



HAL
open science

Exploration de stratégies de commande visuelle prédictive pour le contrôle des mouvements de manipulateurs mobiles

Hugo Bildstein

► **To cite this version:**

Hugo Bildstein. Exploration de stratégies de commande visuelle prédictive pour le contrôle des mouvements de manipulateurs mobiles. Robotique [cs.RO]. UPS Toulouse, 2024. Français. NNT: . tel-04909915

HAL Id: tel-04909915

<https://laas.hal.science/tel-04909915v1>

Submitted on 24 Jan 2025

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.

Doctorat de l'Université de Toulouse

préparé à l'Université Toulouse III - Paul Sabatier

Exploration de stratégies de commande visuelle prédictive
pour le contrôle des mouvements d'un manipulateur mobile

Thèse présentée et soutenue, le 16 juillet 2024 par

Hugo BILDSTEIN

École doctorale

SYSTEMES

Spécialité

Robotique et Informatique

Unité de recherche

LAAS - Laboratoire d'Analyse et d'Architecture des Systèmes

Thèse dirigée par

Viviane CADENAT et Adrien DURAND-PETITEVILLE

Composition du jury

M. Patrick DANÈS, Président, Université Toulouse III - Paul Sabatier

M. François CHAUMETTE, Rapporteur, INRIA Rennes

M. Guillaume ALLIBERT, Rapporteur, Université Côte d'Azur

M. Olivier KERMORGANT, Examineur, Centrale Nantes

Mme Viviane CADENAT, Directrice de thèse, Université Toulouse III - Paul Sabatier

M. Adrien DURAND-PETITEVILLE, Co-directeur de thèse, Universidade Federal de Pernambuco

Table des matières

1	Introduction	7
1.1	Stratégies de commande	8
1.1.1	Stratégies de commande s'appuyant sur le paradigme Perception-Décision-Action (Sense-Plan-Act ou SPA)	8
1.1.2	Stratégies de commande réactive	8
1.1.3	Inscription du travail de thèse dans ce contexte	9
1.2	Asservissement visuel	9
1.2.1	De la commande réactive à l'asservissement visuel	9
1.2.2	État de l'art des stratégies de commande par asservissement visuel	10
1.2.3	Conclusion	14
1.2.4	Inscription du travail de thèse dans ce contexte	15
1.3	Asservissement visuel prédictif	15
1.3.1	De l'asservissement visuel à l'asservissement visuel prédictif	15
1.3.2	Stratégies d'asservissement visuel prédictif (AVP, ou VPC) pour le contrôle de manipulateurs mobiles	17
1.3.3	Inscription du travail de thèse dans ce contexte	22
2	Modélisation de la plateforme robotique	25
2.1	Présentation de la plateforme	25
2.2	Modélisation du robot	26
2.3	Modélisation cinématique de la base mobile	27
2.4	Modélisation cinématique du bras	29
2.5	Modélisation cinématique de la tête	30
2.6	Modélisation cinématique du manipulateur mobile	31
2.6.1	Terme relatif au mouvement du bras	31
2.6.2	Terme relatif au mouvement de la base mobile	31
2.6.3	Jacobienne du manipulateur mobile	32
2.7	Modélisation des caméras	32
3	Asservissement visuel prédictif mono-caméra pour manipulateur mobile	35
3.1	Présentation de la cible	36
3.2	Choix des indices visuels	37
3.2.1	Matrice d'interaction	38

