Interactions multi sensorielles entre un système robotique multidimensionnel et multi capteur

Ayman Beghdadi

▶ **To cite this version:**

Ayman Beghdadi. Interactions multi sensorielles entre un système robotique multidimensionnel et multi capteur. Automatique. Université Paris-Saclay, 2023. Français. NNT : 2023UPASG076 . tel-04420384

HAL Id: tel-04420384
https://hal.science/tel-04420384

Submitted on 12 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

université PARIS-SACLAY

Interactions multi sensorielles entre un système robotique multidimensionnel et multi capteur

Multi-sensory interactions between a multidimensional and multi-sensor robotic system

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580, sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat : Robotique
Graduate School : Sciences de l'ingénierie et des systèmes
Référent : Université d'Évry Val d'Essonne

Thèse préparée dans l'unité de recherche **IBISC (Université Paris-Saclay, Univ Evry)**, sous la direction de **Malik MALLEM**, Professeur des universités, la co-direction de **Lotfi BEJI**, Maître de conférénce HDR et de **Ali AMOURI**, Enseignant chercheur PCE.

Thèse soutenue à Paris-Saclay, le 23 Novembre 2023, par

Ayman BEGHDADI

Composition du jury

Membres du jury avec voix délibérative

Eric MONACELLI Professeur des universités, Université Paris-Saclay	Président
Jenny BENOIS-PINEAU Professeur des universités, Université de Bordeaux	Rapporteuse & Examinatrice
Adriana TAPUS Professeur des universités, ENSTA Paris	Rapporteuse & Examinatrice
Faïz BEN AMAR Professeur des universités, Sorbonne Université	Examinateur
David FOFI Professeur des universités, Université de Bourgogne	Examinateur

THESE DE DOCTORAT

NNT : 2023UPASG076

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Interactions multi sensorielles entre un système robotique multidimensionnel et multi capteur

Mots clés : Distorsion, Interaction robot-robot, Motion Cueing Algorithm, Perception, Simulateur, SLAM visuel

Résumé : Dans le cadre de simulateur de réalité mixte, la qualité de l'immersion ne peut être évaluée qu'à travers des questionnaires post-simulations faits sur les utilisateurs. Afin de s'affranchir de cette limite, nous proposons un système reproduisant la capacité de l'humain à percevoir son mouvement en vue d'évaluer la qualité de l'immersion de manière qualitative. Dans notre système, l'humain est ainsi remplacé par un robot humanoïde NAO et un modèle de perception visuo-inertiel du mouvement permettant de simuler par biomimétisme la fonction cognitive visuo-spatiale de l'humain. Ce système soulève des problématiques relatives aux traitements des flux d'information visuelle et inertielle réalisés respectivement par le cortex visuel et le système vestibulaire. En conséquence, une méthode de SLAM visuel robuste aux dynamiques de scène est proposée en substitue des tâches réalisées par le cortex visuel. Cette méthode exploite les informations spatiales, sémantiques et d'interactions présentes dans la scène observée dans le but d'atteindre un niveau de robustesse similaire à l'humain. Une base de données contenant des images photo-réalistes dégradées de manière globale et locale a également été produite pour rendre notre méthode de SLAM moins sensible aux phénomènes de distorsions liées aux conditions d'acquisition. Les estimations visuelles et inertielles sont ensuite intégrées à un framework liant notre robot NAO et le modèle de perception utilisé en vue de déterminer le mouvement perçu selon l'humain et ainsi émettre un avis sur la qualité de l'immersion. Ce framework offre alors la possibilité de calibré des scénarios pour le simulateur qui garantissent à la fois une restitution des sensations et l'intégrité physique de l'utilisateur. Enfin, dans l'intention d'optimiser l'immersion, une méthode d'apprentissage par renforcement pour l'optimisation de la restitution des sensations inertielles à travers le *Motion Cueing Algorithm* est proposée. Cette méthode nouvelle permet une meilleure retranscription des sensations inertielles par la plateforme robotique du simulateur.

Title : Multi-sensory interactions between a multidimensional and multi-sensor robotic system
Keywords : Distortion, Motion Cueing Algorithm, Perception, Robot-robot interaction, Simulator, Visual SLAM

Abstract : In mixed-reality simulators, immersion quality can only be assessed by means of post-simulation user questionnaires. To overcome this limitation, we propose a system that reproduces a human's ability to perceive its self-motion so that immersion quality can be assessed qualitatively. In our system, the human is replaced by a NAO humanoid robot and a visuo-inertial motion perception model that simulates human visuo-spatial cognitive function. This system raises issues related to the processing of visual and inertial information flows by the visual cortex and the vestibular system, respectively. Therefore, a visual SLAM method that is robust to scene dynamics is proposed to replace the tasks performed by the visual cortex. This method exploits the spatial, semantic and interaction information present in the observed scene to achieve a human-like level of robustness. A database of globally and locally degraded photorealistic images has also been created to make our SLAM method less sensitive to distortions related to the acquisition conditions. The visual and inertial estimates are then integrated into a framework that links our NAO robot to the perceptual model used to determine the motion perceived by the human and thus provide an opinion on the quality of the immersion. This framework then makes it possible to calibrate scenarios for the simulator that guarantee both the restoration of sensation and the physical integrity of the user. Finally, with the aim of optimising immersion, a reinforcement learning method is proposed to optimise the restitution of inertial sensations through the Motion Cueing Algorithm. This new method enables the simulator's robotic platform to better reproduce inertial sensations.

Table des matières

1 - Remerciements

Je souhaite tout d'abord remercier mon directeur de thèse, Prof. Malik MALLEM, pour son encadrement bienveillant, son aide continue et ses conseils avisés dans le cadre professionnel et également personnel. J'ai eu le privilège de l'avoir comme directeur de thèse, qui par son soutien indéfectible et instantané a permis le bon déroulement de mon travail. À travers nos discussions, j'ai pu mieux appréhender les subtilités du travail de recherche et me conforter dans mes orientations.

Je souhaite également remercier mon co-encadrant de thèse, Prof. Lotfi BEJI, qui a su m'orienter vers la réussite par des remarques et conseils avisés.

J'ai également une attention particulière pour Prof. Ali AMOURI, qui m'a ouvert la voie vers le doctorat auquel à l'origine je ne me destinais pas. Je le remercie pour ses encouragements perpétuels et son énergie communicative de tous les instants qui m'ont permis de me lancer dans cette aventure.

Et enfin, je souhaite remercier tous mes proches, dont en particulier mon père, qui m'ont supporté tout ce temps et m'ont donné la motivation pour réussir. J'ai une attention particulière pour mes filles, qui par leur joie de vivre ont su me donner la force de surmonter les moments difficiles. À ma femme qui m'a épaulé toutes ces années, et sans qui rien n'aurait été possible.

2 - Abstract

The research topic of this thesis, "Multi-sensory interactions between a multi-dimensional, multi-sensor robotic system", is rooted in the rehabilitation of people with motor disabilities through the use of an adapted skiing simulator. The simulator used, a complex system made up of a robotic platform and a virtual reality (VR) scenario, should enable virtual skiing that is sufficiently immersive in terms of visual, inertial and physical sensations. In general, the performance of VR simulators is evaluated by means of subjective post-simulation questionnaires on the quality of the user experience. However, this approach does not allow for the generalisation of the immersion evaluation process, nor does it provide access to numerical data for possible immersion optimisation. These limitations raise questions about the means available to measure and optimise the quality of immersion in terms of movement through the interactions operating in the simulator.

Indeed, it is essential to identify and evaluate the factors and modalities of interaction in our system that affect immersion through the perception of one's own movement. To access this information, we propose a simulation of the human perceptual system, with sensors for organs and algorithms/models for cognitive functions. These cognitive models should support the acquisition of a sense of motion and the evaluation of immersion in a standardised way using biomimicry.

As a result, in our case study, the human has been replaced by the humanoid robot NAO in order to provide access to quantifiable data and to maintain human-like body mobility. The first objective of our work is therefore to design and integrate a system that uses sensor information to simulate the complex processing of sensory information by the human brain that enables it to perceive its own movement. We propose a computer vision algorithm that reproduces the processing of visual information by the visual cortex. However, the complexity and robustness of human visual perception means that our algorithm must be adaptable to real environments, which are complex and dynamic. In this way, this hybrid perceptual model provides an estimate of the sensations perceived by the human cognitive bias. Secondly, these perceived sensations are linked to feedback from a 'virtual sensor', allowing immersion to be assessed on the basis of the difference between the accelerations and velocities applied to the NAO robot and those perceived by our hybrid model. This tool for assessing immersion from a dynamic point of view could be used in the future for simulator calibration.

Finally, preliminary studies we have carried out have shown that the re-

production of acceleration profiles from the scenario by the platform does not allow optimal immersion to be achieved. In fact, the main problem with simulators is the reproduction of the sensation of movement on the user for trajectories from scenarios that are poorly adapted to the simulator workspace. The use of the MCA filter (*Motion Cueing Algorithm*) allows the decomposition of an acceleration profile corresponding to a trajectory that cannot be reproduced by the simulator into a set of acceleration profiles for each of the simulator's motion actors (XY table, translation and rotation at the level of the Gough Stewart platform) that reproduce the sensations induced by the scenario's acceleration profile. Consequently, we are looking for an approach that modulates the simulator's input acceleration profile so that the acceleration perceived by our cognitive model is as close as possible to that derived from the scenario. A reinforcement learning model is proposed to perform this modulation and to ensure the generalisation of this approach for any type of acceleration profile.

3 - Introduction

3.1 . Introduction générale

Au cours des dernières décennies, l'évolution croissante des techniques et technologies dans le domaine de l'informatique, la robotique et l'ensemble de ses domaines sous-jacents ont bouleversé le rapport de l'humain à son environnement. La robotisation et digitalisation de la société dans le cadre industriel comme personnel ont poussé l'humain à intégrer progressivement les nouvelles technologies dans son quotidien. En effet, l'amélioration constante des capacités des ordinateurs et la normalisation de l'utilisation du smartphone ont conduit les industriels à concevoir des algorithmes toujours plus complexes pour effectuer des tâches de haut niveau au quotidien. L'ensemble de ces innovations ont fait émerger des axes de recherches autour des principaux besoins de la population, soit pour l'amélioration de la condition humaine, soit pour l'aide à l'exécution de certaines tâches complexes du quotidien.

La robotique classique assimilée à la production industrielle à travers les automates s'est progressivement orientée vers la robotique autonome avec deux principales réalisations, la voiture [KW17 ; Van+18] et le drone [KM12 ; Men+17] autonomes. Chacun de ces véhicules autonomes ouvrant un nouveau champ de possibilités et de défis pour la société. Leur développement a été rendu possible grâce à l'amélioration des capteurs et actionneurs ainsi que des méthodes de vision par ordinateur permettant une perception de l'environnement dans lequel ces robots évoluent. En parallèle, le déploiement croissant des systèmes embarqués, IoT et applications mobiles ont permis l'essor des méthodes de vision par ordinateur intégrant davantage l'apprentissage automatique pour effectuer des tâches complexes de traitement d'image et de compréhension de scène.

La diversité des applications [Tor12 ; Gao+18a ; Sze22] issues de ces nouvelles méthodes telles que la vidéo-surveillance [OS15 ; Shi+19], la reconnaissance d'objets et d'actions, l'imagerie médicale et la reconstruction 3D, pour ne citer que les principales, ont fait émerger de nombreux axes de recherche en robotique, vision par ordinateur, intelligence artificielle et réalité virtuelle. Les progrès majeurs des performances des unités de traitement graphique (GPU) ont offert un renouveau à des technologies relativement anciennes, plus particulièrement l'apprentissage profond et la réalité virtuelle, ne disposant pas de la puissance de calcul nécessaire à l'époque pour atteindre un niveau d'efficacité suffisant pour leur déploiement. La nature du fonctionne-

ment de l'apprentissage profond couplée à l'utilisation croissante de données a permis l'émergence d'un vaste champ d'applications [DY+14 ; Naj+15 ; SS18] qui touchent l'ensemble des domaines de production de la société (finance, marketing, automatisation de procédé, etc...). La réalité mixte, qui est une technologie basée essentiellement sur l'informatique et de vision par ordinateur, simule un environnement virtuel, autrement dit artificiel, pour y intégrer un utilisateur réel en immersion à travers des stimulus sensoriels essentiellement basés sur la vision, le toucher et l'ouïe. Cette technologie, récemment remise sur le devant de la scène par les acteurs majeurs des nouvelles technologies [Wei22], introduit des challenges et domaines d'application [MNB14 ; Wex14 ; SC18] tels que le divertissement [CFP18], l'apprentissage interactif [Gre+17 ; Rad+20 ; Xie+21], les simulateurs [VGD16 ; OD17 ; Neu+18], les technologies médicales [Mog+16 ; Li+17 ; JH20] et la réhabilitation [SR01 ; Kim+05 ; Ros+11 ; How17].

Les secteurs de la technologie médicale et de la santé sont en plein essor grâce aux continuelles innovations des domaines de recherche cités précédemment. Ce secteur gagne sans cesse des parts de marché dans la recherche et développement en raison de l'intérêt sociétal qu'il suscite. L'émergence de nouvelles technologies dans la santé [Kyr+21] a fortement amélioré la prise en charge des maladies, accidents et soins apportés aux patients. Dans le cadre de la robotique médicale, la chirurgie robot-assistée [Pet+18 ; Pet+18 ; Pan+19] a offert une solution prometteuse pour la généralisation de l'aide aux interventions chirurgicales ou même à la télé-opération. Cette solution joint des techniques de haute technologie issues des domaines de robotique, automatique, apprentissage profond et traitement du signal.

Dans un même temps, l'activité de l'imagerie médicale est en progression constante grâce à la fois aux nouvelles techniques d'imagerie résultant de système d'acquisition plus performant, mais également aux méthodes de traitement des images [Gao+18a ; MBM16] permettant une amélioration de l'extraction et caractérisation des informations. L'intégration croissante de l'apprentissage profond dans les solutions d'imagerie médicale [RNZ18 ; Anw+18] a permis d'augmenter la fiabilité des informations extraites et des analyses produites. Ces procédés et méthodes fournissent une aide au diagnostic pour les spécialistes de la santé afin de détecter de nombreuses maladies, en particulier pour le cancer qui reste à ce jour la principale cause de mortalité.

Les applications de réalité virtuelle se sont développées ces dernières années [How17 ; Tie+18 ; RNC18] pour la réhabilitation de personnes victimes de maladies ou accidents engendrant des situations de handicap mental ou physique. Ces applications prennent la forme d'environnements virtuels si-

mulés dans le cadre de réhabilitation pour des troubles mentaux [KK20] en exploitant souvent uniquement les équipements rudimentaires du domaine comme les casques de réalité virtuelle ou les joysticks. Pour la réhabilitation des capacités motrices, les méthodes proposées [Lav+17 ; Far+18 ; Aye+19] peuvent incorporer, en plus des casques et joysticks, une plateforme ou un système robotique pour augmenter l'immersion à travers le toucher, et surtout garantir une exécution contrôlée d'exercices gestuels satisfaisant des critères de sécurité corporelle et d'efficacité de la rééducation.

Enfin, le développement croissant de la robotique intégrant l'humain et la réalité virtuelle a conduit à étudier les interactions homme-robot (IHR) dans les systèmes robotique multimodaux. Rappelons que l'interaction correspond à l'action mutuelle et réciproque de plusieurs acteurs (robotique ou humain) dans un système multimodale où des informations de nature différente sont échangées. Des travaux de synthèse ont été publiés [JS07 ; GS+08 ; Tur14] pour régulièrement mettre à jour l'état de l'art. D'autres introduisent le principe de l'interaction [DLO09 ; Bar+20] ou soulignent les principaux challenges et résultats [De +08 ; She16 ; KS20 ; Fio+20]. De manière générale, les travaux de recherches sont focalisés sur l'amélioration de la capacité de perception sensorielle et cognitive, et de la conception mécanique et algorithmique des systèmes robotiques.

3.2 . Contexte

Dans le cadre de cette thèse, nous nous sommes appliqués à l'étude des interactions multi-sensorielles au sein d'un système robotique multidimensionnel et multi-capteur. Ce sujet de recherche s'ancre dans une thématique de réhabilitation de personnes en situation de handicap moteur au niveau des membres inférieurs dans le but de fournir un outil de rééducation à travers les sports de glisse, et plus spécifiquement le ski. Cette rééducation est rendue possible par l'utilisation d'un système de simulation mixte liant un scénario de réalité virtuelle et une plateforme robotique réelle. Le simulateur proposé s'insère dans un contexte de descente de ski où le scénario et les mouvements de la plateforme tentent de restituer au mieux les sensations d'immersion résultant des trajectoires réelles et virtuelles via les ressentis kinesthésique et inertiel, et la perception visuelle. La sensation d'immersion selon le critère de perception du mouvement induit à l'utilisateur, aussi appelé perception du mouvement propre, lors de la simulation, est rendue possible à travers le ressenti inertiel des accélérations par le système vestibulaire de l'oreille interne. L'emploi d'un scénario virtuel introduit un ressenti fondé sur la vision à travers la perception du mouvement par le système visuel humain, à savoir l'œil et le cortex visuel. Dans cette étude, le scénario virtuel intègre des exigences

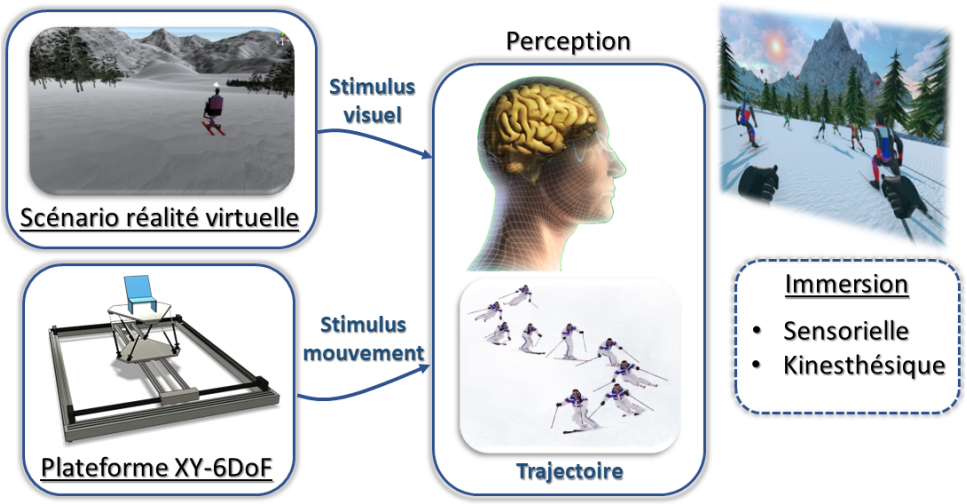


Figure 3.1 – Simulateur de réalité virtuelle pour la réhabilitation.

de réalisme en reproduisant au mieux les conditions de ski, par la prise en compte des phénomènes physiques, et le réalisme des trajectoires générées et de l'environnement visuel créé.

Une plateforme robotique développée au sein du laboratoire nommée XY-6DoF est utilisée pour les expérimentations de simulation. Entièrement conçue par l'équipe de recherche du laboratoire et étudiée auparavant à travers des travaux portant sur sa modélisation et paramétrisation puis son contrôle [Hou+19c ; Hou+19a], cette plateforme offre un accès aux contenus bas et haut niveaux de la plateforme pour une prise en main optimale. La configuration hybride de celle-ci, associe un robot série sous la forme d'une table XY (2 degrés de liberté) avec une plateforme de Gough-Stewart (6 degrés de liberté) de type robot parallèle composée de six actionneurs linéaires pour atteindre 8 degrés de motorisation pour l'ensemble du système (voir figure 3.2). L'association de ces deux types de robot garantit une redondance du mouvement découplant ainsi les effets d'accélérations et la zone de travail de la structure robotique. La partie inférieure de la plateforme, correspondant à la table XY, est allouée à l'exécution des mouvements d'accélération élevée et de haute fréquence. La partie supérieure, relative à la plateforme de Gough-Stewart que nous appelons nacelle, réalise les mouvements de rotation et/ou d'accélération moindre et de plus basse fréquence.

Afin d'évaluer la qualité de la retransmission des sensations de mouvement, il est nécessaire d'effectuer une analyse des sensations de l'humain dûs aux retours d'information extéroceptifs et proprioceptifs par rapport aux

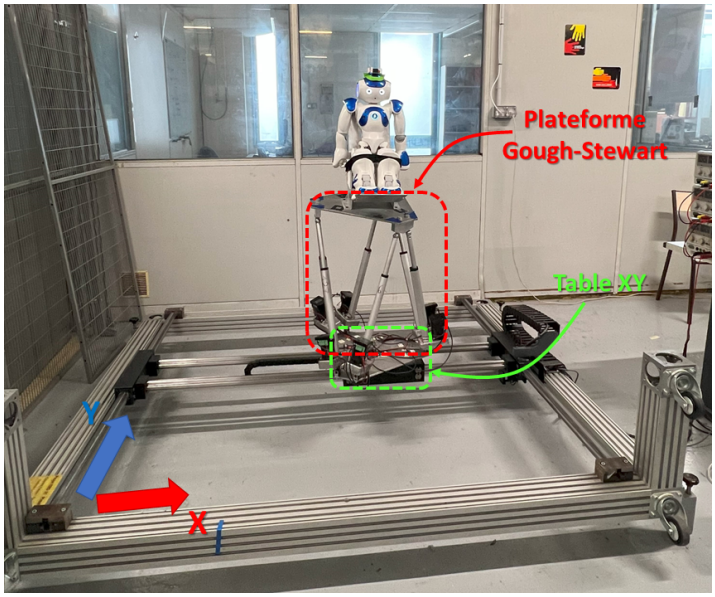


Figure 3.2 – Plateforme robotique XY-6DoF.

différentes trajectoires réalisées par la plateforme hybride. D'un point de vue physiologique, la perception du mouvement de soi (mouvement propre) à travers les ressentis, s'effectue chez l'être humain grâce au système vestibulaire de l'oreille interne, et aux récepteurs musculaires et ligamentaires. Les informations visuelles sont traitées de manière plus profonde et précise offrant davantage une "estimation" qu'un "ressenti" du mouvement. De manière générale, l'humain utilise trois types d'informations :

- Les informations visuelles : provenant du système visuel humain qui inclut la rétine et le système sensori-moteur.
- Les informations inertielles : issues du système vestibulaire de l'oreille interne.
- Les informations kinesthésiques/proprioceptives : qui permettent la perception consciente ou non du mouvement du corps à travers des organes proprioceptifs tels que les muscles, articulations et tendons qui réagissent à leurs propres "réactions" au mouvement.

Comme illustré dans la figure 3.3, l'ensemble des informations issues de la simulation sont captés par les organes sensoriels ou proprioceptifs respectivement adaptés à la nature de l'information (visuelle, inertielle, articulaire, etc.). Une perception du mouvement global (accélération, vitesse, orientation et position du corps dans sa globalité) et des positions relatives des différents membres du corps (articulations, squelette) est déduite des ressentis provenant des différentes sensations corporelles et sensorielles. L'intelligence humaine et son expérience parviennent à déduire le mouvement propre de manière naturelle à travers l'analyse de ces sensations plus ou moins abstraites.

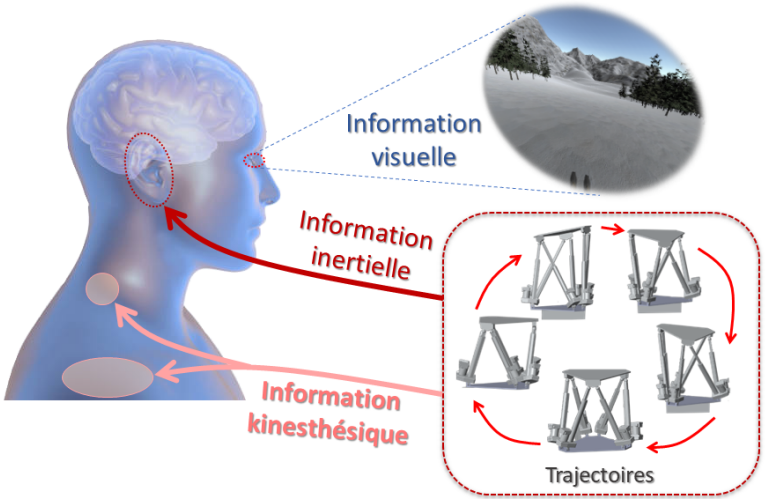


Figure 3.3 – Relation entre source d'information et organes sensoriels ou proprioceptifs.

De fait, ces informations sont inaccessibles lors de simulation avec l'humain hormis par des questionnaires post-simulation qui de plus ne fournissent pas de données à exploiter. En conséquence, l'humain a été substitué par un robot humanoïde NAO présentant l'avantage d'avoir une morphologie similaire à l'homme et de posséder de nombreux capteurs. La structure du robot NAO conçue sur le schéma du squelette humain garantit une restitution assez fidèle des effets du mouvement de la plateforme sur le corps. En outre, la présence des nombreux capteurs, dont en particulier articulaires et inertiels, permettent une estimation précise du mouvement de son squelette dans l'espace et au cours du temps. Pour satisfaire le contexte de descente de ski pour personne en situation de handicap moteur, la nacelle de la plateforme a été adaptée par l'ajout d'un siège pour correspondre au handiski et accueillir le robot NAO (voir figure 3.4). Le cadre d'application lié au handicap et la réhabilitation exige le respect de conditions sécuritaires et contextuelles impactant à la fois le contenu virtuel du scénario et réel du système robotique. En effet, la vulnérabilité des personnes en situation de handicap amplifiée par la pratique du ski, qui est de fait dangereux, impose des contraintes de simulation :

- Les membres inférieurs du robot NAO doivent être immobilisés et fixés au siège pour satisfaire les critères de handicap moteur des membres inférieurs et de pratique du ski avec handiski (voir figure 3.4).
- Le scénario virtuel doit contenir un environnement adapté aux personnes à mobilité réduite (pente faible, peu d'obstacles, etc...) et donc des trajectoires sécurisées adaptées à leur capacité.

Figure 3.4 – Homothétie du robot NAO sur siège et de l'handiski.

La thématique plurielle du sujet de recherche axée sur l'interaction combinée au contexte et aux conditions d'application apporte un ensemble de problématiques complexes. Notons également que la diversité des domaines de recherche liée au sujet offre un large choix d'orientations de travaux de recherche pour l'amélioration de l'existant.

3.3 . Motivations

Le sujet de recherche de cette thèse, "Interactions multi sensorielles au sein d'un système robotique multidimensionnel et multi capteur", s'ancre dans un contexte de réhabilitation de personne en situation de handicap moteur grâce à l'utilisation d'un simulateur de ski adapté. Le simulateur utilisé, un système complexe composé d'une plateforme robotique et d'un scénario de Réalité Virtuelle (RV), doit permettre une pratique de ski virtuel qui soit suffisamment immersive au niveau des ressentis visuels, inertiels et corporels. De manière générale, les performances des simulateurs de RV sont évaluées par des questionnaires post-simulation subjectifs portant sur la qualité d'expérience utilisateur. Cependant, cette approche ne permet ni une généralisation du processus d'évaluation de l'immersion, ni l'accès à des données numériques pour une possible optimisation de l'immersion. Ces limitations soulèvent des problématiques quant aux moyens disponibles pour mesurer et optimiser la qualité de l'immersion sur le plan du mouvement à travers les interactions opérant au niveau du simulateur.

En effet, il est primordial d'identifier et évaluer les facteurs et modalités d'interaction dans notre système affectant l'immersion à travers la perception du mouvement propre. Pour accéder à ces informations, nous proposons une

simulation du système de perception humain, capteurs pour organes et algorithmes/modèles pour fonctions cognitives. Ces modèles cognitifs doivent offrir un support pour l'acquisition de ressenti du mouvement et l'évaluation de l'immersion de manière standardisée par biomimétisme comme illustré dans la figure 3.5.

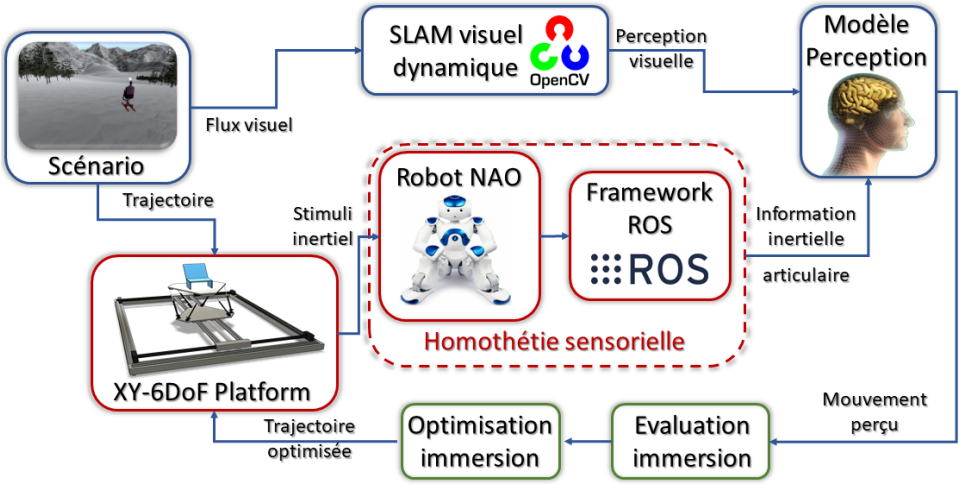


Figure 3.5 – Architecture globale de notre système.

En conséquence, l'humain a été substitué par le robot humanoïde NAO dans notre cas d'étude pour offrir un accès à des données quantifiables et conserver une mobilité corporelle proche de celle de l'humain. Dans un premier temps, l'objectif de notre travail est donc de concevoir et intégrer un système qui exploite les informations capteurs pour simuler les traitements complexes par le cerveau humain des informations sensorielles permettant la perception de son mouvement propre. Nous proposons un algorithme de vision par ordinateur reproduisant le traitement de l'information visuelle par le cortex visuel. Néanmoins, la complexité et la robustesse de la perception visuelle humaine impose à notre algorithme une adaptabilité à des environnements réels qui sont complexes et dynamiques. Cet algorithme de vision est ensuite associé à un modèle de perception inertiel [HCB11] afin de garantir la restitution des interactions visio-vestibulaires lors de la perception du mouvement propre. Ainsi, ce modèle de perception hybride produit une estimation des sensations perçues par le biais cognitif humain. Dans un second temps, ces sensations perçues sont apparentées à des retours d'information de "capteur virtuel" permettant d'évaluer l'immersion à partir de l'écart entre les accélérations et vitesses appliquées au robot NAO et celles perçues à travers notre modèle hybride. Cet outil d'évaluation de l'immersion d'un point de vue de la dynamique pourra à l'avenir être exploité pour la calibration de simulateur.

Enfin, des études préliminaires que nous avons menées ont montré que la reproduction, des profils d'accélérations provenant du scénario, par la plateforme ne permettait pas d'atteindre une immersion optimale. En effet, la problématique principale des simulateurs est la reproduction des sensations de mouvement sur l'utilisateur pour des trajectoires provenant de scénarios qui sont inadaptées à l'espace de travail des simulateurs. L'utilisation du filtre MCA (*Motion Cueing Algorithm*) permet la décomposition d'un profil d'accélération correspondant à une trajectoire non-reproductible par le simulateur en un ensemble de profils d'accélération pour chacun des acteurs de mouvement du simulateur (table XY, translation et rotation au niveau de la plateforme de Gough Stewart) qui restitue les sensations induites par le profil d'accélération du scénario. En conséquence, nous cherchons une approche qui module le profil d'accélération d'entrée du simulateur pour que l'accélération perçue par notre modèle cognitif soit la plus proche possible de celle issue du scénario. Un modèle d'apprentissage par renforcement est proposé pour effectuer cette modulation et garantir une généralisation de cette approche pour tout type de profil d'accélération.

3.4 . Contributions et publications

3.4.1 . Contributions

Ce travail de thèse a pour objectif l'étude des interactions multi sensorielles entre un système robotique multidimensionnel et multi capteur opérant dans le cadre d'un simulateur de réalité mixte pour la pratique de ski en vue de la réhabilitation de personnes en situation d'handicap moteur. Dans ce contexte, un modèle cognitif a été implémenté pour servir d'outil d'évaluation de la qualité d'immersion proposée par le simulateur et de retour d'information en vue d'optimiser l'immersion. Afin de mener cette étude, nous avons apporté les contributions suivantes :

1. Tout d'abord, une étude des principaux concepts et de l'état de l'art du SLAM visuel a été menée afin de déterminer un axe de recherche pertinent pour notre sujet d'étude et le domaine du SLAM. En conséquence, un algorithme de SLAM visuel robuste aux dynamiques de scène a été proposé dans le but de reproduire les capacités cognitives humaines relatives aux fonctions visuo-spatiales. Cette méthode offre une solution performante et fiable pour la localisation et la cartographie d'environnement dynamique en prenant en compte les éléments de la scène en mouvement afin de réduire l'accumulation d'erreurs d'estimation du mouvement de la caméra. Des raisonnements géométrique et d'apprentissage profond sont combinés à une étude des interactions objet-objet présentes dans la scène afin d'estimer l'état des points d'intérêt (statique/dynamique) utilisés dans le processus de SLAM. Cette ap-

proche innovante permet de prendre en considération davantage de points par rapport aux méthodes de l'état de l'art et de rejeter les points considérés dynamiques. L'algorithme développé a été évalué sur la base publique TUM RGBD [Stu+12] dédiée à cet effet, permettant de démontrer la supériorité de notre approche [BM22; BMB22b; AML23].

2. Le contexte d'application de notre étude a mis en évidence la présence de perturbations photométriques liées au mouvement de la plateforme. Une baisse des performances du modèle d'apprentissage profond réalisant la segmentation d'objet dans l'algorithme de SLAM a été observée dans ces situations, ce qui a conduit à chercher une méthode permettant d'améliorer la robustesse du modèle contre les distorsions d'images. L'approche basée sur l'augmentation de donnée a été privilégiée en raison de sa simplicité de mise en place. Cependant, afin de contribuer au domaine et d'améliorer l'apport de cette approche pour notre modèle de segmentation, nous avons développé un nouveau type de distorsion, dit localisé, afin de fournir des distorsions davantage photoréalistes. Une première base de données a été créée à cet effet, contenant l'ensemble des distorsions habituellement présentes dans les bases de données perturbées ainsi que nos distorsions localisées appliquées de manière locale sur les images. L'apport de cette approche a été évalué par un benchmarking réalisé sur les principaux modèles de détection d'objet, révélant l'efficacité de l'augmentation de donnée par l'utilisation d'image perturbé localement et globalement [BMB22a]. Une seconde base de données dérivée de la première a été produite en incorporant le contexte de scène dans la génération des distorsions. La nature de la scène et des objets, et leurs orientations et proximités par rapport à la caméra ont été considéré pour appliquer des distorsions locales et globales, plus réalistes en termes de nature, magnitude, orientation et répartition dans l'image. Cette base pourra servir pour l'entraînement de modèles appliqués à la vision par ordinateur, réalisant la détection et/ou segmentation d'objet, et l'estimation de pose d'individu, qui soient robustes aux distorsions. (Accepté à EUVIP)

3. Une étude approfondie des capacités cognitives humaines a été réalisée pour identifier les fonctions cognitives relatives à la perception du mouvement. En outre, la configuration de notre système robotique nous a amené à identifier les formes d'interaction Homme-Robot multimodale afin de spécifier les informations majeures du système. En parallèle, les seuils de perception vestibulaires humain ont été extraits de travaux de référence dans le domaine et les seuils de perception articulaire humaine ont été déterminés à l'aide d'une étude expérimentale menée sur une vingtaine d'individus. L'ensemble de ces seuils étant indispensable à l'intégration des limites humaines dans le modèle de per-

ception du mouvement propre proposé. Ce modèle basé sur les informations visuelles et vestibulaires, et considérant le mouvement corporel dû aux sensations articulaires, a intégré l'algorithme de SLAM visuel dynamique pour restituer les capacités d'observation visuelle humaine (Soumission ROBIO).

4. Un environnement de simulation réaliste et complexe a été conçu, intégrant un scénario de descente de ski en réalité virtuelle, une plateforme robotique reproduisant les accélérations du scénario et un robot humanoïde servant de substitut à l'humain et imitant les fonctions cognitives humaines grâce au modèle de perception proposé. Le modèle cognitif reproduisant les aptitudes humaines de perception du mouvement propre a été incorporé dans un framework d'évaluation de la qualité de l'immersion pour les simulateurs de réalité mixte. Partant du constat que la sensation d'immersion était liée aux ressentis du mouvement d'un point de vue inertiel et visuel, le framework proposé calculait l'écart entre les accélérations et vitesses perçues à travers le modèle cognitif et celles réelles issues des trajectoires de la plateforme et du scénario de réalité virtuelle. Enfin, un modèle d'apprentissage par renforcement a été développé pour optimiser l'immersion en réduisant l'écart de ressentis visuel et inertiel grâce à la modification des profils d'accélération de la plateforme (en cours de réalisation).

3.4.2 . Publications

Dans le cadre de mes travaux de recherche, mes publications se résument par :

- [BM22] Beghdadi, A., Mallem, M. (2022). A comprehensive overview of dynamic visual SLAM and deep learning : concepts, methods and challenges. Machine Vision and Applications, 33(4), 54.
- [BMB22a] Beghdadi, A., Mallem, M., Beji, L. (2022, October). Benchmarking performance of object detection under image distortions in an uncontrolled environment. In 2022 IEEE International Conference on Image Processing (ICIP) (pp. 2071-2075). IEEE.
- [Hou+23a] Houda, T., Beghdadi, A., Beji, L., Amouri, A. (2023, February). Complex Multi Robot Hybrid Platform for Augmented Virtual Skiing Immersion. In 3rd IFSA Winter Conference on Automation, Robotics & Communications for Industry 4.0/5.0 (ARCI 2023).
- [AML23] Ayman, B., Malik, M., & Lotfi, B. (2023). DAM-SLAM : depth attention module in a semantic visual SLAM based on objects interaction for dynamic environments. Applied Intelligence, 1-14.
- [Hou+23b] Houda, T., Beghdadi, A., Beji, L., & Amouri, A. (2023). Multi-Complex Robot Interaction with Homothetic Human Feedback in a Virtual Environment. Sensors & Transducers, 260(2), 15-24..

- [Beg+23] Beghdadi, A., Beghdadi, A., Mallem, M., Beji, L., Alaya Cheikh, F. (2023). CD-COCO : A Versatile Complex Distorted COCO Database for Scene-Context-Aware Computer Vision. European Workshop on Visual Information Processing (EUVIP).

Les articles suivants sont soumis :

- Beghdadi, A., Houda, T., Beji, L., Mallem, M., Amouri, A. (2023).Human Homothetic Perception Framework Based on Robot multi-modal Interaction with Mixed Reality. IEEE International Conference on Robotics and Biomimetics (IEEE ROBIO 2023).

3.5 . Organisation de la thèse

La thèse est organisée de la manière suivante :

- Le chapitre 1 introduit le sujet et le contexte afin de présenter les motivations de ce travail et les publications résultantes.
- Le chapitre 2 détaille l'état de l'art des méthodes d'interactions Homme-Robot de manière générale pour exposer ses concepts et modalités, et plus particulièrement pour la réhabilitation. Un court état de l'art des simulateurs est également proposé pour identifier les plateformes robotiques existantes semblables à notre système.
- Le chapitre 3 présente les contributions dans la partie vision de notre sujet de thèse à travers notre méthode de SLAM visuelle dynamique. L'algorithme de SLAM visuel y est décrit et évalué sur une base de données dédiée à cet effet dans le domaine. Ses limitations sont exposées et une solution est décrite dans le chapitre suivant.
- Le chapitre 4 est dédié à la description des travaux sur l'amélioration de la robustesse des modèles de segmentation à travers la génération de distorsions photo-réalistes pour une augmentation de données. L'impact de cette augmentation de données est évalué à la fois pour l'amélioration de la robustesse du modèle de segmentation et de notre algorithme de SLAM.
- Le chapitre 5 est dédié à la description du framework réalisant l'évaluation de l'immersion lors des simulations à l'aide du modèle de perception, et la présentation de la méthode d'optimisation d'immersion via l'utilisation du *Motion Cueing Algorithm*.
- Le chapitre 6 expose l'environnement de simulation utilisé lors de notre thèse ainsi que les expérimentations menées et les résultats obtenus. Ce chapitre regroupe des explications sur l'ensemble des composants du simulateur, à savoir, les caractéristiques de la plateforme robotique, du scénario de ski en réalité virtuelle et des interactions entre environnements réels et virtuels.

- Le chapitre 7 conclut ces travaux de recherche et offre des perspectives quant à la démonstration de l'apport de l'approche pour l'amélioration des simulateurs et l'intégration de modalités supplémentaires pour une interaction plus forte.

4 - Interaction Homme-Robot et simulateurs

4.1 . Introduction

La nature de notre système de simulation composé de robots et d'humain (d'un point de vue conceptuel avant sa substitution par le robot NAO) soulève de fait des interrogations à propos des Interactions Homme-Robot (IHR). Il est ainsi primordial de définir le principe, les modalités et les cadres d'application des interactions Homme-Robot dont une illustration est donnée dans la figure 4.1. Dans ce chapitre, le principe de l'IHR et ses différentes variantes en fonction de sa configuration seront détaillés. Afin de mieux appréhender les

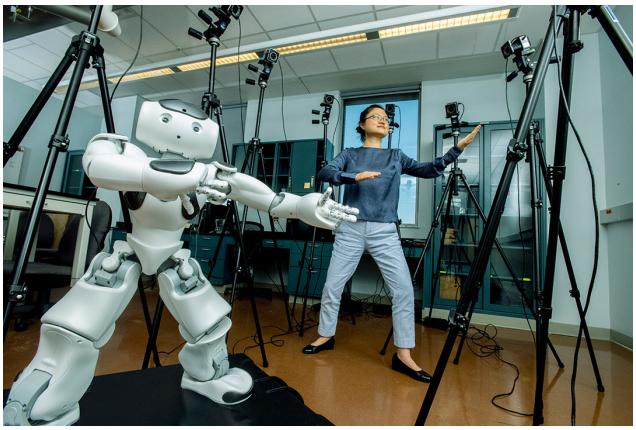


Figure 4.1 – Système robotique d'interaction Homme-Robot.

enjeux des interactions étroitement liées aux capacités de perception et raisonnement humaines, l'ensemble des fonctions cognitives seront détaillées. Également, les niveaux d'autonomie des systèmes robotiques seront énoncés pour mettre en avant l'impact de l'autonomie sur le type et le degré d'interaction qui en résulte. De même, les différentes sources d'information et leurs différentes associations seront exposées de manière à définir le concept et les enjeux de l'interaction multimodale. Rappelons que notre sujet de recherche s'ancre dans un contexte de réhabilitation d'une personne présentant un handicap moteur. Ainsi, les modalités des travaux d'IHR pour des applications d'entraînement, d'assistance et de réhabilitation seront énumérées pour en comprendre les problématiques et enjeux. Une attention particulière sera portée à l'étude de l'état de l'art des méthodes de réhabilitation basées sur des interfaces (ou simulateur) intégrant l'IHR. Pour compléter cette étude, l'état de l'art des simulateurs conçus avec une configuration robotique proche de la plateforme de Gough-Stewart ou similaire sera fourni pour mieux appréhender l'évolution et les singularités de ce type de plateforme.

4.2 . Interaction Homme-Robot

L'Interaction Homme-Robot (IHR) est le domaine d'étude des relations existantes entre des systèmes robotiques utilisés par ou en collaboration avec l'humain. L'interaction est donc une mise en relation d'humains et robots à travers la transmission d'informations ou de stimulus créant une influence mutuelle ou d'un opérateur sur l'autre. De fait, il existe plusieurs catégories d'interaction en fonction de la localisation des opérateurs l'un par rapport à l'autre (interaction à distance/proximité) et du type d'application requérant une manipulation physique, de la mobilité ou de l'interaction sociale. Les cadres d'application de l'interaction en fonction de la catégorie d'interaction et du type de système robotique sont illustrés dans la figure 4.2

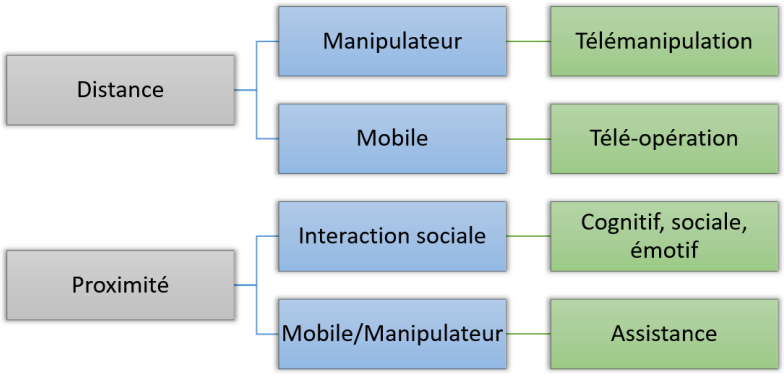


Figure 4.2 – Cadre applicatif de l'interaction Homme-Robot.

Notons que la téléopération est également appelée contrôle de supervision et que l'interaction sociale est l'échange entre humains et robots en tant que pairs ou partenaires pour des aspects cognitifs (apprentissage, stimulation), sociales (création de lien) ou émotifs (thérapie pour le bien-être). Le domaine de recherche de l'IHR a émergé au cours des années 90 où ses fondements ont été établis à travers une approche multidisciplinaire composée de robotique, sciences cognitives et psychologie. Ce domaine a émergé sous des formes d'interfaces diverses (voir figure 4.3) du fait du développement croissant de la robotique au cours de cette période, en particulier de la robotique autonome, l'algorithmie comportementale et des plateformes robotiques.

Dans un premier temps, des robots mobiles pour téléopération ont été conçus pour des missions exploratoires essentiellement par des agences spatiales telles que la NASA [FT01; Dif+03]. Ces robots mobiles de configurations diverses, humanoïdes ou motorisées, ont permis l'émergence de domaines de recherche parallèles axés sur le développement de méthode permettant la mise en mouvement autonome et la retranscription du raisonnement humain. La robotique humanoïde regroupe les robots autonomes de structure

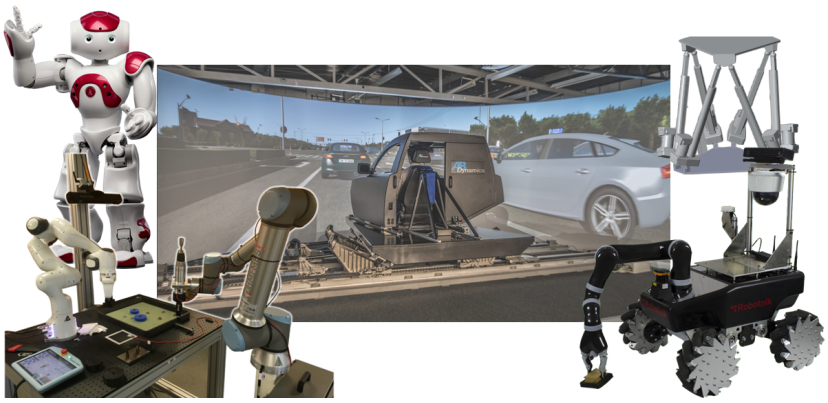


Figure 4.3 – Interfaces d'interaction Homme-Robot.

proche du squelette humain [Hir+98] qui intègre des algorithmes imitant les comportements, raisonnement et capacités de mouvement de l'humain [Sch99; Atk+00; Ozt+05; WLJ08; PPD16]. Cette discipline s'est fortement développée au cours des dernières décennies proposant des solutions de téléopération [SNK08; GCB13; AM13; Abi+18] et dernièrement une autonomisation dynamique des robots [Tan+18; RKD18; Dag20; Sta22] grâce à l'intelligence artificielle intégrant une solution de mimétisme des capacités humaines. Une branche secondaire a également vu le jour sur la téléopération appliquée à des bras robotiques pour des applications possibles dans le secteur médical et industriel. En parallèle, les travaux sur la mise en mouvement autonome se sont focalisés sur la perception de l'environnement et le contrôle des systèmes robotiques pour leur déplacement dans des environnements inconnus.

L'algorithmie comportementale [HR77; Pap11] consiste en l'intégration de boucles de réponse adaptées au sens des stimulus externes auquel est soumis le système. Le développement au cours du temps d'architectures toujours plus complexes a permis d'atteindre un niveau de reproduction de qualité des raisonnements cognitifs et des comportements réactifs humains garantissant une interaction Homme-Robot. Deux domaines majeurs résultent de la robotique comportementale, la science comportementale anthropologique (sociale) et la science cognitive (perception, émotion, etc...). Le premier domaine tente de développer des comportements sociaux réalistes intégrables à des robots pour satisfaire des conventions sociales établies pour différents cadres applicatifs [BF04; Dau07; KM14; Kl17]. Le second domaine tente de retranscrire les mécanismes cognitifs à l'origine des capacités de raisonnement et perception humaines. Ce domaine en plein essor est porté par l'intérêt grandissant pour les sciences cognitives et le développement de l'intelligence artificielle permettant conjointement la génération de modèles cognitifs très réaliste [MFT01; LL08; Wan10; LJO13; WBK17].

Les plateformes robotiques ont eu un intérêt conséquent dans le domaine spatial et industriel au cours du 20^{eme} et 21^{eme} siècles [ADV93 ; KH95 ; Che+01 ; Gra+01 ; OD17]. Ces systèmes robotiques ont souvent été développés sous la forme de simulateur permettant d'évaluer les aptitudes humaines lors de la conduite de véhicules terrestre et aérien [HV76 ; Hee+05]. Ces études permettant ensuite d'optimiser les véhicules d'un point de vue mécanique et contrôle pour garantir un meilleur confort de conduite et de ressenti. En outre, l'essor des IHR a permis d'intégrer des interfaces de plus en plus complexes dans les plateformes robotiques pour garantir une interaction davantage dynamique. De nombreuses interfaces d'IHR ont vu le jour pour proposer des solutions performantes pour la réhabilitation et l'assistance [ST14 ; Bec+17] de personne. Parmi ces interfaces de réhabilitation à l'aide d'interaction homme-robot, les exosquelettes [Rah+15 ; Amb+17], les systèmes de mobilité [Dev+17 ; SH17 ; ARO17 ; Vai+20b ; Ort+21] et les plateformes de rééducation [San+16 ; Bec+18 ; Bin+19] sont les plus répandues.

Il est important de noter que toutes les applications de robot n'ont pas les mêmes degrés et formes d'interaction. En effet, un système télé-opéré est supervisé par un humain qui contrôle ou rectifie le comportement du système de manière ponctuelle lorsque la situation le requiert. Ce type d'interaction a été largement développé dans le domaine industriel et dans les premiers travaux de recherche en IHR [NS91 ; Kof+05 ; HS06]. Pour les systèmes entièrement autonomes, l'interaction est plus subtile car indirecte. Elle consiste en une supervision conceptuelle du robot à travers la formulation d'objectifs et contraintes que le robot devra prendre en compte pour trouver lui-même la solution. Un exemple récent est la recherche sur les robots collaboratifs [FTB03 ; Che+16b ; Hen+19 ; Zac+20] qui interagissent de manière autonome avec un humain ou un autre robot lors de l'exécution d'une tâche commune ou disjointe.

Ces champs de recherche lèvent un certain nombre de problématiques qui permettent de modéliser les interactions entre humains et robots quel que soit leur type d'application ou leur forme d'interaction. Ces problématiques offrent la possibilité de concevoir et évaluer des systèmes d'interaction pour atteindre des objectifs établis à travers des critères connus :

- L'autonomie du système
- Le niveau de collaboration
- La formulation et l'exécution de l'objectif
- La forme des informations
- La forme d'apprentissage

La conception d'interfaces IHR s'applique à définir explicitement l'ensemble de ces critères afin d'optimiser les échanges humains-robots et ainsi atteindre l'objectif souhaité. La recherche en IHR est un vaste domaine, en conséquence, nous nous restreignons uniquement au détail des concepts en lien avec notre sujet d'étude. Pour cela, l'état de l'art décrit dans la section 4.2.5 sera uniquement alloué aux méthodes d'interface Homme-Robot destinées à la réhabilitation et l'assistance de personne en situation de handicap.

4.2.1 . Fonctions cognitives

Les fonctions cognitives sont l'ensemble des capacités du cerveau humain permettant d'interagir avec son environnement en analysant l'information, raisonnant et s'adaptant. Ces fonctions ont été étudiées au cours de nombreux travaux [BP49 ; Boy98 ; JCL06] où les principales tâches et objectifs correspondants ont été décris et analysés. Les fonctions cognitives peuvent être réunies en sept domaines cognitifs :

- L'attention
- La mémoire et l'apprentissage
- Le langage
- Les fonctions exécutives
- La cognition sociale
- La perception
- Les fonctions motrices

Une représentation sous forme de carte est proposée dans la figure 4.4 pour illustrer les principales fonctions cognitives et leurs tâches et/ou objectifs respectifs.
L'attention est le processus permettant de focaliser nos capacités mentales pour un raisonnement sur certains côtés définis de l'environnement qui contiennent les informations les plus utiles à la déduction d'une solution, ou l'exécution d'une action adaptée à la situation. L'attention est donc la faculté de gérer la concentration nécessaire pour le traitement adéquat de l'information et le bon déroulement du raisonnement. En tout, cette fonction cognitive se présente sous quatre formes différentes : **Attention sélective** pour la focalisation sur l'information nécessaire, **Attention soutenue** pour la capacité de concentration sur un temps long, **Attention alternée** pour la gestion de l'orientation de la concentration et **Vitesse de traitement** pour la vitesse d'exécution du cerveau.

La mémoire et l'apprentissage sont les facultés de codifier et de stocker une information ou un événement puis à le récupérer en fonction des besoins et situations. Il existe une distinction entre les deux types de mémoire :

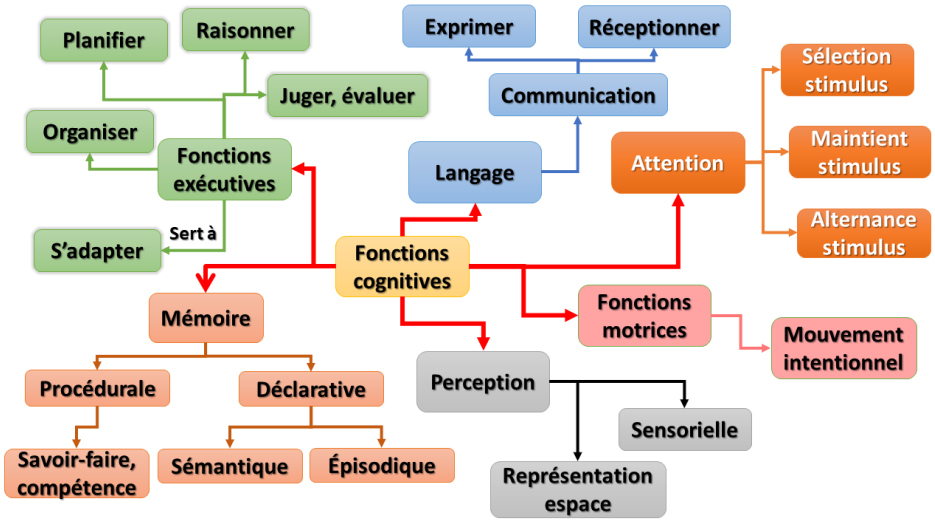


Figure 4.4 – Carte des fonctions cognitives humaines.

- La mémoire déclarative (explicite) : pour la mémorisation des informations exactes à travers la formulation par des mots des événements, procédures et faits. Elle permet la transmission d'une connaissance de manière explicite et de conceptualiser par la pensée chaque étape d'une action. Elle se décompose en deux formes, la mémoire sémantique pour les connaissances générales et la mémoire épisodique pour les événements ou expériences contextualisés dans l'espace et dans le temps.
- La mémoire procédurale (implicite) : pour l'apprentissage de la manière, elle sert à la mémorisation des méthodes, savoir-faires et compétences souvent effectués de manière naturelle.

Le langage (aussi appelé phasie) est la capacité à communiquer par un langage structuré à travers des phases d'expression (écriture et parole) et de réception (écoute, lecture, compréhension) Ce processus est rendu possible par la faculté de codage et décodage, de matérialiser et dématérialiser des informations. Le langage comporte de nombreuses compétences telles que l'expression, la compréhension, l'écriture, la lecture, le vocabulaire, etc.

Les fonctions exécutives sont les fonctions cognitives les plus complexes car incorporant toutes les autres fonctions cognitives en vue de planifier, organiser, évaluer, guider et contrôler le comportement afin de l'adapter aux situations et à l'environnement pour atteindre des objectifs. Pour réaliser ces tâches, les fonctions exécutives utilisent trois principaux processus :

- La planification qui fournit les objectifs, élabore les plans d'actions pour les atteindre sous la forme d'étapes successives, évalue le plan qui semble le plus adéquat en fonction de l'environnement et de ses répercussions sur la situation puis génère une décision.

- La flexibilité qui permet de comparer les résultats obtenus et d'en conclure le besoin ou non d'une nouvelle planification.
- L'inhibition est la capacité d'abstraction aux informations secondaires ou superficielles, ou la capacité d'arrêt de l'exécution d'un processus.

La cognition sociale est l'ensemble des capacités cognitives et émotionnelles grâce auxquelles nous analysons et interprétons les interactions sociales avec autrui. Cette fonction cognitive est étroitement liée aux codes sociétaux mémorisés par la fonction de mémoire et à notre manière de raisonner à travers notre façon de penser et nos agissements à travers notre comportement issu des fonctions exécutives.

La perception, aussi appelée gnosie ou fonctions visuo-spatiales, est l'aptitude à s'orienter dans l'espace, à percevoir et conceptualiser les objets présents dans l'environnement. Dans ce cas, l'orientation est la capacité d'être conscient de sa position et du contexte dans lequel on s'ancre par rapport à un référentiel personnel, temporel et spatial. De manière générale, la perception recouvre la faculté de reconnaître et d'identifier par les sens (vision, ouïe, toucher, goût et odorat) les éléments de son environnement, mais également de repérer son corps dans l'espace de manière globale (où je suis) et de manière individuelle à travers un schéma corporel (où sont chacun de mes membres les uns par rapport aux autres).

Les fonctions motrices, aussi appelés praxie, sont l'ensemble des capacités de mouvements intentionnels que nous effectuons pour mener un plan d'action à terme ou atteindre un objectif. Il existe quatre fonctions motrices principales : la praxie visuo-constructives qui consiste à planifier des mouvements pour atteindre un objectif dans le domaine spatial, la praxie idéomotrice qui est la faculté de réaliser des gestes simples souhaités, la praxie idéatoire pour la manipulation d'objet et la praxie bucco-faciale qui est le contrôle du visage (yeux, sourcils, bouche, etc.).

4.2.2 . Autonomie des systèmes robotiques

L'autonomie des systèmes robotiques est la capacité d'analyse et d'interprétation de stimulus ou informations extérieures provenant de l'environnement sans intervention humaine pour en déduire des décisions ou tâches à exécuter. Dans la littérature, l'autonomie en IHR est caractérisée de plusieurs manières [Alb91; BH04] avec comme critère principal, le niveau d'autonomie du système. De manière formelle, le niveau d'autonomie est la proportion de temps durant laquelle le système n'a pas besoin de supervision humaine. Autrement dit, l'autonomie est étroitement liée à la capacité du système à s'auto-gérer dans divers environnements et situations. Rappelons tout de même qu'en IHR, l'autonomie n'est pas un objectif, mais plutôt une aide à

l'interaction afin d'alléger la charge de l'utilisateur humain et ainsi optimiser l'interaction. En conséquence, le niveau d'autonomie est corrélé au type d'application, aux besoins de l'interaction et à l'objectif du système.

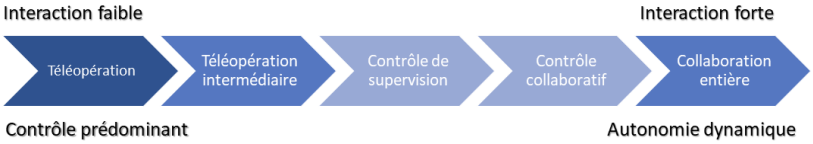


Figure 4.5 – Niveau d'autonomie et d'interaction.

La figure 4.5 décrit le parallèle entre les différents niveaux d'autonomie et les niveaux d'interaction résultants. De fait, un système hautement autonome induira souvent une interaction relativement forte ce qui conduit à définir une stratégie d'interaction pour exploiter au mieux la relation Homme-Robot en fonction des situations. Cette stratégie soulève des problématiques de conception de l'interface pour faciliter l'utilisation du système par l'homme. À l'inverse, le choix du niveau d'autonomie du robot induit le niveau de compétences cognitives [Bur+05 ; CFG08 ; Lem+10] à lui adjoindre facilitant l'interaction avec l'humain. L'autonomie des robots est rendue possible grâce à des méthodes de contrôle et de traitement de signal ancrées dans le cadre des sciences cognitives, et dernièrement grâce à des méthodes d'intelligence artificielle. L'approche générale consiste à développer des modèles haut niveau de prise de décision [Mur04 ; She16 ; Mur19] effectuant les tâches suivantes : percevoir, planifier et agir. Une approche différente de bas niveau [AA+98 ; Bro86] propose de corréler l'autonomie à des comportements réactifs associant les données capteurs aux actions/décisions directement au niveau matériel. Il existe de nos jours des systèmes hybrides [Pet+97 ; Mur19] basés sur les modèles de prise de décision juxtaposés à des approches comportementales.

Les avancées en robotique, algorithmie de raisonnement et vision par ordinateur ont permis d'organiser les travaux sur l'autonomie des systèmes robotiques autour de deux axes de recherche liés aux capacités cognitives humaines d'objectif et complexité différentes. Tout d'abord, la perception de l'environnement pour la localisation, la détection d'obstacle et le déplacement du robot dans son environnement. Cette perception est basée sur des algorithmes de vision par ordinateur comme le SLAM [MMT15a ; MT17b], la détection d'objet [RF18 ; BWL20] et la compréhension de scène [Ria+20 ; De +22 ; FZL22] réalisant des tâches cognitives telles que les fonctions visuo-spatiales (l'orientation) et la gnosie. À un niveau cognitif supérieur et plus complexe, la perception de situation à travers les fonctions cognitives exécutives relatives aux modèles de décision. Cet axe relatif au raisonnement, la planification et la prise de décision offre des cadres de recherche tels que la planification

de trajectoire [AG92 ; SKD10 ; FON14 ; ZLC18], la théorie de l'intention conjointe [CL91 ; Kum+02 ; DWH14], ou la compréhension de scène pour la collaboration Homme-Robot en IHR [Ria+20 ; FZL22 ; De +22].

4.2.3 . Information et interaction multimodale

L'échange des informations relatives à l'interaction entre hommes et robots est un critère primordial de la qualité de l'interaction. En effet, la qualité de l'interaction peut être évaluée selon des critères de charge cognitive appliquée à l'utilisateur lors de l'interaction [She+02], de niveau de collaboration et de compréhension commune entre les acteurs [LT02 ; Kle+05 ; LJO13 ; Cha+17] et de vitesse de communication entre les systèmes [Goo+01 ; Zie+10]. L'échange d'informations étant le dénominateur commun de chacun de ces critères, il est important de rappeler que pour l'humain, l'essentielle des informations s'acquièrent grâce aux cinq sens. Dans les systèmes d'IHR, seuls, 3 de ces sens opèrent à travers la vue, l'ouïe et le toucher :

- La vision intervient via des interfaces graphiques ou de réalités augmentées/virtuelles, et des algorithmes de vision par ordinateur.
- Le langage et la parole appliqués à travers des systèmes de communication sonore ou écrit pour l'interaction par le dialogue [HM15 ; CC20a]. Les avancées dans le traitement neuronal du langage par l'apprentissage profond ont permis des progrès importants dans ce domaine.
- La gestuelle correspond à la déduction d'intention ou de situation par la reconnaissance de geste, d'expression faciale ou d'action à l'aide de méthodes d'apprentissage appliquées à l'image [DS94 ; AS16 ; TA20].
- L'haptique est le retour d'information physique relative au toucher et à la kinesthésie, très utilisé en téléopération [TPM05 ; Tav+07 ; HB12] et télémanipulation [AA+98]. L'acquisition de ces informations est rendue possible grâce à des capteurs/actionneurs spécifiques permettant d'acquérir les informations de pression de contact et de restituer la sensation de toucher par des interfaces de retour d'effort.
- Le sonore par l'émission de bruit non-vocale servant essentiellement d'avertisseur.

La complexification des systèmes robotiques par l'intégration croissante de capteurs de sources d'information différentes a conduit la communauté à concevoir des interfaces d'interaction multimodale. Ce domaine de recherche est bien renseigné à travers les travaux suivants [JS07 ; Tur14]. La combinaison d'informations multiples et complémentaires a permis de rendre l'interaction plus naturelle en s'approchant de la cognition multimodale humaine. Le tableau 4.1 est une liste non-exhaustive d'exemples existants pour les principales modalités : En effet pour la perception de l'environnement ou de situation, l'homme utilise les informations issues de plusieurs organes sensoriels

Table 4.1 – Exemple d'application de modalité sensorielle humaine.

Modalité	Exemple
Vision	Reconnaissance d'action Analyse de scène Reconnaissance de geste Détection de visage Reconnaissance d'expression faciale Reconnaissance de caractéristiques humaines Estimation de l'orientation du regard
Langage et parole	Compréhension de texte (NLP pour chatbot) Compréhension de parole (NLP pour assistance vocale)
Haptique	Pression de contact Retour d'effort pour la téléopération/télémanipulation Acquisition de gestuelle
Sonore	Signalisation Environnement d'immersion

de manière complète ou partielle et séquentielle ou continue. Il exploite les modalités existantes entre les différentes sources d'information pour améliorer, compléter ou corriger son analyse favorisant ainsi son interaction avec son environnement [Que+02; VGP05].

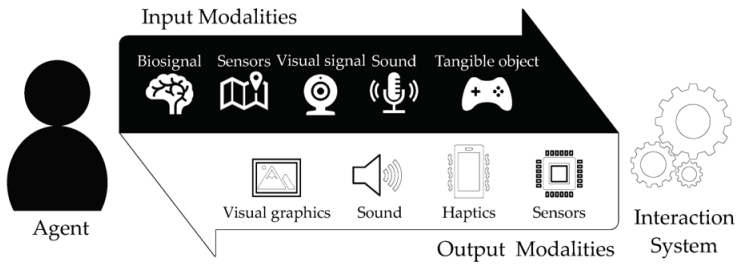


Figure 4.6 – Interaction multimodale en termes d'entrées et sorties.

Les interfaces multimodales tentent de retranscrire cette faculté humaine d'exploitation de données diverses en vue d'optimiser l'interaction comme démontré dans l'étude [DLO09]. En procédant de cette manière, les interactions multimodales offrent les avantages suivants :

- Une adaptabilité à la réception des informations par leur utilisation com-

plète ou partielle et séquentielle ou continue.

- Une possibilité d'alternative d'approche d'interaction pour un meilleur confort d'utilisation.
- Une meilleure flexibilité aux différents utilisateurs, situations, environnements et tâches.

Dans la communauté de l'interaction multimodale, le terme mode/modalité fait référence à la perception d'une information acquise par l'un des cinq sens. Ainsi, le terme multimodal fait référence à des interfaces opérant avec plusieurs modalités ou plusieurs sources d'information d'un même sens (canaux multiples). De fait, les systèmes multimodaux ont des configurations variant en fonction des modalités d'entrée, du nombre de canaux de communication, du séquencement de l'utilisation des modalités, etc. La configuration est alors adaptée à l'application, au modèle cognitif utilisé, aux sources d'information disponibles et à l'environnement d'interaction. Reeves [Ree+04] propose une liste de critères pour la conception d'une interface multimodale fonctionnelle, avec comme critères principaux :

- Le besoin de flexibilité du système multimodal pour s'adapter au mieux au contexte, besoins et capacités de l'utilisateur pour garantir la meilleure interaction possible avec l'environnement.
- L'optimisation des capacités cognitives humaines par la connaissance du raisonnement humain et de ses limites.

4.2.4 . Évaluation et métriques communes pour l'interaction homme-robot

L'interaction Homme-Robot est un domaine de recherche avec certains critères relativement abstraits. En conséquence, il peut être difficile d'évaluer et de noter une interaction de manière formelle. Certains travaux ont tenté de proposer des méthodes d'évaluation ou des métriques adaptées à ce domaine. Scholtz [Sch03] a proposé de théoriser l'interaction Homme-Robot en vue d'établir des critères utiles pour l'établissement d'interfaces utilisateur intuitives et adaptées aux différentes situations et niveaux d'interaction. En complément, il propose une méthodologie d'évaluation de l'interaction fondée sur le niveau de la conscience de la situation. Cette conscience de la situation est décomposée en trois niveaux :

- La perception des indices les plus importants pour pouvoir continuer l'interaction.
- La faculté d'analyser et d'incorporer plusieurs éléments d'informations afin d'en déduire leurs apports pour atteindre les objectifs de l'utilisateur.
- La capacité de prédiction d'événements ou situations futures en fonction de sa compréhension et perception de la situation actuelle.

Le niveau de conscience atteint par l'utilisateur indique ainsi la qualité de l'interaction fournie par l'interface.

Table 4.2 – Métriques pour les tâches de navigation et perception visuelle.

Tâche	Critère	Métrique
Navigation	Réalisation tâche	Pourcentage de tâches de navigation
		Couverture de la zone
		Écart par rapport à l'itinéraire prévu
		Obstacles évités avec succès ou non
	Temps nécessaire	Temps pour terminer la tâche
		Temps de l'opérateur pour la tâche
	Quantité de boucle	Nombre d'interventions de l'opérateur
		Rapport du temps opérateur/robot
Perception passive	Précision objets	Mesures de détection (source de la détection)
		Mesures de reconnaissance (classification)
	Spatial	Estimation absolue de distance, taille et longueur
		Estimation relative de distance, taille et longueur
		Estimation relative de distance dans l'environnement
	Mouvement	Vitesse absolue du robot
		Mouvement relatif dans l'environnement
Perception active	Reconnaissance	Vitesse de confirmation de l'identification
		Quantité de mouvement de la caméra
		Précision détection
		Temps de recherche d'objets

Steinfeld [Ste+06] tente d'identifier des mesures communes pour l'interaction Homme-Robot à travers des métriques intégrant les facteurs de biais existant. Pour cela, il identifie d'abord un ensemble de tâches existantes en IHR : la navigation, la perception, la gestion de la collaboration, la manipulation et l'interaction sociale. À travers l'ensemble de ces tâches, des effets de biais apparaissent comme :

- La communication à travers des problèmes de latence, mauvaise synchronisation et limites de bande passante au niveau de l'interface.
- La réponse du robot qui correspond au délai de traitement de l'information par le robot par rapport à celui de l'humain induisant un décalage temporel.
- L'état de l'utilisateur qui influe sur ses performances et son comportement. Les facteurs opérationnels, de matériel, de la tâche et et de l'environnement affectent de fait la perception de l'humain entraînant une fluctuation de l'interaction.

Ces biais sont intégrés dans l'ensemble des tâches considérées dans l'étude. Le tableau 4.2 récapitule les différentes métriques pour les tâches de navigation et perception en raison de leur lien avec notre propre sujet d'étude.

4.2.5 . Interaction Homme-Robot : réhabilitation et assistance

La réhabilitation à travers la robotique est un domaine de recherche pluridisciplinaire orienté vers la conception de systèmes robotiques, d'environnements et de procédures permettant de recouvrir entièrement ou partiellement des capacités motrices ou mentales réduites à la suite d'accident ou de maladie. La réhabilitation robotique est de plus en plus utilisée dans le domaine médical du fait de sa double utilité. En effet, la réhabilitation sert à la fois de support pour l'entraînement thérapeutique et d'outil de suivi du niveau de rééducation motrice atteint. Dans le cadre médical, les recherches à ce sujet tentent de répondre à un besoin social croissant d'autonomisation et de réinsertion des personnes ayant des capacités motrices réduites à travers une assistance au mouvement ou une rééducation [San21]. La conception de systèmes robotiques d'assistance ou de réhabilitation, que ce soit de type exosquelette ou interface robotique, se décompose sous plusieurs formes comme illustrées dans la figure 4.7. Les systèmes de réhabilitation thérapeutique ont souvent une configuration robotique lourde et complexe afin de disposer des ressources mécaniques et sensorielles nécessaires à l'exécution d'exercices thérapeutiques et à l'analyse de leur déroulement. À l'inverse, les systèmes de réhabilitation par l'assistance sont souvent plus légers et flexibles afin d'être déployables et adaptés à la vie quotidienne des utilisateurs.

Dans un premier temps, ces systèmes ont souffert d'un manque de considération pour les contraintes et aptitudes de l'utilisateur humain au niveau conception et intégration du contrôle. Ces premières méthodes requéraient un comportement passif du patient qui subissait le mouvement de ses extrémités imposé par le système robotique pour recouvrir une certaine mobilité mécanique [Rah+15 ; San+16 ; Amb+17]. La participation passive des patients simplifiait la conception du système, mais ne permettait pas de pleinement stimuler la rééducation motrice au niveau neuromusculaire ou sensoriel. En

conséquence, une intégration progressive des caractéristiques physiologique et cognitive de l'humain dans la conception des systèmes robotiques de réhabilitation a été nécessaire pour atteindre un niveau de performance et de sécurité suffisant. L'implication des patients a ainsi été grandissante à travers des stratégies d'application de forces de maintien ou de résistance permettant d'augmenter progressivement au fil des approches leur niveau de participation [Chi+16 ; WWD16 ; Win+16 ; Bec+18 ; Bin+19].

Figure 4.7 – Systèmes IHR de réhabilitation thérapeutique et d'assistance.

L'intégration progressive des interactions Homme-Robot et l'étude de son impact sur ces système [ST14 ; Bec+17] a permis d'identifier les modalités d'interaction afin de concevoir des interfaces adaptées et exploitant au mieux les capacités de l'utilisateur et du système robotique. L'ensemble des capteurs et actionneurs incorporés ont fourni un support pour une interaction à double sens au sein des systèmes afin d'insérer davantage l'humain dans l'environnement d'interaction. Les systèmes de caméra (stéréovision, monoculaire, de profondeur, etc.) et capteurs proprioceptifs (centrale inertielle, accéléromètre, capteurs articulaires, etc.) renseignent sur les positions et mouvements des humains et robots dans l'environnement de simulation pour analyser l'interaction s'opérant. D'un point de vue de la communication, le développement de modèles de traitement neuronal du langage (NLP) pour l'interaction [Ami+16 ; Mav+19 ; Jay21] écrite ou orale permet l'ajout d'une modalité supplémentaire au niveau du système. La considération des forces, couples et contraintes mécaniques s'appliquant au système lors des interactions est rendue possible grâce aux capteurs de force et pression [LP17].

De nos jours, les systèmes robotiques pour la réhabilitation sont adaptés pour la rééducation de troubles physiques ou neurologiques résultant

de lésions de la moelle épinière ou des cervicales, d'accidents vasculaires cérébraux, d'erreurs lors d'interventions chirurgicales, etc. L'intensité et la fréquence des séances thérapeutiques ont un impact direct sur la récupération et la rééducation des patients en raison de la tonification des cellules musculaires par l'exercice physique et de la régénération neurologique grâce à la neuroplasticité humaine. Burgar a démontré dans son étude [Bur+oo] que la rééducation effectuée avec des robots de réhabilitation permet d'obtenir de meilleurs résultats qui sont plus durables dans le temps. En effet, l'utilisation de systèmes robotiques pour la réhabilitation présente de multiples avantages :

- Un accès simplifié aux exercices thérapeutiques par la possibilité d'ajuster le nombre et la fréquence des séances pour optimiser les résultats.
- Un outil qui réduit la charge de travail des thérapeutes, que ce soit en nombre d'heures à effectuer ou comme système d'assistance lors des séances.
- La récupération d'information sur les performances des patients pour effectuer un suivi de leur progrès et déterminer les futurs exercices.

L'apprentissage et la formation sont l'une des branches principales de l'interaction Homme-Robot. Cependant, cette branche recouvre des objectifs différents en fonction de l'opérateur qui est en position d'apprenant. En effet, la

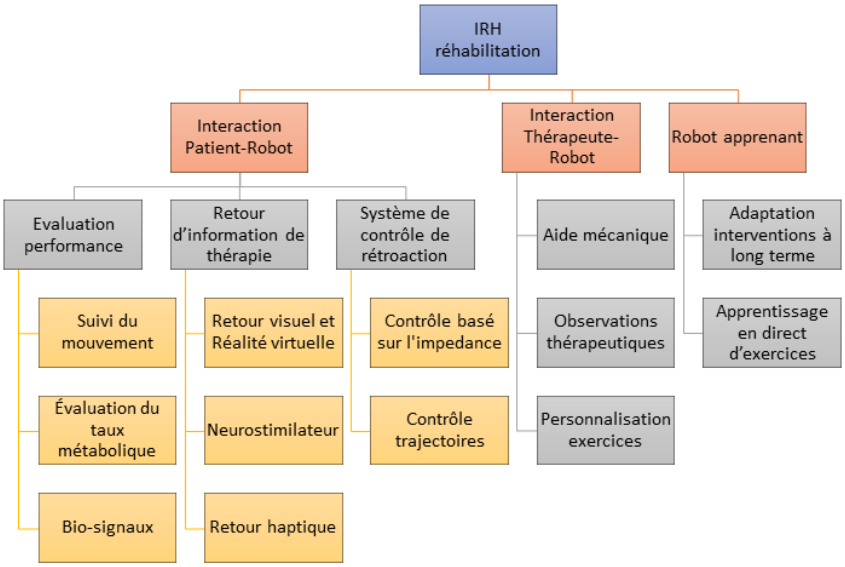


Figure 4.8 – Forme d'interaction pour la réhabilitation via Interaction Homme-Robot.

configuration de l'interface et de l'environnement seront différentes dans le cas où l'humain apprend du robot, de celui où le robot apprend et de celui dont l'apprentissage est collaboratif entre les deux types d'opérateurs.

Dans le contexte de la réhabilitation, ces systèmes robotiques peuvent être de trois formes différentes en fonction des opérateurs de l'interaction : robot-praticien, robot-patient et robot apprenant.

L'interaction robot-patient qui nous intéresse est la plus courante et regroupe l'ensemble des systèmes où le robot sert d'outil thérapeutique directement pour l'humain. Cette forme d'interaction collecte les informations de performance du patient et de retour d'information capteur pour le contrôle et l'interaction Homme-Robot comme représenté dans la figure 4.8.

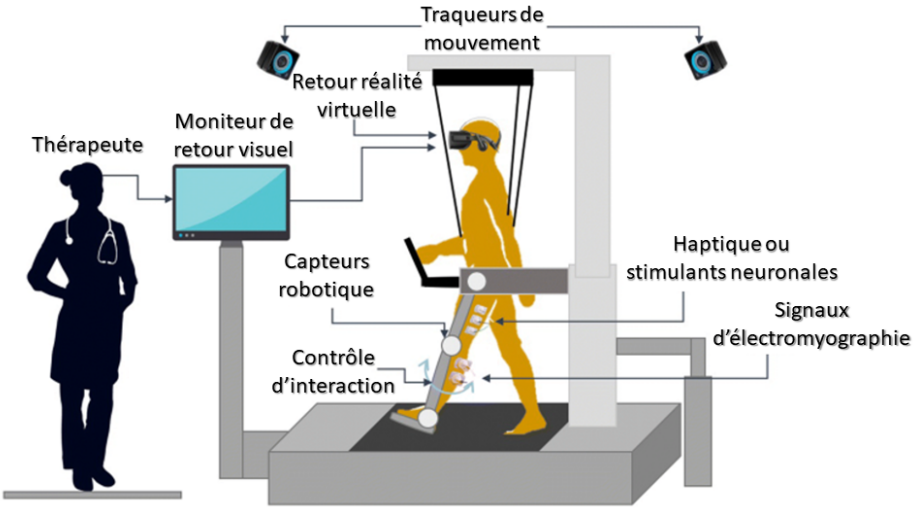


Figure 4.9 – Plateforme de réhabilitation par exosquelette d'interaction multimodale.

Pour la réhabilitation thérapeutique des capacités motrices, les plateformes de réhabilitation incorporent très souvent des exosquelettes intégrés à des systèmes d'interaction Homme-Robot multimodal basés sur une multitude de capteurs et interfaces complémentaires comme schématisé dans la figure 4.9. Ces plateformes de réhabilitation prennent parfois la forme de simulateur de réalité par l'ajout d'interface graphique (écran ou casque de Réalité Virtuelle) afin de fournir au patient un stimulis visuel offrant une meilleure immersion pour la réalisation d'exercice plus réaliste. En outre, l'interface graphique offre également une scénarisation des séances thérapeutiques pour l'exécution de tâches spécifiques stimulant un ou des organes à travers des objectifs concrets favorisant l'implication du patient. Enfin, l'ajout de cette modalité d'interaction permet de stimuler plus fortement les fonctions cognitives du patient pour une meilleure rééducation.

Dans le cadre de la réhabilitation par l'utilisation d'exosquelette, deux formes existent comme illustrées dans la figure 4.10 : les exosquelettes portatifs sous

forme de combinaison et les exosquelettes intégrés à des plateformes fixes. De nombreux travaux ont proposé des solutions de réhabilitation basées sur les Electro-Encéphalo-Gramme (EEG) et/ou ElectroMyoGramme (EMG) sans support visuel ou de Réalité Virtuelle [Yu+15 ; Wen+20 ; Shi+21]. Une étude sur l'impact de l'incorporation de la Réalité Virtuelle dans un système robotique pour la réhabilitation à la marche a été menée [Cal+17]. Elle a démontré l'apport considérable de cette modalité d'interaction supplémentaire qui sollicite des zones du cerveau responsables des fonctions cognitives de la planification, de la praxie et de l'apprentissage.

Ocampo [OT19] propose un système de rééducation des membres supérieurs grâce à l'association d'un environnement en Réalité Augmentée (RA) et de retour haptique pour une rétroaction. Un système robotique de perception intelligente est proposé par Weiguang [MZY16] pour distinguer les phases de marche du patient en se basant sur les retours capteurs fournissant des informations d'électromyographie de surface, des angles articulaires et de la force d'interaction Homme-Robot. La distinction des phases de marche permettant ensuite d'adapter les commandes de marche du système de contrôle et la force d'assistance du système.

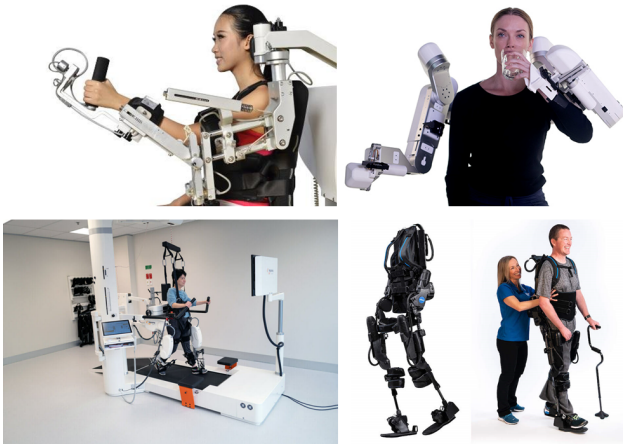


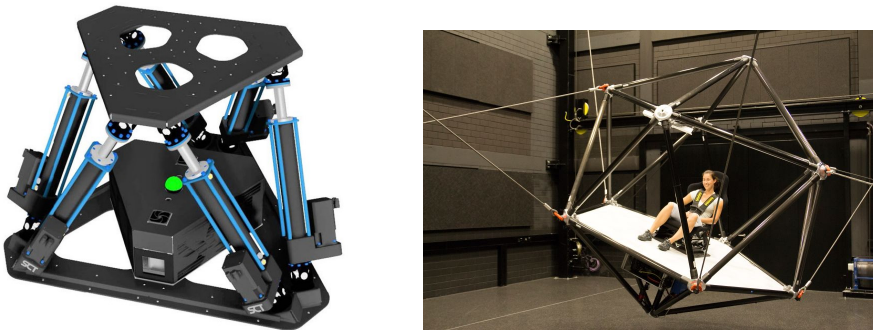
Figure 4.10 – Formes d'exosquelettes pour la réhabilitation des membres inférieurs et supérieurs.

Les travaux de l'étude [GLZ17] tentent d'intégrer directement plusieurs modèles de marche contrôlable de manière active par le patient grâce à une IHR basée sur l'électromyographie. Pour les personnes en situation de handicap moteur ou cognitif, des interfaces ont été conçues afin de permettre leur rééducation. La plateforme robotique [Gom+17] intègre de nombreux capteurs pour proposer différentes thérapies adaptées aux types et aux degrés de handicap. Pour cela, la modalité de l'interaction est ajustée en fonction de l'objectif et de la forme de la thérapie.

4.3 . Simulateur en réalité mixte

4.3.1 . Simulateur de conduite/vol

La plupart des simulateurs de conduite et de vol sont conçus à l'aide d'une plateforme robotique parallèle, que ce soit de type Gough-Stewart (voir figure 4.11a) ou robot câble (voir figure 4.11b) . Cette forme de structure offre 6 degrés de liberté et permet la mise en mouvement de charge lourde tout en garantissant une précision élevée.



(a) Plateforme de Gough-Stewart (b) Robot câble

Figure 4.11 – Plateformes robotiques parallèles

Dans un premier temps, les simulateurs de conduite et de vol ont intégré des plateformes robotiques parallèles classiques leur offrant une amplitude de mouvement relativement limitée. Les constructeurs automobiles ont mené des recherches afin de surmonter cette limite ce qui a permis le développement de filtres appelés *washout* [GR97 ; HLA04] permettant de reproduire des trajectoires amples dans des espaces de travail plus réduits.

(a) Simulateur de conduite de Renault [Rey+00] (b) Simulateur de conduite Virttex de Ford [Gra+01]

Figure 4.12 – Première génération de simulateurs de conduite

Le simulateur de Renault [Rey+00] illustré dans la figure 4.12a, offrait une amplitude de mouvement linéaire de 22cm et angulaire de +/-15° avec des accélérations et vitesses maximales respectivement de 0.5g et $0.4m.s^{-1}$ en translation, et $5.27rad.s^{-2}$ et $0.52rad.s^{-1}$ en rotation. Ce simulateur était composé d'une Clio directement montée sur la plateforme de Gough-stewart et d'une interface graphique sous forme d'écrans.

Le constructeur automobile Ford a développé le simulateur Virttex (VIRtual Test Track EXperiment) [Gra+01] au cours des années 2000 (voir figure 4.12b) sous la forme d'un dôme intégrant une interface graphique. Le simulateur offrait une amplitude de mouvement de translation de 1.6m et une vitesse de $1.5m.s^{-1}$ pour un temps de réponse de 15 ms. L'incorporation de plateforme robotique hybride liant robot parallèle et robot série a permis d'augmenter la zone de travail des robots et d'atteindre des vitesses et accélérations de translation bien supérieures.

Le groupe Chrysler a conçu un simulateur Daimler-Chrysler [KH95] (figure 4.13a) composé d'un dôme installé sur une plateforme de Gough-Stewart montée sur un rail rectiligne de 3.8 m permettant d'atteindre des accélérations de 0.7g. Sous une architecture similaire, la plateforme NADSdyna [Che+01] (figure 4.13b), d'une forme hybride plus complexe, associait un dôme monté sur une plateforme parallèle à 6 degrés de liberté avec une table XY d'amplitude latérale et longitudinale de 20 mètres. Ce simulateur, incorporant une interface, permettait de reproduire des trajectoires plus amples avec des accélérations plus importantes.

(a) Simulateur hybride Daimler-Chrysler [KH95]

(b) Simulateur hybride NADSdyna [Che+01]

Figure 4.13 – Simulateurs de conduite hybrides

L'architecture hybride des plateformes de Gough-stewart a connu une utilisation croissante dans le domaine de la robotique ce qui a conduit à la réalisation de nombreux travaux de recherche à ce sujet [Hou+19c; Hou+19a; Hou+19b; Nie+13].

4.3.2 . Simulateur de sport

Quelques simulateurs de sport de glisse ont été développés ces dernières années liant plateforme robotique et environnement de Réalité Virtuelle (voir figure 4.14). Wu et Nozawa [Wu+19] proposent un simulateur d'entraînement au ski en Réalité Virtuelle qui synchronise les mouvements réels des skis et les trajectoires virtuelles induites grâce à deux traqueurs de mouvements. Pour optimiser l'apprentissage du ski, des mouvements de ski de professionnels sont intégrés dans l'environnement virtuel sous forme de repères visuels que l'utilisateur peut tenter de reproduire. Dans leur travaux suivante [Wu+20], des aides visuelles supplémentaires pour l'apprentissage sont intégrées telle que l'angle des pieds ou la position latérale. Une évaluation qualitative et quantitative de cet apport pour l'apprentissage a été effectuée, établissant les leviers disponibles pour l'entraînement au ski.

Figure 4.14 – Simulateur de ski en Réalité Virtuelle [Wu+19 ; Wu+20].

Hu [Hu+19] propose un cadre modélisant les trajectoires de ski et le contact ski-neige dans le but de générer des mouvements de ski réalistes pour le développement de simulateur en réalité virtuel. Les modélisations de cette étude pourront être reprises pour la création de notre scénario de ski garantissant ainsi un niveau de réalisme satisfaisant.

Figure 4.15 – Simulateur en réalité mixte d'aviron VR4VRT [Del+20].

Quelques travaux traitent de simulateur d'aviron en réalité mixte pour la pratique de sport en environnement intérieur contrôlé. Le système VR4VRT [Del+20] propose un simulateur de réalité mixte pour la formation à la pratique de l'aviron dans un cadre sécurisé et permettant d'avoir accès à des données exploitables pour évaluer le niveau du pratiquant (voir figure 4.15). Ce système intègre des ergomètres dynamiques en référence aux rameurs afin de fournir un système interactif multimodale via retour d'effort et stimulis visuels. Le mouvement réel de l'utilisateur est tracké puis reproduit dans l'environnement virtuel à travers une interface logicielle-matérielle garantissant une immersion optimale.
Une autre solution est proposé par Shoib [Sho+20] pour la pratique de l'aviron via un simulateur. Le simulateur est d'une configuration similaire mais l'étude sur des sujets à propos de la qualité de l'immersion est davantage poussée mettant en avant l'efficacité de ce type de simulateur.

Enfin Kim [KKK06] propose un simulateur en réalité mixte à la fois dédié à la pratique de sport et à la réhabilitation pour l'équilibre postural. Ce simulateur de réalité mixte associe une interface graphique et un vélo d'appartement pour interfacer un scénario de réalité virtuel avec le réel. Ce système a pour but d'offrir une solution de réhabilitation des fonctions motrices et de la capacité d'équilibre pour des sujets âgées. Des capteurs intégrés au vélo renseignent sur la vitesse de pédalage, le sens des poignées et les transferts de poids au niveau du support pour permettre une interaction optimale avec l'environnement virtuel. Une étude de l'impact du retour visuel des trois informations capteurs ont mis en évidence la possibilité d'utiliser ce simulateur comme outil de rééducation pour l'équilibre postural.

4.3.3 . Simulateur pour la réhabilitation

Le projet de recherche *WheelSims structuring* mené par le laboratoire "Mobilité et sport adapté" s'est focalisé sur la conception d'un simulateur d'entraînement pour la conduite de chaise roulante en environnement mixte (voir figure 4.16).
Ce simulateur, un fauteuil manuel immobile avec retour haptique entièrement conçu en laboratoire, permet de simuler la propulsion du fauteuil dans un environnement contrôlé. Un modèle ergomètre du fauteuil roulant basé sur la robotique haptique a été proposé [CBA13; CBA15] afin de pouvoir simuler les modèles linéaires et non-linéaires existants des fauteuils roulants. Ce design particulier permet de pouvoir simuler des manœuvres de virage très souvent présentes lors de déplacement. Par la suite, une étude approfondie [Blo+14] a permis d'intégrer un système de retour d'information haptique à travers un modèle optimisant la force mécanique efficace nécessaire à la réhabilitation et prise en main de chaise roulante.

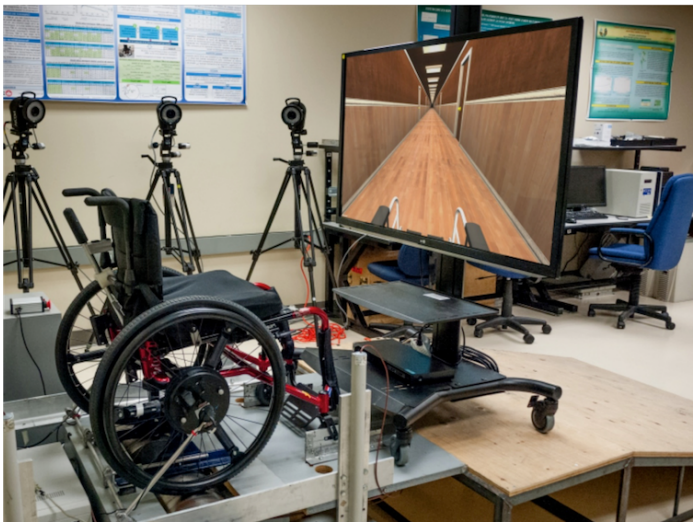


Figure 4.16 – Simulateur de réalité mixte du projet *WheelSims structuring*.

Des travaux ont traité de la réhabilitation à travers des simulateurs dans le cas de personne en situation de handicap utilisant des fauteuils roulants électriques [Arc+17 ; SH17 ; Vai+20b]. Ces études se focalisent sur le développement d'environnement de simulation ou sur l'analyse des perceptions des patients lors de l'utilisation de simulateur. Sørensen et Hansen [SH17] présentent un prototype à faible coût de simulateur de Réalité Virtuelle d'un fauteuil roulant manuel pour apprendre le maniement de ce fauteuil. Des rouleaux et encodeurs servent d'interface pour fournir à l'environnement de Réalité Virtuelle les entrées correspondant aux mouvements réels effectués par l'utilisateur. Des tests menés sur une vingtaine de sujets ont cependant montré les limites de ce prototype en termes de manque de rétroaction.

Devigne [Dev+17] a conçu un simulateur immersif pour l'assistance de conduite en fauteuil roulant électrique des personnes soufrant de handicaps visuels et/ou cognitifs. Le simulateur intègre un fauteuil roulant semi-autonome qui corrige automatiquement et progressivement les dérives de trajectoire et évite les obstacles en cas d'inaction de l'utilisateur. Ce simulateur en Réalité Virtuelle peut ainsi servir d'outil de prototypage de fauteuils ou d'outil d'entraînement de prise en main de fauteuils électriques par des patients dans un environnement virtuel contrôlé et sécurisé. Alshaer [ARO17] a mené une étude de l'impact de facteurs d'immersion sur la perception et le comportement de sujets utilisant un simulateur de fauteuil roulant électrique en Réalité Virtuelle. À travers des expérimentations effectuées sur 72 sujets, il a démontré que le type d'affichage affecte à la fois la perception et le comportement tandis que le champ de vision n'influe que sur le comportement.

Vailland [Vai+20a ; Vai+20b] propose des études sur les retours d'information vestibulaire d'utilisateurs dans le cadre d'un simulateur de Réalité Virtuelle pour fauteuil roulant électrique. Pour cela, une rétroaction vestibulaire a été intégrée au simulateur afin de garantir un sentiment de présence élevé et un cyber-malaise relativement faible. Cette approche a été testée sur des sujets soumis à des simulations avec et sans rétroaction, démontrant l'apport du système comme futur outil de formation à l'utilisation de fauteuil.

L'entreprise BTS Bioengineering propose des solutions pour la réhabilitation motrice et cognitive de personnes souffrant de handicap ou blessure. Le système NIRVANA (voir figure 4.17)

Figure 4.17 – Simulateur NIRVANA pour la réhabilitation motrice et cognitive.

Le simulateur NIRVANA est un simulateur en réalité mixte pour la neuro-réhabilitation motrice et cognitive de patients atteints de troubles neurologiques. Ce simulateur prend la forme d'une "salle sensorielle" permettant l'immersion du patient dans des scénarios stimulant et créé en fonction des objectifs de rééducation. Un écran sert d'interface graphique et un système d'analyse du comportement du patient permet d'ajuster l'environnement en fonction pour une stimulation et une rétroaction audiovisuelle adaptées.

L'entreprise Virage Simulation offre un simulateur de réhabilitation de personne en situation de handicap moteur pour la formation à la conduite 4.18. Cette solution proche des simulateurs de conduite se décompose sous la forme d'une plateforme de mouvement linéaire associé à une interface graphique et pour offrir une interaction multimodale à l'utilisateur. L'utilisateur peut ainsi se former à la conduite dans un environnement contrôlé à travers

divers scénarios permettant une formation complète à la conduite en vue d'une insertion accrue dans la société.

Figure 4.18 – Simulateur vs500m de Virage Simulation pour la réhabilitation via la conduite.

Les travaux de Kerm [Ker+19] propose une application de Réalité Virtuelle intégrant du contenu procédural générant des paysages variés en vue d'offrir une solution de réhabilitation pour la marche sur tapis roulant (voir figure 4.19). L'impact de l'immersion du simulateur sur la motivation et l'affect des utilisateurs est évalué par rapport à la pratique de la marche réelle. Dans la continuité, Winter [Win+21] a mené une étude sur l'impact de l'utilisation du simulateur de réalité mixte de marche pour la réhabilitation de troubles de la marche causés par des scléroses en plaques (SEP) et AVC. Cette étude a été effectuée sous trois conditions expérimentales correspondant à des configurations spécifiques du simulateur pour des degrés d'immersion différents :

- Immersive : marche avec tapis et scénario virtuel via un casque de Réalité Virtuelle.
- Semi-immersive : marche avec tapis et scénario virtuel via un moniteur.
- Non-immersive : marche sur tapis roulant sans Réalité Virtuelle.

L'apport de l'entraînement à la marche immersif par rapport aux deux autres a permis de confirmer l'efficacité de cette méthode de réhabilitation pour garantir une motivation et implication des patients.
Borrego [Bor+16] propose un environnement de simulation mixte pour l'évaluation du mal du simulateur dans le cadre de la réhabilitation pour la marche. Le système est composé de l'environnement réel contrôlé CAVE intégrant des projecteurs, caméras infrarouges de tracking, un flystick, et un casque de Réalité Virtuelle incorporant un scénario adapté aux dimensions de l'environnement réel (voir figure 4.20).

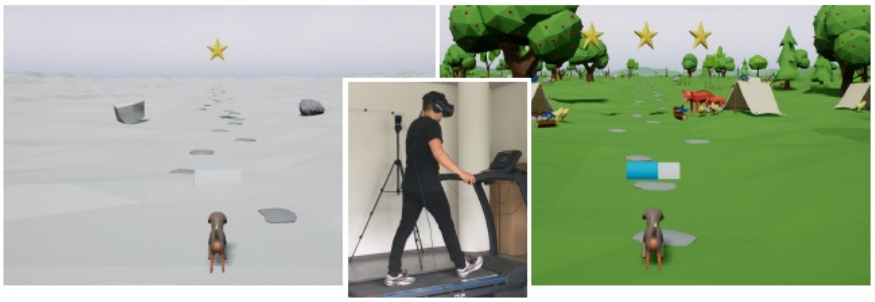


Figure 4.19 – Simulateur de réhabilitation pour la marche.

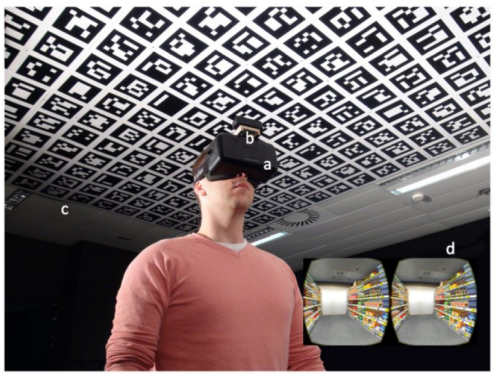


Figure 4.20 – Système CAVE pour la réhabilitation via un simulateur de marche.

Des fudiciales projetés par les projecteurs, le joystick tenu par l'utilisateur et les marqueurs du casque permettent de suivre la position de l'utilisateur dans l'environnement réelle pour retranscrire son mouvement dans l'environnement virtuel afin de garantir un sentiment de présence élevé. Ce système permet ainsi d'évaluer la qualité de la présence, la précision et le décalage de suivi pour des scénarios de marche via des questionnaires post-simulation.

4.4 . Conclusion

Dans ce chapitre, nous avons introduit les principaux concepts de l'interaction Homme-Robot nécessaires à la compréhension des modalités d'interaction opérant dans notre système de simulation. Les notions sous-jacentes de l'IHR ont été détaillées afin de définir et d'identifier les biais cognitifs présents dans notre système, le niveau d'autonomie de l'IHR utilisée, la forme de multimodalité présente, et des pistes pour évaluer les interactions agissantes. Notre simulateur incorpore des interactions Homme-obot pour la réhabilitation sous la forme d'interaction patient-robot. Partant de ce constat, une attention particulière sera apportée aux retours d'information et l'évaluation

des performances (voir figure 4.8). L'étude en profondeur des composantes de notre système de simulation permettra de déterminer les modèles cognitifs à concevoir par la suite afin d'incorporer la capacité de perception du mouvement propre au robot humanoïde NAO. Un état de l'art des applications de réhabilitation basées sur l'IHR a mis en lumière les problématiques et enjeux de ce domaine. Plus particulièrement, les travaux menés sur les ressentis des sujets lors de simulation pour la réhabilitation à l'aide d'un fauteuil électrique ont fourni des indications et orientations pour la suite de nos travaux à propos de la perception inertielle du mouvement propre dans un environnement simulé.

5 - Vers un SLAM visuel dynamique

5.1 . Introduction

L'étude des interactions multi-sensorielles de notre système robotique liée à la perception du mouvement est rendue possible grâce à la substitution de l'humain par le robot humanoïde NAO. Dans le cadre physiologique, la perception du mouvement propre via le système visuel humain est rendue possible grâce au cortex visuel qui analyse et traite les informations à travers des raisonnements complexes. Notons que les capacités cognitives visuo-spatiales humaines garantissent un niveau de robustesse aux environnements complexes (dynamique, surchargé, peu texturé, etc.) permettant une perception du mouvement propre fiable et précise. En conséquence, le passage d'un système humain à un système robotique implique une nécessité de reproduire les capacités humaines via des algorithmes ou modèles mathématiques robustes. En effet, l'utilisation d'un robot humanoïde permet l'acquisition d'informations exploitables, dont en particulier visuelles, en vue de percevoir son mouvement, mais nécessite de fait le développement d'algorithme imitant le raisonnement humain. L'algorithme de vision par ordinateur proposé devra ainsi être apte à appréhender la complexité des informations visuelles pour en extraire des données utiles, mais également atteindre une certaine robustesse indispensable à la compréhension d'environnements réels et complexes.

La perception chez l'humain se fait à l'aide de deux processus joints et complémentaires, l'estimation du flux optique qui est un raisonnement plutôt abstrait, et l'estimation de la position par l'identification de point d'ancrage dans la scène observée qui permettent la localisation au cours du temps par un raisonnement de vision plus global (profondeur, coins, lignes, particularités, etc.). La deuxième approche, plus sophistiquée, correspond à une méthode bien connue de vision par ordinateur appliquée à la robotique : le SLAM (*Simultaneous Localization And Mapping*). Cette méthode repose sur le suivi de points d'ancrage présent dans la scène au cours du temps pour déterminer par des raisonnements géométriques appliqués à la vision la pose d'une caméra dans un environnement inconnu. En raisonnant par homothétie, l'œil humain correspond à la caméra tandis que le cortex visuel correspond à l'algorithme de SLAM visuel dynamique. Pour se conformer aux capacités humaines, l'algorithme de SLAM proposé est conçu pour être robuste aux dynamiques de scène résultant de mouvements d'éléments. Ce chapitre détaillera l'algorithme proposé, son concept, ses limitations et les solutions apportées, et les résultats obtenus.

5.2 . État de l'art

Le SLAM visuel, thème majeur dans le domaine de la robotique autonome, est un axe de recherche pluridisciplinaire liant des concepts de vision par ordinateur à des besoins et fonctionnalités propres à la robotique. Il présente l'avantage d'offrir une solution efficace, facile à déployer et peu onéreuse du fait de l'utilisation de capteur optique (caméra), pour la conception de systèmes robotique autonomes capables de se localiser et de se déplacer dans des environnements inconnus. Dans cette section, nous présentons l'état de l'art et fournissons une description globale du système de SLAM visuel, des concepts associés et de ses principaux processus.

Dans un premier temps, une esquisse de son évolution au cours du temps est présentée 5.2.1 pour synthétiser les différentes phases qui ont permis son arrivée à maturation. Puis, une description détaillée de l'architecture vSLAM permet de comprendre le principe de cette méthode de robotique et les différents éléments caractérisant les techniques de vSLAM. Les principaux algorithmes de vSLAM sont ensuite introduits et brièvement discutés, en soulignant les aspects les plus pertinents qui ont un impact sur leurs performances. Enfin, un regard critique sur le vSLAM et son évolution à moyen et long terme, à la lumière des dernières avancées technologiques et conceptuelles, met l'accent sur les avancées techniques et algorithmiques les plus significatives.

5.2.1 . Évolution du SLAM visuel : un regard sur le passé et le présent

Au fil du temps, les concepts de vSLAM ont évolué pour s'adapter aux objectifs de performance dans différents thèmes de recherche. En effet, on peut diviser le développement du SLAM en 3 phases comme le montre la figure 5.1.

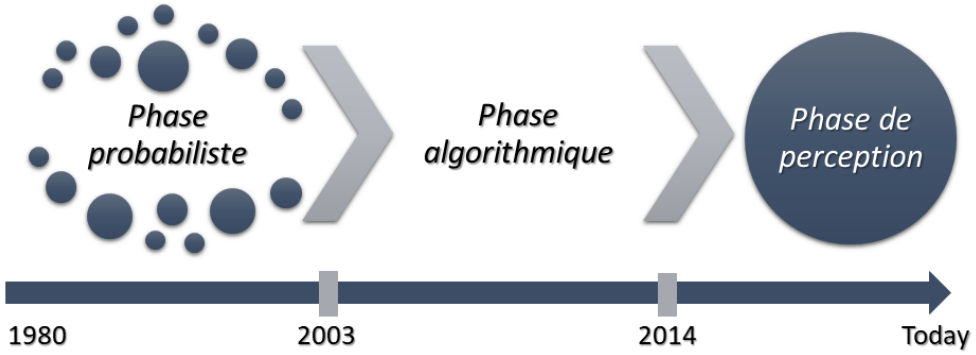


Figure 5.1 – Aperçu de l'évolution des systèmes vSLAM.

Premièrement, la phase probabiliste des années 80 à 2003, qui utilisait une formulation probabiliste pour cartographier et localiser simultanément un environnement via les méthodes du filtre de Kalman étendu (EKF) [Dav+07] et de l'estimation de la vraisemblance maximale (MLE), respectivement. Cette phase était essentielle pour le SLAM classique utilisant le LIDAR et d'autres capteurs. Ensuite, la phase algorithmique et vision de 2003 à 2014 était basée sur l'amélioration des compétences en vision par ordinateur. Cette période a établi l'architecture des systèmes de vSLAM et des principaux algorithmes. En outre, il a introduit des critères d'observabilité et de cohérence ainsi que le terme d'efficacité du SLAM et la densité de la carte. Enfin, la phase Perception de 2014 à nos jours vise à améliorer la robustesse des systèmes par l'apprentissage. Il permet d'améliorer la perception de l'environnement pour effectuer des tâches autonomes.

Le développement de l'apprentissage profond et son application en vision par ordinateur ont permis de faire des progrès significatifs dans la détection et la reconnaissance d'objets via le Deep Neural Network (DNN) [Gir+14 ; Gir15 ; Ren+16] notamment. En effet, l'apprentissage a été l'élément principal de la troisième phase à travers la compréhension de l'environnement et la sélection des fonctionnalités. Ces tâches d'apprentissage sont basées sur l'utilisation de bases de données issues des travaux de la communauté de la vision par ordinateur. Ainsi, la robustesse des méthodes de vSLAM a été améliorée avec un meilleur processus de détection des points d'intérêt et quelques optimisations de l'architecture du système. Enfin, un autre domaine de recherche est présent dans cette phase avec l'association de capteurs par fusion de données entre les informations de la caméra et l'unité de mesure inertielle (IMU). Cette approche fournit un processus d'estimation plus fiable qui exploite les avantages des deux capteurs pour être plus robuste pour des environnements et des conditions complexes.

5.2.2 . Architecture du SLAM visuel

Au départ, l'architecture vSLAM était composée de trois modules initiaux dédiés aux tâches de base, à savoir l'**initialisation**, la **localisation** et la **cartographie** [Dav+07 ; SLL02]. Par la suite, de nombreux modules supplémentaires ont été ajoutés pour améliorer les performances et notamment la cohérence globale de la localisation et de la cartographie. Ces trois tâches initiales sont interconnectées pour atteindre différents objectifs, tels que l'estimation de la pose de la caméra et la reconstruction 3D de l'environnement inconnu. Pour cela, une première estimation de la pose de la caméra est obtenue, grâce à la détection de points d'intérêt qui sont également utilisés pour construire une carte initiale. Le suivi de points d'intérêt et la cartographie sont effectués simultanément et en continu pour estimer la position de la caméra 3D dans la

carte de manière cohérente. Cette estimation est réalisée grâce à la méthode Structure from Motion (SfM) [HZ06 ; Nis03 ; Tri+99 ; ESN06] en exploitant la relation entre les coordonnées de l'image 2D et les coordonnées de la carte 3D à partir des points d'intérêt détectés.

L'appariement et le suivi de ces points sont réalisés par le biais du processus SfM. Par conséquent, la localisation et la cartographie sont simultanément effectuées et intégrées dans un seul modèle cohérent. Il est à noter que ces tâches correspondent respectivement à l'estimation de la pose courante de la caméra dans l'environnement et aux observations partielles de l'environnement. Enfin, il est établi qu'une localisation précise est possible si la carte est précise et inversement.

Par la suite, deux modules supplémentaires, à savoir la **relocalisation** et l'**optimisation globale de la carte**, ont été développés pour améliorer la précision et la robustesse du processus. Tout d'abord, la relocalisation est lancée en cas d'échec du suivi en raison d'un mouvement rapide de la caméra, d'une scène sans texture ou d'un environnement dynamique. Une phase de relocalisation implique l'absence de connaissance sur la pose actuelle de la caméra. L'absence de ces informations critiques conduit à redémarrer le processus depuis le début en utilisant la détection de points d'intérêt et de repères ou les processus de détection en boucle fermée. L'objectif étant de se localiser par rapport à des points de la carte déjà cartographiés afin de synchroniser l'estimation de la position courante avec l'existant. Un schéma résumant les tâches du système global est donné dans la figure 5.2.

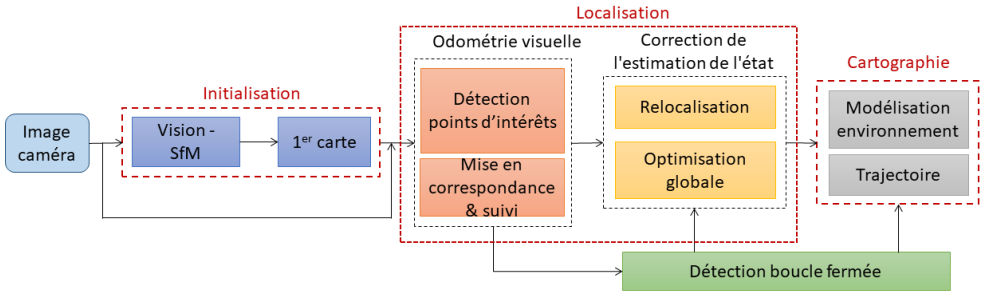


Figure 5.2 – Schéma de l'architecture des systèmes SLAM visuels.

Ces nouveaux modules ont conduit à prendre en compte la stratégie de sélection d'images clés pour réduire le coût de calcul. Ainsi, les points d'intérêt de l'étape de suivi sont utilisés pour définir les images clés, y compris uniquement les images avec des points d'intérêt différents de l'image clé précédente. Par conséquent, les images qui se chevauchent sont rejetées et le nombre d'images clés est diminué, ce qui permet ainsi de réduire la charge de calcul.

Diverses solutions pratiques pour la mise en œuvre de vSLAM ont été proposées pour améliorer la cartographie. Cependant, ces solutions souffrent d'un manque de robustesse lorsqu'on revisite un lieu déjà exploré et cartographié. L'étape de détection de fermeture de boucle a été conçue pour vérifier si un lieu a déjà été cartographié. À cet effet, la détection de fermeture de boucle utilise des repères de l'image courante et les compare aux images clés précédentes. Ce processus est appliqué à la cartographie via les modules d'optimisation et de relocalisation de la carte globale.

Enfin, l'optimisation globale de la carte supprime les erreurs d'estimation accumulées grâce à la détection de fermeture de boucle, ce qui donne une carte géométriquement cohérente. Ces processus d'optimisation tels que la détection de fermeture de boucle, le Bundle Adjustment (BA ou ajustement de faisceaux) [Tri+99 ; ESN06] ou l'optimisation de graphique de pose [Jur+21] ont été développés en fonction de la croissance de la puissance des ordinateurs.

5.2.3 . Conception et analyse de l'algorithme du vSLAM

De nos jours, l'architecture des algorithmes vSLAM a atteint un niveau de maturité et de fiabilité qui garantit des performances satisfaisantes. Les algorithmes vSLAM modernes effectuent simultanément les tâches d'Odométrie Visuelle (VO), de correction d'estimation d'état et de cartographie via différents threads (fils d'exécution) pour accélérer le processus.

Cette architecture est constituée des modules suivants (voir figure 5.2) :

1. Acquisition de l'image
2. Initialisation : première cartographie utilisant les points d'intérêt extraits d'une paire d'images.
3. Odométrie visuelle (VO) : Estimation de la pose de la caméra 3D et de la position du repère 3D dans l'environnement entre des images consécutives.
4. Estimation de l'état : un raffinement global des poses estimées à partir de la VO et de la détection de fermeture de boucle.
5. Relocalisation : détermination de l'emplacement actuel en cas d'échec du suivi.
6. Optimisation globale : Préservation de la cohérence géométrique de l'ensemble de la carte en utilisant l'ajustement de faisceau (BA) ou d'autres processus.
7. Détection de fermeture de boucle : vérification si des lieux ont déjà été cartographiés.

8. Cartographie : Intégration des observations partielles de l'environnement dans un modèle cohérent pour proposer un chemin de navigation possible.

L'algorithme de vSLAM, fourni ci-dessous, met en évidence les liens entre les différentes tâches.

Algorithm 1 Visual Simultaneous Localization and Mapping

Require: Image : Acquisition d'images
  while Image **do**
    $Initialization \leftarrow$SfM et $1^{st} mapping$
    $Tracking \leftarrow 1$
    $Newplace \leftarrow 1$
    while New place = 1 and Tracking = 1 **do**
      $VO \leftarrow Executed$
      $Loop\ closure\ detection \leftarrow Executed$
      if Tracking = Failed **then**
        $Tracking \leftarrow 0$
        $Relocalization \leftarrow Executed$
      else if Place already mapped **then**
        New $place \leftarrow 0$
        $Relocalization \leftarrow Executed$
      end if
      $Global\ optimization \leftarrow Executed$
      $Mapping \leftarrow Executed$
    end while
  end while

Ces étapes peuvent être regroupées en quatre parties : initialisation, front-end, back-end (relocalisation, optimisation globale et détection de fermeture de boucle) et cartographie. L'initialisation est nécessaire pour obtenir la première carte grâce à la méthode Structure from Motion [HZ06; Nis03] appliquée aux images consécutives. L'estimation du mouvement de la caméra et de la structure 3D de l'environnement est réalisée de manière discontinue et sans aucune considération globale. Par conséquent, l'incorporation des parties front-end et back-end surmonte les limitations de la méthode SfM.

La partie front-end, composée de l'Odométrie visuelle (VO) [NNB04], extrait les points d'intérêt des images successives. Ces points permettent de ne considérer que le mouvement de la caméra sans prendre en compte l'ensemble de la carte. Cette estimation de mouvement est possible grâce à un processus d'appariement basé sur la méthode RANdom SAmple Consensus

(RANSAC) [RFP08] appliqué aux points d'intérêt pendant le processus de suivi. Cette mise en correspondance permet de déterminer la transformation homographique entre des images consécutives et d'estimer les poses courantes de la caméra et les points de repère de l'environnement. Il est à noter que ce processus souffre du manque de cohérence globale de la cartographie du fait de sa portée limitée. En effet, elle repose sur une estimation locale qui n'intègre pas une vision globale du milieu exploré.

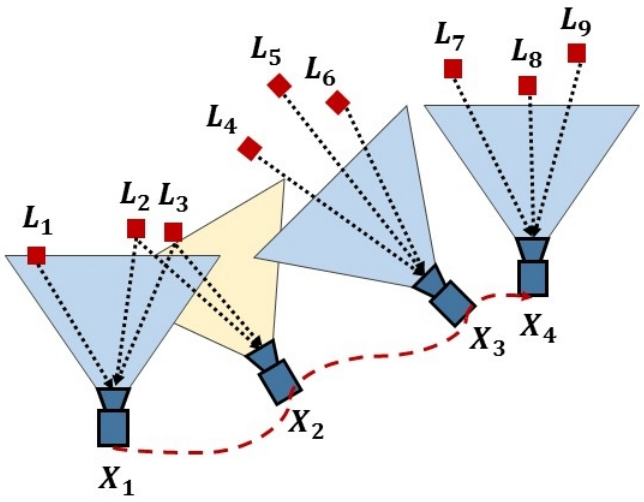


Figure 5.3 – Illustration graphique de la stratégie de sélection des images clés. Les repères rouges représentent les repères, les trapèzes bleus les images clés sélectionnées.

L'étape principale affine cette estimation via des tâches de relocalisation, d'optimisation globale ou de détection de fermeture de boucle. La relocalisation est effectuée en cas d'échec du suivi ou en présence d'un lieu déjà cartographié. Dans le premier cas, l'étape VO est relancée pour suivre de nouveaux points d'intérêt valides. Dans le second cas, l'estimation est affinée en effectuant une globalisation via l'étape de détection de fermeture de boucle. Cette étape tente d'optimiser la reprojection de l'estimation. Pour accomplir cette tâche, le processus de détection de fermeture de boucle applique l'optimisation de graphe basée sur le BA, ou une approche RANSAC Perspective-n-Point [Nis03] pour minimiser les erreurs de reprojection. Ensuite, une optimisation globale est réalisée pour améliorer la qualité de la cartographie en ajoutant de nouveaux points de repère dans la carte mise à jour ou en supprimant des points de carte incorrects. Une carte globale, avec des estimations locales de la VO pour réaliser une localisation contextuelle qui améliore conjointement les tâches de localisation et de cartographie, est ensuite utilisée. Cette stratégie permet d'optimiser le processus d'estimation.

Il est à noter que l'insertion ou la mise à jour d'images clés servent à mettre à jour la carte. La dérive, qui peut résulter de l'erreur d'estimation, peut être corrigée en utilisant de nouveaux repères au voisinage de l'image clé courante. De plus, une stratégie de sélection d'images clés est menée pour éviter la redondance et accélérer le processus. En effet, il est plus approprié de sélectionner des images clés sans recouvrement comme représenté sur la figure 5.3 dans laquelle le cadre orange n'est pas considéré selon ce modèle. Ainsi, une comparaison des points d'intérêt entre l'image courante et les images clés est effectuée pour définir les images clés suivantes. Par conséquent, les estimations de la position de la caméra et de l'emplacement des points de repère et la tâche de détection de la fermeture de la boucle nécessitent des points d'intérêt précis pour sélectionner efficacement les images clés, et pour effectuer une localisation et une cartographie précises. Enfin, un processus de cartographie est effectué. Il consiste à gérer des images clés et des points d'intérêt pour construire une carte globale cohérente mise à jour automatiquement. Par conséquent, cette architecture permet d'optimiser simultanément et continuellement la localisation et la cartographie par des processus de fermeture de boucle et d'optimisation globale.

5.2.4 . Méthodes basées sur les points d'intérêt

Les méthodes basées sur les points d'intérêt, concept de base de la vision par ordinateur, qui correspondent à des informations visuelles localisées ayant des propriétés de niveaux de gris, voisinage et forme qui permettent de les caractériser pour les repérer de manière continue. La caractérisation de ces points par des descripteurs permet leur suivi d'une image à l'autre pour effectuer des tâches de vision plus complexes telles que le calcul de pose, le suivi d'objet, etc. Ces méthodes reposent sur deux approches différentes : celles **basées sur les filtres** et celles **basées sur les BA**. Chacune utilise des détecteurs de point d'intérêts et des descripteurs pour établir un processus de suivi fournissant une estimation fiable de la trajectoire de la caméra. Cette approche a évolué pour augmenter considérablement son efficacité.

Tout d'abord, ces méthodes ont utilisé un suivi et une cartographie séquentiels associés à une optimisation à travers un filtre de Kalman étendu (EKF). La parallélisation des processus de suivi et de cartographie a été introduite par la méthode PTAM [KM09], ce qui a permis de gérer davantage de points d'intérêt et, par conséquent, d'améliorer la précision. Par la suite, le processus de suivi et de cartographie parallèle a été complété par une étape de détection de fermeture de boucle. De même, l'optimisation a été rendue plus efficace en ajoutant les processus de graphe de pose et d'ajustement de bundle (BA). Le passage d'une optimisation basée sur des filtres à une opti-

misation basée sur le BA a considérablement amélioré l'efficacité.

Mono-SLAM [Dav+07] :] Mono-SLAM a été le premier SLAM visuel monoculaire développé en 2007 par Davison et al. Cette méthode de filtrage est basée sur l'estimation simultanée de la pose de la caméra et de la structure 3D de l'environnement en utilisant un EKF. Ici, l'EKF prend en charge les 6 degrés de liberté (DoF) du mouvement de la caméra et les positions des points d'intérêt dans un vecteur d'état. Il linéarise le mouvement grâce à un modèle prédictif utilisant ce vecteur. Dans un premier temps, cet algorithme effectue une initialisation de la carte à partir des observations d'un objet 3D connu. Enfin, le modèle prédictif estime le mouvement de la caméra et les positions 3D des points d'intérêt. Cependant, cette méthode présente certaines limites dues à son coût de calcul, qui est proportionnel à la taille de l'environnement. De ce fait, ces limites empêchent d'atteindre un processus en temps réel.

PTAM [KM07 ; KM09] : Parallel Tracking And Mapping a été introduit en 2007 par G.Klein et al. pour surmonter le problème de coût de calcul de la méthode Mono-SLAM [Dav+07]. Les auteurs ont séparé les tâches de suivi et de cartographie en deux threads différents sur le CPU, exécutés en parallèle. Cette contribution permet de diminuer le coût de calcul sans réduire l'efficacité. La capacité de calcul libérée est ensuite utilisée pour appliquer un processus d'optimisation (BA) pour la cartographie. Cette nouveauté contribue à atteindre une estimation en temps réel du mouvement de la caméra grâce au tracking. Ainsi, l'algorithme PTAM peut se résumer en 4 étapes :

- Initialisation de la carte : elle est effectuée par l'algorithme à cinq points [Nis03].
- Estimation des poses de la caméra 3D : elle est effectuée à partir de points d'intérêt appariés entre les points de la carte et l'image d'entrée.
- Positions 3D des points d'intérêt : elles sont estimées par la triangulation des images clés avec la pose estimée de la caméra 3D via le processus d'optimisation BA [Tri+99 ; ESN06].
- Tracking : Effectué par une recherche aléatoire basée sur un arbre.

Enfin, PTAM a introduit le concept d'image clé, qui correspond à un écart important mesuré entre une image d'entrée et une image plus ancienne classée comme image clé. Une grande disparité améliore la précision de la triangulation, ce qui implique une importance cruciale pour le choix de l'image clé. De plus, deux optimisations BA sont appliquées pour estimer la structure 3D. En effet, une BA locale avec quelques images clés courantes et une BA globale avec toutes les images clés pour une meilleure cohérence globale de la carte.

ORB-SLAM [MMT15b] : est une extension de l'algorithme PTAM avec quelques

améliorations et nouveautés. En effet, ORB-SLAM suit les étapes précédentes, mais ajoute un troisième thread parallèle pour effectuer une détection de fermeture de boucle [MT14a ; MT14b]. Ce troisième thread vérifie si un lieu a déjà été cartographié. Dans ce cas, il applique une optimisation du graphe de pose sur les contraintes de similarité détectées pour obtenir une cohérence globale de la cartographie et de la localisation. De plus, pour le thread de suivi, une tâche globale de relocalisation [MT14a] est effectuée à la fois en cas de suivi perdu et pour classer les nouvelles images en images clés ou non.

ORB-SLAM présente de nombreux avantages par rapport aux méthodes précédentes. Premièrement, l'utilisation des mêmes points d'intérêt pour le suivi, la cartographie, la relocalisation et la fermeture de boucle rendent le système plus efficace et plus fiable. En effet, le descripteur ORB permet d'atteindre un processus en temps réel et d'être plus robuste aux rotations, translations et illuminations. Il présente l'avantage d'être plus précis et répétable par rapport aux descripteurs précédemment utilisés comme le SIFT, SURF, etc. Cette amélioration de la robustesse corrige les erreurs de suivi et permet de réutiliser la carte. De plus, la nouvelle procédure d'initialisation automatique est plus robuste et flexible, ce qui lui permet d'être utilisée pour des scènes planes et non-planes. La modification de la stratégie de sélection des images clés augmente l'efficacité du processus en limitant les redondances d'images clés et les coûts de calcul. Enfin, le module de fermeture de boucle et le graphe de covisibilité séparent la cartographie locale et globale, permettant des opérations en temps réel dans de grands environnements.

ORB-SLAM2 [MT17a] : est une version mise à jour de la précédente méthode ORB-SLAM [MMT15b]. Il présente de nombreuses améliorations et innovations. Tout d'abord, cette méthode est maintenant adaptée pour les caméras monoculaires, stéréo ,RGB-D et permet de réutiliser les cartes précédentes. Un pré-traitement d'entrée est ajouté pour prendre en compte les systèmes stéréo et RVB-D pour l'extraction des points clé. Il existe encore trois fils d'exécutions parallèles : premièrement, l'étape de suivi localise la position de la caméra en déterminant les correspondances d'entités entre les points d'intérêt des cadres et les points d'intérêt de la carte locale. Ensuite, il minimise l'erreur de re-projection grâce au Bundle Adjustment. Deuxièmement, la cartographie locale est effectuée par une optimisation BA locale. Ensuite, le processus de fermeture de boucle est effectué pour corriger la dérive accumulée à l'aide de l'optimisation du graphe de pose. L'association de plusieurs systèmes de vision augmente considérablement la précision.

L'algorithme **OpenVSLAM [SSS19] :** a été conçu par S. Samikura et al. en 2019. Il est basé sur la même architecture que les méthodes SLAM visuelles basées

sur les points d'intérêt via des modules de suivi, de cartographie et d'optimisation globale. Ces modules construisent une carte locale et globale et fournissent la pose de la caméra et la position des points d'intérêt 3D. Cette méthode atteint de bons résultats avec une grande précision. De plus, il a l'avantage d'être adapté à différents types de modèles de caméras (perspective, fisheye, équirectangulaire) et est personnalisable pour d'autres modèles de caméras. Enfin, il possède un système pour enregistrer et charger les cartes créées en tant que cartes pré-construites et localiser de nouvelles images en fonction de celles-ci. Les résultats d'OpenVSLAM sont très encourageants. En effet, le temps du processus est inférieur au temps d'ORB-SLAM et fournit une précision équivalente à l'ORB-SLAM pour le jeu de données EuRoC MAV (caméra monoculaire) et le jeu de données KITTI Odometry (stéréo vision).

UcoSLAM [MM19] : a été développé par R.Munoz et al. en 2019. Il introduit une nouvelle approche pour le SLAM visuel monoculaire utilisant des points d'intérêt naturels et artificiels. Les points d'intérêt artificiels issues des marqueurs fiduciaux fournissent des points de repère stables supplémentaires, rendant le processus plus fiable et le suivi plus robuste. L'apport principal est de rendre l'échelle accessible tant qu'un repère est observé. De plus, il permet d'utiliser ensemble ou séparément des marqueurs et des points clé naturels. Cette flexibilité permet de s'adapter à la présence ou non de marqueurs et de maintenir la robustesse en cas de trop faible nombre de points d'intérêt naturels. De même, le problème d'ambiguïté visuelle due à l'environnement répétitif est résolu pour le module de relocalisation, grâce aux marqueurs. Enfin, en plus de l'estimation d'échelle, UcoSLAM permet de charger et de stocker les cartes générées. Il est très utile pour de nombreuses applications réelles, car il permet d'économiser les ressources informatiques.

La méthode **ORB-SLAM3** [Cam+20] a été développée en 2020 par C.Campos et al. Il associe plusieurs types de caméras et IMU dans un système basé sur le vSLAM pour atteindre une meilleure précision et robustesse en temps réel par rapport à l'état de l'art. Il s'agit du premier système à réaliser un SLAM visuel et visuo-inertiel et à créer plusieurs cartes. L'algorithme peut être configuré à l'aide de modèles d'objectifs pinhole ou fisheye pour plusieurs systèmes de caméras. La méthode ORB-SLAM3 apporte quatre contributions principales :

- Il introduit un SLAM visuel-inertiel basé sur les points d'intérêt et une estimation du Maximum-a-Postériori (MAP). Cette méthode y parvient de manière robuste en temps réel pour tous les environnements avec la meilleure précision par rapport aux méthodes de pointe.
- Il utilise une reconnaissance de lieu basée sur la bibliothèque de "Bag of Words" DBoW2 [GT12]. Cette reconnaissance de lieu améliore l'algorithme de la manière suivante : les images clés candidates sont d'abord

vérifiées selon leur cohérence géométrique, puis par leur cohérence locale avec trois images clés covisibles qui sont déjà dans la carte la plupart du temps. Ce processus augmente la remise en perspective globale et densifie l'association de données en améliorant la précision de la carte à un coût de calcul légèrement supérieur.

- L'ORB-SLAM Atlas est le premier système complet de SLAM multi-carte capable de gérer des systèmes visuels et visuels inertiels dans des configurations monoculaires et stéréos. Il utilise et combine des cartes construites à différents moments pour effectuer un SLAM multi-session incrémentale. Ainsi, il effectue des opérations de cartographie telles que la reconnaissance de lieux, la relocalisation de caméras, la fermeture de boucles et la fusion de cartes précises et transparentes.
- La flexibilité du système de caméra se fait grâce à la mise en œuvre des modèles sténopé et fisheye.

De nos jours, ORB-SLAM3 présente les meilleures performances par rapport aux méthodes présentes dans l'état de l'art. En effet, il est 2 à 5 fois plus précis que le précédent ORB-SLAM2 de 2017, considéré comme l'une des meilleures approches SLAM visuelles classiques.
Le SLAM visuel a été amélioré au fil du temps pour mieux estimer la trajectoire et l'environnement des robots autonomes. Aujourd'hui, les SLAM visuels simples comme l'ORB-SLAM2 et l'UcoSLAM présentent les meilleures performances pour les caméras monoculaires, stéréos ou RGB-D. Le tableau 5.1 résume les méthodes en source ouverte les plus célèbres de SLAM visuel dans un diagramme. L'association des systèmes visuels et inertiels augmente considérablement la précision et la robustesse de la localisation et de la cartographie. De nos jours, le VINS-Mono/fusion [Qin+19], et l'ORB-SLAM3 [Cam+20] sont les méthodes basées sur la VI avec les meilleures performances.
Tout d'abord, de nombreuses recherches ont été menées pour améliorer l'estimation de trajectoire et la cartographie en développant de nouvelles méthodes d'optimisation telles que l'ajustement de faisceau local et global (BA). Ces optimisations sont réalisées après la triangulation des points 3D ou l'optimisation du graphe de pose pour produire une carte locale avec une cohérence globale.

Ensuite, de nombreuses stratégies sur le choix des images clés ont permis d'augmenter la vitesse du système. De même, l'ajout croissant de capteurs dont les capteurs inertiels et caméras ont permis d'améliorer sa robustesse. La fusion visuelle-inertielle devient l'un des sujets majeurs dans le domaine de la robotique à travers le SLAM visuel en raison de ses avantages et de sa mise en œuvre assez facile. Enfin, le dernier domaine de recherche dans les systèmes SLAM visuels modernes peut se décomposer en trois thèmes : la perception de l'environnement, l'optimisation de l'estimation et la robustesse

à l'environnement. Dans l'ensemble, ces thèmes ont été développés grâce à l'émergence de l'apprentissage automatique et de l'apprentissage en profondeur, et à l'amélioration des ordinateurs.

5.2.5 . Méthodes directes

Contrairement aux méthodes basées sur les points d'intérêt, les méthodes directes utilisent les images d'entrée sans aucun traitement d'image au niveau des pixels pour extraire les points d'intérêt et leur description. Ces méthodes sont basées sur l'étude de la cohérence photométrique en tant que mesure d'erreur pour estimer les changements séquentiels de la position de la caméra dans le temps.

DTAM [NLD11] : La méthode Dense Tracking And Mapping a été introduite en 2011 par R.A.Newcombe et al. pour fournir la première méthode directe. DTAM suit ces trois étapes :

- Initialisation de la carte par des mesures stéréo.
- Estimation du mouvement de la caméra en générant des vues synthétiques à partir de la carte reconstruite.
- Estimation des informations de profondeur pour chaque pixel en utilisant la stéréo multi-ligne de base, puis cette estimation est optimisée en fonction de la prise en compte de la continuité spatiale.

À noter que cette approche présente l'avantage de fournir une carte dense.

LSD-SLAM [ESC15 ; CEC15] : La méthode SLAM direct à grande échelle a été développée en 2014 par J.Engel et al. Elle propose une nouvelle méthode de suivi direct basé sur l'algorithme d'algèbre de Lie et un VO semi-dense. Il permet de sélectionner des zones utiles pour reconstituer l'environnement. Par conséquent, il ne considère pas les zones sans texture en raison de la difficulté d'estimer exactement l'information de profondeur de ces zones. Le mouvement de la caméra est ensuite estimé en minimisant les erreurs photométriques au lieu de suivre les points d'intérêt. Enfin, cette méthode fonctionne en temps réel et atteint une assez bonne précision.

SVO [FPS14; For+17] : L'odométrie visuelle semi-directe (SVO) est une méthode monoculaire de 2014 qui utilise deux modules distincts nommés estimation de mouvement et cartographie. Cette division permet d'améliorer la vitesse d'exécution, de suivre les points d'intérêt et d'étendre la carte. Tout d'abord, une image creuse basée sur un modèle d'alignement permet une initialisation de la pose pour estimer le mouvement. Ensuite, un raffinement de l'estimation de la pose et de la structure est effectué en minimisant l'erreur de reprojection de l'alignement des points d'intérêt correspondant. Enfin, le

module de cartographie est basé sur un filtre de profondeur probabiliste initialisé pour chaque point d'intérêt 2D ayant un point 3D correspondant à estimer. L'initialisation du filtre est effectuée dans le cas d'une nouvelle image clé dans laquelle des correspondances 3D à 2D sont trouvées. Cette nouvelle approche permet d'atteindre un processus rapide et précis par rapport aux techniques précédentes. De même, la cartographie probabiliste rend la méthode robuste aux mesures aberrantes.

DSO [EKC18; Mat+18; WSC17] : Direct Sparse Odometry (DSO) est une méthode entièrement directe avec une carte de reconstruction clairsemée. DSO tente de supprimer l'erreur cumulée en diminuant les facteurs d'erreur des points de vue géométrique et photométrique. Ainsi, une sélection est appliquée aux images d'entrée décomposées en plusieurs blocs pour déterminer les points candidats à la reconstruction en fonction de leur intensité. Ainsi, il répartit les points sur toute l'image. Un processus de fenêtre glissante est également effectué pour relier les différentes images et estimer la pose. Enfin, DSO utilise les paramètres géométriques et photométriques de la caméra pour estimer avec précision tous les paramètres du modèle de caméra (pose de la caméra, paramètres intrinsèques et géométriques).

LDSO [Gao+18b] : Le Direct Sparse Odometry (LDSO) avec le module de fermeture de boucle de 2018 introduit de nombreuses étapes d'optimisation des méthodes DSO [EKC18; Mat+18; WSC17]. En effet, LDSO améliore la stratégie de sélection des points du DSO pour ne conserver que les points d'intérêt de coin répétables, augmentant ainsi la robustesse. Ces points d'intérêt sont injectés dans une détection de fermeture de boucle à l'aide d'un module [GT12] basé sur les points d'intérêt Bag-of-Words (BoW) afin de vérifier les points candidats. Ces points sont utilisés pour calculer les contraintes de pose relatives Sim(3). Ensuite, un graphe de covisibilité des poses relatives utilise ces contraintes pour estimer la pose globale pour l'image clé courante. Enfin, l'optimisation intégrée du graphe de pose diminue les dérives accumulées (de translation, de rotation et d'échelle) en mettant à jour les poses globales de l'ancienne partie de la trajectoire.

5.2.6 . Robustesse des méthodes de vSLAM dans les environnements dynamiques

Le dernier axe de recherche comprend des travaux sur la robustesse des systèmes de vSLAM dans des environnements dynamiques et inconnus. L'utilisation des deux axes de recherche précédents qui mélangent perception et optimisation améliore les performances du vSLAM. Dans les approches classiques de vSLAM, les hypothèses sur l'environnement sont fournies. En effet, l'environnement est considéré comme statique avec une caméra en mouvement. Le vSLAM moderne utilise la complémentarité entre l'architecture du vSLAM et la détection d'objets pour détecter les points d'intérêt considérés dynamiques, puis les rejeter du processus d'estimation de la pose de la caméra. Le mouvement global des objets dynamiques présents dans la scène est pris en compte, diminuant l'erreur cumulée de l'estimation de pose et de la cartographie. L'environnement est analysé pour extraire des informations géométriques, contextuelles et dynamiques sans aucun modèle d'objet préalable. Ces algorithmes fournissent donc les informations nécessaires pour rendre le système autonome lors de sa navigation.

De nos jours, les méthodes vSLAM modernes tentent d'estimer les objets en mouvement présents dans des environnements dynamiques pour les représenter dans une carte spatio-temporelle. Les méthodes de SLAM dynamiques intègrent des techniques de détection/segmentation d'objets dans l'architecture vSLAM pour fournir des informations supplémentaires pour la tâche d'Odométrie Visuelle (VO). En effet, la détection d'objets permet de réaliser une stratégie de sélection de points d'intérêt pour rejeter ceux appartenant à des objets préalablement considérés comme en mouvement de par leur nature. Par conséquent, le processus de VO est optimisé, ce qui améliore la précision de la localisation et de la cartographie en réduisant l'impact des points d'intérêt dynamiques. Le tableau 5.2 représente un résumé complet de ces méthodes pour fournir une compréhension globale de leurs stratégies et avantages.

RDSLAM [Tan+13] : est l'une des premières méthodes de vSLAM monoculaire temps réel adaptées aux environnements dynamiques. Elle comprend une nouvelle représentation d'images clés et une stratégie de mise à jour prenant en compte les changements environnementaux dus aux mouvements d'objets.

Detect-SLAM [Zho+18] : développe une approche basée sur le système de vSLAM associé à un réseau de neurones profond pour la détection d'objets. La méthode met en évidence la complémentarité de ces deux fonctions pour

les environnements dynamiques, ce qui améliore les deux tâches. Les principales contributions de cette approche sont :

- La détection d'objet sélectionne les points d'intérêt utilisables malgré le mouvement de l'objet pour améliorer la précision et la robustesse du système pour les environnements dynamiques.
- La complémentarité des deux fonctions permet de construire en temps réel une carte sémantique au niveau de l'instance de l'environnement dynamique.
- Il augmente le nombre de situations pratiques dans lesquelles la détection d'objet est effectuée efficacement malgré les variations d'éclairage et le flou de mouvement grâce à l'effet de levier de la carte d'objet.

DS-SLAM [Yu+18] : est basé sur les précédents travaux ORB-SLAM2 [MT17a] associés à une tâche de segmentation sémantique via la méthode Segnet [BKC16]. Cette méthode construit une carte sémantique dense qui rend le système plus robuste dans les environnements dynamiques. Cette méthode apporte les contributions suivantes :

- Il diminue l'impact des mouvements d'objets sur l'estimation de la pose en appliquant un vSLAM sémantique complet en temps réel.
- Le système est intégré dans ROS (Robot Operating System) et évalué sur le jeu de données TUM RGB-D et dans un environnement réel.
- Le système se divise en 5 threads parallèles, dont un réseau de segmentation sémantique. Son réseau de segmentation sémantique est combiné à l'approche classique vSLAM pour déterminer les objets en mouvement de l'environnement afin de les rejeter. Cela améliore les tâches de localisation et de cartographie pour les environnements dynamiques.
- Un nouveau framework génère une carte sémantique dense en 3D qui sélectionne des voxels stables et les met à jour pour ajuster les significations sémantiques.

DynaSLAM [Bes+18] : est une méthode basée sur le framework ORB-SLAM2 [MT17a] et la détection d'objets dynamiques qui complètent l'arrière-plan occulté avec des informations statiques provenant d'images clés précédentes. Ce travail présente les contributions suivantes :

- Un système vSLAM flexible pour les configurations monoculaire, stéréo et RGB-D.
- Une méthode en ligne qui traite des objets dynamiques en appliquant une géométrie multi-vue ou un apprentissage en profondeur.
- Une stratégie différente selon la configuration de la caméra : Utilisation d'un CNN [He+18] qui segmente pixel par pixel les objets dynamiques

dans les configurations monoculaires et stéréos. Et inversement, la combinaison de modèles de géométrie multi-vue et d'algorithmes basés sur l'apprentissage profond pour détecter des objets dynamiques en configuration RVB-D.

- Une carte statique de l'environnement peut représenter l'arrière-plan du cadre occlus par les mouvements d'objets, ce qui est préférable pour les applications à long terme dans des environnements réels.

DMS-SLAM [Liu+19a] : propose une méthode d'initialisation et de suivi des poses pour la configuration des caméras monoculaires, stéréo et RGB-D. L'algorithme basé sur le travail ORB-SLAM2 [MT17a] est combiné avec la méthode Grid-based Motion Statistics (GMS). Cette approche apporte les contributions suivantes :

- Une nouvelle initialisation de vSLAM combine l'algorithme GMS (Grid-based Motion Statistics) et la fenêtre glissante pour obtenir les points de correspondance des points d'intérêt 3D statiques.
- Ces points 3D statiques sont inclus dans la carte locale via l'algorithme de correspondance des points d'intérêt GMS entre les images clés, puis une carte 3D globale statique est construite.
- L'étape de suivi de la pose ajoute des points 3D dans la carte locale si les images clés correspondent aux points d'intérêt. Ensuite, ces points 3D sont reprojetés.

ROBIO-SLAM [Liu+19b] : est une approche combinant la reconnaissance d'objets et le flux optique pour les systèmes vSLAM dans des environnements dynamiques qui permet de diminuer l'influence des objets dynamiques. Les points dynamiques sont détectés en calculant la distance entre les points d'intérêt et les lignes épipolaires. L'étape suivante consiste à détecter les objets dynamiques en utilisant le réseau de neurones YOLOv3 [RF18]. Enfin, dans le cas de fonctionnalités dynamiques présentes sur les objets dynamiques, ces points d'intérêt sont supprimés du processus vSLAM.

DM-SLAM [Che+20] : combine un réseau de segmentation d'instance [He+18] avec des informations de flux optique pour améliorer la précision de localisation dans des environnements dynamiques. Il comprend trois modules : la segmentation sémantique pour obtenir une segmentation pixel par pixel des objets potentiellement dynamiques, l'ego-motion pour estimer la pose initiale, et la détection dynamique de points selon deux stratégies. Enfin, le cadre du vSLAM intègre les points d'intérêt statiques précédents détectés pour réaliser son processus. Les principales contributions de cet article sont résumées comme suit :

- Un système vSLAM complet qui élimine l'impact des objets dynamiques sur l'estimation de pose dans des environnements dynamiques en utili-

sant conjointement un réseau de segmentation d'instance et d'estimation du flux optique.

- Une stratégie adaptée selon la configuration de la caméra permet d'extraire des points dynamiques : une comparaison entre les points d'intérêt avec des informations de profondeur et leurs vecteurs d'offset de reprojection pour les caméras RVB-D/stéréo ou l'utilisation de la contrainte épipolaire pour effectuer cette tâche.
- Une évaluation du système sur les ensembles de données publiques TUM et KITTI met en évidence son bon fonctionnement dans des scénarios très dynamiques.

[Ai+20] : est un algorithme vSLAM en temps réel basé sur les travaux de l'ORB-SLAM2 [MT17a] et une méthode de détection d'objets d'apprentissage profond [BWL20] pour diminuer l'impact des objets dynamiques et rendre le système plus robuste. Ainsi, un modèle de probabilité d'objet dynamique est appliqué pour améliorer la réalisation du module de détection d'objet en utilisant un réseau neuronal profond. Les principaux apports de cette approche sont résumés ci-dessous.

- Un réseau de neurones profonds pour la détection d'objets est ajouté au cadre ORB-SLAM2 pour effectuer une étape de pré-traitement afin de supprimer les objets dynamiques de la scène statique. Il améliore l'estimation de la pose de la caméra et génère une carte de points 3D dense.
- Un modèle de probabilité d'objet dynamique améliore la capacité à séparer les objets dynamiques des scènes statiques.
- Les points clé sont intégrés dans le module de suivi du framework ORB-SLAM2 et mis à jour.

DynaSLAM2 [Bes+20] : est une mise à jour de la méthode DynaSLAM [Bes+18] qui ajoute l'intégration de la capacité de suivi multi-objets à l'aide du framework ORB-SLAM2 [MT17a] et d'une segmentation d'instance. Une nouvelle approche d'ajustement de faisceaux optimise la structure de la scène statique et des objets dynamiques avec les trajectoires de caméra et les objets dynamiques conjointement. Cette méthode présente les apports suivants :

- Un système vSLAM en source ouverte pour les configurations de caméras stéréo et RVB-D dans des environnements dynamiques.
- Une estimation simultanée de la pose de la caméra, de la carte et des trajectoires des objets dynamiques.
- Une nouvelle solution d'ajustement de faisceaux qui optimise conjointement les poses de la caméra, la structure de la scène et les trajectoires dynamiques des objets dans une fenêtre temporelle locale.

- Une estimation des dimensions et des 6 degrés de liberté de la pose des objets dynamiques.
- Une évaluation sur le jeu de données KITTI et une comparaison avec l'état de l'art.

Dynamic Object Tracking for visual SLAM (DOT) [Bal+21] : a été conçu en 2021 par I.Ballester, A Fontan et al. pour prendre en compte les objets dynamiques dans le processus vSLAM. Il est basé sur un suivi de masque de propagation qui supprime les masques d'objets statiques d'une première opération de segmentation à l'aide de la méthode Mask-RCNN [He+18]. Les points caractéristiques de la scène sont séparés selon les points de la scène statique et les points des objets dynamiques. Ensuite, le mouvement de la caméra est estimé à travers les points statiques, tandis que les mouvements dynamiques des objets sont estimés indépendamment. Un critère de géométrie multi-vue est utilisé pour vérifier si les objets dynamiques potentiels sont réellement en mouvement et, par conséquent, pour ne retenir que les masques des objets dynamiques. Ainsi, la première scène segmentée est mise à jour itérativement image par image sans processus de segmentation supplémentaire. Cette méthode fournit les contributions suivantes :

- Une méthode qui résout le problème du coût de calcul de la segmentation d'instance.
- Un module frontal indépendant facile à brancher sur les systèmes vSLAM existants.

Le SLAM dynamique est l'un des sujets de recherche les plus récent et les plus en vogue dans le domaine du vSLAM. Il intègre les meilleurs algorithmes de vSLAM [MT17a ; Cam+20] combinés aux derniers algorithmes d'apprentissage profond appliqués à la détection/segmentation d'objets. Cette association permet une compréhension sémantique de haut niveau de l'environnement permettant d'enrichir sa perception. En effet, ces avancées ont fourni des algorithmes robustes pour des environnements dynamiques et complexes. Chaque étape de l'architecture, du front-end au back-end, tire parti de l'apprentissage en profondeur pour améliorer leurs performances. De plus, passer d'une approche par point à une approche par entité permet de concevoir l'environnement de manière plus humaine et globale.

Cependant l'ensemble de ces méthodes possèdent des limitations. La plupart de ces méthodes surmontent leurs limites en ajoutant des informations sémantiques supplémentaires dans leur système. En effet, elles ont besoin d'informations sémantiques supplémentaires pour estimer l'état des points clés non appariés via la segmentation d'objets supplémentaires (chaises, moniteurs, etc.) Ces approches ne sont donc pas généralisables car elles sont limitées par le nombre d'étiquettes utilisées pour entraîner le modèle de seg-

mentation. Pour ces raisons, nous avons concentré nos efforts dans la conception d'une méthode de SLAM dynamique qui surpasse ces limites.

5.2.7 . SLAM dynamique : apprentissage profond

L'apprentissage profond a bouleversé tous les domaines de la recherche en proposant de nouveaux outils puissants pour la réalisation de tâches complexes. Dans le cas de vSLAM, il a considérablement amélioré la précision et la robustesse du processus grâce à la détection d'objets, à la prédiction et au raisonnement géométrique. Ainsi, l'apprentissage profond ouvre une nouvelle voire pour l'amélioration de l'odométrie visuelle, la compréhension de la scène et la cartographie dynamique grâce à des cadres de SLAM sémantiques fiables [Liu+20 ; Jia+19a]. Cependant, malgré le grand nombre de travaux actuels sur l'apprentissage en profondeur, ces méthodes manquent d'interprétabilité et de capacité de généralisation pour atteindre une fiabilité indispensable dans diverses situations et environnements. De plus, le SLAM dynamique n'est pas encore utilisable dans les systèmes vSLAM embarqués en raison de sa faible vitesse de calcul. En effet, l'intégration de modules de détection d'objets augmente la complexité de calcul, limitant ainsi le développement de solutions temps réel. À l'avenir, la recherche se concentrera sur l'intégration de méthodes puissantes capables de traiter des conditions d'environnement réelles telles que des scènes dynamiques, un environnement sans texture, des changements d'éclairage ou un flou dû à un mouvement rapide. De plus, les futures méthodes devraient se concentrer sur l'optimisation des processus pour permettre la mise en œuvre de systèmes vSLAM dans des scénarios en temps réel. En effet, ce critère est crucial pour développer des applications vSLAM temps réel fiables et pratiques. Notez que les travaux récents majeurs suivants [Cam+20 ; Qin+19 ; Bes+20 ; Zha+20 ; LM21b] fournissent une base de départ solide pour développer des architectures vSLAM modernes et dynamiques.

Table 5.1 – Systèmes de SLAM visuel en source ouverte.

Méthode	Année	Caméra/ Modèle	Type vSLAM	Process.	Carto.	Reloc.	Boucle	Carte
Mono-SLAM [Dav+07]	2007	Mono/ Pers	Points	Filtre	Éparse	×	×	×
PTAM [KM07] [KM09]	2007	Mono/ Pers	Points	Optim.	Éparse	√	×	×
DTAM [NLD11]	2011	Mono/ Pers	Direct	-	Dense	-	-	-
SVO [FPS14; For+17]	2014	Mono/ Pers	Direct	-	-	×	×	×
LSD-SLAM [ESC15; CEC15]	2014	Mono/ Pers	Direct	Optim.	Semi dense	√	√	×
ORB-SLAM [MMT15b]	2015	Tous/ Pers	Points	Optim.	Éparse	√	√	×
ORB-SLAM2 [MT17a]	2017	Tous/ Pers	Points	Optim.	Éparse	√	√	×
DSO [Mat+18] [WSC17]	2017	Mono/r Pers	Direct	Optim.	Semi dense	×	×	×
LDSO [Gao+18b]	2018	Mono/ Pers	Direct	Optim.	Semi dense	√	√	×
OpenSLAM [SSS19]	2019	Tous/ Tous	Points	Optim.	Éparse	√	√	√
UcoSLAM [MM19]	2019	Tous/ Persp	Points	Optim.	Éparse	√	√	√
ORB-SLAM3 [Cam+20]	2020	Tous/ Tous	Points	Optim.	Éparse	√	√	√

Caméra : système de vision monoculaire, stéréo ou RGBD ; Modèle de caméra : perspective, fisheye ou équirectangulaire ; Type vSLAM : basée sur les points d'intérêt ou la cohérence photométrique (direct) ; Carto. : type de cartographie ; Reloc. : Relocalisation ; Boucle : Fermeture de boucle ; Carte : sauvegarde/chargement de carte ; (×) : non pris en charge, (√) : pris en charge.

Table 5.2 – État de l'art des méthodes de SLAM visuel dynamique

Méthode	Année	Caméra	Carte	Stratégie	Architecture	Avantages
RDSLAM [Tan+13]	2013	Mono	Éparse statique	RANSAC adaptatif avec les points 3D	SIFT GPU	Résistant aux changements de texture
Detect-SLAM [Zho+18]	2018	Mono	Niveau Instance	Détecteur d'objets pour sélectionner des points dynamiques	DNN	Résistant au flou et variation d'éclairage
DS-SLAM [Yu+18]	2018	Mono	Dense 3D octo-tree	Segmentation sémantique pour rejeter objets dynamiques	ORB-SLAM2 + SegNet [BKC16]	Améliore l'estimation de la pose
DynaSLAM [Bes+18]	2018	All	Statique	Segmentation sémantique des objets dynamiques	ORB-SLAM2 + Mask-RCNN [He+18]	peint l'arrière-plan occlus
DMS-SLAM [Liu+19a]	2019	All	3D globale statique	Combine GMS et fenêtre glissante	ORB-SLAM2 +GMS	Obtenir les points 3D statiques appariés
ROBIO-SLAM [Liu+19b]	2019	All	Statique	Reconnaissance d'objet avec le flux optique	ORB-SLAM2 + YOLOv3 [RF18]	Rejet des points dynamiques
DM-SLAM [Che+20]	2020	All	Éparse	Segmentation d'instance et flux optique	ORB-SLAM2 + Mask-RCNN	Estimation de pose robuste
vSLAM [Ai+20]	2020	Stereo, RGBD	point 3D Dense	Modèle de détection d'objet et de probabilitél	ORB-SLAM2 + YOLOv4 [BWL20]	Rejète les objets dynamiques
DynaSLAM2 [Bes+20]	2020	Stereo, RGBD	Éparse statique	Trajectoires caméra et objets dynamiques	ORB-SLAM2 + Mask-RCNN	Donne la pose de la caméra et le MDOT

Caméra : systèmes de vision monoculaire, stéréo ou RGBD, CNN : Convolutional Neural Network, Deep Neural Network, R-CNN : Recurrent Convolutional Neural Network, DO : Dynamic Objects, MDoT : Trajectoires cartographiques et objets dynamiques.

5.3 . SLAM visuel dynamique

La prise en compte des dynamiques de scène dans les algorithmes de SLAM visuel est un critère indispensable au déploiement de robot autonome en environnement réel. Ces algorithmes dits "dynamique" nécessitent l'emploi de méthodes et concepts pluri-disciplinaires afin de pouvoir considérer les dynamiques de scène dans les processus de localisation et de cartographie. Nous détaillons donc par la suite le principe de fonctionnement et les évolutions du SLAM visuel dynamique. Nous soumettrons ensuite des hypothèses liées à des observations qui seront alors justifiées et qui permettront de proposer une approche innovante permettant une meilleure estimation de l'état des points d'intérêt (statique/dynamique), et en conséquence, de meilleures performances.

5.3.1 . Principe

Les méthodes de SLAM visuel dynamique reposent sur la prise en compte des dynamiques de scène dans les modules de suivi de points d'intérêt, d'estimation de pose de la caméra et de cartographie de l'environnement. Ces méthodes tentent de distinguer les points d'intérêt qui correspondent à des éléments statiques de la scène, de ceux qui correspondent à des éléments en mouvements entre les différentes images. Cette distinction permet de rejeter les points considérés dynamiques des modules du SLAM afin de n'estimer la pose de la caméra et de ne cartographier l'environnement qu'avec les points dont l'état est considéré statique. Grâce à ce procédé, nous réduisons l'erreur d'estimation de la localisation ce qui rend alors le système plus robuste dans des environnements réels qui contiennent généralement des dynamiques de scène. De manière générale, il existe deux approches pour déterminer quels points d'intérêt sont dynamiques, les méthodes géométriques et les méthodes par apprentissage via la segmentation et/ou détection d'objet :

- **Méthodes géométriques :** Elles impliquent l'estimation de la dynamique des points en effectuant des calculs basés sur la géométrie projective et épipolaire. Ils tentent d'estimer la distance du point correspondant d'une image antérieur par rapport à la projection de la droite épipolaire de ce point dans l'image actuelle. La distance entre le point correspondant de l'image actuelle et la droite épipolaire du point projeté de l'image précédente indique la dynamique du point. Une distance élevée impliquant un mouvement importante, et inversement.
- **Méthodes par apprentissage :** L'estimation des dynamiques par l'apprentissage se fait par l'introduction d'hypothèses concernant la probabilité de mouvement liée à la nature des objets présents dans la scène. Par ce procédé, nous effectuons une segmentation de scène où nous

estimerons qu'une classe d'objets donnée est directement considérée comme dynamique. Cette approche peut être complétée par des méthodes d'apprentissage réalisant l'estimation du flux optique de l'image ou un raffinement de l'estimation des dynamiques de scènes par l'emploi de méthodes géométriques.

L'état des points d'intérêt (dynamique ou statique) est déterminé avant le module de suivi de sorte qu'il puisse s'exécuter sans prendre en compte les points dynamiques. Par effet de cascade, le rejet des points dynamiques améliore les performances et la fiabilité du suivi qui à son tour améliore la fiabilité de l'estimation de pose de la caméra et de la cartographie de l'environnement à travers le processus d'ajustement des faisceaux (BA).

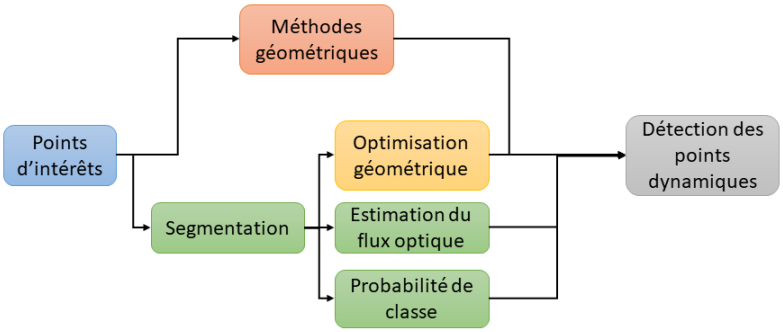


Figure 5.4 – Stratégies d'estimation de la dynamique des points.

Pour atteindre une détection fiable des points dynamiques, il existe plusieurs combinaisons d'approches illustrées dans la figure 5.4. L'approche géométrique repose sur la comparaison entre les coordonnées image du projeté d'un point d'intérêt de l'image précédente dans l'image actuelle et le point de l'image actuelle correspondant à celui de la précédente. Ce procédé permet alors de déterminer la nature des points à étudier. Notons que seuls les points d'intérêt qui ont des correspondants d'une image à l'autre sont retenus dans le module de suivi et donc d'estimation de l'état des points d'intérêt.

Afin d'atteindre cet objectif, il est indispensable de passer d'une hypothèse de scène statique à une hypothèse de scène dynamique dans la méthode d'estimation de la pose de la caméra. La figure 5.5 illustre les hypothèses avec les cas de points statique et dynamique. Le point réel p_{t-1} est projeté dans le plan image I_{t-1} correspondant à sa trame temporelle pour fournir le point image $m_{t-1,j}$, et également dans le plan image I_t de l'image suivante fournissant le point image $m'_{t-1,j}$. Entre le moment t et $t-1$ la caméra s'est déplacée selon la matrice $H_{t-1,t}$ et le point p^j_{t-1} selon la matrice $R^j_{t-1,t}$ pour devenir le point réel p^j_t et son projeté image $m_{t,j}$. Nous avons également représenté la

ligne épipolaire $l_{m_{t-1}}$ correspondant à la projection épipolaire du point p^j_{t-1} dans le plan image I_t. La distance minimale entre la ligne épipolaire $l_{m_{t-1}}$ et le point image m_t correspondant du point m_{t-1} au moment t est notée Δ_d et est représentative du mouvement réelle effectué par le point réel p entre les images des moments $t-1$ et t.

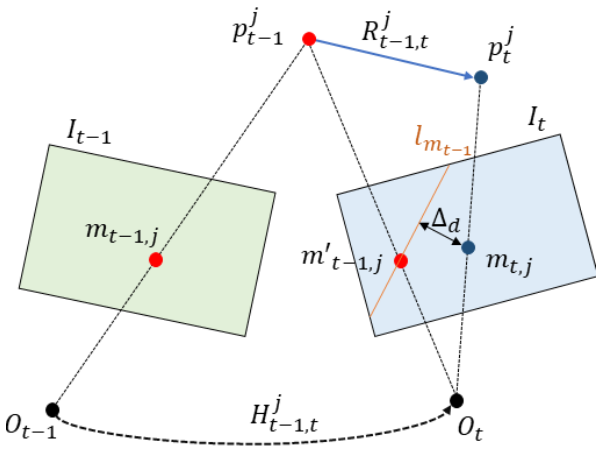


Figure 5.5 – Hypothèse de scène pour la projection de points statique et dynamique.

En utilisant les notions de géométrie projective et épipolaire, nous pouvons formaliser l'erreur de reprojection $e_{t,j}(\epsilon)$ dans un premier temps pour l'hypothèse de scène statique :

$$e_{t,j}(\epsilon) = m_{t,j} - \pi(H^w_t(\epsilon), p^j_{t-1}) \tag{5.1}$$

Dans le cas d'une scène statique où nous considérons le point réel statique p^j_{t-1} du repère caméra, l'erreur de reprojection entre ses correspondants aux images $t-1$ et t selon la fonction de projection π est nulle. De ce fait, l'algorithme de BA utilise l'erreur de reprojection de l'équation 5.1 pour optimiser l'estimation de la pose et la position des points dans la carte en l'appliquant à la fonction de coût suivante :

$$C = \sum_{t,j} \rho_h(e_{t,j}(\epsilon)^T \Omega^{-1}_{t,j} e_{t,j}(\epsilon)) \tag{5.2}$$

Rappelons que $\Omega^{-1}_{t,j}$ est la matrice de covariance et ρ_h est la fonction Huber Kernel. À l'inverse, dans une scène dynamique où nous considérons que le point réel p^j_{t-1} du repère caméra s'est déplacé d'un mouvement inconnu noté R^t_{t-1} pour devenir le point p^j_t, l'erreur de reprojection sera non nulle et aura la formulation suivante :

$$e_{t,j}(\epsilon) = m_{t,j} - \pi(H^w_t(\epsilon), p^j_t) = m_{t,j} - \pi(H^w_t(\epsilon), R^t_{t-1} p^j_{t-1})$$

La valeur de l'erreur de reprojection, qui correspond à la distance entre la droite épipolaire du point $m_{t-1,j}$ et son projeté $m_{t,j}$ à l'instant t, décrit l'importance du mouvement du point réel m entre les deux images successives. L'incertitude liée à l'état du point considéré a mené à ajouter une composante probabiliste à la fonction de coût précédente 5.2. Le poids W_j est incorporé à la fonction pour indiquer l'état du point de manière binaire (1 pour dynamique et 0 pour statique, ou inversement) :

$$C = \sum_{t,j} W_j \rho_h(e_{t,j}(\epsilon)^T \Omega_{t,j}^{-1} e_{t,j}(\epsilon))$$

Les points considérés comme dynamiques sont alors soit directement rejetés du reste du processus ou bien affinés et traités dans le temps à l'aide d'approches probabilistes.

5.3.2 . Approches probabilistes

De nombreuses études ont intégré des fonctions probabilistes pour affiner l'estimation de l'état des points et les catégoriser. Le poids probabiliste W_j est alors nuancé selon la grille de valeur suivante :

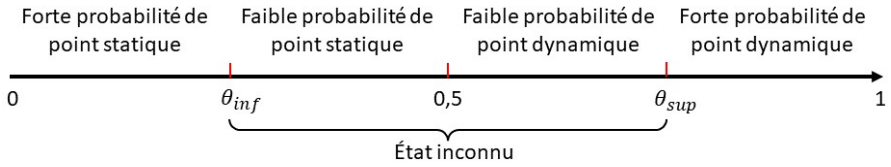


Figure 5.6 – Les différents états des points d'intérêt par probabilité de mouvement.

L'état des points étudiés sont alors répartis en quatre catégories illustrées dans la figure 5.6. Les méthodes d'estimation de l'incertitude de l'état varient d'une étude à l'autre, mais mettent très souvent en place des stratégies de sélection des points en fonction de leur état et/ou une mise à jour de la probabilité de mouvement des points par l'usage d'un filtre Bayésien. La première catégorie de méthode applique une stratégie de sélection des types d'états de point en fonction des contraintes résultantes des situations et besoins de chacun des modules du SLAM (suivi de points, estimation de pose et cartographie).

Cette catégorie, la plus rudimentaire, applique une stratégie d'élargissement des ensembles d'états utilisés selon les besoins. Ainsi, sont retenus en premier lieu uniquement les points considérés statiques, puis si besoin, les points ayant un état inconnu, et enfin les points considérés dynamiques. Cette approche permet de contourner le problème lié au rejet des points dyna-

miques qui peut conduire à un nombre insuffisant de points pour effectuer le suivi de point ainsi que le BA pour l'estimation de pose et la cartographie. Malheureusement, elle ne garantit pas une estimation très fiable au cours du temps de l'état des points.

Pour résoudre ce problème, les filtres Bayésiens sont utilisés pour mettre à jour au cours des images les états de chacun des points et donc étaler l'incertitude d'estimation d'état. Ce filtrage sur plusieurs images rectifie les erreurs d'estimation avant de considérer un point comme dynamique et donc de le rejeter. L'ensemble de ces approches permet alors de rendre plus robuste les systèmes de SLAM visuels aux dynamiques de scène, mais reste toujours limités par :

- La qualité de la mise en correspondance de points d'intérêt entre les images successives. Une mauvaise mise en correspondance entraîne de fait un suivi de point d'intérêts incohérent et donc une mauvaise estimation de la pose et cartographie.
- La précision de segmentation du modèle qui s'applique aux classes d'objets considérées comme potentiellement en mouvement. En effet, une segmentation imprécise impliquera la prise en compte de points statiques comme étant dynamiques, car compris dans le masque généré par la segmentation, et inversement, des points dynamiques de l'objet en mouvement comme étant statiques car exclus du masque.
- Le nombre de points retenus après le processus d'estimation de la dynamique des points et leur rejet. Un nombre trop petit de points mis en correspondance entre deux images successives entraînera une diminution de l'efficacité du module de suivi ou son dysfonctionnement.

La plupart des méthodes s'efforcent de surmonter ces limitations en employant des lois probabilistes dépendant essentiellement de la géométrie épipolaire à travers la prise en compte de la distance minimale Δ_d entre la ligne épipolaire d'un point d'une image précédente et son point correspondant dans l'image actuelle (voir figure 5.5).

Une variante consiste à ajouter une composante probabiliste à l'étude des points situés dans les masques générés et proches de leurs bords. En effet, il arrive que certains points présents dans une zone segmentée soient d'abord estimés comme dynamique en ne prenant en compte que le modèle géométrique. Un processus de vérification reprend l'emploi de modèle probabiliste lié à une distance, avec dans ce cas, la distance minimale entre le bord du masque et le point incertain, pour ajuster son estimation. Ces méthodes ont le mérité d'optimiser l'estimation de la dynamique des points, mais manquent de mise en perspective avec le contexte de scène.

5.4 . Hypothèses et formulations mathématiques

5.4.1 . Hypothèses

Les méthodes d'estimation de la dynamique des points examinés ne sont pas en corrélation avec les informations de scène. En effet, ces méthodes se réfèrent uniquement à des informations pixels qui ont l'avantage d'être faciles à exploiter, mais pas suffisamment fiables car en 2D. Il semble alors pertinent de remettre en perspective les informations dans leur contexte réel qui s'appuie sur la géométrie 3D. À ce titre, nous avons formulé deux hypothèses liées au contexte de scène qui enrichit l'estimation probabiliste du mouvement de points.

Hypothèse 1 *La profondeur de scène peut influer sur la fiabilité de l'estimation de points dynamiques. Nous considérons ici la profondeur de scène comme étant la distance séparant le point réel de la caméra. En conclusion, d'après la géométrie projective, plus un point est loin, moins il est fiable.*

Prendre en compte cette hypothèse fait sens afin de surmonter les limitations évoquées précédemment 5.3.2. En effet, il est avéré que plus un objet est loin de la caméra, plus sa représentation image sera petite et plus les détails sont atténués. Ainsi, nous constatons les conséquences suivantes :

- Plus un objet est distant, plus sa représentation image est petite, et plus un objet est représenté petit, plus la segmentation perd en précision.
- De même, plus un objet est distant, plus sa représentation image est petite ce qui provoque de fait une résolution pixel plus faible et donc une réduction de la fiabilité d'estimation du dynamisme.

Les performances de l'ensemble des modèles de détection et segmentation d'objet confirment le premier constat. En effet, les scores AP_S, correspondant à l'application de ces modèles sur des objets de petite taille et donc possiblement lointains, sont relativement faible tandis que les scores AP_L pour les gros objets souvent plus proches ont les meilleurs scores. Le deuxième constat peut être confirmé par l'utilisation de la géométrie projective dans des cas décrits dans les sous-sections suivantes 5.4.2 et5.4.3.

Hypothèse 2 *Plus un point image correspond à un point réel distant, plus l'erreur de quantification spatiale aura un impact sur l'estimation de la dynamique de ce point.*

L'erreur de quantification spatiale est le phénomène résultant de la capacité des capteurs CCD des caméras à digitaliser les ondes lumineuses dans des éléments spatiaux quantifiés avec des dimensions finies. L'erreur de quantification spatiale intervient lors du passage des coordonnées physiques de

l'image sur le capteur, à l'approximation des coordonnées pixels correspondantes.

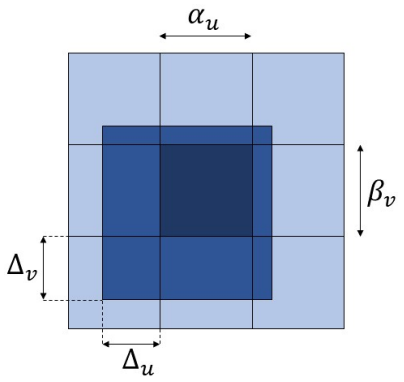


Figure 5.7 – Erreur de quantification spatiale.

La figure 5.7 représente le phénomène physique du passage des coordonnées capteurs aux coordonnées pixels. Le carré central représente le pixel qui correspond de façon certaine à l'image réelle, le bleu foncé à l'image réelle qui s'étale sur plusieurs cellules du capteur et enfin le bleu clair l'ensemble des pixels qui peuvent potentiellement être affecté à la représentation de l'image réelle. δ_{uu} et δ_{vv} désignent respectivement la taille horizontale et verticale des pixels, et Δ_u et Δ_v les dépassements de l'image réelle sur les pixels incertains. Ainsi, tout élément digitalisé sous forme de pixel contient une erreur de quantification de $\left[\frac{-1}{2}; \frac{1}{2}\right]$ pixel. En considérant les hypothèses 1 et 2, nous pouvons formuler que la profondeur de scène a un impact sur la fiabilitié des processus de segmentation d'image et de raisonnement par contrainte géométrique épipolaire.

5.4.2 . Effet de la profondeur sur la représentation image.

Afin de pouvoir intégrer la profondeur dans la détermination de l'état des points d'intérêt, il est indispensable de formaliser mathématiquement l'impact de la profondeur sur la représentation pixel. Dans ce but, nous avons établi un cas de référence dans la figure 5.8 où sont illustrés les points diagonaux de la face visible d'un cube dans le repère caméra (p_1 et p_2) et leurs projetés respectifs dans le repère image (m_1 et m_2).

Dans un premier temps, le cube de taille Δ_x et Δ_y est placé à une distance Δ_z de la caméra puis déplacé à une distance Δ'_z. Nous obtenons alors les points images correspondant m_1 et m_2 qui représentent la projection 2D du cube de taille Δ_u et Δ_v pour le cube à une distance Δ_z puis Δ'_u et Δ'_v pour le cube à la distance Δ'_z. Notons également que les coordonnées des points images et réels sont respectivement de la forme $m_i = (u_i \quad v_i)^T$ et $p_i =$

$(x_i \quad y_i \quad z_i)^T$, et que leurs correspondants après déplacement du cube d'une distance Δ'_z sont notés m'_i et p'_i.

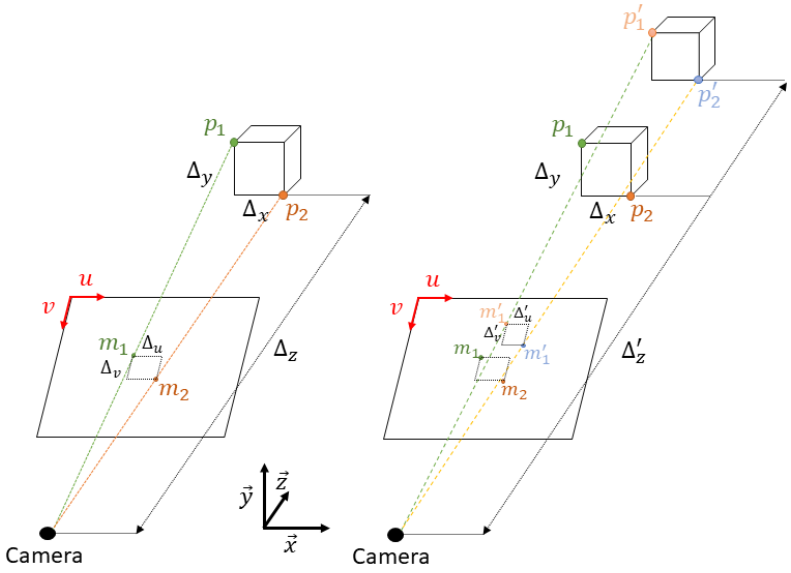


Figure 5.8 – Impact de la profondeur sur la représentation pixel.

Nous pouvons alors exprimer les coordonnées pixels de chacun des points de la manière suivante, f étant la focale de la caméra :

$$\begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = \begin{pmatrix} f\delta_u & 0 & u_0 \\ 0 & f\delta_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{x_1}{\Delta_z} \\ \frac{y_1}{\Delta_z} \\ 1 \end{pmatrix} \tag{5.3}$$

$$\begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} = \begin{pmatrix} f\delta_u & 0 & u_0 \\ 0 & f\delta_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{x_2}{\Delta_z} \\ \frac{y_2}{\Delta_z} \\ 1 \end{pmatrix} = \begin{pmatrix} f\delta_u & 0 & u_0 \\ 0 & f\delta_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{x_1+\Delta_x}{\Delta_z} \\ \frac{y_1+\Delta_y}{\Delta_z} \\ 1 \end{pmatrix} \tag{5.4}$$

Nous pouvons alors exprimer Δ_u et Δ_v en fonction de u_1, u_2, v_1 et v_2 à l'aide des systèmes 5.3 et 5.4. Réécrivons tout d'abord de manière plus formelle les équations de u_1, u_2, v_1 et v_2 :

$$\begin{cases} u_1 &= f\delta_u \frac{x_1}{\Delta_z} + u_0 \\ v_1 &= f\delta_v \frac{y_1}{\Delta_z} + v_0 \\ u_2 &= f\delta_u \frac{x_1+\Delta_x}{\Delta_z} + u_0 \\ v_2 &= f\delta_v \frac{y_1+\Delta_y}{\Delta_z} + v_0 \end{cases} \tag{5.5}$$

Ce qui nous permet d'exprimer la taille pixel du cube à travers Δ_u et Δ_v de la manière suivante :

$$\begin{cases} \Delta_u &= |u_1 - u_2| &= f\delta_u \frac{\Delta_x}{\Delta_z} \\ \Delta_v &= |v_1 - v_2| &= f\delta_v \frac{\Delta_y}{\Delta_z} \end{cases} \tag{5.6}$$

L'expression de la taille réelle du cube (Δ_x et Δ_y) en fonction de la taille pixel (Δ_u et Δ_v), la distance à la caméra Δ_z et les paramètres intrinsèques de la caméra (f et δ_u) est de la forme suivante :

$$\left\{ \begin{array}{rcl} \Delta_x & = & \Delta_u \Delta_z f \delta_u \\ \Delta_y & = & \Delta_v \Delta_z f \delta_v \end{array} \right. \tag{5.7}$$

Il est maintenant nécessaire d'exprimer les coordonnées des points m'_1 et m'_2 afin d'en extraire l'expression de la taille du projeté du cube dans le plan image à travers les longueurs Δ'_u et Δ'_v en fonction des expressions 5.6 et 5.7 :

$$\begin{pmatrix} u'_1 \\ v'_1 \\ 1 \end{pmatrix} = \begin{pmatrix} f\delta_u & 0 & u_0 \\ 0 & f\delta_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{x_1}{\Delta'_z} \\ \frac{y_1}{\Delta'_z} \\ 1 \end{pmatrix} \tag{5.8}$$

$$\begin{aligned} \begin{pmatrix} u'_2 \\ v'_2 \\ 1 \end{pmatrix} = \begin{pmatrix} f\delta_u & 0 & u_0 \\ 0 & f\delta_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{x_2}{\Delta'_z} \\ \frac{y_2}{\Delta'_z} \\ 1 \end{pmatrix} &= \begin{pmatrix} f\delta_u & 0 & u_0 \\ 0 & f\delta_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{x_1+\Delta_x}{\Delta'_z} \\ \frac{y_1+\Delta_y}{\Delta'_z} \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} f\delta_u & 0 & u_0 \\ 0 & f\delta_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{x_1+\Delta_u \Delta_z f \delta_u}{\Delta'_z} \\ \frac{y_1+\Delta_v \Delta_z f \delta_v}{\Delta'_z} \\ 1 \end{pmatrix} \end{aligned} \tag{5.9}$$

En remplaçant Δ_x et Δ_y par leurs expressions respectives de l'équation 5.7, nous pouvons réécrire u'_2 et v'_2 comme :

$$\begin{pmatrix} u'_2 \\ v'_2 \\ 1 \end{pmatrix} = \begin{pmatrix} f\delta_u & 0 & u_0 \\ 0 & f\delta_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{x_1+\Delta_u \Delta_z f \delta_u}{\Delta'_z} \\ \frac{y_1+\Delta_v \Delta_z f \delta_v}{\Delta'_z} \\ 1 \end{pmatrix} \tag{5.10}$$

Les expressions de u'_1, v'_1, u'_2 et v'_2 sont alors de la forme suivante :

$$\left\{ \begin{array}{rcl} u'_1 & = & f\delta_u \frac{x_1}{\Delta'_z} + u_0 \\ v'_1 & = & f\delta_v \frac{y_1}{\Delta'_z} + v_0 \\ u'_2 & = & f\delta_u (\frac{x_1+\Delta_u \Delta_z f \delta_u}{\Delta'_z}) + u_0 \\ v'_2 & = & f\delta_v (\frac{y_1+\Delta_v \Delta_z f \delta_v}{\Delta'_z}) + v_0 \end{array} \right. \tag{5.11}$$

Il est alors possible d'exprimer les nouvelles longueurs pixels du cube Δ'_u et Δ'_v en fonction des distances du cube à la caméra Δ_z et Δ'_z :

$$\left\{ \begin{array}{rcl} \Delta'_u & = & |u'_1 - u'_2| \quad = \quad \Delta_u f^2 \delta_u^2 \frac{\Delta_z}{\Delta'_z} \\ \Delta'_v & = & |v'_1 - v'_2| \quad = \quad \Delta_v f^2 \delta_v^2 \frac{\Delta_z}{\Delta'_z} \end{array} \right. \tag{5.12}$$

La longueur pixel d'un même segment projeté d'un segment réel ayant subi un éloignement Δ'_z de la caméra est alors proportionnelle à cet éloignement selon le rapport : $\frac{\Delta_z}{\Delta'_z}$. Ainsi, d'après l'équation 5.12, lorsque Δ_z est supérieur à Δ'_z (ce qui correspond à un rapprochement) alors Δ'_u et Δ'_v augmentent proportionnellement au rapport $\frac{\Delta_z}{\Delta'_z}$. A l'inverse, lorsque Δ_z est inférieur à Δ'_z alors Δ'_u et Δ'_v diminuent proportionnellement.

5.4.3 . Effet de l'erreur de quantification spatiale et de la profondeur

L'erreur de numérisation, aussi appelée erreur de quantification spatiale dans le domaine de la vision par ordinateur, est le phénomène résultant de la conversion d'un flux lumineux photonique par les capteurs CCD lors de l'acquisition d'image et sa conversion en information pixel. Des travaux ont tenté de quantifier cette erreur de conversion [Kam89 ; KS89] pour déterminer la probabilité d'état des cellules des capteurs CCD. À l'aide de ces travaux, nous avons formalisé l'impact de l'erreur de quantification spatiale d'un pixel sur sa projection dans le repère caméra en fonction de la profondeur de scène. Dans notre cas, nous considérons les valeurs de profondeurs z_i et z'_i connues. Rappelons que l'erreur de quantification spatiale est estimée à un demi-pixel horizontalement et verticalement. Nous projetons cette erreur sur deux plans parallèles distants de Δ_d en considérant les surfaces de certitude de couleurs foncées et les surfaces d'incertitude de couleurs claires. Nous obtiendrons alors des valeurs maximales et minimales décrivant l'impact de l'erreur de quantification en fonction de la profondeur. Soit m_i=$(u_i\ v_i\ 1)^T$ le point image de taille un pixel et ses points réelles correspondant notés p_i=$(x_i\ y_i\ z_i)^T$ et p'_i=$(x'_i\ y'_i\ z'_i)^T$.

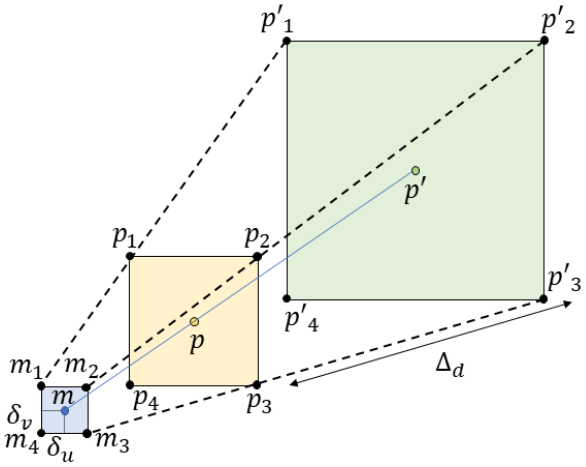


Figure 5.9 – Légende de mon image.

En utilisant la géométrie projective nous pouvons écrire les coordonnées du point m :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f\delta_u & 0 & u_0 \\ 0 & f\delta_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{x}{z} \\ \frac{y}{z} \\ 1 \end{pmatrix} = \begin{pmatrix} f\delta_u & 0 & u_0 \\ 0 & f\delta_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{x'}{z'} \\ \frac{y'}{z'} \\ 1 \end{pmatrix} \tag{5.13}$$

Une manière plus formelle d'écrire l'équation 5.13 permet d'extraire plus faci-

lement les coordonnées images :

$$\left\{ \begin{array}{rcccl} u & = & f\delta_u \frac{x}{z} + u_0 & = & f\delta_u \frac{x'}{z'} + u_0 \\ v & = & f\delta_v \frac{y}{z} + v_0 & = & f\delta_v \frac{y'}{z'} + v_0 \end{array} \right. \tag{5.14}$$

Or le phénomène d'erreur de quantification spatiale qui s'opère au niveau des pixels induit une incertitude de valeur comprise dans l'intervalle : $\left[\frac{-\delta}{2}; \frac{\delta}{2}\right]$. Nous obtenons alors les intervalles de confiance suivants pour les coordonnées pixels du point m :

$$\left\{ \begin{array}{ccccc} u - \frac{\delta_u}{2} & \leq & u & \leq & u + \frac{\delta_u}{2} \\ v - \frac{\delta_v}{2} & \leq & v & \leq & v + \frac{\delta_v}{2} \end{array} \right. \tag{5.15}$$

Ainsi, il est alors possible d'exprimer les coordonnées image des points m_1, m_2, m_3 et m_4 résultant du phénomène de quantification spatiale en fonction des coordonnées du point m.

$$m_1 = \begin{pmatrix} u - \frac{\delta_u}{2} \\ v - \frac{\delta_v}{2} \\ 1 \end{pmatrix}, \; m_2 = \begin{pmatrix} u + \frac{\delta_u}{2} \\ v - \frac{\delta_v}{2} \\ 1 \end{pmatrix}, \; m_3 = \begin{pmatrix} u + \frac{\delta_u}{2} \\ v + \frac{\delta_v}{2} \\ 1 \end{pmatrix}, \; m_4 = \begin{pmatrix} u - \frac{\delta_u}{2} \\ v + \frac{\delta_v}{2} \\ 1 \end{pmatrix} \tag{5.16}$$

Maintenant, toutes les coordonnées du point m précédemment formulées ainsi que leur correspondant lié aux incertitude m_i nous permettent de déterminer les coordonnées réelles du point p et de ses équivalent p_i qui prennent en compte l'erreur de quantification spatiale. Extrayons tout d'abord l'équation correspondante au point p de l'équation 5.13 ce qui permettra d'exprimer les points d'incertitudes p_i en fonction de p :

$$\left\{ \begin{array}{rcl} x & = & \frac{(u-u_0)z}{f\delta_u} \\ y & = & \frac{(v-v_0)z}{f\delta_v} \end{array} \right. \tag{5.17}$$

De la même manière, nous exprimons les coordonnées p_i à l'aide des équations 5.13 et 5.17 :

$$\begin{array}{cccc} \left\{ \begin{array}{rcl} x_1 & = & \frac{(u_1-u_0)z}{f\delta_u} \\ y_1 & = & \frac{(v_1-v_0)z}{f\delta_v} \end{array} \right., & \left\{ \begin{array}{rcl} x_2 & = & \frac{(u_2-u_0)z}{f\delta_u} \\ y_2 & = & \frac{(v_2-v_0)z}{f\delta_v} \end{array} \right., \\ \left\{ \begin{array}{rcl} x_3 & = & \frac{(u_3-u_0)z}{f\delta_u} \\ y_3 & = & \frac{(v_3-v_0)z}{f\delta_v} \end{array} \right., & \left\{ \begin{array}{rcl} x_4 & = & \frac{(u_4-u_0)z}{f\delta_u} \\ y_4 & = & \frac{(v_4-v_0)z}{f\delta_v} \end{array} \right. \end{array} \tag{5.18}$$

Afin d'exprimer les coordonnées p_i en fonction de p, il suffit de remplacer les valeurs u_i et v_i par leurs expressions correspondantes provenant de l'équation 5.16.

$$\left\{ \begin{array}{rcccl} x_1 & = & \frac{(u_1-u_0)z}{f\delta_u} & = & \frac{(u-u_0-\frac{\delta_u}{2})z}{f\delta_u} \quad = \quad x - \frac{z}{2f} \\ y_1 & = & \frac{(v_1-v_0)z}{f\delta_v} & = & \frac{(v-v_0-\frac{\delta_v}{2})z}{f\delta_v} \quad = \quad y - \frac{z}{2f} \end{array} \right. \tag{5.19}$$

Par analogie les expressions de p_2, p_3 et p_4 s'écrivent :

$$\left\{\begin{array}{lcl} x_2 &=& x+\frac{z}{2f} \\ y_2 &=& y-\frac{z}{2f} \end{array}\right. , \quad \left\{\begin{array}{lcl} x_3 &=& x+\frac{z}{2f} \\ y_3 &=& y+\frac{z}{2f} \end{array}\right. , \quad \left\{\begin{array}{lcl} x_4 &=& x-\frac{z}{2f} \\ y_4 &=& y+\frac{z}{2f} \end{array}\right. \tag{5.20}$$

Pour le cas du plan P', nous procédons aux mêmes étapes en considérant que $z'_i = z_i + \Delta_d$. Nous exprimons d'abord les coordonnées du points p' en fonction de celle du point p :

$$\left\{\begin{array}{lcl} x' &=& \frac{(u-u_0)(z'+\Delta_d)}{f\delta_u} \quad = \quad x+\frac{(u-u_0)\Delta_d}{f\delta_u} \\ y' &=& \frac{(v-v_0)(z'+\Delta_d)}{f\delta_v} \quad = \quad y+\frac{(v-v_0)\Delta_d}{f\delta_v} \end{array}\right. \tag{5.21}$$

Par le même procédé, nous exprimons les coordonnées du point p'_1 en fonction de son projeté image m_1 pour le formuler selon p :

$$\left\{\begin{array}{lcl} u_1 &=& f\delta_u \frac{x'_1}{z'+\Delta_d} + u_0 \\ v_1 &=& f\delta_v \frac{y'_1}{z'+\Delta_d} + v_0 \end{array}\right. \tag{5.22}$$

$$\left\{\begin{array}{lcccl} x'_1 &=& \frac{(u_1-u_0)(z'+\Delta_d)}{f\delta_u} &=& \frac{(u-\frac{\delta_u}{2}-u_0)(z'+\Delta_d)}{f\delta_u} \quad = \quad x+\frac{(u-u_0)\Delta_d}{f\delta_u}-\frac{z+\delta_d}{2f} \\ y'_1 &=& \frac{(v_1-v_0)(z'+\Delta_d)}{f\delta_v} &=& \frac{(v-\frac{\delta_v}{2}-v_0)(z'+\Delta_d)}{f\delta_v} \quad = \quad y+\frac{(v-v_0)\Delta_d}{f\delta_v}-\frac{z+\delta_d}{2f} \end{array}\right. \tag{5.23}$$

Et enfin, également par analogie, nous pouvons exprimer les coordonnées de p'_2, p'_3 et p'_4 :

$$\begin{array}{l} \left\{\begin{array}{lcl} x'_2 &=& x+\frac{(u-u_0)\Delta_d}{f\delta_u}+\frac{z+\delta_d}{2f} \\ y'_2 &=& y+\frac{(v-v_0)\Delta_d}{f\delta_v}-\frac{z+\delta_d}{2f} \end{array}\right. \\ \left\{\begin{array}{lcl} x'_3 &=& x+\frac{(u-u_0)\Delta_d}{f\delta_u}+\frac{z+\delta_d}{2f} \\ y'_3 &=& y+\frac{(v-v_0)\Delta_d}{f\delta_v}+\frac{z+\delta_d}{2f} \end{array}\right. \\ \left\{\begin{array}{lcl} x'_4 &=& x+\frac{(u-u_0)\Delta_d}{f\delta_u}-\frac{z+\delta_d}{2f} \\ y'_4 &=& y+\frac{(v-v_0)\Delta_d}{f\delta_v}+\frac{z+\delta_d}{2f} \end{array}\right. \end{array} \tag{5.24}$$

Ces relations mathématiques permettent de formuler l'effet linéaire de la profondeur sur la taille du champ de projection et ainsi sur l'impact de l'erreur de quantification spatiale. Ces formulations mathématiques peuvent aussi être interprétées dans le cadre de la projection d'un élément image carré de taille (δ_u, δ_v) à différentes profondeurs pour observer le champ de projection de l'objet.