3.2.2	Projection perspective	38
3.3	Synthèse du contrôleur	43
3.3.1	Méthodes de résolution en contrôle optimal	44
3.3.2	Présentation du problème	44
3.3.3	Analyse du schéma VPC	46
3.3.4	Définition du coût F_{vs}	47
3.3.5	Définition des modèles de prédiction	48
3.3.6	Définition des contraintes physiques	50
3.3.7	Contraintes de stabilité et de faisabilité	51
3.3.8	Résolution du problème d'optimisation	58
3.4	Résultats	60
3.4.1	Introduction à la tâche étudiée et implémentation du contrôleur	60
3.4.2	Comparaison des performances relatives au choix des indices visuels : S_p^p vs S_p^m	63
3.4.3	Résultats du schéma complet en simulation	67
3.4.4	Intérêt du calcul de gradient symbolique	68
3.4.5	Résultats expérimentaux	69
3.5	Conclusion	74
4	Asservissement visuel prédictif multi-caméra pour manipulateur mobile	77
4.1	Introduction	77
4.2	Stratégie multi-caméra et choix des indices visuels	78
4.2.1	Projection sphérique et moments 3D	79
4.3	Choix de la tâche secondaire - Mesure de manipulabilité	85
4.4	Synthèse du contrôleur	89
4.4.1	Présentation du problème	89
4.4.2	Définition du coût de la tâche visuelle F_{vs}	90
4.4.3	Définition du coût de la tâche de maximisation de manipula- bilité F_w	91
4.4.4	Définition des modèles de prédiction	92
4.4.5	Ensemble de contraintes de positionnement variant dans le temps	93
4.4.6	Autres contraintes	99
4.5	Schéma d'asservissement visuel prédictif 2D	99
4.5.1	Stratégie de sélection de la méthode de prédiction	100
4.5.2	Récapitulatif du processus	101
4.6	Résultats obtenus avec le schéma d'asservissement visuel prédictif 2D	103
4.6.1	Résultats de simulation	103
4.6.2	Résultats expérimentaux	114
4.7	Schéma d'asservissement visuel prédictif 2D 1/2	117
4.8	Résultats obtenus avec le schéma d'asservissement visuel prédictif 2D 1/2	120
4.8.1	Résultats de simulation	120

4.8.2	Résultats expérimentaux	124
4.9	Conclusion	129
5	Conclusion générale	133

Chapitre 1

Introduction

L'essor de la robotique et les défis auxquels elle est confrontée depuis ces dernières années appelle au développement de systèmes robotisés mobiles, d'architecture complexe, capables de réaliser des tâches avancées de manière de plus en plus autonome. Les robots évoluent dans des environnements très variés, souvent peu connus et hautement dynamiques. Ainsi, en robotique agricole par exemple, l'une des applications considérées au sein de mon équipe de recherche, la demande pour des robots capables d'intervenir dans des champs, des serres, des vergers, des vignes est forte. Les tâches considérées sont très variées : taille, désherbage, pollinisation, récolte de fruits, etc. Des capacités de navigation et de manipulation sont requises pour mener à bien ces tâches dont le succès dépend de l'aptitude du système à intervenir dans un environnement agricole très complexe : l'aspect évolutif en fonction des saisons ou de la météorologie, les irrégularités des terrains et des plantations, la perturbation du signal GPS nécessaire pour la localisation sont autant de contraintes que le système de commande du robot doit intégrer. De même dans l'industrie, l'automatisation des tâches d'intervention, d'inspection ou d'entretien nécessite de donner au robot des capacités de navigation et de manipulation dans un environnement intérieur ou extérieur, fortement contraint et hautement dynamique (présence de l'homme, de différents véhicules, etc.). Bien d'autres applications pourraient être citées : industrie du futur, robotique de service (aide aux personnes âgées ou handicapées, etc.), transport autonome, robotique d'intervention dans des sites dangereux ou inaccessibles, etc. Ainsi, les missions des robots évoluent et se complexifient : elles se réduisent maintenant rarement à une tâche de manipulation ou de navigation pure, mais intègrent ces deux composantes. Comme l'ont montré les exemples précédents, le robot est appelé à intervenir dans des environnements complexes au sens où : (i) ils sont difficilement modélisables, peu instrumentés et donc peu connus avec les conséquences déjà évoquées sur la variabilité ; (ii) ils sont fortement dynamiques, car partagés avec l'homme et d'autres véhicules mobiles (camions, tracteurs, autres robots, etc.).

1.1 Stratégies de commande

En robotique mobile comme en robotique de manipulation, on distingue généralement deux types de stratégies de commande : les stratégies s'appuyant sur le paradigme Perception-Décision-Action, et les stratégies dites réactives [Siciliano et al., 2008]. Bien que les architectures de contrôle actuelles se révèlent souvent hybrides, les deux types de stratégie apparaissent bien clairement distinctes.