5.5 . Méthode

La plupart des méthodes de vSLAM dynamique exploite le framework ORB-SLAM3 comme algorithme de base pour ajouter leurs contributions sous la forme de modules supplémentaires ou d'optimisations. Ce framework présente l'avantage d'être en source ouverte, performant et implémenté pour de nombreuses configurations de système de caméra, dont les systèmes RGB-D (caméra couleurs avec également caméra de profondeur). Nous y avons alors incorporé des modules supplémentaires permettant de détecter et de rejeter les points d'intérêt considérés comme ayant un état dynamique. L'architecture globale de notre système illustrée par la figure 5.10 est composée d'un module de segmentation sémantique des objets et d'un module géométrique basé sur le raisonnement par contrainte géométrique épipolaire.

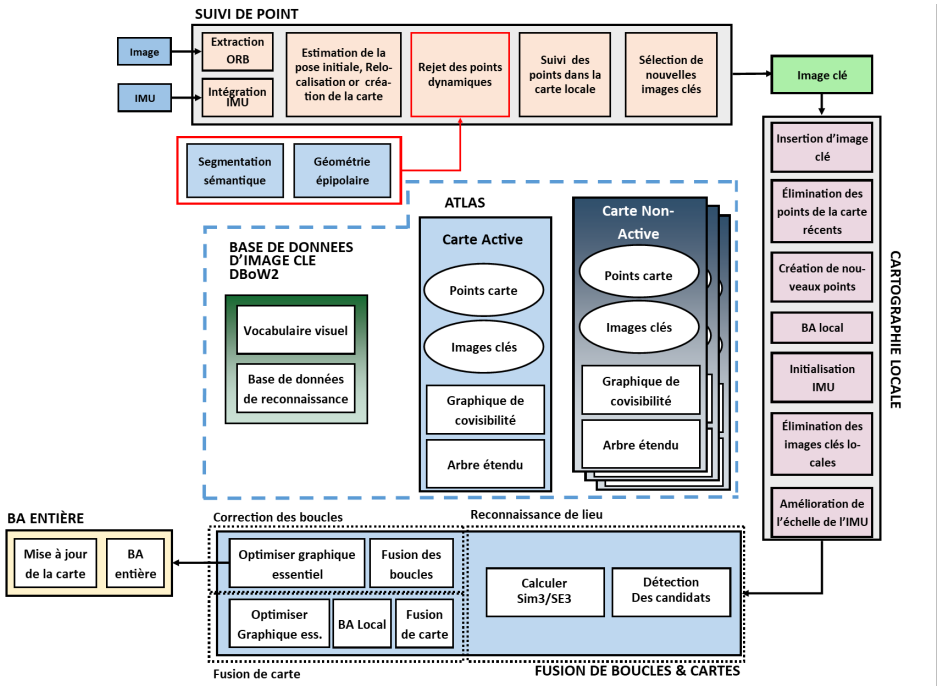


Figure 5.10 – Architecture du SLAM dynamique pour le framework ORB-SLAM3.

De nombreux travaux utilisent des variantes plus ou moins importantes de cette architecture. Notre premier module effectue une segmentation sémantique des objets définis comme potentiellement en mouvement et donc dynamique de par leur nature. Ainsi, en environnement intérieur, l'ensemble de la communauté scientifique s'accorde sur le fait que seul l'humain peut-être considéré comme dynamique et donc être segmenté de fait. En environnement extérieur, davantage d'objets sont considérés dynamiques (personne,

véhicule, animal, etc...) ce qui élargit le champ de considération.
Ce premier module permet de définir des zones de certitude où les points d'intérêt sont dynamiques pour l'ensemble des images. En effet, la nature des objets considérés dynamiques implique de fait la présence de mouvements de manière continuelle ou ponctuelle. Ces objets étant des corps non-rigides, il est alors nécessaire de considérer l'ensemble de leurs formes comme dynamique.
Le second module, le module géométrique, vérifie par raisonnement géométrique l'état des points d'intérêt mis en correspondance entre deux images successives. En se basant sur les hypothèses précédentes 1 et 2 , nos deux modules intègrent l'impact de la profondeur dans l'estimation de l'état des points d'intérêt, ce qui permet d'affiner l'estimation d'état des points.

Cependant, ces deux modules présentent des limites puisqu'ils ne permettent pas de pouvoir considérer l'ensemble des points d'intérêt d'une image. En effet, le module de segmentation sémantique ne fournit d'information que sur l'état des points d'intérêt contenus dans les objets segmentés, ce qui limite de fait son champ d'application. Le module géométrique permet d'étudier l'ensemble de l'image, mais est limité uniquement aux points d'intérêt qui ont un correspondant entre deux images successives. Il nous a semblé indispensable de trouver un moyen d'estimer l'état de points d'intérêt en dehors de ces deux cadres. Nous avons établi le constat qu'en environnement intérieur, la grande majorité des dynamiques existantes sont la résultante d'une action humaine. Ainsi, des objets inertes (chaises, livres, etc.) ne peuvent être en mouvement que s'ils sont en contact avec un humain et donc en interaction avec lui. Partant de ce constat, nous avons proposé une architecture d'algorithme de SLAM dynamique qui surmonte ces deux limitations.

5.5.1 . Architecture

Notre méthode de vSLAM dynamique est construite sur le framework ORB-SLAM3 comme illustré dans la figure 5.11. Elle y intègre trois modules distincts, respectivement le **module géométrique**, le **module de segmentation sémantique** et le **module d'interaction des objets**, qui tentent d'estimer l'état de plus grand nombre possible de points d'intérêt dans une image.
Le système proposé est divisé en deux étapes :

- Les modules géométrique et de segmentation effectuent une première estimation de l'état des points dans l'ensemble de l'image pour les points inclus dans les zones segmentées et pour les points mis en correspondance avec l'image précédente.
- Le module d'interaction des objets utilisent les points considérés dynamiques préalablement par les modules précédents pour estimer l'état des points au voisinage des objets segmentés (humains).

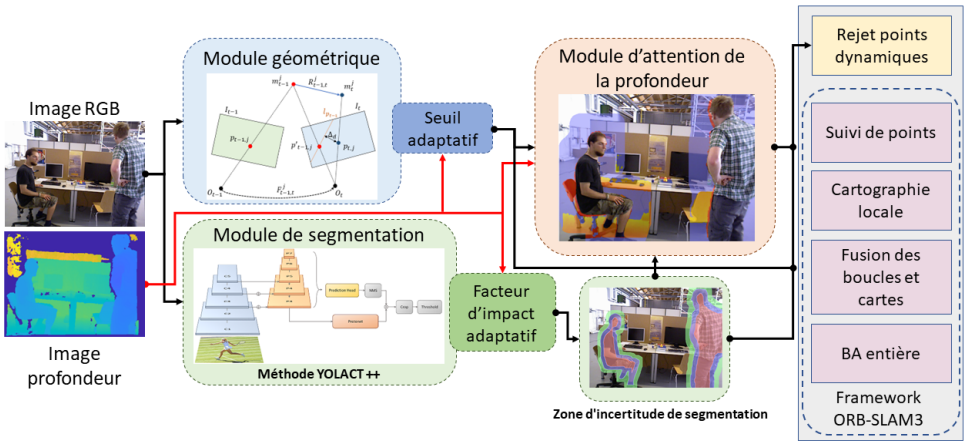


Figure 5.11 – Architecture de la méthode de SLAM dynamique proposée.

Le module géométrique effectue le calcul de l'erreur de reprojection selon la matrice fondamentale entre deux points correspondants d'une image à l'autre en utilisant le raisonnement par contrainte géométrique épipolaire. De manière générale, cette erreur de reprojection renseigne sur l'état du point de l'image actuelle. En effet, une erreur relativement élevée implique que le point a été en mouvement entre les deux images étudiées et est donc dynamique. L'ensemble des méthodes de vSLAM dynamique compare cette erreur à un seuil constant pour définir l'état d'un point.

Cependant, cette approche manque de flexibilité puisqu'elle n'effectue aucune distinction entre les points étudiés. En se basant sur les hypothèses exprimées 1 et 2, nous affinons l'estimation de l'état des points en incorporant un seuil adaptatif qui prend en compte la profondeur à laquelle se trouve le point considéré. Un point avec une erreur supérieure à ce seuil adaptatif est directement considéré comme dynamique et rejeté. Dans le cas contraire, cette erreur est utilisée dans une fonction de probabilité qui déterminera la probabilité d'état du point.

Le module de segmentation sémantique utilise le modèle de segmentation YOLACT++ [Bol+19] pour obtenir en temps réel le masque des objets définis comme dynamique présents dans la scène. D'après l'hypothèse communément admise dans la communauté, seul les humains sont considérés de fait comme dynamique et donc segmentés. Malheureusement, la précision de la segmentation n'est pas forcément élevée en fonction des images. Ainsi, nous affinons l'estimation de l'état des points des masques en étudiant leurs distances par rapport aux bords des masques à l'intérieur et à l'extérieur de ceux-ci. Ainsi plus un point contenu dans un masque est loin du bord, plus la probabilité qu'il soit dynamique est forte.

À l'inverse, plus un point à l'extérieur d'un masque est loin du bord, plus la probabilité qu'il soit dynamique est faible. Comme pour le module géométrique, cette distance nous renseigne sur l'état des points. Au-delà d'une certaine distance, nous considérons que le point est dans une zone de certitude d'état (statique à l'extérieur du masque et dynamique à l'intérieur). Dans le cas où le point est à une distance inférieure qu'un certain seuil donné, il est dans la zone d'incertitude de la segmentation. Nous modulons cette zone d'incertitude de segmentation correspondant à la distance aux bords des masques par un facteur d'impact de la profondeur qui prend en compte la profondeur à laquelle le point se trouve.

Ce raffinement de l'estimation est rendu possible à travers une fonction probabiliste prenant en compte cette distance adaptative aux bords des masques afin de considérer les points ayant une probabilité d'état dynamique compris entre 25% et 75%. Les points dont l'état est incertain sont alors affinés à travers une fonction de probabilité de mouvement des points qui intègre les probabilités d'état provenant des modules géométrique et sémantique. Enfin, le module d'interaction des objets de la seconde étape estime l'état des points incertains restants, c'est-à-dire ceux qui sont toujours considérés comme incertain et ceux qui n'ont été considéré par aucun des deux modules précédents.

Ce module estime l'état des points qui sont dans la zone d'influence de l'humain, soit d'un point de vue image, à une position et profondeur proches de celle des masques de l'humain. Nous utilisons alors l'état des points déjà estimé au voisinage des points de cette zone d'influence pour déterminer leur état en corrélant leur proximité en position et profondeur. Un point étudié entouré de nombreux points considérés dynamiques et ayant une profondeur assez proche peut donc être considéré à son tour comme dynamique car étant très probablement en interaction avec l'humain à proximité.

5.5.2 . Module géométrique

Le module géométrique est basé sur les contraintes géométriques épipolaires qui relient des points mis en correspondance entre deux images comme illustré dans la figure 5.12. Les contraintes géométriques épipolaires sont des hypothèses selon lesquelles deux points correspondants doivent satisfaire une contrainte qui régit une équation de reprojection d'un point par rapport à l'autre selon la matrice fondamentale entre ces deux images. Ainsi, un point d'intérêt noté q_i^{t-1} d'une image précédente $t-1$ et son correspondant p_i^t de l'image actuelle t peut être exprimé sous la forme d'une contrainte qui incorpore la matrice fondamentale F décrivant le mouvement de la caméra entre

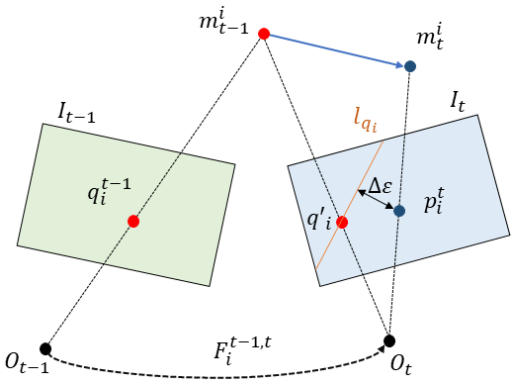


Figure 5.12 – Illustration de l'erreur de reprojection.

ces deux images comme ci-après :

$$(p_i)^T \cdot F \cdot (q_i) = 0 \tag{5.25}$$

Nous avons $F \cdot (q_i)$ qui correspond à la ligne épipolaire l_{q_i} étant la reprojection du point q_i dans l'image t. Cette ligne épipolaire peut être exprimée comme ci-après :

$$l_{q_i} = F \cdot (q_i) = \begin{bmatrix} X_{q_i} \\ Y_{q_i} \\ Z_{q_i} \end{bmatrix} \tag{5.26}$$

Comme expliqué précédemment, l'erreur de reprojection noté $\Delta\,\varepsilon(p_i)$ décrit la distance entre la ligne épipolaire l_{q_i} du point q_i et son correspondant p_i de l'image actuelle. Cette erreur est exprimée par l'équation suivante :

$$\Delta\varepsilon(p_i) = \frac{|(p_i)^T F(q_i)|}{\sqrt{\|X_{qi}\|^2 + \|Y_{qi}\|^2}} \tag{5.27}$$

Le point p_i est directement considéré dynamique si son erreur de reprojection $\Delta\varepsilon(p_i)$ est supérieur à un seuil adaptatif $\alpha(z_{p_i})$ lié à la profondeur du point p_i notée Z_{p_i}. Ce seuil adaptatif est basé sur l'hypothèse, que plus un point est loin, plus son erreur de reprojection représente un mouvement important en raison des contraintes de la géométrie projective. Ainsi, ce seuil adaptatif $\alpha(z_{p_i})$ est inversement proportionnel à la profondeur comme décrit par $z_{p_i} \in [D_{min}; D_{max}]$:

$$\alpha(z_{p_i}) = \frac{\alpha_{min} - \alpha_{max}}{D_{max} - D_{min}} \cdot (Z_{p_i} - D_{min}) + \alpha_{max} \tag{5.28}$$

Où α_{max} et α_{min} correspondent respectivement à 90% et 10% de la profondeur maximale D_{max} dans l'image de profondeur, D_{min} correspond à la profondeur minimale, et α_{min} et $\alpha(z_{p_i}) \in [0.5; 0.9]$. Dans le cas où l'erreur est

inférieure à ce seuil, cette erreur est intégrée à la fonction de densité de probabilité normale suivante pour estimer la probabilité d'état du point p_i :

$$P(g_{p_i}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\Delta\varepsilon(p_i))^2}{2\sigma^2}\right) \tag{5.29}$$

σ prend la valeur 1 et représente la déviation standard de cette distribution. La probabilité de mouvement $P(g_{p_i})$ est ensuite normalisée par le seuil adaptatif.
Notons M l'ensemble contenant les points appariés p_i de l'image actuelle. L'algorithme 2 permet d'estimer l'état des points inférieur au seuil adaptatif $\alpha(z_{p_i})$, ayant donc un état indéterminé, en calculant leur probabilité d'état par raisonnement géométrique comme décrit ci-après :

Algorithm 2 Algorithme du module géométrique

Entrée : Points q_i et p_i des images précédente et actuelle ; M : ensemble des points appariés
Sortie : Etat des points (Dynamique) ou probabilité de mouvement $P(g_{p_i})$

1: Trouver les points appariés
2: Calculer la matrice Fondamentale F
3: Utiliser le seuil adaptatif $\alpha(z_{pi}) \in [0.5; 0.9]$
4: **for** chaque $p_i \in M$ **do**
5: $\Delta\ \varepsilon(p_i) = \frac{|p_i F(q_i)^T|}{\sqrt{\|X_{q_i}\|^2 + \|Y_{q_i}\|^2}}$
6: **if** $\Delta\varepsilon(p_i) > \alpha(z_{p_i})$ **then**
7: $Dynamique \leftarrow p_i$
8: **else**
9: $P(g_{p_i}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\Delta\varepsilon(p_i))^2}{2\sigma^2}\right)$
10: **end if**
11: **end for**

5.5.3 . Module de segmentation sémantique

Le module de segmentation sémantique effectue tout d'abord une segmentation de l'image par classe d'objet afin d'identifier les zones correspondant aux humains (voir figure 5.13). Dans ces zones et à proximité, l'état des points d'intérêt est estimé à travers l'étude de leur probabilité d'être statique ou dynamique. De fait, d'après l'hypothèse établissant que l'humain est considéré dynamique, les points contenus dans les masques ont une forte probabilité d'être dynamique. Cependant, la précision de la segmentation n'est

pas toujours élevée ce qui conduit à devoir considérer la fiabilité de la segmentation au niveau des bords du masque (intérieur/extérieur) comme illustré dans la figure 5.13. Ainsi, la probabilité d'état des points à travers le mo-

Figure 5.13 – Illustration des imprécisions de segmentation.

dule de segmentation sémantique est obtenue en calculant leur probabilité de mouvement en prenant en compte leur distance par rapport aux bords des masques. La distance minimale Δd_m entre le point p_i et un point m_i^n sur

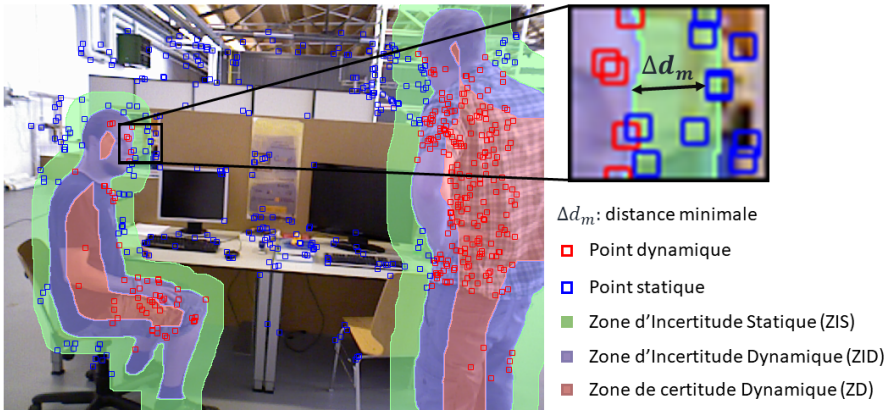


Figure 5.14 – Zone d'incertitude de segmentation.

le bord du masque m_n illustrée dans les figures 5.14 et 5.15 est définie comme ci-après :

$$\Delta d_m = \min \|p_i - m_i^n\| \tag{5.30}$$

Un modèle de régression logistique binomiale utilise cette distance pour estimer la probabilité d'état des points par rapport au masque m_n comme suit :

$$P(S_{p_i^n}) = \frac{1}{exp(-\beta(z_{m_i}) \cdot \Delta d_m) + 1} \tag{5.31}$$

Où $\beta(z_{m_i})$ est le facteur d'impact adaptatif proportionnel à la profondeur moyenne z_{m_i} du masque m auquel se trouve le point clé p_i liés et exprimés comme suit :

$$\beta(z_{m_i}) = \frac{\beta_{max} - \beta_{min}}{D_{max} - D_{min}} \cdot (z_{m_i} - D_{min}) + \beta_{min} \tag{5.32}$$

Le facteur d'impact adaptatif $\beta(z_{m_i}) \in [0.06; 0.25]$ assigne la taille de la zone d'incertitude de segmentation à l'intérieur et l'extérieur du masque (voir figure 5.14).

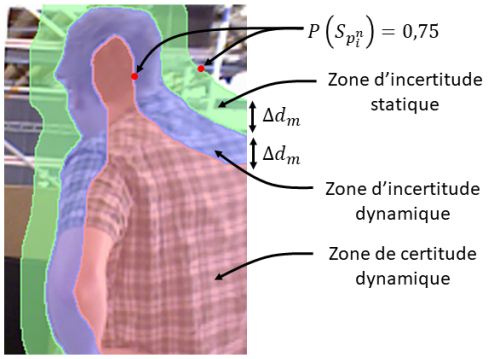


Figure 5.15 – Zone d'incertitude de segmentation liée à la distance Δd_m

Ainsi, plus la profondeur est élevée, plus la zone d'incertitude est petite, et inversement. Nous exprimons la zone d'incertitude du masque m comme étant la distance Δd_m du point p_i au bord du masque m pour que la probabilité d'état soit inférieure à 75% tel que :

$$\Delta d_m < \frac{\ln \frac{1}{3}}{\beta(z_{m_i})} \tag{5.33}$$

Les points avec une probabilité supérieure à 75% (donc en dehors de la zone d'incertitude) prennent alors l'état de la zone d'étude, à savoir statique pour les points extérieurs au masque et dynamique pour ceux à l'intérieur (voir figure 5.15). Ainsi, plus l'humain segmenté est à une distance lointaine, plus la zone d'incertitude statique et dynamique sera petite pour tenir compte de la précision de la segmentation liée à la taille des objets.

L'équation 5.31 permet alors d'exprimer la probabilité d'état des points par rapport à leurs distances aux bords du masque le plus proche. La probabilité d'état obtenue nous renseigne sur l'état du point en fonction de sa position par rapport au masque (intérieur/extérieur) et à la zone d'incertitude (en dehors/dedans). La probabilité pour un point dans le masque indique sa probabilité d'être dynamique tandis que celle d'un point à l'extérieur indique sa probabilité d'être statique. Ainsi, la probabilité d'état des points extérieurs

$P_{ext}(S_{p_i^n})$ nécessite d'être changé en une probabilité d'être dynamique $P(S_{p_i^n})$ en effectuant une simple inversion :

$$P(S_{p_i^n}) = 1 - P_{ext}(S_{p_i^n}) \tag{5.34}$$

Avec $P(S_{p_i^n}) \in [0; 0.5]$. En résumé, l'intervalle de probabilité d'un point d'être dynamique peut être défini comme suit :

$$P(S_{p_i^n}) = \begin{cases} [0.0; 0.25] & \text{if } p_i \in \ Zone\ Statique\ (ZS) \\ ]0.25; 0.5[& \text{if } p_i \in \ Zone\ d'Incertitude\ Statique\ (ZIS) \\ [0.5; 0.75[& \text{if } p_i \in \ Zone\ d'Incertitude\ Dynamique\ (ZID) \\ [0.75; 1.0] & \text{if } p_i \in \ Zone\ Dynamique\ (ZD) \end{cases} \tag{5.35}$$

Le modèle de régression logistique binomiale nous informe sur la probabilité d'état du point en suivant l'algorithme 3 détaillé ci-après :

Algorithm 3 Algorithme du module de segmentation sémantique

Donnée : Points p_i et masques m_n de l'image courante
Résultat : Probabilité sémantique de l'état du point $P(S_{p_i^n})$

Calculer la distance minimale Δd_m
Déterminer le facteur d'impact adaptatif $\beta(z_{p_i})$
Calculer la probabilité d'état $P(S_{p_i^n})$
for chaque p_i **do**
    if $p_i \in m_n$ **then**
        if $P(S_{p_i^n}) > 0.75$ **then**
            $Dynamique \leftarrow p_i$ et $p_i \in DA$
        else
            $p_i \in UDA$
        end if
    else
        if $P(S_{p_i^n}) < 0.25$ **then**
            $Statique \leftarrow p_i$ et $p_i \in SA$
        else
            $p_i \in USA$
        end if
    end if
end for

Le principe de fonctionnement des modèles de segmentation d'objet et la méthode choisie sont détaillés dans la section 6.4. Nous justifierons le choix de la méthode par rapport à l'existant et soulignerons ses limites. Une solution adaptée au contexte d'utilisation des méthodes de SLAM dynamique sera proposée pour surmonter ces limites.

5.5.4 . Mise à jour des probabilités

La probabilité de mouvement $P(p_i)$ représente la probabilité d'état du point p_i avec $P(p_i) \in Etat$ où $Etat = \{statique(s), dynamique(d)\}$. Elle intègre lorsque le point considéré a un correspondant, les probabilités $P(g_{p_i})$ issue du module géométrique et $P(S_{p_i})$ issue du module de segmentation sémantique comme suit :

$$P(p_i) = \omega P(g_{p_i}) + (1 - \omega)P(S_{p_i}) \tag{5.36}$$

Avec ω comme poids qui décrit la pertinence du modèle probabiliste pour diverses situations de point. Notons que pour les points n'ayant pas de correspondant, le poids ω prend la valeur 0 et donc la probabilité $P(p_i)$ correspond uniquement à la probabilité sémantique $P(S_{p_i})$.

Dans les zones de segmentation incertaine (ZIS et ZID), le poids ω est mis à 0.5 tandis que dans les zones de certitude dynamique (ZD) il prend la valeur 0.1. Dans le cas de points appariés qui se trouvent dans la zone de certitude statique (ZS), nous mettons ω à 0.9, s'il est en dehors de la zone de segmentation la probabilité $P(p_i)$ vaut directement la probabilité géométrique $P(g_{p_i})$. Nous mettons à jour la fonction de probabilité de mouvement en utilisant un filtre Bayésien de l'expression suivante :

$$bel(p_i) = \eta P(\Omega_i|p_i) \int P(p_i|q_i)bel(q_i)dq_i \tag{5.37}$$

Où $\eta = \frac{1}{bel(p_i=d)+bel(p_i=s)}$ est un facteur d'impact pour normaliser les probabilités obtenues. La probabilité à priori initiales p_0 et la probabilité d'observation $P(\Omega_i|p_i)$ sont mises à 0.5. Enfin, les points d'intérêt avec une probabilité de déplacement supérieure à 0,5 sont considérés comme des points dynamiques et donc rejetés, les autres sont conservés comme points ayant un état inconnu et seront pour certains à nouveau estimés à travers le module d'interaction des objets.

La probabilité d'état des points est obtenue en suivant l'algorithme 4 :

Algorithm 4 Algorithme de calcul de la probabilité d'état des points d'intérêt

Entrée : Probabilité géométrique $P(g_{p_i})$ et sémantique $P(S_{p_i^n})$ de l'état des points

Output : Probabilité d'état des points $P(p_i)$

for chaque p_i **do**
  if $p_i \in M$ **then**
    if $p_i \in m_n$ **then**
      if $P(S_{p_i^n}) > 0.75$ **then**
        $\omega \leftarrow 0.1$
      else
        $\omega \leftarrow 0.5$
      end if
    else
      if $P(S_{p_i^n}) < 0.25$ **then**
        $\omega \leftarrow 1$
      else
        $\omega \leftarrow 0.5$
      end if
    end if
  else
    if $P(S_{p_i^n}) > 0.75$ **then**
      $\omega \leftarrow 0.1$
    else
      $\omega \leftarrow 0.5$
    end if
  end if
  $P(p_i) = \omega P(g_{p_i}) + (1 - \omega) P(S_{p_i})$
end for

5.5.5 . Module d'interaction des objets

Le module d'interaction des objets tente de déterminer les interactions entre humains et objets inertes en corrélant leur proximité de position et de profondeur. Ce module est basé sur l'hypothèse que plus un point considéré statique ou d'état inconnu a une position image et une profondeur proche d'un humain, plus il est probable qu'il interagisse avec lui et donc soit dynamique. À cette fin, nous définissons une zone d'interaction liée à la position et à la profondeur des humains présents dans la scène (voir zones colorées de la figure 5.16).

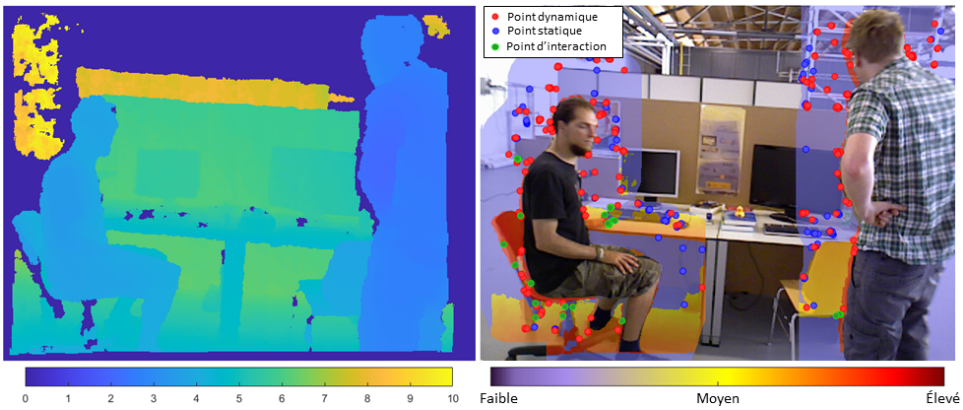


Figure 5.16 – Module d'interaction des objets avec corrélation de la profondeur et la position. (Image de gauche) Image de profondeur où o représente le manque de données. (Image de droite) Carte de probabilité d'interaction où les points d'interaction sont des points dynamiques résultant du module.

La probabilité d'état des points d'intérêt des objets inertes (tout point en dehors des masques considérés comme statique ou d'état inconnu) présents dans cette zone est à nouveau estimé en examinant leur voisinage. Cette zone corrèle des informations 2D des images à travers la distance en pixel entre le point étudié et l'humain le plus proche avec des informations 3D correspondant à la distance en profondeur entre ces deux éléments.

Ainsi, dans cette zone d'interaction, nous évaluons la proximité en termes de position et de profondeur entre le point étudié et les points dans son voisinage définis comme dynamiques par les modules précédents. Cette proximité combinée à la position du point étudié par rapport à son voisinage nous renseigne sur l'état du point.

Un point étudié p_i, considéré comme préalablement statique, dans cette zone aura une distance en termes de position et de profondeur $\Delta z(i)$ inférieure aux seuils adaptatifs définis. Nous considérerons tout point dynamique p_i^d comme étant dans le voisinage du point étudié p_i, s'il satisfait aux conditions suivantes :

$$\begin{cases} \|p_i - p_i^d\| < \delta(z_{p_i}), & \delta(z_{p_i}) \in [11; 48] \\ \Delta z(i) = |p_i(z) - p_i^d(z)| < \rho, & \rho = 0.9 \end{cases} \tag{5.38}$$

Où $\delta(z_{p_i}) = 48 - 4 * z_{p_i}$ est un seuil adaptatif correspondant à une distance de position en pixels, et ρ est un seuil de distance de profondeur en mètres. Comme le montre la figure 5.17, seuls les points clés dynamiques satisfaisant les conditions de l'équation 5.38 sont pris en compte pour estimer l'état antérieur des points clés statiques.

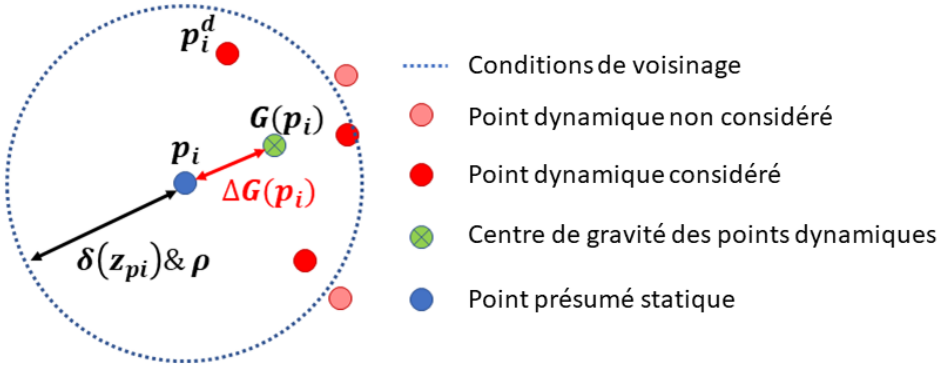


Figure 5.17 – Voisinage des points clés statiques satisfaisant les conditions.

Notons que nous effectuons un sous-échantillonnage suivi d'un ré-échantillonnage de l'image de profondeur pour compléter les informations de profondeur manquantes (voir la figure 5.18). Le sous-échantillonnage a été effectué à travers une fenêtre glissante de 10x10 à partir d'une image de profondeur originale de taille 640x480 et une image sous-échantillonnée de taille 64x48.

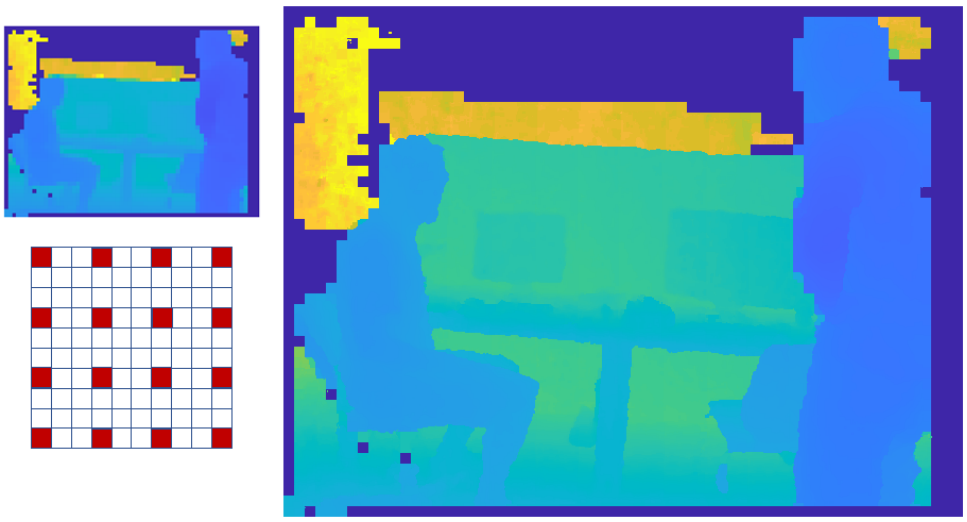


Figure 5.18 – Image de profondeur re-échantillonnée (à droite). Image de profondeur sous-échantillonnée (en haut à gauche). Filtre médian (en bas à gauche)

Nous appliquons un filtre médian qui calcule la profondeur moyenne sans tenir compte des pixels vides pour chaque fenêtre d'observation (pixels rouges illustrés sur la figure 5.18) .

Nous calculons le centre de gravité $G(p_i)$ de ces points dynamiques présents dans le voisinage en les pondérant par leur distance en profondeur comme suit :

$$G(p_i) = \frac{\sum_{i=1}^{k} p_i^d \cdot \frac{\rho - \Delta z(i)}{\rho}}{\sum_{i=1}^{k} \frac{\rho - \Delta z(i)}{\rho}} \tag{5.39}$$

Le centre de gravité G_{p_i} des points clés considérés dynamiques est ensuite utilisé pour calculer la distance $\Delta(G_{pi})$ de ce centre par rapport au point statique précédent p_i comme suit :

$$\Delta(G_{(p_i)}) = |G_{(p_i)} - p_i| \tag{5.40}$$

Cette distance $\Delta(G_{pi})$ fournit des informations sur l'état de p_i qui sont utilisées pour affiner l'état du point clé statique précédent.

La distance $\Delta(G_{p_i})$ entre le centre de gravité calculé $G(p_i)$ et le point étudié p_i nous renseigne sur l'état du point. En effet, cette distance nous renseigne sur la pertinence du lien caractérisant ces points dynamiques et le point p_i. Une faible distance implique pour le point statique p_i soit une très grande proximité avec ces points dynamiques soit que ces points l'englobent. Ainsi, le module d'interaction change l'état des points en l'état dynamique quand $\Delta(G_{p_i})$ est plus petit qu'un seuil adaptatif $\gamma(z_{p_i})$ inversement proportionnel à la profondeur où $\gamma(z_{p_i}) \in [10; 28]$:

$$\alpha(z_{p_i}) = \frac{\gamma_{min} - \gamma_{max}}{D_{max} - D_{min}} \cdot (z_{p_i} - D_{min}) + \gamma_{max} \tag{5.41}$$

Si le voisinage du point étudié p_i n'a pas de point préalablement dynamique p_i^d, alors nous étudions la proximité du point étudié à travers sa probabilité de segmentation sémantique $P(S_{p_i})$ et sa proximité de profondeur $\Delta\rho$. Nous exprimons alors sa probabilité de déplacement $P_{inter}(p_i)$ de la manière suivante :

$$P_{inter}(p_i) = \omega_d \cdot (1.5 - P(S_{p_i}) + (1 - \omega_d) \cdot \Delta\rho \tag{5.42}$$

Avec $\Delta\rho = \frac{(\rho - |p_i(z) - D_h|)}{\rho}$ et D_h qui correspond à la profondeur moyenne de l'humain. ω_d est un poids décrivant la fiabilité de l'information.

L'algorithme 5 décrit le processus du module d'interaction des objets.

Algorithm 5 Algorithme module d'interaction des objets

Entrée : Point statique étudié : p_i ;
Point dynamique préalable : p_i^d ; Profondeur du point : $p_i(z)$; Zone d'interaction : L ; Profondeur moyenne de l'humain : D_h
Sortie : Probabilité d'état du point étudié $P(p_i)$

for chaque p_i **do**
  if $p_i \in L$ **then**
    for chaque p_i^d **do**
      $\Delta p(i) = \|p_i - p_i^d\|$
      $\Delta z(i) = |p_i(z) - p_i^d(z)|$
      if $\Delta p(i) < \delta(z_{pi}) \quad \& \quad \Delta z(i) < \rho$ **then**
        $G(p_i) = \frac{\sum_{i=1}^{k} p_i^d \cdot \frac{\rho - \Delta z(i)}{\rho}}{\sum_{i=1}^{k} \frac{\rho - \Delta z(i)}{\rho}}$
      end if
    end for
    $\Delta(G_{p_i}) \quad = \quad |G_{p_i} - p_i|$
    if $\Delta(G_{p_i}) < \gamma(z_{pi})$ **then**
      $Dynamique \leftarrow p_i$
    else
      $P_{inter}(p_i) = \omega_d \cdot (1.5 - P(S_{p_i}) + (1 - \omega_d) \cdot \Delta\rho$
      if $P_{inter}(p_i) > 0.5$ **then**
        $Dynamique \leftarrow p_i$
      end if
    end if
  else
    $Statique \leftarrow p_i$
  end if
end for

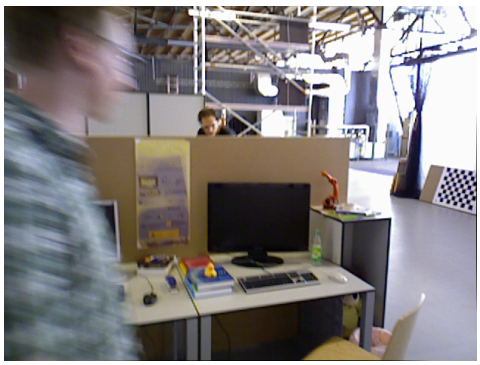
5.5.6 . Module de segmentation sémantique : limitations et propositions

La qualité et la précision jouent un rôle prépondérant dans l'efficacité de notre algorithme de SLAM dynamique. Le module de segmentation sémantique repose entièrement sur la bonne segmentation des humains présents dans la scène observée. Ainsi, un humain mal et/ou pas segmenté impliquera un dysfonctionnement du module qui ne pourra déterminer les zones de certitude de dynamisme lié à la présence d'humain. Ce dysfonctionnement peut provenir de trois causes :

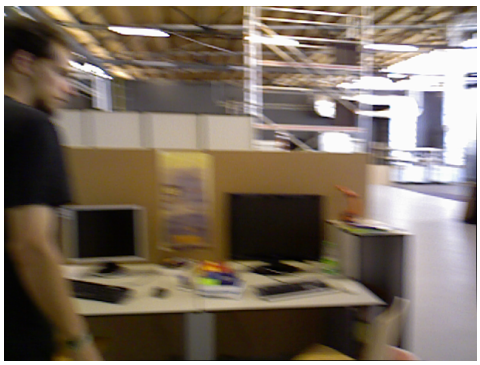
- La précision du modèle de segmentation : directement liée à l'architecture du modèle, elle ne peut être améliorée qu'en modifiant le modèle.

- Le contexte de scène : un objet partiellement occulté ou hors du champ de l'image pourra causer le dysfonctionnement de la segmentation qui sera incapable de reconnaître l'humain.
- La robustesse du modèle de segmentation : la présence de flou de mouvement lié au mouvement de la caméra ou de l'humain peut causer une baisse des performances du modèle de segmentation. Le phénomène de flou de défocalisation peut également provoquer une diminution des performances du modèle. Dans ces deux cas, on parle de modèle non-robuste.

Les deux dernières causes, à savoir le contexte de scène et la robustesse, peuvent être supprimées en jouant sur le contenu de la base de données utilisée pour entraîner le modèle. Ce procédé, utilisé en apprentissage profond, s'appelle l'augmentation de données. Il consiste à modifier directement le contenu de la base de données pour y incorporer des images (dans le cas de modèles de vision) ayant des caractéristiques photométriques adaptées au besoin de robustesse du modèle pour le type d'application souhaitée. La

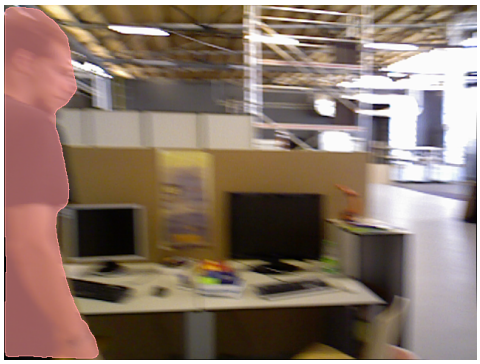


(a) Flou de mouvement de l'objet



(b) Flou de mouvement de la caméra

(c) Inefficacité de la segmentation



(d) Imprécision de la segmentation

Figure 5.19 – Illustration de la robustesse du modèle de segmentation

figure 5.19 illustre le manque de robustesse du modèle de segmentation qui segmente mal ou pas du tout certains humains présents dans la scène. Ce

constat nous a amené à étudier l'impact des distorsions sur les modèles de détection et segmentation (voir section 6.3.8), proposé des algorithmes de génération de distorsions complexes et réalistes (voir section 6.3), et enfin évaluer l'apport d'une augmentation de données sur le modèle de segmentation d'objet choisie, à l'aide de notre base de données perturbée générée (voir section 6.3.9).

5.6 . Expérimentations et résultats

Notre méthode DAM-SLAM a été évaluée sur les séquences fre3 du jeu de données public TUM RGB-D [Stu+12] dédié à l'évaluation des méthodes de vSLAM dans des environnements dynamiques. Ces séquences dédiées fournissent des images RVB et de profondeur, ainsi que des trajectoires de vérité terrain. Elles se décomposent en deux types de séquences, à savoir assis(s) et marche(w), qui correspondent respectivement à des scénarios à bas et haut niveaux de dynamique humaine. Les scénarios assis (s) correspondent à des séquences avec des humains restant assis tandis que les scénarios de marche (w) correspondent à des séquences avec des humains en mouvement.

Table 5.3 – Caractéristiques des scénarios de séquence fre3 dans le jeu de données TUM RGB-D

Séquences	État humains	Trajectoires de la caméra
fr3/w/xyz	Mouvement	Translation
fr3/w/half	Mouvement	Translation/rotation fortes
fr3/w/static	Mouvement	Mouvement faible
fr3/w/rpy	Mouvement	Rotations Statiques
fr3/s/xyz	Faible Mouvement	Translation
fr3/s/half	Faible Mouvement	Translation/rotation fortes
fr3/s/static	Faible Mouvement	Mouvement faible
fr3/s/rpy	Faible Mouvement	Rotations Statiques

Les séquences xyz consistent en des mouvements de translation de la caméra le long de trois axes (axe xyz) sans rotations. Les séquences demi-sphère (half) sont des scénarios composés de petits mouvements de demi-sphère d'environ un mètre de diamètre avec des translations et des rotations. Les séquences rpy incluent des mouvements de rotation le long des axes principaux (roulis-tangage-lacet) à la même position. Les séquences statiques représentent l'acquisition d'images avec un état relativement stationnaire (sans rotations ni translations). Les caractéristiques des scénarios sont résumées dans le tableau 5.3.

Figure 5.20 – Évaluation du module d'interaction objets (RMSE) sur le jeu de données RGB-D TUM [Stu+12]

Séquences	Notre méthode (w/o-DAM)			Notre méthode (w/-DAM)			Amélioration		
	ATE	T. RPE	R. RPE	ATE	T. RPE	R. RPE	ATE	T. RPE	R. RPE
fr3/w/xyz	0.0162	0.0236	0.648	**0.0149**	**0.0217**	**0.6194**	7.41%	7.63%	4.24%
fr3/w/half	0.0291	0.0413	0.903	**0.0262**	**0.0369**	**0.8608**	10.31%	10.65%	4.67%
fr3/w/static	0.0160	0.0282	0.565	**0.0068**	**0.0097**	**0.2789**	57.5%	65.60%	50.64%
fr3/w/rpy	0.0326	0.0466	1.047	**0.0316**	**0.0453**	**0.9364**	3.10%	2.79%	10.56%
fr3/s/xyz	**0.0141**	**0.0202**	0.633	0.0150	0.0217	**0.6264**	-6.38%	-7.43%	1.04%
fr3/s/half	0.0204	0.0293	0.842	**0.0198**	**0.0283**	**0.8156**	2.94%	3.41%	3.14%
fr3/s/static	0.0069	0.0101	0.3320	**0.0064**	**0.0093**	**0.3222**	7.25%	7.92%	2.95%

Toutes les expériences ont été réalisées sur un ordinateur Intel XENON avec 16 Go de RAM et un GPU Nvidia RTX2080 SUPER. Pour cela, nous avons utilisé le framework officiel de la méthode ORB-SLAM 3 [Cam+20] sur une version 20.04 d'Ubuntu. Nous avons utilisé le modèle officiel YOLACT++ [Bol+19] avec le backbone Resnet50-FPN pré-entraîné sur le jeu de données COCO pour le module de segmentation. Ce modèle prend des images d'entrée de taille 550×550, ce qui est assez similaire à la taille des images des séquences fr3.

Notre méthode a été comparée à d'autres méthodes vSLAM dynamiques de pointe selon les mesures d'erreur de trajectoire absolue (ATE) et d'erreur de pose relative (RPE). Rappelons que, l'ATE est l'écart moyen par rapport de la trajectoire estimée par rapport à la trajectoire de vérité terrain par image. Le RPE est une mesure de l'écart moyen entre les transformations relatives entre les poses estimées et les poses provenant de la vérité de terrain. L'ATE et le RPE de translation (T.RPE) sont exprimés en mètres (m), tandis que le RPE de rotation (R.RPE) est exprimé en degré (°).

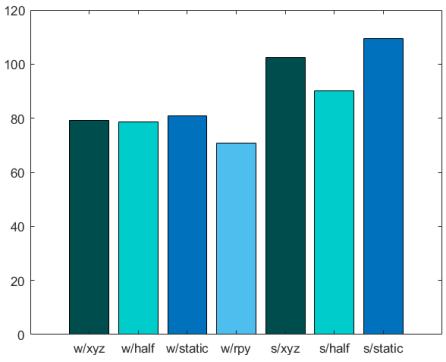
5.6.1 . Évaluation de notre méthode

Dans un premier temps, l'apport du Depth Attention Module (DAM) à notre système a été mis en évidence en évaluant les performances de notre méthode avec (w/-DAM) et sans (w/o-DAM) ce module (voir colonne améliorations du tableau de la figure 5.20).
Une amélioration nette a été observée pour les scénarios de scène à haute dynamique, en particulier pour la séquence *walking static*, qui inclut une scène avec des humains dynamiques, mais un mouvement de caméra peu important. Cette amélioration est due à la composition de la séquence w/static, qui inclut de nombreuses interactions entre les humains et les objets (voir la colonne ratio dans le tableau de la figure 5.21).
Le DAM permet de considérer dans ces séquences des objets en contact avec

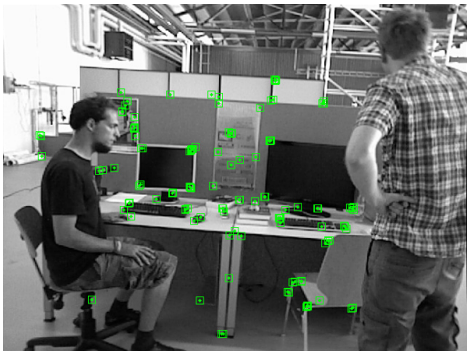
Figure 5.21 – Interaction visible des humains avec les objets par séquence

Séquences	Nombre d'objets							Nombre d'image	Ratio
	Chaise	Souris	Clavier	Livre	Boîte	Webcam	Total		
fr3/w/xyz	477	31	133	0	0	0	641	859	0.746
fr3/w/half	313	0	69	470	0	11	863	1067	0.809
fr3/w/static	655	0	71	0	60	0	786	743	1.058
fr3/w/rpy	225	0	102	0	0	0	327	910	0.359

l'individu, tels que chaises, livres, webcam, souris et claviers, sans aucune connaissance sémantique préalable en dehors de l'homme. Nous avons évalué la contribution du DAM dans le rejet dynamique des points d'intérêt par l'interaction des objets. Notre module peut traiter et rejeter environ 80 points



(a) Nombre moyen de points d'intérêt dynamiques rejetés par image via le DAM.



(b) points d'intérêt statiques conservés avec le DAM.

Figure 5.22 – Impact du module DAM pour la détection des points dynamiques.

d'intérêt dynamiques supplémentaires par image pour des scénarios dynamiques dans le cas d'objets segmentés (voir fig. 5.22a). Le tableau 5.21 synthétise le nombre d'objets en interaction avec les humains dans les différentes séquences. Nous observons une corrélation entre le ratio du nombre d'objets en interaction et le nombre de points d'intérêt rejetés via le DAM, ce qui met en évidence l'efficacité de notre module DAM dans l'estimation des interactions.
Les figures 5.23 et 5.22b illustrent la contribution du DAM pour la détection de points d'intérêt dynamiques grâce à l'interaction d'objets.
Comme le montre le tableau 5.4, les résultats obtenus montrent que les performances de notre méthode sont meilleures que celles des techniques considérées.

Table 5.4 – Évaluation de l'erreur absolue de trajectoire en mètres (m). Les meilleurs résultats sont en gras

Séquences	RMSE				
	ORB-SLAM3	DynaSLAM	DS-SLAM	RDMO	DAM-SLAM(ours)
fr3/w/xyz	0.725	0.0164	0.0247	0.0226	**0.0149**
fr3/w/half	0.399	0.0296	0.0303	0.0304	**0.0262**
fr3/w/static	0.358	**0.0068**	0.0081	0.0126	**0.0068**
fr3/w/rpy	0.807	0.0354	0.4442	0.1283	**0.0316**
fr3/s/static	0.0096	0.0108	0.0065	0.0066	**0.0064**

Cependant, les résultats de l'état de l'art doivent être mis en perspective en les comparant à travers une métrique cohérente. En effet, prendre des résultats bruts d'ATE et de RPE sans tenir compte des performances initiales des frameworks ORB-SLAM sur ce jeu de données n'est pas rigoureux puisque les performances dépendent des librairies utilisées (OpenCV, Eigen) et de la puissance de l'ordinateur. Les résultats sont globalement cohérents pour un utilisateur, mais varient d'une configuration à l'autre. Ainsi, nous avons choisi de comparer notre taux d'amélioration avec notre implémentation du framework ORB-SLAM3. Nous avons fait de même avec les méthodes DynaSLAM [Bes+18], DS-SLAM [Yu+18] et RDMO-SLAM en prenant les résultats du benchmarking de l'article RDMO-SLAM [LM21a]. DP-SLAM [Li+21] fournit directement les taux d'amélioration que nous avons pris tels quels.

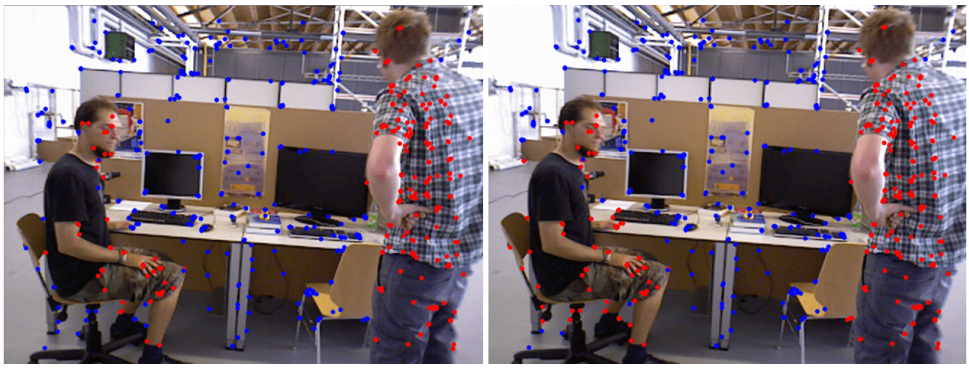


Figure 5.23 – Résultats de la détection des points d'intérêt dynamiques sans (gauche) et avec (droite) le DAM (points d'intérêt statiques et dynamiques, respectivement, en bleu et rouge)

5.6.2 . Benchmarking

Nous avons utilisé les résultats ATE et RPE des articles RDMO-SLAM et DP-SLAM pour notre analyse comparative. Il est important de noter que certaines méthodes ne sont pas en temps réel, ce qui explique dans certains cas les différences de performances. Le fonctionnement en temps réel et l'exigence supplémentaire de connaissances sémantiques préalables des méthodes étudiées sont résumés dans le tableau 5.5.

Table 5.5 – Résumé des caractéristiques de la méthode

Méthodes	Temps réel	Connaissance sémantique supplémentaire
DynaSLAM	×	✓
DS-SLAM	×	✓
DP-SLAM	×	✓ / ×
RDMO-SLAM	✓	✓
Ours	×	×

Dans cette étude, nous distinguons les méthodes utilisant des connaissances sémantiques antérieures autres que celles de l'homme en raison de leur manque de généralisation. Nous avons calculé le taux d'amélioration Ψ des méthodes à l'aide de l'équation suivante :

$$\Psi = 1 - \frac{S_m}{S_o} \tag{5.43}$$

Où S_m et S_o sont les scores des méthodes évaluées et du cadre ORB-SLAM3 pour chaque séquence. Les résultats de l'amélioration qualitative sont présentés dans les tableaux des figures 5.24,5.25,5.26, dans lesquels les meilleurs résultats sont mis en évidence en gras.

Comme indiqué dans le tableau de la figure 5.24, notre méthode sans aucune connaissance sémantique préalable supplémentaire est meilleure ou légèrement inférieure aux autres méthodes pour la métrique ATE. En effet, nous obtenons les deuxièmes meilleurs résultats pour les séquences w/xyz et w/rpy.

Les tableaux des figures 5.25 et 5.26 mettent en évidence la meilleure efficacité de notre méthode par rapport à d'autres ayant moins de connaissances pour estimer la pose de la caméra en fonction de sa translation et de sa rotation. DAM-SLAM augmente considérablement les performances du cadre ORB-SLAM3 pour les scénarios à haute dynamique puisque la localisation est encore plus perturbée par les scènes fortement dynamiques. En effet, nous enregistrons des gains relativement importants en ATE et RPE, qui

Figure 5.24 – Évaluation du taux d'amélioration de l'ATE en mètres (m). Les meilleurs résultats sont en gras et (*) désigne les méthodes nécessitant des connaissances sémantiques préalables supplémentaires.

Séquences	RMSE					
	ORB-SLAM3	DynaSLAM*	DS-SLAM*	DP-SLAM*	RDMO-SLAM*	DAM-SLAM
fr3/w/xyz	0.725	**98.2%**	97.3%	97.9%	97.5%	97.9%
fr3/w/half	0.463	**95.5%**	95.4%	94.1%	95.4%	94.3%
fr3/w/static	0.358	**98.1%**	97.8%	98.0%	96.5%	**98.1%**
fr3/w/rpy	0.807	**96.8**%	60.0%	94.9%	88.4%	96.1%
fr3/s/static	0.0096	-2.0%	27.7%	29.8%	26.7%	**33.3%**

Figure 5.25 – Évaluation du taux d'amélioration du RPE de translation en mètres (m).

Séquences	RMSE					
	ORB-SLAM3	DynaSLAM*	DS-SLAM*	DP-SLAM*	RDMO-SLAM*	DAM-SLAM
fr3/w/xyz	1.0817	94.9%	92.2%	95.8%	93.0%	**98.0%**
fr3/w/half	0.6103	91.3%	90.9%	61.8%	91.0%	**94.0%**
fr3/w/static	0.5083	**98.9%**	98.7%	51.8%	97.9%	98.1%
fr3/w/rpy	1.129	89.7%	65.6%	54.4%	68.0%	**96.0%**
fr3/s/static	0.0155	-23.5%	23.5%	3.8%	11.8%	**40.0%**

Figure 5.26 – Évaluation du taux d'amélioration du RPE de rotation en degrés (°).

Séquences	RMSE					
	ORB-SLAM3	DynaSLAM*	DS-SLAM*	DP-SLAM*	RDMO-SLAM*	DAM-SLAM
fr3/w/xyz	20,193	92.0%	89.5%	40.8%	89.9%	**96.9%**
fr3/w/half	14,53	89.2%	88.7%	55.8%	89.1%	**94.1%**
fr3/w/static	8,7533	95.7%	95.5%	37.1%	94.4%	**96.8%**
fr3/w/rpy	0,5757	88.7%	65.7%	34.7%	70.9%	**94.4%**
fr3/s/static	0,3783	-13.6%	9.0%	2.4%	3.2%	**17.1**%

sont meilleurs pour certaines séquences et légèrement inférieurs à l'état de l'art pour d'autres. Elle démontre l'intérêt de considérer les paramètres liés à la proximité avec l'homme dans l'estimation de la trajectoire et de la pose de la caméra. Le gain en précision dans l'estimation de la pose de la caméra est encore plus important par rapport aux autres méthodes dans le cas où la caméra a de grandes amplitudes de mouvement.

De plus, nous avons réalisé une comparaison des performances entre notre méthode et la méthode DGS-SLAM [Yan+22], qui considère également la profondeur dans son approche et est basée sur le framework ORB-SLAM3. Cette comparaison n'a été effectuée qu'à travers le taux d'amélioration ATE en raison du manque d'informations dans l'article DGS-SLAM sur les performances ORB-SLAM3 pour la métrique RPE.

Table 5.6 – Taux d'amélioration RMSE de l'ATE

Séquences	DGS-SLAM* [Yan+22]	DAM-SLAM
fr3/w/xyz	**98.1%**	97.9%
fr3/w/half	93.3%	**94.3%**
fr3/w/statique	69.4%	**98.1%**
fr3/w/rpy	95.8%	**96.1%**
fr3/s/statique	**35.2%**	33.3%

La comparaison illustrée dans le tableau 5.6 met en évidence les performances de notre méthode où les meilleurs résultats sont en gras, et (*) désigne les méthodes nécessitant des connaissances sémantiques préalables supplémentaires. Notre méthode a surpassé les performances de la méthode DGS-SLAM pour les scènes dynamiques avec moins d'informations requises. Nous avons effectué une analyse en temps réel de notre système vSLAM, que nous avons résumée dans le tableau 5.7.

Table 5.7 – Temps d'exécution de chaque module (en ms)

Méthode	YOLACT	Prob.	DAM	Suivi
ORB-SLAM3	×	×	×	32
DAM-SLAM	29	101	8	109

Nous n'obtenons pas de fonctionnement en temps réel comme la plupart des méthodes, mais nous avons réussi à réduire le temps de traitement de 43%

sans réduire les performances de nos méthodes. Malheureusement, cette optimisation entraîne la perte de suivi pour la séquence w/rpy. Nous avons préféré l'efficacité à la rapidité pour notre méthode, mais d'autres investigations seront menées dans le futur pour dépasser cette limite. Les trajectoires estimées de chacune des séquences sont illustrées pour notre méthode et la méthode ORB-SLAM3 dans la figure 5.27.

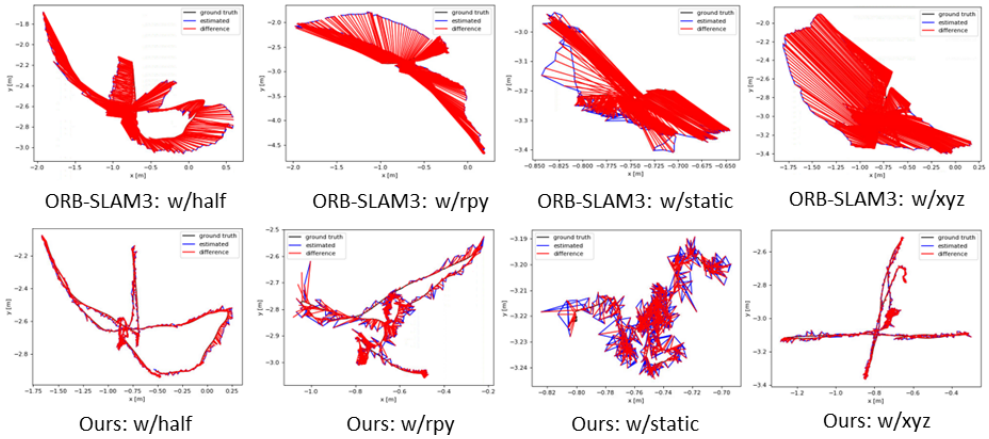


Figure 5.27 – Les trajectoires de caméra estimées à travers la métrique ATE de l'ORB-SLAM3 et notre méthode

La méthode proposée permet de s'affranchir des limitations des modules géométriques et sémantiques utilisés dans les méthodes existantes. À cette fin, nous considérons la profondeur à plusieurs niveaux pour améliorer la fiabilité des modules de segmentation géométrique et sémantique. Cela augmentera également le nombre de points clés pouvant être intégrés dans l'algorithme via le module Depth Attention. Il convient de noter que cette approche affine l'estimation de l'état des points clés en étendant la probabilité d'état liée au voisinage de chaque point clé. La méthode proposée surpasse l'état de l'art en termes de précision et avec un temps de calcul relativement faible. De plus, cette approche est plus généralisable puisqu'elle peut être utilisée dans un environnement avec des scènes contenant des objets non reconnus, c'est-à-dire sans aucune information sémantique préalable.

Cependant, cette méthode est relativement longue en raison de la complexité du processus de mise en correspondance des points clés. La première solution permet de simplifier la mise en correspondance des points clés, mais avec une légère baisse des performances. Ainsi, des recherches supplémentaires devraient être menées sur la stratégie d'appariement pour réduire la complexité de calcul. Une solution consisterait à limiter la zone de recherche des points clés correspondants autour de leur voisinage dans la trame précé-

dente. Une autre stratégie consisterait à utiliser un schéma multi-résolution afin de réduire la taille des trames. Ces pistes potentielles pourraient être explorées dans un futur travail.

5.7 . Conclusion

En nous basant sur l'existant, nous avons proposé une approche qui surpasse les performances de l'état de l'art pour la localisation et la cartographie d'environnements dynamiques. En effet, elle améliore fortement l'estimation de la position relative d'un point de vue translationnel et rotationnel par rapport à la méthode ORB-SLAM originale du framework. Notre méthode se base sur un module de segmentation d'objet localisant au niveau pixel les parties de l'image incluant des objets considérés comme dynamiques afin de caractériser l'état de leurs points d'intérêt. Ces informations extraites sont combinées à des informations provenant d'un module basé sur un raisonnement géométrique permettant d'estimer l'état de paires de points entre deux images successives. La prise en compte de l'impact de la profondeur sur l'incertitude de l'état des points permet d'améliorer la précision de l'estimation en considérant les informations visuelles contextuelles.

Notre méthode présente l'avantage d'estimer l'état d'un nombre plus important de points d'intérêt que les autres méthodes et requièrent moins d'information sémantique grâce à l'association des deux modules et des informations de profondeur. Une estimation de l'interaction inter-objet est rendue possible pour étendre le nombre de points considérés en étudiant le voisinage des individus et des points préalablement considérés comme dynamiques. Cette approche permet d'obtenir un algorithme de SLAM visuel robuste aux environnements dynamiques comme l'est le cortex visuel humain. Ainsi, cette solution pourra servir de système homothétique au système visuel humain pour la conception du futur algorithme de perception du mouvement propre. Cependant, les limitations du module de segmentation présentées en section 5.5.6 et l'importance de ce module pour le bon fonctionnement de notre approche ont mis en lumière l'impact de sa défaillance sur les performances. En effet, il a été observé que la segmentation produite par le module était peu précise ou défaillante en cas d'images perturbées. Afin de surmonter cette limite, une robustification du module de segmentation dans le cas d'acquisition d'images perturbées a été effectuée et détaillée dans le chapitre suivant.

6 - Vers un SLAM dynamique et robuste

6.1 . Introduction

La robustesse des méthodes de détection et segmentation d'objet en présence de distorsions d'images est un enjeu majeur en vision par ordinateur. En effet, l'intégration de ces modèles dans des systèmes embarqués opérant dans des environnements réels a mis en lumière leur faible robustesse contre des situations particulières ou critiques peu présentes dans les bases de données utilisées pour les entraîner. Partant de ce constat, deux solutions existent pour rendre ces modèles robustes dans des environnements non-contrôlés ; la conception d'un modèle intrinsèquement robuste à un ou plusieurs types de distorsion, ou l'application d'une augmentation de données contenant diverses distorsions lors de l'entraînement de ces modèles.

La première approche permet d'obtenir une robustesse plus importante, mais nécessite une architecture du réseau spécifique généralement pour un seul type de distorsion. Cette approche plus complexe nécessite une modification en profondeur des modèles de segmentation pour atteindre le résultat souhaité. La seconde approche, moins sophistiquée, permet une amélioration moins importante mais plus globale de la robustesse. L'intégration de plusieurs types de distorsion dans la base de données permet d'entraîner le modèle sur tous les cas de figure.

Dans notre cas, nous avons privilégié la deuxième approche en raison de sa flexibilité de déploiement et son impact plus global sur les performances du processus de détection d'objets. Ainsi, dans ce chapitre, nous détaillons l'étude de l'impact des distorsions sur les performances des méthodes de détection et segmentation d'objets, et plus particulièrement les approches basées sur l'apprentissage profond appliquées au SLAM. Pour ce faire, nous avons été amenés à construire une base de données originale et dédiée à cette étude. Nous détaillons la base de données créée pour l'amélioration de la robustesse de la segmentation/détection d'objet dans le cas d'images affectées de diverses dégradations. Cette base de données est construite à partir de la base de données connues COCO que nous avons fortement enrichie de différents contenus sémantiques et de diverses distorsions photo-réalistes.
Les contributions majeures de cette base de données sont résumées ci-après :

- L'introduction et la prise en compte, pour la première fois, de distorsions appliquées localement sur les objets ou zones d'intérêt.
- Des distorsions cohérentes avec le contexte de la scène pour un meilleur

réalisme.

- Une application photo-réaliste des perturbations atmosphériques (pluie, neige et brouillard).
- Une classification de scènes/environnements intérieur et extérieur.

Les avantages de cette base de données et sa conception ainsi que l'étude approfondie sur les méthodes de l'état de l'art de détection d'objet seront développés plus loin. Tout d'abord, une évaluation de l'impact des distorsions sur ces méthodes soulignera le besoin de les rendre davantage robustes. Cela conduit à la solution d'augmentation de données pour montrer, à travers des résultats expérimentaux, l'amélioration de la robustesse de l'un des modèles de détection d'objets, considérés dans cette étude [Bol+19], vis à vis de l'ensemble des distorsions affectant la qualité des images.

Rappelons, que cette augmentation de données a pour objectif de rendre la méthode de segmentation YOLACT [Bol+19] utilisée dans notre SLAM plus performante dans le cas d'acquisition d'images perturbées par diverses dégradations. Ainsi, la méthode de segmentation d'objet choisie sera détaillée et l'impact de l'augmentation de données sur sa robustesse dans le cas d'images perturbées sera évaluée pour mettre en évidence l'efficacité de cette approche. Enfin, le modèle entraîné sur cette base perturbée sera intégré à notre méthode de SLAM visuel afin de mesurer l'apport de l'augmentation de données sur les performances de notre approche par rapport à ce même modèle entraîné sur des images non perturbées.

6.2 . Étude des distorsions du signal image dans le cadre du vSLAM

L'utilisation croissante de méthodes d'apprentissage profond dans le domaine du SLAM visuel, et plus particulièrement les modèles de détection et segmentation d'objets ont conduit la communauté scientifique à considérer la robustesse de ces modèles dans des environnements non contrôlés et qui correspondent à des cas souvent rencontrés en vision par ordinateur. En effet, la présence de mouvements brusques de caméra et d'objets, le mauvais réglage de la focale de la caméra, l'éclairage non-uniforme ou le contre-jour influent fortement à la fois sur les performances des algorithmes de vSLAM et les modèles de détection et segmentation d'objets. Partant de ce constat, des travaux ont été menés pour analyser l'impact de ces perturbations ou distorsions sur les modèles de segmentation d'images et de détection d'objets afin d'évaluer leur robustesse pour une meilleure optimisation de leur efficacité.

Nous considérons comme distorsion toute aberration contenue dans une image qu'elle soit liée aux conditions d'acquisition ou compression de l'image

(bruit d'acquisition, artefacts de compression), aux conditions atmosphériques (pluie, neige, brouillard), limitations et mauvais réglage des capteurs et systèmes optiques (flou de défocalisation, flou de mouvement d'objets ou instabilité de la caméra) ou d'origine photo-métrique (éclairage insuffisant, éclairage non-uniforme, saturation photo-métrique, etc). Dans la suite de notre étude, nous considérons ces distorsions comme des distorsions "globales" du fait de leur application homogène et relativement uniforme sur l'ensemble des images à l'inverse du nouveau type de distorsions que nous proposons par la suite et qui concerne les dégradations dites locales.

6.2.1 . Distorsions liées à l'acquisition et la compression de l'image

Les distorsions purement liées à l'acquisition du signal image sont nombreuses et peuvent être classées en deux catégories. La première concerne celles liées directement aux limitations des capteurs d'images. La seconde catégorie correspond aux dégradations dues aux conditions d'acquisition du signal image. En général, les deux sont liées du fait des limitations des capteurs d'images dans certains environnements non contrôlés. Ces distorsions se manifestent sur l'image sous différentes formes impactant ainsi la qualité de l'image. C'est au niveau de la première opération de conversion du signal optique à travers la chaîne d'acquisition et de stockage de l'image qu'apparaissent les dégradations majeures de l'image. La figure ci-dessous illustre quelques distorsions typiques du signal image (voir la figure 6.1).

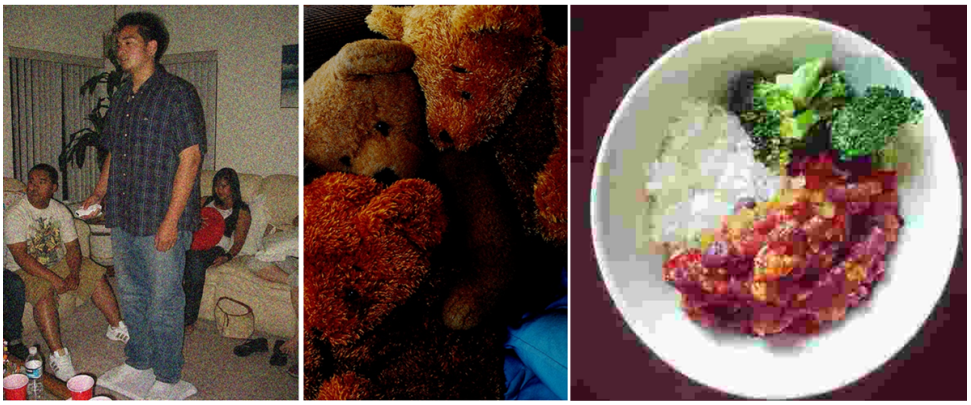


Figure 6.1 – Distorsions au niveau de l'acquisition de l'image.

- Bruit dans les images numériques : Le bruit est présent dans les images numériques pour diverses raisons liées aux propriétés physiques des capteurs [BJ15]. On peut considérer comme bruit tout signal indésirable ou perturbant le signal utile. Il y a deux types de bruit : le bruit dit aléatoire et le bruit structuré souvent lié à des phénomènes physiques ou

artefacts inhérents aux systèmes d'acquisition, de stockage ou transmission du signal. Le bruit peut être additif, multiplicatif, corrélé ou non corrélé au signal. Dans ce qui suit, nous nous intéressons essentiellement au bruit blanc Gaussien additif et non corrélé au signal qui est probablement le plus fréquent [Bon09]. Il est souvent utilisé pour modéliser divers bruits d'origine thermique, quantique ou liés aux conditions d'acquisition (environnement, source d'éclairage, limitations des capteurs opto-électroniques). Dans le cas du signal image, le bruit Gaussien se manifeste comme un phénomène de dégradation perceptuelle d'aspect granuleux. Il affecte les pixels de façon aléatoire aussi bien dans le domaine spatial que photométrique.

- Artefacts de compression : Les artefacts de compression et plus particulièrement "l'effet de bloc" et le phénomène de Gibbs 2d (ou ringing effect) sont dus au processus de compression et codage avec perte tels que JPEG ou JPEG2000 [PB02]. En effet, la réduction d'information inhérente a la compression irréversible étant un processus de sélection de composantes pertinentes du signal utile à un effet de filtre passe bas. Cela s'accompagne inévitablement de réduction de qualité d'image en raison de la suppression de certains détails fins du signal original. Les effets de bloc générés par la compression JPEG est un artefact dû essentiellement au traitement et quantification des coefficients de la TCD-2D (Transformée en Cosinus Discret) par bloc de façon indépendante. L'effet d'oscillations irrégulières au niveau des contours d'objets ou "ringing effect" qui apparaît dans le codage JPEG2000 est du essentiellement a la décimation des coefficients d'ondelettes [SCE01].
- Défaut de contraste : la visibilité des détails peut être affectée en raison du faible contraste dû à un mauvais éclairage, de la faible dynamique du capteur image ou sa réponse non-linéaire ayant un effet de compression de la dynamique, ou d'une source d'éclairage parasitant l'acquisition du signal image. Le faible contraste peut être aussi lié à la faible résolution du capteur produisant ainsi un effet de flou au niveau des contours et des détails fins du signal image. Un autre problème de contraste peut provenir des conditions d'acquisition d'image telles que le contre-jour, saturation de couleurs ou la réflexion spéculaire pour ne citer que les plus limitations les plus connues.

L'ensemble de ces distorsions est décrit par des phénomènes bien connus de la communauté de la vision par ordinateur qui les a caractérisés et a proposé des méthodes pour les corriger. Cependant, avant l'avènement du paradigme de l'apprentissage profond, les méthodes existantes employaient des approches blasées sur la modélisation du signal soumis au phénomène de dégradation et une information a priori du signal idéal a estimer [DTV08]. Actuellement, les approches utilisées reposent essentiellement sur la construction

de bases de données nécessaires au développement de solutions basées sur les modèles et architecture de réseaux de neurones convolutionnels [RJ22].

6.2.2 . Distorsions dues aux conditions atmosphériques

Les distorsions liées aux conditions atmosphériques intègrent les dégradations résultant de phénomènes atmosphériques agissant sur l'environnement capté par la caméra (voir la figure 6.2). Dans cette étude, nous nous limitons à la pluie, le brouillard et la neige qui peuvent être présents en environnement extérieur. La prise en compte de ces distorsions est particulièrement importante dans le cadre de véhicule ou de systèmes robotiques autonomes opérant en extérieur tels que la voiture autonome, le drone, etc.

La génération de ces distorsions se fait le plus souvent à l'aide de filtre extrait d'image reproduisant en laboratoire ces phénomènes (gouttelette d'eau ou fumée sur fond noir). La génération de telles distorsions par ordinateur s'appuie sur la modélisation de ces phénomènes atmosphériques. Cependant, ces modèles ne fournissent que des images plus ou moins proches de la réalité. Par exemple le modèle de génération de fumée ou de brouillard base sur le modèle de bruit de Perlin (Perlin noise model) néglige plusieurs paramètres physiques liés au phénomène réel [Per85]. Une idée plus attrayante consiste à utiliser une technique hybride où l'on combine une image réelle du phénomène considéré avec l'image originale par la technique de mélange de signaux appelée "image blending" [Kha+20b].

Figure 6.2 – Exemples de distorsions liées aux conditions atmosphériques (pluie à gauche et brouillard à droite).

C'est cette dernière solution que nous avons adaptée et améliorée en tenant compte de l'information profondeur de la scène, issue du capteur RGB-D, pour moduler le poids de la dégradation. Cette technique permet de générer des images avec plus de photo-réalisme [BMB22a]. La prise en compte de l'information profondeur dans la génération d'images dégradées constitue une de nos contributions originales dans ce travail, d'après nos connaissances des bases de données images et vidéos connues à ce jour.

6.2.3 . Distorsions liées au défaut de réglage et aux limitations optiques

Les distorsions liées au mauvais réglage optique (défocalisation ou défaut de mise au point) ou aux limitations de la réponse temporelle du capteur (voir la figure 6.3) se manifestent par un effet de flou. Dans le cas de mouvements brusques de la caméra, la vitesse de l'obturateur de la caméra peut être insuffisante pour effectuer une acquisition rapide permettant d'éviter un temps d'intégration du signal trop long conduisant ainsi à élargir la réponse optique (l'image d'un point ou petit élément est étalée).

Ce phénomène produit alors un effet de flou dans les zones où le temps d'acquisition a été relativement trop long par rapport à la vitesse de déplacement de l'objet ou de la caméra. De fait, cette perturbation s'applique à l'ensemble de l'image de manière plus ou moins homogène en fonction du type de mouvement de la caméra (translation/rotation) et du contexte de scène (position des éléments de la scène par rapport à la caméra). Dans le cas du déplacement d'un ou plusieurs objets le flou se manifeste uniquement au niveau des contours et détails de ces derniers. Dans ce dernier cas, le flou est dit local. C'est ce ce dernier cas qui est très peu considéré dans la littérature et que nous prenons en compte dans notre étude. Cela constitue une autre contribution originale de notre étude.

Figure 6.3 – Distorsions photo-métriques. Flou de mouvement à gauche et flou de défocalisation à droite.

6.2.4 . Étude de l'impact des distorsions sur la détection d'objet

La détection d'objets dans des séquences vidéo ou des images fixes est un sujet de recherche de grand intérêt compte tenu des nombreuses applications dans le domaine de la vision par ordinateur [Dol+11; PP99]. Avec le développement des méthodes d'apprentissage profond et la disponibilité de nombreuses bases de données dédiées à cette problématique, ce domaine

de recherche a connu une réelle avancée. Une étude complète sur les approches de détection d'objet basées sur l'apprentissage profond est fournie dans [Jia+19b]. Cependant, la plupart du temps, les bases de données disponibles ne tiennent pas compte des scénarios réels et en particulier des images et des vidéos capturées dans un environnement non contrôlé et donc affectées par divers types de distorsions. En effet, il a été montré dans de nombreuses études que les performances des modèles de détection d'objet sont fortement affectées par la qualité des images [BK19 ; Lin+23 ; CC20b ; Che+21 ; BMB22a]. Il convient de noter que le nombre et les types de distorsions pris en compte dans ces études et l'ensemble de données dédié existant sont limités. De plus, le cas de distorsions multiples apparaissant simultanément n'a pas été pris en compte dans les études d'évaluation des performances des méthodes de détection d'objet.

Plusieurs scénarios de distorsion ont été envisagés dans quelques études sur l'évaluation de la qualité vidéo, mais dans des contextes limités [Kha+20a ; Gom+22 ; NRC19]. Certains travaux intéressants ont étudié l'impact de diverses distorsions sur les performances des architectures des modèles de détection d'objet basés sur les réseaux de neurones convolutionels [AW19 ; Mic+19]. Certains articles traitent de l'impact de diverses distorsions sur les performances de détection, comme l'article [AW19], qui tente d'évaluer la robustesse de certaines architectures réseau de neurones convolutionels face aux transformations géométriques. Une étude comparative entre les architectures basées sur les réseaux de neurones profonds (ResNet-152, VGG-19, GoogLeNet) et la performance de l'observateur humain a été menée dans [Gei+18]. Dans cette étude, douze distorsions courantes sont considérées dans l'évaluation des performances des méthodes de reconnaissance d'objets.

Il a été démontré que ces architectures basées sur des réseaux de neurones profonds fonctionnent remarquablement bien dans le cas des distorsions sur lesquelles l'apprentissage a été effectué. Cependant, cette performance diminue, par rapport à celle de l'observateur humain, pour des distorsions peu perceptiblement visible ou complètement invisibles. Une autre étude intéressante sur l'évaluation de la robustesse des modèles de classification des images de la base de données Imagenet et soumises à quinze distorsions artificielles à cinq niveaux de sévérité, a révélé l'effet de la dégradation de la qualité de l'image sur la stabilité du processus d'apprentissage pour diverses architectures basées sur réseaux de neurones profonds [HD19]. Récemment, une étude de référence sur la robustesse de nombreux modèles de détection d'objets vis a vis de 15 types de distorsions globales et d'images stylisées via un processus d'augmentation des données a été réalisée dans [Mic+19].

Cependant, toutes ces études se limitent à un faible nombre de types de distorsions et ne prennent pas en compte les distorsions locales qui correspondent réellement aux scénarios présents dans des environnements réels. En effet, si l'on prend par exemple le flou dû au mouvement, il est le plus souvent simulé de manière globale dans les bases de données existantes par application globale d'un filtre passe bas de type Gaussien, par exemple, a tous les pixels de l'image sans distinction (pixels mobiles versus pixels fixes). Or, nous savons que dans une scène observée, il peut y avoir des objets se déplaçant à différentes vitesses et dans différentes directions ; cela engendre du flou de mouvement localisé au niveau des contours d'objets appelé aussi flou cinétique. Ce flou s'applique uniquement aux objets en mouvement de manière plus ou moins proportionnelle à leur vitesse de mouvement, leur éloignement de la caméra et leur localisation dans le champ de vision (centrale/périphérique). Il en va de même pour le flou de défocalisation, qui dépend de la profondeur des objets de la scène filmée. La prise en compte de ces aspects importants souvent négligés, en raison de leur complexité, dans le études connues a ce jour, d'après notre connaissance, constitue une de nos contributions majeures réalisées dans ce travail.

En conséquence, nous avons pense qu'il était pertinent et judicieux de mener une étude complémentaire sur l'impact des distorsions locales et globales sur les méthodes de détection d'objet en intégrant plusieurs niveaux de complexité ; à savoir la prise en compte de la profondeur des objets, la discrimination entre les cinétiques des objets, l'aspect local de certaines distorsions et d'autres paramètres physiques et géométriques permettant de donner a notre étude plus de réalisme et de crédibilité. À cet effet, nous avons conçu une base de données issue de la base MS-COCO [Lin+14], dans laquelle nous avons appliqué les deux types de distorsions en considérant le contexte de scène tel que la profondeur, l'environnement (intérieur/extérieur), la nature des objets, etc. Dans notre base de données, nous avons pris en compte ces aspects et d'autres tels que les effets d'éclairage qui varient avec la profondeur et la géométrie des objets pour déduire l'orientation des perturbations locales. Nous avons adopté la même approche concernant les distorsions dues aux phénomènes atmosphériques tels que la pluie et le brouillard qui s'appliquent en respectant le principe d'atténuation ou accentuation en fonction de la profondeur des différents objets et régions de la scène observée.

À la lumière de cette étude indispensable pour une meilleure conception de solutions robustes de segmentation et de détection d'objet, nous introduisons notre méthode d'augmentation de données destinée à l'amélioration des performances de notre méthode de SLAM visuel en exploitant les résul-

tats et données développés dans la partie précédente.

6.2.5 . Méthodes d'apprentissage profond appliquées au SLAM

Au cours des dix dernières années, le SLAM visuel et de nombreux domaines de recherche appliquée en vision artificielle ont été de plus en plus repensés à la lumière des nouveaux développements des méthodes basées sur l'apprentissage automatique. Il est communément établi que le SLAM visuel a un lien direct avec la vision par ordinateur. Ce dernier domaine, qui a plus de 50 ans, a été révolutionné avec l'avènement des méthodes et des technologies d'apprentissage automatique. Nous avons observé une amélioration des performances des méthodes de SLAM avec l'amélioration des méthodes basées sur l'apprentissage profond. Afin de migrer vers les processus vSLAM modernes, des méthodes basées sur l'apprentissage profond ont été intégrées au processus du SLAM, ce qui a optimisé certains modules du SLAM spécifiques (relocalisation, détection de fermeture de boucle, etc.) ou amélioré les performances globales.

Habituellement, des optimisations spécifiques tentent de mieux estimer la localisation et la cartographie en améliorant les estimations d'échelle et de profondeur, ainsi que l'extraction et l'association des points caractéristiques. La prise en compte de la dynamique de l'environnement est l'un des sujets du moment pour améliorer les performances du SLAM en environnement réel. En effet, les tâches de localisation et de cartographie SLAM visuelles utilisent les points caractéristiques de l'environnement, mais elles ont besoin de points de repère statiques pour obtenir une estimation pertinente. Cependant, les environnements statiques sont assez rares dans le monde réel, ce qui implique que de nombreux points de caractéristiques dynamiques sont pris en compte lors du module d'Odométrie Visuelle. Le SLAM dynamique consiste à déterminer les objets dynamiques présents dans un environnement pour les rejeter du module d'Odométrie Visuelle.

Ainsi, l'amélioration de la sélection des points caractéristiques dans le SLAM moderne diminue l'erreur d'estimation accumulée de la localisation et de la cartographie. Pour distinguer les objets statiques des objets dynamiques, de nombreuses méthodes basées sur l'apprentissage profond ont été introduites dans le processus du SLAM [Bes+18; Bes+20; Li+21]. Le processus de discrimination utilisé dans ces procédés est réalisé en opérant en trois étapes, à savoir :

- La détection d'objets dynamiques potentiels selon des classes prédéfinies d'objets dynamiques.
- La segmentation de ces objets détectés (facultatif).
- La détection parmi les objets potentiellement mobiles de ceux qui sont

réellement dynamiques par estimation du flux optique ou raisonnement géométrique.

Ainsi, la vision par ordinateur et les méthodes basées sur l'apprentissage profond sont nécessaires pour exécuter les différentes tâches du processus de SLAM moderne, que ce soit l'extraction de l'information, l'optimisation ou la perception de l'environnement. Cette section présente les principaux concepts des méthodes basées sur l'apprentissage profond pour la détection d'objets et la segmentation d'objets.

La détection d'objet

La détection d'objets est l'un des problèmes les plus importants et les plus étudiés en vision par ordinateur. Cependant, malgré le nombre important de travaux consacrés a cette problématique, ce domaine de recherche est encore un sujet ouvert et suscite de nombreux débats [Zou+23]. En effet, une des difficultés majeures dans la recherche de solutions efficaces pour la détection d'objets réside dans la définition et la compréhension de la notion "objet dans une image". En général, le concept d'objet en vision par ordinateur est lié a l'application et le but recherchés dans l'analyse et l'interprétation de la scène observée. Les méthodes classiques de détection d'objet reposent en général sur le concept fond-versus-objet et opèrent une opération de segmentation basée sur des critères et métriques construits a partir de descripteurs locaux ou globaux tels que la forme, la couleur, ou d'autres informations statistiques. Certaines des méthodes basées sur l'apprentissage automatique exploitent des descripteurs ou attributs de bas niveau, tel que l'histogramme de gradient orienté (HOG), pour la détection d'objet. Ces méthodes sont souvent associées à une base de données ou comparées à des modèles structurés. L'introduction de l'apprentissage profond et l'amélioration des unités de traitement graphique (GPU) ont permis de concevoir des algorithmes capables de détecter et de segmenter efficacement certaines classes d'objets de manière plus fiable et généralisable. L'utilisation des **réseaux de neurones profonds** appliqués à la détection d'objets a permis d'atteindre une meilleure précision et robustesse en extrayant de manière automatique les informations pertinentes dans les images pour caractériser leurs contenus sémantiques et différencier entre les divers types d'objets les constituants. La détection d'objet se compose des étapes essentielles présentées dans la figure 6.4, qui permettent l'exploitation des différentes informations necessaires a la realisation de cette tache importante pour l'analyse et l'interprétation de la scène observée.
La détection d'objet peut être réalisée en deux étapes. La première est la classification des objets présents dans l'image suivie de leur caractérisation en vue de leur localisation dans l'espace image 2D. L'étape optionnelle suivante

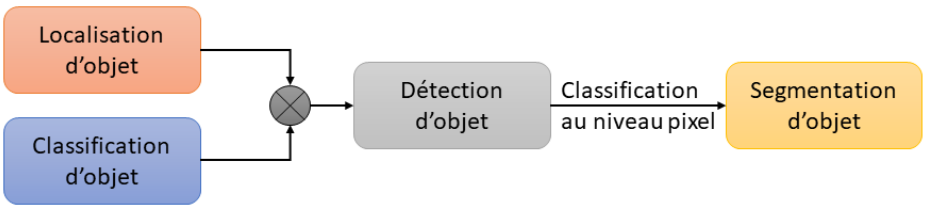


Figure 6.4 – Illustration du processus de reconnaissance d'objet.

est la segmentation de l'objet détecté, qui est le raffinement de sa localisation pour obtenir sa forme. Cette forme est ajoutée à l'image via un masque d'objet qui informe de son emplacement au niveau du pixel. De nombreuses méthodes de détection d'objets ont été intégrées au processus de SLAM dynamique comme Mask-RCNN [He+17a], YOLOv3 [RF18] et YOLOv4 [BWL20]. La littérature récente détaille l'état de l'art des méthodes de détection d'objets [Zou+19 ; Liu+20].

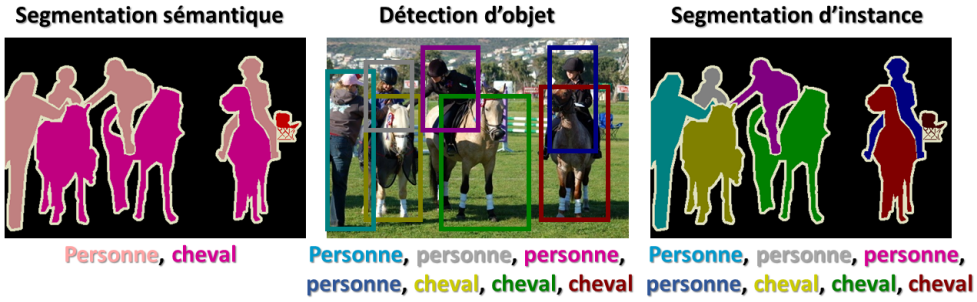


Figure 6.5 – Illustration des différents types de segmentation d'objet.

La segmentation d'objet

La segmentation a l'avantage sur la détection de fournir des informations sur la forme et l'emplacement des objets. Un masque pour chaque objet de l'image est créé, permettant d'analyser et d'interpréter la scène observée. Il est utilisé dans diverses applications telles que le diagnostic basé sur l'imagerie médicale, la navigation autonome, la reconnaissance faciale, la robotique, les systèmes de contrôle du trafic et la localisation d'objets. De ce fait, nous le considérons comme une étape critique pour les tâches ultérieures. Il existe deux principaux types de segmentation, à savoir la **segmentation sémantique** et la **segmentation d'instance**. Le résultat de la segmentation sémantique est un processus de classification où tous les pixels appartenant à une classe donnée sont regroupés comme une entité. À l'inverse, la segmentation

d'instance distingue les objets d'une même classe. Ainsi, la segmentation sémantique ne catégorise pas les pixels, mais regroupe et identifie les objets similaires comme une seule classe à la place. Les classes d'objets identifiées sont ensuite discriminées par segmentation d'instance. La figure 6.5 illustre ces deux types de segmentation.

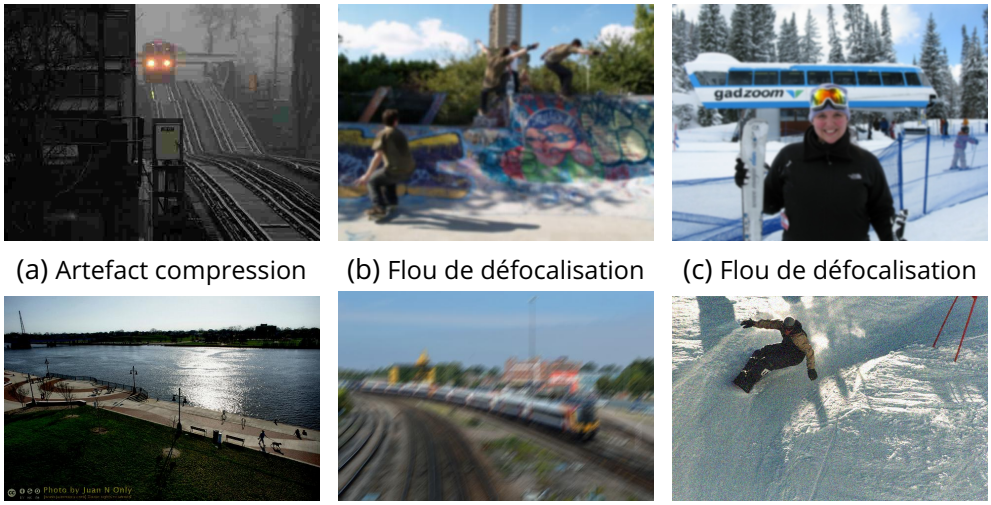
Grâce à leur simplicité, de nombreuses méthodes de segmentation classiques étaient couramment utilisées par la communauté de la vision par ordinateur. Mais, en raison de certaines de leurs limites, telles que leur manque de flexibilité et leur sensibilité aux environnements variables, d'autres solutions basées sur des approches d'apprentissage profond ont été développées comme DeepLab [Che+16a; Che+18], MASK-RCNN [He+17b], SegNet [BKC16] et PANet [Liu+18]. La complexité des scènes nécessite des techniques plus sophistiquées pour atteindre une meilleure robustesse et précision. Ainsi, l'introduction de l'apprentissage automatique via les CNN a permis d'atteindre ces objectifs. Les nouvelles méthodes de segmentation sont évaluées selon deux catégories de critères : les performances techniques en termes de précision et robustesse, et le temps d'exécution. L'étude menée dans [Min+21] donne un aperçu sur les méthodes de segmentation et propose un apprentissage global rapide pour effectuer cette tâche primordiale dans les systèmes de vision par ordinateur. Parmi les méthodes existantes, seule la méthode YOLACT [Bol+19] satisfait le critère de temps-réel pour une précision de segmentation satisfaisante, ce qui nous a conduit à nous orienter vers l'utilisation de cette méthode.

6.3 . Distorsions complexes : prise en compte du contexte de scène

Une base de données issue de la base MS-COCO [Lin+14] a été créée [Beg+23] afin d'étudier l'apport d'une augmentation de données sur les performances du modèle de segmentation d'objet intégré à notre méthode de SLAM. Cette base d'image a été perturbée selon des distorsions globales, locales et atmosphériques pour tenter de fournir au modèle de segmentation une robustesse à de nombreuses conditions d'acquisition d'image.

6.3.1 . Distorsions globales

Dans notre cas, le terme distorsion globale désigne les distorsions classiques qui s'appliquent de manière plus ou moins homogène à l'ensemble de l'image sans aucune considération du contexte de la scène. Ainsi, nous avons appliqué des distorsions globales pour certaines images résultant de conditions d'acquisition d'image (bruit, compression, changement de contraste) ou de caméra (flou de mouvement et de défocalisation) sans tenir compte du contexte de la scène (voir figure 6.6). Cependant, certaines images sont ac-



(a) Artefact compression (b) Flou de défocalisation (c) Flou de défocalisation

(d) Défaut de contraste (e) Flou de mouvement (f) Bruit Gaussien

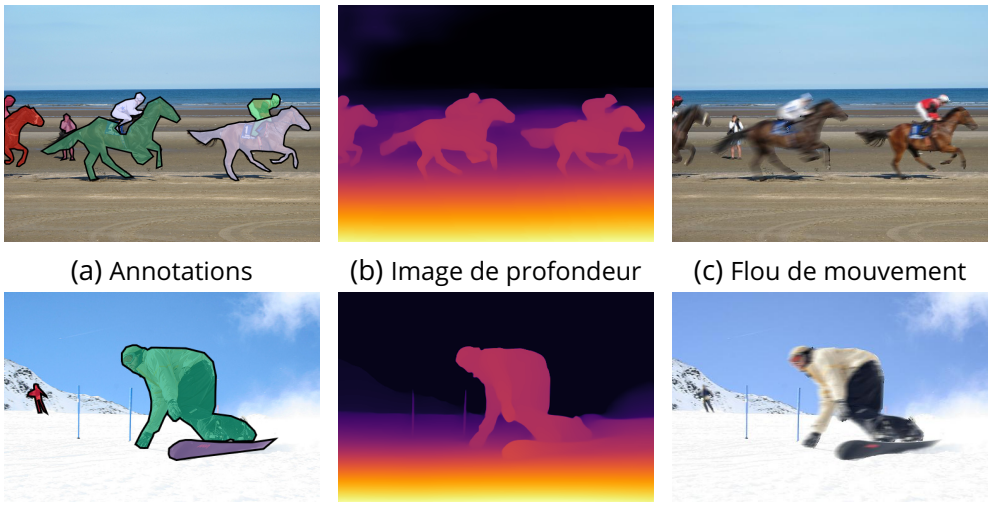
Figure 6.6 – Exemple de distorsions globales

quises dans des contextes particuliers qui nécessitent l'application de distorsions locales avec des approches plus sophistiquées et notamment dans le cas de dégradations atmosphériques. Les distorsions complexes générées selon notre approche exploitent les informations de profondeur de la scène, les informations de vérité terrain des annotations de la base d'images COCO (masques d'objet) et les types d'objet et de scène pour produire des distorsions complexes et photo-réalistes. Les informations sur la profondeur de la

scène sont obtenues à l'aide de la méthode d'estimation de profondeur MiDaS [Ran+20].

6.3.2 . Distorsion locale : Flou de mouvement local

Le flou de mouvement local est dû au mouvement des objets filmes par la caméra. Dans notre approche, aux annotations des objets contenus dans les images acquises nous appliquons un effet de flou localisé et qui tient compte d'autres paramètres physiques tels que la vitesse et l'orientation. Ce type de distorsion locale est illustré dans la figure 6.7. Cette distorsion locale nécessite



(a) Annotations (b) Image de profondeur (c) Flou de mouvement

(d) Annotations (e) Image de profondeur (f) Flou de mouvement

Figure 6.7 – Conception du flou de mouvement local.

que les masques de vérité terrain définissent la zone de pixels où le mouvement de flou doit être appliqué. De plus, le masque d'objet est également utilisé pour déterminer l'orientation de la distorsion grâce à une stratégie propre à la nature de l'objet (classes d'objets). Une autre double stratégie nous permet de calculer l'amplitude du mouvement à appliquer à chaque objet de l'image. L'orientation de l'objet est obtenue en calculant l'angle entre l'axe des abscisses et le grand axe de l'ellipse contenant l'objet. Ensuite, une stratégie de vérification de l'orientation est adaptée pour appliquer un flou de mouvement selon la nature de l'objet, comme indiqué dans le tableau 6.1.

Table 6.1 – Condition d'orientation par type d'objet.

Type d'objet	Orientations applicable
Personne	$[15°; 90°] \cap [-15°; 90°]$
Véhicule	$[-60°; 60°] \cap [120°; 240°]$
Animal	$360°$

Il est à noter que nous tenons compte de l'orientation relative de l'objet/personne en mouvement dans l'application du flou local. Par exemple, dans le cas d'un déplacement latéral d'un objet ou d'une personne, il faudrait ajuster l'estimation de l'orientation en ajoutant $90°$ pour caractériser le mouvement. Tout d'abord, un intervalle d'amplitude de mouvement est déduit de la nature de l'objet et des connaissances préalables sur la vitesse du type d'objet (voir tableau 6.2). Ensuite, une valeur d'amplitude de distorsion est calculée en considérant les profondeurs de l'objet considéré et celle des autres pour mettre cette valeur dans le contexte global de la scène.

Table 6.2 – Connaissance préalable de l'intervalle de vitesse par type d'objet.

Type d'objet	Intervalle de vitesse en km/h
Personne (intérieur)	[3 ; 7]
Personne (extérieur)	[5 ; 25]
Voiture / moto	[30 ; 130]
Camion/Bus	[30 ; 90]
Vélo/Chien/Chat/Ours	[10 ; 50]
Cheval	[20 ; 90]
Ski	[30 ; 60]
Surf/Bateau	[10 ; 30]

Ainsi, chaque valeur d'amplitude est obtenue en corrélant la nature et la profondeur des objets, ce qui garantit la cohérence globale des distorsions de mouvement de chaque flou local les unes par rapport aux autres. De plus, une vérification de l'interaction des objets est réalisée pour donner la priorité à l'amplitude et à l'orientation des objets de niveau supérieur sur les objets de niveau inférieur, comme indiqué sur la figure 6.8.

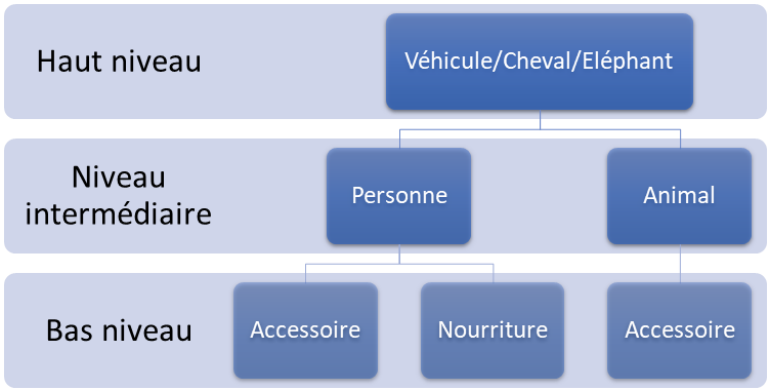
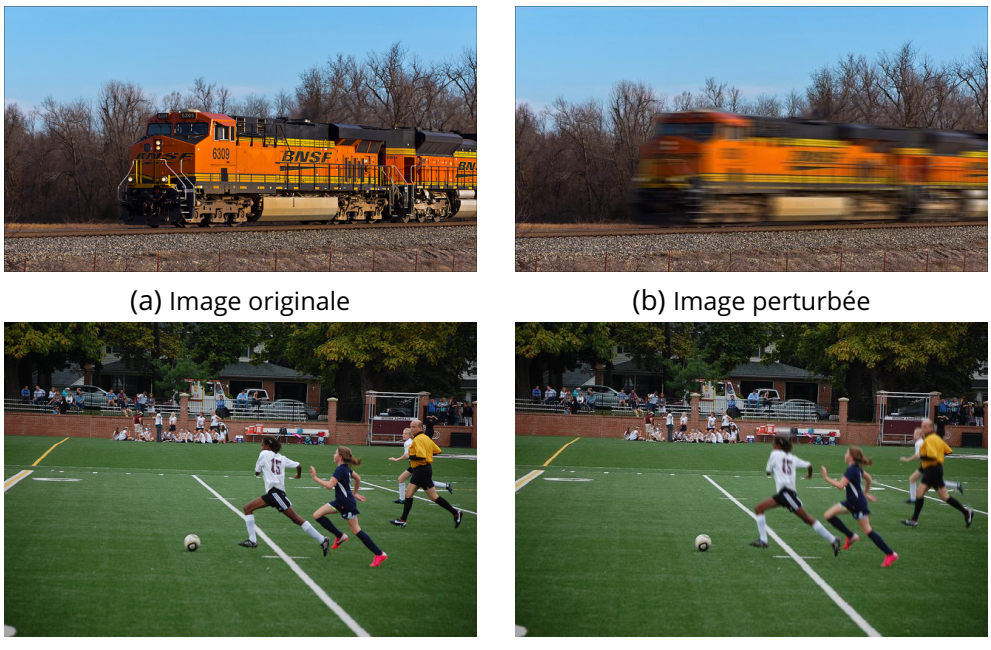


Figure 6.8 – Hiérarchie des interactions liée au type d'objet

Cette étape est effectuée en corrélant leur proximité de profondeur et les

boîtes englobantes qui se chevauchent pour garantir la cohérence de la distorsion des objets liés. Ainsi, l'amplitude et l'orientation des objets de niveau supérieur sont appliquées à ces objets en interaction de niveau inférieur. L'algorithme complet suit les étapes suivantes :

1. Trouver le contexte de la scène : ski, équitation, sport, skate ou surf en fonction des objets présents dans l'image.
2. Classer les objets : créer une super-classe d'objets en regroupant des objets afin de raisonner de manière globale (véhicule, personne, animal, nourriture, etc.).
3. Calculer la profondeur moyenne de chaque objet annoté.
4. Calculer l'amplitude et l'orientation en tenant compte de la profondeur et de la nature des objets pour la distorsion de chaque objet individuellement.
5. Déterminer l'interaction entre les objets en exploitant leur proximité spatiale (positions x,y,z) via leur profondeur et les boîtes englobantes se chevauchant afin d'appliquer la même distorsion a ces objets en interaction.
6. Trier les objets selon leur profondeur pour ajuster leur amplitude de mouvement pour une cohérence globale de la scène et des distorsions.



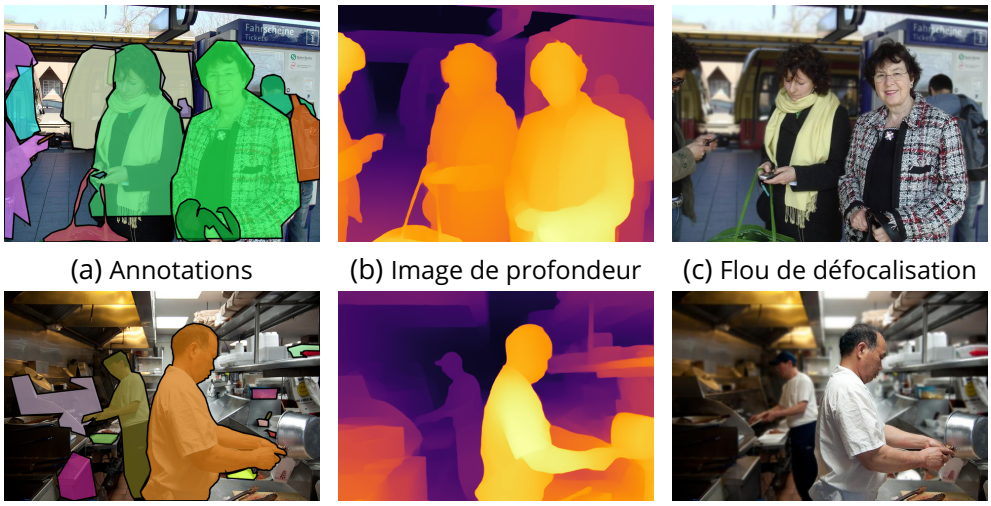
(a) Image originale (b) Image perturbée

(c) Image originale (d) Image perturbée

Figure 6.9 – Illustration du flou de mouvement local.

6.3.3 . Distorsion locale : Flou de défocalisation local

Le flou de défocalisation local est la distorsion représentant les phénomènes de flou résultant de la mise au point de l'objectif optique de la caméra sur l'arrière plan ou l'avant-plan des scènes. Pour produire une distorsion de flou de défocalisation réaliste, nous avons effectué un multi-seuillage qui fournit trois zones distinctes liées à la profondeur de la scène. Le rendu de cette approche est illustré en figure 6.10 où l'on distingue les masques, les images de profondeur et les images perturbées. Ce processus de multi-seuillage est



(a) Annotations (b) Image de profondeur (c) Flou de défocalisation

(d) Annotations (e) Image de profondeur (f) Flou de défocalisation

Figure 6.10 – Conception du flou de défocalisation locale.

effectué à l'aide d'une fonction non linéaire Ω donnée par :

$$\Omega(x) = 1 - \frac{1}{1 + \exp\left(-15(x - 0.5)\right)} \tag{6.1}$$

Où x représente la profondeur du point d'intérêt normalisée par la profondeur moyenne de l'objet le plus proche et estimée comme suit :

$$x = \frac{threshold - p_i}{threshold} \tag{6.2}$$

La figure 6.11 illustre la subdivision de l'image en trois classes (plans) différentes par multi-seuillage de la profondeur de la scène observée. Le premier plan correspond aux profondeurs avec des coefficients de seuil supérieurs au seuil haut, le plan intermédiaire aux coefficients compris entre les seuils haut et bas, et le fond aux coefficients inférieur au seuil bas (voir fig.6.11). Le résultat de ce processus de seuillage est illustré en fig.6.12 pour l'exemple de la première ligne de la figure 6.10. Ensuite, les profondeurs moyennes δ, δ_m et δ_b des trois sols sont calculées pour effectuer un flou de défocalisation proportionnel lié à la profondeur.

Nous avons appliqué des amplitudes de flou de défocalisation cumulatives λ, λ_m et λ_b allant du premier plan à l'arrière plan sur chaque zone.

$$\lambda = 0.5 + \frac{\delta_f - threshold}{threshold} \cdot 1.5 \tag{6.3}$$

$$\lambda_m = \lambda + \frac{\delta_m - threshold}{threshold} \cdot 1.2 \tag{6.4}$$

$$\lambda_b = \lambda_m + \frac{\delta_b - threshold}{threshold} \cdot 1.2 \tag{6.5}$$

Les trois zones de l'image sont ainsi dégradées de façon différenciée et locale en fonction de l'amplitude de flou de défocalisation correspondant (λ, λ_m et λ_b). L'image dégradée résultante I_d est obtenue par fusion des trois composantes comme illustré sur la figure 6.10.

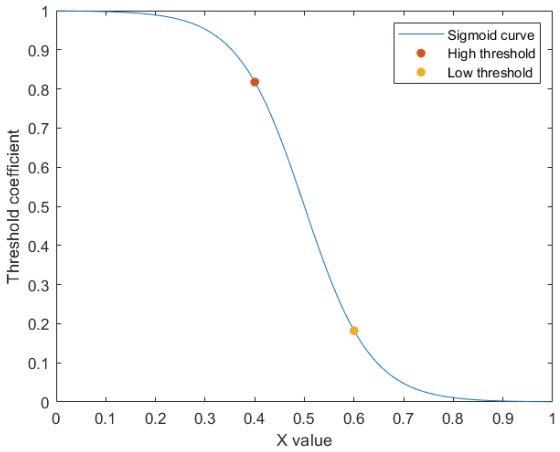


Figure 6.11 – Allure de la fonction utilisée dans le processus de multi-seuillage

Figure 6.12 – Seuils liés à la profondeur (premier plan en bleu, plan intermédiaire en vert et arrière plan en rouge)

Le flou de défocalisation généré ainsi est davantage réaliste, car l'application de la distorsion se fait de manière douce et en tenant compte de l'information

spatiale et notamment la profondeur. En opérant ainsi, la prise en compte de la position relative des objets dans les scènes permet d'ajuster le calcul des différents plans afin de générer de manière automatique selon l'algorithme 6 des distorsions adaptées au contexte de la scène observée.

Notons que le contexte de scène correspond à la position relative des objets dans la scène et les uns par rapport aux autres, mais également à la nature et l'orientation des objets. La méthode de génération de la distorsion de flou de défocalisation est ainsi automatique et généralisable à n'importe quel contexte de scène. Le premier plan intégrera toujours l'objet le plus proche de la caméra pour éviter un dysfonctionnement lorsque les objets les plus proches sont lointains. La figure 6.13 illustre l'impact de la distorsion ainsi générée par notre méthode sur une image originale et l'algorithme de défocalisation local proposé et décrit dans l'algorithme 6.

Algorithm 6 Algorithme de flou de défocalisation locale

Input : Image perturbée I_d

Trouver la profondeur de l'objet le plus proche : threshold
High threshold th_f = 0.8176
Low threshold th_b = 0.182
for each $p_i \in I$ **do**
  $\sigma = \frac{threshold - p_i}{threshold}$
  $\Delta(p_i) = 1 - \frac{1}{1+\exp(-15(\sigma-0.5))}$
  if $threshold > p_i$ **then**
    $Premier\ plan \leftarrow p_i$
  end if
  if $th_f <= \Delta(p_i)$ **then**
    $Premier\ plan \leftarrow p_i$
  end if
  if $th_b <= \Delta(p_i)$ **then**
    $Plan\ intermediaire \leftarrow p_i$
  end if
  if $th_f >= \Delta(p_i)\ \ \&\ \ th_b > \Delta(p_i)$ **then**
    $Arriere\ plan \leftarrow p_i$
  end if
  $\delta_f \leftarrow$ Profondeur moyenne du premier plan
  $\delta_m \leftarrow$ Profondeur moyenne du plan intermédiaire
  $\delta_b \leftarrow$ Profondeur moyenne de l'arrière plan
end for

(a) Image originale

(b) Image perturbée

(c) Image originale

(d) Image perturbée

(e) Image originale

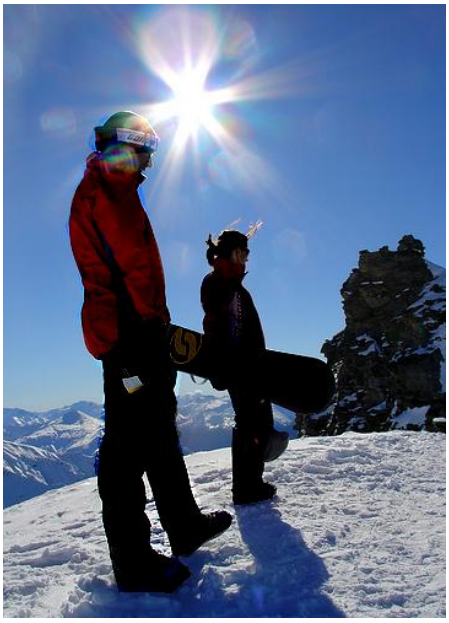
(f) Image perturbée

Figure 6.13 – Illustration du flou de défocalisation locale.

6.3.4 . Distorsion locale : Le contre-jour

La distorsion de contre-jour local implique l'application d'une modification de contraste local dans la zone du masque de l'objet. On essaie de reproduire l'effet de contre-jour résultant de la position des objets en fonction de celle du soleil ou la source de lumière. Pour cela, une remise a l'échelle de la dynamique de niveau de gris de l'image d'origine est réalisée pour la normaliser dans une gamme de valeurs prédéfinie par des intervalles choisis par ajustement aléatoire. Cette transformation de contraste est illustrée dans la figure 6.14.

(a) Image originale



(b) Image perturbée

Figure 6.14 – Distorsion de contre-jour

Les intervalles de remise à l'échelle sont déterminés par une recherche manuelle et sont appliqués de manière aléatoire. L'opération de transformation de l'échelle de niveaux de gris produit un effet d'assombrissement de l'image. De cette transformation résulte deux niveaux d'assombrissement, comme indiqué dans le tableau 6.3.

Table 6.3 – Intervalles de remise a l'échelle.

Intensité	Intervalles de remise a l'échelle			
Sombre	$[0.2; 1.0]$	$[0.2; 0.9]$	$[0.3; 0.9]$	$[0.3; 1.0]$
	$[0.35; 1.0]$			
Très sombre	$[0.2; 0.8]$	$[0.3; 0.7]$	$[0.3; 0.8]$	$[0.35; 0.9]$
	$[0.4; 0.9]$			

6.3.5 . Distorsion atmosphériques : la pluie

Synthétiser la pluie de manière homogène sans prendre en compte l'information de profondeur de scène conduit a des images peu réalistes et susceptibles d'affecter la robustesse des solutions basées sur l'apprentissage à partir des données ainsi générées. En effet, la taille et la densité de la pluie dépendent de la distance à laquelle elle tombe par rapport à la caméra. Un objet proche sera affecté par une pluie qui apparaît plus dense et épaisse qu'elle ne l'est pour un objet éloigné. Notre algorithme de génération de pluie est ainsi

basé sur la méthode de l'algorithme 6 pour effectuer la classification des composantes de profondeur de la scène en trois classes, a savoir le premier plan, le plan intermédiaire et l'arrière plan. Chaque plan se voit attribuer un niveau d'intensité de pluie qui reproduit la densité de la pluie en fonction de la distance par rapport à la caméra (voir figure 6.15).

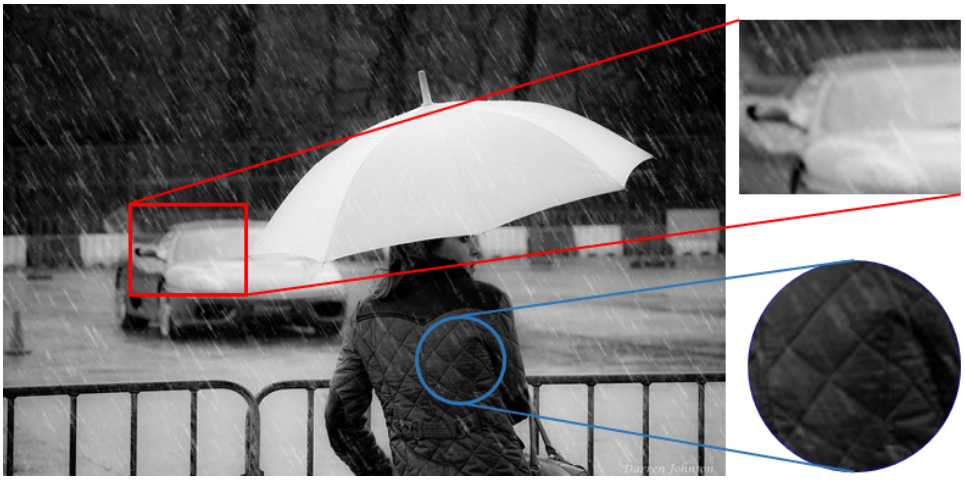


Figure 6.15 – Effet localisé de la distorsion de pluie

Cela implique l'utilisation de masques de pluie obtenus à partir d'images de la pluie produite dans des conditions expérimentales (voir figure.6.16). Nous extrayons de ces masques trois densités de pluie en effectuant un processus d'érosion et de dilatation de l'image de pluie.

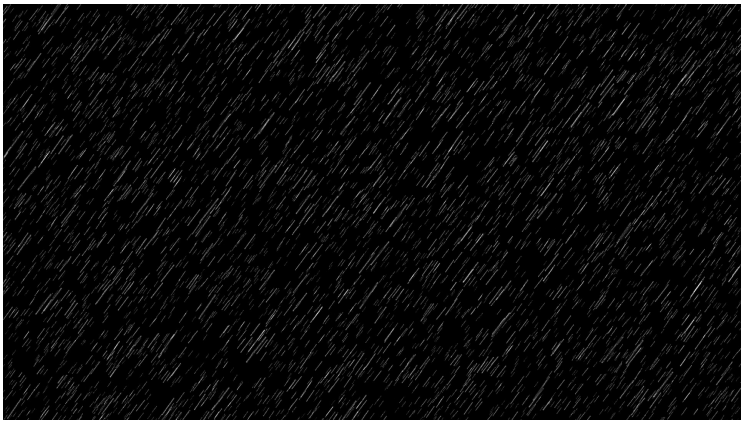


Figure 6.16 – Masque de pluie

Ces trois sous-masques sont appliqués pour chaque plan selon une constante aléatoire α de mélange, réalisant une mixture d'images comme illustré par la figure.6.15.

Il convient de noter que les sous-masques de pluie sont appliqués de manière cumulative du premier plan vers l'arrière plan selon le processus décrit ci-après.

$$I_f = 1 - ((1 - I) \cdot (1 - (\alpha \cdot R_f))) \tag{6.6}$$
$$I_m = 1 - ((1 - I_f) \cdot (1 - (\alpha \cdot R_m))) \tag{6.7}$$
$$I_d = 1 - ((1 - I_m) \cdot (1 - (\alpha \cdot R_b))) \tag{6.8}$$

Où I, I_f, I_m et I_d sont respectivement les images originales, de premier plan, plan intermédiaire et arrière plan dégradées. De même, R_f, R_m et R_b sont les trois sous-masques de pluie. Dans cette approche la pluie fine est appliquée uniquement a l'arrière plan de la scène distante pour un meilleur réalisme, comme le montre la figure 6.15.

Le processus de génération d'images avec pluie que nous proposons ici améliore la cohérence globale de la distorsion en considérant le lien spatial entre la profondeur de la scène et le phénomène atmosphérique comme illustré dans la figure 6.17.

(a) Image originale

(b) Image perturbée

(c) Image originale

(d) Image perturbée

Figure 6.17 – Exemples de distorsions atmosphérique : la pluie.

6.3.6 . Distorsion atmosphérique : Le brouillard

La génération de brouillard synthétique est une tâche complexe en vision par ordinateur. En effet, les modèles mathématiques de génération de brouillard existants, et notamment ceux basés sur le modèle de bruit de Perlin (Perlin Noise) [Per85], n'incluent pas certains paramètres physiques et de ce fait fournissent des images peu réalistes. Cela vient du fait que la prise en compte des paramètres physiques pertinents rend la modélisation complexe et dépendante de réglages multiples. Nous avons ainsi opté pour une solution plus pragmatique qui consiste à combiner l'image originale à un vrai phénomène de brouillard produit en laboratoire ou issu d'images de brouillard réel disponibles sur internet.

Un autre aspect non pris en compte dans les modèles de génération de brouillard dans les images est la profondeur des objets constituant la scène. Dans notre méthode, nous incluons cet aspect en utilisant des masques de brouillard extraits d'images de créations expérimentales (voir fig.6.18). Cela constitue une autre contribution originale dans notre travail. De nombreux masques ont été extraits d'un flux vidéo de fumée réelle pour fournir un large éventail de brouillard avec diverses densités et formes. Ces masques sont appliqués aux images d'origine, en les mélangeant de manière transparente grâce à un processus de pondération en fonction de la distance de l'objet par rapport à la caméra.

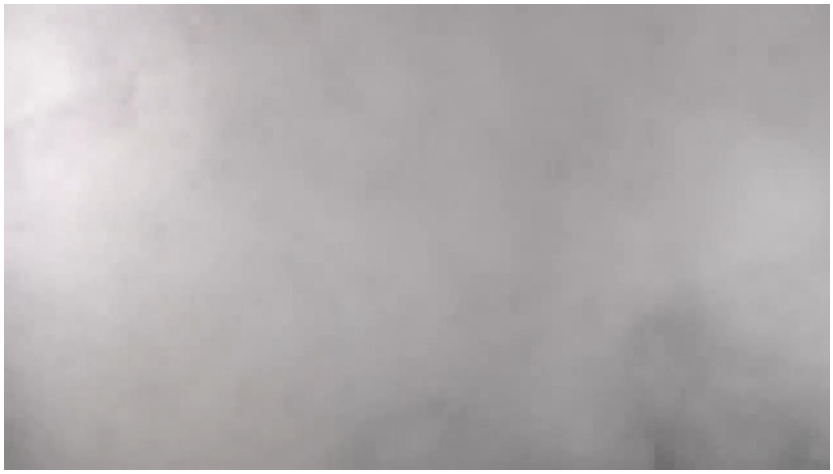


Figure 6.18 – Exemple de masque de brouillard utilisé dans notre base de données

Notons que l'application d'un masque de manière homogène produit un brouillard non réaliste. Il convient alors de considérer la profondeur de la scène pour appliquer le masque à l'effet de brouillard pour donner plus de réalisme a la scène ainsi générée. Ce masque de brouillard H est une matrice ou chaque

élément $\kappa(i,j)$ est proportionnel à la profondeur normalisée $Depth_n(i,j)$ de chaque pixel image $I(i,j)$ et une valeur constante α comme résumé dans l'algorithme 7. L'épaisseur du brouillard est alors proportionnelle à la profondeur, ce qui permet de générer des images de scènes beaucoup plus réalistes, comme le montre la figure 6.19.

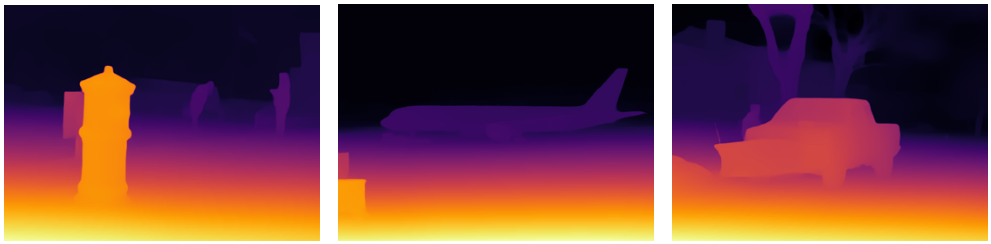
Algorithm 7 Algorithme de génération de brouillard

Input : Image I, masque de brouillard H
Output : Image perturbée I_d

α = 0.95
for chaque pixel $i, j \in I$ **do**
  $Depth_n(i,j) = \frac{Depth(I(i,j))}{Depth_{max}}$
  $\kappa(i,j) = \alpha \cdot Depth_n(i,j)$
  $I_d(i,j) = (1 - (1 - I(i,j)) \cdot (1 - (\kappa(i,j) \cdot H(i,j))) \cdot 255$
end for

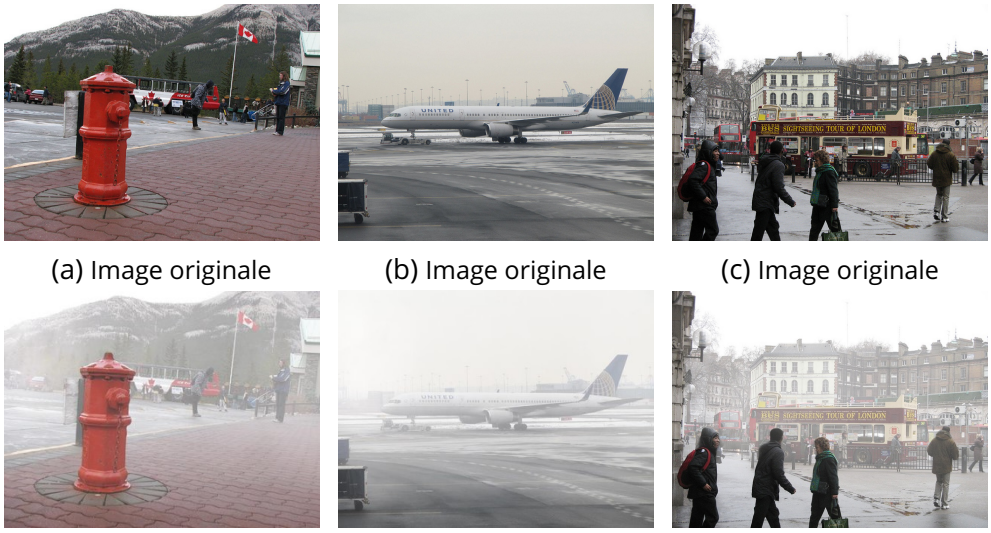
Cette stratégie produit une impression d'accumulation de brouillard qui semble plus épaisse selon la profondeur comme dans les scénarios réels. La figure 6.20 illustre bien ce phénomène d'accumulation de brouillard. L'image originale est aussi représente dans cette figure pour mieux percevoir le photoréalisme produit dans l'image dégradée générée par notre algorithme.

(a) Distorsion de brouillard (b) Distorsion de brouillard (c) Distorsion de brouillard



(d) Image de profondeur (e) Image de profondeur (f) Image de profondeur

Figure 6.19 – Distorsion atmosphérique : quelques exemples illustrant la distorsion de brouillard.



(a) Image originale (b) Image originale (c) Image originale

(d) Image perturbée (e) Image perturbée (f) Image perturbée

Figure 6.20 – Exemples de distorsions atmosphérique : le brouillard.

6.3.7 . Base de données CD-COCO

L'ensemble des distorsions décrites précédemment ont permis de construire notre base de données CD-COCO contenant des images perturbées synthétiquement suivant notre méthode et issue de la base MS-COCO [Lin+14]. Notre base de données CD-COCO comprend ainsi des jeux de données d'entraînement, de validation et de test avec une vérité terrain d'annotation complète provenant de l'ensemble de données MS-COCO d'origine. La base de données se compose de plus de 123 000 images avec 80 classes d'objets. Les trois jeux de données sont organisés comme suit :

- Jeu d'entraînement : 95 000 images
- Jeu de validation : 5 000 images
- Jeu de tests : 23 000 images

L'annotation de vérité terrain fournit les classes d'objets, les cadres de délimitation et les masques pour chaque image, qui peuvent être utilisés pour former des modèles de classification, de détection d'objets et de segmentation. La répartition des classes d'objets met en évidence la présence de nombreux objets ayant une forte probabilité d'être en mouvement dans la scène observée (voir figure 6.21). Par conséquent, ces objets, probablement en mouvement, pourraient produire une distorsion de flou de mouvement local dont l'étendue et l'amplitude augmentent avec leur vitesse.

Notre base d'images dégradées est composée de dix types de distorsion, cinq distorsions globales, deux distorsions atmosphériques globales et trois distorsions locales. Afin de générer les différentes distorsions de manière cohérente et pertinente, un premier balayage de toutes les images est effectué pour préparer le protocole d'attribution des distorsions en fonction du

contenu sémantique de la scène et du contexte. Ensuite, à l'aide d'un algorithme spécifique, les différentes distorsions sont automatiquement appliquées aux images préalablement annotées lors du premier traitement. Les images annotées en tant que distorsions globales sont ensuite dégradées par l'un des types de distorsion globale choisi au hasard avec une distribution uniforme. Une stratégie similaire est utilisée pour appliquer des distorsions atmosphériques appropriées aux images originales.

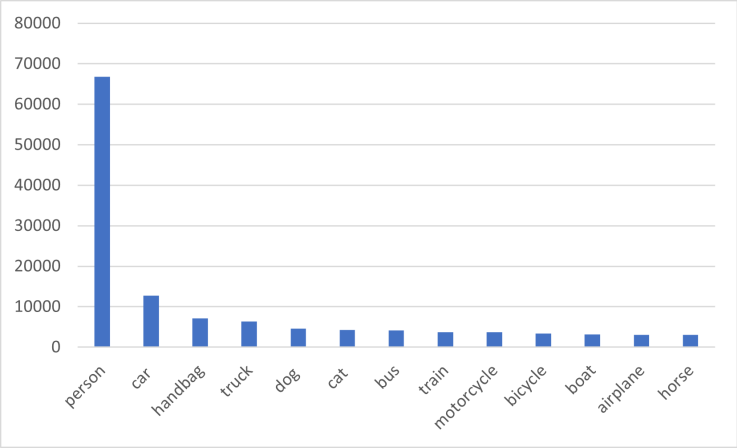


Figure 6.21 – Répartition des principales classes d'objets en nombre d'images

Le tableau 6.4 récapitule le contenu de notre base CD-COCO d'un point de vue distorsion en fonction de leur répartition.

Table 6.4 – Répartition des distorsions.

Type de distorsion	Nombre d'images	Rapport
Artefact de compression	17989	15.3%
Changement de contraste	18038	15.4%
Bruit gaussien	18055	15.4%
Flou de mouvement global	18018	15.3%
Flou de défocalisation global	17792	15.1%
Brouillard	787	0.7%
Pluie	845	0.7%
Rétroéclairage local	296	0.3%
Flou de défocalisation local	7061	6.0%
Flou de mouvement local	18625	15.9%

En parallèle, les scènes observées sont classées en scènes intérieures et extérieures en fonction du contexte. Les scènes d'intérieur sont attribuées aux images où la plupart des informations visuelles est incluse dans des environnements intérieurs (pièce, bâtiment, hall, intérieur de véhicule, etc.). À l'in-

Table 6.5 – Classification des scènes.

Type de scène	Nombre d'images
Scène d'intérieur	45884
Scène extérieure	72404
Scène de ski	4434
Scène de surf	3635
Scène de patinage	3603
Scène sportive	11965

verse, les scènes extérieures correspondent à des environnements ouverts. Il convient de noter que la classification des courts de tennis en scènes intérieures/extérieures est une tâche complexe, même pour les humains. En effet, les différences visuelles entre les courts de tennis intérieurs et extérieurs sont à peine perceptibles. Ainsi, nous avons arbitrairement affecté tous les courts de tennis à des scènes extérieures.

6.3.8 . Étude de la robustesse des modèles de détection d'objet vis a vis des distorsions

Dans un premier temps, nous avons étudié la robustesse de quelques modèles de détection d'objets de l'état de l'art dans les images perturbées pour mettre en avant la pertinence de la base de données unique que nous avons créée. Dans cette étude de la robustesse des algorithmes de détection d'objets, nous nous sommes focalisés sur les modèles YOLOv4 [BWL20], Mask R-CNN [He+17b] et EfficientDet [TPL20] ayant des architectures de réseau principal distinctes. Les performance de ces modèles sont alors comparées dans le cas de notre base de données et l'une des bases les plus connues MS-COCO. Le tableau 6.6 résume les différents modèles et leurs composants qui sont utilisés dans cette étude comparative.
Nous avons alors évalué la robustesse des modèles par rapport aux distorsions sur l'ensemble de validation MS-COCO 2017. Ce jeu de donnée de 5000 images a été perturbé selon les 10 types de distorsion et à 10 niveaux de sévérité de distorsion. Il est à noter que le niveau 0 de distorsion correspond à des images non dégradées. L'ensemble de ces modèles ont été évalués selon la métrique mAP (mean Average Precision), qui permet de calculer la précision de détection des modèles, à savoir la précision de la boîte englobante

Table 6.6 – Modèles de détection d'objet de l'état de l'art

Méthodes	Architecture	Taille	Bloques supplémentaires
Mask R-CNN [He+17b]	Resnet-101	512	FPN, RPN [Lin+17 ; Ren+15]
YOLOv4 [BWL20]	CSPDarknet53 YOLOv4-tiny	512 416	PAN [Liu+18]
EfficientDet [TPL20]	EfficientNet-B0 EfficientNet-B1 EfficientNet-B2 EfficientNet-B3 EfficientNet-B4	512 640 768 896 1024	Bi-FPN [TPL20]

FPN : Feature Pyramid Network, RPN : Region Proposal Network, PAN : Path Aggregation Network, Bi-FPN : Bi-directional FPN.

des objets détectés. Le mAP est la métrique qui permet de comparer la boîte englobante de la vérité de terrain à la boîte prédite et qui renvoie un score décrivant la précision de la détection. Le mAP est calculé en prenant en compte la précision moyenne notée AP pour chaque classe N du modèle, puis en calculant la moyenne de l'ensemble des classes comme ci-après :

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i = \frac{1}{N}\sum_{i=1}^{N}(R(i) - R(i+1)) \cdot P(i) \tag{6.9}$$

La précision P représente le pourcentage de prédiction correcte du modèle tandis le recall R indique le pourcentage d'objet de la vérité de terrain que le modèle est capable de prédire. La précision et le recall sont definis par :

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN} \tag{6.10}$$

Où TP correspond au nombre de boîtes englobantes correctement prédites chevauchant les boîtes de la vérité de terrain et ayant un score IoU (Intersection over Union) supérieur à un seuil fixé, FP le nombre de boîtes faussement prédites et ayant un score IoU inférieur au seuil et FN le nombre de boîtes faussement non prédites pour des boîtes de la vérité de terrain existantes.

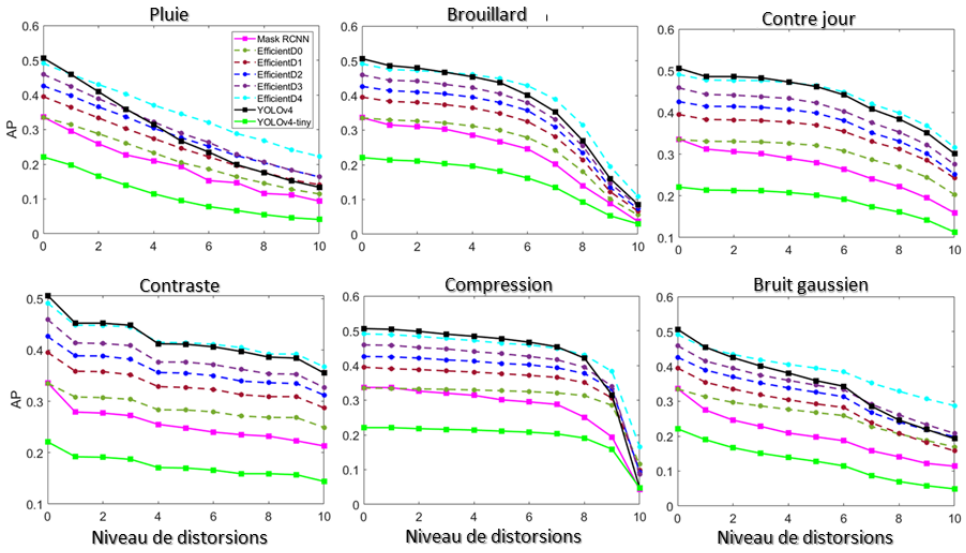


Figure 6.22 – Robustesse des méthodes de détection d'objets contre les distorsions

Les figures 6.22 et 6.23 illustrent l'évolution du score mAP des modèles de détection d'objet considérés en fonction de chaque type et niveau de distorsion. Ces résultats montrent que les distorsions réduisent les performances des méthodes de détection d'objets de 20% à 89,2%. De plus, ils mettent en évidence les faiblesses de la méthode YOLOv4 vis-à-vis des distorsions (courbes fortement décroissantes dans les figures 6.22 et 6.23), qui nécessitent une étude plus approfondie.

L'analyse des courbes de la figure 6.23 indiquent que les distorsions globales ont plus d'impact que les distorsions locales sur les performances des méthodes de détection d'objets. En effet, les performances globales des modèles sont fortement affectées par les images dégradées, mais pas de manière uniforme. À l'inverse, les perturbations locales et atmosphériques ont un impact plus ou moins cohérent sur les méthodes malgré leurs architectures différentes. Notons que chaque type de distorsion a un taux d'impact différent sur les performances des modèles en fonction du niveau de distorsion.

Afin de mieux évaluer l'impact des distorsions, nous avons calculé le taux de robustesse qui représente le rapport entre les scores mAP des modèles liés à chaque type et niveau de distorsion, et les images non dégradées (niveau 0). En recontextualisant ce constat par rapport à notre conception d'une méthode de SLAM dynamique, nous pouvons déduire que les scènes avec de forts flous de mouvement et de défocalisation globaux et/ou locaux impacteront fortement les performances du module de segmentation d'objet. Ainsi, l'utilisation d'une méthode d'augmentation de données par synthèse

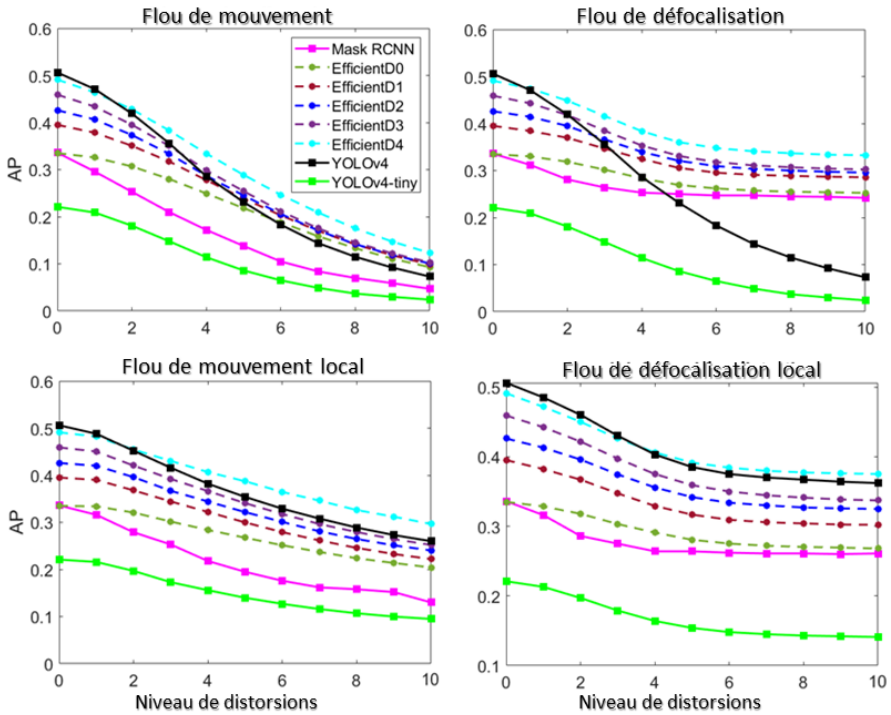


Figure 6.23 – Robustesse des méthodes de détection d'objets contre les distorsions de flou

de multiples distorsions photo-réaliste selon un schéma qui tient compte du contexte et de la profondeur de scène, permettra d'améliorer la robustesse de ce module.

6.3.9 . Augmentation de données

D'après notre étude, le modèle YOLOv4-tiny est le moins robuste aux distorsions avec une diminution moyenne avec un score AP de 36,7% entre les niveaux de distorsion 0 et 10 (taux de robustesse moyen le plus bas à 63,3%, voir score mAP dans les figures 6.22 et 6.23). Cette observation ainsi que la capacité de notre carte graphique nous a conduit à choisir ce modèle pour évaluer l'amélioration de la robustesse à l'aide d'une augmentation de données telle que proposé dans notre jeu de données. Cette section présente les résultats expérimentaux sur le jeu de donnée d'évaluation MS-COCO 2017 (5000 images). Le modèle YOLOv4-tiny a été entraîné selon le protocole du défi MS-COCO, c'est-à-dire en utilisant le jeu de donnée trainval35k qui comprend 118 000 images.
Nous avons d'abord effectué cet entraînement avec le jeu de donnée MS-COCO trainval35k original pour obtenir des valeurs de référence. Ensuite, nous l'avons effectué de nouveau avec le jeu de données dégradées, ce qui nous a permis d'évaluer la contribution de l'augmentation des données sur la ro-

Table 6.7 – Amélioration des performances en pourcentage du modèle YOLOv4-tiny entraîné sur notre jeu de données dégradées.

Type de distorsion	Niveau de distorsion					Amélioration moyenne
	2	4	6	8	10	
Bruit gaussien	9.58%	22.3%	37.9%	80.3%	114%	47.2%
Contraste	-0.54%	1.8%	2.47%	3.23%	5.71%	1.99%
Compression	-2.36%	-0.48%	-0.5%	4.84%	35.4%	4.26%
Pluie	12.7%	40.4%	71.4%	101.8%	114.3%	61.7%
Brouillard	-0.49%	3.14%	10.5%	29.8%	51.7%	15.8%
Flou de mouvement	6.11%	39.5%	95.3%	151.4%	183.3%	86.3%
Flou de défocalisation	1.02%	14.2%	22.4%	26.3%	26.7%	16.5%
Flou.mvmnt local	10.3%	31.6%	46.9%	59.3%	67.0%	39.8%
Flou.déf local	3.08%	16.0%	23.3%	25.5%	25.9%	17.2%
Contre-jour	1.45%	5.85%	19.1%	40.3%	76.8%	24.1%
Moyenne par niveau	4.11%	17.9%	32.9%	52.3%	70.1%	**31.5%**

bustesse du modèle. Chaque type de distorsion représente 5% de la base de données utilisée pour le processus d'apprentissage, soit 5,9 K images pour chaque distorsion. Dans notre cas, les hyper-paramètres sont issus de la méthode YOLOv4 mais avec quelques ajustements. Le pas d'apprentissage est de 500 050, avec le taux d'apprentissage initial multiplié par 0,1 à 400 040 pas puis 450 045 pas.

L'évaluation de l'amélioration de la formation est obtenue en comparant les scores d'évaluation des deux modèles formés sur les images originales et dégradées, respectivement, sur les ensembles de validation MS-COCO dégradés pour chaque niveau de distorsion. Ces résultats présentés dans le tableau 6.7 (Colonnes 2 à 6) sont obtenus en calculant le rapport entre les scores AP COCO du modèle dégradé sur le modèle original.

Nous avons observé que l'entraînement a un impact significatif sur l'amélioration de la robustesse avec une augmentation moyenne de 31,5% des performances (voir dernière colonne du tableau 6.7). De plus, l'étude a mis en évidence que l'entraînement améliore la robustesse pour les distorsions de haut niveau de sévérité (voir dernière ligne du tableau 6.7). Cette étude a révélé l'impact de la qualité des images acquises dans le développement des méthodes de détection. En effet, à travers cette étude, nous avons montré que l'augmentation de la base de données d'apprentissage avec des scénarios complexes contenant différentes distorsions améliore les performances des modèles.

Nous avons montré que l'augmentation de données pour le module de détection d'objet, à l'aide de notre base de données perturbée par des distorsions complexes et liées au contexte de scène (voir section 6.3) est une solution pertinente pour améliorer la robustesse des méthodes de détection d'objet. Il est à noter que l'application directe et logique aurait été l'étude de l'impact des distorsions sur les modèles de segmentation en vue de l'intégrer la méthode robuste dans notre méthode de VSLAM. Cependant, les études présentes dans l'état de l'art ne traitent que de l'impact des distorsions sur les performances des modèles de détection d'objet, ce qui a renforcé notre choix d'étude. De plus, l'absence au sein du laboratoire de matériel a la puissance de calcul nécessaire pour l'entraînement des modèles de segmentation d'objets empêchait d'effectuer les tests de cette étude. Pour contourner cette difficulté, il nous a semblé judicieux de baser notre étude sur les méthodes de détection d'objets qui au vu de leur architecture plus légère satisfaisaient aux conditions matérielles d'entraînement. Le fait que les modèles de détection et de segmentation d'objets aient une architecture très similaire a justifié notre démarche et permis de conjecturer l'apport de l'augmentation de donnée pour le modèle YOLACT++. Pour des raisons de limitation de capacité de calcul, nous avons dû réaliser l'entraînement du modèle YOLACT++ au sein d'un autre laboratoire. Le modèle de segmentation d'objet est détaillé dans la section suivante 6.4.

6.4 . Apprentissage profond : segmentation d'objet

6.4.1 . Principe

Les modèles de segmentation d'objets par les approches basées sur l'apprentissage profond ont pour objectif de fournir une représentation simplifiée de l'image où les pixels sont regroupés en classes au moyen d'étiquette ou labels fourni par le processus d'apprentissage. À l'inverse des modèles de détection d'objets, les modèles de segmentation fournissent une information précise de la localisation des objets permettant ainsi d'opérer des tâches complexes de vision par ordinateur. Les modèles actuels de segmentation d'objets reposent sur une architecture de réseau de neurones constituée de trois parties distinctes que nous nommerons :

- Réseau principal : réseau qui extrait la majorité des informations d'une image à travers des couches de convolution successives.
- Réseau d'extraction : les informations extraites par le réseau principal sont ensuite affinées lors d'un parcours inversé du réseau.
- En-tête du réseau : les informations affinées sont utilisées pour effectuer des prédictions réalisant la ou les tâche(s) du modèle (classification, détection, segmentation, etc.).

Les modèles de segmentation prédisent ensuite la localisation des objets segmentés en produisant des masques qui englobent la silhouette des objets au niveau pixel.

Nous décrivons brièvement ci-après le modèle YOLACT++, sur lequel repose notre schéma de VSLAM, un des modèles de segmentation les plus performants.

6.4.2 . Modèle YOLACT++

La méthode YOLACT++ [Bol+19] est le premier modèle de segmentation d'objet fonctionnant en temps réel avec de bonnes performances, ce qui a influencé notre choix pour l'intégrer à notre méthode de SLAM.

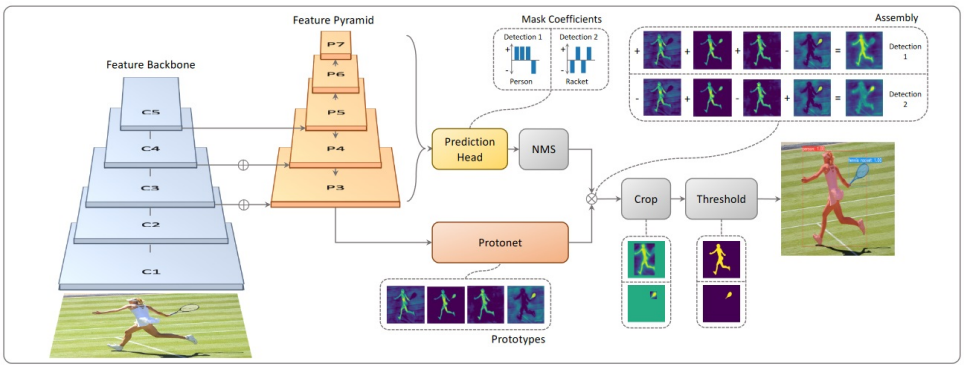


Figure 6.24 – Architecture du modèle YOLACT [Bol+19].

Cette prouesse est rendue possible par l'ajout d'une branche dédiée à la génération de masques à un modèle de détection d'objet existant. Ce générateur de masque opère à travers deux tâches exécutées en parallèle. Tout d'abord, un module qui produit des "masques prototypes" sans aucune considération pour les instances et classes des objets. Puis un second module supplémentaire ajouté à l'en-tête du réseau permettant d'obtenir un vecteur de "coefficients de masque" pour chaque détection d'objet afin d'encoder les "masques prototypes" selon un critère d'instance. Enfin, après application d'une suppression des non-maxima, les objets détectés en fin de processus servent d'identifiants pour combiner linéairement les résultats des masques et leurs coefficients respectifs.

L'architecture du modèle YOLACT est de la forme des modèles de détection d'objet à un niveau tel que les modèles YOLO [Red+16 ; RF18 ; BWL20] ou le modèle Mask-RCNN [He+17a] sans son réseau secondaire RPN (Region Proposal Network) de la méthode Faster-RCNN [Ren+15]. Le réseau principal intégré au modèle YOLACT se décompose en trois variantes, les réseaux Resnet 50 et 101 [He+16], et Darknet-53 [Wan+20]. Un réseau d'extraction est ajouté à la suite du réseau principal sous la forme d'un parcours inversé des cartes des

caractéristiques via la méthode FPN (Feature Pyramid Network) [Lin+17]. Ce réseau d'extraction permet de conserver la fiabilité des informations caractéristiques obtenues à l'aide du réseau principal et d'augmenter la précision de la localisation de ses informations en exploitant la précision des informations du début de réseau principal. La figure 6.24 illustre l'architecture du modèle YOLACT.

Le module "Protonet" en charge de la génération des "masques prototypes" prédit k masques prototypes pour l'ensemble de l'image. Ce module est basé sur un réseau FCN (Fully Convolutional Networks) [LSD15] qui prend en entrée la couche P3 du FPN en raison de sa plus grande précision d'informations caractéristiques et pixels (localisation). La sortie du FCN fournit une carte de k canaux correspondant aux k masques prototypes qui est suréchantillonnée et convoluée comme illustré dans la figure 6.25.

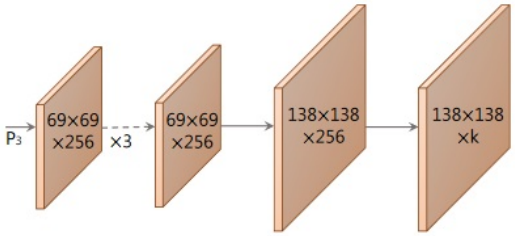


Figure 6.25 – Architecture du module Protonet [Bol+19].

Pour illustrer le fonctionnement du module *Protonet*, la figure 6.26 montre les masques prototypes de diverses images pour 6 masques prototypes.

Le second module de prédiction des "coefficients de masque" consiste en l'ajout d'une branche supplémentaire au réseau d'en-tête déjà composé d'une branche dédiée à la prédiction de la confiance des c classes d'objets et d'une autre branche pour la prédiction de 4 boîtes englobantes le tout pour chaque ancre de prédiction. Cette branche supplémentaire prédit alors k coefficients de masque correspondant à $(k+c+4)$ coefficients pour chacune des a ancres (voir figure 6.27). Pour éviter les problèmes de non-linéarité, une fonction d'activation $tanh$ est ajouté aux k coefficients de masques les rendant plus stables pour une future soustraction avec les k masques du module *Protonet*.

La combinaison linéaire des sorties des modules *Protonet* et *coefficient des masques* est réalisée par simple multiplication matricielle suivie d'une sigmoïde :

$$M = \psi(PC^T) \tag{6.11}$$

Ou P est la matrice des masques prototypes de taille $k \times w \times h$ et C à une

matrice des coefficients de masque de taille $k \times n$ relative à n instance après suppression des non-maximums et seuillage des scores.

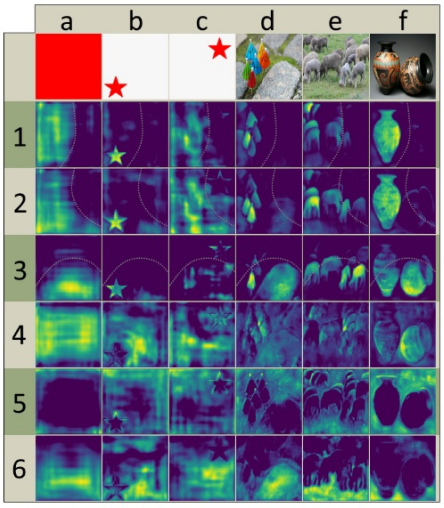


Figure 6.26 – Comportement du module Protonet [Bol+19].

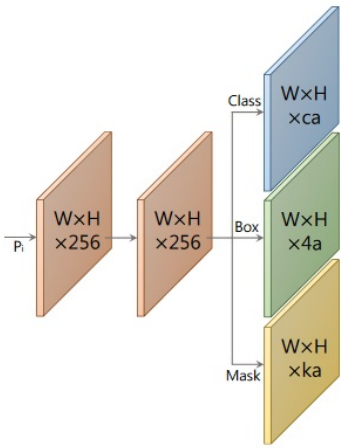


Figure 6.27 – Architecture de l'en-tête du modèle YOLACT [Bol+19].

Les fonctions de perte associées à chacune des branches de l'en-tête, une perte de régression de boîte englobante et une perte de masque de poids respectifs 1, 1.5 et 6.125. Notons que les fonctions de perte de classification et régression sont extraites du modèle SSD (Single Shot Detection) [Liu+16].

La fonction de masque correspond à l'entropie croisée binaire pixel par pixel entre les masques prédits et les masques de vérité terrain. Enfin, les masques finaux sont recadrés à l'aide des boîtes englobantes prédites lors de l'évaluation et à l'aide de ceux de la vérité de terrain lors de l'entraînement.

6.4.3 . Évaluation de l'augmentation de données

Afin d'évaluer l'apport de l'augmentation de données sur la robustesse du modèle YOLACT dans le cas d'images perturbées, nous avons entraîné un des modèles YOLACT avec notre base de données CD-COCO créée à cet effet. Cette base décrite précédemment, contient 95K images pour l'entraînement et 4926 images pour la validation. L'entraînement a été effectué sur un ordinateur possédant un processeur inter-core i9 et une carte graphique GPU NVIDIA RTX A6000. Afin d'avoir une base comparative, nous avons entraîné le modèle YOLACT basé sur l'architecture Resnet101-FPN de taille d'entrée 550 avec notre base CD-COCO et la base MS-COCO originale en conservant le protocole original de la méthode à savoir :

- Un entraînement de 800 000 itérations avec des changements du taux d'apprentissage aux itérations 280 000, 600 000, 700 000, 750 000.
- Un taux d'apprentissage de 0.001 et un facteur de 0.1 pour chacun des changements de taux.
- Un *momentum* et *decay* posés respectivement à 0.9 et 0.0005.

Les tests ont été effectués sur le jeu de test de la base CD-COCO pour le modèle original et notre modèle entraîné. Le tableau 6.8 reprend les résultats de ces deux modèles sur la base de données CD-COCO. Entraîner le modèle YOLACT avec notre base de données CD-COCO améliore de 17.7% les performances du modèle.

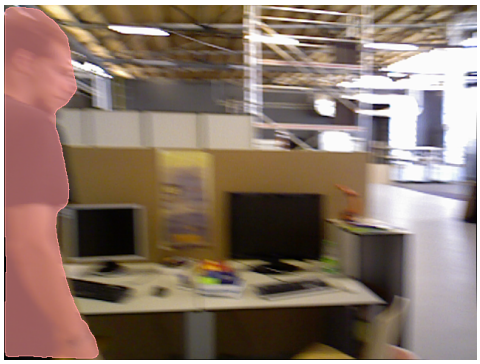
Table 6.8 – Performance du modèle YOLACT sur la base CD-COCO

Modèles	Architecture	Taille	mAP
YOLACT [Bol+19]	Resnet101-FPN	550	0.243
YOLACT + CD-COCO	Resnet101-FPN	550	0.286

Cependant, il est important de noter qu'en raison de contraintes matérielles, seul le modèle YOLACT Resnet101-FPN de taille d'entrée 550 et de précision 29.8 mAP a été entraîné. En effet, le modèle YOLACT++ Resnet101-FPN de taille 550 et précision 34.6 mAP requérait une configuration ordinateur indisponible en laboratoire. Le modèle YOLACT++ est une optimisation du modèle YOLACT existant et atteint donc de meilleures performances. Au vu des travaux existants [Mic+19 ; BMB22a] sur l'amélioration de la robustesse des modèles de détection et segmentation d'objet, nous pouvons aisément conjecturer de l'impact positif de l'utilisation de la base CD-COCO pour l'amélioration de la robustesse du modèle YOLACT++ contre les distorsions.

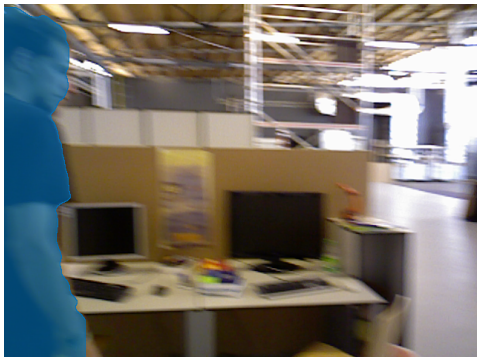
La figure 6.28 reprend l'illustration des limitations du module de segmentation appliqué sur la base d'évaluation du SLAM. On observe une amélioration nette des performances sur ces images perturbées pour le modèle YOLACT entraîné avec la base CD-COCO par rapport au modèle YOLACT++ qui est pourtant, de base, plus performant.

(a) Inefficacité de la segmentation avec YOLACT++



(b) Imprécision de la segmentation avec YOLACT++

(c) Efficacité de la segmentation avec YOLACT CD-COCO



(d) Précision équivalente avec YOLACT CD-COCO

Figure 6.28 – Illustration de l'amélioration de la robustesse du modèle de segmentation

Le modèle YOLACT entraîné sur la base CD-COCO est plus performant pour la segmentation de scène présentant des distorsions comme illustré dans la figure 6.28c pour une distorsion locale de flou de mouvement et dans la figure 6.28c pour une distorsion globale de flou de mouvement.

6.5 . Expérimentations et résultats pour le SLAM

L'apport de l'utilisation de modèle YOLACT entraîné avec notre base CD-COCO sur les performances de notre méthode de SLAM dynamique, est évalué sur la base TUM RGBD [Stu+12] avec le modèle entraîné sur la base originale MS-COCO puis avec le modèle entraîné sur notre base CD-COCO. Ces

expérimentations ont été faites sur les 5 séquences qui composent la base TUM-RGBD pour évaluer la méthode selon les métriques de ATE (*Absolute Trajectory Error*), RPE (*Relative Pose Error*) de rotation et RPE de translation.

Il est important de noter que le modèle YOLACT étant moins optimisé que le modèle YOLACT++, la précision de la segmentation sera inférieure. Notre méthode de SLAM requiert une segmentation précise pour effectuer une estimation sémantique de l'état des points qui soit précise (module segmentation sémantique). Le module d'attention de la profondeur nécessite une bonne précision de segmentation, ou au mois et surtout un résultat de segmentation qui indique la présence d'une personne dans la scène pour pouvoir ensuite estimer l'interaction inter-objet dans la scène observée. Toujours au vu des travaux sur l'amélioration de la robustesse des modèles de détection et segmentation d'objet existants, nous pouvons conclure quant à l'impact positif de l'utilisation de la base CD-COCO pour l'amélioration de la robustesse du modèle YOLACT++ vis à vis des distorsions du signal image.

6.5.1 . Impact de l'augmentation de données

L'évaluation de notre méthode de SLAM dynamique avec l'utilisation du modèle YOLACT simple et notre modèle YOLACT CD-COCO est résumée dans le tableau de la figure 6.29.

Figure 6.29 – Évaluation de l'impact de l'augmentation de données sur notre méthode de SLAM dynamique

Séquences	YOLACT			YOLACT CD-COCO			Amélioration		
	ATE	T. RPE	R. RPE	ATE	T. RPE	R. RPE	ATE	T. RPE	R. RPE
fr3/w/xyz	0.0164	0.0238	0.644	**0.0151**	**0.0218**	**0.627**	8.6%	9.2%	2.7%
fr3/w/half	0.0278	0.0394	0.898	**0.0266**	**0.0378**	**0.8611**	4.5%	4.2%	4.3%
fr3/w/static	0.0075	0.0108	0.298	**0.0072**	**0.0106**	**0.289**	4.2%	1.9%	3.1%
fr3/w/rpy	0.0497	0.0698	1.469	**0.0339**	**0.0496**	**1.052**	46.6%	40.7%	39.6%
fr3/s/static	0.0067	0.0101	0.329	**0.0063**	**0.0095**	**0.319**	6.3%	6.3%	3.1%

Les performances de la méthode sont accrues de 5.9% pour l'ATE, 5.4% pour le RPE de translation et 3.3% pour le RPE de rotation. Notons cependant que l'amélioration pour la séquence rpy ne peut être pris en considération du fait de valeur trop importante. En effet, l'écart entre les performances de notre méthode utilisant YOLACT et YOLACT++ est considérable, cela est certainement dû à la nature même de la séquence rpy où de nombreuses distorsions de flou de mouvement sont présentes. Le modèle YOLACT étant de nature moins performant que le modèle YOLACT++, il est de fait plus sensible aux perturbations. En faisant une projection de cette amélioration sur notre mé-

thode de SLAM utilisant le modèle YOLACT++, nous atteindrions les performances suivantes :

Figure 6.30 – Projection des performances de notre méthode de SLAM dynamique basé le modèle YOLACT++ entraîné avec CD-COCO

Séquences	YOLACT++			YOLACT++ CD-COCO			Amélioration		
	ATE	T. RPE	R. RPE	ATE	T. RPE	R. RPE	ATE	T. RPE	R. RPE
fr3/w/xyz	0.0149	0.0217	0.6194	0.0136	0.0197	0.6026	98.1%	98.2%	97.0%
fr3/w/half	0.0262	0.0369	0.8608	0.0250	0.0353	0.8237	94.6%	94.2%	94.3%
fr3/w/static	0.0068	0.0097	0.2789	0.0059	0.0095	0.2702	98.3%	98.1%	96.9%
fr3/s/static	0.0064	0.0093	0.3222	0.0060	0.0087	0.3122	37.5%	43.9%	21.2%

Notons qu'aucune projection n'est faite pour la séquence rpy en raison du problème de performance évoqué précédemment. Nous pouvons cependant conjecturer que l'utilisation du modèle YOLACT++ entraîné avec CD-COCO améliore les performances de notre méthode de SLAM pour la séquence rpy. La colonne "Amélioration" du tableau 6.30 représente le taux d'amélioration de la méthode de SLAM par rapport à la méthode originale ORB-SLAM3 [Cam+20].

6.6 . Conclusion

À travers cette étude, nous avons montré que la prise en compte de divers scénarios, correspondant a des conditions d'acquisition d'images dans des environnements difficiles et incontrôlables, est une approche pertinente pour la construction de bases de données, nécessaires au développement d'architectures basées sur l'apprentissage profond pour la résolution de problèmes lies à notre méthode de SLAM. En effet, les tests préliminaires menés sur notre méthode de SLAM dynamique ont mis en évidence une défaillance partielle ou complète du module de segmentation composé du modèle YOLACT lors du traitement d'images contenant du flou de mouvement local et global.

Ces scénarios sont relativement présents dans la base de test TUM RGBD et dans le cadre d'applications réelles lors de rotation brusque de la caméra ou de mouvement d'objet proche de la caméra. Ainsi, l'incorporation d'une étape d'entraînement du modèle YOLACT intégré dans notre méthode de SLAM dynamique avec notre base d'image perturbées CD-COCO permet d'augmenter la robustesse du module de segmentation. En effet, une augmentation de la robustesse de ce module implique une meilleure capacité à segmenter les objets flous permettant ainsi de mieux considérer l'ensemble des personnes présentes dans la scène par notre module d'attention de profondeur

en charge d'évaluer l'interaction inter-objet. De fait, une précision de segmentation équivalente et un plus grand nombre de segmentation efficace entraînent une meilleure estimation de l'état d'objet en interaction avec les personnes présentes dans la scène. Cette meilleure estimation entraînant à son tour de meilleures performances de notre méthode de SLAM dynamique comme le montrent les tableaux 6.29 et 6.30.

7 - Perception visuo-inertielle pour l'évaluation et l'optimisation d'un simulateur immersif

7.1 . Introduction

Tout d'abord, il est important de rappeler que nos travaux ont deux principaux objectifs à atteindre, à savoir, la conception d'un algorithme de perception du mouvement propre imitant les fonctions cognitives visuo-spatiales de l'humain et la conception d'une méthode d'optimisation de l'immersion basée sur l'algorithme de repérage de mouvement (*Motion Cueing Algorithme* MCA). Également, rappelons que les informations liées au mouvement sont soit de nature visuelle à travers l'utilisation d'un algorithme de vision par ordinateur, soit de nature kinesthésique à travers la récupération de données des capteurs articulaires et inertiels provenant du robot NAO et de la centrale inertielle Xsens. Nous tentons dans ce chapitre de formaliser notre système dans une architecture globale qui souligne les liens et enchaînements des différents éléments qui le composent (voir la figure 7.1).

Nous pouvons ainsi distinguer quatre modules inter-connectés permettant d'atteindre ces objectifs. Chacun de ces modules est décrit séparément dans les sections suivantes et tente de réaliser les tâches suivantes :

- Module NAO : acquiert les informations provenant des capteurs articulaires et inertiels de manière synchronisée. Il formalise et interprète ces informations selon les modélisations du robot.
- Module Vision : incorpore le processus d'acquisition des informations visuelles et leur interprétation à travers une méthode de SLAM visuel robuste dans le but de réaliser une localisation du robot dans son environnement réel.
- Module Perception : réalise l'association des informations provenant des modules NAO et Vision selon une retranscription du procédé et des capacités de perception de l'humain.
- Module Optimisation d'immersion : détermine les paramètres du MCA pour optimiser les ressentis du mouvements propre à l'aide d'une décomposition du profil d'accélération d'entrée du simulateur en des profils pour la table XY, l'hexapode et une rotation de tilt.

Les modules NAO et Vision sont réalisés en parallèle afin de fournir en même temps au module perception leurs informations qui sont complémentaires et utiles pour estimer l'impact du mouvement de la plateforme sur le robot. Cette estimation du mouvement perçu sert alors d'entrée l'outil d'évaluation

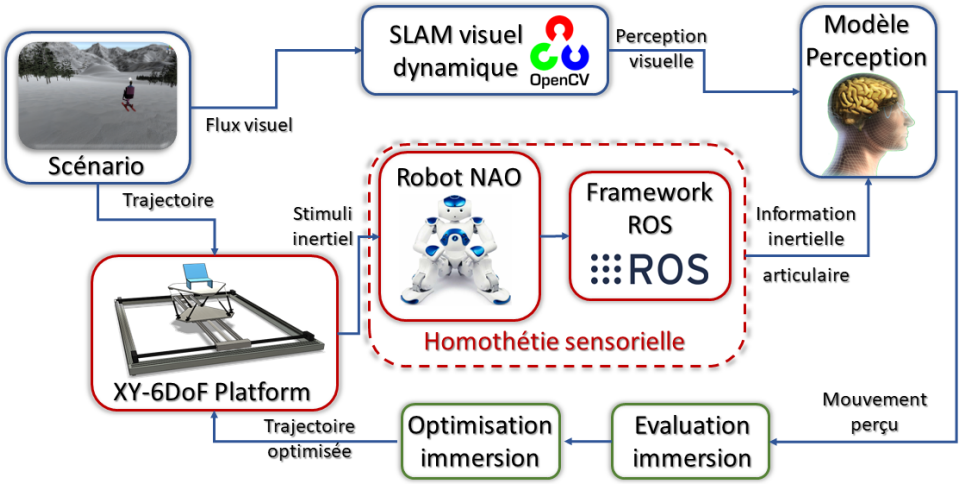


Figure 7.1 – Architecture globale de notre système.

de l'immersion dans le but de définir des profils d'accélération augmentant les sensations. L'ensemble de ces modules a été implémenté sous ROS (Robot Operating System), qui est un système pour le développement d'application robotique, afin de satisfaire le besoin de synchronisation des différentes informations.

L'ensemble des caractéristiques et de nos réalisations portant sur le robot NAO sont détaillées par la suite. Notre méthode de SLAM visuel dynamique développé est basé sur la méthode ORB-SLAM3 [Cam+20] à laquelle nous avons intégré des processus de segmentation et/ou détection d'objet associé à des raisonnements géométriques exploitant également les informations de profondeur afin de la rendre plus robuste aux dynamiques de scène.

La substitution de l'humain par le robot humanoïde NAO permet l'étude des interactions multi-sensorielles de notre système robotique liée à la perception du mouvement. Le système robotique de substitution et les algorithmes de vision et perception exploitent au mieux les informations visuelles et inertielles pour reproduire les capacités cognitives humainse. Notre système robotique est destiné à la réhabilitation de personne présentant un handicap moteur, de ce fait, le robot reproduira les contraintes corporelles liées au handicap. Les parties inférieures seront donc bloquées physiquement avec des sangles et un blocage logiciel des actionneurs de chacune des articulations affectées. Ne seront pas concernés par la suite de l'étude uniquement le bassin et la partie supérieure du corps pour l'étude de la perception du mouvement.

7.2 . État de l'art

La perception du mouvement propre de l'humain correspond à la perception de son mouvement corporel dans son environnement à travers des retours de sensations provenant des organes sensoriels et proprioceptifs. Dans notre cas d'étude, la substitution de l'humain par le robot NAO nous a conduit à chercher la manière dont l'humain perçoit le mouvement. Notre système robotique nous offrait l'accès aux informations visuelles via les caméras, inertiel via les centrales inertielles et proprioceptives à travers les capteurs articulaires. Il a été ainsi nécessaire d'étudier les capacités humaines de perception à travers ses limites et les possibles modèles ou algorithmes existants. L'étude de ce processus a débuté au milieu du 20^{eme} siècle avec comme premier objectif de déterminer quelles étaient les seuils de perception du mouvement de l'humain à travers le système vestibulaire. Par la suite, des travaux expérimentaux sur des plateformes robotiques dédiées à la simulation de mouvement ont permis de modéliser le processus de perception du mouvement d'après le système vestibulaire. Ces travaux ont ensuite été enrichis par l'intégration des informations visuelles dans le modèle afin de reproduire au mieux la manière dont l'humain perçoit son mouvement. Nous détaillons dans les sections suivantes les principaux travaux et expérimentations menées à cet effet.

7.2.1 . Système vestibulaire

Le système vestibulaire est l'organe sensoriel responsable de la perception des mouvements d'un point de vue inertiel et de l'équilibre chez l'humain. Le système vestibulaire est le composant principal de l'oreille interne (voir figure 7.2) et est relié au système nerveux par le nerf vestibulaire.

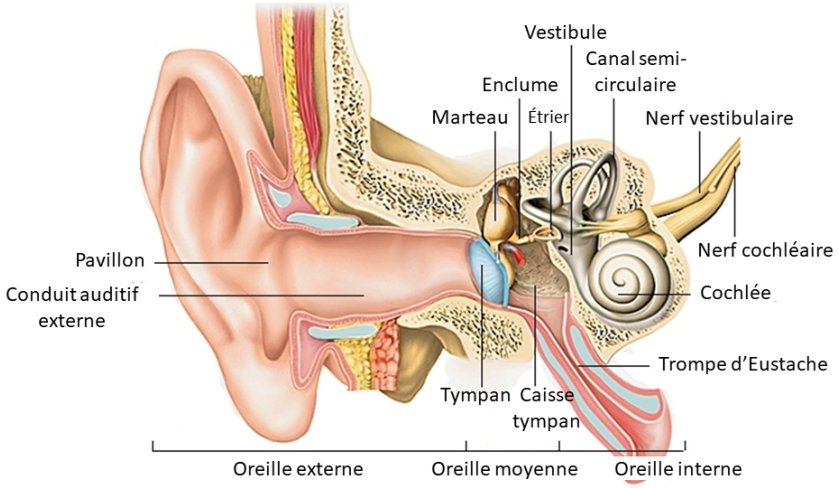


Figure 7.2 – Oreille interne humaine.

Il est localisé dans le labyrinthe de l'oreille interne, cavité permettant l'équilibre et le ressenti de sensation de mouvement inertiel. Ce système intègre deux éléments principaux responsables de la perception inertielle : les canaux semi-circulaires et le vestibule via les otolithes. La perception des mou-

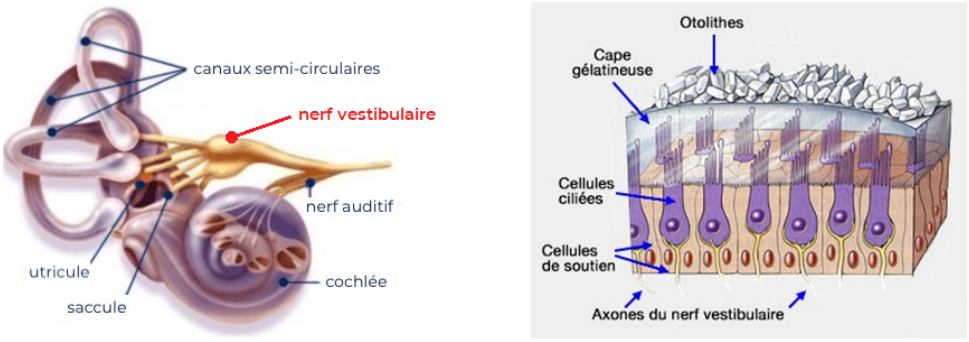


Figure 7.3 – Vestibule composé de macules.

vements linéaires est rendue possible par les otolithes présents dans le vestibule du système vestibulaire (voir figure 7.3). Le vestibule, organe otolithique, est composé de macules constituées de cellules de soutien raccordées aux axones du nerf vestibulaire. Il existe deux macules dans le vestibule, les cavités utriculaires et sacculaires, chacune responsable de la perception de mouvement selon un axe donné. Les otolithes présents dans les macules sont des cristaux de calcium déposés sur une couche membranique gélatineuse appelée membrane otoconiale (ou membrane otolithique) dans laquelle flotte une multitude de cellules sensorielles dites cellules ciliées. Ces cellules ciliées, également appelées épithélium cilié, sont raccordées aux cellules de soutien pour transmettre la perception du mouvement au système nerveux humain.

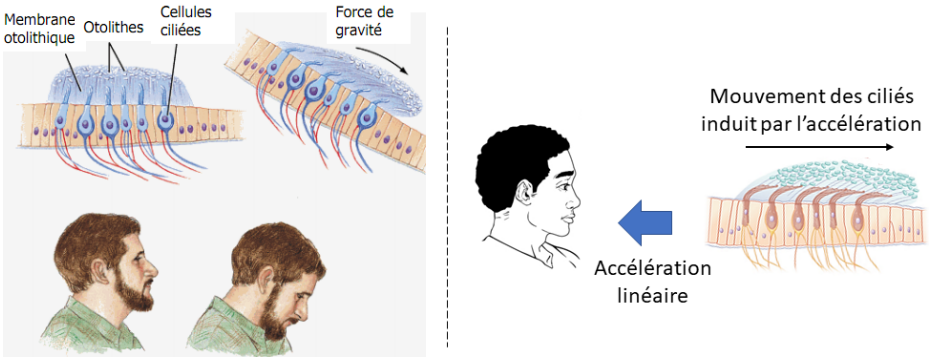


Figure 7.4 – Impact des accélérations sur les ciliés.

Les ciliés présents dans les macules sont sensibles aux accélérations linéaires, dont la gravité terrestre. En cas de mouvement, les ciliés se courbent dans le

sens inverse du mouvement en flottant dans la membrane otolithique. Les ciliés de l'utricule détectent les accélérations antéropostérieures et latérales à la tête tandis que ceux du saccule décèlent les accélérations verticales. La longueur des ciliés n'étant pas identique, ils sont capable de distinguer les accélérations à haute fréquence et amplitude (mouvement réel) de ceux à basse fréquence (gravité ou mouvement léger). Ce mouvement de ciliés est alors recueilli par les cellules de support qui convertissent cette information mécanique en signal électrique nerveux qui est ensuite interprété et transformé en sensation comme expliqué dans les travaux de Benson [Ben+74] et Telban [HTC05]. Le phénomène de déplacement des ciliées en cas de mouvement de la tête est illustré dans la figure droite 7.4 tandis que l'impact de la gravité est illustré dans la figure gauche 7.4.

De nombreux travaux ont tenté de modéliser mathématiquement ce processus de perception des mouvements linéaires [Mei65 ; YM68 ; BD82 ; TCG00]. De manière générale, la dynamique du système otolithique est considérée comme étant un système d'absorption de choc sous la forme d'un système masse-ressort. Une première modélisation de ce système a été proposée par Meiry [Mei65] sous la forme du filtre du second ordre suivant :

$$\frac{\hat{v}(s)}{v(s)} = \frac{K_{OTO}\tau_1 s}{(1+\tau_1 s)(1+\tau_2 s)} \tag{7.1}$$

Avec $\hat{v}(s)$ et $v(s)$ sont respectivement les vitesses linéaires de stimuli et ressentis. Les constantes τ_1 et τ_2 fixées respectivement à 10 et 0.66 secondes, et K_{OTO} un gain fixé en fonction des amplitudes mesurées. Cependant, cette modélisation ne considère pas les forces spécifiques provoquant une mauvaise prédiction de la réponse du modèle otolithique pour les accélérations soutenues. Un modèle revisité est proposé par Young [YM68] qui ne se base plus sur la vitesse, mais sur les forces appliquées au système. Dans ce cas, la force spécifique f correspond à la différence entre l'accélération linéaire a issue du mouvement et l'accélération constante g induite issue de la gravité terrestre, soit $f = a - g$. Cette approche permet de considérer l'impact continu de la gravité (accélération verticale) sur le système otolithique même en cas d'absence de mouvement de l'individu. Le modèle revisité s'écrit alors :

$$\frac{\hat{f}(s)}{f(s)} = \frac{K_{OTO}\tau_L s + 1}{(1+\tau_1 s)(1+\tau_2 s)} \tag{7.2}$$

Où f et $\hat{f}$ sont respectivement la force spécifique effective et celle ressentie. Les constantes τ_1, τ_2 et τ_L sont fixées différemment dans les travaux suivants [YM68 ; BD82 ; TCG00] tandis que le gain K_{OTO} est déterminé à l'aide après mesure.

Les canaux semi-circulaires sont les composants du système vestibulaire permettant la détection des rotations s'opérant sur l'axe crânien du corps. Les canaux sont localisés dans la partie haute de l'oreille interne (voir figure 7.5), directement reliés à une ampoule appelée "cupule" remplie d'une substance gélatineuse dans laquelle flottent des cellules sensorielles. Les canaux sont remplis d'un liquide appelé l'endolymphe qui bouge en inertie avec les mouvements de la tête provoquant une pression sur la substance gélatineuse présente dans les ampoules (cupule) entraînant à son tour le mouvement de cellules ciliées. De même que le système otolithique, le mouvement méca-

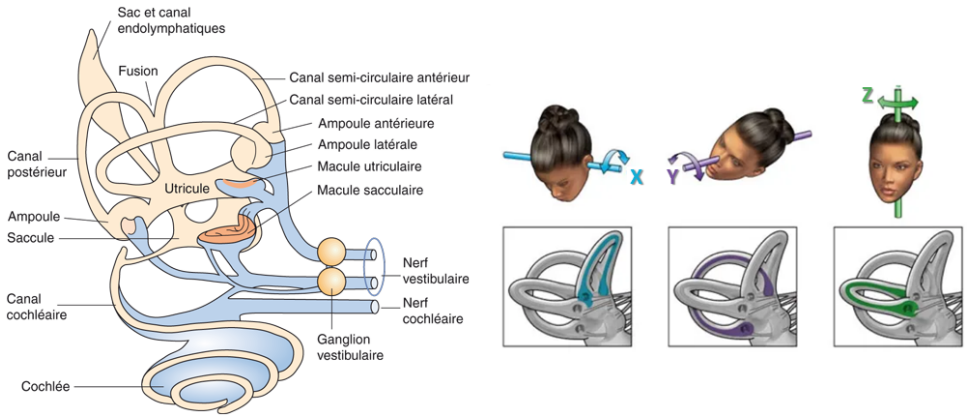


Figure 7.5 – Canaux semi-circulaires de l'oreille interne.

nique des cellules ciliées est convertis en message électrique nerveux transmis au système nerveux à travers le nerf vestibulaire. Ce message est ensuite traité et analysé pour en déduire une sensation de mouvement angulaire. Les canaux du système vestibulaire sont au nombre de trois canaux, disposés de manière plus ou moins orthogonale, comme illustré dans la figure 7.6, offrant ainsi une perception à 3 degrés de liberté.

Chacun des canaux semi-circulaires illustrés dans la figure 7.5 permet la perception d'un mouvement angulaire distinct comme indiqué ci-après :

- Le canal semi-circulaires antérieur (vertical) : relié à l'ampoule antérieure, permet de détecter les mouvements de rotation de tangage (rotation x sur la figure 7.5).
- Le canal semi-circulaires latéral (horizontal) : relié à l'ampoule latérale, permet de détecter les mouvements de rotation de lacet (rotation z sur la figure 7.5).
- Le canal semi-circulaires postérieur (vertical) : relié à l'ampoule, permet de détecter les mouvements de rotation de roulis (rotation y sur la figure 7.5).

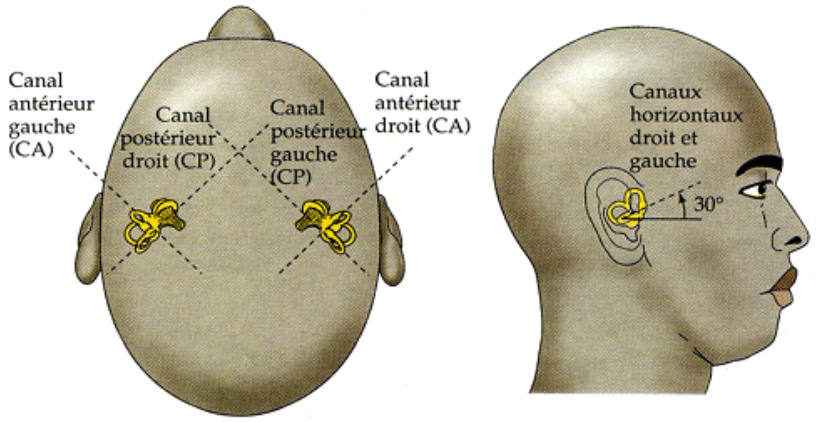


Figure 7.6 – Orientation des canaux semi-circulaires.

Benson et Mayne [Ben+74] ont proposé un premier modèle mathématique de la réponse des canaux semi-circulaires à des accélérations angulaires en formulant le phénomène mécanique comme un système de pendule sur-amorti. La fonction de transfert de second ordre de ce système a été formulée de la manière suivante :

$$\frac{\theta_e(s)}{\alpha(s)} = \frac{\tau_1\tau_2}{(1+\tau_1 s)(1+\tau_2 s)} \tag{7.3}$$

Où $\alpha(s)$ et $\theta_e(s)$ sont respectivement l'accélération angulaire subit par la tête et le déplacement du liquide endolymphe des canaux dans leurs ampoules respectives. Rappelons que le déplacement de ce liquide entraîne le mouvement des cellules ciliées contenues dans les ampoules qui permet de déduire le mouvement de la tête. Les constantes de temps τ_1 et τ_2 sont liées aux caractéristiques physionomiques de l'élasticité de la tasse et du moment d'inertie du liquide endolymphe. Plusieurs variantes de ce modèle existent [Mei65 ; YM68] avec des constantes de temps et de gain définies, mais également sans considération pour la sensibilité des canaux à certaines plages d'accélération.

Hosman [HV78] propose un modèle obtenu à l'aide de mesures expérimentales effectuées sur des pilotes devant déterminer l'accélération angulaire seuil à partir de laquelle ils percevaient le mouvement. Les pilotes étaient installés sur une plateforme robotique effectuant des mouvements de rotation connus, permettant de déduire la fonction de transfert des canaux semi-circulaires. Le modèle proposé est décrit par la fonction H_{SCC} suivante :

$$H_{SCC}(s) = \frac{\tau_L s + 1}{(1+\tau_1 s)(1+\tau_2 s)} \tag{7.4}$$

Où les constantes τ_1, τ_2 et τ_L sont fixées respectivement à 5.924, 0.005 et 0.1097.

Par la suite, d'après la supposition faite par Zacharias [Zac78], la vitesse angulaire $\hat{\omega}$ perçue par l'humain à travers les canaux semi-circulaires est proportionnelle au mouvement de la membrane gélatineuse de la cupule. Une fonction de transfert correspondante est décrite ci-après :

$$\frac{\hat{\omega}(s)}{\omega(s)} = \frac{\tau_1 s}{(1+\tau_1 s)(1+\tau_2 s)} \tag{7.5}$$

Houck et Telban [TC01; HTC05] prennent en compte dans leur étude la sensibilité des canaux aux accélérations sous forme d'intervalle. En raisonnant sous forme de vitesse angulaire, ils ont relevé que la sensibilité était accrue pour les vitesses angulaires comprises entre $11.45 deg/s^{-1}$ et $572.95 deg/s^{-1}$. De même, la sensibilité aux accélérations angulaires est équivalente, mais pour une bande passante basse fréquence comprise entre $0,26 deg/s^{-2}$ et $79,64 deg/s^{-2}$. Ainsi la perception du jerk correspondant à la dérivée de l'accélération, les accélérations au-delà de $103,7 deg/s^{-3}$ sont perçues. Pour satisfaire ces conditions, le modèle en vitesse des canaux semi-circulaires s'exprime de la manière suivante :

$$\frac{\hat{\omega}(s)}{\omega(s)} = \frac{K_{SCC}\tau_1\tau_a s^2(1+\tau_L s)}{(1+\tau_a s)(1+\tau_1 s)(1+\tau_2 s)} \tag{7.6}$$

Comme proposé dans le modèle de Telban [TC01], la fonction de transfert précédente peut être d'ordre réduit du fait que les valeurs des constantes τ_2 et τ_L sont au-delà de l'intervalle des mouvements possibles de la tête. De même, la valeur du pas doit être très inférieur à celle de la plus petite constante de temps. La fonction de transfert peut alors se réécrire :

$$\frac{\hat{\omega}(s)}{\omega(s)} = \frac{\tau_1\tau_a s^2}{(1+\tau_a s)(1+\tau_1 s)} \tag{7.7}$$

7.2.2 . Seuils de perception inertielle

Les seuils de perception des mouvements linéaire et angulaire ont été déterminés par de nombreuses expérimentations faites sur des individus suivant des protocoles définis et rigoureux. Chacun de ces tests a été effectué sur des cobayes ayant les yeux bandés afin de n'estimer que la perception selon les informations inertielles provenant du système vestibulaire humain. Dans ces études, le seuil de perception correspond à l'accélération ou vitesse angulaire minimale à partir de laquelle le mouvement est perçu par l'humain. Dans le cas d'accélérations non constantes, ce seuil peut également correspondre à la différence d'accélération perceptible.