1.1.1 Stratégies de commande s'appuyant sur le paradigme Perception-Décision-Action (Sense-Plan-Act ou SPA)

Traditionnellement, ce sont les méthodes SPA qui sont le plus souvent rencontrées. La réalisation de tâches robotisées est alors effectuée en trois étapes : la localisation, la planification de trajectoire et l'exécution [Siciliano et al., 2008] [Choset et al., 2005]. L'étape de planification est généralement réalisée hors ligne, sur la base de données prélevées sur l'environnement. Elle nécessite donc une connaissance a priori de ce dernier via l'utilisation d'un modèle interne qui doit être suffisamment précis. Ces modèles peuvent être de différentes natures : métriques, topologiques, etc. La planification de trajectoire peut être réalisée de manière relativement simple selon le niveau de modélisation. Sur un modèle topologique par exemple, un algorithme de Dijkstra permet de trouver la séquence optimale. Tout cela se complexifie lorsqu'un modèle métrique et de nombreuses contraintes sont considérées.

La commande, en général de type retour d'état, assure ensuite le suivi de la trajectoire planifiée et requiert une localisation du robot dans le modèle. Ce type d'approche s'avère limité face aux problématiques actuelles relatives à une automatisation plus avancée. Les tâches des robots se complexifient, les environnements sont très souvent dynamiques et inconnus. Les robots doivent être capables de s'adapter rapidement. Des événements imprévus peuvent se produire, venant remettre en cause des trajectoires pré-planifiées. Bien qu'il ait été proposé des améliorations, par exemple [Koenig and Likhachev, 2005], il est clair que ce type d'approche n'offre pas toute la flexibilité requise pour répondre aux défis actuels de la robotique. De plus, la phase de localisation nécessaire au calcul du retour d'état est très sensible aux erreurs de mesure et aux imprécisions de modèle, ce qui peut réduire la qualité de la réalisation de la tâche [Chaumette and Hutchinson, 2006].

1.1.2 Stratégies de commande réactive

Dans ce contexte, les solutions locales et réactives, basées sur des retours de sortie, semblent plus adaptées. Elles consistent à exprimer la tâche à partir des informations sensorielles pertinentes pour sa réalisation [Chaumette and Hutchinson, 2006]. Le robot est alors commandé par un retour de sortie et se déplace en fonction de l'objectif et des données sensorielles perçues pendant la mission. On parle alors de commande référencée capteur. Ces stratégies s'opposent donc aux précédentes et n'utilisent aucun modèle de l'environnement connu a priori. Il est alors possible

de détecter les événements imprévus et de modifier les mouvements du robot en conséquence. Ces solutions étaient initialement difficiles à implémenter compte tenu de la puissance de calcul nécessaire pour disposer des données sensorielles en un temps compatible avec la fréquence de la boucle de contrôle, mais les avancées en termes de processeurs, de cartes graphiques et de machine learning remettent en cause ces conclusions des dernières décennies.

La commande référencée capteurs permet donc de s'affranchir de l'étape de localisation dans le modèle interne de l'environnement. Elle consiste à exprimer les tâches à réaliser directement dans l'espace du capteur sous la forme d'une relation locale entre le robot et son environnement [Chaumette, 1990]. Elle a donc pour objectif de définir des lois de commande en boucle fermée sur les informations extéroceptives et ne nécessite plus la reconstruction de l'état du robot. Néanmoins, cette approche requiert l'utilisation de capteurs performants. Encore une fois, l'évolution de la technologie les rend désormais disponibles et économiques.

1.1.3 Inscription du travail de thèse dans ce contexte

Ces travaux de thèse s'inscrivent dans une démarche d'adaptabilité à des tâches variées, et cherchent à éviter l'instrumentation de l'environnement de travail et l'utilisation d'informations connues a priori sur celui-ci. Ils s'intègrent donc naturellement dans le champ des stratégies de commande réactives pour des manipulateurs mobiles.

1.2 Asservissement visuel

1.2.1 De la commande réactive à l'asservissement visuel

Les stratégies de commande réactives définissent des lois de commande en boucle fermée basées sur les informations fournies par des capteurs extéroceptifs. Ces capteurs sont de nature diverse et sont employés à des fins très variées. Les lidars permettent d'obtenir des informations sur la géométrie de l'environnement. Les caméras 2-D/3-D offrent une vue de cet environnement, et peuvent fournir des informations riches : forme, couleur, texture, présence, localisation et identification d'objets d'intérêt. Les micros sont, eux aussi, utilisés, ouvrant ainsi la voie à l'audition en robotique [Magassouba, 2016].

Parmi les capteurs extéroceptifs existants, la vision apparaît comme un capteur privilégié pour réaliser des tâches robotiques. L'intégration de la vision au sein des stratégies de commande réactive forme le domaine de l'asservissement visuel. Ce dernier a pour objectif de définir des lois de commande en boucle fermée sur les informations provenant des caméras. La recherche dans ce domaine a été très active depuis la fin du 20ème siècle et plusieurs types d'asservissement visuel existent à présent [Chaumette and Hutchinson, 2006] [Chaumette and Hutchinson, 2007]. Leurs caractéristiques principales sont détaillées et confrontées aux enjeux de la problématique étudiée dans la section suivante.

1.2.2 État de l'art des stratégies de commande par asservissement visuel

Asservissement visuel 3D, ou asservissement visuel basé position (ou 3D Visual Servoing, ou 3D-VS, Positioned-Based Visual Servoing, ou PBVS)

On distingue généralement deux types principaux de structure de commande référencée vision [Chaumette and Hutchinson, 2006]. La première repose sur un asservissement de situation de la caméra par rapport à l'environnement. La tâche est alors décrite dans un espace de primitives cartésiennes 3D : en l'occurrence, généralement, la situation de la caméra qu'elle doit atteindre dans l'environnement. Il est alors nécessaire d'effectuer une reconstruction 3D de la situation courante du système robotique. Cette stratégie, basée sur la position et l'orientation de l'organe piloté dans son espace cartésien, est généralement appelée asservissement visuel 3D (Position-Based Visual Servoing ou PBVS). Elle rejoint les techniques de planification de trajectoire en reconstruisant les primitives traditionnellement utilisées, le tout étant obtenu avec des informations issues des caméras. Cette étape de reconstruction de primitives présente des problèmes similaires à ceux induits par la localisation du robot comme l'inhérente présence d'erreurs d'estimation [Chaumette, 2007]. De plus, [Chaumette and Hutchinson, 2006] montre qu'une estimation correcte de la situation de la caméra est cruciale au sein d'un PBVS, à la fois pour le régime transitoire de l'asservissement que pour sa précision en régime permanent. En effet, le schéma n'est pas très robuste à des imprécisions de reconstruction 3D et des petites erreurs peuvent grandement affecter la précision après convergence. Bien que des avancées aient été faites, cet aspect reste un problème important du schéma classique de PBVS.

Asservissement visuel 2D, ou asservissement visuel basé image (ou 2D Visual Servoing, ou 2D-VS, ou Image-Based Visual Servoing, ou IBVS)

La deuxième structure principale d'asservissement visuel est basée sur la régulation d'une erreur exprimée dans l'image. On parle alors ici d'asservissement visuel 2D (Image-Based Visual Servoing ou IBVS). Dans l'asservissement visuel 2D, la tâche n'est donc plus définie dans l'espace cartésien, mais plutôt dans l'espace image [Chaumette and Hutchinson, 2006]. La commande est alors uniquement synthétisée grâce à des informations visuelles. L'étape de localisation de la caméra n'est donc plus nécessaire. L'IBVS présente une remarquable robustesse par rapport aux erreurs de calibration et aux erreurs dans l'image, et permet d'obtenir une excellente précision après convergence grâce à l'utilisation de primitives au plus proche des informations capteur. Concernant la stabilité et le régime transitoire, les deux stratégies se rejoignent, car des informations 3D sont tout de même nécessaires pour le calcul de la loi de commande. Ce problème est exacerbé lorsqu'on s'intéresse à des tâches nécessitant des déplacements sur de longues distances [Chaumette and Hutchinson, 2006].