J.J.Groen et L.B.W.Jongkees ont mené une étude [GJ48] en 1948 pour déterminer le seuil de perception angulaire de l'humain à travers diverses expériences. Ils ont ainsi pu critiquer ou conforter les seuils préalablement trouvés

lors d'expérimentations antérieures. Leur première expérience a été menée sur 30 cobayes humains les yeux bandés et placés sur une chaise effectuant une rotation de lacet. Les individus étaient soumis à un mouvement de rotation relativement long (30 secondes) et devaient indiquer le moment où une sensation de mouvement était ressentie. Ce relevée permettant ainsi de déterminer la valeur minimale à partir de laquelle le mouvement était perçu par le système vestibulaire. Cette expérimentation a établi une valeur de seuil pour une accélération angulaire de 0.8°/s² en moyenne pour l'ensemble des 30 sujets, qui allait de 0.28°/s² à 2.0°/s². La seconde expérimentation consistait à une étude de la perception angulaire dans le cas d'un mouvement oscillatoire issue d'une torsion chez les cobayes précédents. Une diminution moyenne de 30% a été relevé au niveau du seuil de perception pour ce type de rotation, avec une valeur de 0.55°/s².

B.Clark et J.D.Stewart ont effectué une étude comparative [CS68] en 1968 afin d'émettre une critique objective des seuils établis précédemment en raison des conditions expérimentales dans lesquelles elles ont été réalisées . En effet, la technologie utilisée dans les précédentes études ne permettait pas d'avoir des résultats suffisamment fiables. De plus, le manque de variété dans le profil des accélérations empêche la généralisation des seuils préalablement déterminés. Dans cette étude, trois expérimentations ont été réalisées sur 16 hommes avec les yeux fermés pour déterminer l'impact du type d'accélération sur la perception du mouvement chez l'humain. L'ensemble de ces expérimentations ont été menées avec des individus sains et à l'aide d'une chaise de rotation. Tout d'abord, une expérimentation a été effectuée pour des accélérations constantes où l'individu devait indiquer le sens de la rotation de la chaise toutes les 10 secondes afin de relever la présence de sensation. Cette expérience a permis d'établir un seuil à 0.39°/s² en moyenne pour une plage de valeur allant de 0.11°/s² à 1.18°/s².

La seconde expérience reprenait le protocole précédent, mais pour des intervalles d'accélération. Si le candidat répondait correctement aux stimuli des valeurs minimale et maximale de deux intervalles d'accélération, alors l'écart entre ces intervalles était réduit pour affiner la valeur de seuil. Ce procédé était reproduit jusqu'à ce que les intervalles se chevauchent. Par ce procédé, la valeur de seuil a été estimé à 0.43°/s² en moyenne avec un intervalle de réponse allant de 0.07°/s² à 1.19°/s².

La dernière expérimentation consistait en l'application de rampes d'accélération en constante augmentation pour une durée de 20 secondes suivies d'un palier à accélération d'une durée de 60 secondes. Il existait quatre profils de rampes : 0.003°/s^3, 0.006°/s^3, 0.010°/s^3 et 0.020°/s^3 où l'individu devait

détecter quel était le sens de la rotation de la chaise. L'expérimentation a permis de relever les seuils 0.20°/s², 0.33°/s², 0.44°/s² et 0.63°/s² suivant pour les profils d'accélération respectifs.

Alain Berthoz et Jaques Droulez [BD82] ont étudié en 1982 la perception du mouvement propre linéaire. Au cours de leurs travaux, ils ont synthétisé les travaux existant à ce sujet en proposant un tableau récapitulatif des seuils pour diverses configurations et expérimentations. Nous présentons les seuils de perception dans ce tableau 7.1 synthétique, pour des fréquences de fonctionnement et des périodes différentes. Le mouvement décrit dans le tableau représente successivement l'orientation de la gravité terrestre et des oscillations selon le sujet de l'expérience, puis l'orientation des oscillations selon la gravité. Notons que les repères sont définis comme étant X (en avant), Y (sur le côté) et Z (en haut). Ce tableau permet de mettre en avant l'incertitude des

Table 7.1 – Tableau récapitulatif de la perception du mouvement propre linéaire.

Auteurs	Année	Nombre de sujet	Posture	Mouvement	Seuil (cm/s²)
Mach	1875	2		Z, Z (Vert.)	10-12
Travis	1928	2	Assis	Z, X (Horiz.)	20-25
		2	Debout	Z, X (Horiz.)	8
		2	Debout	Z, Y (Horiz.)	15
Gurnee	1934	3	Assis	Z, Z (Vert.)	8-10
Jonkees et Groen	1946	2		X, Z (Horiz.)	6-13
Walsh	1962	6	Allongé	X, Z (Horiz.)	7-8
		6	Allongé	X, Z (Horiz.)	9
Benson	1975	12	Assis	Tilt Z, X (Vert.)	30
Hosman	1978	3	Assis	Z, Z (Vert.)	3-7

seuils qui varient en fonction du protocole choisi (posture, type de mouvement), du matériel utilisé (précision, fréquence de fonctionnement, etc.) et de la fréquence de simulation. Ainsi la perception assise et debout n'est pas la même que celle allongée où la gravité terrestre n'est plus dans le même axe.

R.Fitzpatrick et D. I. McCloskey ont synthétisé dans l'article "*Proprioceptive, visual and vestibular thresholds for the perception of sway during standing in humans*" [FM94] la capacité humaine à percevoir les mouvements de balancement. Cette étude comporte 4 expérimentations menées sur des sujets sains et a pour but d'estimer le seuil de vitesse angulaire perceptible par l'humain. Dans le cas de l'expérimentation basée uniquement sur le système vestibulaire (pas d'information visuelle ni somato-sensorielle), le seuil a été relevé à une vitesse angulaire de 0.002 rad/s^1 soit 0.115°/s^1.

L.E.Zaichik, dans l'article "*Acceleration perception*" [Zai+99], a étudié les seuils de perception absolue et différentielle de l'humain à l'aide de simulation réalisé avec une plateforme à 6 dégrés de liberté. Les mouvements effectués par la plateforme étaient des accélérations de nature sinusoïdale à diverses fréquences. Le tableau 7.2 résume le relevé des valeurs de seuils absolus pour chacun des types de mouvement. Cependant, ces relevés de seuils absolus

Table 7.2 – Tableau récapitulatif des valeurs absolues de seuil de perception du mouvement propre.

Mouvement	Fréquence	Seuil de perception
Longitudinale	Basse	0.098m/s²
Latérale	Basse	0.025m/s²
Verticale	Basse	0.128m/s²
Longitudinale	Haute	0.004m/s²
Latérale	Haute	0.007m/s²
Verticale	Haute	0.008m/s²
Lacet	Basse	0.75°/s
Roulis	Basse	0.40°/s
Tangage	Basse	0.95°/s
Lacet	Haute	0.50°/s
Roulis	Haute	0.30°/s
Tangage	Haute	0.70°/s

ne permettent pas de les généraliser pour des forces spécifiques avec des amplitudes variantes au cours du temps.

Pour cela, les seuils différentiels, correspondant à l'incertitude perception liée à la variation de force spécifique (accélération sinusoïdale), sont calculés

pour chacun des axes linéaires de sorte d'obtenir une linéarité entre le seuil absolu Δ_a et le seuil différentiel Δ_d de la manière suivante :

$$\Delta_d = \Delta_a + ka \tag{7.8}$$

Où k est une constante liée à l'axe linéaire considéré et a l'accélération résultant du total des forces spécifiques (accélérations hautes et basses fréquences). Notons que k vaut 0.9, 0.6 et 0.4 respectivement pour les mouvements verticaux, longitudinaux et latéraux.

Une étude des seuils de perception effectuée sur le simulateur SIMONA [Hee+05] ont permis d'affiner l'estimation des seuils grâce à la considération de deux seuils secondaires : le seuil inférieur et le seuil supérieur. Le seuil supérieur correspond à l'accélération à partir de laquelle l'individu est capable de percevoir le mouvement à l'aide de son système vestibulaire. Le seuil inférieur est celui correspondant à l'accélération à partir de laquelle il n'est plus capable de percevoir un mouvement lorsqu'il le percevait au préalable.

Ces travaux ont été repris par E.L.Groen et M.Wentink [Gro+06] pour en extraire une valeur de seuil moyenne pour les 6 degrés de liberté offerts par le simulateur SIMONA. Cette étude simplifie les conditions d'estimation des seuils de perception en considérant la fréquence du système comme constante, ce qui permet d'obtenir une valeur moyenne transposable à toutes les situations et proche des valeurs réelles. Ces seuils ont ensuite été intégrés à un modèle de perception du mouvement en ne considérant que les accélérations satisfaisant à ces seuils.

Table 7.3 – Tableau récapitulatif des seuils de perception du mouvement propre.

Organe sensoriel	Mouvement	Seuil de perception
Otholith	Longitudinale	7.42 $\cdot 10^{-2} m/s^2$
	Latérale	7.43 $\cdot 10^{-2} m/s^2$
	Verticale	1.23 $\cdot 10^{-1} m/s^2$
Canaux Semi-Circulaire	Roulis	5.21 $\cdot 10^{-3}$ rad/s
	Tangage	7.34 $\cdot 10^{-3}$ rad/s
	Lacet	1.66 $\cdot 10^{-2}$ rad/s

7.2.3 . Modèles de perception

R.J.Telban [TC01] a intégré un modèle de perception qui considère les interactions visuelles et vestibulaires. Une comparaison de l'existant dans l'état de l'art a permis de mettre en évidence qu'une simple addition des informations visuelles et inertielles ne permettait pas de reproduire le processus de perception du mouvement propre. Ainsi, l'étude menée a établi l'intérêt de pondérer ces informations complémentaires. Le modèle proposé (voir figure 7.7) intègre les modèles mathématiques des otholites et canaux semi-circulaires ainsi que la vection provenant du système visuel sous la forme d'un filtre passe-bas du premier ordre. Il considère la latence issue de la vection ce qui permet entre autres de pondérer l'influence des deux sources d'information. Dans ces travaux, les modèles de perception angulaire et linéaire sont considérés comme indépendants et donc implémentés séparément.

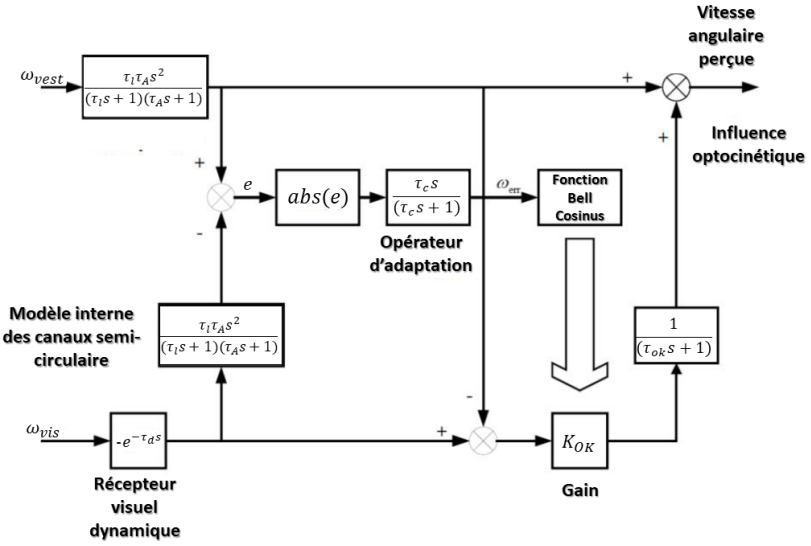


Figure 7.7 – Modèle de perception angulaire de Telban.

Une modélisation plus complète du processus de perception du mouvement est fourni dans l'article [BBH01] où les signaux provenant du système vestibulaire sont traités de sorte à reproduire la manière dont l'humain sépare l'accélération due au mouvement de celle liée à la gravité terrestre. Ces travaux de recherche mettent en évidence l'existence d'un processus de filtrage des basses fréquences chez l'humain qui permet de ne pas considérer l'accélération constante de la gravité. L'article [BB02] présente une approche théorique pour la résolution de ce problème de distinction d'accélération. Il met en avant l'interaction existante entre les canaux semi-circulaires et les otolithes sous une forme tridimensionnelle basée sur la description bidimen-

sionnelle de ce phénomène par Mayne [Ben+74]. Cette approche décompose l'accélération totale en une composante de mouvement et une composante de gravité terrestre en utilisant les informations d'inclinaison et de translation et un filtre passe-bas.

Le modèle de perception retenu et nommé *TNO model* (voir figure 7.8), sépare les informations issues des mouvements linéaires et circulaires pour combiner les signaux visuels et vestibulaires avec des coefficients de poids correspondant à l'importance de chacune des informations pour l'humain. Le modèle permet de déterminer les vitesses linéaire et angulaire perçues ainsi que le vecteur idiotropique (verticale subjective liée entre autres à la gravité) perçue.

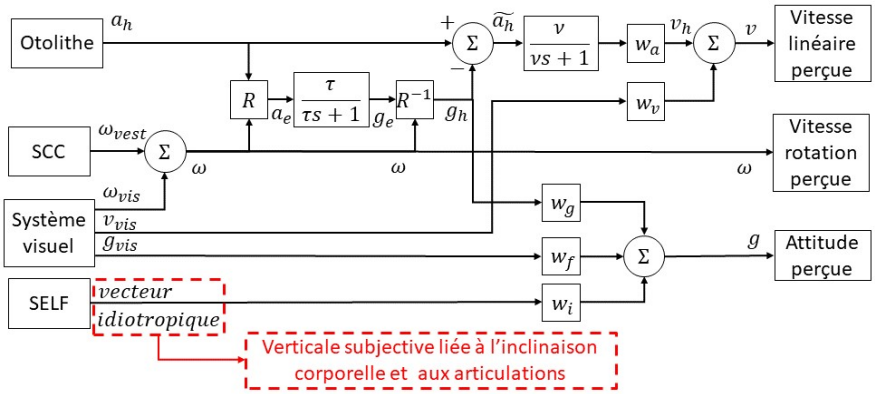


Figure 7.8 – Modèle de perception TNO.

Notons que la perception idiotropique correspond à l'association d'information visuelle concernant les lignes (verticale/horizontale) d'une scène observée, les informations vestibulaires relatives à la perception de l'orientation de la gravité, et les informations corporelles à travers la verticale subjective provenant de l'axe longitudinal du corps et les ressentis articulaires. Mittelstaedt a étudié dans ses travaux [Mit83 ; Mit88] ce rapport à la verticale subjective à travers le ressenti corporel, travaux qui ont été repris dans le modèle TNO.

Ce modèle a été évalué et validé sur la plateforme robotique de simulation *SIMONA* (voir figure 7.9) des travaux de E.L.Groen et R.J.Hosman [GCH00]. Cette plateforme robotique de Gough-Stewart offrait 6 degrés de liberté et des capacités de mouvement et d'accélération adaptées à la simulation de vol. De plus, la présence d'un écran permettant l'affichage de contenu visuel rendait possible l'évaluation des interactions des systèmes visuel et vestibulaire pour la perception du mouvement.

Figure 7.9 – Simulateur SIMONA.

J.E.Bos et R.J.Hosman ont approfondi cette recherche dans des travaux [BHB02] intégrant le modèle TNO pour des expérimentations sur une plateforme robotique de Gough-Stewart et un simulateur à base fixe (centrifugeuse). Cette étude avait pour objectifs de valider la fiabilité du modèle TNO dans des conditions expérimentales différentes, et d'optimiser les simulateurs utilisés en déterminant les paramètres optimaux des filtres de mouvement permettant de reproduire au mieux les sensations de mouvement simulés (essentiellement des décollages d'avion).

Le modèle TNO a également été éprouvé dans les travaux suivants de E.L.Groen et R.J.Hosman [GSH07] à l'aide du simulateur de vol NLRresearch (GFORCE) (voir figure 7.10) a 6 degrés de liberté. Lors de cette étude, la perception du mouvement de onze pilotes d'avion a été évaluée pour un ensemble de manœuvres d'atterrissage en cas de vent transversal, dans des conditions de simulation avec ou sans filtre de *washout* pour les mouvements du simulateur.

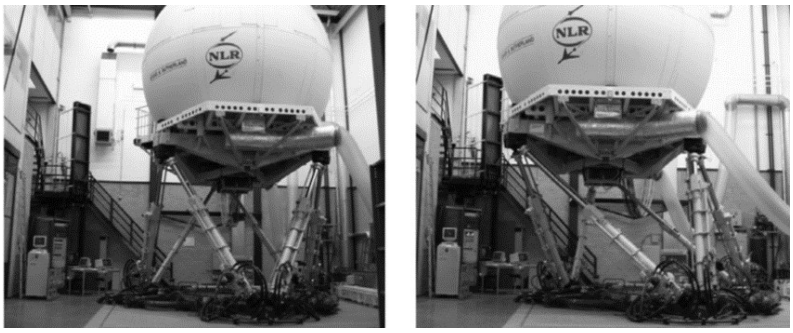


Figure 7.10 – Simulateur NLRresearch (GFORCE).

L'effet des informations visuelles a été considéré en effectuant les tests avec, puis sans une vue extérieure simulée. L'écart de perception dans ces deux

cas pour un même mouvement du simulateur a mis en évidence l'importance de l'information visuelle dans la perception du mouvement. De même, cette étude a démontré que la bonne perception du mouvement était réalisée pour les mouvements de balancement et de roulis du simulateur à l'inverse du mouvement de lacet. Pour les mouvements non filtrés par le filtre *washout*, les mouvements de balancement provoquaient des ressentis de mouvement trop fort même malgré une réduction de l'espace de travail de la plateforme par rapport au mouvement réel de l'avion. Les données recueillies ont permis également de valider le modèle TNO et d'optimiser ses paramètres.

Des études complémentaires [HCB11] menées par la même équipe de recherche a évalué le modèle TNO par rapport à ceux existant dont le Telban [TC01]. Cette étude comparative a mis en évidence la supériorité du modèle TNO qui considère à la fois l'impact de la gravité dans la perception, le fonctionnement du système nerveux central et également les sensations de verticale subjective. L'intégration des seuils de perception vestibulaire a également été proposée dans l'article [Gro+06] afin de présenter un modèle incorporant les limites physionomiques de perception du mouvement chez l'humain. Les résultats obtenus démontrent que les informations visuelles augmentent les seuils de perception du mouvement pour un stimulus inertiel donné.

La corrélation entre le système robotique utilisé dans ces travaux de recherche (plateforme de Gough-Stewart) et celui utilisé dans nos travaux ainsi que les nombreux travaux réalisés sur le modèle TNO sur de vrais simulateurs mettent en évidence l'importance de considérer ce modèle pour nos futurs travaux.

7.3 . Perception du mouvement propre : humain

7.3.1 . Organes sensoriels et proprioré cepteurs

La perception de son propre mouvement chez l'humain est la capacité à percevoir son mouvement de manière globale en considérant le corps comme rigide dans son environnement et de manière localisée en considérant le corps comme un élément non-rigide composé d'articulations. Cette perception est rendue possible grâce à la combinaison des organes sensoriels (perception globale) et des organes de proprioception et kinesthésie (perception localisée). Seulement deux organes sensoriels interviennent dans le processus de perception, le système visuel à travers l'œil, et le système vestibulaire à travers l'oreille interne.

L'oreille interne illustrée dans la figure 7.11 fait partie de l'organe de l'ouïe. Il est plus particulièrement composé du système vestibulaire, un sous-système qui est un organe sensoriel barosensible capable de percevoir les accélérations linéaires et les mouvements de rotation. Le système vestibulaire est l'organe responsable de la capacité de perception du mouvement et du maintien de l'équilibre chez l'humain.

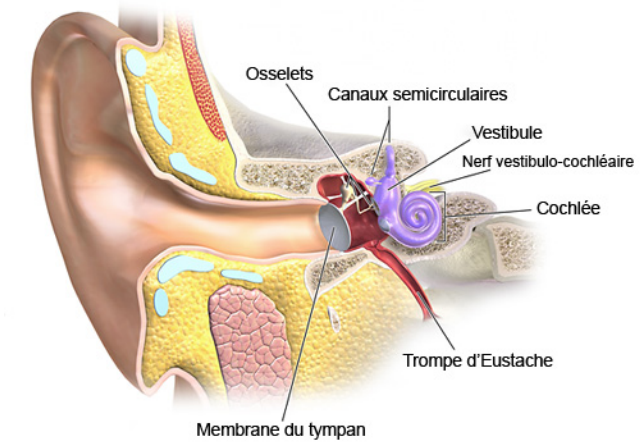


Figure 7.11 – Illustration de l'oreille interne humaine.

Au sein du système vestibulaire, trois canaux semi-circulaires sont responsables de la perception des mouvements de rotation, et le saccule et l'utricule composant les organes otolithiques permettent la perception des accélérations linéaires. Tous ces composants sont remplis d'un liquide appelé endolymphe permettant de réguler les impulsions électrochimiques des cellules ciliées. Chacun des trois canaux semi-circulaires est relié à une ampoule osseuse contenant une crête ampulaire composée d'une capsule gélatineuse épaisse appelée cupule et de nombreuses cellules ciliées. Lors d'un mouve-

ment de rotation de la tête, le liquide endolymphe se déplace dans le sens inverse du mouvement dans l'ensemble des canaux. Ce mouvement est alors estimé par les cils des cellules ciliées présentes dans la crête ampulaire qui relèvera l'orientation et l'intensité du mouvement. La combinaison des trois canaux semi-circulaires orthogonaux, à savoir antérieur (supérieur), latéral (horizontal) et postérieur, permet une estimation du mouvement dans un repère en trois dimensions adapté aux déplacements de l'humain.

- Canaux semi-circulaires antérieurs : mouvement d'avant/arrière correspondant à un hochement de la tête (tangage).
- Canaux semi-circulaires latéraux : mouvement de gauche/droite de la tête (lacet).
- Canaux semi-circulaires postérieurs : mouvement d'inclinaison à gauche/droite de la tête (roulis).

Les organes otolithiques situés dans le vestibule perçoivent les accélérations linéaires horizontale pour l'utricule et verticale pour le saccule provoquant le mouvement des macules baignant dans le liquide endolymphe. Ces macules sont composées d'une membrane otolithique recouverte d'une couche d'otolithe dans laquelle baignent des cils reliés à des cellules ciliées. À l'inverse des canaux semi-circulaires où les cellules ciliées sont à l'extérieur des canaux, les macules sont situées dans le saccule et l'utricule.

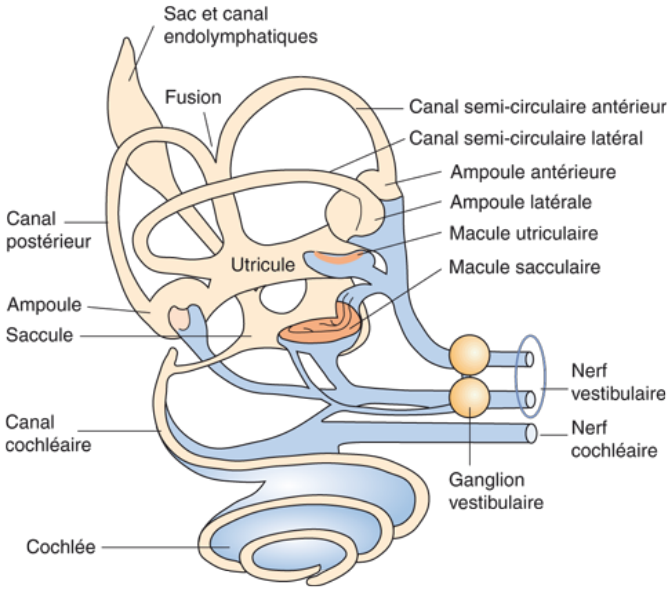


Figure 7.12 – Système vestibulaire.

La membrane otolithique ainsi que la couche d'otolithe permettent le mouvement des cils converti en message nerveux lors d'accélérations linéaires.

Les accélérations latérale et longitudinale perçues par le système maculaire sont mises en perspective avec l'accélération verticale résultant de la pesanteur pour déterminer l'orientation de la tête dans son environnement. Ainsi, la perception du mouvement linéaire est une résultante de l'impact du mouvement corrélé à la gravité terrestre sur le système vestibulaire. Le système

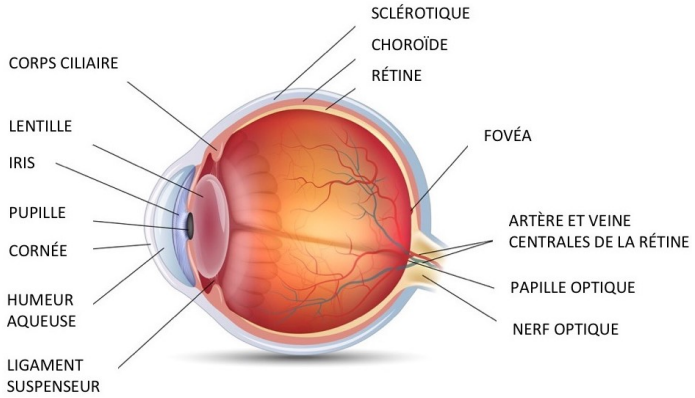


Figure 7.13 – Système visuel humain.

visuel humain est l'organe sensoriel intégrant le sens de la vue permettant la perception de l'environnement en trois dimensions à travers l'acquisition et l'interprétation de deux images en deux dimensions. Ce système est composé de l'œil rattaché au cortex visuel du cerveau par des nerfs optiques, un chiasma optique, et un noyau du corps genouillé latéral. Les systèmes caméra et photographique ont été conçu en s'inspirant du fonctionnement de l'œil humain. L'œil est ainsi constitué d'une pupille qui diaphragme la lumière qui est ensuite déviée par la cornée et le cristallin. Par homothétie, le diaphragme de l'appareil photo correspond à la pupille de l'œil tandis que la lentille correspond au cristallin. La lumière déviée par le cristallin est projetée sur la rétine sous une forme conique inversée de la scène observée. La lumière projetée sur la rétine sous forme de photon est interprétée en zone de contraste qui est ensuite convertie en signaux chimiques de potentiels d'action (influx nerveux) transmis à travers le nerf optique vers le cortex visuel. Le système visuel humain est constitué de deux sous-systèmes visuels permettant le transport de l'information visuelle du photon jusqu'au cortex visuel.

- Le premier système de transport est le système parvo-cellulaire qui transmet les signaux nerveux permettant la distinction des formes, couleurs et textures. C'est le système permettant l'analyse des informations visuelles sous forme d'image selon des critères de contenus, caractéristiques, etc. Il récupère les informations issues du centre de la rétine qui contient l'essentiel des cônes et cellules ganglionnaires de type P permettant la reconnaissance de l'information.

- Le second système appelé magno-cellulaire traite les informations de mouvement (flux visuel), de profondeur (vision binoculaire) et de vision globale (coin, ligne, etc.) . Il est donc le système permettant la détection de l'information avec une forte sensibilité au changement à l'inverse du système parvo-cellulaire. Il récupère les informations visuelles qui sont projetées sur la zone périphérique de la rétine, c'est dans cette zone que se trouve la majorité des bâtonnets et des cellules ganglionnaires de type M.

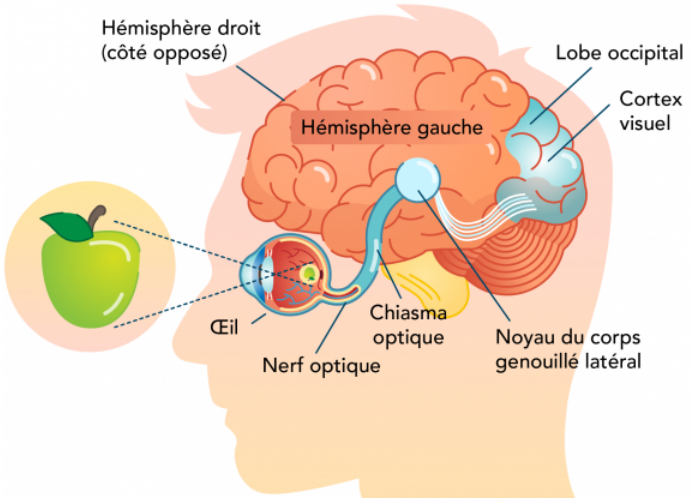


Figure 7.14 – Cortex visuel du cerveau.

Les cônes et bâtonnets de ces deux sous-systèmes sont activés par la lumière de manière discontinue en raison du temps des réactions photo-chimiques s'opérant pour la conversion des photons en agents chimiques au niveau des pigments. Les informations visuelles provenant de ces deux systèmes sont combinées par le cerveau au sein du cortex visuel qui synthétise ces informations avec ses connaissances préalables afin d'obtenir une perception spatio-temporelle de l'environnement observé.

Les organes proprioré cepteurs sont l'ensemble des organes responsable de la perception du mouvement des différentes parties du corps humain. Cette perception est rendue possible grâce aux informations nerveuses provenant des muscles, tendons, ligaments, os et articulations, qui sont acheminés par le système nerveux humain. Ces récepteurs, à savoir les fuseaux neuromusculaires et organes neurotendineux, sont considérés comme proprioceptifs car réagissant à des stimuli provenant de l'organe même (contraction, mouvement, etc.). Ces influx nerveux sont transmis au système nerveux central qui les traite de manière plus ou moins consciente du fait de la "pro-

fondeur" de la sensation. Il en résulte une perception de la contraction ou tonus des muscles et la position relative des différentes parties du corps (bras, jambes, tête, etc.) d'après les informations articulaires et osseux.

7.3.2 . Modèle visuo-inertiel : perception de son mouvement

Comme détaillé dans l'état de l'art, de nombreux travaux ont tenté de modéliser la capacité humaine de perception de son propre mouvement à travers divers organes sensoriels et/ou proprioréceptifs. Tout d'abord décrit uniquement selon les informations inertielles provenant du système vestibulaire, les modèles les plus complets ont tenté d'intégrer le flux d'information visuelle provenant du système visuel. La modélisation des capacités de perception de mouvement chez l'humain requiert une étude expérimentale à effectuer sur des humains soumis à des mouvements connus où leur sensibilité aux mouvements est recueillie et analysée. Cette analyse permet d'extraire une fonction de transfert pour chacun des éléments du système vestibulaire correspondant aux capacités humaines de perception du mouvement.

Dans notre cas, la capacité de charge de la plateforme XY-6Dof de notre laboratoire ne permet pas d'y installer un humain pour effectuer ces expérimentations. En conséquence, il a été nécessaire de reprendre un modèle d'une étude préalablement faite par un autre laboratoire possédant le matériel requis. Le modèle de perception TNO [HCB11] obtenu à l'aide de nombreux tests réalisés sur des plateformes robotiques de Gough-Stewart a permis de modéliser la perception du mouvement propre d'après uniquement les informations inertielles provenant du système vestibulaire. Les canaux semi-circulaires du système vestibulaire permettent la perception des mouvements rotationnels s'opérant sur la tête ont été modélisés par de nombreuses fonctions de transfert [**empty citation**]. Une approximation de la fonction de transfert des canaux semi-circulaires ω_{CSC} est donnée selon la vitesse angulaire de la tête ω_t et une constante de temps notée $\tau_c = 10s$:

$$\omega_{CSC} = \frac{\tau_c s}{\tau_c s + 1}\omega_t \tag{7.9}$$

Les mouvements linéaires sont captés par les otolithes présents dans le système vestibulaire qui correspondent à trois accéléromètres placés de manière orthogonale pour garantir une perception 3D du mouvement. Dans un premier temps, la fonction de transfert [**empty citation**] du système otolithique était modélisé par l'équation suivante :

$$\frac{\hat{f}(s)}{f(s)} = K_{OTO}\frac{\tau_L s + 1}{(1 + \tau_1 s)(1 + \tau_2 s)} \tag{7.10}$$

Où f et $\hat{f}$ sont respectivement la force spécifique effective et celle ressentie. Cependant, cette fonction ne permet pas de reproduire le traitement com-

plexe s'opérant au sein du système nerveux central (SNC) lors de la perception de mouvement. En effet, sur Terre, lors d'un mouvement, l'humain est soumis à la force f_a provoquant sa mise en mouvement, mais également à la force gravitationnelle terrestre g qui est constante. Ainsi, une force spécifique notée f, correspondant à une combinaison de l'accélération du corps et de la gravité comme étant $f = a + g$, induit une réponse linéaire du système otolithique. En prenant en compte le principe d'équivalence d'Enstein, il a été déduit que le SNC filtre la gravité puisque nous ne la ressentons pas lors d'un mouvement. La gravité terrestre étant constante, Mayne [Ben+74] a proposé de modéliser le processus de filtrage du SNC par un filtre passe-bas. Néanmoins, le repère du système otolitique correspond à celui de la tête tandis que celui de la gravité est compris dans le repère terrestre. Ainsi l'accélération a correspondant au mouvement réel de l'individu doit être extrait de la force spécifique f en considérant la matrice de passage $R_{O/T}$ du repère otolithe au repère Terre, de la manière suivante :

$$a = R_{O/T}(f) - g \tag{7.11}$$

En se basant sur les travaux de Bos et Bles (2001) [BB02], la matrice de rotation $R_{O/T}$ peut être extraite des informations rotationnelles issues des canaux semi-circulaires et/ou du système de vision. Cette étude, basée sur le constat de Mayne [Ben+74] concernant la gravité, exprime la matrice de rotation $R_{O/T}$ comme étant celle permettant de passer l'Accélération Gravito-Inertielle (AGI) f, aussi appelée force spécifique, du repère tête en un vecteur accélération dans le repère Terre. Une fois l'accélération projetée dans le repère terre, la gravité $\tilde{g}$ peut être estimée par un filtre passe-bas appliqué à la force spécifique f. En notation Laplacienne, l'accélération perçue $\tilde{a}$ peut être exprimée de la manière suivante :

$$\tilde{a} = f - \tilde{g} = f(1 - \frac{1}{\tau s + 1}) = \frac{\tau s}{\tau s + 1} f \tag{7.12}$$

L'accélération perçue $\tilde{a}$ est de fait la réponse passe-haut du ressenti de la verticale détecté issue de la perception de la gravité par le SNC. L'estimation de la gravité par le SNC est illustrée dans la figure 7.15.

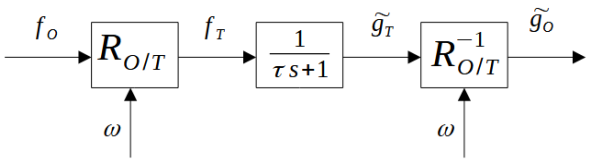


Figure 7.15 – Modèle de l'estimation de la gravité par le SNC.

L'information inertielle provenant du système otolithique est transformée selon $R_{O/T}$ pour obtenir une accélération exprimée selon le repère terre, l'accélération est ensuite déduite puis supprimée à l'aide d'un filtre passe bas pour

n'extraire que la gravité estimée $\tilde{g_T}$ selon le repère terre, puis une rotation inverse $R_{O/T}^{-1}$ est appliquée à cette gravité $\tilde{g_T}$ pour l'exprimer selon le repère tête. Finalement, cette gravité perçue $\tilde{g}$ est soustraite à l'AGI f pour déduire l'accélération a comme illustré dans la figure 7.16. Dans ce modèle, ω, f_O, f_T, $\tilde{g_T}$ et $\tilde{g_O}$ correspondent respectivement à l'orientation de la tête provenant des canaux semi-circulaires, la force spécifique selon le repère Otolithique, la force spécifique selon le repère terre, la gravité estimée selon le repère terre et la gravité estimée selon le repère Otolithique.

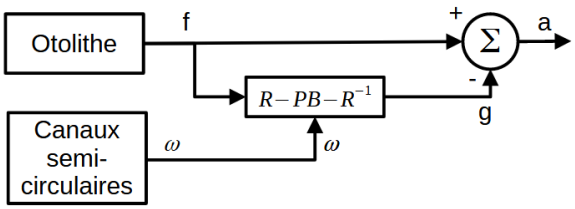


Figure 7.16 – Modèle globale du filtrage de la gravité par le SNC.

Dans la figure 7.16, $R - PB - R^{-1}$ fait référence aux étapes de changement de repères par rotations R et R^{-1}, et d'application du filtre Passe-Bas (PB) décrites dans la figure précédente 7.15.

Ce modèle a été exprimé mathématiquement dans [BB02] sous la forme d'une équation différentielle tridimensionnelle donnant une estimation de la gravité $\tilde{g}$. En effet, de manière analogue à l'équation 7.10, le filtrage de la gravité peut être exprimé en notation Laplacienne comme suit :

$$\frac{d\tilde{g_T}}{dt} = \frac{1}{\tau}(f - \tilde{g_T}), \quad avec \;\; \tilde{g_T} = \frac{1}{\tau s + 1} f_T \tag{7.13}$$

Étant donné que $\tilde{g_O} = R^{-1}\tilde{g_T}$, il en résulte que $\tilde{g_T} = R\tilde{g_O}$, ce qui nous permet de réécrire l'équation précédente 7.13 en prenant compte ces notations :

$$\frac{dR\tilde{g_O}}{dt} = \frac{1}{\tau}(Rf_O - R\tilde{g_O}) \tag{7.14}$$

La matrice de Rotation R peut être remplacé par la vitesse angulaire ω détectée par les canaux en appliquant la règle du produit (règle de Leibniz) à l'équation 7.13. Pour simplifier l'écriture, la gravité et la force spécifique perçues seront notées g et f, l'équation s'écrit alors :

$$\frac{dR}{dt}\tilde{g} + R\frac{d\tilde{g}}{dt} = \frac{1}{\tau}(Rf - R\tilde{g}) \tag{7.15}$$

En appliquant la matrice inverse R^{-1}, l'équation obtenue est :

$$\frac{d\tilde{g}}{dt} = \frac{1}{\tau}(f - \tilde{g}) - R^{-1}\frac{dR}{dt}\tilde{g} \tag{7.16}$$

Si nous considérons le cas d'un vecteur x auquel nous appliquons uniquement une rotation R donnant $y = Rx$, la dérivé de y s'écrit :

$$\frac{dy}{dt} = \frac{dR}{dt}x = \omega \times y \tag{7.17}$$

En ré-exprimant l'équation 7.17 selon la gravité estimé $\tilde{g}$ et la vitesse angulaire ω :

$$\frac{dR}{dt}\tilde{g} = R\omega \times R\tilde{g} \tag{7.18}$$

L'équation 7.16 peut être exprimée selon l'équation 7.18 de la manière suivante :

$$\frac{d\tilde{g}}{dt} = \frac{1}{\tau}(f - \tilde{g}) - R^{-1}(R\omega \times R\tilde{g}) \tag{7.19}$$

Ainsi le modèle d'estimation de la gravité par le Système Nerveux Central s'écrit :

$$\frac{d\tilde{g}}{dt} = \frac{1}{\tau}(f - \tilde{g}) - \omega \times \tilde{g} \tag{7.20}$$

Par la suite, les simulateurs de recherche SIMONA [ADV93] et ont permis d'incorporer les informations visuelles aux expérimentations en intégrant des écrans dans le cockpit du simulateur. La perception du mouvement par le système visuel humain correspond à l'acquisition du flux lumineux par la rétine qui est ensuite transformé en signal électrique à travers un procédé chimique qui produit des potentiels d'action transmis au cerveau par les nerfs optique puis traité par le cortex visuel. Ce processus est relativement long comparé à celui du système vestibulaire du fait de la nature du signal visuel et des informations qui sont davantage complexes et abstraites. Le traitement des informations par le cortex visuel et le transport de l'information via les nerfs optiques induisent des temps de retard dans le traitement comme expliqué dans la section précédente 7.3.1.

Le contexte de notre étude appliquée à l'étude du mouvement dans un simulateur induit de fait une nécessité de duper la perception visuelle du mouvement. En effet, dans un simulateur, le flux d'information visuelle fournit au cortex visuel des informations de mouvements erronées induisant une interprétation de mouvement de l'individu en se basant sur son système visuel tandis que son corps à travers le mouvement de la plateforme robotique ne suit pas le même mouvement ou n'effectue pas de mouvement. Ce phénomène de sensation erronée de mouvement basé sur la vision s'appelle la "vection".

À l'inverse du système vestibulaire, le système visuel humain ne perçoit pas les accélérations. En effet, le traitement de l'information par le cortex visuel n'est adapté que pour l'acquisition et l'analyse de vitesse et position. Dans le cas de la vection linéaire (mouvement linéaire induit), nous considérons que le processus résultant est une fonction de transfert du premier ordre. Dans

ce système, v_T correspond à la vitesse dans le référentiel terrestre, v_{vis} la vitesse linéaire perçue résultant de la vection linéaire et τ_v est la constante de temps représentant le temps de traitement de l'information visuel par le cortex visuel.

$$v_{vis} = \frac{1}{\tau_v s + 1} v_T \tag{7.21}$$

Pour les rotations au niveau de la tête, le phénomène de perception visuelle du mouvement est appelé vection circulaire. Dans le cadre de simulation en réalité mixte, flux visuel virtuel et mouvement réel, le temps d'analyse est relativement long. Étant la perception inverse de la vitesse angulaire du système vestibulaire, la fonction de transfert résultante reprend la constante de temps τ_c de la fonction des canaux semi-circulaires. Notons que, ω_O et $\omega_v is$ sont respectivement la vitesse angulaire déduite par le cortex visuel et la vitesse angulaire perçue selon le référentiel de la tête. La fonction représentant cette estimation est exprimée par :

$$\omega_{vis} = \frac{1}{\tau_c s + 1} \omega_O \tag{7.22}$$

Enfin, d'un point de vue de la vision, la verticale subjective correspond à la direction de la gravité estimée d'après une analyse visuelle du contenu de la scène observée. En effet, l'humain est habitué à évoluer dans un environnement naturellement composé de structure et d'éléments rectilignes. L'horizon, les bâtiments, les meubles et certains éléments terrestres (arbres, animaux, etc.) ont des structures majoritairement constituées de lignes perpendiculaires étant dans l'orientation de la gravité (normale à l'horizon donc verticale) ou dans l'orientation de l'horizon donc horizontale. L'analyse de ces orientations permet d'estimer la verticale subjective en se référant au repère tête par rapport à l'environnement (repère terrestre). Cette estimation de l'orientation de la gravité fournit de fait une estimation partielle de l'orientation de la tête (roulis et tangage) induite par la rotation qui s'y applique. Ce processus rapide est également modélisé par une fonction de transfert de premier ordre avec comme constante de temps τ_v :

$$g_{vis} = \frac{1}{\tau_v s + 1} g_T \tag{7.23}$$

L'humain utilise les interactions visuelles-vestibulaires s'opérant lors de la perception de son mouvement propre pour fournir une estimation fiable de son mouvement. Cette estimation permet son équilibre et son déplacement dans son environnement en garantissant une robustesse aux situations de mouvement ou de scène complexe ou contradictoire en tirant parti des avantages et inconvénients de chacun des systèmes de perception à travers la complémentarité de ses informations.

Le modèle TNO [BBH01] proposé pour nos travaux est un modèle intégrant les interactions visuel-vestibulaire, mais également le vecteur idiotropique relatif à l'alignement corporel ressenti essentiellement au niveau de l'articulation des cervicales. Un écart angulaire entre l'alignement des vertèbres dorsales et l'orientation de la tête au niveau des vertèbres cervicales perçu par les terminaisons nerveuses localisé à cet endroit apporte une estimation de l'orientation de la tête par rapport au corps. Également, les informations articulaires du bassin, des genoux et chevilles permettent d'estimer un alignement du corps de manière individuelle, puis une mise en perspective de l'orientation de la tête par rapport au contact du sol en considérant l'ensemble des articulations du corps (voir la figure 7.17). Le système vestibulaire fournit

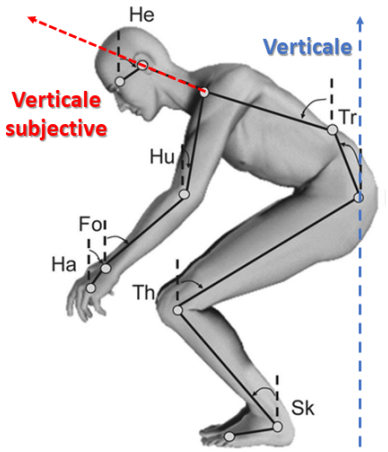


Figure 7.17 – Estimation de l'alignement corporel pour l'estimation de la verticale et l'horizontale subjectives.

une estimation de la magnitude et l'orientation de la gravité tandis que les systèmes visuels et proprioceptifs (articulations) n'indiquent que l'orientation de la gravité perçue. Des travaux de Mittelstaedt [Mit83 ; Mit88] ont mis en lumière le processus d'estimation de la verticale subjective ainsi que l'addition des sources d'information des trois systèmes à travers des coefficients d'importance.

Dans ce modèle, les informations de mouvement angulaire issues des systèmes visuel et vestibulaire sont combinées. Elles fournissent ensuite une estimation de la vitesse de rotation perçue ω par l'humain qui sert ensuite d'indicateur de l'orientation R s'opérant sur la tête d'après les informations inertielles. Cet indicateur permet l'estimation de gravité $\tilde{g}$ pour extraire l'accélération linéaire perçue $\tilde{a}$ de la force spécifique (AGI) f. Le système visuel percevant uniquement les vitesses, il est nécessaire d'estimer la vitesse perçue en intégrant l'accélération perçue. Cependant, une simple intégration ne garantirait pas une vitesse nulle dans le cas d'accélération subjective remise

à zéro liée au contexte de simulation. En conséquence, une intégration suivie d'un filtre passe-haut impliquera une remise à zéro de la vitesse vestibulaire dans cette situation. La vitesse vestibulaire perçue $\tilde{v}$ est alors modélisée d'après l'accélération vestibulaire perçue $\tilde{a}$ et la constante de temps $\nu = 5s$ par :

$$\tilde{v} = \frac{1}{s}\frac{\nu s}{\nu s + 1}\tilde{a} = \frac{\nu}{\nu s + 1}\tilde{a} \tag{7.24}$$

L'intégration suivie du filtre passe-haut induit en conséquence l'application d'un filtre passe-bas sur l'accélération vestibulaire perçue $\tilde{a}$. Des poids w_n sont assignés à chacune des sources estimées pour combiner linéairement celles en interaction tout en garantissant une cohérence de l'importance du type de source. D'après de nombreuses études [GJH02; Gra+12; KLM14], les informations visuelles sont considérées comme prédominantes par rapport aux informations vestibulaire et articulaire, car provenant d'un processus de traitement de l'information plus complexe et précis.
Le modèle TNO est schématisé dans la figure 7.18 synthétisant l'ensemble des fonctions, combinaisons et prédominances des sources décrites précédemment.

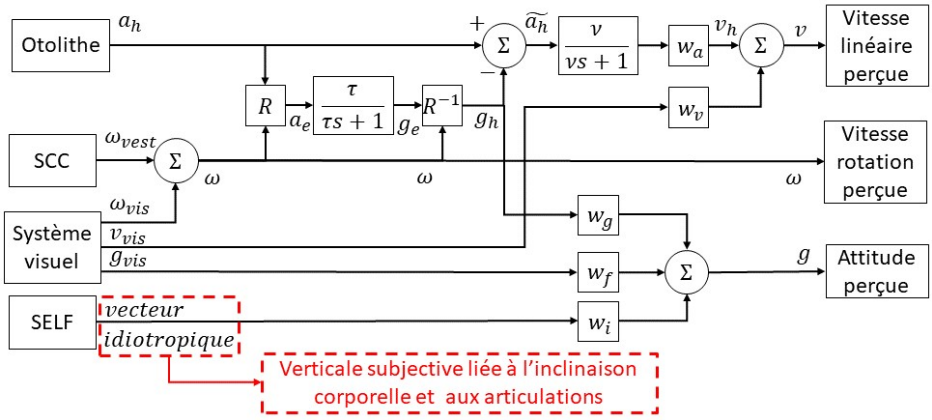


Figure 7.18 – Modèle de perception TNO.

Le tableau 7.4 récapitule les poids de prédominance de chacune des sources en fonction des situations de simulation, à savoir avec ou sans vision.

7.3.3 . Seuil de perception articulaire de l'humain

L'estimation de la verticale subjective, autrement dit de la gravité, est liée à la combinaison d'information visuelle, vestibulaire et proprioceptive à travers le vecteur idiotropique relatif aux ressentis articulaires. De ce fait, les ressentis articulaires offrent une source d'information permettant une double perception du mouvement :

- Une perception globale de son corps en définissant son mouvement

Table 7.4 – Poids de prédominance des sources pour le modèle de perception .

Coefficient de poids	Vitesse linéaire		Orientation		
	w_a	w_v	w_g	w_f	w_i
Avec vision	0.2	0.8	0.2	0.75	0.05
Sans vision	1	0	0.95	0	0.05

par rapport à des ressentis internes et une analyse de son environnement au cours du temps.

- Une perception individuelle à travers un schéma corporel qui localise la position de chacun des membres les uns par rapport aux autres.

Pour le premier type de perception, les informations articulaires sont analysées par le cerveau humain selon le référentiel tête afin de déduire la verticale subjective en fonction de chacun des ressentis allant des articulations des membres (pied si debout, fessier si assis et dos si allongé) en contact avec une surface plane de référence (sol, chaise, lit, etc.) jusqu'aux articulations de la nuque. Cette surface de contact étant considérée comme l'horizontale et plan de référence, la verticale subjective est ainsi la normale par rapport au segment corporel en contact avec le plan de référence. Le second type de perception est basé sur la reconstruction d'un schéma corporel utilisant les ressentis articulaires pour déterminer la position de chacun des segments du corps les uns par rapport aux autres et par rapport à l'environnement. La perception du mouvement propre intègre le premier type de perception pour déduire la verticale subjective, tandis que le second est davantage utile pour les déplacements et mouvements.

Comme tout ressenti corporel humain, la perception est relativement abstraite, imprécise et légèrement fluctuante d'un individu à l'autre. Partant de ce constat et du manque de travaux au sujet de cette imprécision de perception articulaire, il nous a semblé pertinent de mener une étude à ce sujet. En conséquence, des expérimentations sur des sujets ont été menées pour déterminer articulation par articulation l'incertitude d'estimation de position angulaire des membres supérieurs. Les expérimentations consistaient à relever l'estimation angulaire faite par les sujets de l'expérience en se basant uniquement sur leurs ressentis articulaires. Pour cela, les sujets étaient équipés d'une tenue MTw Awinda Xsens tracker de mouvement composée de sangles incluant des centrales inertielles permettant de relever au cours du temps de manière précise la pose de l'utilisateur segment par segment. Les sujets devaient estimer la position angulaire de certaines articulations données en fermant les yeux pour ne considérer que les sensations articulaires. Le sys-

tème Awinda fournissait en temps réel la position angulaire de chacun des segments des membres supérieurs du corps qui permettait ainsi de faire un comparatif entre la position angulaire perçue et la position réelle.

Le système de relevé de la cinématique 3D humaine MTw Awinda Xsens est une unité de mesure inertielle miniature et sans fil composée d'accéléromètres, gyroscopes, magnétomètres et baromètre. Ces unités possèdent des processeurs intégrés qui gèrent la synchronisation, l'échantillonnage, l'étalonnage et la communication sans fil via le protocole IEEE 802.15.4 PHY avec la station ou le Dongle du système. Le logiciel MT du système estime la pose 3D de l'utilisateur en temps réel grâce à un filtre de Kalman breveté de l'entreprise Xsens. Ce logiciel permet la gestion de 20 unités inertielles en simultané et le maintien de l'estimation de la pose même en cas de perte de données temporaire. Awinda échantillonne les données à 1Khz dans les unités avant de les transmettre de manière synchronisée à la fréquence souhaitée à la station (jusqu'à 120 Hz). L'ensemble des capteurs permettent une capture précise du mouvement des segments en conservant les contraintes des liaisons du corps humain et une estimation de l'angle de chacune des articulations. Une fois calibré, le système garantit une précision d'estimation d'angle articulaire de 0.5° et 0.75° pour les situations respectivement statique et dynamique. Lors de nos expérimentations, seuls les incertitudes liées à l'estimation d'angle articulaire de la nuque, les épaules et des coudes ont été étudiées. La figure 7.19 représente la combinaison du système MTw Awinda Xsens composée des unités inertielles permettant l'estimation de la cinématique 3D du sujet et le système de coordonnées utilisé pour chacun des segments de notre étude.
Rappelons que les rotations autours des axes X, Y et Z sont respectivement le roulis, le tangage et le lacet. Les rotations de tangage au niveau des coudes n'ont pas été considérées au cours de notre étude par souci de simplification et en raison de la faible amplitude des mouvements de cette articulation pour cette rotation.
Ces expérimentations ont été menées sur 14 sujets en suivant le protocole décrit ci-après.

- Équipements : Les sujets étaient équipés d'une tenue MTw Awinda Xsens composée de 18 centrales inertielles garantissant l'enregistrement de sa cinématique 3D en temps réel.
- Sujets : Un panel de 14 sujets (6 femmes et 8 hommes) en bonne santé et ne présentant aucun handicap pouvant altérer le bon déroulement des tests ont effectué les expérimentations.
- Procédure : Les sujets ont été équipés des combinaisons MTw Awinda Xsens puis ont suivi le protocole de calibration du système Xsens permettant la bonne synchronisation de chacune des centrales inertielles

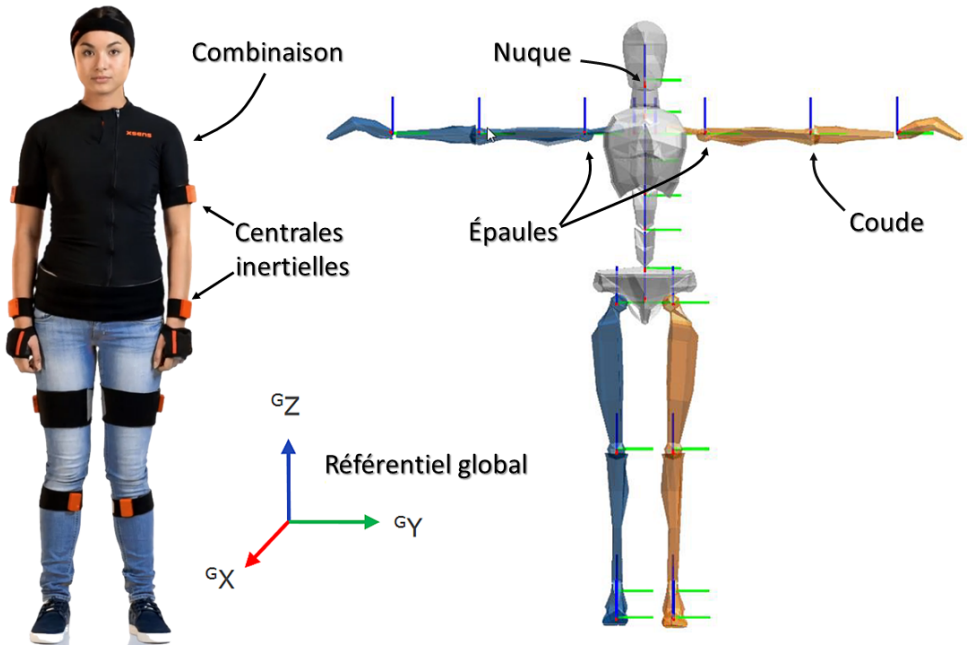


Figure 7.19 – Systèmes de combinaison MTw Awinda Xsens (à gauche) et de coordonnées de chaque segment (à droite).

pour une bonne récupération de la cinématique 3D. Une fois la calibration effectuée, les expérimentations d'estimation de l'incertitude de perception articulaire ont été menées de la manière suivante pour chacune des articulations :

- Une étape d'étalonnage pour que le sujet détermine son point d'origine (position angulaire à $0°$ selon le sujet) pour chacune des articulations de la tête. Ce point devait être atteint 10 fois avant de commencer l'expérimentation avec une erreur de $1°$ autorisée afin de garantir la fiabilité du repère. Ainsi, les mouvements articulaires effectués par la suite étaient tous relatifs à ce point d'origine subjectif.
- Le déplacement angulaire selon les différents axes (roulis, tangage et lacet) pour des positions données. L'angle effectué et mesuré était ensuite comparé à l'angle de consigne pour déterminer l'incertitude de perception articulaire. Cette opération était effectuée 10 fois pour des angles différents afin d'obtenir une moyenne représentative de l'individu pour chacun des axes des articulations étudiée. Des retours à la position d'origine étaient demandés par moment pour vérifier l'absence de dérive du repère d'origine du sujet. Dans le cas échéant, un processus de réétalonnage était effectué.

- À noter que pour les mouvements des épaules et coudes seuls des angles de $45°$ et $90°$ étaient demandés du fait de leur facilité de représentation pour l'humain. Pour la tête, l'amplitude de mouvement angulaire était de $[-30°; 30°]$.

Les expérimentations menées ont permis de déterminer l'incertitude moyenne de perception de position angulaire (en degré) de chaque articulation de la partie haute du corps. Le tableau 7.5 synthétise l'incertitude pour chacune des articulations en intégrant également l'écart-type moyen. De manière globale, l'humain estime relativement bien la position angulaire de chacune de ses articulations avec une incertitude $2.8deg$ en moyenne après une étape de calibration. Ces incertitudes de perception pourront être intégrées par la suite dans un modèle d'intervalle d'incertitude des données capteurs articulaires du robot NAO pour intégrer une composante humaine à la perception articulaire.

Table 7.5 – Incertitude de perception articulaire.

Segment	Tête			Épaules		Coudes
	Roulis	Tangage	Lacet	Roulis	Lacet	Lacet
Moyenne	2.364°	2.829°	2.807°	2.962°	3.162°	3.767°
Écart-type	0.785°	0.649°	0.830°	0.587°	0.980°	0.721°

7.4 . Framework pour l'évaluation de l'immersion

L'intégration du robot NAO dans notre interface de simulation de réalité mixte d'interaction multimodale a conduit à prendre en considération des problématiques liées à la robotique humanoïde. Dans un premier temps, une homothétie sensorielle a été réalisée pour permettre une incorporation des fonctions cognitives humaines dans notre robot NAO. Cette homothétie a été rendue possible par l'association jointe de capteurs et d'algorithmes cognitifs décrits par la suite. Une description du robot NAO est également fournie en annexe pour les caractéristiques techniques et framework de programmation afin d'appréhender au mieux les avantages et limitations du robot NAO.

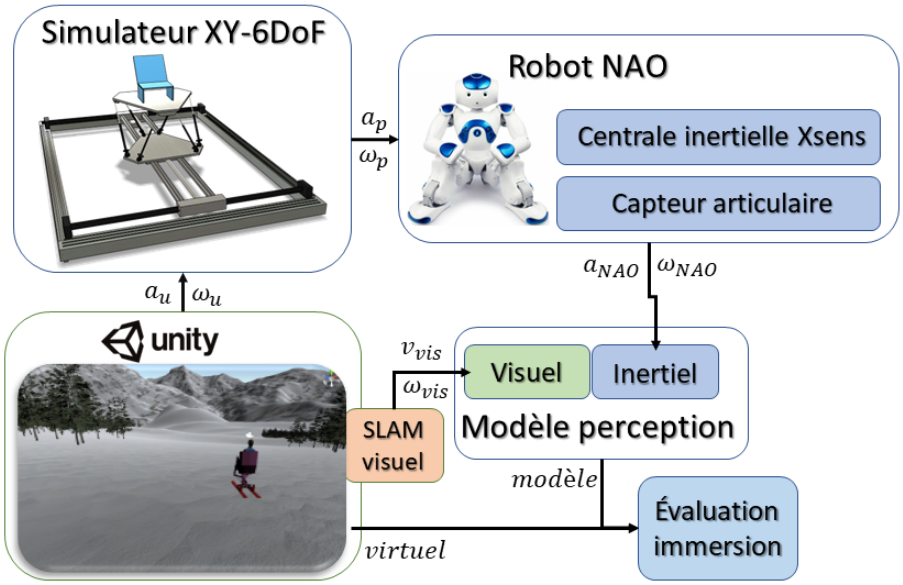


Figure 7.20 – Architecture du système

Le cadre proposé, illustré sur la figure.7.20, intègre les capacités de perception du mouvement propre de l'humain dans un système robotique complexe et multimodal. Ce cadre propose un système multisensoriel reposant sur une homothétie des organes sensoriels humains permettant de reproduire la capacité humaine à percevoir son propre mouvement. Le système vestibulaire est remplacé par une centrale inertielle, tandis que le système visuel est substitué par un algorithme de SLAM visuel. L'intégration des méthodes de SLAM visuel dépend du type d'environnement visuel (avec ou sans objets en mouvement).

Ainsi, pour les environnements à faible scène dynamique, l'ORB -SLAM3 [Cam+20] est une solution efficace, tandis que pour les environnements à scène contenant des dynamiques élevées, les méthodes DynaSLAM [Bes+18] ou D2SLAM [BMB22b] offrent de meilleures performances. Le mouvement du

simulateur XY-6DoF résulte de la trajectoire virtuelle Unity observée par les informations du capteur inertiel issues du mouvement résultant du robot NAO.

Le modèle géométrique direct du robot NAO et les données de ses capteurs articulaires sont combinés pour estimer la pose actuelle de NAO pendant la simulation. Cette estimation permet d'ajuster la position de la caméra du scénario Unity hors ligne pour faire correspondre les champs de vision virtuel et réel. De plus, cette estimation de la pose du corps fournit une solution efficace pour prédire l'impact du mouvement de la plateforme sur le corps du futur patient afin de prévenir les blessures.

Pour simplifier le démonstrateur, nous avons rendu immobile tout le corps de NAO. Un corps sans rigidité signifie que la posture doit être contrôlée pour maintenir une vision cohérente de l'environnement. La tête du robot NAO et les poses de la caméra Unity sont alors harmonisées pour une interaction inter-système cohérente.

7.4.1 . Homothétie sensorielle

L'intégration d'un module de perception du mouvement propre dans un système robotique tentant d'imiter l'aptitude humaine à effectuer cette fonction cognitive a soulevé de nombreuses problématiques. La principale problématique a été de déterminer quelles homothéties sensorielles étaient à opérer lors de la substitution de l'humain par le robot NAO. La fonction cognitive humaine de perception visuo-spatiale repose à la fois sur des organes sensoriels dédiés à l'acquisition d'informations extérieures et sur des traitements plus ou moins complexes de ces informations. Pour la perception du mouvement propre (fonction cognitive visuo-spatiale), trois principaux sens sont requis :

- La vue pour les informations visuelles
- Le système vestibulaire pour les informations inertielles
- La proprioception pour les informations corporelles

Comme détaillée précédemment, la perception visuelle est rendue possible grâce à l'acquisition de stimuli visuels par l'œil humain qui sont ensuite traités par le cortex visuel. L'œil humain induit une sensation de mouvement, appelée *vection*, provoquée par l'observation de l'environnement en mouvement (humain en mouvement dans un décors fixe, l'inverse ou les deux). Cette sensation de mouvement est exploitée dans les simulateurs pour induire un mouvement imaginaire de l'utilisateur en faisant défiler l'environnement virtuel qu'il observe. Il existe deux types de vection en fonction du mouvement de l'environnement : la vection linéaire pour les mouvements de translation et la vection circulaire pour les rotations.

Dans notre cas, nous avons remplacé l'œil humain par un système de caméra qui est le capteur homothétique de celui-ci, et le cortex visuel qui traite l'information visuelle par un algorithme de SLAM qui reprend les tâches du système magno-cellulaire. Ce système estime le mouvement par la perception du flux visuel et la détection de repère (coin, bord, etc...) comme le fait le SLAM visuel.

La perception du mouvement à travers les informations inertielles se fait par le système vestibulaire qui perçoit les mouvements linéaires à l'aide des otolithes et circulaire avec les canaux semi-circulaires (CSC). Ce processus est beaucoup moins complexe et plus rapide que celui de la vision, mais d'un poids inférieur dans la perception globale du mouvement propre.

Le capteur homothétique du système vestibulaire est une centrale inertielle qui renseigne de la même manière sur les accélérations linéaires et les vitesses angulaires. Les modèles mathématiques des otolithes et canaux semi-circulaires permettent de transformer les accélérations linéaires et vitesses angulaires en sensations inertielles correspondants aux aptitudes humaines.

Enfin, les sensations proprioceptives au niveau des articulations peuvent être reproduites par homothétie en exploitant les capteurs articulaires présents dans le robot NAO. Ces informations permettent de déduire la verticale subjective, aussi appelée vecteur idiotropique, d'après l'inclinaison corporelle et les positions angulaires de chacune des articulations. Ce vecteur idiotropique renseigne sur la perception de l'attitude, autrement dit de la gravité, en étant combiné avec les perceptions de la gravité par les systèmes de vision (détection de l'horizon, des lignes de l'environnement) et vestibulaire à travers l'extraction de la gravité.

Au final, l'homothétie sensorielle présente lors de la substitution de l'humain par NAO peut être résumée par type de perception, organes/capteurs et tâche cognitive comme dans le tableau 7.6.

Table 7.6 – Homothétie sensorielle.

Perception	Acquisition		Analyse	
	Organe	Capteur	Organe	Modèle / Algorithme
Vision	Oeil	Caméra	Cortex visuel	SLAM visuel
Inertielle	Vestibulaire	Centrale inertielle	Cervelet	Modèle Otolithe et CSC

Les modèles et algorithmes d'analyse reproduisent par homothétie les capacités humaines de perception du mouvement à travers les deux formes de stimuli de notre simulateur, à savoir, l'accélération pour la perception inertielle et la vitesse pour la perception visuelle. L'algorithme de SLAM visuel renseignera sur les vitesses linéaire et angulaire perçues à travers l'analyse du scénario de réalité virtuel de ski tandis que les modèles mathématiques du système vestibulaire renseigneront sur les accélérations linéaire et angulaire réelles perçues par la centrale inertielle lors des mouvements de la plateforme hybride.

Le modèle des otolithes fournit les accélérations linéaires notées $a_i = (a_{i_x}, a_{i_y}, a_{i_z})^T$ tandis que le modèle des canaux semi-circulaires fournit les vitesses angulaires notées $\omega_i = (\omega_{i_x}, \omega_{i_y}, \omega_{i_z})^T$. Pour la partie vision, l'algorithme de SLAM visuel fournit les vitesses linéaires notées $v_v = (v_{v_x}, v_{v_y}, v_{v_z})^T$ et les vitesses angulaires notées $\omega_v = (\omega_{v_x}, \omega_{v_y}, \omega_{v_z})^T$ relative respectivement à la vection linéaire et circulaire.

7.4.2 . Architecture du framework

Le cadre de la perception homothétique humaine se compose de quatre parties distinctes, comme le montre la Figure.7.21. Tout d'abord, un module scénario virtuel est développé à l'aide du moteur de jeu multiplateforme Unity. Il fournit les informations de flux visuelles nécessaires à la perception visuelle du mouvement le tout à une fréquence d'images de 50 Hz requise par l'algorithme cognitif (SLAM visuel) pour traiter les informations visuelles. La deuxième partie utilise les trajectoires du scénario virtuel comme instructions de mouvement pour le système de plateforme XY-6DoF.

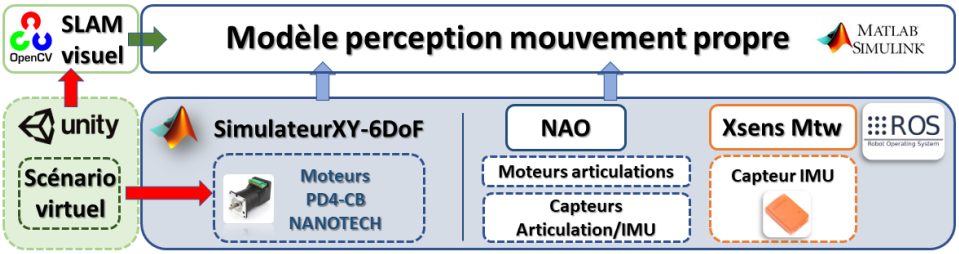


Figure 7.21 – Architecture du framework : niveaux matériel et logiciel.

Le niveau matériel de la plateforme XY-6DoF est contrôlé par six moteurs PD4-CB via MATLAB, permettant la mise en mouvement de la plateforme à l'aide d'une interface de communication basée sur une carte PCI National Instrument. La mise en mouvement de la plateforme impliquera un mouvement de robot NAO qui est observé à travers ses capteurs internes et la centrale inertielle supplémentaire Xsens Mtw. L'acquisition des données des capteurs et la communication avec les actionneurs du robot humanoïde sont réalisées

via ROS (Robot Operating System) qui synchronise les systèmes NAO et Xsens à la fréquence de fonctionnement la plus petite des composants (50Hz).

Ensuite, la troisième partie consiste en l'estimation du mouvement visuel via un algorithme de SLAM visuel utilisant la bibliothèque OpenCV Computer Vision pour effectuer des tâches de vision par ordinateur de haut niveau. Il estime la pose de la caméra virtuelle (position et orientation) qui est ensuite dérivée pour obtenir une estimation des vitesses visuelles linéaire et angulaire. La dernière partie est l'implémentation du modèle de perception du mouvement propre humain dans Simulink, qui prend en entrée la vitesse linéaire et angulaire visuelle du SLAM visuel et l'accélération linéaire et la vitesse angulaire inertielles du système multi-robot. Il fournit une reproduction de la perception du mouvement en fonction du comportement humain. L'évaluation de l'immersion est alors obtenue en calculant la différence absolue e_p entre le mouvement perçu $p = (p_x, p_y, p_z)$ et le mouvement réel $p_g = (p_{gx}, p_{gy}, p_{gz})$ exprimée comme ci-après :

$$e_p(t) = \|p(t) - p_g(t)\| \tag{7.25}$$

Cette différence nous renseigne sur la qualité de la restauration sensorielle pour parvenir à une optimisation future.

7.4.3 . Robot NAO / Centrale inertielle Xsens

La synchronisation des données capteur du robot NAO et de la centrale inertielle sur la tête du robot est rendue possible par l'utilisation de ROS (Robot Operating System). Une architecture middleware (intergicielle) sous ROS permet la gestion et la récupération des données capteurs interne au robot NAO et externe utilisées pour les simulations à travers le framework.
Notre système est décomposé en deux *packages* pour la gestion du robot nao et de la centrale inertielle Xsens, respectivement *nao_data*, *nao_sync* et *xsens_mtw_driver* :

```
/
├── dossier: catkin_nao
│   ├── package: nao_data
│   │   └── Fichier de lancement: set_nao_data.launch
│   │       └── Script python:publish_nao.py
│   └── package: nao_sync
│       └── Fichier de lancement: get_nao_sync.launch
│           └── Script python:get_nao_xsens_data_sync.py
└── dossier: catkin_xsens
    └── package: xsens_mtw_driver
        └── Script c++: mt_w_main.c++
```

L'architecture de notre système sous ROS décrite dans la figure 7.22 est composée de 3 noeuds (nodes) permettant la publication ou récupération des différentes données capteurs. Le noeud *mt_w_driver* est dédié à l'envoi des données capteurs du capteur Xsens dans le système ROS à travers un type de message prédéfini *sensor_msg/IMU* pour la gestion des informations inertielles.

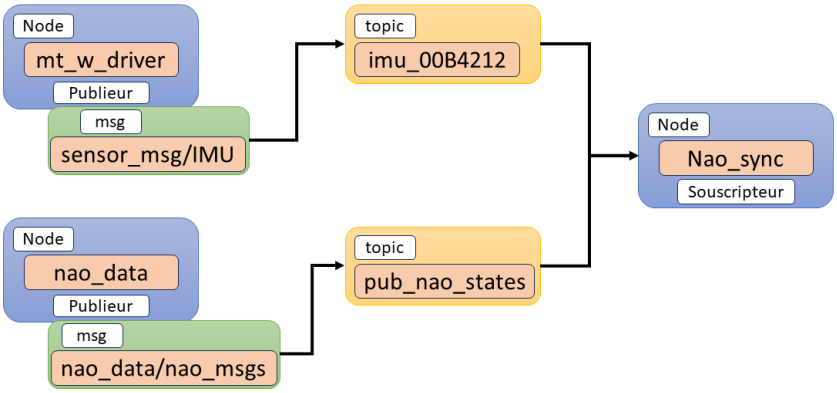


Figure 7.22 – Architecture du système sous ROS.

Ces messages sont envoyés à la fréquence de fonctionnement de la centrale, à savoir 100Hz. Ce message renseigne sur l'orientation à travers un quaternion, et sur la vitesse angulaire et l'accélération linéaire à travers des vecteurs de dimension 3. Le topic *imu_00B4212* correspond alors à l'identifiant de la centrale inertielle Xsens utilisée.

Le noeud *nao_data* est dédié à l'acquisition et l'envoi des données capteurs provenant du robot NAO à une fréquence de 50Hz. Un type de message a été défini à cet effet, *nao_data/nao_msgs* pour contenir de manière structurée l'ensemble des données capteurs à exploiter. Ce message contient l'ensemble des positions angulaires des membres supérieurs du robot ainsi que ses données inertielles issues de sa propre centrale inertielle.

Notons que ces informations sont exploitées à l'aide du framework NAOqi garantissant un accès aux ressources du robot NAO. Nous nous servons des fonctions d'accès aux données en mémoire du robot pour récupérer de manière synchronisée sous ROS les informations capteurs. Un relevé de la séquence et du marqueur temporel d'acquisition garantissent la détermination du moment d'acquisition dans le domaine temporel.

Les capteurs du robot NAO et ceux de la centrale inertielle Xsens étant de fréquence de fonctionnement différente, une synchronisation sous ROS des données capteurs a été effectuée en ajustant la fréquence du capteur à la plus haute fréquence (centrale inertielle) sur celle des capteurs à plus basse fréquence (capteurs NAO).

La fonction *ApproximateTimeSynchronizer* de la bibliothèque *message_filters* permet la concordance des données sur une échelle temporelle. Elle prend en entrée les messages des deux publieurs et fait concorder les données pour un écart maximal donné comme illustré dans la figure 7.23.

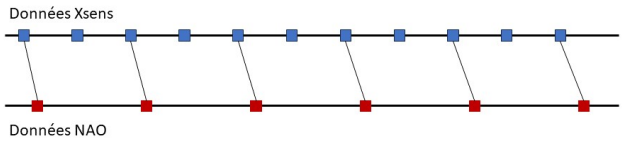


Figure 7.23 – Synchronisation des données NAO et Xsens.

L'idée étant d'associer les données de capteurs de fréquence différente ayant le moment d'acquisition le plus proche. Finalement, l'ensemble des données récupérées sont enregistrées dans un fichier exploité pour la simulation sous Simulink pour évaluer la perception du mouvement propre selon le critère de l'équation 7.25.

7.4.4 . Modèle de perception

Le modèle de perception est implémenté directement sur Simulink en fonctionnement hors ligne. Le système prend en entrée les vitesses linéaire et angulaire issues du SLAM visuel, et l'accélération linéaire et vitesse angulaire provenant de la centrale inertielle Xsens pour le modèle de perception. Un modèle 3D du robot NAO est également intégré à travers Simscapes qui permet un rendu visuel de la simulation (voir figure 7.24).

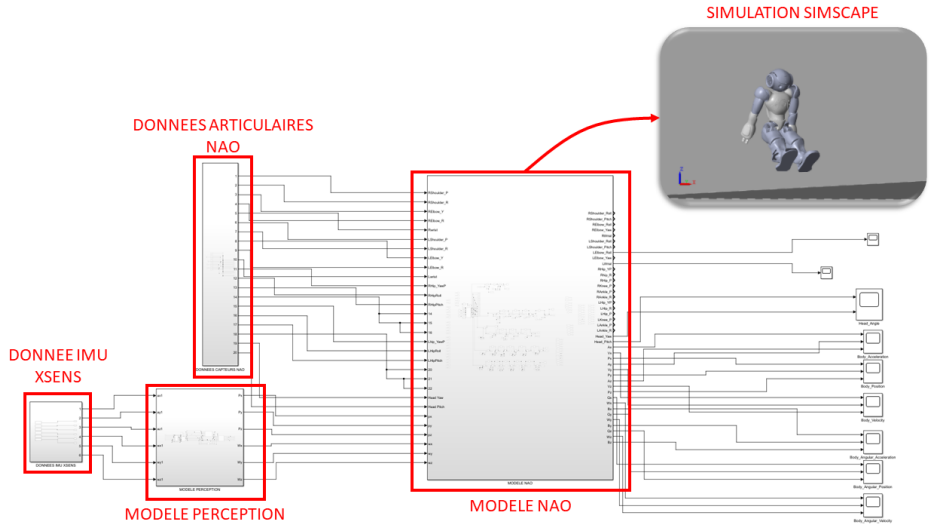


Figure 7.24 – Intégration du modèle de perception dans Simulink.

Le modèle de perception du mouvement propre est composé de deux sous-systèmes réalisant le traitement des informations d'accélération linéaire et de vitesse angulaire correspondant au modèle TNO. Notons qu'un pré-traitement des données d'accélération de la centrale inertielle Xsens est fait pour décomposer l'accélération. Pour cela, une décomposition de la force spécifique est opérée pour distinguer l'accélération subie de la gravité terrestre.
La figure 7.25 illustre l'implémentation des fonctions de transfert de la perception des mouvements linéaire et circulaire.

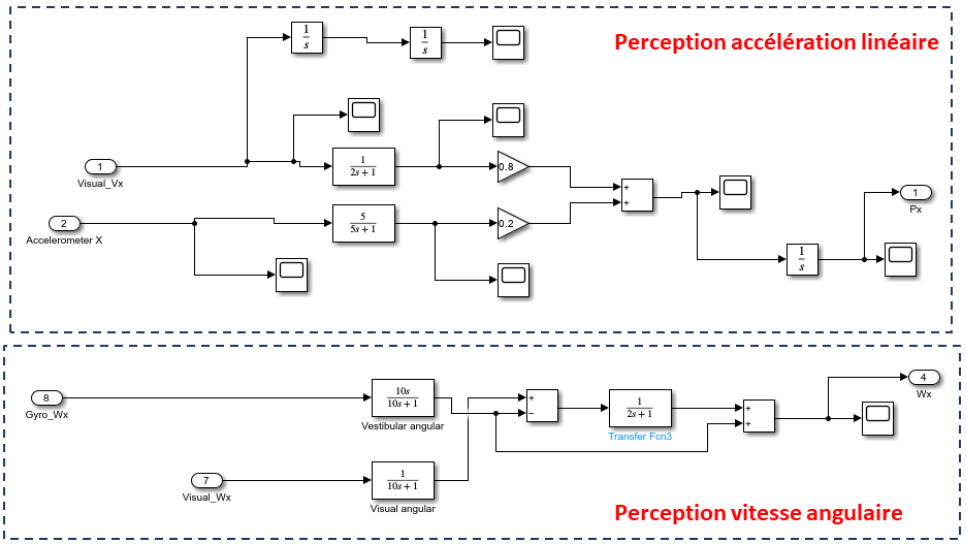


Figure 7.25 – Modèle de perception des mouvements linéaire et circulaire.

Ces informations inertielles sont ensuite associées aux informations visuelles correspondant à la vection sous la forme d'une sommation de ces sensations perçues. Il en résulte une vitesse perçue convertie en position perçue par simple intégration de celle-ci.

Le modèle du robot NAO est implémenté sous SIMULINK grâce au logiciel Simscape intégré dans MATLAB. Nous recomposons l'ensemble de la structure de NAO comme illustré dans la figure 7.26, où chaque articulation du robot est représentée par une rotule aux contraintes mécaniques correspondant à l'articulation réelle. Chaque segment du robot NAO est modélisé en 3D par CAO puis intégré dans notre modèle Simulink pour fournir le support visuel du robot. Le système prend alors en entrée les données provenant des capteurs articulaires du robot NAO réel afin de restituer dans l'interface visuel les mouvements effectués par le robot lors de tests.

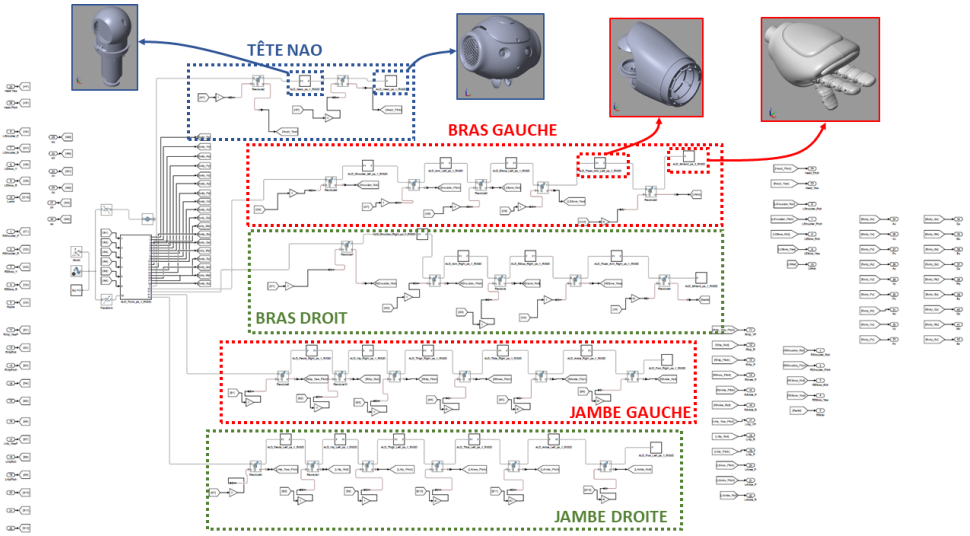


Figure 7.26 – Modèle du système NAO dans SIMULINK.

7.5 . Optimisation de l'immersion

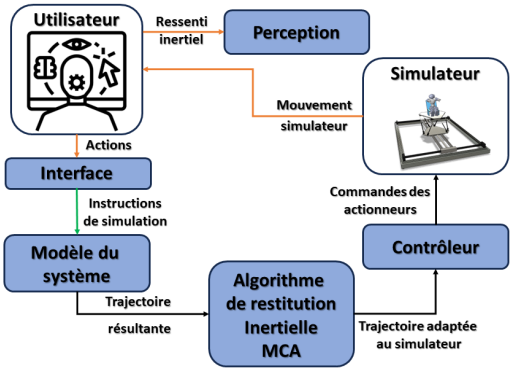
Les simulateurs sont des plateformes robotiques servant à la restitution de sensations inertielles dues à un mouvement défini par un scénario permettant l'immersion de l'utilisateur dans un contexte particulier. La structure robotique des simulateurs induit des limites d'un point de vue systémique. En effet, l'objectif d'un simulateur est de reproduire les trajectoires, ou du moins les sensations, d'un scénario défini relatif à un environnement vaste. Cependant, l'espace de travail limité des simulateurs ne permet donc pas de reproduire des trajectoires de mouvement issues d'environnements infinis ce qui pose la question de leur déploiement.

Dès lors, des travaux ont été menés pour lever cette limitation à l'aide d'algorithme de restitution inertielle tel que le *Motion Cueing Algorithm* (MCA). Partant de ce constat, plusieurs problématiques apparaissent :

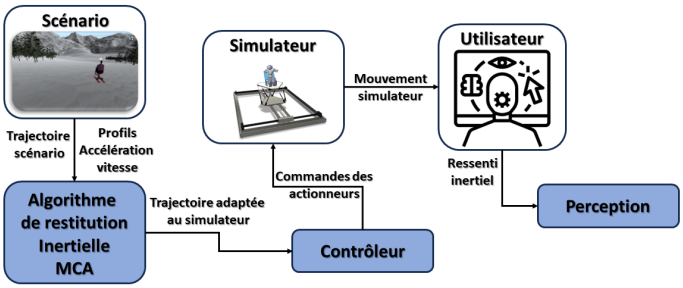
- Par quel procédé peut-on restituer les sensations inertielles d'un mouvement d'amplitude supérieur à la zone de travail du simulateur ?
- Quels sont les algorithmes de restitution inertielle existants ?
- Existe-t-il des leviers pour optimiser cette restitution inertielle ?

À l'heure actuelle, aucune solution ne permet une restitution parfaite des sensations inertielles par des simulateurs. Les approches existantes permettent d'améliorer l'immersion, mais sont limitées par les performances des algorithmes et par la modélisation et la compréhension du processus de perception. L'organe sensoriel, à savoir le système vestibulaire, est sensible aux

mouvements linéaires à travers l'accélération pour les otolithes et aux mouvements circulaires via la vitesse angulaire pour les canaux semi-circulaires. La pertinence de ces grandeurs justifie leur utilisation pour caractériser les sensations inertielles des trajectoires à restituer à travers ces algorithmes.



(a) Système de simulation active



(b) Système de simulation passive

Figure 7.27 – Architecture des systèmes de simulation.

De fait, un système de simulation intègre un ensemble de modules permettant la bonne restitution des sensations inertielles provenant de trajectoires définies par le simulateur. En fonction de la nature active ou passive de l'utilisateur du simulateur, des modules supplémentaires peuvent être incorporés. La figure 7.27 illustre les différents liens existants entre les modules du système de simulation. Rappelons que dans notre cas, l'utilisateur du simulateur est passif car soumis à la trajectoire prédéfinie du scénario à laquelle il doit adapter sa posture afin de rééduquer ses muscles et ses automatismes intervenants pour le maintien postural (voir figure 7.27b). L'algorithme de restitution inertielle prend alors le rôle d'un outil de transformation des trajectoires initialement appliqué au simulateur en des profils d'accélération réalisable par le simulateur tout en conservant au mieux les sensations inertielles.

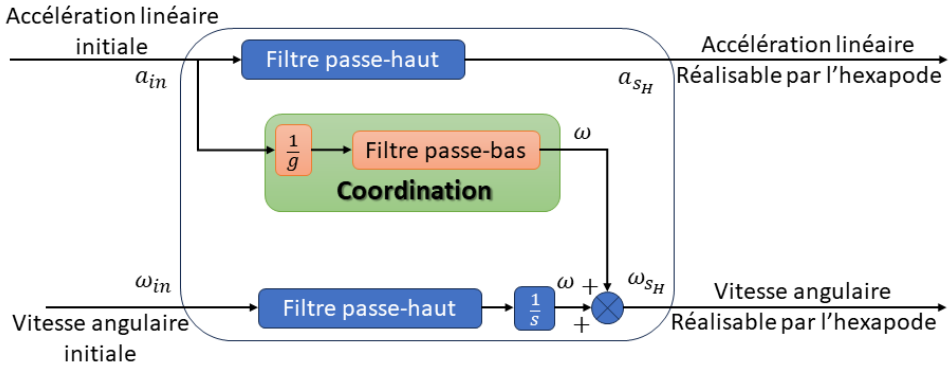
7.5.1 . *Motion Cueing Algorithm*

Le problème de restitution des sensations inertielles dans les simulateurs a été étudié au cours des dernières décennies et en partie résolu par l'algorithme du MCA. Le MCA transforme les trajectoires initiales imposées au si-

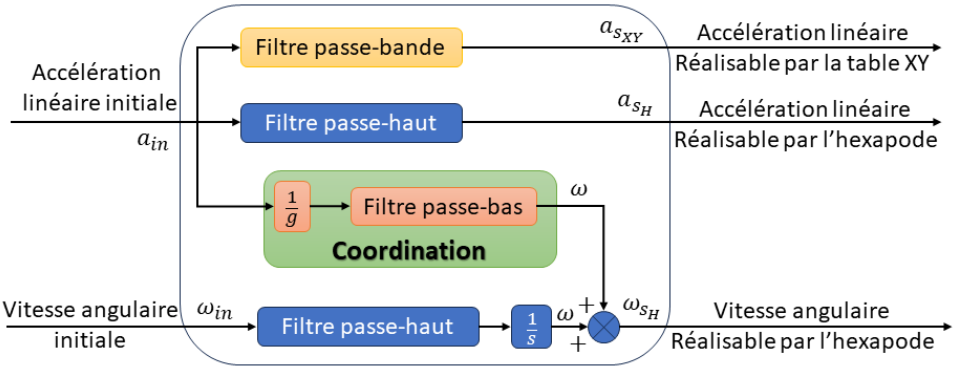
mulateur en trajectoires qui soient réalisables et suffisamment immersives. Pour cela, l'algorithme décompose la trajectoire initiale en plusieurs trajectoires spécifiques à chacune des parties de la plateforme de simulation en fonction de sa configuration :

- Plateforme de Gough-Stewart : décomposition du mouvement selon des mouvements de rotation et de translation de la nacelle haute (hexapode).
- Plateforme de Gough-Stewart hybride : décomposition du mouvement selon des mouvements de rotation et de translation de la nacelle haute, et de translation de la table XY.

De manière générale, le MCA est basé sur la séparation fréquentielle de l'accélération linéaire et de la vitesse angulaire, afin d'exploiter au mieux le processus de perception du mouvement propre humain intervenant via le système vestibulaire (voir figure 7.28).



(a) MCA pour la plateforme de Gough-Stewart classique



(b) MCA pour la plateforme de Gough-Stewart hybride

Figure 7.28 – Principe du *Motion Cueing Algorithm*.

Les principales approches de cet algorithme peuvent être résumées dans la liste suivante :

- L'approche classique [CS70] utilise des coefficients constants pour l'ensemble des filtres afin de séparer les contenus fréquentiels de l'accélération et de la vitesse angulaire. Cette approche non-optimale incorpore la coordination pour faire coïncider les mouvements de rotation et translation.
- L'approche adaptative [Par+75] est une amélioration du MCA classique qui minimise en temps réel une fonction de coût traduisant le meilleur compromis entre le respect des contraintes mécaniques et la meilleure restitution des sensations inertielle. Cependant, cette approche détermine les coefficients de chacun des filtres pour une trajectoire définie et spécifique.
- L'approche optimale [SIH82] est une généralisation de l'approche adaptative pour une famille de trajectoires. À l'instar de l'approche adaptative, cette approche prend en considération les seuils de perception humains dans le calcul des coefficients du filtre.
- L'approche prédictive [Dag+04 ; Dag05] prend en considération les contraintes et intègre un superviseur en charge du retour de la plateforme en position neutre. Cette approche permet alors de maximiser la restitution inertielle pour une trajectoire donnée.
- L'approche basée sur le PSO (*Particle Swarm Optimisation*) [Cas+18] est une métaheuristique d'optimisation permettant de trouver une solution optimale pour chacun des coefficients du filtre du MCA. Ces coefficients optimaux permettent l'optimisation de la restitution inertielle d'une trajectoire donnée.

Nous décrirons les approches classiques et optimales du MCA qui sont respectivement les approches détaillant le principe de base du MCA et l'approche permettant à ce jour la meilleure optimisation de la restitution inertielle. Pour la suite de l'étude des différentes approches, nous utiliserons les notations suivantes pour décrire le mouvement du simulateur avec l'indice s et le mouvement de la trajectoire réelle via l'indice r :

- x, y et z : respectivement les variables de translations longitudinale, transversale et verticale.
- ϕ, θ et ψ : respectivement les variables de rotation de roulis, tangage et lacet.
- g correspond à la gravité terrestre et vaut 9.81 m.s^{-2}.
- υ, a et ω : respectivement les variables de vitesse linéaire, accélération linéaire et vitesse angulaire dans le cas de mouvement unidirectionnel.

7.5.2 . Approche classique du MCA

L'approche classique du MCA introduite par Shmidt [CS70] tente de surmonter les contraintes liées à l'espace de travail limité des simulateurs en effectuant une séparation fréquentielle des trajectoires réelles. Le MCA exploite les capacités liées à la structure de la plateforme de Gough-Stewart en répartissant les mouvements lents (ou de basse fréquence) et les mouvements rapides (ou de haute fréquence) entre des mouvements de rotation et de translation de l'hexapode. En effet, pour des trajectoires linéaires réelles, la plateforme n'est capable que de reproduire les fortes accélérations à court terme (haute fréquence) à travers des mouvements de translation. À l'inverse, la coordination permet de reproduire la composante basse fréquence des accélérations en exploitant les failles du système de perception vestibulaire humain. Dans le cas d'un mouvement rectiligne longitudinal, le MCA générera un mouvement de translation et de rotation (tangage apparenté à du *tilt*) qui s'exprime comme suit :

$$\begin{aligned} x_s &= (\tfrac{1}{s^2} F_{H1}) a_r \\ \theta_s &= (\tfrac{1}{s} F_{H2}) \omega_r + (\tfrac{1}{g} F_B) a_r \end{aligned} \tag{7.26}$$

Les variables F_{H1}, F_{H2} et F_B font référence aux filtres illustrés dans la figure 7.29 où F_H et F_B correspondent respectivement à des filtres passe-haut et passe-bas.

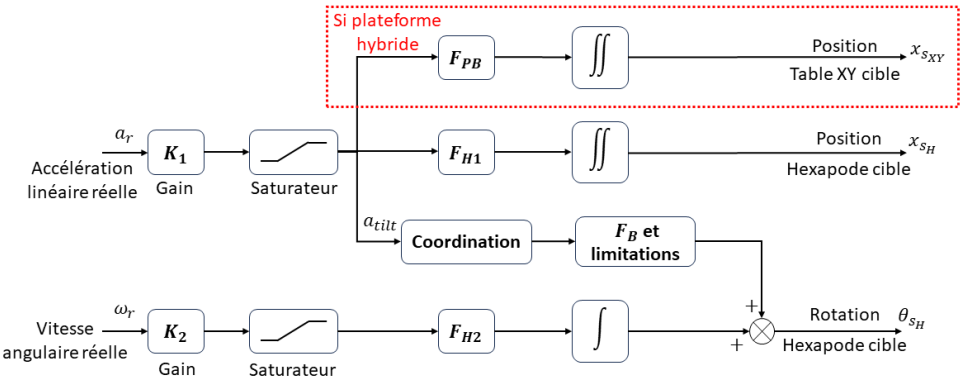


Figure 7.29 – MCA classique pour un mouvement longitudinal appliqué à une plateforme de Gough-Stewart (normal ou hybride).