Il a été montré que le type de caractéristiques visuelles utilisé a un impact si-

gnificatif sur les trajectoires dans les espaces cartésien et image [Chaumette and Hutchinson, 2006]. Pour cette raison, il est proposé dans de nombreux travaux de s'appuyer sur les moments d'image comme caractéristiques visuelles pour définir l'erreur [Tahri, 2004]. Ces caractéristiques visuelles offrent une meilleure traduction des informations cartésiennes que les coordonnées des points d'intérêt dans l'image habituellement utilisées. Les moments permettent par exemple de mieux contrôler les rotations autour des axes non-optiques. Il est alors possible de se rapprocher des performances du PBVS en optimisant les trajectoires dans l'espace cartésien et non plus dans l'image, tout en conservant l'absence d'information 3D au sein de l'erreur. Cependant, les trajectoires obtenues n'assurent plus nécessairement la visibilité des points d'intérêt. Il devient nécessaire de reformuler le problème pour prendre en compte ces contraintes, comme dans [Kermorgant and Chaumette, 2013]. D'autres contraintes sont généralement à prendre en compte : les contraintes de butées articulaires ou d'évitement d'obstacle comme dans [Cherubini and Chaumette, 2011] ou [Durand-Petiteville, 2012]. La reformulation des schémas d'asservissement visuel est une opération souvent complexe lorsque plusieurs contraintes sont à prendre en compte.

Asservissement visuel hybride ou 2D 1/2 (ou Hybrid Visual Servoing, ou HVS, ou 2D 1/2 Visual Servoing, ou 2D 1/2-VS)

Les stratégies d'asservissement visuel 2D et 3D peuvent également être utilisées de manière complémentaire pour former le domaine de l'asservissement visuel 2D 1/2 [Chaumette and Hutchinson, 2007]. Ce nom lui est donné en référence à la définition du vecteur d'erreur sur lequel il s'appuie. En effet, une utilisation astucieuse des données cartésiennes 3D et des données issues de l'image permet de bénéficier de combiner leurs avantages et d'éviter leurs inconvénients respectifs [Chaumette and Hutchinson, 2007]. Alors qu'une mauvaise estimation de la pose peut affecter la stabilité du PBVS, l'IBVS peut être sujet à des minima locaux, des singularités, une trajectoire de robot inadéquate et un domaine de convergence limité. L'ensemble de ces problèmes est de plus accentué lorsque des grands déplacements sont nécessaires pour atteindre la position désirée. L'utilisation couplée d'erreurs cartésiennes et dans l'image permet de répondre habilement à ces enjeux [Malis et al., 1999]. Il devient alors possible de partitionner la loi de commande pour découpler les différents degrés de liberté de la tâche [Corke and Hutchinson, 2001]. Les trajectoires obtenues classiquement par PBVS peuvent être améliorées en favorisant un centrage des points d'intérêts par la prise en compte d'indices visuels 2D [Corke et al., 2000]. Les trajectoires peuvent ainsi être optimisées simultanément d'un point de vue cartésien et dans l'image. Ces techniques permettent de gérer une rotation importante autour de l'axe optique par exemple. Cependant, le problème du contrôle des rotations autour des axes perpendiculaires à l'axe optique persiste et [Cong and Hanh, 2022] combinent plus récemment plusieurs stratégies d'asservissement visuel hybrides pour mieux gérer ce problème.

Les informations 3D utilisées peuvent alors être obtenues de plusieurs manières. La stratégie peut par exemple s'appuyer sur un modèle 3D de la cible. Les travaux

fondateurs de l'asservissement visuel 2D 1/2 [Malis et al., 1999] reposent sur une estimation partielle de la situation de la caméra basée sur la géométrie épipolaire ou sur la matrice d'homographie. Enfin, une caméra 3D peut également fournir les informations de profondeur nécessaires. Dans tous les cas, il est important de minimiser les étapes d'estimation et de reconstruction pour augmenter la robustesse et les performances de positionnement final des lois de commande.