L'algorithme du MCA est illustré dans la figure 7.29 pour des configurations de la plateforme normale et hybride. En nous basant sur les travaux de Nehaoua [Neh+06] sur la modélisation de filtre passe-haut appliqué au MCA, l'élimination des composantes basse fréquences est rendue possible par l'utilisation

d'un filtre passe-haut du troisième ordre F_{h_i} combinant un filtre passe-haut normal et un filtre *washout* comme suit :

$$F_{h_i} = \frac{a_s(s)}{a_r(s)} = K_1 \underbrace{(\frac{s^2}{s^2 + 2\zeta_i\omega_{1_i}s + \omega_{1_i}^2})}_{\text{Filtre PH}} \underbrace{(\frac{s}{s + \omega_{2_i}})}_{\text{Washout}} \tag{7.27}$$

Le filtre passe-bas F_B permettant l'extraction de la composante basse fréquence de l'accélération linéaire est formulé sous la forme d'un filtre passe-bas du second ordre comme ci-après :

$$F_B = K_2(\frac{1}{s^2 + 2\zeta_i\omega_1 s + \omega_1^2}) \tag{7.28}$$

Où K_1, K_2, ω_{1_i}, ω_1, ω_{2_i} et ζ_i correspondent aux paramètres des $i^{ieme} = 1, 2$ filtres passe-haut et du filtre passe-bas à ajuster pour optimiser la restitution inertielle. Les paramètres K sont les gains des filtres, ζ est le coefficient d'amortissement qui permet d'atténuer la composante négative de l'accélération cible, ω_1 les pulsations permettant d'ajuster le niveau de restitution inertielle et la prise en compte des contraintes mécaniques, et ω_2 est la pulsation du filtre *washout* permettant le retour à une position neutre de la plateforme en ajustant la vitesse de retour tout en considérant les seuils de perception inertielle humains. La figure 7.30 illustre la restitution de l'accélération provenant de la trajectoire réelle à l'aide de l'hexapode (plateforme de Gough-Stewart).

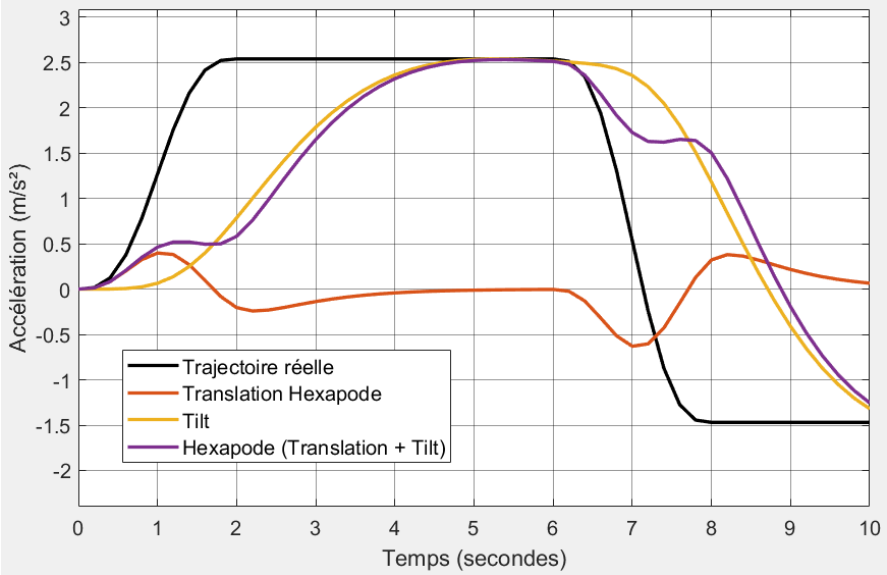


Figure 7.30 – Restitution de la trajectoire réelle via l'hexapode.

Dans le cas d'une plateforme de Gough-Stewart hybride, une partie de la composante de l'accélération linéaire est reproduite à travers la table XY. Pour ce faire, Houda [Hou20] propose un filtre passe-bande F_{PB} du second-ordre qui extrait la composante moyenne fréquence de l'accélération linéaire correspondant au déplacement plus long. Ce filtre passe-bande est formulé de la manière suivante :

$$F_{PB} = K_1\left(\frac{H_0 \frac{f_0}{Q} s}{s^2 + H_0 \frac{f_0}{Q} s + f_0^2}\right) \tag{7.29}$$

Où H_0, f_0 et Q sont respectivement le gain du filtre, la pulsation de résonance correspondant au gain maximal et au facteur de qualité.

En reprenant la trajectoire réelle précédente et les mêmes paramètres du MCA, nous pouvons démontrer l'apport de l'intégration de la table XY pour l'optimisation de la restitution. Comme le montre la figure 7.31, nous remarquons que l'intégration de la table XY dans l'utilisation de la plateforme de Gough-Stewart permet de décomposer davantage la composante de l'accélération linéaire conduisant à une meilleure restitution des profils d'accélération. La table XY permet la retranscription des dynamiques de moyenne fréquence qui ne sont ni considérées par la translation ni par la rotation de l'hexapode.

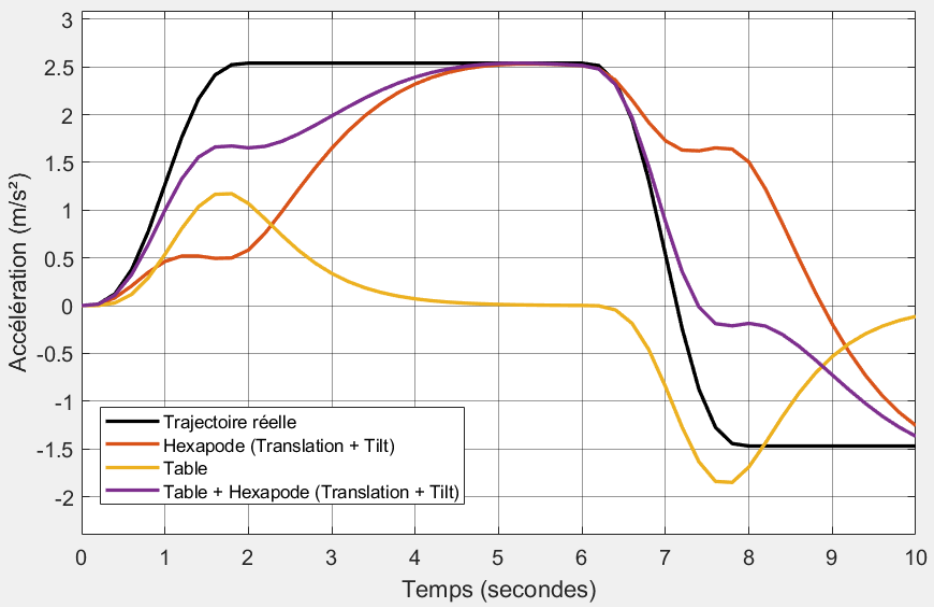


Figure 7.31 – Restitution de la trajectoire réelle via l'hexapode et la table XY.

La figure 7.32 montre la restitution des forces spécifiques en fonction de la configuration du simulateur (Hexapode ou Hexapode + table).

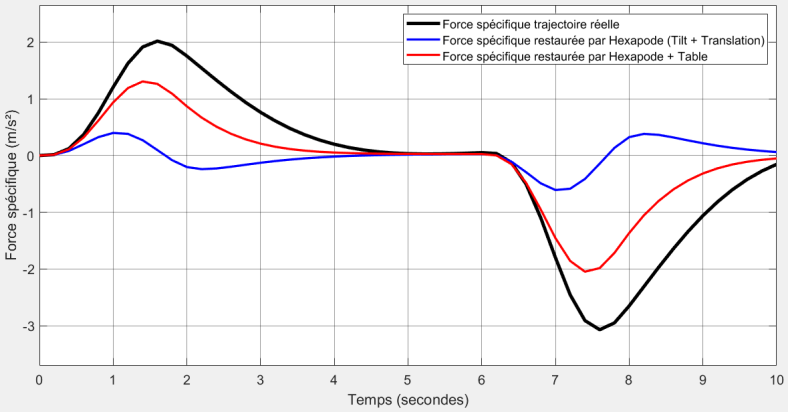


Figure 7.32 – Force spécifique restituée par le simulateur.

En raisonnant selon l'aspect restitution des sensations inertielles, la force spécifique partiellement restaurée à l'aide MCA est assez fidèle à celle provenant des trajectoires réelles comme illustré dans la figure 7.32.

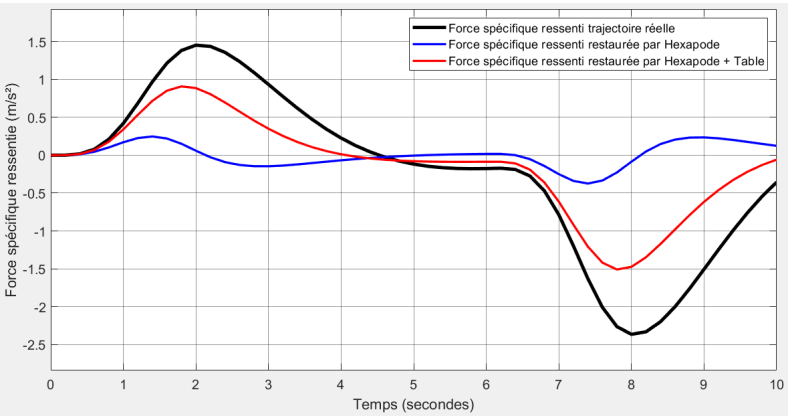


Figure 7.33 – Ressenti de la force spécifique restituée par le simulateur.

L'erreur de perception inertielle obtenue via la fonction de transfert du système vestibulaire illustrée dans la figure 7.34 montre l'apport de l'intégration de la table XY dans la restitution des sensations inertielles.
Cette étude montre l'intérêt d'avoir incorporé une table XY à notre plateforme de Gough-Stewart pour améliorer la restitution des sensations inertielles en plus de l'espace de travail.

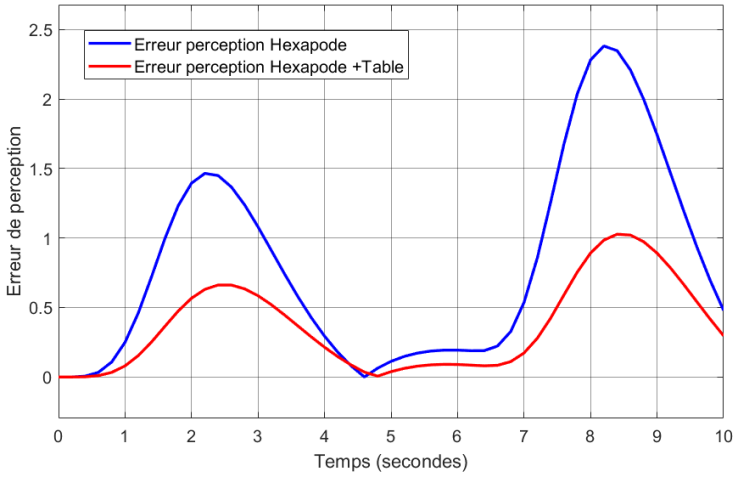


Figure 7.34 – Restitution de la trajectoire réelle via l'hexapode et la table XY.

7.5.3 . Approche optimale du MCA

L'approche optimale du MCA a été introduite par Sivan [SIH82] en 1982 avec de nombreuses implémentations dans des simulateurs dont le SIMONA par Telbian et Cardullo [Tel+00 ; TC01]. Cette approche tente d'obtenir le meilleur compromis entre la restitution des sensations inertielles et les contraintes mécaniques du simulateur en calcul hors ligne un filtre optimal H_O de restitution inertiel. La qualité de la restitution inertielle est définie par l'erreur de perception notée e représentant la différence entre les sensations inertielles induites par un filtre de perception inertielle H_V appliqué à la trajectoire réelle et celle du simulateur comme illustré dans la figure 7.35.

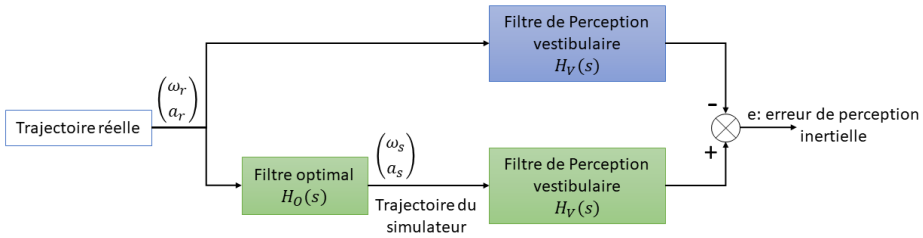


Figure 7.35 – Architecture de l'algorithme de MCA optimal.

La différence de perception inertielle correspondant à l'erreur de perception e est exprimée par :

$$e(s) = H_V(s)(H_O(s)\begin{pmatrix}\omega_r\\a_r\end{pmatrix}) - H_V(s)\begin{pmatrix}\omega_r\\a_r\end{pmatrix} = H_V(s)\begin{pmatrix}\omega_s\\a_s\end{pmatrix} - H_V(s)\begin{pmatrix}\omega_r\\a_r\end{pmatrix} \tag{7.30}$$

Où les vitesses angulaires ω_s et ω_r, et les accélérations linéaires a_s et a_r sont respectivement les états d'entrée du simulateur et de la trajectoire réelle. Le

filtre de perception inertielle H_V lié au système vestibulaire est défini de la manière suivante :

$$H_V = \begin{pmatrix} H_{CSC} & 0 \\ \frac{1}{gs} H_{oto} & H_{CSC} \end{pmatrix} \tag{7.31}$$

Ce filtre H_V est obtenu par la combinaison des modèles des organes du système vestibulaire, à savoir les canaux semi-circulaires notés H_{CSC} et les otolithes notés H_{oto}. L'équation 7.30 peut ensuite être réécrite comme suit :

$$\begin{aligned} \dot{\chi_\nu} &= A_\nu \chi_\nu + B_\nu \begin{pmatrix} \omega_s - \omega_r \\ a_s - a_r \end{pmatrix} \\ e &= C_\nu \chi_v + D_\nu \begin{pmatrix} \omega_s - \omega_r \\ a_s - a_r \end{pmatrix} \end{aligned} \tag{7.32}$$

Où χ_ν est l'erreur d'état du système vestibulaire. Dans le but de contraindre le mouvement du simulateur à revenir à sa position d'équilibre et rester dans sa zone de travail, la fonction de coût χ_d caractérisant le déplacement du simulateur doit incorporer l'intégrale du déplacement, la vitesse de translation, le déplacement et la position angulaire. Ainsi, le simulateur est caractérisé par les deux termes supplémentaires suivants dans son espace d'état :

$$\begin{aligned} \chi_d &= [\int \int \int a_s dt^3 \ \int \int a_s dt^2 \ \int a_s dt \ \omega_s]^T \\ \dot{\chi_d} &= A_d \chi_d + B_d \begin{pmatrix} \omega_s \\ a_s \end{pmatrix} \end{aligned} \tag{7.33}$$

Où les états χ_d (le vecteur de déplacement) et $\dot{\chi_d}$ sont liés aux états d'entrée du simulateur ω_s et a_s à travers les matrices suivantes :

$$A_d = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \ B_d = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{7.34}$$

Les deux critères de compromis du MCA, à savoir la restitution inertielle et les contraintes mécaniques du simulateur, sont établis par les équations 7.32 et 7.33 à travers les variables d'état e, $\int x$, x, ν et θ respectivement l'erreur de sensation inertielle, la distance parcourue par rapport à la position neutre, la position, la vitesse et l'angle de rotation. Le MCA optimal est une approche hors ligne qui requiert donc une connaissance préalable de la trajectoire réelle.

Cependant, dans le cas d'une simulation active (voir figure 7.27a) où l'utilisateur est maître des trajectoires réelles via l'envoi d'instructions de simulation en temps réel, les trajectoires réelles ne peuvent être connues à l'avance et donc optimisées. Pour surmonter cette limitation, Sivan [SIH82] propose d'utiliser une entrée perturbée par un bruit blanc filtré comme trajectoire réelle pour le MCA. Cette solution permet de diversifier les trajectoires réelles

prise en compte par le MCA en incorporant une variable d'état supplémentaire χ_n :

$$\begin{aligned} \dot{\chi_n} &= A_n\chi_n + B_n d \\ \chi_n &= \begin{pmatrix} \omega_r \\ a_r \end{pmatrix} \end{aligned} \tag{7.35}$$

Où χ_n représente les états de bruit blanc filtrés et d le bruit blanc. Les matrices A_n et B_n sont définies par les fréquences de coupure f_1 et f_2 par :

$$A_n = -\begin{pmatrix} f_1 & 0 \\ 0 & f_2 \end{pmatrix} \; et \; B_n = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} \tag{7.36}$$

Les trois équations d'état peuvent être combinées pour former l'équation du système global comme suit :

$$\begin{aligned} \dot{\chi} &= A\chi + B\begin{pmatrix} \omega_s \\ a_s \end{pmatrix} + Nd \\ y &= C\chi + D\begin{pmatrix} \omega_s \\ a_s \end{pmatrix} \end{aligned} \tag{7.37}$$

Avec $\chi = [\chi_\nu \; \chi_d \; \chi_n]^T$ représentant l'ensemble des états du système et $y = [e \; \chi_d]^T$ la sortie désirée du MCA. Les matrices de l'équation du système global 7.37 sont données par :

$$A = \begin{pmatrix} A_\nu & 0 & -B_\nu \\ 0 & A_d & 0 \\ 0 & 0 & A_n \end{pmatrix} \; , \; B = \begin{pmatrix} B_\nu \\ B_d \\ 0 \end{pmatrix} \; , \; N = \begin{pmatrix} 0 \\ 0 \\ B_n \end{pmatrix} \; , \; C = \begin{pmatrix} C_\nu & 0 & -D_\nu \\ 0 & I & 0 \end{pmatrix} \; et \; D = \begin{pmatrix} D_\nu \\ 0 \end{pmatrix} \tag{7.38}$$

L'approche du MCA optimal tente d'obtenir les paramètres optimaux en lien avec les états du mouvement du simulateur et de la perception inertielle à travers la résolution de l'équation de Riccati. Le compromis optimal entre l'erreur de perception inertielle minimale e et les contraintes de déplacement χ_d est obtenu par la minimisation de la fonction de coût J. La fonction de coût J est minimisée à l'aide du LQR pour l'intervalle de simulation $[t_0 \; t_1]$ de la manière suivante :

$$J = E\left\{ \int_{t_1}^{t_0} (e^T Q e + \chi_d^T R_d \chi_d + \begin{pmatrix} \omega_s \\ a_s \end{pmatrix}^T R \begin{pmatrix} \omega_s \\ a_s \end{pmatrix}) dt \right\} \tag{7.39}$$

Cette fonction de coût considère le coût de l'erreur de perception à travers le terme $e^T Q e$ et le coût lié au déplacement du simulateur via $\chi_d^T R_d \chi_d$ et $\begin{pmatrix} \omega_s \\ a_s \end{pmatrix}^T R \begin{pmatrix} \omega_s \\ a_s \end{pmatrix}$. Notons que minimiser J consiste à minimiser l'espérance E de l'ensemble des coûts des variables d'état de sortie du MCA (e et χ_d). Les matrice Q et R_d et R sont des matrices de pondération.

L'équation 7.39 a donc un double objectif, minimiser l'erreur de sensation inertielle entre le mouvement de la trajectoire réelle et celui reproduit par le simulateur, et minimiser le déplacement de la plateforme afin de satisfaire

ses contraintes mécaniques. L'équation du système globale 7.37 et la fonction de coût 7.39 peuvent être reformulées avec la variable d'état du système global χ de la manière suivante :

$$\begin{aligned} \dot{\chi} &= A_t\chi + B_t u_t N d \\ J &= E\left\{ \int_{t_1}^{t_0} (\chi^T R_1 \chi + u^T R_2 u) dt \right\} \end{aligned} \tag{7.40}$$

Où les termes de la fonction 7.40 s'expriment par :

$$\begin{aligned} A_t &= A - BR_2^{-1}R_{12}^T \\ u &= \tbinom{\omega_s}{a_s} + R_2^{-1}R_{12}^T\chi \\ R_1 &= C^T GC - R_{12}R_2^{-1}R_{12}^T \;\; avec \;\; G = \begin{pmatrix} Q & 0 \\ 0 R_d & \end{pmatrix} \\ R_2 &= R + D^T GD \;\; et \;\; R_{12} = C^T GD \end{aligned} \tag{7.41}$$

Cette reformulation permet une résolution du problème de la forme $u = -K\chi$ où K déduit des matrices du système global et permet de minimiser la fonction de coût exprimée selon la variable globale χ lorsque u vaut :

$$u = -R_2 B^T P \chi \tag{7.42}$$

Où P est la solution de l'équation de Riccati permettant cette minimisation :

$$R_1 PBR_2^{-1}B^T P + A_t^T P + PA_t = 0 \tag{7.43}$$

Ainsi en combinant l'expression de u de l'équation 7.41 et sa valeur minimale souhaitée de l'équation 7.42, nous pouvons exprimer $\binom{\omega_s}{a_s}$ par :

$$\begin{pmatrix} \omega_s \\ a_s \end{pmatrix} = -R_2^{-1}(B^T P + R_{12}^T)\chi = -K\chi \tag{7.44}$$

Avec K la solution de ce problème de minimisation pouvant s'exprimer par :

$$K = R_2^{-1}(B^T P + R_{12}^T) \tag{7.45}$$

En décomposant la variable globale χ et la matrice K, nous pouvons réécrire la solution optimale précédente 7.44 sous la forme suivante :

$$\begin{pmatrix} \omega_s \\ a_s \end{pmatrix} = -K_1\chi_\nu - K_2\chi_d - K_3\chi_n \tag{7.46}$$

Étant donné que $\chi_n = [\omega_r \; a_r]^T$, l'équation globale du système 7.37 exprimée par les variables d'état $[\chi_\nu \; \chi_d \; \chi_n]$ peut alors s'écrire uniquement selon les variables $[\chi_\nu \; \chi_d]$ donnant ainsi :

$$\begin{pmatrix} \dot{\chi_\nu} \\ \dot{\chi_d} \end{pmatrix} = \begin{pmatrix} A_\nu & 0 & -B_\nu \\ 0 & A_d & 0 \end{pmatrix} \begin{pmatrix} \chi_\nu \\ \chi_d \\ u_r \end{pmatrix} + \begin{pmatrix} B_\nu \\ B_d \end{pmatrix} u_s \tag{7.47}$$

Où $u_r = [\omega_r \; a_r]^T$ et $u_s = [\omega_s \; a_s]^T$. En reprenant la formulation de u_s de l'équation 7.46 en fonction de K, l'équation précédente peut s'écrire :

$$\begin{aligned} \dot{\chi_\nu} &= A_\nu \chi_\nu - B_\nu u_r + B_\nu(-K_1\chi_\nu - K_2\chi_d - K_3 u_r) \\ \dot{\chi_d} &= A_d \chi_d + B_d(-K_1\chi_\nu - K_2\chi_d - K_3 u_r) \end{aligned} \tag{7.48}$$

Soit en réécrivant cette équation en fonction des variables d'état définissant l'entrée du MCA et ses critère de perception et de déplacement, nous obtenons :

$$\begin{pmatrix} \dot{\chi_\nu} \\ \dot{\chi_d} \end{pmatrix} = \begin{pmatrix} A_\nu - B_\nu K_1 & -B_\nu K_2 \\ -B_d K_1 & A_d - K_2 B_d \end{pmatrix} \begin{pmatrix} \chi_\nu \\ \chi_d \end{pmatrix} + \begin{pmatrix} -B_\nu(I + K_3) \\ B_d K_3 \end{pmatrix} u_r \tag{7.49}$$

Ainsi le filtre optimal H_O liant la trajectoire du simulateur à celle réelle, correspond au passage de ces équations dans le domaine fréquentiel de la manière suivante :

$$u_s = \begin{pmatrix} \omega_s \\ a_s \end{pmatrix} = H_O(s) \begin{pmatrix} \omega_r \\ a_r \end{pmatrix} = H_O(s) u_r = H_O(s) \chi_n \tag{7.50}$$

Où le filtre optimal H_O s'exprime :

$$H_O(s) = \begin{pmatrix} K1 & K2 \end{pmatrix} \begin{pmatrix} sI - A_\nu + B_\nu K_1 & B_\nu K_2 \\ B_d K_1 & sI - A_d + K_2 B_d \end{pmatrix}^1 \begin{pmatrix} -B_\nu \\ B_d \end{pmatrix} - K_3 \tag{7.51}$$

7.5.4 . Méthode proposée : MCA via apprentissage par renforcement

De nos jours, l'apprentissage par renforcement (AR) est de plus en plus utilisé en robotique pour des applications de contrôle de robot ou de prise de décision [PN17 ; Luo+19 ; Cor+20]. Sutton et Barton [SB18] ont introduit le concept d'apprentissage par renforcement via l'expérience et l'étude du comportement de système soumis à des prises de décision. De nombreux travaux [KBP13 ; Lil+15 ; Kai+19 ; DMH19 ; Yar+21] ont démontré l'efficacité de l'AR pour résoudre des problèmes dans divers domaines, dont la robotique. Partant de ce constat, nous proposons un MCA basé sur l'apprentissage par renforcement (AR) afin de décomposer un profil d'accélération linéaire souhaité en deux mouvements linéaires au niveau de la table et de l'hexapode, et un mouvement angulaire au niveau de l'hexapode. À ce jour, quelques travaux [Moh+16 ; Ren+18 ; Koy+20 ; Qaz+20] traitent du MCA à travers l'apprentissage mais uniquement sous la forme de modèle prédictif de contrôle (MPC). Seul Scheidel [Sch+23] traite la problématique du MCA dans le cadre de simulateur à l'aide de l'apprentissage par renforcement. Nous nous inspirerons de cette approche pour concevoir notre méthode tout en l'adaptant à la configuration de notre plateforme de Gough-Stewart hybride.

L'apprentissage par renforcement est une approche d'apprentissage profond basée sur un processus de décision Markovien permettant à un agent d'apprendre de l'interaction qu'il a avec un environnement modélisé au cours d'un entraînement basé sur l'expérimentation. L'interaction existante entre l'agent et l'environnement est décrite par un vecteur d'état E et un vecteur d'action A pour chacun des temps relevés. L'action A_t de l'agent sur l'environnement entraîne une observation des états E_t des sorties de l'environnement représentant la cause à effet entre ces deux vecteurs en vue d'une formulation du système sous la forme d'un problème de décision. La qualité de l'impact de l'action sur l'environnement par rapport aux objectifs souhaités est formulée sous la forme d'une fonction de récompense R_t basée sur les observations d'état E_t.

L'agent gagne en expérience au cours de l'entraînement grâce à l'observation du comportement du système en fonction de ses actions. Le processus de décision Markovien est traduit alors par la probabilité de transition $\Gamma(E_{t+1}|E_t, A_t)\ E_t$ représentant la probabilité de l'état E_t au moment t de transiter vers l'état E_{t+1} pour une action donnée A_t. L'algorithme d'AR tente ainsi de mettre en relation les états et actions dans un réseau de neurones afin d'optimiser une stratégie de contrôle $\Pi(A_t|E_t)$ garantissant cette transition. Le comportement du système est alors lié à la distribution de probabilité $p_\theta(\tau)$ reliant stratégie de contrôle et probabilité de transition, et définit par :

$$p_\theta(\tau) = \mu_0(E_0) \prod_{t=1}^{\infty} \Pi_\theta(A_t|E_t)\ \Gamma(E_{t+1}|E_t, A_t) \tag{7.52}$$

Où θ correspond aux paramètres du réseau de neurones et μ_0 la distribution de probabilité de départ de l'état initial E_0. Afin d d'optimiser la stratégie de contrôle Π, l'entraînement de l'agent doit conduire à obtenir la plus grande valeur de récompense cumulée $C(\tau)$, où $C(\tau)$ est définie par :

$$C(\tau) = \sum_{t=0}^{\infty} \gamma^t R_{t+1} \tag{7.53}$$

Ici, γ, aussi appelé *discount factor*, définit la sensibilité de l'agent à la valeur de la récompense au cours du temps. Tout comme Scheidel [Sch+23], nous avons opté pour l'utilisation de la méthode de politique de gradient PPO (*Proximal Policy Optimization*) [Sch+17] qui s'appuie sur une implémentation de type acteur-critique. Pour chaque expérimentation, aussi appelée épisode, les observations des sorties de l'environnement varies en fonction des modifications stochastiques de la politique de contrôle. Cette même politique de contrôle est mise à jour en fonction de la direction du gradient qui dépend des observations faîtes de l'environnement. En nous basant sur l'article [Wil92], nous pouvons exprimer l'objectif de la mise à jour de la politique comme étant de

faire tendre la fonction d'objectif $L(\theta)$ vers la valeur optimale de récompense cumulée comme ci-après :

$$L(\theta) = \mathbb{E}_{\tau \sim p_{theta}(\tau)} \left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \right] = \mathbb{E}_{\tau \sim p_{theta}(\tau)} \left[C(\tau) \right] \tag{7.54}$$

La valeur optimale étant pour une trajectoire donnée τ. Le gradient de cette fonction d'objectif peut être exprimé pour chacun des paramètre θ comme suit :

$$\nabla_\theta L(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[C(\tau) \sum_{t=0}^{\infty} \nabla_\theta log[\pi_\theta(A_t|E_t)] \right] \tag{7.55}$$

L'implémentation acteur-critique consiste en un double réseau de neurones artificiels où le critique est dédié à l'approximation de la valeur attendue afin d'optimiser le gradient de la fonction d'objectif $\nabla_\theta L(\theta)$ par rapport aux récompenses obtenues et la valeur attendue. L'algorithme PPO met à jour la politique de contrôle de l'approximation du critique selon une certaine marge ξ définie par le paramètre η correspondant à l'écart possible avec la politique précédente.

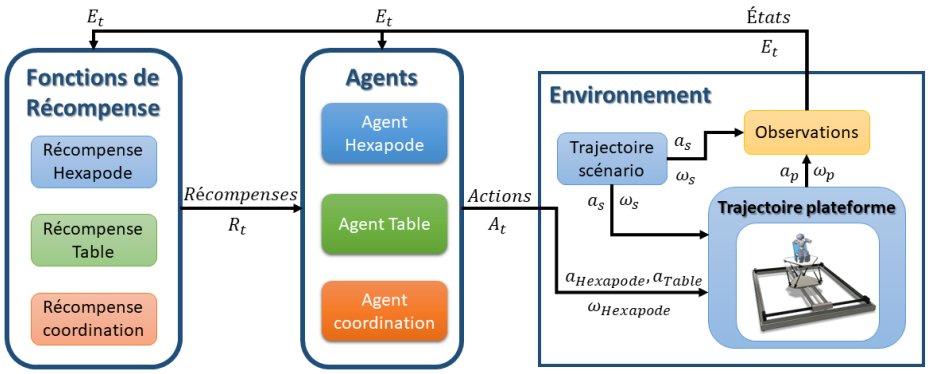


Figure 7.36 – Architecture de l'algorithme de MCA optimal.

Notre système illustré dans la figure 7.36 est composé d'un environnement simulant le comportement du simulateur et d'agents fournissant les actions appliquées à l'environnement. Le vecteur d'action noté A_t fournit la valeur de variation d'action au moment t pour la table et l'hexapode notés respectivement ta et h.

$$A_t = [\Delta a_t^{ta},\ \Delta a_t^{h},\ \Delta \omega_t^{h}]^T \tag{7.56}$$

L'objectif est de déterminer le vecteur d'état A_t permettant de restituer au mieux le profil d'accélération linéaire a_t^s et la vitesse angulaire ω_t^s du simulateur. Pour cela, nous décomposons notre modèle d'apprentissage par renforcement en 3 agents indépendant en charge de fournir chacun un des composants du vecteur d'état. Les profils d'accélérations a_{Table} et $a_{Hexapode}$ proposés

par les agents Table et Hexapode sont combinés au profil d'accélération résultant de la vitesse angulaire $\omega_{Hexapode}$ proposée par l'agent coordination.

L'environnement sert à la fois à simuler des profils d'accélération linéaire et de vitesse angulaire souhaités et à intégrer les actions issues des agents produisant le mouvement restitué. L'environnement fournit en sortie pour chaque pas de temps t un vecteur d'état E_t définit par :

$$E_t = [x_t^r,\ v_t^r,\ a_t^r,\ x_t^{ta},\ x_t^h,\ f_t^r,\ \theta_t^r,\ \omega_t^r,\ a_t^s,\ f_t^s,\ \omega_t^s,\ \Delta a_t^{ta},\ \Delta a_t^h,\ \Delta\omega_t^h]^T \quad (7.57)$$

Où $x_t^r,\ v_t^r$ et a_t^r représentent la position, la vitesse et l'accélération du simulateur, x_t^{ta} et x_t^h la position absolue de la table et l'hexapode, f_t^r la force spécifique restituée par le simulateur, θ_t^r la position angulaire du simulateur et ω_t^r la vitesse angulaire restituée par le simulateur. Les variables issues de la trajectoire de simulation sont définies par $a_t^s,\ f_t^s$ et ω_t^s respectivement pour l'accélération, la force spécifique et la vitesse angulaire souhaitées. Enfin, les actions sont également intégrées au vecteur d'état pour fournir une observation de l'impact .

Ces observations permettent ensuite le calcul des récompenses de chacun des 3 agents permettant d'atteindre les objectifs. Rappelons que modèle à un double objectif :

- Restituer au mieux le profil d'accélération en fournissant en entrée de l'environnement deux profils d'accélérations pour la table et l'hexapode.
- Satisfaire aux limites mécaniques de la plateforme et ainsi restituer au mieux l'accélération avec des amplitudes de mouvements réalisables.

Nos fonctions de récompense R_{i_t} prennent en considérations ces objectifs à travers des fonctions de récompense intermédiaire notées r_{i_n} servant de pénalité pour retranscrire les différents buts. Les fonctions de récompense R_{i_t} de chacun des i agents sont alors définies de la manière suivante :

$$R_t = 2 + r_t - r_a - r_f - r_\omega - r_x - r^i_{\Delta a^h} - r^i_{\Delta a^{ta}} - r^i_{\Delta\omega} - r^i_{lxh} - r^i_{lxta} - r^i_{l\theta} \quad (7.58)$$

Notre fonction de récompense prend deux fonctions de récompense intermédiaires composées d'une constante et d'un critère de temps r_t qui servent à valoriser l'avancement de la simulation dans le temps. La fonction de récompense liée au temps est définie comme suit :

$$r_t = \Delta_t t \quad (7.59)$$

Où $\Delta_t = \frac{10}{Tf}$ un coefficient lié au temps total de simulation Tf permettant d'augmenter la valeur de récompense proportionnellement au temps de simulation afin d'encourager le modèle à aller au bout de l'expérience. L'idée

étant ensuite d'imposer des pénalités à ces récompenses en fonctions de critères liés à la restitution et au mouvement de la plateforme. Un ensemble de fonctions de récompense communes à chacun des agents permettent de formaliser les critères caractérisant le mieux la restitution inertielle à travers une optimisation des paramètres d'entraînement. Le critère principal, à savoir l'écart de restitution d'accélération, est défini par la fonction r_{i_a} comme suit :

$$r_a = W_a|a_t^r - a_t^s| \tag{7.60}$$

La restitution de la force spécifique est caractérisée par la fonction r_f définie par :

$$r_f = W_f|f_t^r - f_t^s| \tag{7.61}$$

Enfin, la différence de vitesse angulaire entre la simulation et le mouvement de la plateforme permet de prendre en considération la composante angulaire présente lors de la coordination. Cette fonction de récompense r_ω est dédiée à la l'incitation de la réalisation de la coordination et définie comme suit :

$$r_\omega = \begin{cases} W_\omega|\omega_t^r - \omega_t^s|, & \text{si } \omega \geq 3 \\ 0, & \text{sinon} \end{cases} \tag{7.62}$$

La fonction de récompense r_x pénalise les déplacements inutiles de la plateforme afin d'inciter les actions à ramener la plateforme en position de confort (x =0) :

$$r_x = W_x|x_t^r| \tag{7.63}$$

Où W_a, W_f, W_ω et W_x sont les poids définissant l'importance de chacun de ces critères.

Les fonctions de récompense liées aux actions permettent de minimiser les écarts de valeurs entre les pas de temps afin d'atténuer les oscillations. Pour ces fonctions, les poids diffèrent d'un agent à l'autre afin de prioriser la minimisation en fonction du composant (table et hexapode). La fonction de récompense $r^i_{\Delta a^{ta}}$ pour les actions d'accélération appliquées à la table est définie comme suit :

$$r_{\Delta a^{ta}} = W^i_{a^{ta}}|\Delta a_t^{ta}| \tag{7.64}$$

Celle pour les actions d'accélération appliquées à l'hexapode $r_{\Delta_i a^h}$ est définie comme suit :

$$r^i_{\Delta a^h} = W^i_{a^h}|\Delta a_t^h| \tag{7.65}$$

Enfin, la fonction de récompense $r^i_{\Delta a^{ta}}$ pour les actions d'accélération appliquées à la table est définie comme suit :

$$r^i_{\Delta\omega} = W^i_\omega|\Delta\omega_t^h| \tag{7.66}$$

Où $W^i_{a^{ta}}$, $W^i_{a^h}$ et W^i_{ω} sont les poids priorisant l'impact des actions des agents i afin d'adapter la fonction à la composante du modèle avec i $\in$ Agent Table, Agent Hexapode, Agent Coordination $\}$.

Les limites mécaniques de la plateforme sont prises en compte via les fonction de récompense r^i_{lxh}, r^i_{lxta} et $r^i_{l\theta}$ destinées à pénaliser l'amplitude du mouvement de translation et de rotation effectuées au niveau de la table et de l'hexapode. La fonction dédiée à la limite de mouvement de translation de l'hexapode r_{lxh} est définie par :

$$r^i_{lxh} = W^i_{lxh} \frac{|x_{t^h}|}{lxh} \tag{7.67}$$

Pour les mouvements de translation de la table, la fonction r_{lxt} est définie comme suit :

$$r^i_{lxta} = W^i_{lxta} \frac{|x^{ta}_t|}{lxta} \tag{7.68}$$

Et celle pour les rotations par :

$$r^i_{l\theta} = W^i_{\theta} \frac{|\theta^r_t|}{l\theta} \tag{7.69}$$

Où lxh, $lxta$ et $l\theta$ sont les limites d'amplitude de mouvement de translation et rotation de l'hexapode et la table, et W^i_{lxh}, W^i_{θ} et W^i_{lxh} les poids assignés. Dans chacune de ces fonctions de récompense, la pénalité augmente proportionnellement au pourcentage d'amplitude de mouvement effectué par le composant par rapport à sa capacité maximale. Enfin, une condition de fin de simulation est ajoutée au modèle afin de stopper l'épisode lorsque l'une des limites d'amplitude est atteinte.

Afin d'inciter l'agent à aller au bout de la simulation, une fonction ponctuelle de récompense r_p est ajoutée pour valoriser la décomposition du profil d'accélération lorsqu'elle ne provoque pas une amplitude de mouvement supérieur ou égale aux limites mécaniques de la plateforme. Cette fonction s'exprime de la manière suivante :

$$r_p = \begin{cases} 10000, & \text{si } t \geq Tf \ \ \& \ \ 0.1 \geq m_a \\ 300 + \frac{300}{m_a}, & \text{si } t \geq Tf \ \ \& \ \ m_a \geq 0.1 \\ 300 - \frac{100}{(1-m_a)+0.1}, & \text{si } t \geq Tf \ \ \& \ \ m_a \leq 1 \ \ \& \ \ m_a > 0.1 \\ -300, & \text{si } t \geq Tf \ \ \& \ \ m_a > 1 \\ 0, & \text{sinon} \end{cases} \tag{7.70}$$

Où t représente le temps, Tf la durée totale de la simulation et m_a l'écart moyen entre l'accélération initiale et celle restituée.

La restitution du profil d'accélération et de la force spécifique pour un profil d'accélération donné sont illustrés dans le figure chapitre suivant.

7.6 . Conclusion

Nous avons proposé un framework intégrant un modèle de perception visuo-vestibulaire dans un système robotique humanoïde dans le but de reproduire le mécanisme de perception du mouvement propre de l'humain. Ce framework implémenté sous Simulink pour l'interprétation du mouvement et en Python pour la communication avec les capteurs du robot NAO fournit un outil d'évaluation de la qualité de l'immersion. Cette évaluation peut ensuite être utilisée comme information dédiée à la calibration de scénarios qui garantissent une bonne restitution des ressentis ainsi que l'intégrité physique de l'utilisateur. En outre, nous avons proposé une méthode de *Motion Cueing Algorithm* basée sur l'apprentissage par renforcement qui permet au simulateur de reproduire les ressentis vestibulaire correspondant à des trajectoires d'amplitudes supérieurs à sa zone de travail. Cette méthode repose sur un modèle d'apprentissage multi-agent qui décompose la trajectoire du scénario en 3 sous-trajectoires destinées au mouvement de translation au niveau de la table et de l'hexapode, et au mouvement de rotation pour l'hexapode. Chacun de ces composants tentant de restituer au mieux le profil d'accélération de la trajectoire du scénario. Notre méthode permet une restitution de X % du profil d'accélération via les différents composants de la plateforme. Cette approche nouvelle appliquée à une plateforme de Gough-Stewart hybride ouvre la voie à l'intégration de modèle d'apprentissage par renforcement pour des systèmes de simulations complexes.

8 - Expérimentation et résultat de simulation

8.1 . Introduction

Le développement de l'industrie automobile et aéronautique au cours des dernières décennies a poussé les industriels à concevoir des plateformes de simulation permettant d'expérimenter certaines innovations dans des environnements maîtrisés. Ces simulateurs de mouvement [Gra+01 ; Che+01 ; Nie+13], largement étudiés dans de nombreux travaux de recherche, possèdent des propriétés et avantages distincts en fonction de leur configuration robotique et du contexte d'application. Dans un premier temps, les travaux et recherches ont été focalisés essentiellement sur la composante robotique de ces simulateurs afin de contourner leurs contraintes de mouvement [GR97 ; HLA04] et optimiser celui-ci pour de meilleures restitutions inertielles [Tel+00 ; RK00 ; GB13]. Par la suite, une composante visuelle a été incorporée aux simulateurs, tout d'abord à travers des écrans intégrés dans les cockpits.

Cette incorporation a enrichi l'immersion en apportant une information supplémentaire à l'utilisateur, ce qui a conduit la communauté à entreprendre des recherches en lien avec la vision par ordinateur pour optimiser cette composante supplémentaire [HCB11 ; TC01]. Récemment, les écrans des simulateurs ont été remplacés par les casques de réalité virtuelle enrichissant ainsi l'immersion. La réalité virtuelle a connu un essor significatif ces dernières années grâce aux investissements massifs dans le domaine par de nombreuses entreprises technologiques. L'intégration progressive des dernières innovations en apprentissage profond dans les outils de développement de réalité virtuelle tels que Unity et Unreal Engine ont permis une amélioration importante du réalisme des environnements créés.

Le simulateur de ski proposé dans notre étude intègre deux composantes distinctes : un scénario virtuel générant l'environnement d'immersion visuelle et les trajectoires de l'utilisateur, et une plateforme robotique de Gough-Stewart, nommée simulateur XY-6DoF, qui reproduira les trajectoires du scénario.

L'environnement de simulation créé devra satisfaire certaines conditions afin de garantir la faisabilité de la simulation par le futur utilisateur et la plateforme robotique. En outre, l'environnement créé et les trajectoires proposées devront atteindre un niveau de réalisme correspondant à une situation réelle de descente de ski. Une description du simulateur, des contraintes liées à la simulation et du scénario de réalité virtuelle est donnée pour établir les propriétés de l'environnement de simulation utilisé pour l'étude de la perception du mouvement.

Enfin l'ensemble des expérimentations seront présentées pour exposer les résultats obtenus pour l'évaluation de l'immersion à travers notre simulateur et pour l'optimisation de l'immersion à l'aide du *Motion Cueing Algorithm* (MCA). Nos expérimentations seront tout d'abord menées en simulation sous Matlab pour évaluer les concepts d'évaluation de l'immersion puis reproduites avec notre plateforme robotique et le scénario virtuel. Pour l'optimisation d'immersion, les expérimentations seront uniquement simulées sous Matlab pour valider les concepts d'optimisation du MCA via apprentissage par renforcement.

8.2 . Simulateur XY-6DoF

8.2.1 . Plateforme hybride

Le simulateur XY-6DoF illustré dans la figure 8.1 et conçu au sein du laboratoire IBISC, sous une configuration hydride, associe une plateforme parallèle de Gough-Stewart pour la partie haute du simulateur et une table XY pour la partie basse, le tout augmentant la zone de travail et les capacités d'accélération. La partie haute de la plateforme, aussi appelée nacelle, est mise en mouvement à l'aide de six vérins hydrauliques à embouts cylindriques raccordés aux rotules de la nacelle et aux cardans de la table XY. La table XY est composée de deux actionneurs permettant les mouvements de translation selon les axes X et Y.

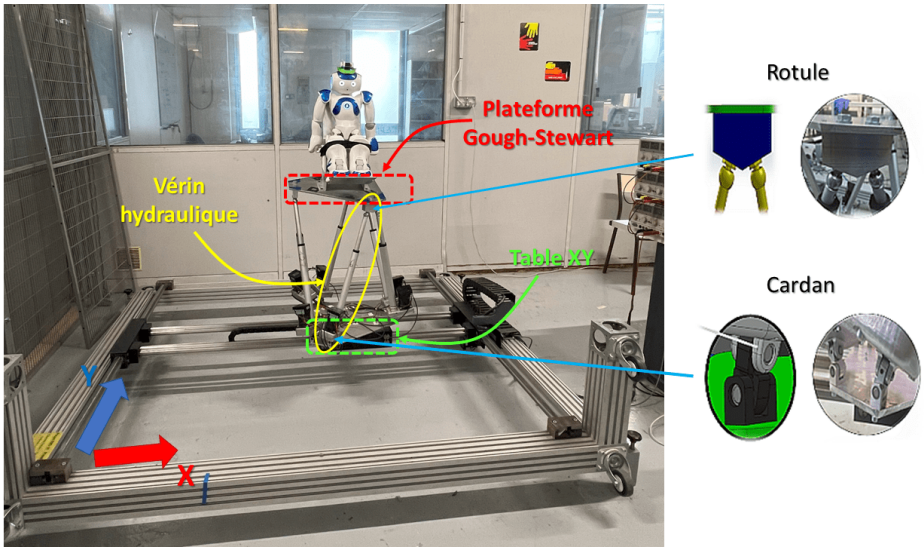


Figure 8.1 – Simulateur XY-6DOF : plateforme hybride.

L'utilisation d'articulations à rotules et cardan permet d'effectuer des translations et rotations de la nacelle offrant ainsi 6 degrés de liberté, la table XY

fournie 2 degrés de liberté supplémentaire augmentant la zone de travail et les capacités d'accélération de la plateforme.

Les actionneurs utilisés pour la mise en mouvement de la table XY et de la nacelle sont des moteurs sans balai PD4-CB du constructeur NANOTEC. Les mouvements de translation longitudinaux (selon l'axe X) et transversaux (selon l'axe Y) de la table XY sont contrôlés par un moteur installé sur chacun des axes. Les six vérins qui raccordent la table XY à la nacelle incorporent chacun un moteur permettant leur mise en mouvement. Ces six moteurs permettent ainsi d'effectuer des mouvements de translation (X_p, Y_p, Z_p) et de rotation (roulis θ_x , tangage θ_y et lacet θ_z) de la nacelle offrant un comportement plus diversifié à la plateforme. Au total, 8 moteurs permettent le contrôle du comportement de la plateforme accordant ainsi 8 degrés de liberté au système. Le tableau 8.1 récapitule les caractéristiques de la plateforme en termes d'amplitude de mouvement, de vitesse et d'accélération.

Table 8.1 – Caractéristiques mécaniques de la plateforme.

Axes	Amplitude	Vitesse maximale	Accélération maximale
X	$\pm 0.55m$	1.2 m/s	$1.25m/s^2$
Y	$\pm 0.575m$	1.2 m/s	$0.71m/s^2$
X_p, Y_p	$\pm 0.6m$	0.8 m/s	$1.1m/s^2$
Z_p	$\pm 0.6m$	0.8 m/s	$1.1m/s^2$
θ_x, θ_y	$\pm 15deg$	48.7 deg/s	$71.6deg/s^2$
θ_z	$\pm 15deg$	106 deg/s	$154deg/s^2$

8.2.2 . Architecture du système

Le contrôle du comportement de la plateforme est assuré par le pilotage des 8 moteurs du système. Les moteurs PD4-CB, présentés dans la figure 8.2, ont la particularité d'intégrer directement dans leur boîtier les parties mécaniques pour la génération des mouvements et électroniques pour le contrôle des moteurs. Ces moteurs intègrent également un encodeur magnétique de position absolue renseignant sur l'incrémentation angulaire du moteur ensuite convertie en position dans le repère plateforme.

L'encodeur dispose d'une précision d'incrémentation correspondant à un déplacement angulaire θ de $0.18°$. En considérant le rayon r de l'arbre du moteur de taille 4 mm, nous pouvons calculer le déplacement d de la plateforme correspondant à un déplacement angulaire θ de la plateforme de la manière suivante :

$$d = sin(\theta) \times r = sin(0.18) \times 4 \cdot 10^{-3} = 1.26 \cdot 10^{-5} m \tag{8.1}$$

Une révolution du moteur équivaut à 2000 incrémentations. De ce fait, nous pouvons déterminer un rapport entre position incrémentale P_i du moteur et position plateforme P_p pour la table XY en mètre : $P_p = P_i \cdot 10^{-3}$.
Le rapport entre position incrémentale P_i du moteur et position plateforme

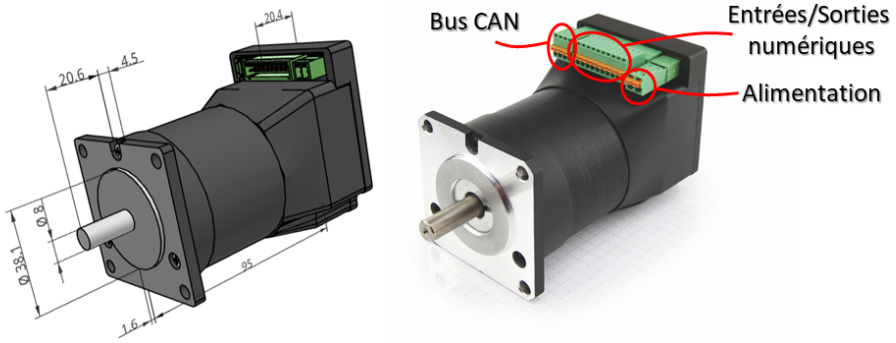


Figure 8.2 – Moteurs PD4-CB sans balai.

P_p de la nacelle est différent en raison de la nature de la transmission du mouvement. En effet, pour la table XY, le mouvement des moteurs est transmis à la plateforme basse grâce à des courroies tandis que les mouvements sont transmis à la nacelle par des vérins. Pour la nacelle, le ratio moteur/plateforme est alors : $P_p = 0.25P_i \cdot 10^{-3}$.

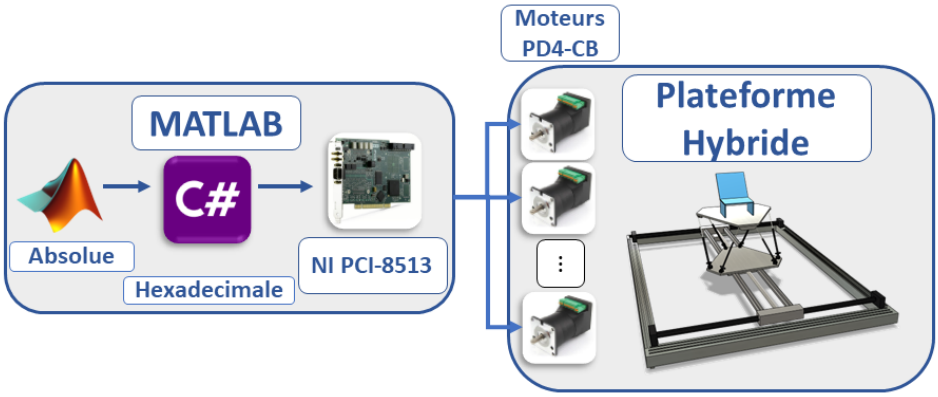


Figure 8.3 – Architecture système de la plateforme hybride.

Les moteurs PD4-CB sont fournis avec les pilotes adaptés et un kit de développement logiciel (SDK) en $C\#$ permettant l'envoi d'instructions aux moteurs et la réception d'informations capteurs. Cet kit est à l'origine adapté pour les communications en bus CANopen avec un appareil USB-to-CAN qui cependant ne permet pas d'atteindre une vitesse de communication suffisante. En

conséquence, la mise en communication entre l'unité de calcul et les moteurs du système a été rendue possible grâce l'utilisation d'une carte PCI 8513 du fabricant National Instrument. La carte PCI est une interface CAN FD (Controller Area Network Flexible Data Rate) monofilaire permettant l'émission et réception à haute vitesse de signaux et trames CAN (jusqu'à 1 MBd). Cette carte offre l'avantage d'être configurable et utilisable sous Matlab, pour l'envoi et la réception de trames CAN. Il a donc été nécessaire d'interfacer Matlab avec les programmes en $C\#$ mis à disposition par NANOTEC pour contrôler les moteurs via la carte PCI 8513. Sous Matlab, la librairie .NET permet d'interagir avec d'autres applications .NET, dont notre kit de développement en $C\#$, depuis Matlab. Dans notre cas, nos programmes Matlab font appel à la classe *controlLib* définie dans le programme $C\#$ pour utiliser les fonctions de contrôle des moteurs. La figure 8.3 illustre l'architecture logicielle et matérielle de notre plateforme avec les différents liens entre les supports logiciels.

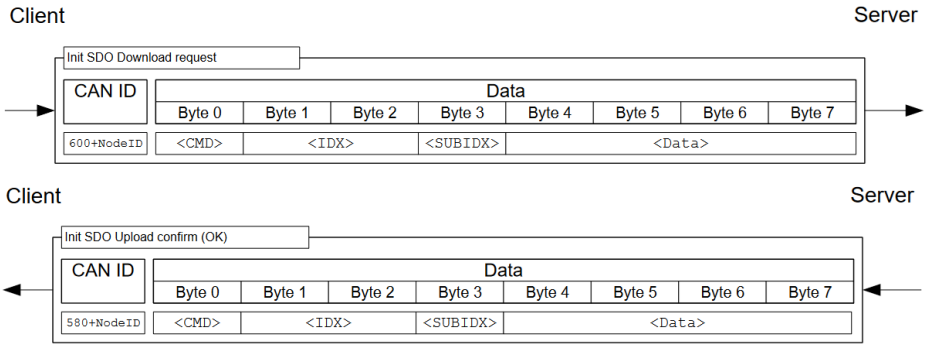


Figure 8.4 – Communication via le service SDO (émission en haut et réception en bas).

La communication avec les moteurs se fait via le protocole CANopen, une couche applicative de la communication en bus CAN, qui repose sur des dictionnaires d'objet décrivant la fonctionnalité d'un appareil de manière standardisée. L'ensemble de ces objets sont définis par un index principal *IDX* codé en 16 bits et un sous-index *SUBIDX* en 8 bits. Le protocole CANopen repose sur trois services de communication : NMT, SDO et PDO. Ces services permettent l'exécution de tâches indexées dans un registre de dictionnaire d'objet qui sont orientées vers l'appareil souhaité par l'indicatif *NodeID* qui dans notre système indique le moteur à contrôler.

- NMT (*Network Management*) : il définit l'état du bus (pré-opérationnel, opérationnel, stoppé, etc.) à l'aide de 2 octets. Le premier pour définir l'état via une commande *CMD* et le second pour définir l'appareil concerné.

- SDO (*Service Data Object*) : il permet d'accéder aux informations d'un nœud spécifique du réseau en indiquant l'objet souhaité par les index et sous-index contenus dans le dictionnaire d'objet. Les trames SDO sont codées sur 8 octets de manière spécifique pour l'envoi et la réception de données.
- PDO (*Process Data Object*) : il ne contient que les données à transmettre grâce à un système d'auto-configuration en mode émission ou réception de données de 8 octets.

En particulier, le service SDO permet d'indiquer le mode de fonctionnement de moteur à l'aide du premier octet *CMD*. Les 4 derniers octets sont alloués à l'émission ou à la réception de données permettant donc l'envoi d'instruction et la réception d'information. L'envoi et la réception de données via le service SDO sont illustrés sous une forme schématisée dans la figure 8.4. Les informations transmises via l'interface $C\#$ sont au format hexadécimal qui n'est pas intuitif. Ainsi des fonctions de conversion ont été intégrées à nos programmes Matlab pour faciliter la communication avec les moteurs (voir figure 8.3).

8.2.3 . Protocole de sécurisation de la plateforme

Le principe de fonctionnement des encodeurs incrémentaux est problématique pour réussir correctement la localisation de la plateforme au cours du temps. En effet, ces capteurs magnétiques sont d'une grande précision, mais fournissent une position absolue basée sur le relevé du nombre d'incrémentations par rapport à une position de départ inconnu dans l'environnement de la plateforme (table XY). En conséquence, il est nécessaire de concevoir un système qui permet d'établir un référentiel connu pour ces capteurs.

Ce système décrit dans la figure 8.5, consiste à utiliser les limites physiques de la structure de la plateforme pour se situer par rapport à celle-ci. Des déplacements successifs selon les axes x et y sont commandés à la plateforme dans le but de déterminer les positions auxquelles la plateforme est en contact avec la structure. Une fois en contact avec la structure selon l'axe X, le moteur de l'axe X est réinitialisé à 0 (étape 2 sur la figure 8.5), et le procédé est répété pour l'axe Y (étape 3 sur la figure 8.5). Les moteurs étant réinitialisés à 0 pour les axes X et Y, la position de l'étape 3 devient alors le point d'origine du référentiel plateforme.

Dans le but de garantir l'intégrité physique de la plateforme lors des futurs mouvements, une sécurité logicielle est incorporée à l'ensemble des fonctions de mouvement créées. Une zone de sécurité virtuelle (voir figure droite 8.5) de 5 cm est ainsi instaurée pour empêcher tout choc entre la plateforme et la structure.

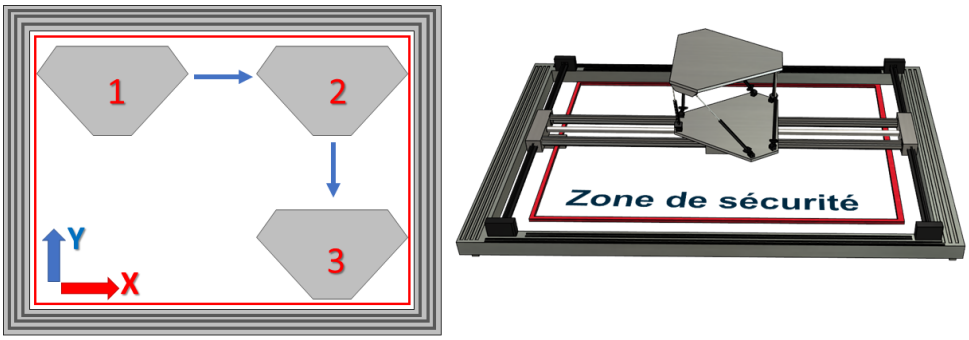


Figure 8.5 – Protocole de sécurisation de la plateforme.

La détection du contact plateforme-structure lors du protocole de sécurisation est rendue possible par le relevé et l'analyse des positions incrémentales des moteurs. Un contact implique de fait l'absence de mouvement qui se traduit par une différence de position ΔP nulle ou très faible entre le moment t et $t-1$ définie par :

$$\Delta P_t = |P_t - P_{t-1}| \tag{8.2}$$

Cette différence de position est ensuite comparée à un seuil α permettant de déterminer la présence de mouvement ou non, et donc de contact. Ce seuil α décrit la valeur minimale de déplacement entre deux acquisitions de position pour considérer la plateforme en mouvement. En conséquence, ce seuil est indexé sur la fréquence d'acquisition des positions (166Hz) f et la vitesse du mouvement assignée au moteur ω. La vitesse des moteurs ω indiquée en tour/min doit être converti en radian/s afin d'exprimer le seuil de la manière suivante :

$$\alpha = 0.3 \times r \times \frac{60\pi}{1000}\omega \times \frac{1}{f} \tag{8.3}$$

Le seuil α est donc proportionnel à la vitesse attribuée à chacun des moteurs et permet de considérer que la plateforme est en contact avec la structure si le déplacement entre deux acquisitions est inférieur à 30% du déplacement théorique induit par la vitesse du moteur. L'algorithme 8 décrit l'ensemble du protocole pour la table XY qui permet l'établissement d'un repère d'origine pour la plateforme.

Le procédé est réitéré pour les moteurs de chacun des vérins afin d'obtenir un référentiel d'origine pour la nacelle et également de mettre en place une sécurisation logicielle de ses mouvements. Notons que dans ce cas, la butée basse des vérins est recherchée comme zone de contact pour définir le repère.

Algorithm 8 Algorithme de sécurisation de la plateforme

Donnée : Constante α

Résultat : Définition de la position de référence (o ;o)

Instruction de mouvement selon l'axe X
Récupérer la position incrémentale du moteur X : P_{i_x}
Initialiser $\Delta X = \alpha$
while $\Delta X \geq \alpha$ **do**
    Récupérer la position actuelle du moteur X : P_{a_x}
    Calculer la différence de position : $\Delta X = |P_{i_x} - P_{a_x}|$
    $P_{i_x} \Longleftarrow P_{a_x}$
end while
Arrêt du moteur X et réinitialisation de l'encodeur à o
Instruction de mouvement selon l'axe Y
Récupérer la position incrémentale du moteur Y : P_{i_y}
Initialiser $\Delta Y = \alpha$
while $\Delta Y \geq \alpha$ **do**
    Récupérer la position actuelle du moteur Y : P_{a_y}
    Calculer la différence de position : $\Delta Y = |P_{i_y} - P_{a_y}|$
    $P_{i_y} \Longleftarrow P_{a_y}$
end while
Arrêt du moteur Y et réinitialisation de l'encodeur à o

8.3 . Contraintes de simulation : contexte et simulateur

Il est important de prendre en considération les contraintes de simulation dans l'élaboration des scénarios de réalité virtuelle. En effet, le cadre d'application et l'outil de simulation imposent des limites qui doivent être intégrées pour garantir une reproductibilité des trajectoires issues du scénario et un impact adapté sur l'utilisateur. Ainsi, il existe une contrainte liée à l'objectif de réhabilitation sur l'aspect contexte, et des contraintes mécaniques au niveau du simulateur.

8.3.1 . Réhabilitation

La partie réhabilitation consiste à prendre en considération l'impact des profils d'accélération sur l'évolution corporelle du robot NAO. Pour cela, divers profils d'accélération de mouvements longitudinaux ont été appliqués à la plateforme avec le robot totalement désarticulé et en position d'équilibre assis. La plateforme était alors soumise à des trajectoires d'en avant puis d'en arrière pour déterminer l'impact du sens du mouvement sur l'équilibre du robot. L'évolution de ses positions articulaires est récupérée à l'aide de notre framework sous ROS à une fréquence de 30Hz.

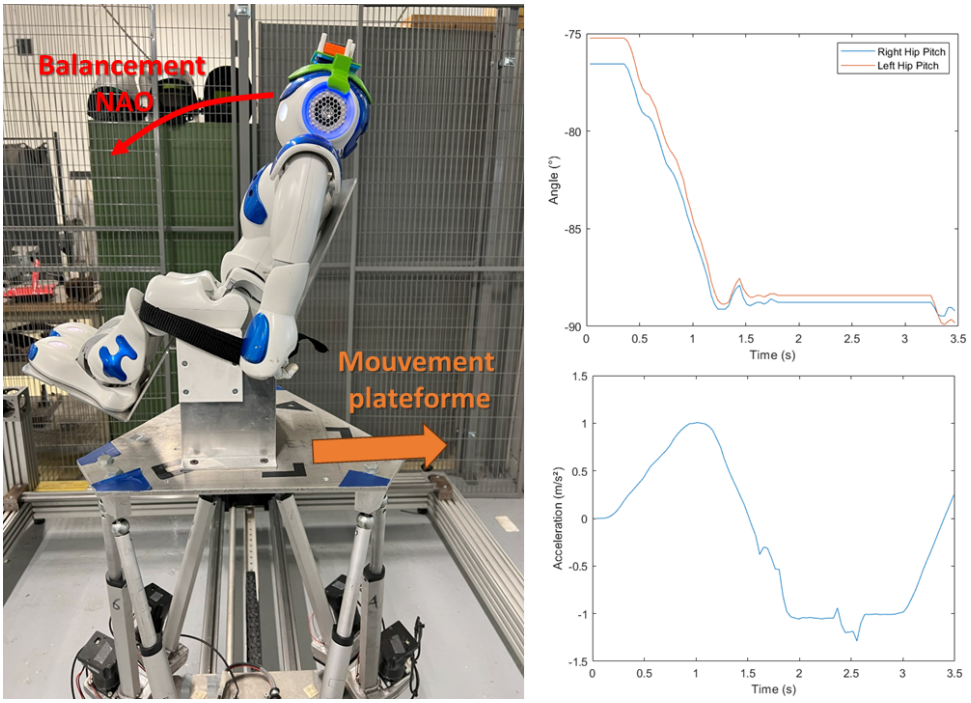


Figure 8.6 – Évolution corporelle pour un mouvement longitudinal en arrière.

Notons qu'en position assise d'équilibre, le robot NAO est à une position angulaire de $-75\,\mathrm{deg}$ et que son amplitude maximale de balancement en avant

liée à ses limites angulaires est de $-88\,\mathrm{deg}$ et qu'en balancement arrière limité par le dossier de son siège, son amplitude maximale est de $-45\,\mathrm{deg}$. La légère différence de position angulaire entre la hanche gauche et droite s'explique par la position assise du robot NAO qui n'est pas parfaitement symétrique.

Table 8.2 – Profils vitesse et accélération des mouvements longitudinaux.

Profil	Mouvement plateforme		Vitesse angulaire maximale	
	Vitesse maximale	Accélération maximale	En avant	En arrière
1	0.79 m/s	Na	20.5 deg/s	Na
2	0.87 m/s	Na	20.1 deg/s	Na
3	0.92 m/s	0.93 m/s^2	25.2 deg/s	-17.4 deg/s
4	0.97 m/s	0.95 m/s^2	27.1 deg/s	-22.3 deg/s
5	0.9 m/s	0.83 m/s^2	23.3 deg/s	-19.8 deg/s
6	1 m/s	1 m/s^2	28.4 deg/s	-23.9 deg/s

Les expérimentations ont été menées pour des mouvements longitudinaux d'en avant et d'en arrière pour 6 profils de vitesse et accélération différents (voir tableau 8.2). Le couple τ résultant du balancement du robot NAO peut être obtenu d'après le moment d'inertie I_{NAO} du robot et son accélération angulaire α. La formule du couple est donné par l'expression suivante :

$$\tau = I_{NAO} \times \alpha \tag{8.4}$$

Où le couple τ est exprimé en Nm, le moment d'inertie I_{NAO} en $kg.s^2$ et l'accélération angulaire en $rad.s^{-2}$. Le couple renseigne indirectement sur la force qui correspondrait au mouvement de balancement du robot, du fait de la relation suivante :

$$\tau = F \times r \tag{8.5}$$

Où F est la force appliquée au robot considérée perpendiculaire au rayon r du cercle correspondant au mouvement de rotation selon un point fixe considéré ici comme le bassin du robot. Les constantes I_{NAO} et r correspondant aux propriétés du robot NAO sont disponible dans la documentation. Le rayon r correspondant à la distance entre le centre d'inertie et le bassin vaut 8.5cm tandis que le moment d'inertie I_{NAO} vaut :

$$I_{NAO} = 1.0496 \begin{Bmatrix} 0.0051 & 1.4312e^{-5} & 0.0002 \\ 1.4312e^{-5} & 0.0049 & -2.7079e^{-5} \\ 0.0002 & -2.7079e^{-5} & 0.0016 \end{Bmatrix} \tag{8.6}$$

Le tableau 8.3 récapitule les couples et forces correspondant aux différents profils extraits des relations précédentes.

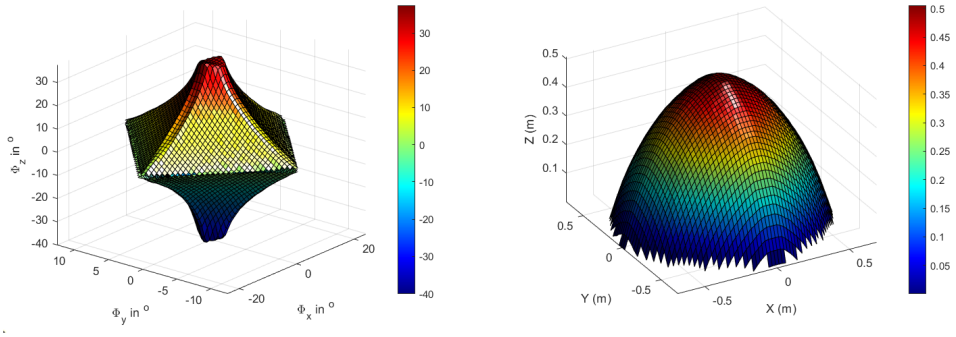
Table 8.3 – Couples et forces appliqués au robot NAO pour les différents profils.

Profil	Couple		Force	
	En avant	En arrière	En avant	En arrière
1	1.012 Nm	0.208 N	Na	Na
2	1.418 Nm	0.291 N	Na	Na
3	0.743 Nm	0.156 N	1.02 Nm	0.199 N
4	1.29 Nm	0.265 N	0.769 Nm	0.162 N
5	2.47 Nm	0.507 N	1.069 Nm	0.215 N
6	3.04 Nm	0.621 N	0.786 Nm	0.182 N

Il est ainsi possible de déterminer si un profil de vitesse et d'accélération satisfait aux conditions d'intégrité physique d'un utilisateur à mobilité réduite.

8.3.2 . Limites mécaniques et Singularités de la plateforme

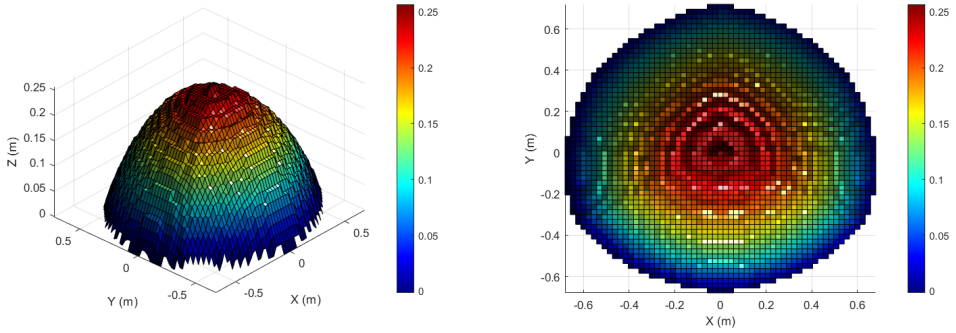
Tout d'abord, la conception mécanique de la plateforme hybride du laboratoire impose certaines limites de mouvement en raison de la taille de sa structure de table XY et de la nature de ses articulations (cardans et rotules). La combinaison de la table XY et de la plateforme parallèle de Gough-Stewart induit un espace de travail correspondant à l'espace où la plateforme est susceptible de se déplacer. Pour la plateforme parallèle de Gough-Stewart, la figure 8.7a illustre la zone de travail accessible en rotation tandis que la figure 8.7b illustre celle pour les translations.



(a) Zone de travail de rotation (b) Zone de travail de translation

Figure 8.7 – Zone de travail de la plateforme parallèle de Gough-Stewart.

Rappelons que la zone de travail accessible correspond à l'espace que la plateforme peut atteindre dans au moins une orientation. En combinant les mouvements de translation et de rotation, nous obtenons la zone de travail habile où la plateforme parallèle peut atteindre l'ensemble des points pour chaque configuration d'orientations (voir figure 8.8).



(a) Zone de travail habile (b) Vue de haut de la zone de travail

Figure 8.8 – Zone de travail habile de la plateforme parallèle de Gough-Stewart.

En prenant notre simulateur XY-6DoF dans son intégralité, à savoir la table XY et la plateforme de Gough-Stewart, nous obtenons alors la zone de travail présentée dans la figure 8.9.

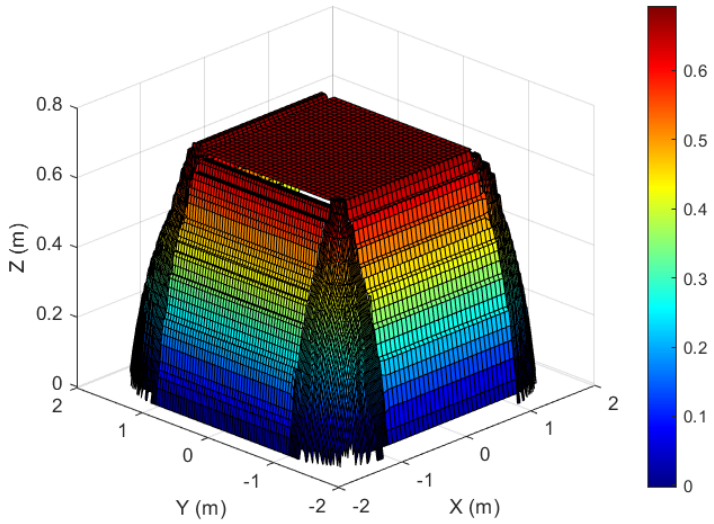


Figure 8.9 – Zone de travail habile du simulateur XY-6DoF.

Cependant, du fait de contraintes mécanique au niveau de la nacelle (plateforme de Gough-Stewart), certaines singularités de mouvement apparaissent,

empêchant certaines combinaisons de rotation et translation au niveau de la nacelle.

Dans le cas de trajectoires provenant de notre scénario de réalité virtuelle, seuls trois types de mouvement peuvent être réalisés : les mouvements longitudinaux, transversaux et verticaux. Chacun de ces mouvements peut être composé d'un mouvement de rotation et translation provenant de la décomposition de la trajectoire linéaire par le MCA. De nombreux travaux [SGoo; Li+06] ont tenté de modéliser les équations de singularité de la plateforme de Gough-Stewart en utilisant 20 coefficients en fonction des angles d'orientation. Cependant, ces approches décrivent les singularités de manière trop complexe et manquent de généralisation en raison de leur expression pour un point donné d'un repère mobile.

Afin de généraliser l'approche précédente, Jiang [Jiao8] considère un point de référence, comme le centre de gravité de la plateforme, pour minimiser le nombre de paramètres décrivant la plateforme haute et basse (nacelle haute et basse). L'équation de singularité est exprimée sous la forme d'une surface du troisième ordre dans l'espace 3D de la manière suivante :

$$f_1x^3 + f_2x^2y + f_3x^2yz + f_4x^2 + f_5y^2x + f_6xyz + f_7xy + f_8xz^2x + f_9xz + f_{10}x + f_{11}y^3 + f_{12}y^2z + f_{13}y^2 + f_{14}yz^2 + f_{15}yz + f_{16}y + f_{17}z^3 + f_{18}z^2 + f_{19}z + f_{20} = 0 \tag{8.7}$$

Où chacun des 20 coefficients f_i sont liés aux paramètres géométriques et aux angles d'orientation (ϕ, θ, ψ), et exprimés dans [Jiao8] sous la forme de calcul de déterminants.

En nous basant sur l'approche de Jiang, le repère O_{xyz} de la plateforme est choisi comme étant le milieu du segment de deux extrémités de la base de la nacelle (hexagone semi-régulier du bas) noté B_1 et B_2. Le segment B_1B_2 indique l'axe x tandis que l'axe y est confondu avec le plan nacelle basse et normal à l'axe x (Voir figure 8.10), l'axe z est lui normal au plan O_{xy}.
En suivant le même procédé, le repère O'_{xyz} de la nacelle haute est le point symétrique du point O. Il est alors possible d'exprimer chacun des points $B_i\ i = (1, 2, \ldots, 6)$ et $P_i\ i = (1, 2, \ldots, 6)$ en fonction de leur repère respectif et de l'autre repère. L'aspect semi-régulier des nacelles haute et basse permet d'exprimer chacune de ces parties à l'aide de 5 paramètres géométriques notés $t_i,\ i = (1, 2, \ldots, 5)$ pour la partie basse et $t_i,\ i = (6, 7, \ldots, 10)$ pour la partie haute.

Lors de mouvements longitudinaux, seule la translation en x et l'inclinaison de tangage θ selon l'axe de rotation peuvent intervenir. En effet, ces mou-

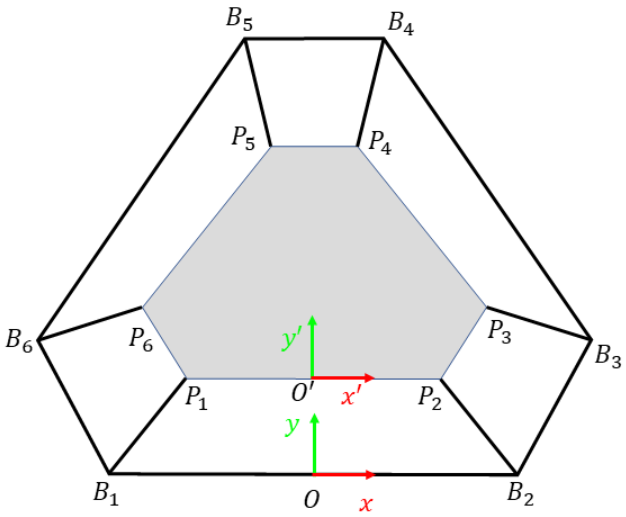


Figure 8.10 – Repère de la plateforme de Gough-Stewart.

vements linéaires peuvent être suivis d'une inclinaison permettant d'accentuer les sensations inertielles. Ainsi y, z, ϕ et ψ sont nuls ce qui permet de réécrire l'équation 8.7 de la manière suivante :

$$f_1 x^3 + f_4 x^2 + f_{10} x + f_{20} = 0 \tag{8.8}$$

Notons que les coefficient f_i sont ici liés uniquement à la variable angulaire θ. Nous pouvons ainsi observer l'absence de singularité dans cette zone de travail.

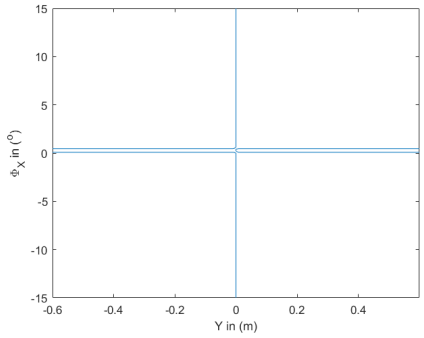
Les mouvements transversaux sont composés de translations selon l'axe y pouvant être suivis de rotations ϕ selon x correspondant à des mouvements de balancier. Dans cette configuration, les variables $x,\ y,\ \theta\ et\ \psi$ sont nulles ce qui permet de réécrire l'équation 8.7 comme suit :

$$f_{11} y^3 + f_{13} y^2 + f_{16} y + f_{20} = 0 \tag{8.9}$$

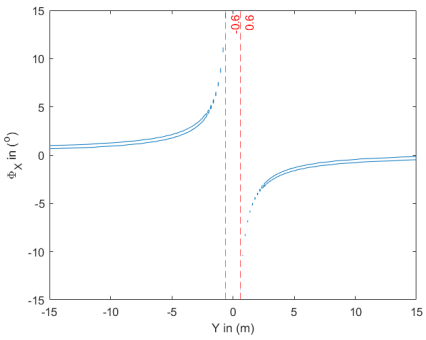
L'équation pour les mouvements transversaux présente des singularités pour $x = z = 0$ comme illustré dans la figure 8.11a. Rappelons que la plateforme possède une amplitude de translation de $\pm 0.6m$ et de rotation de $\pm 15\,\mathrm{deg}$ d'après le tableau 8.1. Dans le cas de l'ajout d'un déplacement vertical de la plateforme, z n'est plus nul et l'équation 8.7 s'écrit alors :

$$f_{11} y^3 + f_{12} y^2 z + f_{13} y^2 + f_{14} y z^2 + f_{15} y z + f_{16} y + f_{17} z^3 + f_{18} z^2 + f_{19} z + f_{20} = 0 \tag{8.10}$$

En considérant les limites d'amplitude de translation de la nacelle haute, nous observons dans la figure 8.11b qu'élever la nacelle de $z = 0.2\ m$ permet de lever les singularités des mouvements transversaux.



(a) Singularité pour $x = z = 0$



(b) Singularité levée pour $x = 0$ et $z = 0.2m$

Figure 8.11 – Singularités de la plateforme de Gough-Stewart pour les mouvements transversaux.

Une combinaison des mouvements transversaux et longitudinaux entraînent une valeur nulle pour z et ψ ce qui permet d'exprimer l'équation de singularité selon ϕ et θ comme étant :

$$f_1x^3 + f_2x^2y + f_4x^2 + f_5y^2x + f_7xy + f_{10}x + f_{11}y^3 + f_{13}y^2 + f_{16}y + f_{20} = 0 \quad (8.11)$$

Cette combinaison de mouvements dépend de 4 variables ce qui ne permet pas sa représentation. De plus, le nombre important de singularités nous conduit à lever celle-ci en utilisant la table XY pour faire les mouvements de translation diagonaux, ce qui implique que $x = y = 0$. Ainsi, l'équation 8.7 dépend uniquement de ϕ et θ et s'écrit :

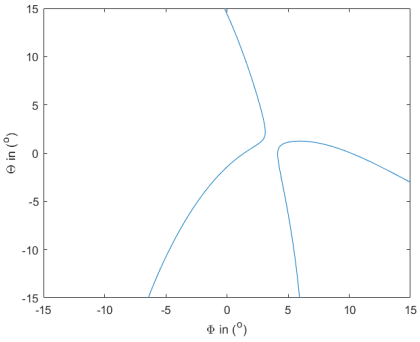
$$f_{20} = 0 \quad (8.12)$$

En utilisant cette astuce, nous levons les singularités liées à l'association de mouvements longitudinaux et transversaux. Cependant, des singularités liées aux rotations subsistent dans le cas où $x = z = 0$ comme illustré dans la figure 8.12a. Afin d'éliminer cette singularité, il est nécessaire d'élever la nacelle haute d'une certaine hauteur $z \geq 0.35m$ ce qui implique une réécriture de l'équation de singularité 8.7 de la manière suivante :

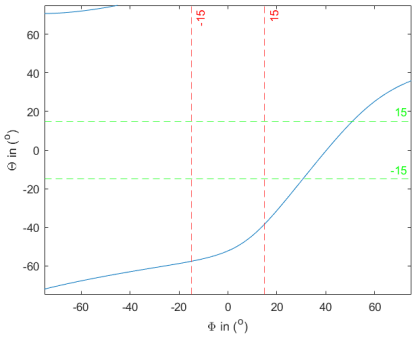
$$f_{17}z^3 + f_{18}z^2 + f_{19}z + f_{20} = 0 \quad (8.13)$$

L'équation de singularité ne dépend alors que des angles ϕ et θ. La figure 8.12b illustre cette élévation pour $z = 0.35m$.

Enfin, la combinaison des mouvements longitudinaux, transversaux et verticaux induit l'utilisation de l'ensemble des variables en dehors du lacet ψ. Afin de réduire la complexité de l'équation de singularité résultante, les mouvements longitudinaux et transversaux peuvent être effectués par la table XY



(a) Singularité pour $x = z = 0$



(b) Singularité levée pour $x = 0$ et $z = 0.35m$

Figure 8.12 – Singularités pour les mouvements combinés longitudinaux et transversaux.

ce qui permet de réduire le degré de liberté de la plateforme de 5 à 3 degrés ($z,\ \phi\ et\ \theta$). L'équation de singularité dépend uniquement de ϕ et θ, et est donc de la même forme que la précédente, à savoir :

$$f_{17}z^3 + f_{18}z^2 + f_{19}z + f_{20} = 0 \tag{8.14}$$

La figure 8.13 illustre les singularités pour la combinaison du mouvement vertical avec les mouvements de rotation, de roulis et tangage.

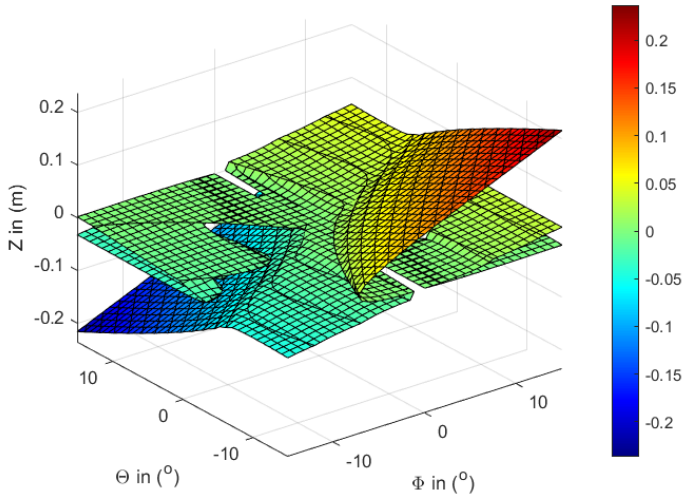


Figure 8.13 – Singularités du mouvement combinant les trois déplacements.

Il apparaît que les singularités sont présentes uniquement pour les mouvements transversaux inférieurs à $0.20m$ dans le champ d'amplitude angulaire $\pm 15\,\mathrm{deg}$ pour la plateforme. L'utilisation de la table XY pour effectuer des mouvements combinés de translation longitudinale et transversale est indispensable à l'exécution de mouvements complexes.

8.4 . Scénario de réalité virtuelle

Le scénario de réalité virtuelle a été développé à l'aide de l'outil-logiciel de création de jeu Unity utilisant le langage de programmation $C\#$ sur la plateforme $.NET$. Unity associe une interface graphique et un support de programmation pour la création d'environnements 3D incorporant des objets 3D inertes ou acteurs soumis à des instructions/interactions. L'environnement sous Unity, appelé scène, contient l'ensemble des éléments visuels nommés GameObjects correspondant aux éléments du moteur comme les images, les objets 3D modélisés par CAO, les scripts d'instruction, etc. Des comportements peuvent être attribués à ces GameObjects grâce à l'utilisation de composants aux propriétés modifiables.
Les scripts servent, eux, à développer des propriétés nouvelles pour les composants afin de créer des comportements nouveaux pour les GameObjects. L'ensemble des scripts sont reliés au fonctionnement du moteur de Unity par l'implémentation de classes dérivées de la classe prédéfinie **MonoBehaviour** permettant un fonctionnement cohérent de l'ensemble du système. En outre, Unity offre la possibilité d'intégrer des modélisations de phénomènes physiques via les scripts standardisant le réalisme produit dans l'environnement de simulation. L'ensemble de ces fonctionnalités ont permis de créer l'environnement dédié à notre étude en garantissant une fiabilité du réalisme des phénomènes physiques et actions présents dans notre scénario.

8.4.1 . Environnement sous Unity

L'environnement sous Unity est composé d'un terrain sur lequel sont disposés des GameObjects évoluant dans une scène définie et régie par des modèles physiques. Le terrain (relief), le modèle de notre handiski, les éléments du paysage (arbres, marqueurs, etc.), la luminosité et la position du repère caméra fournissent les informations visuelles exploitées par l'utilisateur pour s'immerger dans le scénario. Cependant, le terrain conçu composera l'essentiel de l'environnement et influera aussi bien sur l'immersion que sur le scénario. Partant de ce constat, la qualité et la stratégie de la création du terrain ont un rôle majeur sur notre scénario et devront donc satisfaire à une exigence de réalisme.
Le relief du terrain est ainsi calqué de la topologie d'une piste de ski réelle afin de faciliter la création de trajectoire de ski à la fois réaliste et réalisable. La figure 8.14 présente la topographie du terrain de notre scénario. Dans cet environnement simulé relativement grand, notre skieur évolue dans une zone restreinte localisée au niveau des icônes blancs de caméra et soleil de la figure 8.14. Cette parcelle du terrain a été choisie en raison du relief adapté à la pratique du ski pour personne en situation de handicap comme détaillé dans la sous-section 8.3.

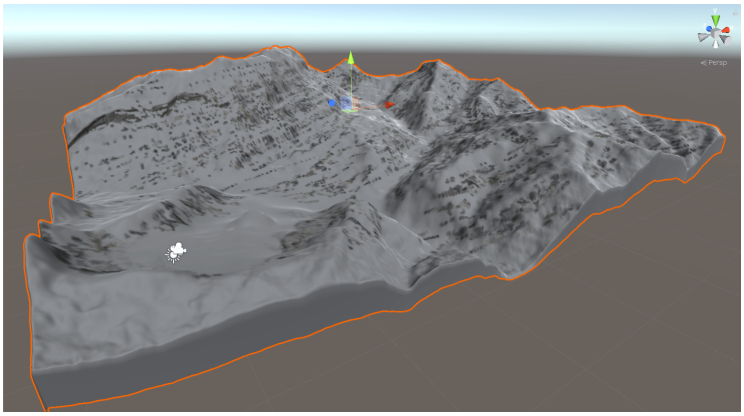


Figure 8.14 – Topologie du terrain.

Afin de rendre cette piste de ski plus réaliste et fournir des repères visuels pour se localiser, des éléments de décors ont été ajoutés comme des arbres et marqueurs (voir figure 8.15). La piste de ski s'étend sur X km pour un dénivelé de X m et une pente moyenne de $11°$. Le siège et l'avatar de skieur ont été modélisés sous Unity en reprenant les caractéristiques des modèles double-ski d'handiski (voir figure 8.16). Ce type de modèle est composé d'un châssis monté et relié à deux skis parallèles par des jointures et pivots à ressorts, et d'un stabilisateur qui permet de déporter le poids lors des virages. L'ensemble est raccordé à un siège d'handiski sur lequel est installé la personne en situation de handicap maintenu par des sangles.

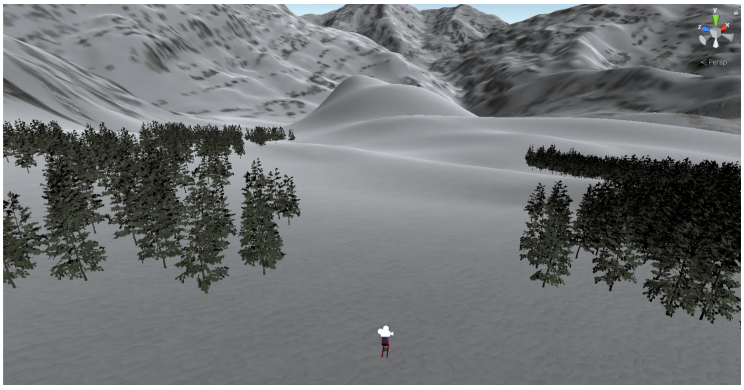


Figure 8.15 – Piste de ski du scénario.

Dans le but de fournir une vue immersive adaptée à notre contexte, le repère caméra est fixé au niveau de la tête du skieur modélisé comme illustré dans la figure 8.17. Ce modèle caméra est doublé et placé à chaque extrémité du visage pour faire homothétie au système de vision humain composé de 2 yeux

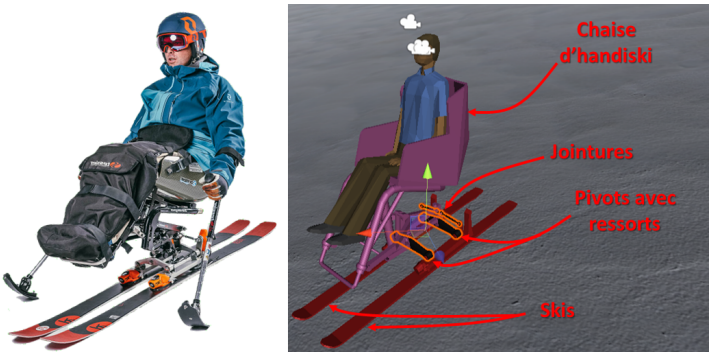


Figure 8.16 – Modélisation CAO d'un modèle double-ski d'handiski.

(les deux icônes caméra de l'image en haut à gauche de la figure 8.17). Notons également que le repère caméra est désigné comme système caméra de l'objet pour que lors de simulations de l'environnement, la vue projetée corresponde au champ de vision du skieur.

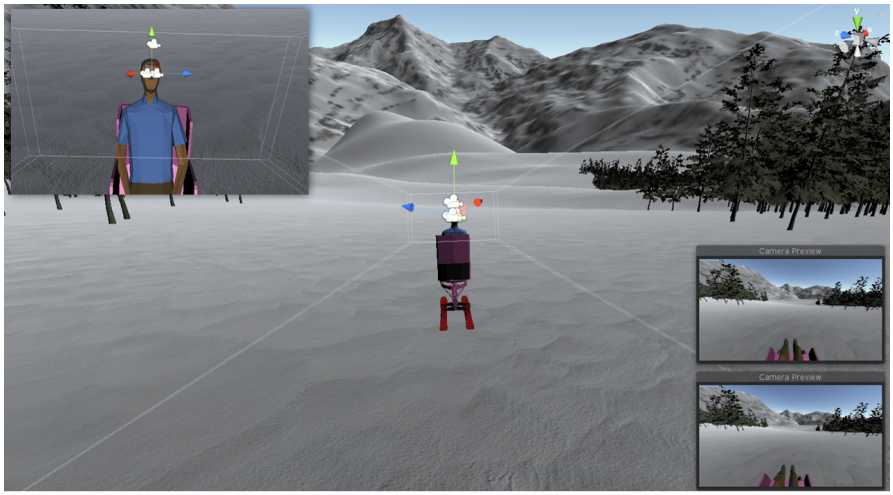


Figure 8.17 – Repères caméra pour l'homothétie du système visuel humain.

L'élaboration des trajectoires est l'élément majeur du scénario, car les trajectoires influent sur l'ensemble de notre système, que ce soit la perception visuelle ou les instructions de mouvement pour la plateforme influant elles-mêmes la perception inertielle. Les trajectoires définies doivent satisfaire à un certain nombre de critères indispensables pour une immersion de qualité. Les mouvements induits par la trajectoire doivent être fluides et coïncider avec le relief du terrain afin d'éviter toute aberration. Unity propose trois solutions pour la génération de trajectoires :

- La génération de trajectoires grâce à une liste de points. Cette solution est en apparence la plus simple, mais ne permet ni de générer des mou-

vements fluides, ni d'avoir des trajectoires qui coïncident suffisamment avec le relief du terrain.

- L'utilisation d'interpolation de spline pour générer des trajectoires fluides adaptées à la montagne virtualisée. Cette solution fluidifie les mouvements, mais ne considère pas les aspects physique et dynamique liés à la pratique du ski et à l'état du skieur.
- Les trajectoires induites par les forces agissant sur le skieur comme la gravité, la résistance au sol, les forces de frottement des skis avec le sol et du skieur avec l'air, et les forces produites par le skieur.

La dernière solution semble être la plus appropriée car elle satisfait les critères de fluidité du mouvement et de cohérence avec le terrain. En outre, la pratique du ski a la particularité d'être une activité sportive produisant une mise en mouvement qui n'est pas directement le fait de l'individu. En effet, le skieur n'est pas acteur de son mouvement, mais se sert de la gravité terrestre subie et des pentes de son environnement pour se déplacer. Il exerce des forces au niveau des skis et de ses hanches et genoux pour diriger sa descente. Ainsi, générer des trajectoires basées sur un raisonnement lié aux forces est cohérent avec la pratique réelle du ski et permettra d'atteindre un meilleur réalisme. En conséquence, de nombreux modèles physiques ont été intégrés à notre scénario dans Unity, que ce soit des phénomènes naturels induis de la pratique du ski (la gravité, la réaction du support ou les frottements), ou des phénomènes liés à l'action de l'avatar (tourner, s'arrêter, ralentir, etc.).

8.4.2 . Modèles physique pour le ski

Quelques travaux traitent de la modélisation des phénomènes physiques appliqués à la pratique du ski en vue d'effectuer des tests en simulation. Korecki effectue une recherche sur la physique et le comportement des skieurs dans son étude [KPW16] à propos des forces sociales lors de la descente d'une piste empruntée par d'autres skieurs.

L'ensemble des modèles physique des forces opérant dans le cadre de la pratique du ski y sont détaillés et repris dans notre scénario. La gravité étant directement incorporée dans le moteur Unity, le modèle de cette force n'a pas eu besoin d'être intégré. Également, la propriété glissante de la neige est intégrée directement à l'environnement en définissant le terrain comme en étant un matériau à base de neige, composant aux propriétés connues dans Unity. Ainsi, comme force globale résultant d'un phénomène naturel, seules la résistance de l'air notée F_{air} et la force de friction des skis sur la neige $F_{ski/neige}$ doivent être exprimés par :

$$F_{air} = -\frac{1}{2m} C_d S_d \rho \|v\|^2 \tag{8.15}$$

$$F_{ski/neige} = -\mu \cos\alpha \|v\|^2 \tag{8.16}$$

Avec m la masse du skieur, C_d le coefficient de traînée, S_d la surface frontale du corps du skieur, ρ la masse volumique de l'air, v la vitesse longitudinale du skieur, μ le coefficient de frottement des skis sur la neige et α l'angle de la pente de descente.

8.4.3 . Trajectoire de ski

Pour soutenir la validation de notre cadre, un scénario de ski virtuel a été développé dans le moteur de jeu multi-plateforme Unity à une fréquence d'images de 50 Hz. Un environnement montagnard a été modélisé directement (voir figure 8.15), ainsi qu'un avatar et son double handiski, comme le montre la figure.8.18.

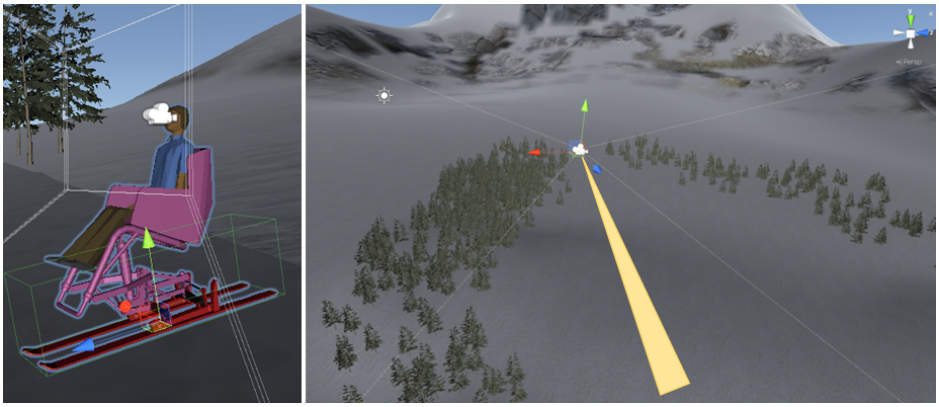


Figure 8.18 – Environnement de réalité virtuelle.

Dans un premier temps, les trajectoires virtuelles générées sont entièrement reproduites par la plateforme robotique XY-6DoF afin de pouvoir évaluer la qualité d'immersion pour des scénarios sans stratégie de restitution inertielle requérant de MCA (Motion Cueing Algorithm). Une trajectoire adaptée aux capacités de la plateforme a ainsi été développée pour soutenir visuellement la simulation, comme le montre la Figure 8.19. Dans ce cas de figure, seuls les mouvements verticaux sont effectués par le simulateur XY-6DoF en raison de sa limitation d'amplitude pour les mouvements longitudinaux. Des mouvements de translation latérale sur un plan incliné de $10°$ sont alors simulés sous Unity et reproduis par la plateforme.

La plateforme de Gough-Stewart est donc inclinée dès le début de $10°$ selon l'axe Y pour correspondre à l'inclinaison du plan du scénario virtuel. Dans la trajectoire virtuelle, le skieur descend la piste de manière linéaire avec une amplitude longitudinale de $73m$ et des déplacements latéraux d'amplitude $+/-0.5m$ qui permettent de rester dans le champ d'amplitude réplicable par la plateforme XY-6DoF (voir figure 8.19). La plateforme XY-6DoF de Gough-

Stewart reproduit de manière synchronisée les profils d'accélérations latérales issus de la trajectoire tandis que l'axe longitudinal reste inactif.

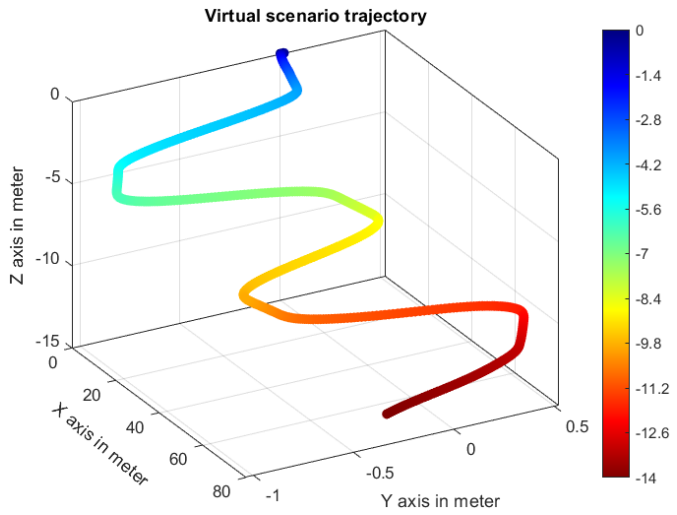


Figure 8.19 – Trajectoire virtuelle

Par la suite, l'application de l'algorithme de MCA permettra la reproduction de l'intégralité des trajectoires virtuelles via la restitution inertielle des sensations induites par la trajectoire. Ainsi, les accélérations du scénario virtuel selon les 3 axes du repère pourront être reproduites par la plateforme grâce au MCA qui permet de surmonter la problématique de la limitation de l'espace de travail des simulateurs. Une accélération induisant un mouvement supérieur à la capacité d'amplitude de la plateforme XY-6DoF pourra alors être exécutée.

8.5 . Expérimentation pour l'évaluation de la perception

8.5.1 . Expérimentation en simulation

Tout d'abord, une étape de simulation illustrée à la figure 8.20 a été réalisée dans SimMechanics (Matlab) pour observer le comportement du robot NAO pour la trajectoire provenant du scénario virtuelle avec le modèle de perception du mouvement propre de l'humain. Ainsi, une trajectoire prédéfinie selon un seul axe de la plateforme a été simulée pour obtenir des données inertielles au niveau du robot NAO. En parallèle, la trajectoire correspondante a été simulée sur Unity afin de fournir des informations visuelles adéquates pour le modèle de perception.
Les articulations du robot NAO sont immobilisées afin de simplifier la simulation et ne pas avoir à implémenter un contrôle posturale qui permettrait au

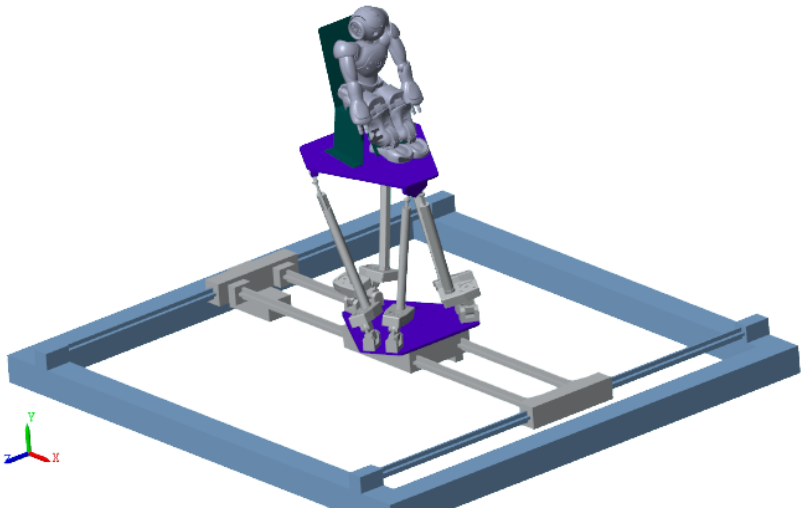


Figure 8.20 – Système multi-robot SimMechanics.

robot NAO de se remettre en position d'équilibre. Un capteur IMU virtuel a été intégré dans la tête du robot NAO pour correspondre à notre véritable système multi-robot. Les données inertielles obtenues sont filtrées afin d'éliminer la composante de gravité et le bruit lié aux vibrations de la plateforme lors du mouvement. Les données inertielles et visuelles sont ensuite introduites dans le modèle de perception du mouvement propre pour émuler la perception humaine de ce mouvement.

Nous observons une erreur absolue significative de perception de la position à travers le modèle TNO par rapport aux données inertielles brutes. En effet, la figure 8.21 illustre l'erreur d'estimation de la position perçue par le modèle et la position intégrée d'après les données inertielles par rapport à la trajectoire réelle connue. Il apparaît que le modèle est plus sensible au changement brusque de direction, mais se stabilise également plus rapidement lors des mouvements linéaires. À l'inverse, l'intégration des données brutes de la centrale inertielle semble moins sensible aux changements brusques de direction, mais davantage sujet à l'accumulation de l'erreur. Ces observations illustrent l'effet de la latence du traitement du flux visuel et des performances de perception inertielle du système vestibulaire. L'expérimentation menée souligne l'importance d'intégrer un algorithme de MCA efficace qui puisse reproduire le plus fidèlement possible les ressentis inertiels afin de restituer au mieux les sensations de mouvement correspondant à la trajectoire virtuelle. Ces ressentis inertiels étant ensuite intégrés au modèle de perception qui reproduit le ressenti humain en vue de l'optimisation du simulateur pour une meilleure immersion.

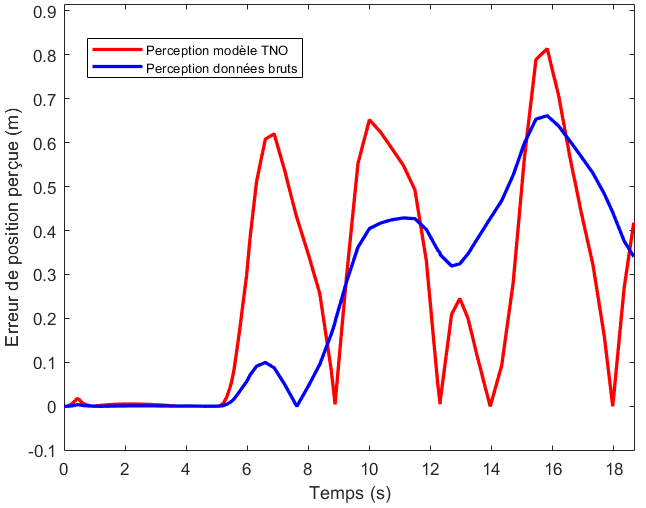


Figure 8.21 – Erreur absolue de position en simulation.

8.5.2 . Expérimentation réelle sans MCA

Tous les composants matériels et logiciels sont gérés dans ce cadre complet à l'aide des plateformes ROS ou Matlab/Simulink. Ici, seul le mouvement de l'axe Y de la trajectoire virtuelle est réalisé en raison des limitations physiques de la plateforme. Dans ce cas applicatif, la plateforme reproduit à l'identique les mouvements de translation latérale des trajectoires du scénario de réalité virtuelle sans utilisation du MCA. La plateforme XY-6DoF réalise des mouvements oscillatoires d'environ 0,5 mètre sur sa partie inférieure avec une inclinaison fixe de 10 degrés le long de l'axe de tangage, comme le montrent les figures 8.18, 8.19 et 8.23.

Les capteurs Xsens IMU disposés sur la tête du robot NAO et sur la nacelle de la plateforme renvoient des données inertielles synchronisées liées au mouvement de la plateforme et du robot, comme illustré à la figure 8.22. En utilisant la méthode décrite dans l'équation $a = R_\omega(f) - g$, la composante de gravité est filtrée à partir des données inertielles. L'accélération filtrée de l'IMU est introduite dans le modèle de perception dans Simulink pour émuler le comportement humain et fournir l'accélération perçue $\tilde{a_h}$ à travers les otolithes homothétiques. De plus, la perception visuelle du mouvement résultant du processus vSLAM est ajoutée au modèle de perception pour prendre en compte les interactions visuo-vestibulaires.

Un retard dans la perception du mouvement est observé sur la figure 8.23 en raison de la latence du processus de vision lors de la vection et de la diminution de la perception des informations inertielles au fil du temps. Par conséquent, une dérive de la perception a été observée au cours de nos ex-

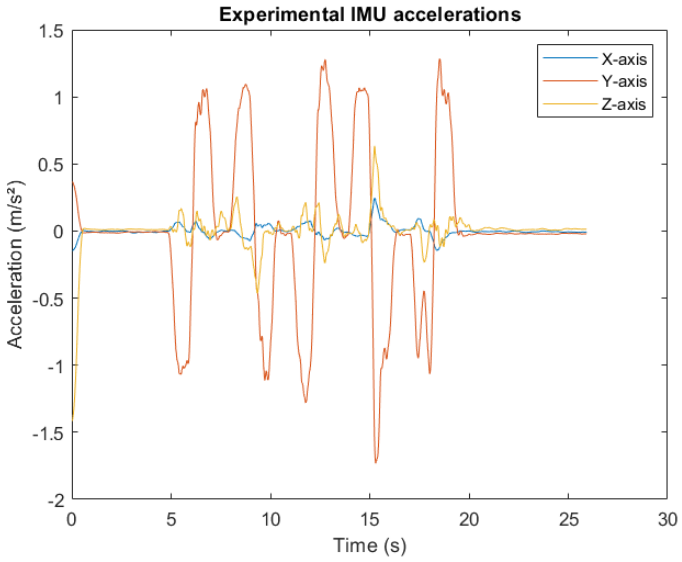


Figure 8.22 – Mesure de l'IMU Xsens pour la trajectoire réelle.

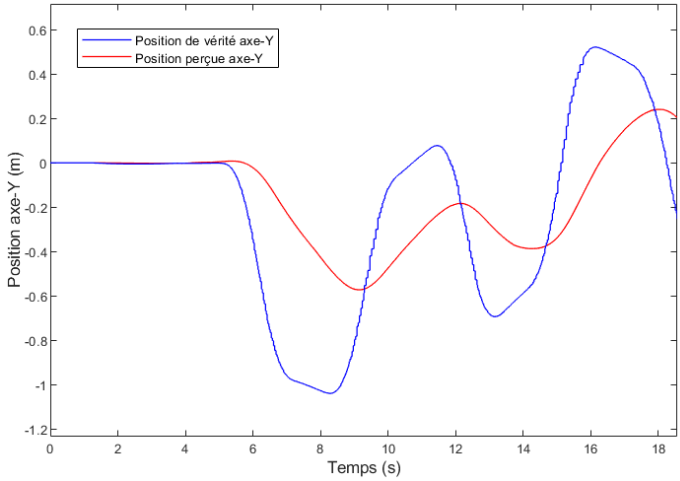


Figure 8.23 – Position perçue selon l'axe Y par le modèle de perception.

périences. Le poids de l'information visuelle par rapport à l'information inertielle dans le processus de perception du mouvement propre chez l'humain explique le retard de perception par le modèle.

Le mouvement perçu est ensuite comparé à la vérité terrain de la trajectoire virtuelle pour évaluer la qualité de l'immersion. L'erreur absolue de position perçue (voir fig.8.24) est calculée pour indiquer la qualité de l'immersion du point de vue du mouvement. Lors du déplacement, nous avons observé une erreur absolue moyenne de 25cm, avec quelques pics lors de changements brusques d'accélération et de direction. Cet indicateur sera utilisé

comme capteur virtuel pour développer un algorithme d'optimisation de la sensation de mouvement prenant en compte à la fois les sensations visuelles et inertielles.

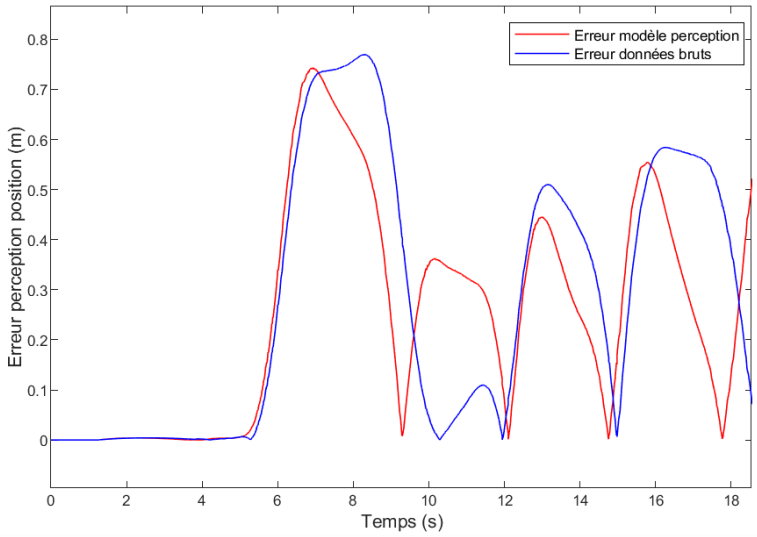


Figure 8.24 – Erreur absolue de position perçue pour le mouvement selon l'axe Y.

8.6 . Expérimentation et résultats de l'optimisation de l'immersion

Une étude approfondie a été menée à propos de la restitution visuo-vestibulaire des sensations de mouvement au sein d'un système de réalité mixte via notre méthode de MCA par apprentissage. Le scénario fournit la trajectoire au système visuel qui l'interprète comme tel tandis que le système inertiel décompose le profil d'accélération de la trajectoire via le MCA afin d'être en capable de reproduire les sensations inertielles dans la zone de travail restreinte de la plateforme robotique.

Le modèle d'apprentissage par renforcement du MCA a été entraîné selon le protocole décrit par la suite pour un profil d'accélération donné. Des expérimentations ont été effectuées tout d'abord uniquement selon l'aspect inertiel du ressenti du mouvement afin d'évaluer les performances des différents modèles de MCA à restituer les profils d'accélérations pour une famille de profil d'accélération donnée. Ensuite, des expérimentations combinant les modèles de MCA et le modèle TNO ont permis d'évaluer la qualité de l'immersion à travers la restitution des sensations inertielles.

8.6.1 . Modèle d'apprentissage par renforcement

Entraînement du modèle

Notre modèle d'apprentissage par renforcement appliqué à la réalisation d'un MCA a été implémenté sous Matlab/Simulink dans un ordinateur avce un processeur intel-core i9 et une carte graphique NVIDIA RTX 2080. Le processus d'entraînement comporte 10 000 épisodes du profil d'accélération illustré dans la figure 8.25 d'une durée de 4s. Le profil d'accélération correspond à un déplacement longitudinale de 13 m qui de fait n'est pas réalisable par la plateforme robotique. L'utilisation du MCA permet alors de lever cette limitation via la décomposition du profil d'accélération.

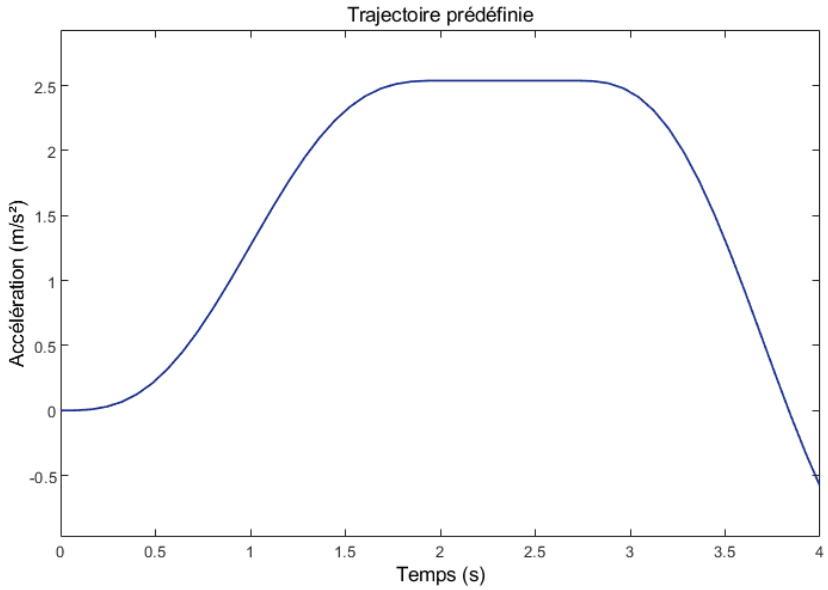


Figure 8.25 – Profil d'accélération pour l'entraînement du modèle.

Les réseaux de neurones des acteurs et critiques sont composés de couches entièrement connectées de 128 noeuds chacune suivies de fonction d'action ReLu. Chacun des réseaux est entraîné avec un taux d'apprentissage de 0.001, un facteur GAE de 0.95, un facteur de réduction de 0.999 et pour 1 période (epoch). L'entraînement du modèle n'est pas encore optimale car un plus grand nombre d'épisode et une incorporation de trajectoires supplémentaires sont nécessaire. Le tableau 8.4 récapitule l'ensemble des poids utilisés pour le calcul de chacune des fonctions de récompense des acteurs Table, Hexapode et Coordination.

Table 8.4 – Poids des fonctions de récompenses des différents agents.

Poids	Agent Table	Agent Hexapode	Agent Coordination
W_a	4	4	4
W_f	1.5	1.5	1.5
W_ω	0	0	2
W_x	1.5	1.5	1.5
$W^i_{a^{ta}}$	0.5	0.25	0.25
$W^i_{a^h}$	0	0.5	0
W^i_ω	0	0	0.5
W^i_{lxh}	0	1	0.25
W^i_{lxta}	1	0.25	0.25
W^i_θ	0	0.3	1

Expérimentation de restitution des sensations inertielles

Des expérimentations pour des profils d'accélération donnés ont permis d'évaluer les performances de notre méthode de MCA pour la restitution des sensations inertielles. La figure 8.26 illustre la décomposition du profil d'accélération de la trajectoire prédéfinie en trois profils d'accélération à travers la table, l'hexapode et la coordination.

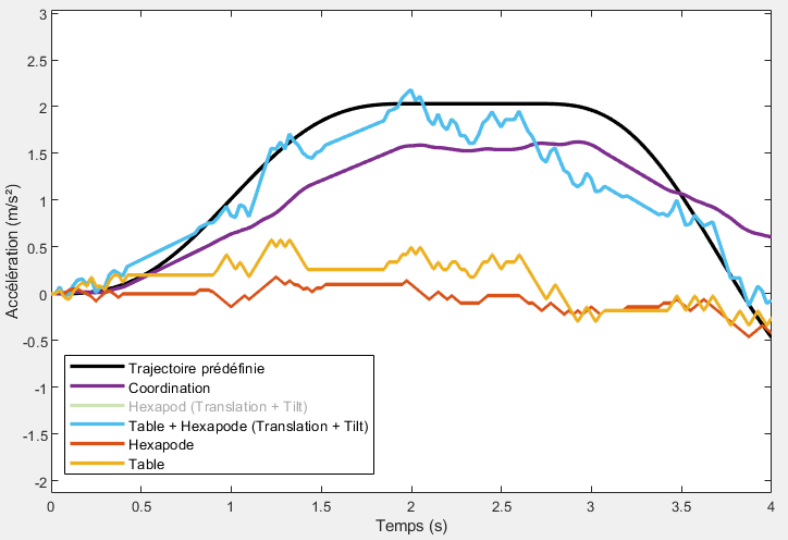


Figure 8.26 – Accélération restituée par le modèle MCA.

L'allure du profil est correctement reproduite, mais des oscillations trop importantes au niveau de la table et l'hexapode sont présentes. De plus, la proportion du profil d'accélération restituée à l'aide de la coordination est trop

importante, ce qui implique un besoin d'amélioration du modèle pour davantage mettre à contribution la table de la plateforme.

D'un point de vue force perçue, la figure 8.27 montre l'efficacité à restituer les forces appliquées à l'individu.

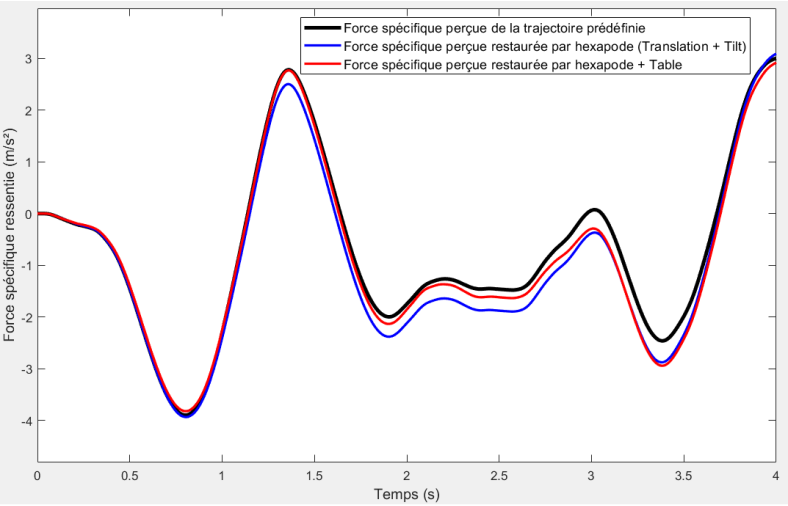


Figure 8.27 – Force spécifique perçue restituée par le modèle MCA.

Les mouvements de translation permettent dans un premier temps d'améliorer la restitution de la force spécifique, mais les profils d'accélération de la table et de l'hexapode imposent une diminution brusque de leurs profils en raison du risque que les composants atteignent leurs limites mécaniques. Le modèle entraîné en l'état ne permet pas une restitution fiable des profils d'accélération pour chaque simulation. En effet, dans certains cas, les limites mécaniques sont atteintes avant la fin de la simulation entraînant l'arrêt de celle-ci. Une étude plus approfondie des coefficients pour chaque fonction de récompense devra être menée. Il est également nécessaire d'augmenter le nombre d'épisodes et de profils d'accélération lors de l'entraînement afin d'atteindre une meilleure convergence et flexibilité du modèle. Pour le moment, le modèle sert de démonstrateur du concept pour un cas simple de trajectoire connue.

8.6.2 . Expérimentations pour l'évaluation de la qualité de l'immersion

Expérimentation en simulation avec le MCA classique et optimal

Une étude préliminaire a été effectuée en simulation dans Simulink afin d'observer l'impact de l'utilisation des approches classique et optimale du MCA pour la restitution des sensations vestibulaires à travers notre modèle de perception visuo-inertiel. Ces expérimentations ont permis d'étendre notre étude à la réalisation de trajectoires issues du scénario par la plateforme qui

permettent de considérer la restitution des sensations vestibulaires de mouvement dépassant la zone de travail de la plateforme. La figure 8.28 illustre la restitution des sensations inertielles perçue obtenues à l'aide de l'approche classique et optimale.

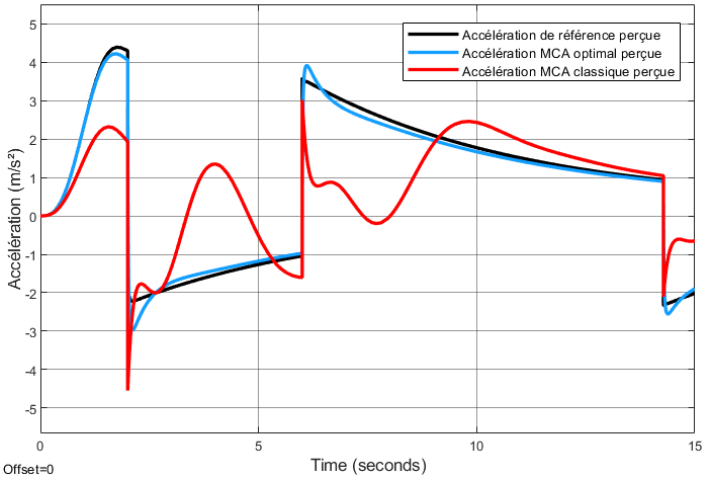
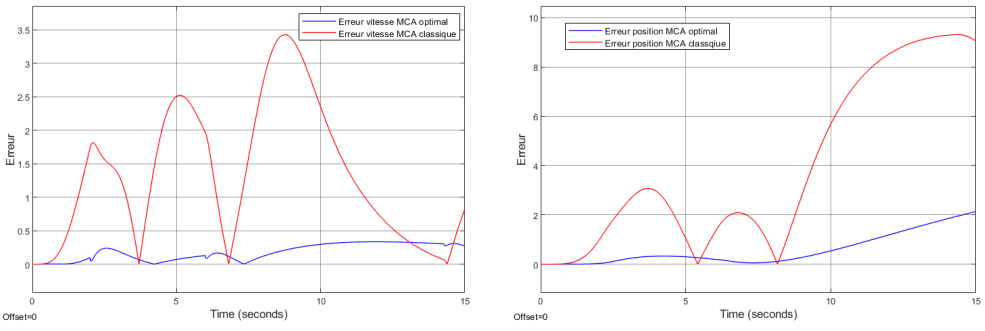


Figure 8.28 – Accélération restituée perçue via les approches classique et optimale.

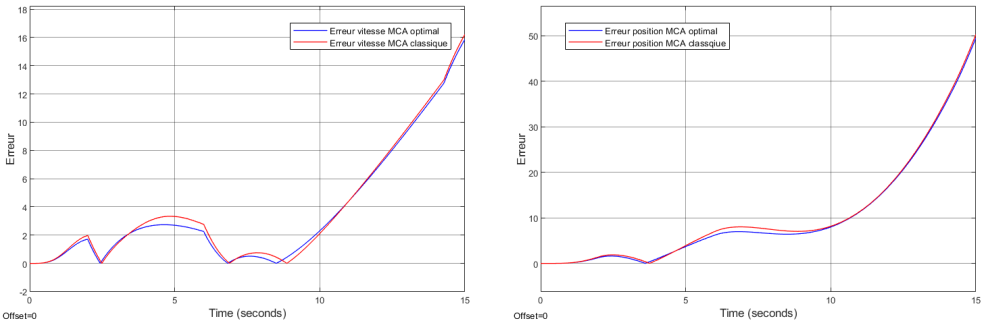
L'approche MCA offre une meilleure restitution des accélérations perçues par rapport à la méthode classique. En nous basant dans un premier temps uniquement sur les informations inertielles, l'erreur de perception de vitesse et de position obtenues par ces deux méthodes permettent de fournir une mesure de la qualité de l'immersion d'un point de vue inertiel (voir figures 8.29a et 8.29b). L'erreur de perception des vitesses est nettement inférieure à celle de l'approche classique garantissant de fait une meilleure cohérence entre la vitesse perçue visuellement et celle perçue à travers le système vestibulaire. De fait, l'erreur de position illustrée dans la figure 8.29b conduit à la même conclusion.



(a) Erreur de perception de la vitesse. (b) Erreur de perception de la position.

Figure 8.29 – Comparatif de performance des approches classique et optimale.

L'expérimentation a été menée pour la même trajectoire afin d'estimer le mouvement perçu à l'aide du modèle de perception visuo-inertiel. À travers cette expérimentation, nous évaluons la qualité de l'immersion selon les critères visuels et inertiels agissant lors du processus de perception du mouvement propre chez l'humain.



(a) Erreur de perception de la vitesse. (b) Erreur de perception de la position.

Figure 8.30 – Comparatif de performance des approches classique et optimale pour la perception visuo-inertielle du mouvement.

L'utilisation du MCA permet d'améliorer les performances du simulateur à restituer de manière plus ou moins précise les ressentis vestibulaire résultant d'une trajectoire où seuls les profils d'accélérations sont reproduits par la plateforme robotique. Les trajectoires du scénario sont utilisées comme telles pour l'élaboration des trajectoires dans le scénario virtuel fournissant le flux d'information visuelle. Les profils d'accélérations correspondant à la trajectoire du scénario sont extraits puis décomposés à l'aide du MCA en trois profils d'accélération qui combinés reproduisent les sensations inertielles originelles tout en garantissant un mouvement réalisable par la plateforme. Le modèle visuo-vestibulaire TNO permet alors une évaluation complète de l'immersion selon les ressentis visuels et inertiels en garantissant une émulation du procédé humain de perception du mouvement.

Cependant, comme l'illustrent les figures 8.30a et 8.30b, l'écart d'erreur de perception de la vitesse et de la position est moins important lors de la prise en compte des informations visuelles. Ce phénomène est tout d'abord dû à la latence du traitement de l'information visuelle par le cortex visuel de l'humain, et au poids de l'information visuelle par rapport à l'inertielle lors du processus de perception du mouvement [HCB11].

Expérimentation en simulation avec notre modèle de MCA

Des expérimentations similaires ont été réalisées avec notre modèle de MCA dans le but d'évaluer la qualité de l'immersion produite à l'aide de notre restitution des sensations inertielles. La restitution des sensations inertielles est représentée dans la figure 8.31 où l'allure et la magnitude des sensations sont bien restituées.

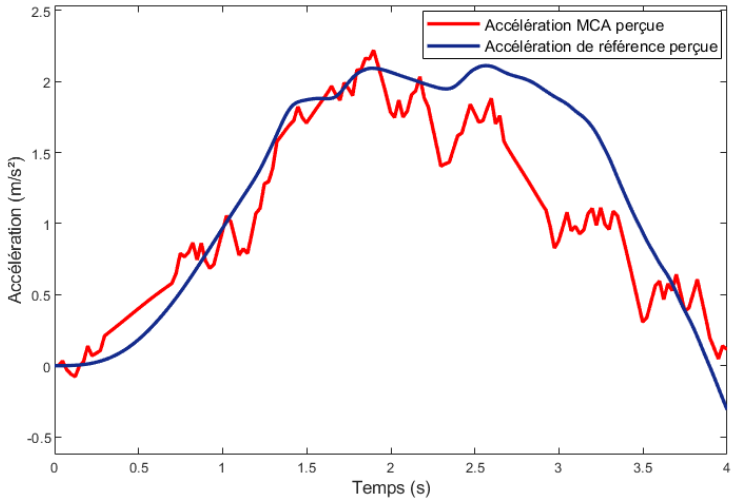
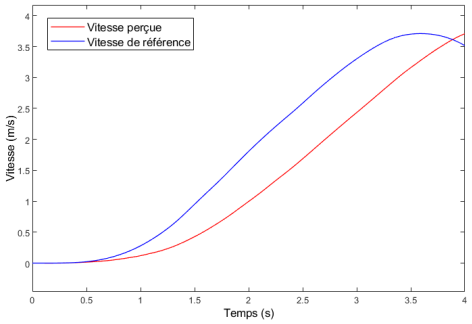
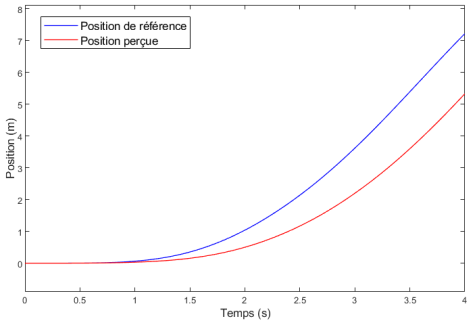


Figure 8.31 – Sensations d'accélération restituées via notre méthode de MCA.

Cependant, nous observons des oscillations liées aux profils d'accélération de la table et l'hexapode eux-même oscillants, créant une erreur cumulée importante. L'analyse des profils vitesse et position perçus est décrite dans les figures 8.32a et 8.32b où nous observons un écart cohérent avec les résultats précédents obtenus pour l'étude du MCA optimal et classique.



(a) Perception de la vitesse via le modèle TNO.



(b) Perception de la position via le modèle TNO.

Figure 8.32 – Résultat de la perception en vitesse et position à l'aide du modèle visuo-inertielle TNO.

8.7 . Conclusion

Le cadre de simulation et la plateforme robotique utilisée comme simulateur ont été détaillés afin de fournir une compréhension des conditions d'expérimentation. En outre, un framework intégrant un modèle de perception visuo-inertiel du mouvement propre a été implémenté et appliqué pour des expérimentations avec notre système robotique réel (plateforme et robot NAO) et des simulations sous Matlab. L'ensemble des expérimentations menées à l'aide de notre framework ont permis d'évaluer la qualité de l'immersion de plusieurs scénarios dont les trajectoires étaient restituées par la plateforme hybride à l'aide des approches du MCA optimale, classique et par apprentissage. Le problème de latence de l'information visuelle et l'erreur de restitution inertielle ont produit une erreur de perception du mouvement qui a été observée lors de nos simulations. À travers ce framework, nous fournissons un outil qui permet de calibrer les scénarios afin d'obtenir la meilleure restitution des sensations visuo-inertielles en adaptant les trajectoires du scénario aux besoins d'immersion.

Une méthode nouvelle d'optimisation de la restitution inertielle par utilisation d'un MCA basé sur l'apprentissage par renforcement a été proposée sur la base de l'existant. La nature hybride de la plateforme robotique nous a mené à concevoir une fonction de récompense adaptée à sa configuration, qui prenne en considération les limites mécaniques de celle-ci. La méthode proposée permet une assez bonne restitution des profils d'accélération à travers les trois composants de la plateforme. Cependant, la méthode nécessite d'être optimisée afin de la rendre flexible à davantage de familles de trajectoire, et de mieux répartir l'accélération entre les différents composants et ainsi limiter l'échec de simulation en raison de limites mécaniques atteintes. Pour cela, il semble indispensable de modifier la fonction de récompense et d'approfondir le processus d'entraînement à travers une plus grande diversité de profils d'accélération et un nombre d'épisodes d'entraînement plus conséquent.

9 - Conclusion

Nous avons commencé cette thèse pour étudier les interactions multi sensorielles entre un système robotique multidimensionnel et multi capteur dans le cadre d'un simulateur de réalité mixte appliqué à la réhabilitation de personne en situation de handicap à travers la pratique de sport de glisse. Il a été nécessaire dans un premier temps d'entreprendre une étude des systèmes de simulation et d'interaction afin d'en comprendre les capacités et limitations. Cette étude nous a conduit à nous focaliser sur l'analyse du processus de perception du mouvement chez l'humain en vue de considérer les interactions multi sensorielles s'opérant au sein de notre système multi robot. En conséquence, nous avons exploré les différents mécanismes de perception liés au sens de la vue et aux ressentis vestibulaires, ce qui nous a conduits à mener des recherches dans le domaine de la vision par ordinateur appliquée à la robotique, et aux méthodes appliquées aux ressentis inertiels à travers les simulateurs. Pour chacune de ces thématiques, nous avons cherché un équilibre entre contributions propres au domaine et intégration dans le cadre de notre sujet d'étude.

La partie vision a été décomposée en deux sous-parties, une méthode de SLAM visuel dynamique servant de substitue au processus de perception du mouvement via le système visuel humain, et une amélioration des performances de cette méthode à l'aide de l'utilisation d'une base de données contenant des images dégradées photo-réalistes. Nous avons ainsi tout d'abord proposé un algorithme de SLAM visuel robuste aux dynamiques de scène inévitables dans les environnements réels et virtuels. Cette méthode contribue à l'amélioration de l'existant en tirant avantage des informations de profondeur et d'interaction entre les objets présents dans la scène afin de réduire l'impact de la dynamique de scène dans la localisation de la caméra et la cartographie de l'environnement. À travers cette méthode, nous introduisons une approche nouvelle basée sur des hypothèses liées à l'impact de la profondeur sur le raisonnement par géométrie épipolaire et sur la segmentation d'objet. Pour cela, cette approche nouvelle utilise les informations de profondeur afin d'améliorer l'estimation de l'état de points d'intérêts lors du processus de SLAM à travers des raisonnements géométrique et sémantiques. Cette prise en compte de la profondeur permet de conserver une cohérence 3D des informations spatiales extraites des images lors du processus réduisant alors l'impact des dynamiques.

Une étape supplémentaire détermine l'état des points restants à travers un raisonnement basé sur l'interaction des objets présents dans la scène ex-

ploitant à la fois les informations géométrique, sémantique et spatiale. Nous introduisons ainsi le module d'attention de la profondeur qui met en rapport l'ensemble de ces informations sous condition de la nature sémantique des objets segmentés. Une hypothèse associée à la nature de la scène (intérieur/extérieur) permet de définir l'état de certaines classes d'objets considérées dynamique de par leur nature. Partant de cette hypothèse, le module proposé analyse le voisinage de ces objets à travers les informations spatiales et géométriques permettant de considérer un plus grand nombre de points. Cette approche permet de surpasser les limitations des méthodes existantes restreintes à la prise en compte des points mis en correspondances d'une image à l'autre ou contenus dans une zone segmentée.

Les expérimentations menées sur la base de données dédiée à cet effet ont montré les limitations de notre approche en présence d'images dégradées. En effet, les méthodes de SLAM dynamique reposent sur le bon déroulement du processus de segmentation d'objet permettant de fournir les informations sémantiques. Partant de ce constat, nous avons tout d'abord entrepris une étude sur l'impact des distorsions d'image sur les performances des modèles de détection d'objet et l'apport de l'utilisation de l'augmentation de donnée pour l'amélioration de leur robustesse. Afin de garder une cohérence avec la problématique de performance de notre processus de segmentation, nous avons créé la première base de données contenant des distorsions d'image appliquées de manière locale et photo-réaliste grâce à la prise en compte du contexte de scène. Cette prise en compte a été rendue possible par l'utilisation des informations de profondeur et des informations sémantiques des objets garantissant une application locale de distorsions qui respectent à la fois l'impact de la profondeur de scène et la nature des objets (position, orientation, etc.) sur les distorsions. Nous avons ainsi introduit, de manière globale et/ou locale, à la fois de distorsions liées à des défauts de réglage ou de limitations optiques, des conditions atmosphériques, et des conditions d'acquisition. Notons qu'à ce jour, cette base de données est la première à proposer des distorsions locales et atmosphériques qui soient photo-réalistes grâce à la prise en compte de la profondeur. L'utilisation de cette base pour l'entraînement du modèle de segmentation d'objet utilisé dans notre algorithme de SLAM a grandement amélioré ses performances, garantissant son bon fonctionnement même en cas d'images fortement dégradées.

La partie optimisation de la simulation a consisté à implémenter sous forme de framework un modèle visuo-inertiel de perception du mouvement exploitant notre méthode de SLAM pour l'estimation du mouvement d'un point de vue visuel et les informations provenant d'une centrale inertielle pour le système vestibulaire. Ce framework offre la possibilité d'exploiter les

informations capteurs provenant du robot humanoïde et des informations visuelles provenant du scénario de réalité virtuel en vue d'estimer le mouvement perçu à travers des critères qualitatifs. De plus, l'utilisation de ce modèle de perception et la substitution de l'humain par le robot NAO offre un outil de calibration des scénarios estimant le mouvement perçu qui s'émancipe du besoins du retour d'information auprès de l'utilisateur via des questionnaires post-simulation. Cette estimation permet à la fois d'évaluer la qualité de l'immersion afin de vérifier la pertinence du scénario conçu, et à terme d'optimiser l'immersion en réduisant l'erreur de perception liée au processus visuel humain et/ou à la qualité de la restitution inertielle du simulateur. En outre, cette partie inclut également l'intégration d'une méthode de MCA basée sur l'apprentissage par renforcement permettant d'exploiter au mieux la redondance du système robotique de simulation. Ce modèle décompose le profil d'accélération issu du scénario en 3 profils pour chacune des parties de la plateforme robotique du simulateur. Cette approche nouvelle doit permettre en théorie de fournir une décomposition du profil d'accélération pour plusieurs familles de profil rendant le système flexible à divers scénario. Cependant, les expérimentations menées jusqu'à maintenant n'ont pas permis d'atteindre ce niveau d'adaptibilité du modèle.

Comme perspectives, une prise en considération de la mauvaise appréciation de la profondeur par l'homme lors des simulations en réalité virtuelle peut être intéressante dans le modèle de perception. L'étude de l'impact de la vision périphérique sur la perception du mouvement en réalité virtuelle pourrait être également réalisée. Enfin, un approfondissement de notre méthode de MCA basé sur l'apprentissage par renforcement est nécessaire pour atteindre des performances satisfaisantes.

Bibliographie

[GJ48] JJ Groen et LBW Jongkees. “The threshold of angular acceleration perception”. In : *The Journal of physiology* 107.1 (1948), p. 1.

[BP49] Jerome Bruner et Leo Postman. “Perception, cognition, and behavior.” In : *Journal of personality* (1949).

[Mei65] Jacob Leon Meiry. “The vestibular system and human dynamic space orientation.” Thèse de doct. Massachusetts Institute of Technology, 1965.

[CS68] Brant Clark et John D Stewart. “Comparison of three methods to determine thresholds for perception of angular acceleration”. In : *The American journal of psychology* 81.2 (1968), p. 207-216.

[YM68] Laurence R Young et Jacob L Meiry. “A revised dynamic otolith model”. In : *Third Symposium on the Role of the Vestibular Organs in Space Exploration, NASA SP-152*. 1968, p. 363-368.

[CS70] Bjorn Conrad et SF Schmidt. *Motion drive signals for piloted flight simulators*. Rapp. tech. NASA, 1970.

[Ben+74] AJ Benson et al. “A systems concept of the vestibular organs”. In : *Vestibular system part 2 : psychophysics, applied aspects and general interpretations* (1974), p. 493-580.

[Par+75] Russell V Parrish et al. “Coordinated adaptive washout for motion simulators”. In : *Journal of aircraft* 12.1 (1975), p. 44-50.

[HV76] RJAW Hosman et JC Van der Vaart. *Thresholds of motion perception measured in a flight simulator*. Technische Hogeschool, 1976.

[HR77] John H Holland et Judith S Reitman. “Cognitive systems based on adaptive algorithms”. In : *Acm Sigart Bulletin* 63 (1977), p. 49-49.

[HV78] RJAW Hosman et JC Van der Vaart. “Vestibular models and thresholds of motion perception. Results of tests in a flight simulator”. In : *Delft University of Technology, Department of Aerospace Engineering, Report LR-265* (1978).

[Zac78] Greg L Zacharias. *Motion cue models for pilot-vehicle analysis*. Rapp. tech. BOLT BERANEK et NEWMAN INC CAMBRIDGE MA CONTROL SYSTEMS DEPT, 1978.

[Ull79] Shimon Ullman. "The interpretation of structure from motion". In : *Proceedings of the Royal Society of London. Series B. Biological Sciences* 203.1153 (1979), p. 405-426.

[FB81] Martin A Fischler et Robert C Bolles. "Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography". In : *Communications of the ACM* 24.6 (1981), p. 381-395.

[Lon81] H Christopher Longuet-Higgins. "A computer algorithm for reconstructing a scene from two projections". In : *Nature* 293.5828 (1981), p. 133-135.

[BD82] Alain Berthoz et Jaques Droulez. "Linear self motion perception". In : *Tutorials on motion perception* (1982), p. 157-199.

[SIH82] Raphael Sivan, Jehuda Ish-Shalom et Jen-Kuang Huang. "An optimal control approach to the design of moving flight simulators". In : *IEEE Transactions on Systems, Man, and Cybernetics* 12.6 (1982), p. 818-827.

[Mit83] Horst Mittelstaedt. "A new solution to the problem of the subjective vertical". In : *Naturwissenschaften* 70 (1983), p. 272-281.

[Per85] Ken Perlin. "An image synthesizer". In : *ACM Siggraph Computer Graphics* 19.3 (1985), p. 287-296.

[Bro86] Rodney Brooks. "A robust layered control system for a mobile robot". In : *IEEE journal on robotics and automation* 2.1 (1986), p. 14-23.

[HS+88] Chris Harris, Mike Stephens et al. "A combined corner and edge detector". In : *Alvey vision conference*. T. 15. 50. Citeseer. 1988, p. 10-5244.

[Mit88] Horst Mittelstaedt. "The information processing structure of the subjective vertical. A cybernetic bridge between its psychophysics and its neurobiology". In : *Processing structures for perception and action*. VCH-Verlagsgesellschaft, 1988, p. 217-263.

[Kam89] Behrooz Kamgar-Parsi. "Evaluation of quantization error in computer vision". In : *IEEE Transactions on Pattern Analysis & Machine Intelligence* 11.09 (1989), p. 929-940.

[KS89] Behzad Kamgar-Parsi et WA Sander. “Quantization error in spatial sampling : comparison between square and hexagonal pixels”. In : *1989 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society. 1989, p. 604-605.

[Alb91] James S Albus. “Outline for a theory of intelligence”. In : *IEEE transactions on systems, man, and cybernetics* 21.3 (1991), p. 473-509.

[CL91] Philip R Cohen et Hector J Levesque. “Teamwork”. In : *Nous* 25.4 (1991), p. 487-512.

[NS91] Günter Niemeyer et J-JE Slotine. “Stable adaptive teleoperation”. In : *IEEE Journal of oceanic engineering* 16.1 (1991), p. 152-162.

[AG92] Christos Alexopoulos et Paul M Griffin. “Path planning for a mobile robot”. In : *IEEE Transactions on systems, man, and cybernetics* 22.2 (1992), p. 318-322.

[Wil92] Ronald J Williams. “Simple statistical gradient-following algorithms for connectionist reinforcement learning”. In : *Machine learning* 8 (1992), p. 229-256.

[ADV93] S ADVANI. “The development of SIMONA-A simulator facility for advanced research into simulation techniques, motion system control and navigation systems technologies”. In : *Flight Simulation and Technologies*. 1993, p. 3574.

[DS94] James Davis et Mubarak Shah. “Visual gesture recognition”. In : *IEE Proceedings-Vision, Image and Signal Processing* 141.2 (1994), p. 101-106.

[FM94] Richard Fitzpatrick et DI McCloskey. “Proprioceptive, visual and vestibular thresholds for the perception of sway during standing in humans.” In : *The Journal of physiology* 478.1 (1994), p. 173-186.

[KH95] Wilfried Käding et Friedrich Hoffmeyer. *The advanced Daimler-Benz driving simulator*. Rapp. tech. SAE Technical Paper, 1995.

[GR97] Peter R Grant et Lloyd D Reid. “Motion washout filter tuning : Rules and requirements”. In : *Journal of aircraft* 34.2 (1997), p. 145-151.

[Pet+97] R Peter Bonasso et al. “Experiences with an architecture for intelligent, reactive agents”. In : *Journal of Experimental & Theoretical Artificial Intelligence* 9.2-3 (1997), p. 237-256.

[AA+98] Ronald C Arkin, Ronald C Arkin et al. *Behavior-based robotics*. MIT press, 1998.

[Boy98] Guy A Boy. *Cognitive function analysis*. T. 2. Greenwood Publishing Group, 1998.

[Hir+98] Kazuo Hirai et al. "The development of Honda humanoid robot". In : *Proceedings. 1998 IEEE international conference on robotics and automation (Cat. No. 98CH36146)*. T. 2. IEEE. 1998, p. 1321-1326.

[Low99] David G Lowe. "Object recognition from local scale-invariant features". In : *Proceedings of the seventh IEEE international conference on computer vision*. T. 2. Ieee. 1999, p. 1150-1157.

[PP99] Constantine P Papageorgiou et Tomaso Poggio. "A trainable object detection system : Car detection in static images". In : (1999).

[Sch99] Stefan Schaal. "Is imitation learning the route to humanoid robots ?" In : *Trends in cognitive sciences* 3.6 (1999), p. 233-242.

[Tri+99] Bill Triggs et al. "Bundle adjustment—a modern synthesis". In : *International workshop on vision algorithms*. Springer. 1999, p. 298-372.

[Zai+99] L Zaichik et al. "Acceleration perception". In : *Modeling and Simulation Technologies Conference and Exhibit*. 1999, p. 4334.

[Atk+00] Christopher G Atkeson et al. "Using humanoid robots to study human behavior". In : *IEEE Intelligent Systems and their applications* 15.4 (2000), p. 46-56.

[Bur+00] Charles G Burgar et al. "Development of robots for rehabilitation therapy : the Palo Alto VA/Stanford experience". In : *Journal of rehabilitation research and development* 37.6 (2000), p. 663-674.

[GCH00] Eric Groen, Mario Clari et Ruud Hosman. "Psychophysical thresholds associated with the simulation of linear acceleration". In : *Modeling and Simulation Technologies Conference*. 2000, p. 4294.

[SG00] Boris Mayer St-Onge et Clément M Gosselin. "Singularity analysis and representation of the general Gough-Stewart platform". In : *The International Journal of Robotics Research* 19.3 (2000), p. 271-288.

[RK00] Gilles Reymond et Andras Kemeny. “Motion cueing in the Renault driving simulator”. In : *Vehicle System Dynamics* 34.4 (2000), p. 249-259.

[Rey+00] Gilles Reymond et al. “Validation of Renault’s dynamic simulator for adaptive cruise control experiments”. In : *Proceedings of the Driving Simulator Conference (DSC00)*. 2000, p. 181-191.

[ST00] Jianbo Shi et Carlo Tomasi. “Good Features to Track”. In : *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 600 (mars 2000). doi : 10.1109/CVPR.1994.323794.

[TCG00] Robert Telban, Frank Cardullo et Liwen Guo. “Investigation of mathematical models of otolith organs for human centered motion cueing algorithms”. In : *Modeling and Simulation Technologies Conference*. 2000, p. 4291.

[Tel+00] Robert J Telban et al. *Motion cueing algorithm development : Initial investigation and redesign of the algorithms*. Rapp. tech. 2000.

[Tri+00] B. Triggs et al. “Bundle adjustment - A modern synthesis”. In : *ICCV ’99 Proceedings of the International Workshop on Vision Algorithms : Theory and Practice* (jan. 2000), p. 198-372.

[BBH01] Jelte Bos, Willem Bles et Ruud Hosman. “Modeling human spatial orientation and motion perception”. In : *AIAA Modeling and Simulation Technologies Conference and Exhibit*. 2001, p. 4248.

[Che+01] LD Chen et al. “NADS at the University of IOWA : A tool for driving safety research”. In : *Proceedings of the 1st human-centered transportation simulation conference*. 2001.

[FT01] Terrence Fong et Charles Thorpe. “Vehicle teleoperation interfaces”. In : *Autonomous robots* 11 (2001), p. 9-18.

[Goo+01] Michael A Goodrich et al. “Experiments in adjustable autonomy”. In : *Proceedings of IJCAI Workshop on autonomy, delegation and control : interacting with intelligent agents*. Seattle, WA. 2001, p. 1624-1629.

[Gra+01] Peter Grant et al. “Motion characteristics of the VIRTTEX motion system”. In : *Proceedings of the 1st human-centered transportation simulation conference*. 2001, p. 4-7.

[MFT01] Reinhard Moratz, Kerstin Fischer et Thora Tenbrink. “Cognitive modeling of spatial reference for human-robot interaction”. In : *International Journal on Artificial Intelligence Tools* 10.04 (2001), p. 589-611.

[SR01] Maria T Schultheis et Albert A Rizzo. “The application of virtual reality technology in rehabilitation.” In : *Rehabilitation psychology* 46.3 (2001), p. 296.

[SCE01] Athanassios Skodras, Charilaos Christopoulos et Touradj Ebrahimi. “The JPEG 2000 still image compression standard”. In : *IEEE Signal processing magazine* 18.5 (2001), p. 36-58.

[TC01] Robert Telban et Frank Cardullo. “An integrated model of human motion perception with visual-vestibular interaction”. In : *AIAA Modeling and Simulation Technologies Conference and Exhibit*. 2001, p. 4249.

[BB02] Jelte E Bos et Willem Bles. “Theoretical considerations on canal–otolith interaction and an observer model”. In : *Biological cybernetics* 86.3 (2002), p. 191-207.

[BHB02] Jelte E Bos, RJ Hosman et W Bles. *Visual-vestibular interactions and spatial (dis) orientation in flight and flight simulation*. Rapp. tech. HUMAN FACTORS RESEARCH INST TNO SOESTERBERG (NETHERLANDS), 2002.

[GJH02] Eric L Groen, Heather L Jenkin et Ian P Howard. “Perception of self-tilt in a true and illusory vertical plane”. In : *Perception* 31.12 (2002), p. 1477-1490.

[Kum+02] Sanjeev Kumar et al. “Toward a formalism for conversation protocols using joint intention theory”. In : *Computational Intelligence* 18.2 (2002), p. 174-228.

[LT02] Corinna E Lathan et Michael Tracey. “The effects of operator spatial perception and sensory feedback on human-robot teleoperation performance”. In : *Presence* 11.4 (2002), p. 368-377.

[PB02] Amal Punchihewa et Donald G Bailey. “Artefacts in image and video systems ; classification and mitigation”. In : *Proceedings of image and vision computing New Zealand*. 2002, p. 197-202.

[Que+02] Francis Quek et al. “Multimodal human discourse : gesture and speech”. In : *ACM Transactions on Computer-Human Interaction (TOCHI)* 9.3 (2002), p. 171-193.

[SLL02] Stephen Se, David Lowe et J.J. Little. “Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks”. In : *The International Journal of Robotics Research* 21 (août 2002), p. 735-760. doi : 10.1177/027836402761412467.

[She+02] Thomas B Sheridan et al. *Humans and automation : System design and research issues*. T. 280. Human Factors et Ergonomics Society Santa Monica, CA, 2002.

[Dif+03] Myron A Diftler et al. “Evolution of the NASA/DARPA robonaut control system”. In : *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*. T. 2. IEEE. 2003, p. 2543-2548.

[FTB03] Terrence Fong, Charles Thorpe et Charles Baur. “Collaboration, dialogue, human-robot interaction”. In : *Robotics research : The tenth international symposium*. Springer. 2003, p. 255-266.

[Nis03] D. Nister. “An Eifficient Solution to the Five-Point Relative Pose Problem”. In : *Proc. of CVPR* 2 (jan. 2003), p. 756-777.

[Sch03] Jean Scholtz. “Theory and evaluation of human robot interactions”. In : *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*. IEEE. 2003, 10-pp.

[BF04] Christoph Bartneck et Jodi Forlizzi. “A design-centred framework for social human-robot interaction”. In : *RO-MAN 2004. 13th IEEE international workshop on robot and human interactive communication (IEEE Catalog No. 04TH8759)*. IEEE. 2004, p. 591-594.

[BH04] Gordon Beavers et Henry Hexmoor. “Types and limits of agent autonomy”. In : *Agents and Computational Autonomy : Potential, Risks, and Solutions 1*. Springer. 2004, p. 95-102.

[Dag+04] M Dagdelen et al. “MPC based motion cueing algorithm : Development and application to the ULTIMATE driving simulator”. In : *DSC 2004 Europe (driving simulation conference)*. 2004, p. 221-233.

[HLA04] Munther A Hassouneh, Hsien-Chiarn Lee et Eyad H Abed. “Washout filters in feedback control : Benefits, limitations and extensions”. In : *Proceedings of the 2004 American control conference*. T. 5. IEEE. 2004, p. 3950-3955.

[Mur04] Robin R Murphy. “Human-robot interaction in rescue robotics”. In : *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 34.2 (2004), p. 138-153.

[NNB04] D. Nister, O. Naroditsky et J. Bergen. “Visual odometry”. In : *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* T. 1. 2004, p. I-I. doi : 10.1109/CVPR.2004.1315094.

[Ree+04] Leah M Reeves et al. “Guidelines for multimodal user interface design”. In : *Communications of the ACM* 47.1 (2004), p. 57-59.

[Bur+05] Catherina Burghart et al. “A cognitive architecture for a humanoid robot : A first approach”. In : *5th IEEE-RAS International Conference on Humanoid Robots, 2005.* IEEE. 2005, p. 357-362.

[Dag05] Mehmet Dagdelen. “Restitution des stimuli inertiels en simulation de conduite”. Thèse de doct. Paris, ENMP, 2005.

[Hee+05] Harm Heerspink et al. “Evaluation of vestibular thresholds for motion detection in the Simona research simulator”. In : *AIAA modeling and simulation technologies conference and exhibit*. 2005, p. 6502.

[HTC05] Jacob A Houck, Robert J Telban et Frank M Cardullo. *Motion cueing algorithm development : Human-centered linear and nonlinear approaches*. Rapp. tech. 2005.

[Kim+05] Gerard Jounghyun Kim et al. “A SWOT analysis of the field of virtual reality rehabilitation and therapy”. In : *Presence* 14.2 (2005), p. 119-146.

[Kle+05] Gary Klein et al. “Common ground and coordination in joint activity”. In : *Organizational simulation* 53 (2005), p. 139-184.

[Kof+05] Jonathan Kofman et al. “Teleoperation of a robot manipulator using a vision-based human-robot interface”. In : *IEEE transactions on industrial electronics* 52.5 (2005), p. 1206-1219.

[Ozt+05] Erhan Oztop et al. “Human–humanoid interaction : is a humanoid robot perceived as a human ?” In : *International Journal of Humanoid Robotics* 2.04 (2005), p. 537-559.

[TPM05] M Tavakoli, RV Patel et M Moallem. “Haptic interaction in robot-assisted endoscopic surgery : a sensorized end-effector”. In : *The International Journal of Medical Robotics and Computer Assisted Surgery* 1.2 (2005), p. 53-63.

[VGP05] Virginie Van Wassenhove, Ken W Grant et David Poeppel. “Visual speech speeds up the neural processing of auditory speech”. In : *Proceedings of the National Academy of Sciences* 102.4 (2005), p. 1181-1186.

[ESN06] Chris Engels, Henrik Stewénius et David Nistér. “Bundle adjustment rules”. In : *Photogrammetric computer vision* 2.32 (2006).

[Gro+06] E Groen et al. “Motion perception thresholds in flight simulation”. In : *AIAA Modeling and Simulation Technologies Conference and Exhibit*. 2006, p. 6254.

[HZ06] Andrew Harltey et Andrew Zisserman. *Multiple view geometry in computer vision (2. ed.).* Jan. 2006. isbn : 978-0-521-54051-3.

[HS06] Peter F Hokayem et Mark W Spong. “Bilateral teleoperation : An historical survey”. In : *Automatica* 42.12 (2006), p. 2035-2057.

[JCL06] Russell E Johnson, Chu-Hsiang Chang et Robert G Lord. “Moving from cognition to behavior : What the research says.” In : *Psychological bulletin* 132.3 (2006), p. 381.

[KKK06] Nam-Gyun Kim, Yong-Yook Kim et Tae-Kyu Kwon. “Development of a virtual reality bicycle simulator for rehabilitation training of postural balance”. In : *Computational Science and Its Applications-ICCSA 2006 : International Conference, Glasgow, UK, May 8-11, 2006. Proceedings, Part I 6*. Springer. 2006, p. 241-250.

[Li+06] Haidong Li et al. “Analytic form of the six-dimensional singularity locus of the general Gough-Stewart platform”. In : (2006).

[Neh+06] Lamri Nehaoua et al. “Motion cueing algorithms for small driving simulator”. In : *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE. 2006, p. 3189-3194.

[RD06] Edward Rosten et Tom Drummond. “Machine learning for high-speed corner detection”. In : *European conference on computer vision*. Springer. 2006, p. 430-443.

[Ste+06] Aaron Steinfeld et al. “Common metrics for human-robot interaction”. In : *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*. 2006, p. 33-40.

[Dau07] Kerstin Dautenhahn. “Socially intelligent robots : dimensions of human–robot interaction”. In : *Philosophical transactions of the royal society B : Biological sciences* 362.1480 (2007), p. 679-704.

[Dav+07] A. J. Davison et al. “MonoSLAM : Real-Time Single Camera SLAM”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.6 (2007), p. 1052-1067. doi : 10.1109/TPAMI.2007.1049.

[GSH07] EL Groen, MH Smaili et RJAW Hosman. “Perception model analysis of flight simulator motion for a decrab maneuver”. In : *Journal of aircraft* 44.2 (2007), p. 427-435.

[JS07] Alejandro Jaimes et Nicu Sebe. “Multimodal human–computer interaction : A survey”. In : *Computer vision and image understanding* 108.1-2 (2007), p. 116-134.

[KM07] G. Klein et D. Murray. “Parallel Tracking and Mapping for Small AR Workspaces”. In : *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. 2007, p. 225-234. doi : 10.1109/ISMAR.2007.4538852.

[Tav+07] Mahdi Tavakoli et al. “High-fidelity bilateral teleoperation systems and the effect of multimodal haptics”. In : *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 37.6 (2007), p. 1512-1528.

[CFG08] Antonio Chella, Marcello Frixione et Salvatore Gaglio. “A cognitive architecture for robot self-consciousness”. In : *Artificial intelligence in medicine* 44.2 (2008), p. 147-154.

[DTV08] Ingrid Daubechies, Gerd Teschke et Luminita Vese. “On some iterative concepts for image restoration”. In : *Advances in Imaging and Electron Physics* 150 (2008), p. 1-51.

[De +08] Agostino De Santis et al. “An atlas of physical human–robot interaction”. In : *Mechanism and Machine Theory* 43.3 (2008), p. 253-270.

[GS+08] Michael A Goodrich, Alan C Schultz et al. “Human–robot interaction : a survey”. In : *Foundations and Trends® in Human–Computer Interaction* 1.3 (2008), p. 203-275.

[Jia08] Qimi Jiang. “Singularity-free workspace analysis and geometric optimization of parallel mechanisms”. Thèse de doct. Citeseer, 2008.

[LL08] Hector Levesque et Gerhard Lakemeyer. “Cognitive robotics”. In : *Foundations of artificial intelligence* 3 (2008), p. 869-886.

[RFP08] Rahul Raguram, Jan-Michael Frahm et Marc Pollefeys. “A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus”. In : *European Conference on Computer Vision*. Springer. 2008, p. 500-513.

[SNK08] Mike Stilman, Koichi Nishiwaki et Satoshi Kagami. “Humanoid teleoperation for whole body manipulation”. In : *2008 IEEE International Conference on Robotics and Automation*. IEEE. 2008, p. 3175-3180.

[WLJ08] Zhiliang Wang, Yaofeng Liu et Xiao Jiang. “The research of the humanoid robot with facial expressions for emotional interaction”. In : *2008 First International Conference on Intelligent Networks and Intelligent Systems*. IEEE. 2008, p. 416-420.

[Bon09] Charles Boncelet. “Image noise models”. In : *The essential guide to image processing*. Elsevier, 2009, p. 143-167.

[DLO09] Bruno Dumas, Denis Lalanne et Sharon Oviatt. “Multimodal interfaces : A survey of principles, models and frameworks”. In : *Human machine interaction : Research results of the mmi program* (2009), p. 3-26.

[KM09] G. Klein et D. Murray. “Parallel Tracking and Mapping on a camera phone”. In : *2009 8th IEEE International Symposium on Mixed and Augmented Reality*. 2009, p. 83-86. doi : 10.1109/ISMAR.2009.5336495.

[Cal+10] Michael Calonder et al. “Brief : Binary robust independent elementary features”. In : *European conference on computer vision*. Springer. 2010, p. 778-792.

[Lem+10] Séverin Lemaignan et al. “ORO, a knowledge management platform for cognitive architectures in robotics”. In : *2010 IEEE/RSJ International conference on intelligent robots and systems*. IEEE. 2010, p. 3548-3553.

[SKD10] Stephan Sehestedt, Sarath Kodagoda et Gamini Dissanayake. “Robot path planning in a social context”. In : *2010 IEEE Conference on Robotics, Automation and Mechatronics*. IEEE. 2010, p. 206-211.

[Wan10] Yingxu Wang. “Cognitive robots”. In : *IEEE robotics & automation magazine* 17.4 (2010), p. 54-62.

[Zie+10] Stéphane Zieba et al. “Principles of adjustable autonomy : a framework for resilient human–machine cooperation”. In : *Cognition, Technology & Work* 12.3 (2010), p. 193-203.

[Dol+11] Piotr Dollar et al. “Pedestrian detection : An evaluation of the state of the art”. In : *IEEE transactions on pattern analysis and machine intelligence* 34.4 (2011), p. 743-761.

[HCB11] Ruud JAW Hosman, Frank M Cardullo et Jelte E Bos. “Visual-vestibular interaction in motion perception”. In : *AIAA Modeling and Simulation Technologies Conference 2011*. 2011, p. 522-534.

[LCS11] Stefan Leutenegger, Margarita Chli et Roland Y Siegwart. “BRISK : Binary robust invariant scalable keypoints”. In : *2011 International conference on computer vision*. Ieee. 2011, p. 2548-2555.

[NLD11] R. A. Newcombe, S. J. Lovegrove et A. J. Davison. “DTAM : Dense tracking and mapping in real-time”. In : *2011 International Conference on Computer Vision*. 2011, p. 2320-2327. doi : 10.1109/ICCV.2011.6126513.

[Pap11] Elpiniki I Papageorgiou. “Learning algorithms for fuzzy cognitive maps—a review study”. In : *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.2 (2011), p. 150-163.

[Ros+11] Nathaniel Rossol et al. “A framework for adaptive training and games in virtual reality rehabilitation environments”. In : *proceedings of the 10th international conference on virtual reality continuum and its applications in industry*. 2011, p. 343-346.

[Rub+11] Ethan Rublee et al. “ORB : An efficient alternative to SIFT or SURF”. In : *2011 International conference on computer vision*. Ieee. 2011, p. 2564-2571.

[GT12] Dorian Gálvez-López et Juan D Tardos. “Bags of binary words for fast place recognition in image sequences”. In : *IEEE Transactions on Robotics* 28.5 (2012), p. 1188-1197.

[Gra+12] Mary Grace Gaerlan et al. “Postural balance in young adults : the role of visual, vestibular and somatosensory systems”. In : *Journal of the American Academy of Nurse Practitioners* 24.6 (2012), p. 375-381.

[HB12] Sandra Hirche et Martin Buss. “Human-oriented control for haptic teleoperation”. In : *Proceedings of the IEEE* 100.3 (2012), p. 623-647.

[KM12] Vijay Kumar et Nathan Michael. “Opportunities and challenges with autonomous micro aerial vehicles”. In : *The International Journal of Robotics Research* 31.11 (2012), p. 1279-1291.

[Stu+12] J. Sturm et al. “A Benchmark for the Evaluation of RGB-D SLAM Systems”. In : *Proc. of the International Conference on Intelligent Robot Systems (IROS)*. 2012.

[Tor12] Carme Torras. *Computer vision : theory and industrial applications*. Springer Science & Business Media, 2012.

[AM13] Ismail Almetwally et Malik Mallem. “Real-time tele-operation and tele-walking of humanoid Robot Nao using Kinect Depth Camera”. In : *2013 10th IEEE International Conference on networking, sensing and control (ICNSC)*. IEEE. 2013, p. 463-466.

[CBA13] Félix Chénier, Pascal Bigras et Rachid Aissaoui. “A new wheelchair ergometer designed as an admittance-controlled haptic robot”. In : *IEEE/ASME Transactions on Mechatronics* 19.1 (2013), p. 321-328.

[GB13] Nikhil JI Garrett et Matthew C Best. “Model predictive driving simulator motion cueing algorithm with actuator-based constraints”. In : *Vehicle System Dynamics* 51.8 (2013), p. 1151-1172.

[GCB13] Michael A Goodrich, Jacob W Crandall et Emilia Barakova. “Teleoperation and beyond for assistive humanoid robots”. In : *Reviews of Human factors and ergonomics* 9.1 (2013), p. 175-226.

[KBP13] Jens Kober, J Andrew Bagnell et Jan Peters. “Reinforcement learning in robotics : A survey”. In : *The International Journal of Robotics Research* 32.11 (2013), p. 1238-1274.

[LJO13] Christian Lebiere, Florian Jentsch et Scott Ososky. “Cognitive models of decision making processes for human-robot interaction”. In : *Virtual Augmented and Mixed Reality. Designing and Developing Augmented and Virtual Environments : 5th International Conference, VAMR 2013, Held as Part of HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part I 5*. Springer. 2013, p. 285-294.

[Nie+13] FM Nieuwenhuizen et al. “Cross-platform Validation for a Model of a Low-cost Stewart Platform”. In : *J Model Simul Identif Control* 1 (2013), p. 1-23.

[Tan+13] Wei Tan et al. “Robust monocular SLAM in dynamic environments”. In : *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE. 2013, p. 209-218.

[Blo+14] Martine Blouin et al. “Characterization of the immediate effect of a training session on a manual wheelchair simulator with Haptic biofeedback : Towards more effective propulsion”. In : *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 23.1 (2014), p. 104-115.

[DWH14] Kevin J DeMarco, Michael E West et Ayanna M Howard. “Autonomous robot-diver assistance through joint intention theory”. In : *2014 Oceans-St. John’s*. IEEE. 2014, p. 1-5.

[DY+14] Li Deng, Dong Yu et al. “Deep learning : methods and applications”. In : *Foundations and trends® in signal processing* 7.3–4 (2014), p. 197-387.

[FON14] HC Fang, SK Ong et AYC Nee. “A novel augmented reality-based interface for robot path planning”. In : *International Journal on Interactive Design and Manufacturing (IJIDeM)* 8 (2014), p. 33-42.

[FPS14] Christian Forster, Matia Pizzoli et Davide Scaramuzza. “SVO : Fast semi-direct monocular visual odometry”. In : *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2014, p. 15-22.

[Gir+14] Ross Girshick et al. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2014. arXiv : 1311.2524 [cs.CV].

[KLM14] Faisal Karmali, Koeun Lim et Daniel M Merfeld. “Visual and vestibular perceptual thresholds each demonstrate better precision at specific frequencies and also exhibit optimal integration”. In : *Journal of Neurophysiology* 111.12 (2014), p. 2393-2403.

[KM14] Yunkyung Kim et Bilge Mutlu. “How social distance shapes human–robot interaction”. In : *International Journal of Human-Computer Studies* 72.12 (2014), p. 783-795.

[Lin+14] Tsung-Yi Lin et al. “Microsoft coco : Common objects in context”. In : *European conference on computer vision*. Springer. 2014, p. 740-755.

[MNB14] Matjaž Mihelj, Domen Novak et Samo Beguš. “Virtual reality technology and applications”. In : (2014).

[MT14a] Raúl Mur-Artal et Juan D Tardós. “Fast relocalisation and loop closing in keyframe-based SLAM”. In : *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, p. 846-853.

[MT14b] Raúl Mur-Artal et Juan D Tardós. “ORB-SLAM : tracking and mapping recognizable features”. In : *Workshop on Multi View Geometry in Robotics (MVIGRO)-RSS*. T. 2014. 2014, p. 2.

[ST14] Andrew Sawers et Lena H Ting. “Perspectives on human-human sensorimotor interactions for the design of rehabilitation robots”. In : *Journal of neuroengineering and rehabilitation* 11 (2014), p. 1-13.

[Tur14] Matthew Turk. “Multimodal interaction : A review”. In : *Pattern recognition letters* 36 (2014), p. 189-195.

[Wex14] Alan Wexelblat. *Virtual reality : applications and explorations*. Academic Press, 2014.

[BJ15] Ajay Kumar Boyat et Brijendra Kumar Joshi. “A review paper : noise models in digital image processing”. In : *arXiv preprint arXiv :1505.03489* (2015).

[CEC15] D. Caruso, J. Engel et D. Cremers. “Large-scale direct SLAM for omnidirectional cameras”. In : *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, p. 141-148. doi : 10.1109/IROS.2015.7353366.

[CBA15] Félix Chénier, Pascal Bigras et Rachid Aissaoui. “A new dynamic model of the wheelchair propulsion on straight and curvilinear level-ground paths”. In : *Computer methods in biomechanics and biomedical engineering* 18.10 (2015), p. 1031-1043.

[ESC15] J. Engel, J. Stückler et D. Cremers. “Large-scale direct SLAM with stereo cameras”. In : *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, p. 1935-1942. doi : 10.1109/IROS.2015.7353631.

[Gir15] Ross Girshick. *Fast R-CNN*. 2015. arXiv : 1504.08083 [cs.CV].

[HM15] Julia Hirschberg et Christopher D Manning. "Advances in natural language processing". In : *Science* 349.6245 (2015), p. 261-266.

[Lil+15] Timothy P Lillicrap et al. "Continuous control with deep reinforcement learning". In : *arXiv preprint arXiv :1509.02971* (2015).

[LSD15] Jonathan Long, Evan Shelhamer et Trevor Darrell. *Fully Convolutional Networks for Semantic Segmentation*. 2015. arXiv : 1411.4038 [cs.CV].

[MMT15a] R. Mur-Artal, J. M. M. Montiel et J. D. Tardós. "ORB-SLAM : A Versatile and Accurate Monocular SLAM System". In : *IEEE Transactions on Robotics* 31.5 (2015), p. 1147-1163. doi : 10.1109/TRO.2015.2463671.

[MMT15b] R. Mur-Artal, J. M. M. Montiel et J. D. Tardós. "ORB-SLAM : A Versatile and Accurate Monocular SLAM System". In : *IEEE Transactions on Robotics* 31.5 (2015), p. 1147-1163. doi : 10.1109/TRO.2015.2463671.

[Naj+15] Maryam M Najafabadi et al. "Deep learning applications and challenges in big data analytics". In : *Journal of big data* 2.1 (2015), p. 1-21.

[OS15] Shipra Ojha et Sachin Sakhare. "Image processing techniques for object tracking in video surveillance-A survey". In : *2015 International Conference on Pervasive Computing (ICPC)*. IEEE. 2015, p. 1-6.

[Rah+15] Mohammad Habibur Rahman et al. "Development of a whole arm wearable robotic exoskeleton for rehabilitation and to assist upper limb movements". In : *Robotica* 33.1 (2015), p. 19-39.

[Ren+15] Shaoqing Ren et al. "Faster r-cnn : Towards real-time object detection with region proposal networks". In : *Advances in neural information processing systems* 28 (2015).

[Yu+15] Haoyong Yu et al. "Human–robot interaction control of rehabilitation robots with series elastic actuators". In : *IEEE Transactions on Robotics* 31.5 (2015), p. 1089-1100.

[Ami+16] Yacine Amirat et al. *Assistance and service robotics in a human environment*. 2016.

[AS16] J Anil et L Padma Suresh. “Literature survey on face and face expression recognition”. In : *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*. IEEE. 2016, p. 1-6.

[BKC16] Vijay Badrinarayanan, Alex Kendall et Roberto Cipolla. *SegNet : A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation*. 2016. arXiv : 1511.00561 [cs.CV].

[Bor+16] Adrián Borrego et al. “Feasibility of a walking virtual reality system for rehabilitation : objective and subjective parameters”. In : *Journal of neuroengineering and rehabilitation* 13 (2016), p. 1-10.

[Che+16a] Liang-Chieh Chen et al. *Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs*. 2016. arXiv : 1412.7062 [cs.CV].

[Che+16b] Andrea Cherubini et al. “Collaborative manufacturing with physical human–robot interaction”. In : *Robotics and Computer-Integrated Manufacturing* 40 (2016), p. 1-13.

[Chi+16] Noelia Chia Bejarano et al. “Robot-assisted rehabilitation therapy : recovery mechanisms and their implications for machine design”. In : *Emerging Therapies in Neurorehabilitation II* (2016), p. 197-223.

[He+16] Kaiming He et al. “Deep residual learning for image recognition”. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, p. 770-778.

[KPW16] Tomasz Korecki, Dariusz Pałka et Jarosław Wąs. “Adaptation of social force model for simulation of downhill skiing”. In : *Journal of computational science* 16 (2016), p. 29-42.

[Liu+16] Wei Liu et al. “SSD : Single Shot MultiBox Detector”. In : *Lecture Notes in Computer Science* (2016), p. 21-37. issn : 1611-3349. doi : 10.1007/978-3-319-46448-0_2. url : http://dx.doi.org/10.1007/978-3-319-46448-0_2.

[MZY16] Weiguang Ma, Xiaodong Zhang et Gui Yin. “Design on intelligent perception system for lower limb rehabilitation exoskeleton robot”. In : *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. IEEE. 2016, p. 587-592.

[MBM16] Alessandro Marro, Taha Bandukwala et Walter Mak. “Three-dimensional printing and medical imaging : a review of the methods and applications”. In : *Current problems in diagnostic radiology* 45.1 (2016), p. 2-9.

[Mog+16] Andrea Moglia et al. “A systematic review of virtual reality simulators for robot-assisted surgery”. In : *European urology* 69.6 (2016), p. 1065-1080.

[Moh+16] Arash Mohammadi et al. “Future reference prediction in model predictive control based driving simulators”. In : *Australasian conference on robotics and automation (ACRA2016)*. 2016.

[PPD16] Maxime Petit, Grégoire Pointeau et Peter Ford Dominey. “Reasoning based on consolidated real world experience acquired by a humanoid robot”. In : *Interaction Studies* 17.2 (2016), p. 248-278.

[Red+16] Joseph Redmon et al. *You Only Look Once : Unified, Real-Time Object Detection*. 2016. arXiv : 1506.02640 [cs.CV].

[Ren+16] Shaoqing Ren et al. *Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks*. 2016. arXiv : 1506.01497 [cs.CV].

[San+16] Oscar Sandoval-Gonzalez et al. “Design and development of a hand exoskeleton robot for active and passive rehabilitation”. In : *International Journal of Advanced Robotic Systems* 13.2 (2016), p. 66.

[She16] Thomas B Sheridan. “Human–robot interaction : status and challenges”. In : *Human factors* 58.4 (2016), p. 525-532.

[VGD16] Neil Vaughan, Bodgan Gabrys et Venketesh N Dubey. “An overview of self-adaptive technologies within virtual reality training”. In : *Computer Science Review* 22 (2016), p. 65-87.

[Win+16] Michael Windrich et al. “Active lower limb prosthetics : a systematic review of design issues and solutions”. In : *Biomedical engineering online* 15.3 (2016), p. 5-19.

[WWD16] Qingcong Wu, Xingsong Wang et Fengpo Du. “Development and analysis of a gravity-balanced exoskeleton for active rehabilitation training of upper limb”. In : *Proceedings of the Institution of Mechanical Engineers, Part C : Journal of Mechanical Engineering Science* 230.20 (2016), p. 3777-3790.

[ARO17] Abdulaziz Alshaer, Holger Regenbrecht et David O'Hare. "Immersion factors affecting perception and behaviour in a virtual reality power wheelchair simulator". In : *Applied ergonomics* 58 (2017), p. 1-12.

[Amb+17] Emilia Ambrosini et al. "The combined action of a passive exoskeleton and an EMG-controlled neuroprosthesis for upper limb stroke rehabilitation : First results of the RETRAINER project". In : *2017 International Conference on Rehabilitation Robotics (ICORR)*. IEEE. 2017, p. 56-61.

[Arc+17] Philippe S Archambault et al. "Development and user validation of driving tasks for a power wheelchair simulator". In : *Disability and rehabilitation* 39.15 (2017), p. 1549-1556.

[Bec+17] Philipp Beckerle et al. "A human–robot interaction perspective on assistive and rehabilitation robotics". In : *Frontiers in neurorobotics* 11 (2017), p. 24.

[Cal+17] Rocco Salvatore Calabrò et al. "Robotic gait training in multiple sclerosis rehabilitation : Can virtual reality make the difference ? Findings from a randomized controlled trial". In : *Journal of the neurological sciences* 377 (2017), p. 25-30.

[Cha+17] Tathagata Chakraborti et al. "Ai challenges in human-robot cognitive teaming". In : *arXiv preprint arXiv :1707.04775* (2017).

[Dev+17] Louise Devigne et al. "Design of an immersive simulator for assisted power wheelchair driving". In : *2017 International Conference on Rehabilitation Robotics (ICORR)*. IEEE. 2017, p. 995-1000.

[For+17] C. Forster et al. "SVO : Semidirect Visual Odometry for Monocular and Multicamera Systems". In : *IEEE Transactions on Robotics* 33.2 (2017), p. 249-265. doi : 10.1109/TRO.2016.2623335.

[Gom+17] Francisco Gomez-Donoso et al. "A robotic platform for customized and interactive rehabilitation of persons with disabilities". In : *Pattern Recognition Letters* 99 (2017), p. 105-113.

[Gre+17] Scott W Greenwald et al. "Technology and applications for collaborative learning in virtual reality". In : Philadelphia, PA : International Society of the Learning Sciences., 2017.

[GLZ17] Kai Gui, Honghai Liu et Dingguo Zhang. “Toward multimodal human–robot interaction to enhance active participation of users in gait rehabilitation”. In : *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 25.11 (2017), p. 2054-2066.

[He+17a] Kaiming He et al. “Mask r-cnn”. In : *Proceedings of the IEEE international conference on computer vision*. 2017, p. 2961-2969.

[He+17b] Kaiming He et al. “Mask r-cnn”. In : *Proceedings of the IEEE international conference on computer vision*. 2017, p. 2961-2969.

[How17] Matt C Howard. “A meta-analysis and systematic literature review of virtual reality rehabilitation programs”. In : *Computers in Human Behavior* 70 (2017), p. 317-327.

[Kl17] Takayuki Kanda et Hiroshi Ishiguro. *Human-robot interaction in social robotics*. CRC Press, 2017.

[KW17] Philip Koopman et Michael Wagner. “Autonomous vehicle safety : An interdisciplinary challenge”. In : *IEEE Intelligent Transportation Systems Magazine* 9.1 (2017), p. 90-96.

[Lav+17] Kate E Laver et al. “Virtual reality for stroke rehabilitation”. In : *Cochrane database of systematic reviews* 11 (2017).

[Li+17] Lan Li et al. “Application of virtual reality technology in clinical medicine”. In : *American journal of translational research* 9.9 (2017), p. 3867.

[Lin+17] Tsung-Yi Lin et al. *Feature Pyramid Networks for Object Detection*. 2017. arXiv : 1612.03144 [cs.CV].

[LP17] Kevin M Lynch et Frank C Park. *Modern robotics*. Cambridge University Press, 2017.

[Men+17] Hamid Menouar et al. “UAV-enabled intelligent transportation systems for the smart city : Applications and challenges”. In : *IEEE Communications Magazine* 55.3 (2017), p. 22-28.

[MT17a] R. Mur-Artal et J. D. Tardós. “ORB-SLAM2 : An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras”. In : *IEEE Transactions on Robotics* 33.5 (2017), p. 1255-1262. doi : 10.1109/TRO.2017.2705103.

[MT17b] R. Mur-Artal et J. D. Tardós. “Visual-Inertial Monocular SLAM With Map Reuse”. In : *IEEE Robotics and Automation Letters* 2.2 (2017), p. 796-803. doi : 10.1109/LRA.2017.2653359.

[OD17] Matthias Oberhauser et Daniel Dreyer. “A virtual reality flight simulator for human factors engineering”. In : *Cognition, Technology & Work* 19 (2017), p. 263-277.

[PN17] Athanasios S Polydoros et Lazaros Nalpantidis. “Survey of model-based reinforcement learning : Applications on robotics”. In : *Journal of Intelligent & Robotic Systems* 86.2 (2017), p. 153-173.

[Sch+17] John Schulman et al. “Proximal policy optimization algorithms”. In : *arXiv preprint arXiv :1707.06347* (2017).

[SH17] Lars Yndal Sørensen et John Paulin Hansen. “A low-cost virtual reality wheelchair simulator”. In : *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments*. 2017, p. 242-243.

[WSC17] Rui Wang, Martin Schworer et Daniel Cremers. “Stereo DSO : Large-scale direct sparse visual odometry with stereo cameras”. In : *Proceedings of the IEEE International Conference on Computer Vision*. 2017, p. 3903-3911.

[WBK17] Jinseok Woo, János Botzheim et Naoyuki Kubota. “System integration for cognitive model of a robot partner”. In : *Intelligent Automation & Soft Computing* (2017), p. 1-14.

[Abi+18] Firas Abi-Farrajl et al. “Humanoid teleoperation using task-relevant haptic feedback”. In : *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, p. 5010-5017.

[Anw+18] Syed Muhammad Anwar et al. “Medical image analysis using convolutional neural networks : a review”. In : *Journal of medical systems* 42 (2018), p. 1-13.

[Bec+18] Sebastian Becker et al. “Comparison of muscular activity and movement performance in robot-assisted and freely performed exercises”. In : *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27.1 (2018), p. 43-50.

[Bes+18] Berta Bescos et al. “DynaSLAM : Tracking, mapping, and inpainting in dynamic scenes”. In : *IEEE Robotics and Automation Letters* 3.4 (2018), p. 4076-4083.

[Cas+18] Sergio Casas et al. “A particle swarm approach for tuning washout algorithms in vehicle simulators”. In : *Applied Soft Computing* 68 (2018), p. 125-135.

[Che+18] Liang-Chieh Chen et al. *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*. 2018. arXiv : 1802.02611 [cs.CV].

[CFP18] Carolina Cruz-Neira, Marcos Fernández et Cristina Portalés. *Virtual reality and games*. 2018.

[EKC18] Jakob Engel, Vladlen Koltun et Daniel Cremers. "Direct Sparse Odometry". In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.3 (2018), p. 611-625. doi : 10.1109/tpami.2017.2658577. url : https://app.dimensions.ai/details/publication/pub.1084824871.

[Far+18] Ana L Faria et al. "Combined cognitive-motor rehabilitation in virtual reality improves motor outcomes in chronic stroke–a pilot study". In : *Frontiers in Psychology* 9 (2018), p. 854.

[Gao+18a] Junfeng Gao et al. *Computer vision in healthcare applications*. 2018.

[Gao+18b] Xiang Gao et al. *LDSO : Direct Sparse Odometry with Loop Closure*. 2018. arXiv : 1808.01111 [cs.CV].

[Gei+18] Robert Geirhos et al. "Generalisation in humans and deep neural networks". In : *Advances in neural information processing systems* 31 (2018).

[He+18] Kaiming He et al. *Mask R-CNN*. 2018. arXiv : 1703.06870 [cs.CV].

[Liu+18] Shu Liu et al. *Path Aggregation Network for Instance Segmentation*. 2018. arXiv : 1803.01534 [cs.CV].

[Mat+18] Hidenobu Matsuki et al. "Omnidirectional DSO : Direct sparse odometry with fisheye cameras". In : *IEEE Robotics and Automation Letters* 3.4 (2018), p. 3693-3700.

[Neu+18] David L Neumann et al. "A systematic review of the application of interactive virtual reality to sport". In : *Virtual Reality* 22 (2018), p. 183-198.

[Pet+18] Brian S Peters et al. "Review of emerging surgical robotic technology". In : *Surgical endoscopy* 32 (2018), p. 1636-1655.

[RNZ18] Muhammad Imran Razzak, Saeeda Naz et Ahmad Zaib. "Deep learning for medical image processing : Overview, challenges and the future". In : *Classification in BioApps : Automation of Decision Making* (2018), p. 323-350.

[RF18] Joseph Redmon et Ali Farhadi. *YOLOv3 : An Incremental Improvement*. 2018. arXiv : 1804.02767 [cs.CV].

[RKD18] James A Reggia, Garrett E Katz et Gregory P Davis. "Humanoid cognitive robots that learn by imitating : implications for consciousness studies". In : *Frontiers in Robotics and AI* 5 (2018), p. 1.

[Ren+18] Carolina Rengifo et al. "Solving the constrained problem in model predictive control based motion cueing algorithm with a neural network approach". In : *Driving Simulation Conference 2018 Europe VR*. 2018, p. 63-69.

[RNC18] Tyler Rose, Chang S Nam et Karen B Chen. "Immersion of virtual reality for rehabilitation-Review". In : *Applied ergonomics* 69 (2018), p. 153-161.

[SC18] William R Sherman et Alan B Craig. *Understanding virtual reality : Interface, application, and design*. Morgan Kaufmann, 2018.

[SS18] Pramila P Shinde et Seema Shah. "A review of machine learning and deep learning applications". In : *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*. IEEE. 2018, p. 1-6.

[SB18] Richard S Sutton et Andrew G Barto. *Reinforcement learning : An introduction*. MIT press, 2018.

[Tan+18] Ana Tanevska et al. "Designing an affective cognitive architecture for human-humanoid interaction". In : *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. 2018, p. 253-254.

[Tie+18] Gaetano Tieri et al. "Virtual reality in cognitive and motor rehabilitation : facts, fiction and fallacies". In : *Expert review of medical devices* 15.2 (2018), p. 107-117.

[Van+18] Jessica Van Brummelen et al. "Autonomous vehicle perception : The technology of today and tomorrow". In : *Transportation research part C : emerging technologies* 89 (2018), p. 384-406.

[Yu+18] Chao Yu et al. "DS-SLAM : A semantic visual SLAM towards dynamic environments". In : *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, p. 1168-1174.

[ZLC18] Han-ye Zhang, Wei-ming Lin et Ai-xia Chen. “Path planning for the mobile robot : A review”. In : *Symmetry* 10.10 (2018), p. 450.

[Zho+18] Fangwei Zhong et al. “Detect-slam : Making object detection and slam mutually beneficial”. In : *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018, p. 1001-1010.

[Aye+19] Ines Ayed et al. “Vision-based serious games and virtual reality systems for motor rehabilitation : A review geared toward a research methodology”. In : *International journal of medical informatics* 131 (2019), p. 103909.

[AW19] Aharon Azulay et Yair Weiss. “Why do deep convolutional networks generalize so poorly to small image transformations ?” In : *J. Mach. Learn. Res.* 20 (2019), 184 :1-184 :25.

[Bin+19] Guo Bingjing et al. “Human–robot interactive control based on reinforcement learning for gait rehabilitation training robot”. In : *International Journal of Advanced Robotic Systems* 16.2 (2019), p. 1729881419839584.

[Bol+19] Daniel Bolya et al. “Yolact : Real-time instance segmentation”. In : *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, p. 9157-9166.

[BK19] Tejas S Borkar et Lina J Karam. “DeepCorrect : Correcting DNN models against image distortions”. In : *IEEE Transactions on Image Processing* 28.12 (2019), p. 6022-6034.

[DMH19] Gabriel Dulac-Arnold, Daniel Mankowitz et Todd Hester. “Challenges of real-world reinforcement learning”. In : *arXiv preprint arXiv :1904.12901* (2019).

[HD19] Dan Hendrycks et Thomas Dietterich. “Benchmarking Neural Network Robustness to Common Corruptions and Perturbations”. In : *Proceedings of the International Conference on Learning Representations* (2019).

[Hen+19] Abdelfetah Hentout et al. “Human–robot interaction in industrial collaborative robotics : a literature review of the decade 2008–2017”. In : *Advanced Robotics* 33.15-16 (2019), p. 764-799.

[Hou+19a] Taha Houda et al. “Dynamic Parameters Optimization and Identification of a Parallel Robot”. In : juin 2019, p. 367-374. isbn : 978-3-030-23131-6. doi : 10.1007/978-3-030-23132-3_44.

[Hou+19b] Taha Houda et al. “Dynamic Parameters Optimization and Identification of a Parallel Robot”. In : *European Congress on Computational Methods in Applied Sciences and Engineering*. Springer. 2019, p. 367-374.

[Hou+19c] Taha Houda et al. “Dynamic parameters optimization of a Gough-Stewart Platform mounted on a 2-DOF moving base”. In : juin 2019.

[Hu+19] Chen-Hui Hu et al. “Skiing Simulation Based on Skill-Guided Motion Planning”. In : *Computer Graphics Forum*. T. 38. 6. Wiley Online Library. 2019, p. 66-78.

[Jia+19a] Licheng Jiao et al. “A Survey of Deep Learning-Based Object Detection”. In : *IEEE Access* 7 (2019), p. 128837-128868. issn : 2169-3536. doi : 10.1109/access.2019.2939201. url : http://dx.doi.org/10.1109/ACCESS.2019.2939201.

[Jia+19b] Licheng Jiao et al. “A survey of deep learning-based object detection”. In : *IEEE access* 7 (2019), p. 128837-128868.

[Kai+19] Lukasz Kaiser et al. “Model-based reinforcement learning for atari”. In : *arXiv preprint arXiv :1903.00374* (2019).

[Ker+19] Florian Kern et al. “Immersive virtual reality and gamification within procedurally generated environments to increase motivation during gait rehabilitation”. In : *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE. 2019, p. 500-509.

[Liu+19a] Guihua Liu et al. “DMS-SLAM : A General Visual SLAM System for Dynamic Scenes with Multiple Sensors”. In : *Sensors* 19.17 (2019), p. 3714.

[Liu+19b] Hanjie Liu et al. “Visual SLAM Based on Dynamic Object Removal”. In : *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE. 2019, p. 596-601.

[Luo+19] Nguyen Cong Luong et al. “Applications of deep reinforcement learning in communications and networking : A survey”. In : *IEEE Communications Surveys & Tutorials* 21.4 (2019), p. 3133-3174.

[Mav+19] Thanassis Mavropoulos et al. “A context-aware conversational agent in the rehabilitation domain”. In : *Future Internet* 11.11 (2019), p. 231.

[Mic+19] Claudio Michaelis et al. “Benchmarking robustness in object detection : Autonomous driving when winter is coming”. In : *CoRR* (2019).

[MM19] Rafael Munoz-Salinas et Rafael Medina-Carnicer. *UcoSLAM : Simultaneous Localization and Mapping by Fusion of KeyPoints and Squared Planar Markers*. 2019. arXiv : 1902.03729 [cs.CV].

[Mur19] Robin R Murphy. *Introduction to AI robotics*. MIT press, 2019.

[NRC19] Roger Gomez Nieto, Hernan Dario Benitez Restrepo et Ivan Cabezas. "How video object tracking is affected by in-capture distortions ?" In : *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, p. 2227-2231.

[OT19] Renz Ocampo et Mahdi Tavakoli. "Visual-haptic colocation in robotic rehabilitation exercises using a 2d augmented-reality display". In : *2019 International Symposium on Medical Robotics (ISMR)*. IEEE. 2019, p. 1-7.

[Pan+19] Sandip Panesar et al. "Artificial intelligence and the future of surgical robotics". In : *Annals of surgery* 270.2 (2019), p. 223-226.

[Qin+19] Tong Qin et al. "A general optimization-based framework for local odometry estimation with multiple sensors". In : *arXiv preprint arXiv :1901.03638* (2019).

[Shi+19] Guruh Fajar Shidik et al. "A systematic review of intelligence video surveillance : trends, techniques, frameworks, and datasets". In : *IEEE Access* 7 (2019), p. 170457-170473.

[SSS19] Shinya Sumikura, Mikiya Shibuya et Ken Sakurada. "OpenVSLAM : A versatile visual SLAM framework". In : *Proceedings of the 27th ACM International Conference on Multimedia*. 2019, p. 2292-2295.

[Wu+19] Erwin Wu et al. "How to vizski : Visualizing captured skier motion in a vr ski training simulator". In : *The 17th International Conference on Virtual-Reality Continuum and its Applications in Industry*. 2019, p. 1-9.

[Zou+19] Zhengxia Zou et al. "Object detection in 20 years : A survey". In : *arXiv preprint arXiv :1905.05055* (2019).

[Ai+20] Yong-bao Ai et al. "Visual SLAM in dynamic environments based on object detection". In : *Defence Technology* (2020).

[Bar+20] Christoph Bartneck et al. *Human-robot interaction : An introduction*. Cambridge University Press, 2020.

[Bes+20] Berta Bescos et al. "DynaSLAM II : Tightly-Coupled Multi-Object Tracking and SLAM". In : *arXiv preprint arXiv :2010.07820* (2020).

[BWL20] Alexey Bochkovskiy, Chien-Yao Wang et Hong-Yuan Mark Liao. *YOLOv4 : Optimal Speed and Accuracy of Object Detection*. 2020. arXiv : 2004.10934 [cs.CV].

[Cam+20] Carlos Campos et al. "ORB-SLAM3 : An accurate open-source library for visual, visual-inertial and multi-map SLAM". In : *arXiv preprint arXiv :2007.11898* (2020).

[Che+20] Junhao Cheng et al. "DM-SLAM : A Feature-Based SLAM System for Rigid Dynamic Scenes". In : *ISPRS International Journal of Geo-Information* 9.4 (2020), p. 202.

[CC20a] KR1442 Chowdhary et KR Chowdhary. "Natural language processing". In : *Fundamentals of artificial intelligence* (2020), p. 603-649.

[Cor+20] Antonio Coronato et al. "Reinforcement learning for intelligent healthcare applications : A survey". In : *Artificial Intelligence in Medicine* 109 (2020), p. 101964.

[CC20b] Sebastian Cygert et Andrzej Czyżewski. "Toward robust pedestrian detection with data augmentation". In : *IEEE Access* 8 (2020), p. 136674-136683.

[Dag20] Evren Daglarli. "Computational modeling of prefrontal cortex for meta-cognition of a humanoid robot". In : *IEEE Access* 8 (2020), p. 98491-98507.

[Del+20] Robby van Delden et al. "VR4VRT : Virtual reality for virtual rowing training". In : *Extended Abstracts of the 2020 Annual Symposium on Computer-Human Interaction in Play*. 2020, p. 388-392.

[Fio+20] Laura Fiorini et al. "Multidimensional evaluation of telepresence robot : results from a field trial". In : *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE. 2020, p. 1211-1216.

[Hou20] Taha Houda. "Human Interaction in a large workspace parallel robot platform with a virtual environment". Thèse de doct. université Paris-Saclay, 2020.

[JH20] Mohd Javaid et Abid Haleem. "Virtual reality applications toward medical field". In : *Clinical Epidemiology and Global Health* 8.2 (2020), p. 600-605.

[Kha+20a] Zohaib Amjad Khan et al. "Residual networks based distortion classification and ranking for laparoscopic image quality assessment". In : *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2020, p. 176-180.

[Kha+20b] Zohaib Amjad Khan et al. “Towards a video quality assessment based framework for enhancement of laparoscopic videos”. In : *Medical Imaging 2020 : Image Perception, Observer Performance, and Technology Assessment*. T. 11316. SPIE. 2020, p. 129-136.

[KK20] Suji Kim et Eunjoo Kim. “The use of virtual reality in psychiatry : a review”. In : *Journal of the Korean Academy of Child and Adolescent Psychiatry* 31.1 (2020), p. 26.

[KS20] Bing Cai Kok et Harold Soh. “Trust in robots : Challenges and opportunities”. In : *Current Robotics Reports* 1 (2020), p. 297-309.

[Koy+20] Ahmet Burakhan Koyuncu et al. “A novel approach to neural network-based motion cueing algorithm for a driving simulator”. In : *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2020, p. 2118-2125.

[Liu+20] Li Liu et al. “Deep learning for generic object detection : A survey”. In : *International journal of computer vision* 128.2 (2020), p. 261-318.

[Qaz+20] Mohammad Reza Chalak Qazani et al. “Prepositioning of a land vehicle simulation-based motion platform using fuzzy logic and neural network”. In : *IEEE Transactions on Vehicular Technology* 69.10 (2020), p. 10446-10456.

[Rad+20] Jaziar Radianti et al. “A systematic review of immersive virtual reality applications for higher education : Design elements, lessons learned, and research agenda”. In : *Computers & Education* 147 (2020), p. 103778.

[Ran+20] René Ranftl et al. “Towards robust monocular depth estimation : Mixing datasets for zero-shot cross-dataset transfer”. In : *IEEE transactions on pattern analysis and machine intelligence* 44.3 (2020), p. 1623-1637.

[Ria+20] Hassam Riaz et al. “Scene understanding for safety analysis in human-robot collaborative operations”. In : *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*. IEEE. 2020, p. 722-731.

[Sho+20] Nur Aqilah Shoib et al. “Rowing Simulation using Rower Machine in Virtual Reality”. In : *2020 6th International Conference on Interactive Digital Media (ICIDM)*. IEEE. 2020, p. 1-6.

[TPL20] Mingxing Tan, Ruoming Pang et Quoc V Le. “Efficientdet : Scalable and efficient object detection”. In : *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, p. 10781-10790.

[TA20] Matthew Turk et Vassilis Athitsos. “Gesture recognition”. In : *Computer vision : a reference guide* (2020), p. 1-6.

[Vai+20a] Guillaume Vailland et al. “Power Wheelchair Virtual Reality Simulator with Vestibular Feedback”. In : *Modelling, measurement and control C* 81 (2020), p. 35-42.

[Vai+20b] Guillaume Vailland et al. “Vestibular feedback on a virtual reality wheelchair driving simulator : A pilot study”. In : *Proceedings of the 2020 ACM/IEEE International conference on human-robot interaction*. 2020, p. 171-179.

[Wan+20] Chien-Yao Wang et al. “CSPNet : A new backbone that can enhance learning capability of CNN”. In : *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020, p. 390-391.

[Wen+20] Wang Wendong et al. “Design and verification of a human–robot interaction system for upper limb exoskeleton rehabilitation”. In : *Medical engineering & physics* 79 (2020), p. 19-25.

[Wu+20] Erwin Wu et al. “Vr alpine ski training augmentation using visual cues of leading skier”. In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, p. 878-879.

[Zac+20] Angeliki Zacharaki et al. “Safety bounds in human robot interaction : A survey”. In : *Safety science* 127 (2020), p. 104667.

[Zha+20] Tianwei Zhang et al. “FlowFusion : Dynamic Dense RGB-D SLAM Based on Optical Flow”. In : *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, p. 7322-7328. doi : 10.1109/ICRA40945.2020.9197349.

[Bal+21] Irene Ballester et al. “DOT : dynamic object tracking for visual SLAM”. In : *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, p. 11705-11711.

[Che+21] Xiangning Chen et al. “Robust and accurate object detection via adversarial learning”. In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, p. 16622-16631.

[Jay21] Prashani Jayasingha. “Nao humanoid for physical therapy rehabilitation”. In : (2021).

[Jur+21] Anđela Jurić et al. “A comparison of graph optimization approaches for pose estimation in SLAM”. In : *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*. IEEE. 2021, p. 1113-1118.

[Kyr+21] Maria Kyrarini et al. “A survey of robots in healthcare”. In : *Technologies* 9.1 (2021), p. 8.

[Li+21] Ao Li et al. “DP-SLAM : A visual SLAM with moving probability towards dynamic environments”. In : *Information Sciences* 556 (2021), p. 128-142.

[LM21a] Yubao Liu et Jun Miura. “RDMO-SLAM : Real-time visual SLAM for dynamic environments using semantic label prediction with optical flow”. In : *IEEE Access* 9 (2021), p. 106981-106997.

[LM21b] Yubao Liu et Jun Miura. “RDS-SLAM : Real-Time Dynamic SLAM Using Semantic Segmentation Methods”. In : *IEEE Access* 9 (2021), p. 23772-23785. doi : 10.1109/ACCESS.2021.3050617.

[Min+21] Shervin Minaee et al. “Image segmentation using deep learning : A survey”. In : *IEEE transactions on pattern analysis and machine intelligence* (2021).

[Ort+21] Jessica S Ortiz et al. “Virtual reality-based framework to simulate control algorithms for robotic assistance and rehabilitation tasks through a standing wheelchair”. In : *Sensors* 21.15 (2021), p. 5083.

[San21] Haute Autorité de Santé. “Rééducation et réadaptation de la fonction motrice de l’appareil locomoteur des personnes diagnostiquées de paralysie cérébrale”. In : *HAS : octobre* (2021).

[Shi+21] Di Shi et al. “Human-centred adaptive control of lower limb rehabilitation robot based on human–robot interaction dynamic model”. In : *Mechanism and Machine Theory* 162 (2021), p. 104340.

[Win+21] Carla Winter et al. “Immersive virtual reality during gait rehabilitation increases walking speed and motivation : a usability evaluation with healthy participants and patients with multiple sclerosis and stroke”. In : *Journal of neuroengineering and rehabilitation* 18.1 (2021), p. 68.

[Xie+21] Biao Xie et al. “A review on virtual reality skill training applications”. In : *Frontiers in Virtual Reality* 2 (2021), p. 645153.

[Yar+21] Denis Yarats et al. “Improving sample efficiency in model-free reinforcement learning from images”. In : *Proceedings of the AAAI Conference on Artificial Intelligence*. T. 35. 12. 2021, p. 10674-10681.

[BM22] Ayman Beghdadi et Malik Mallem. “A comprehensive overview of dynamic visual SLAM and deep learning : concepts, methods and challenges”. In : *Machine Vision and Applications* 33.4 (2022), p. 54.

[BMB22a] Ayman Beghdadi, Malik Mallem et Lotfi Beji. “Benchmarking performance of object detection under image distortions in an uncontrolled environment”. In : *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2022, p. 2071-2075.

[BMB22b] Ayman Beghdadi, Malik Mallem et Lotfi Beji. “D2SLAM : Semantic visual SLAM based on the influence of Depth for Dynamic environments”. In : *arXiv preprint arXiv :2210.08647* (2022).

[De +22] Giorgio De Magistris et al. “Vision-Based Holistic Scene Understanding for Context-Aware Human-Robot Interaction”. In : *AIxIA 2021–Advances in Artificial Intelligence : 20th International Conference of the Italian Association for Artificial Intelligence, Virtual Event, December 1–3, 2021, Revised Selected Papers*. Springer. 2022, p. 310-325.

[FZL22] Junming Fan, Pai Zheng et Shufei Li. “Vision-based holistic scene understanding towards proactive human–robot collaboration”. In : *Robotics and Computer-Integrated Manufacturing* 75 (2022), p. 102304.

[Gom+22] Roger Gomez-Nieto et al. “Quality Aware Features for Performance Prediction and Time Reduction in Video Object Tracking”. In : *IEEE Access* 10 (2022), p. 13290-13310.

[RJ22] Shyam Nandan Rai et CV Jawahar. “Removing atmospheric turbulence via deep adversarial learning”. In : *IEEE Transactions on Image Processing* 31 (2022), p. 2633-2646.

[Sta22] Lev A Stankevich. “Cognitive Technologies and Artificial Mind for Humanoid Robots”. In : *Advances in Neural Computation, Machine Learning, and Cognitive Research V : Selected Papers from the XXIII International Conference on Neuroinformatics, October 18-22, 2021, Moscow, Russia*. Springer. 2022, p. 3-8.

[Sze22] Richard Szeliski. *Computer vision : algorithms and applications*. Springer Nature, 2022.

[Wei22] Markus Weinberger. “What Is Metaverse ?—A Definition Based on Qualitative Meta-Synthesis”. In : *Future Internet* 14.11 (2022), p. 310.

[Yan+22] Li Yan et al. “Dgs-slam : A fast and robust rgbd slam in dynamic environments combined by geometric and semantic information”. In : *Remote Sensing* 14.3 (2022), p. 795.

[AML23] Beghdadi Ayman, Mallem Malik et Beji Lotfi. “DAM-SLAM : depth attention module in a semantic visual SLAM based on objects interaction for dynamic environments”. In : *Applied Intelligence* (2023), p. 1-14.

[Beg+23] Ayman Beghdadi et al. “CD-COCO : A Versatile Complex Distorted COCO Database for Scene-Context-Aware Computer Vision”. In : *2023 11th European Workshop on Visual Information Processing (EUVIP)*. 2023, p. 1-6. doi : 10.1109/EUVIP58404.2023.10323035.

[Hou+23a] Taha Houda et al. “Complex Multi Robot Hybrid Platform for Augmented Virtual Skiing Immersion”. In : *3rd IFSA Winter Conference on Automation, Robotics & Communications for Industry 4.0/5.0 (ARCI 2023)*. 2023.

[Hou+23b] Taha Houda et al. “Multi-Complex Robot Interaction with Homothetic Human Feedback in a Virtual Environment”. In : *Sensors & Transducers* 260.2 (2023), p. 15-24.

[Lin+23] Zhongqi Lin et al. “ML-CapsNet meets VB-DI-D : A novel distortion-tolerant baseline for perturbed object recognition”. In : *Engineering Applications of Artificial Intelligence* 120 (2023), p. 105937.

[Sch+23] Hendrik Scheidel et al. *A novel approach of a deep reinforcement learning based motion cueing algorithm for vehicle driving simulation*. 2023. arXiv : 2304.07600 [cs.RO].

[Zou+23] Zhengxia Zou et al. “Object detection in 20 years : A survey”. In : *Proceedings of the IEEE* (2023).

A - Résultats complémentaires

A.1 . Perception du mouvement propre via MCA : profil d'accélération 1

La figure suivante reprend les résultats de chacune des approches par rapport à la vitesse et la position de la trajectoire référence selon uniquement les informations inertielles pour le profil d'accélération perçue suivant :

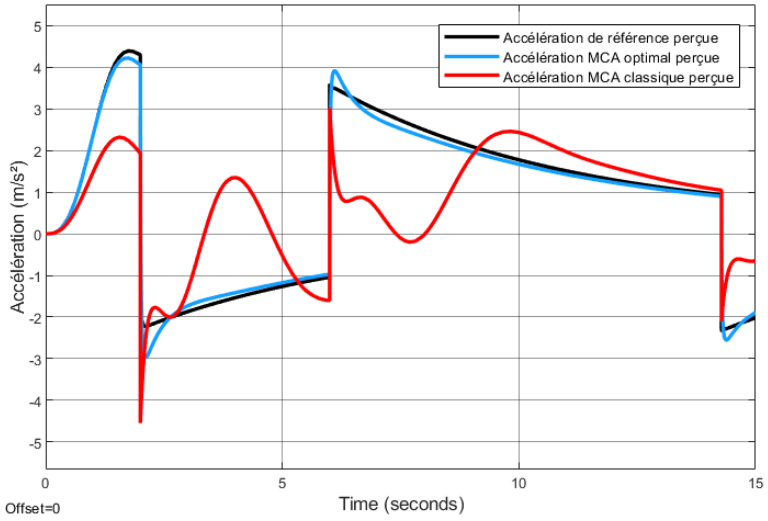
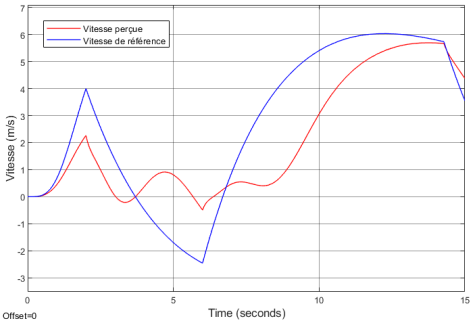
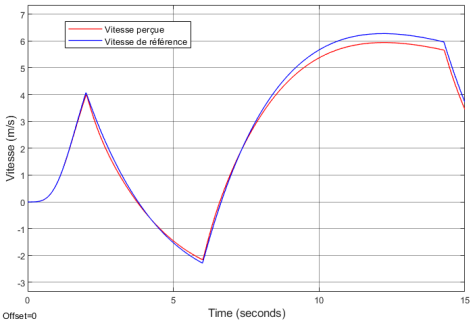


Figure A.1 – Accélération restituée perçue via les approches classique et optimale.

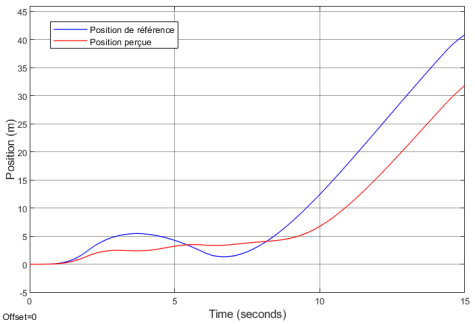
À travers le modèle de perception visuo-inertiel, nous obtenons les perceptions de vitesse et position suivantes pour les approches classique et optimale.



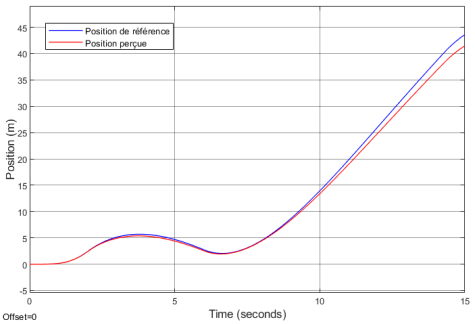
(a) Vitesse perçue selon l'approche classique.



(b) Vitesse perçue selon l'approche optimale.



(c) Position perçue selon l'approche classique.

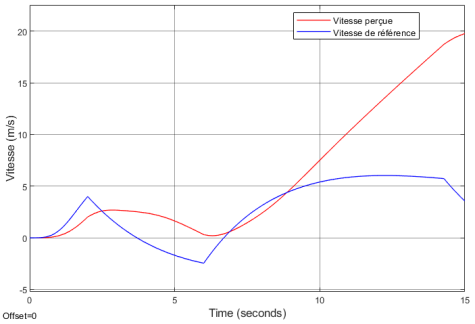


(d) Position perçue selon l'approche optimale.

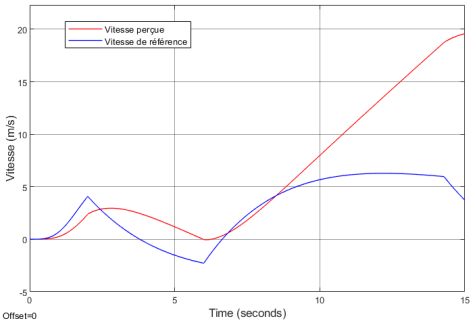
Figure A.2 – Comparatif des perceptions inertielles des approches classique et optimale par rapport à la trajectoire de référence.

A.2 . Perception du mouvement propre via MCA : profil d'accélération 2

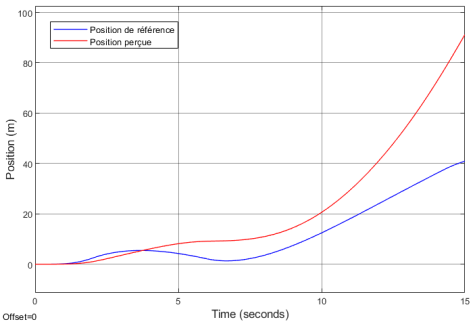
La figure suivante reprend les résultats de chacune des approches par rapport à la vitesse et la position de la trajectoire référence selon uniquement les informations inertielles pour le profil d'accélération perçue suivant. À travers le modèle de perception visuo-inertiel, nous obtenons les perceptions de vitesse et position suivantes pour les approches classique et optimale.



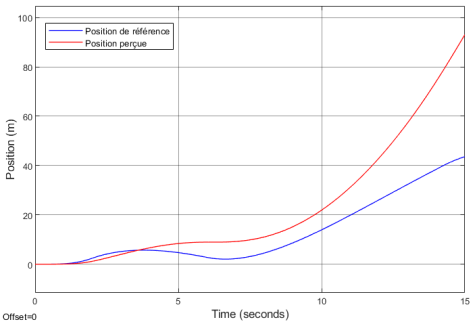
(a) Vitesse perçue selon l'approche classique.



(b) Vitesse perçue selon l'approche optimale.



(c) Position perçue selon l'approche classique.



(d) Position perçue selon l'approche optimale.

Figure A.3 – Comparatif des perceptions visuo-inertielles des approches classique et optimale par rapport à la trajectoire de référence.

A.3 . Perception du mouvement propre via MCA : profil d'accélération 3

La figure suivante reprend les résultats de chacune des approches par rapport à la vitesse et la position de la trajectoire référence selon uniquement les informations inertielles pour le profil d'accélération perçue. À travers le modèle de perception visuo-inertiel, nous obtenons les perceptions de vitesse et position suivantes pour les approches classique et optimale.

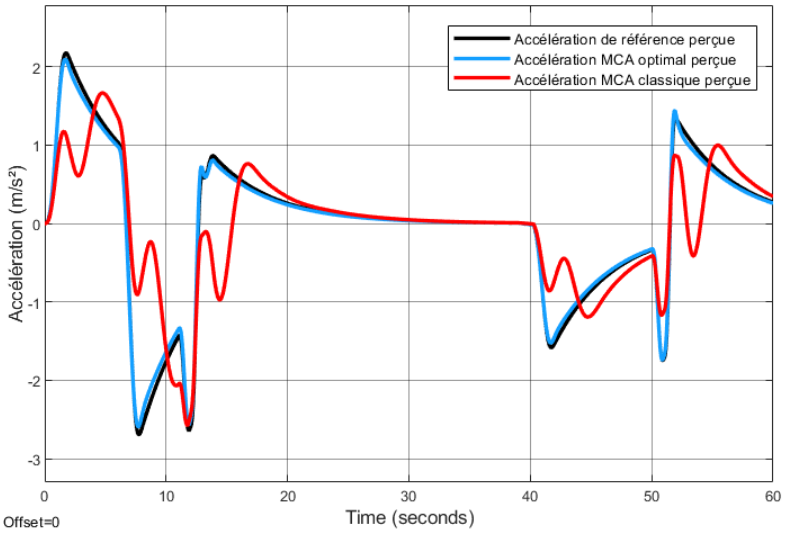
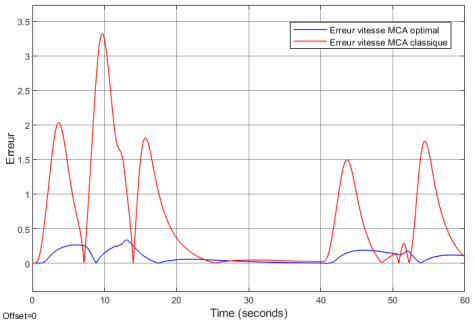
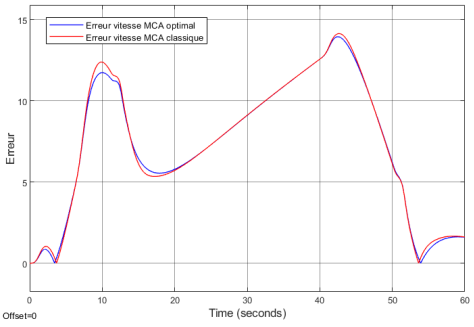


Figure A.4 – Accélération restituée perçue via les approches classique et optimale.

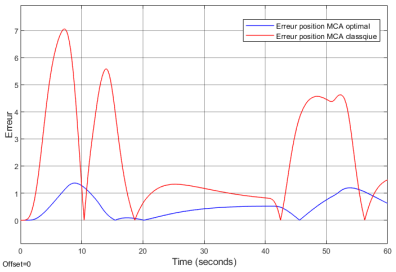


(a) Erreur de vitesse perçue selon le modèle inertiel.

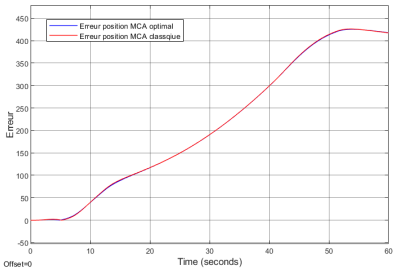


(b) Erreur de vitesse perçue selon le modèle visuo-inertiel.

Figure A.5 – Comparatif des erreurs de perception de vitesse selon les modèles inertielles et visuo-inertielles.

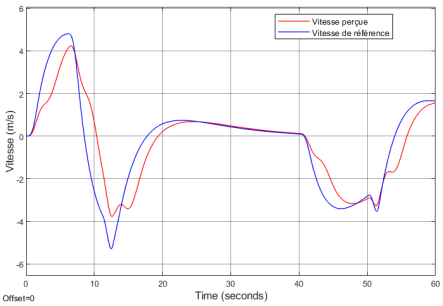


(a) Erreur de position perçue selon le modèle inertiel.

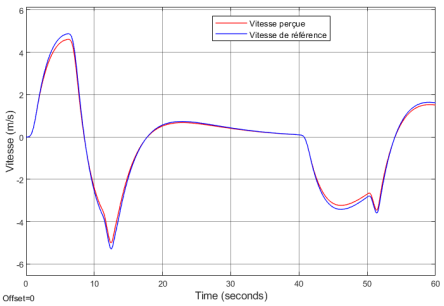


(b) Erreur de position perçue selon le modèle visuo-inertiel.

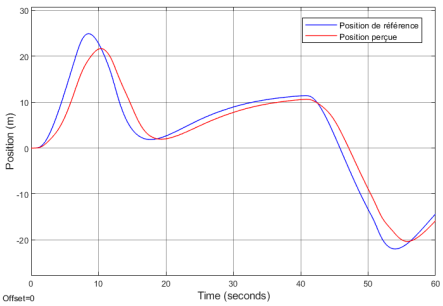
Figure A.6 – Comparatif des erreurs de perception de position selon les modèles inertielles et visuo-inertielles.



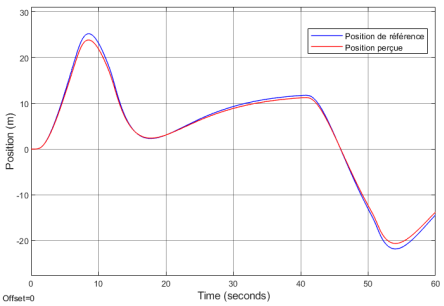
(a) Vitesse perçue selon l'approche classique.



(b) Vitesse perçue selon l'approche optimale.

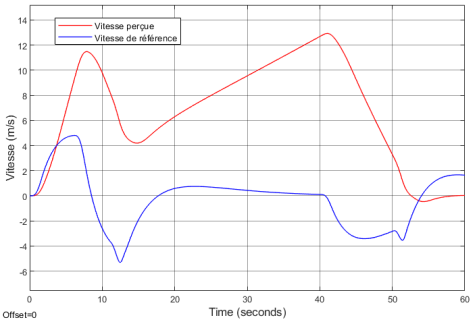


(c) Position perçue selon l'approche classique.

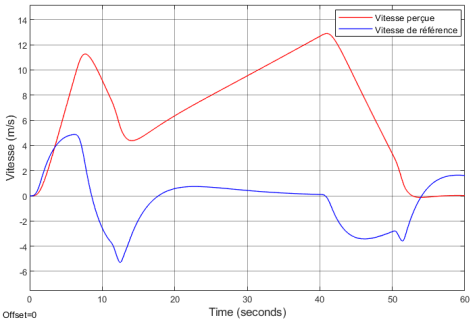


(d) Position perçue selon l'approche optimale.

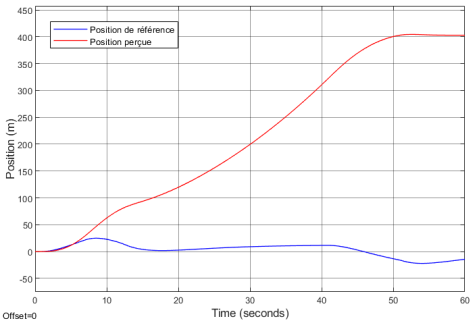
Figure A.7 – Comparatif des perceptions inertielles des approches classique et optimale par rapport au profil d'accélération 2 de référence.



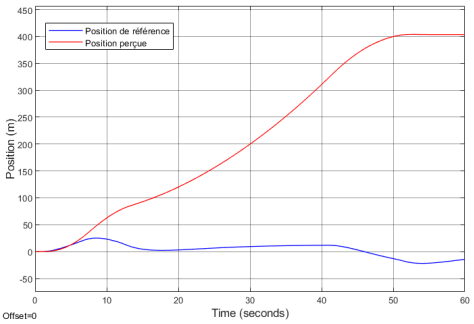
(a) Vitesse perçue selon l'approche classique.



(b) Vitesse perçue selon l'approche optimale.



(c) Position perçue selon l'approche classique.



(d) Position perçue selon l'approche optimale.

Figure A.8 – Comparatif des perceptions visuo-inertielles des approches classique et optimale par rapport au profil d'accélération 2 de référence.

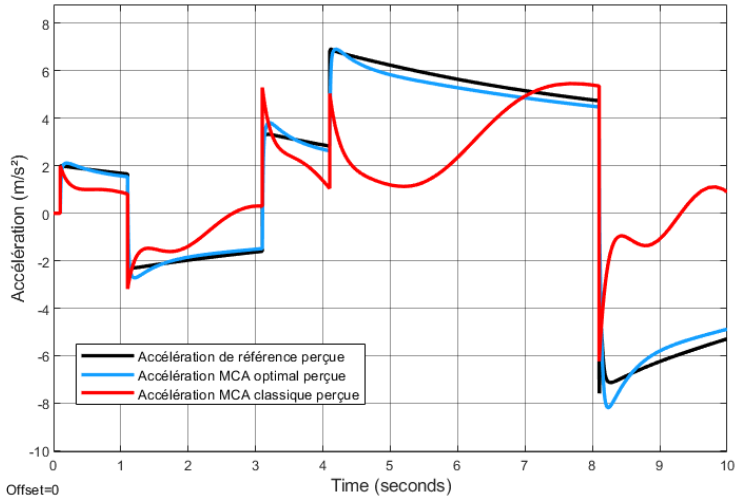
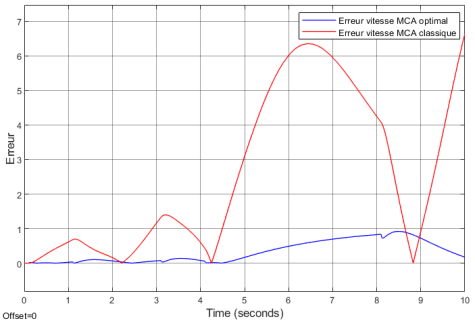
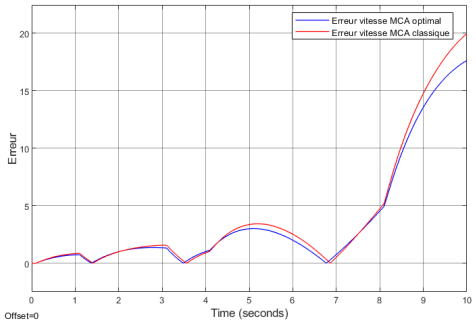


Figure A.9 – Accélération restituée perçue via les approches classique et optimale.

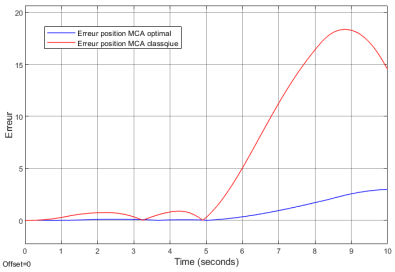


(a) Erreur de vitesse perçue selon le modèle inertiel.

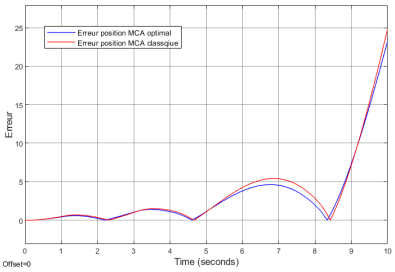


(b) Erreur de vitesse perçue selon le modèle visuo-inertiel.

Figure A.10 – Comparatif des erreurs de perception de vitesse selon les modèles inertielles et visuo-inertielles.

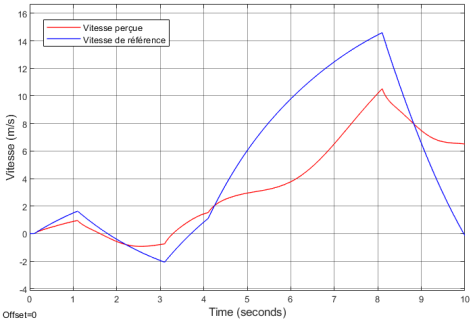


(a) Erreur de position perçue selon le modèle inertiel.

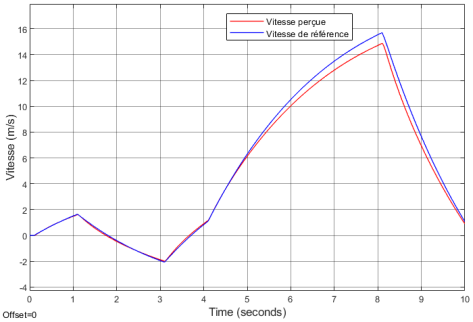


(b) Erreur de position perçue selon le modèle visuo-inertiel.

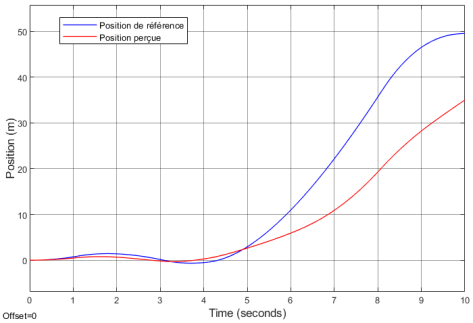
Figure A.11 – Comparatif des erreurs de perception de position selon les modèles inertielles et visuo-inertielles.



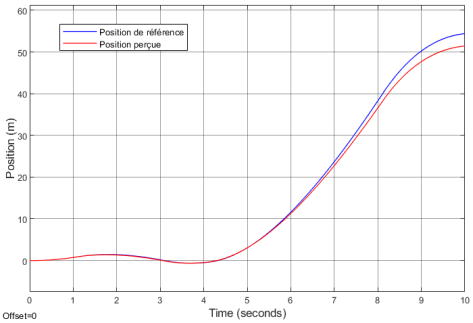
(a) Vitesse perçue selon l'approche classique.



(b) Vitesse perçue selon l'approche optimale.

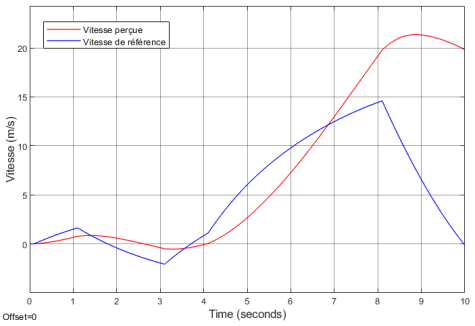


(c) Position perçue selon l'approche classique.

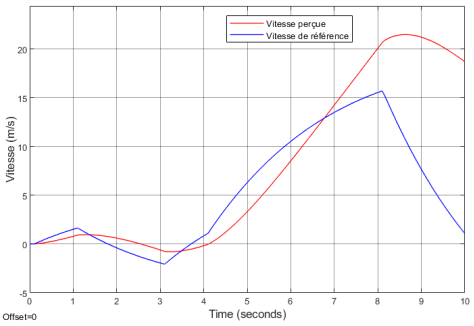


(d) Position perçue selon l'approche optimale.

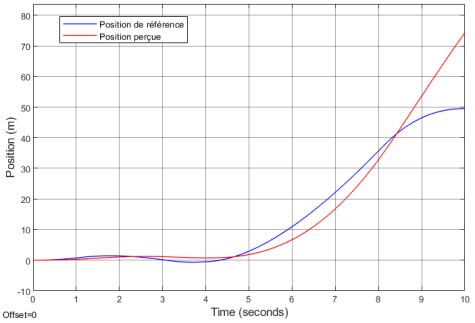
Figure A.12 – Comparatif des perceptions inertielles des approches classique et optimale par rapport au profil d'accélération 3 de référence.



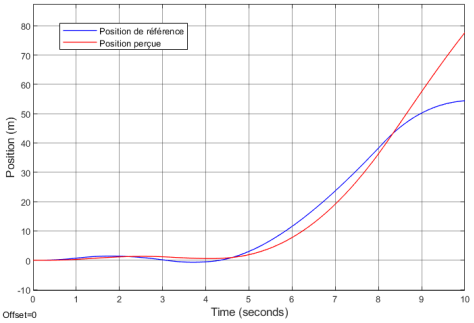
(a) Vitesse perçue selon l'approche classique.



(b) Vitesse perçue selon l'approche optimale.



(c) Position perçue selon l'approche classique.



(d) Position perçue selon l'approche optimale.

Figure A.13 – Comparatif des perceptions visuo-inertielles des approches classique et optimale par rapport au profil d'accélération 3 de référence.

B - Outils pour la vision par ordinateur

B.1 . Fondamentaux de la vision par ordinateurs

Les points d'intérêts sont les primitives de base de la vision par ordinateur. Ils servent de base d'information pour décrire des éléments caractéristiques présents dans une image au niveau pixel afin d'effectuer des tâches de bas à haut niveau de traitement d'image et vision par ordinateur. Ils sont acquis à la suite d'un processus de détection de leur présence suivi d'une caractérisation de leur particularité permettant de les reconnaître et de distinguer selon une métrique connue.

B.1.1 . Détecteurs

Les détecteurs sont des algorithmes de vision par ordinateur permettant d'extraire les points d'intérêts d'une image selon plusieurs approches. Une première approche a été l'étude de leur variation de niveau de gris pour un certain voisinage à travers des méthodes s'appuyant sur des fonctions d'auto-corrélation approximées par l'utilisation de la méthode de la somme des moindres carrés (SSD). La seconde est basée sur le calcul des gradients d'une image selon une ou plusieurs orientations, ce qui permet de détecter les régions entourant un coin ou un bord. Les coins sont retenus, car répondent aux critères fondamentaux de fiabilité des points d'intérêts, à savoir : leurs distractivités, localités, qualités, précisions, fiabilités, robustesses, invariances et répétabilités. En effet, il est important d'établir le processus d'extraction des points d'intérêts sur des éléments caractérisants d'une image qui puisse être facilement identifiable et reconnaissable sur d'autres images liées. La méthode de Harris [HS+88] est la plus connue de la première catégorie, elle repose sur la calcul du changement d'intensité pour une fenêtre d'observation $w(x, y)$, un certain décalage $[u, v]$ et une intensité donnée $I(x, y)$. Ce changement est donné par l'équation suivante :

$$E(u, v) = \sum_{x,y} w(x, y)[I(x + u, y + v) - I(x, y)]^2 \tag{B.1}$$

L'approximation quadratique de $E(u, v)$ est obtenue par le développement en série de Taylor au voisinage de o :

$$E(u,v) = \sum_{x,y} w(x,y)[u^2 I_x^2 + 2uvI_xI_y + v^2 I_y^2] \approx [uv]M \begin{pmatrix} u \\ v \end{pmatrix} \tag{B.2}$$

La matrice M de l'équation (B.2) étant le moment d'ordre 2 obtenue grâce aux dérivées partielles $\frac{\partial^2 E}{\partial^2 u}$ et $\frac{\partial^2 E}{\partial^2 v}$. Nous pouvons alors formulé M par :

$$M = \sum_{x,y} w(x,y) \begin{pmatrix} I_x^2 & I_xI_y \\ I_y^2 & I_yI_x \end{pmatrix} = \begin{pmatrix} \sum I_x^2 & \sum I_xI_y \\ \sum I_y^2 & \sum I_yI_x \end{pmatrix} = \sum \begin{pmatrix} I_x \\ I_y \end{pmatrix} [I_xI_y] = \sum \nabla I (\nabla I)^T \tag{B.3}$$

Ainsi, si les valeurs propres λ_i sont petites et approximativement égale alors la zone d'observation ne correspond pas à un coin, à l'inverse, si les valeurs propres sont élevés et approximativement égale alors il s'agit d'un coin. Enfin, si l'une des valeurs propres est beaucoup plus grande par rapport à l'autre alors il s'agit d'un bord. Afin d'accélérer le processus, le calcul des valeurs propres est délaissé au profit du calcul de la fonction suivante avec α fixé :

$$R = det(M) - \alpha trace(M)^2 = \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2 \tag{B.4}$$

En conséquence, la fenêtre d'observation correspond à un coin si $R > 0$ pour un certain seuil fixé. Enfin, une suppression des non-maxima locaux de la fonction R seuillé est effectuée pour ne conserver que les points les plus pertinents localement et éviter les doublons. L'utilisation d'une fenêtre d'observation donnée fixe pose un problème par rapport aux changements d'échelle de l'image ou la détection de point d'intérêts à des niveaux différents. La méthode SIFT (Scale Invariant Feature Transform) [Low99] introduite en 1999 permet de contourner ce problème en appliquant la construction d'une pyramide multi-échelle de l'image qui permet la détection des extremas, la localisation des points caractéristiques et l'affectation d'une orientation en vue du calcul de leurs descripteurs locaux. Il convient alors de déterminer les extremas dans l'espace-échelle Gaussien d'une image I, cette espace-échelle est défini par la fonction suivante :

$$L(x,y,\sigma) = G(x,y,\sigma*) * I(x,y) \tag{B.5}$$

$$G = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{B.6}$$

Avec σ le paramètre d'échelle . La fonction DoG (Difference of Gradient) est préférée à la fonction LoG pour trouver ces extrema en raison de son temps de calcul plus faible, nous notons l'utilisation d'un paramètre fixe k :

$$DoG(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma) \tag{B.7}$$

Pour finir, une étape de seuillage est effectuée afin d'éliminer les extremas trouvés ayant un faible contraste. Enfin, un rejet des contours est appliqué en utilisant le Hessien de l'image obtenue à l'aide du DoG et l'algorithme de Harris. Enfin, l'algorithme Fast (Features from Accelerated Segment Test) [RD06] est une méthode de détection de coin permettant l'extraction de points caractéristiques pertinent. Cette méthode applique une comparaison de luminosité entre un pixel p et les 16 pixels de son cercle de voisinage. Ces pixels sont classés en fonction de leurs différences de luminosité (plus foncé que p, similaire ou plus clair). Si ce point p possède au moins 8 pixels dans son voisinage présentant une différence de luminosité (plus clair ou foncé) alors ce point sera considéré comme caractéristique. Cependant, l'algorithme FAST n'applique aucune étude multi-échelle de l'image ainsi plusieurs points voisins peuvent être détectés, il est alors nécessaire d'utiliser un processus de suppression des non-maximas détectés uniquement les points adjacents les plus pertinent et caractéristique.

Détecteur	**Invariance**			**Qualité**			
	Rotation	Affine	Scale	Répétabilité	Localité	Fiabilité	Robustesse
Harris	Oui	Non	Non	+++	+++	++	+++
Harris Laplace	Oui	Non	Oui	+++	+++	+	++
Shi-Tomasi	Oui	Non	Non	+++	+++	++	+++
SUSAN	Oui	Non	Non	++	++	+++	++
Hessian	Oui	Non	Non	+++	+++	+	+++
DoG	Oui	Non	Oui	+++	+++	+++	++
FAST	Oui	Non	Non	++	+++	+++	++
SIFT	Oui	Non	Oui	+++	+++	+++	++
SURF	Oui	Non	Oui	++	+++	++	+++
ORB	Oui	Non	Oui	+++	++	++++	+++
BRISK	Oui	Non	Oui	++	+	++++	++

Table B.1 – Classification des algorithmes par performance

En fonction des besoins et du contexte, il est alors plus simple de choisir l'algorithme adéquat même si certains paraissent plus efficaces de manière générale telle que les détecteurs de Harris, MSER, ORB [Rub+11], Shi-Tomasi [ST00] et SIFT [Low99].

B.1.2 . Descripteurs

Un descripteur extrait les informations d'intensités dans la région autour du point d'intérêt provenant du détecteur afin de caractériser ce point. Dans ce cas, une caractéristique est une métrique ou une valeur quantifiable qui est utilisée pour décrire les points d'intérêts d'une image en perspective de

haut niveau.

Il est important de rappeler qu'il est indispensable d'utiliser des points d'intérêts respectant la propriété de répétabilité. Cette caractéristique est très souvent liée à la couleur, la texture, les formes et les coins qui sont contenus dans l'image. La recherche et l'étude des descripteurs est un axe majeur des travaux dans le domaine de la vision par ordinateur, les sources scientifiques y sont très nombreuses [33, 34, 35, 36]. Les descripteurs ont l'avantage d'être distinctif, robuste aux occlusions et ne nécessitent pas de segmentation préalable. Les processus de détection et description sont indispensables à la mise en place d'un système de mise en correspondance d'images entres elles. Ils peuvent être regroupés dans un tableau pour avoir une meilleure compréhension de leurs caractéristiques : D'après la littérature, les descrip-

Descripteurs	Invariances aux rotations	Invariances aux changements d'echelles	Temps de process	Compacité
SIFT	Oui	Oui	++	+
SURF	Oui	Oui	++	++
ORB	Oui	Oui	++++	+++
BRISK	Oui	Oui	++++	+++

Table B.2 – Synthèse des descripteurs

teurs BRISK [LCS11] et l'ORB [Rub+11], tous deux inclus dans la catégorie des blob détections, présentent les meilleures performances. Le descripteur ORB s'appuie sur le détecteur de points d'intérêts FAST [RD06] et le descripteur BRIEF [Cal+10]. La complémentarité de ces deux algorithmes permet de corriger les limitations de la méthode FAST grâce à l'apport des caractérisations de la méthode BRIEF et du calcul multi-échelle de l'ORB. En effet, l'algorithme FAST n'applique aucune étude multi-échelle de l'image ni aucun calcul des orientations du point d'intérêt détecté, tous deux nécessaire à une étude pertinente des informations pixel de l'image et à la bonne caractérisation des éléments détectés.

Le descripteur ORB effectue donc une transformation en pyramide multi-échelle de l'image où chaque niveau de la pyramide correspond à un sous-échantillonnage par rapport au niveau de la pyramide précédente. Ce processus permet d'acquérir des informations élargissant indirectement le champ de comparaison de luminosité appliqué dans l'algorithme FAST à des voisinages plus grand en raison de l'échantillonnage s'opérant à chaque niveau de la pyramide. Ainsi, ce processus supplémentaire rend l'ORB invariant aux changements d'échelle.

Une fois les points d'intérêts localisés, l'algorithme ORB calcule l'orientation de chacun de ces points en étudiant la manière dont les changements de

luminosité s'opèrent au voisinage de ce point. La détection du changement de luminosité est obtenue en utilisant le centroïde d'intensité qui caractérise l'intensité d'un coin et considère que cette intensité peut être décalée par rapport à son centre. Le vecteur résultant est alors exploité pour calculer l'orientation. Les moments d'une observation sont donnés par l'équation :

$$m_{pq} = \sum_{x,y} x^p y^q T(x,y) \tag{B.8}$$

Ces moments permettent ensuite de calculer le centroïde C représentant le centre vectoriel des changements de luminosité de l'observation tel que :

$$C = (\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}) \tag{B.9}$$

L'orientation finale du descripteur correspondant au point d'intérêt correspond alors au vecteur entre le centre du coin noté O et le centroïde C calculé précédemment. L'orientation est donnée par le calcul : $\theta = \mathrm{atan2}(m_{01}, m_{10})$
L'algorithme BRIEF convertit ensuite l'ensemble des points détectés précédemment et les convertit en un vecteur de caractéristiques binaires de 128 à 512 bits permettant une caractérisation robuste des points. Cet algorithme applique un filtre gaussien à l'image pour éliminer le bruit haute fréquence puis sélectionne une paire de pixels aléatoirement au voisinage du point d'intérêt détecté selon une certaine fenêtre d'observation afin d'étudier la différence de luminosité entre ces deux points aléatoires. Si le premier pixel est plus lumineux que le second alors il attribut la valeur 1 au bit correspondant à cette paire aléatoire sinon 0. L'attribution de la valeur binaire τ suit l'algorithme suivant :

$$\tau(p;x,y) = \{ \begin{array}{lcl} 1 & : & p(x) < p(y) \\ 0 & : & p(x) \geq p(y) \end{array} \tag{B.10}$$

Avec $p(x)$ l'intensité lumineuse du pixel x.

Ce processus est ensuite répété n fois compris entre 128 et 512 en fonction du nombre de bits du vecteur caractéristique binaire. Le vecteur caractéristique binaire est alors défini comme :

$$f(n) = \sum_{1<i<n} 2^{i-1} \tau(p;x_i,y_i) \tag{B.11}$$

Enfin, pour rendre le processus invariant aux rotations, le descripteur ORB utilise une variante de la méthode BRIEF nommée rBRIEF (Rotation-aware BRIEF). A cette fin, une matrice S de taille $2xn$ est attribuée au vecteur caractéristique binaire de chacun des pixels (x_i, y_i) :
$S = \begin{pmatrix} x_1 & \cdots & x_n \\ y_1 & \cdots & y_n \end{pmatrix}$
L'orientation Θ calculée par l'algorithme ORB et sa matrice de rotation correspondante noté R_θ sont utilisés pour construire un correspondant S_Θ de S considérant l'orientation, et donné par :
$S_\theta = R_\theta S$
Le vecteur caractéristique obtenu par la méthode BRIEF correspondant au vecteur caractéristique binaire est alors donné par la relation suivante :

$$g_n(R, \Theta) := f_n(p) | (x_i, y_i) \in S_\Theta \tag{B.12}$$

Il est important de noter qu'il est exprimé selon des orientations. Il discrétise ensuite l'angle par un pas de $2/30$ pour construire une table des caractéristiques observés qui est comparée à une table pré-construite pour vérifier la cohérence des observations.

B.1.3 . Mise en correspondance de points d'intérêts

La mise en correspondance de points d'intérêts est basée sur la comparaison de points à travers des formulations mathématiques permettant de calculer la "distance" ou les ressemblances entre leurs caractéristiques. Elle permet d'effectuer des tâches de vision par ordinateur et robotique de plus haut niveau tel que le suivi de points et d'objets, le calcul des transformations homographiques entre des prises de vue et des tâches relevant entre autres de la géométrie épipolaire. Une des premières approches est d'estimer la ressemblance entre deux points en calculant leurs différences d'intensité à l'aide de la méthode de la somme des différences au carré (SSD) appelé également erreur quadratique moyenne. Cette méthode formalise la "distance" entre les caractéristiques de deux points k et k' de deux images respectives I et I', de la manière suivante :

$$S(k, k') = \sum_{k,k'} (I(x_k, y_k) - I'(x_{k'}, y_{k'}))^2 \tag{B.13}$$

La valeur de cette différence quadratique (B.13) indique la similarité entre ces points. Ainsi plus la valeur sera proche de 0 plus les points seront correspondants, et inversement. L'approche la plus performante est l'algorithme non-déterministe RANSAC (RANdom SAmple Consensus) [FB81] qui estime les paramètres d'un modèle mathématique de manière itérative. Cette méthode probabiliste a été initialement introduite dans le domaine de la vision par ordinateur et la robotique pour l'estimation de la position de la caméra, la vérification du bon appariement de points et l'estimation des transformations homographique entre deux scènes. Dans notre cas, les entrées de l'algorithme sont les paires supposées de points correspondants qui sont alors supposés soit comme "inliers" soit comme "outliers" en fonction de s'ils vérifient ou non la transformation s'opérant dans le modèle que l'on tente d'estimer. L'algorithme ajuste alors l'estimation de la transformation de sorte qu'elle s'ajuste à un nombre fixé "d'inliers" pertinent selon l'évaluation de l'estimation de l'erreur de ce modèle pour les données considérées "inlier". Le modèle est ré-évalué itérativement lorsque l'erreur obtenue est plus petite que la précédente. Lorsque l'erreur est jugé suffisamment faible ou que le nombre d'itération est atteint, le meilleur modèle estimé est retourné pour être la solution de notre jeu de données. Le seuil à partir duquel l'erreur est suffisamment faible est fixé, de même, le nombre "d'inliers" minimal pour que le modèle estimer est pertinent pour le jeu de données est donné.

B.2 . Géométrie appliquée à la vision par ordinateur

B.2.1 . Géométrie projective

Il existe plusieurs modèles de géométrie projective pour la formation d'image en vision par ordinateur. Le modèle "pin-hole", le plus simple d'entre eux, applique une approximation qui place le centre optique de la caméra à l'origine du repère 3D et le plan image devant ce centre. Il est alors possible d'utiliser le théorème de Thales afin de passer de coordonnées monde 3D aux coordonnées images qui sont 2D. Dans le cas de ce modèle, le passage de coordonnées monde x, y et z aux coordonnées image u et v s'exprime par rapport à la distance focale f de la manière suivante :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f\frac{x}{z} \\ f\frac{y}{z} \\ 1 \end{pmatrix} \tag{B.14}$$

De manière générale, le modèle géométrique sans l'approximation du modèle "pin-hole" nécessite d'avoir la matrice des paramètres extrinsèques $[R|T]$ qui permet le passage du repère objet au repère caméra (CoP) et la matrice

des paramètres intrinsèques de la caméra noté C. Les matrices R et T représentent respectivement les matrices de rotation et translation. La matrice R reprend les rotations de passage d'un repère à l'autre selon les axes de rotation R_x, R_y et R_z.

$$R = R_x R_y R_z = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\theta & sin\theta \\ 0 & -sin\theta & cos\theta \end{pmatrix} \begin{pmatrix} cos\phi & 0 & sin\phi \\ 0 & 1 & 0 \\ -sin\phi & 0a & cos\phi \end{pmatrix} \begin{pmatrix} cos\gamma & \sin\gamma & 0 \\ -sin\gamma & cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{B.15}$$

La matrice de rotation R issue de (B.15) et la matrice de translation T s'écrivent alors :

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \quad et \quad T = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \tag{B.16}$$

La matrice des paramètres intrinsèques, C est formulée ci-après, où les changements d'échelles pixel/mm horizontale et verticale sont notées respectivement α et β, la longueur focale f et les valeurs du centre de l'axe optique u_0 et v_0 :

$$C = \begin{pmatrix} f\alpha & 0 & u_0 \\ 0 & f\beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{B.17}$$

Le passage de coordonnées monde notées X, Y et Z aux coordonnées image u et v s'exprime alors en exploitant les équations (B.15) et (B.17) de la manière suivante :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = C \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R & T \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = C\,(R|T) \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = P \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{B.18}$$

La matrice de passage notée P permet le passage du repère objet au repère image en prenant en compte les paramètres intrinsèques et extrinsèques de la caméra, où P est noté : $P = C[R|T]$.

B.2.2 . Géométrie épipolaire pour la localisation

La géométrie épipolaire est un procédé de vision appliquée à la robotique basé sur les relations géométriques existantes entre plusieurs vues caméra

d'un même point 3D. Elle s'appuie sur la mise en correspondance d'un même point 3D entre deux différentes images, qui est formalisé par une matrice permettant la projection d'un point image dans l'autre image sous la forme d'une droite dite "ligne épipolaire" qui contient le point correspondant. La matrice de projection d'un repère caméra à l'autre se note différemment en fonction de notre connaissance des paramètres intrinsèques de la caméra :

- Matrice Fondamentale F : paramètres intrinsèques inconnus.
- Matrice Essentielle E : paramètres intrinsèques connus.

La géométrie épipolaire offre la possibilité d'effectuer plusieurs tâches liées à la vision par ordinateur, à savoir la calibration de la caméra, la reconstruction projective 3D, le calcul de pose de la caméra, la triangulation et la vérification de la bonne mise en correspondance entre deux points.
Dans le cas de la matrice fondamentale $F_{3\times3}$, la relation entre un point image q et son correspondant q' respecte la contrainte de projection suivante : $q'^T F q = 0$.
Cette contrainte assure la possibilité d'estimer la vraisemblance de la mise en correspondance de points et/ou d'estimer la matrice fondamentale ou essentielle permettant le passage d'un repère caméra à l'autre.
Les paramètres intrinsèques C de la caméra n'étant pas connus, la matrice fondamentale correspond alors à la projection selon l'ensemble des paramètres C, R et T de la caméra. La relation fondamentale se note alors :

$$q'^T C'^{-T}.[T \times (RC^{-1}q)] = 0 \tag{B.19}$$

À l'aide de propriétés de l'algèbre linéaire, nous pouvons formuler le produit vectoriel $\times$ sous la forme d'une multiplication de matrice/vecteur tel que :

$$m \times n = \begin{pmatrix} 0 & -m_z & m_y \\ m_z & 0 & -m_x \\ -m_y & m_x & 0 \end{pmatrix} \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} = [m_\times]n \tag{B.20}$$

La formulation (B.19) peut alors être réexprimée sous la forme : $q'^T C'^{-T}[T_\times]RC^{-1}q) = 0$
La matrice fondamentale F est donc donnée par l'expression : $F = C'^{-T}[T_\times]RC^{-1}$.
La ligne épipolaire l' projeté du point q dans le repère caméra C' est obtenue par la formule $l' = Fq$, et la ligne épipolaire l projeté du point q' dans le repère caméra C par $l = F^T q'$.
De la même manière, dans le cas de la matrice essentielle $E_{3\times3}$, la relation de correspondance respecte la propriété de reprojection $q'^t Eq = 0$, la matrice essentielle peut ainsi se noter : $E = [T_\times]R$. Les lignes épipolaires respectives l et l' sont données par : $l = E^t q'$ et $l' = Eq$.
L'estimation des matrices fondamentale $\widetilde{F}$ et essentielle $\widetilde{E}$ est réalisée par

l'utilisation de la méthode des 8 points [Lon81] effectuer de manière itérative à travers l'algorithme RANSAC [FB81]. La méthode des 8 points est basée sur la minimisation de l'erreur de reprojection Ĉ pour au moins 8 paires de points provenant d'une même paire d'image selon les propriétés de reprojection $q'^t Eq = 0$ et $q'^t Fq = 0$.
Elle formalise le problème comme une équation linéaire homogène qui lie les paires de points d'après la propriété essentielle précédente où les points q'_i, q_i et la matrice essentielle E sont notés respectivement :

$$q'_i = \begin{pmatrix} q'_{i_u} \\ q'_{i_v} \\ 1 \end{pmatrix}, \quad q_i = \begin{pmatrix} q_{i_u} \\ q_{i_v} \\ 1 \end{pmatrix}, \quad E = \begin{pmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{pmatrix} \tag{B.21}$$

La contrainte de reprojection épipolaire est alors exprimée sous la forme de l'équation linéaire homogène suivante :

$$q'_{iu}q_{iu}e_{11}+q'_{iu}q_{iv}e_{12}+q'_{iu}e_{13}+q'_{iv}q_{iu}e_{21}+q'_{iv}q_{iu}e_{22}+q'_{iv}e_{23}+q_{iu}e_{31}+q_{iv}e_{32}+e_{33} = 0 \tag{B.22}$$

Minimiser l'erreur de reprojection Ĉ en fonction de la matrice essentielle $\widetilde{E}$ permet alors d'estimer en s'appuyant sur la formulation de la contrainte (B.22) , cette minimisation est formulée par :

$$\min_{\widetilde{E}} \mathcal{C}(\widetilde{E}), \quad avec \quad \mathcal{C}(\widetilde{E}) = \sum_{i=1}^{m} (q_i'^t \widetilde{E} q_i)^2 = \sum_{i=1}^{m} \left((q'_{iu}, q'_{iv}, 1) \widetilde{E} \begin{pmatrix} q_{iu} \\ q_{iv} \\ 1 \end{pmatrix} \right)^2 \tag{B.23}$$

Le nombre de paire de points est indiqué par la variable m avec $m \geq 8$ et la paire de points par i. La minimisation de l'erreur de reprojection Ĉ donnée dans (B.23) est reformulée de la même manière que l'équation (B.22) pour vectoriser l'erreur de reprojection Ĉ :

$$\mathcal{C}(\widetilde{E}) = \sum_{i=1}^{m} (a_i^t \widetilde{e})^2 \tag{B.24}$$

Avec a_i^t et $\widetilde{e}$ qui sont exprimés de la manière suivante :

$$\begin{array}{rcl} a_i^t & = & (q'_{iu}q_{iu}q_{iu}q'_{iv}q_{iu}q_{iv}q'_{iu}q'_{iv}q_{iv}q_{iv}q'_{iu}q'_{iv}1) \\ \widetilde{e} & = & (e_{11}e_{12}e_{13}e_{21}e_{22}e_{23}e_{31}e_{32}e_{33}) \end{array} \tag{B.25}$$

La somme des erreurs de reprojection peut alors se noter :

$\mathcal{C}(\widetilde{E}) = ||A\widetilde{e}||^2$, avec $A = \begin{pmatrix} a_1^t \\ \vdots \\ a_m^t \end{pmatrix}$

L'erreur de projection est ensuite minimisée sous la contrainte $||\widetilde{E}|| = ||\widetilde{e}|| =$

1 à l'aide du vecteur singulier correspondant à la plus petite valeur du vecteur singulier A obtenue précédemment. Cette valeur minimale est obtenue en appliquant une décomposition en valeurs singulières (SVD) de A afin d'en extraire $||\widetilde{e}||$. La décomposition de A s'écrit alors :

$$A_{(mx9)} = U_{(mx9)}\ D_{(mx9)}\ V_{(mx9)} = U \begin{pmatrix} \sum_1 & 0 & 0 \\ 0 & \sum_2 & 0 \\ 0 & 0 & 0 \end{pmatrix} V^T$$

Ce procédé est effectué de manière itérative à travers l'algorithme RANSAC qui sélectionne l'estimation de la matrice essentielle $\widetilde{E}$ avec l'erreur de projection la plus petite et donc le plus de candidats dits "inlier" pertinents durant son processus. La dernière colonne de V donne la matrice essetielle E.

B.3 . Triangulation

La triangulation est un procédé très répandu dans la vision par ordinateur. Il utilise les matrices de projection (également appelées matrice essentielle ou fondamentale) pour calculer des points 3D correspondants à des appariements de points image 2D provenant de deux images ou plus. Il est très utilisé en odométrie visuelle et estimation de structure de scène. En théorie, la projection de points correspondants 2D dans le repère 3D doivent se confondre, cependant, en raison des possibles erreurs d'estimation des matrices de projection, appariements de points images et bruit de mesure, il peut apparaitre une certaine distance entre les reprojections de points. La reconstruction par triangulation tente donc conjointement d'estimer la structure et de minimiser l'erreur de reprojection à l'aide

Idéalement, les points 3D doivent se trouver au point d'intersection des rayons rétroprojetés. Cependant, en raison du bruit de mesure, les rayons rétroprojetés ne se croisent généralement pas. Ainsi, les points 3D doivent être choisis de manière à minimiser une métrique d'erreur. L'algorithme de reconstruction "gold standard" minimise la somme des carrés erreurs entre les positions d'image mesurées et prédites du point 3D dans toutes les vues dans lesquelles il est visible, c'est-à-dire

B.4 . Structure from Motion (SfM)

La méthode Structure from Motion (SfM) [Ull79] est une technique de vision par ordinateur qui estime la structure tridimensionnelle d'une scène à travers un nuage de points à partir d'une séquence d'image bidimensionnelle résultant du mouvement de la caméra. Cette méthode tente de projeter des informations image 2D dans un espace 3D en prenant en compte les contraintes de temps et mouvement lié au séquencement des images et au mouvement de la caméra. Cette approche séquentielle tente d'intégrer les

contraintes géométriques épipolaires à travers les matrices fondamentales ou essentielles pour une séquence de vue. Les vues successives sont utilisées une par une en prenant en compte la précédente afin d'estimer la position des points 3D présents dans cette paire de vue et ainsi déterminé la structure et la pose respective à cette paire. Les estimations suivantes sont obtenues par le même procédé, mais en affinant également l'estimation à l'aide des points 3D de la structure déjà construits. Une première initialisation est faite entre les deux premières vues de la séquence par l'emploi des contraintes épipolaires afin d'estimer la pose de la caméra grâce au calcul de la matrice essentielle. L'ensemble de ce processus est affiné de manière itérative à l'aide de l'algorithme de $BundleAdjustment$ (BA) [Tri+00] ce qui permet d'optimiser les estimations de la structure et de la pose en minimisant une fonction de coût décrite dans la section **??**.

B.5 . Ajustement des faisceaux

Comme expliqué précédemment, l'estimation de la structure en fonction d'une certaine pose est obtenue par l'utilisation de la méthode SfM, mais cette approche nécessite une optimisation à travers une méthode non-linéaire nommé $BundleAdjustment$. La méthode BA est une méthode itérative qui minimise une fonction de coût représentant l'erreur pondérée de reprojection au carré. Le BA permet alors d'obtenir une estimation conjointe et optimale de la structure et de la pose de la caméra respectant les contraintes épipolaires retranscrites dans l'estimation de l'erreur de reprojection. Cette approche remet en perspective l'estimation selon le nombre de points 3D observés n dans un certain nombre de vues m. Cette minimisation est formulée avec x_{ij} défini comme la projection du ième point dans l'image j, W_{ij} un poids binaire qui informe de la présence ou non du point i dans l'image j, c_j correspondant au vecteur représentant la pose de la caméra j et p_i le vecteur correspondant au point 3D i :

$$\min_{c_j,p_i} \sum_{i=1}^{n} \sum_{j=1}^{m} W_{ij} d(\Pi(c_j, p_i), x_{ij})^2$$

La fonction de projection est définie par Π et la distance euclidienne par d qui représente l'écart entre le point 3D p_i et la projection de son point 2D équivalent x_{ij}. L'application de cette minimisation à de nombreux points et images permet de généraliser le procédé malgré de possibles projections de points à des images manquantes.

C - Outils pour l'apprentissage profond

C.1 . Réseau de neurones convolutifs

De manière générale, les modèles de segmentation d'objet sont basés sur les réseaux de neurones convolutifs. Ils sont composés d'un réseau principal qui consiste en une succession de couches de convolution, d'activation et de normalisation qui sont par moments suivies de couches de regroupement qui permet la synthèse de l'extraction des informations. Nous détaillerons par la suite le fonctionnement et l'intérêt de chacune de ces types de couches.

Couche de convolution

La couche de convolution est l'élément central des réseaux de neurones convolutifs, car ils sont les couches responsables de l'extraction et la caractérisation des informations. Cette couche est un produit scalaire entre une portion de l'image d'entrée et un filtre de kernel dit noyau. Ce noyau parcourt l'image afin d'extraire une représentation caractéristique de l'image pour chacun des canaux de l'image (RVB). Cette représentation bi-dimensionnelle appelée carte de caractéristiques est alors utilisée comme entrée pour la couche suivante du réseau. Pour une image de taille WxW et une profondeur de canal D, une taille de fenêtre glissante F d'un pas noté S et un remplissage du filtre noté P, la taille de la carte de caractéristiques résultante est exprimé par :

$$W_{out} = \frac{W - F + 2P}{S} + 1 \tag{C.1}$$

Le fonctionnement de l'opération de convolution s'opérant dans une couche de convolution est illustré dans la figure C.1. La couche de convolution per-

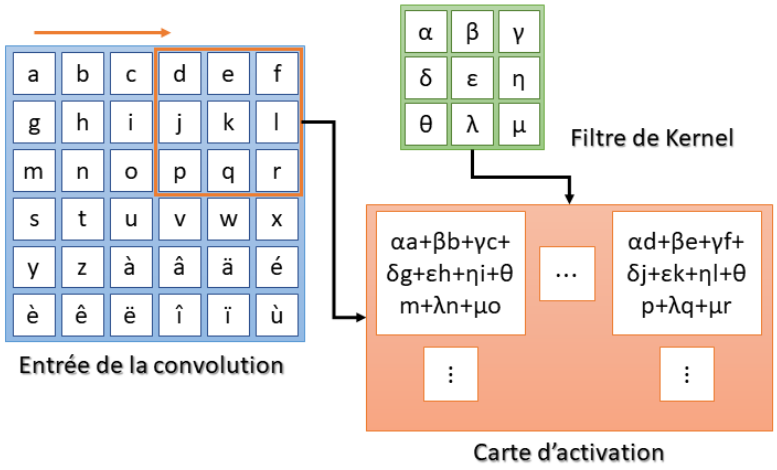


Figure C.1 – Opération résultant d'une couche de convolution

met de considérer plusieurs fois les éléments de la matrice d'entrée offrant l'avantage que les paramètres sont partagés. Ainsi, pour une carte de caractéristiques, les neurones partagent le même ensemble de poids. Cette propriété d'équivariance garantit la relation entre l'entrée et la sortie.

Couche d'activation

Les couches d'activation consistent en l'application de fonctions dites d'activation qui vont générer la carte d'activation issue de la carte de caractéristiques de la couche convolutive précédente. C'est la fonction qui permet d'activer ou non le neurone correspondant permettant ainsi l'apprentissage. La sortie d'une fonction d'activation notée $f(z)$ prend alors en entrée les sorties de la carte de caractéristiques x_i, des poids correspondant W_i et d'un biais donné b combinés en z comme ci-après :

$$z = \sum_i W_i x_i + b \ | f(z) \tag{C.2}$$

Le type de fonction (ReLu, sigmoid, tanh, etc...) influe sur la réaction du modèle pour l'activation du neurone et sur les caractéristiques souhaités de la fonction d'activation (voir figure C.2). De manière générale, une fonction d'activation doit satisfaire les conditions ou résoudre les problèmes suivants :

- Problème de fuite de gradient : l'entraînement des réseaux de neurones se fait à l'aide la méthode de la descente de gradient qui tente d'optimiser la recherche d'une solution. L'apprentissage dans les réseaux de neurones se fait par la propagation vers l'avant puis vers l'arrière des informations pour affecter des valeurs aux poids de manière itérative afin d'approcher de la solution souhaitée. Cependant, pour certaines fonctions, lors de la rétro-propagation qui trouve les dérivées du réseau en déplaçant couche par couche de la couche de la fin de celui-ci vers le début, la multiplication des dérivés peut provoquer une diminution exponentielle du gradient qui ne permettra plus la convergence du modèle. Un gradient petit implique de fait que les biais et poids ne sont pas mis à jour et donc que l'apprentissage ne se fait pas. Pour résoudre ce problème, il faut soit utilisé une fonction qui n'a pas ce problème, soit faire précéder l'activation d'une fonction de normalisation de la carte de caractéristiques.
- Centrage en zéro : la sortie d'une fonction d'activation doit être centrée en zéro pour éviter une dérive du gradient.
- Complexité de calcul : plus la fonction d'activation est complexe, plus elle requiert de temps de calcul. Ainsi, les fonctions peu coûteuses en temps de calcul sont privilégiées ().

Couche de Pooling

La couche de Pooling réalise une synthèse de la couche précédente par l'ap-

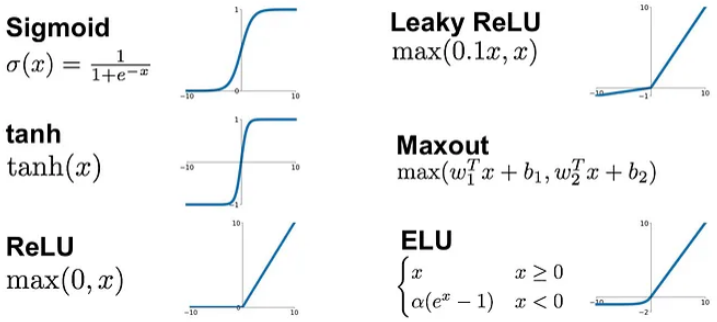


Figure C.2 – Fonctions d'activation

plication d'une opération statistique définie. Cette opération présente l'avantage d'extraire une information récapitulative qui réduit alors la taille de la représentation de la couche d'entrée (carte d'activation). Les opérations statistiques sont appliquées dans une fenêtre glissante rectangulaire sans chevauchement de la carte d'activation. Il existe plusieurs fonctions aux propriétés différentes, la moyenne du voisinage (appelé averagePooling) extrait la valeur moyenne incluse dans la fenêtre d'observation, la moyenne pondérée (appelé GlobalAveragePooling) ou encore la recherche de la valeur maximale (appelé maxPooling). Pour une carte d'activation de taille $WxWxD$, un poo-

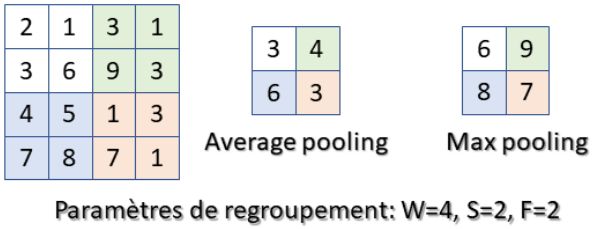


Figure C.3 – Opération résultant d'une couche de regroupement (pooling)

ling de kernel de taille F, un pas S, la taille spatiale W_{out} de la couche de sortie et sa profondeur D_{out} s'expriment par :

$$W_{out} = \frac{W - F}{S} + 1, \quad D_{out} = D \tag{C.3}$$

Ces fonctions sont illustrées dans la figure C.3, où la taille de la représentation est bien diminuée par rapport à celle d'entrée.

D - Outils pour la robotique

D.1 . Robot Humanoïde NAO

Le robot NAO est un système robotique humanoïde programmable conçu en 2006 par la société Aldebaran. Ce robot humanoïde a été développé au cours du temps sous 5 versions (V6, V5, V4, V3.3 et V3.2) avec des caractéristiques techniques assez similaire, mais des améliorations au fil des versions.

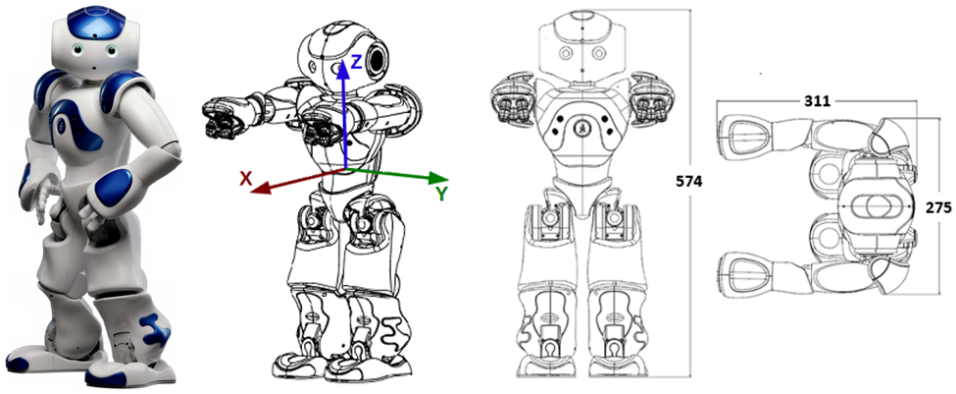


Figure D.1 – Robot humanoïde NAO.

La version V5 utilisée dans nos travaux mesure 57.4cm pour 5.4kg offrant en tout 25 degrés de liberté. Elle intègre un processeur ATOM Z530 1.6 GHz de 1 GB de RAM et une communication possible par cable ethernet RJ45 (10/100/1000 base T) ou wifi via la norme IEEE 802.11 a/b/g/n. La version 5 de NAO a une batterie interne de capacité d'énergie 48.6 Wh offrant une autonomie de 1h en cas d'utilisation intensive et 1h30 en cas d'utilisation normale. Les caractéristiques du robot NAO sont résumées dans le tableau D.1.
La figure D.3 illustre l'arborescence de la structure robotique de NAO où l'ensemble de articulations sont regroupées en trois sous ensemble :

- Corps : inclut l'ensemble des articulations dont les actionneurs des mains.
- Chaînes : correspond aux articulations et actionneurs du segment désigné (tête, bras, jambe) par côté (gauche/droite).
- Effecteurs : inclus tous les noms des chaînes plus le torse.

L'arborescence permet une prise en main ciblée des articulations du robot à travers son framework de programmation. Il est ainsi possible de déterminer précisément le segment à considérer (localement pour les articulations ou globalement pour une chaîne donnée) afin d'optimiser la récupération de

Table D.1 – Caractéristiques du robot NAO.

Caractéristique	Critère	Valeur
Corps	Taille (m) Poids (kg)	0.57 4.5
Batterie	Type Capacité	Lithium-ion 49 Wh
Degrés de liberté : 25	Tête Bras Bassin Jambes Mains	2 5 X 2 1 5 X 2 1 X 2

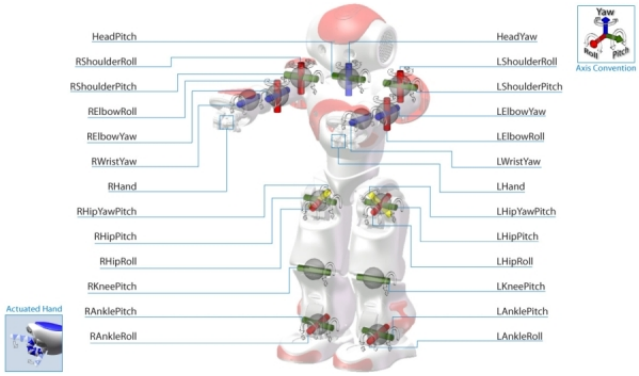


Figure D.2 – Actionneurs du robot NAO.

données capteur ou l'instruction d'actionneurs.

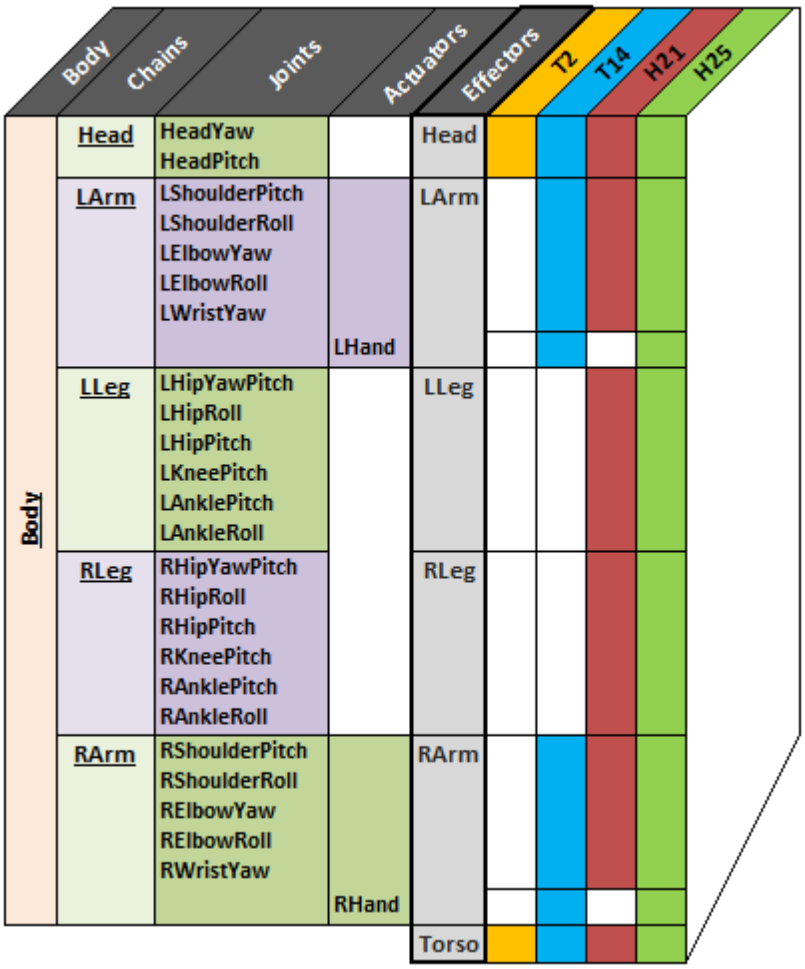


Figure D.3 – Arborescence du robot humanoïde NAO.

D.1.1 . Capteurs et actionneurs

Le robot NAO intègre une multitude de capteurs et actionneurs offrant une capacité d'action et d'interaction considérable. NAO possède des capteurs dédiés à la vision, la parole et l'écoute, la proprioception et la détection d'obstacle. Le tableau D.2 récapitule l'ensemble des capteurs du robot NAO en fonction de leur type de données.
Chacune des articulations du robot NAO est composée d'un ou de plusieurs actionneurs permettant le mouvement du corps. L'ensemble de ces actionneurs se décomposent sous la forme de moteurs de 4 types différents ayant des rapports de réduction de vitesse de type A ou B.

Les caractéristiques des 4 types de moteurs et de leurs types (A/B) de rapports de réduction de vitesse sont décrit dans le tableau D.3. En se référant

Table D.2 – Capteurs du robot NAO.

Type	Capteur	Nombre	Précision/caractéristi
Vision	vidéocaméra CMOS 30FPS	2	1.22 Mp, 1288x968, 61° &
Sonore	Microphone Haut-parleur	4 2	20mV/Pa +/-3dB à 1KHz (150
Inertiel	Accéléromètres Gyroscope	3 3	2g 500°/s
Articulaire	Codeur rotatif magnétique	34	12 bits, soit 0.1°
Haptique	Résistances sensibles à la force	8	0 - 25 N
Onde	Sonar	2	0,20 m - 0,80 m, 60° (4
Lumineux	LED	51	
Communication	Infra-rouge (E/R)	2	940 nm, +/- 60°, 8 m

à la figure D.3 qui fournit une vue d'ensemble de chacun des segments de la structure du robot NAO en fonction des articulations, nous pouvons synthétiser dans le tableau D.4 l'ensemble des actionneurs par segment corporel (Chaîne = Articulations + Actionneurs).

D.1.2 . Framework NAOqi

NAOqi est le framework utilisé pour la programmation du robot NAO garantissant un accès aux ressources du robot ainsi qu'une parallélisation et une synchronisation des tâches. Ce framework est programmé en C++ et Python, et peut être intégré dans un processus sous ROS (Robot Operating System) permettant la synchronisation avec des composants externes au robot NAO. L'Interface de programmation (API) NAOqi est composée de modules aux fonctions définies, exécutables en temps réel et en parallèle, répondant chacun à des besoins spécifiques de la robotique. Dans notre cas d'application, les principaux modules utilisés sont :

- NAOqi Core : qui est le module en charge de la gestion du robot d'un point de vue système, dont sa mémoire. Dans notre cas, nous utilisons particulièrement un de ses sous-modules nommé *ALMemory* dédié à la gestion de la mémoire et des informations stockées tels que les données capteurs et actionneurs.
- NAOqi Motion : qui est alloué à la gestion du mouvement du robot NAO. Il se décompose en plusieurs sous-modules dont *ALMotion* permettant le déplacement du robot selon une approche contrôle ou une approche

Table D.3 – Caractéristiques des moteurs et réducteurs des actionneurs.

Type de moteur	M1	M2	M3	M4
Modèle	22NT82213P	17N88208E	16GT83210E	DCX16S01G
Vitesse à vide	8 300 rpm ±10%	8 400 rpm ±12%	10 700 rpm ±10%	12 700 rpm
Couple de décrochage	68 mNm ±8%	9.4 mNm ±8%	14.3 mNm ±8%	22.4 mNm
Couple nominal	16.1 mNm	4.9 mNm	6.2 mNm	5.53 m
	Rapports de réduction de vitesse			
Type A	201.3	50.61	150.27	150.2
Type B	130.85	36.24	173.22	

réaction d'événement. Il offre également la possibilité de définir le comportement du robot lorsqu'il est inactif et d'avoir accès aux informations relatives aux composants du robot (articulations, limites, etc.).

L'API NAOqi est un exécutable de la forme "*broker*" qui charge un fichier de préférences appelé *autoload.ini* qui définit l'ensemble des bibliothèques à charger pour donner accès aux différentes méthodes contenues dans chacun des modules (voir figure D.4).

Le *broker* permet l'indexation de chacun des modules dans une arborescence afin que chacune de leurs méthodes puissent être trouvées. Rappelons qu'un *broker* est un objet qui fournit à la fois des services d'indexation permettant l'accès aux modules et à leurs méthodes, et un accès au réseau garantissant la possibilité d'appeler des méthodes depuis un processus externe.
La figure D.5 illustre le lien existant entre *Broker*, Modules et Méthodes. Dans NAOqi, les "*Proxy*" sont les objets dédiés à la mise en relation avec les modules définis permettant l'accès aux méthodes de ce module. Ils s'emploient par l'appel du module désiré de la manière suivante :
Nom de l'objet Proxy = ALProxy("Nom du module", "IP du robot", 9559)

Pour l'utilisation du module *ALMotion*, le proxy créé donnant accès à l'ensemble des méthodes de ce module peut s'écrire :
motionProxy = ALProxy("ALMotion", robotIP, PORT)

L'objet *Proxy* nommé *motionProxy* donne ainsi accès aux différentes méthodes du module *ALMotion* tel que stiffnessInterpolation de la manière suivante :
motionProxy.stiffnessInterpolation(names, stiffnessLists, timeLists)

Table D.4 – Capteurs du robot NAO.

Segment	Angle par articulation	Amplitude	Type de moteur
Tête	Lacet	-119.5° à 119.5°	M3TA
	Tangage	-38.5° à 29.5°	M3TB
Bras	Roulis épaule	-18° à 76°	M3TB
	Tangage épaule	-119.5° à 119.5°	M3/4TA
	Lacet coude	-119.5° à 119.5°	M3TA
	Roulis coude	-88.5° à -2°	M3TB
	Lacet poignet	-104.5° à 104.5°	M2TA
Bassin	Lacet tangage hanche	-65.6° à 42.4°	M1TA
Jambe	Roulis hanche	-21.7° à 45.3°	M1TA
	Tangage hanche	-88.0° à 27.7°	M1TB
	Tangage genou	-5.3° à 121.0°	M1TB
	Tangage cheville	-68.2° à 52.9°	M1TB
	Roulis cheville	-22.8° à 44.1°	M1TA

En particulier, la méthode "*getData()*" du module *ALMemory* permet l'accès aux données capteurs inertiels (accélération, vitesse angulaire, position angulaire) et articulaire (position angulaire). Pour la partie gestion du mouvement via le module *ALMotion* dans notre cas d'application, deux principaux groupes de méthodes permettent le contrôle du mouvement : le contrôle de raideur et le contrôle d'articulation. L'API de contrôle de raideur permet la gestion de la rigidité des articulations du robot limitant le couple des moteurs de ces articulations. Une rigidité mise à 0 correspond à une articulation libre de tout mouvement tandis qu'une rigidité mise à 1 correspond à une articulation bloquée ou autorisée à utiliser toute la puissance de couple disponible pour effectuer le mouvement désiré.

L'API de contrôle des articulations est dédiée à la gestion du mouvement du robot d'un point de vue articulaire. Elle permet le contrôle des articulations de manière prédéfinie à travers des fonctions d'interpolation ou de manière réactive à chacun des cycles du système à travers des fonctions non-bloquante telles que setAngles et changeAngles.
La figure D.6 synthétise les principales méthodes du module *ALMotion* utilisées pour le contrôle du robot NAO lors de nos expérimentations.

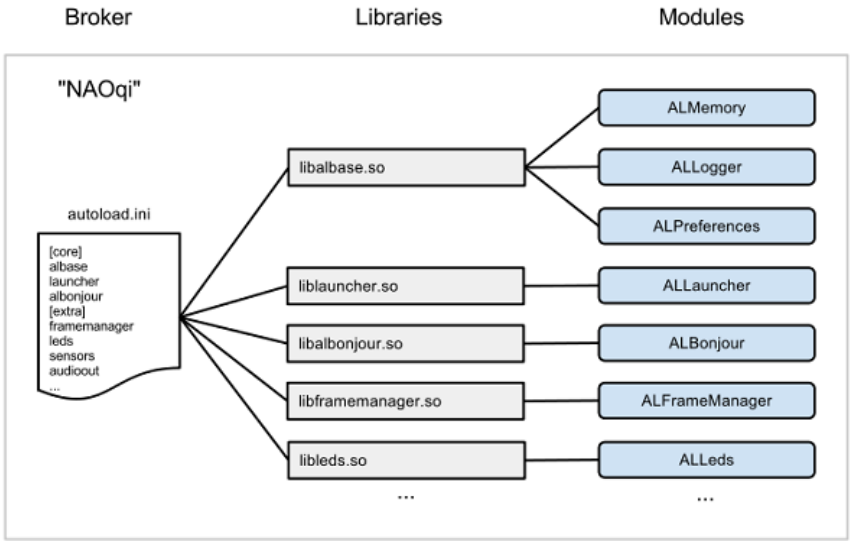


Figure D.4 – Architecture du Framework NAOqi.

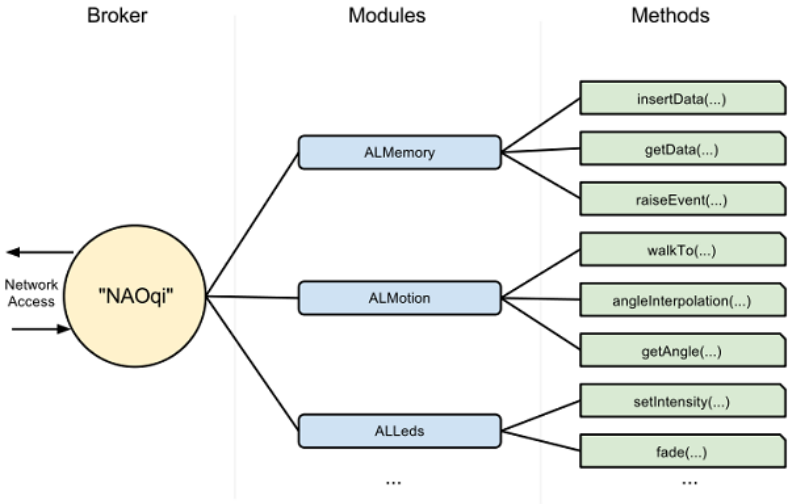


Figure D.5 – Relation *Broker*, Modules et Méthodes dans le framework NAOqi.

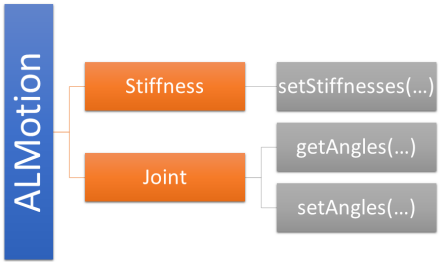


Figure D.6 – Principales méthodes utilisées dans notre implémentation.