Asservissement visuel dense ou asservissement visuel direct (ou Dense Visual Servoing, ou Direct Visual Servoing, ou DVS)

L'asservissement visuel direct constitue une autre catégorie dont le concept mérite d'être mentionné. L'objectif est de minimiser à la fois les étapes de reconnaissance de forme et l'intervention d'informations 3D connues a priori ou issues de capteur. Pour cela, les caractéristiques visuelles utilisées sont directement les intensités lumineuses des pixels. Il est premièrement important de noter que le terme direct peut avoir un double sens dans la littérature. En effet, il peut être lié à la partie contrôle grâce à l'utilisation des informations pixelaires directement au cœur de la loi de commande [Bakthavatchalam et al., 2018], mais peut également ne faire référence qu'à la partie estimation/perception en permettant d'éviter toutes sortes d'extraction de primitives géométrique [Silveira and Malis, 2007]. Ces approches sont dans ce sens au plus *direct* des données capteurs/caméra. Les informations visuelles utilisées s'opposent alors aux simples propriétés géométriques des indices utilisés dans les techniques traditionnelles. Le contraste de l'objet d'intérêt ou de l'image entière est donc également pris en compte. Le terme dense est souvent associé à ces stratégies, et indique généralement que les méthodes d'estimation des informations 3D ne nécessitent pas d'étape d'extraction, de suivi et d'appariement d'un ensemble de points ou de régions bien segmentées.

Ces stratégies de commande sont alors naturellement plus flexibles et robustes. Certaines techniques s'appuient sur les épipoles [Basri et al., 1999], la matrice essentielle, la matrice fondamentale ou l'homographie [Silveira and Malis, 2012] [Silveira and Malis, 2007] [Benhimane and Malis, 2006] et reconstruisent généralement les informations 3D uniquement à partir d'information 2D de l'image. Il ne s'agit donc pas là d'une nouvelle stratégie d'asservissement visuel puisque les informations 3D sont généralement utilisées dans la construction des matrices d'interaction pour l'IBVS [Chaumette and Hutchinson, 2007] ou directement introduits dans un schéma classique de PBVS, donnant ainsi naissance au DVS. La difficulté apparaît alors comme pour le PBVS lors de la phase de reconstruction 3D, ici correspondant à l'estimation des matrices d'homographie ou fondamentales, etc. Ces méthodes nécessitent également un appariement parfait de l'objet considéré dans les images, ce qui limitait fortement leur pertinence pratique. Les nouvelles performances qu'offrent les stratégies d'apprentissage supervisées et renforcées peuvent tout de même remettre en question ce désavantage. Aussi, les méthodes basées sur les matrices fondamentales et essentielles présentent des singularités au niveau de la convergence. Certains travaux proposent des solutions, mais l'utilisation de l'homographie est généralement privilégiée.

L’asservissement visuel direct est donc une manière intéressante pour minimiser l’intervention d’informations 3D connues a priori ou issues de capteur. Cependant, les approches mentionnées précédemment nécessitent des étapes de traitement d’image et d’estimation de matrice d’homographie complexes. Plus récemment, une approche permettant d’éviter toute opération d’appariement ou de reconstruction a été proposée [Collewet and Marchand, 2011]. Aucune autre information que l’intensité des pixels de l’image n’a besoin d’être prise en compte pour contrôler le mouvement du robot. Les indices visuels sont directement l’intensité des pixels de l’image. Ce domaine est appelé asservissement visuel photométrique. Dans ce cas, l’objectif est de minimiser directement l’erreur entre l’image actuelle et l’image désirée en utilisant une formulation analytique de la matrice d’interaction qui relie la variation de l’intensité de l’image au mouvement de la caméra. Le principal inconvénient de ce type de stratégies est leur domaine de convergence par rapport aux techniques classiques. Plusieurs solutions ont été proposées afin d’améliorer la robustesse du DVS en utilisant d’autres caractéristiques visuelles : [Bakthavatchalam et al., 2018] propose d’utiliser des moments photométriques de manière analogue aux stratégies géométriques, [Crombez et al., 2018] utilise des mixtures de gaussiennes, [Ourak et al., 2016] des ondelettes, [Marchand, 2020] la transformation en cosinus discrète, etc. Un défaut inhérent à l’asservissement provient naturellement du fait qu’une partie de l’image désirée doit tout le temps être visible. Certains travaux comme [Caron et al., 2013] proposent d’utiliser des caméras omnidirectionnelles, aussi appelées caméras catadioptriques pour remédier à ce problème. Ces caméras sont basées sur une première étape de projection sphérique suivi d’une reprojection sur le plan image et permettent alors d’obtenir un champ de vue bien plus grand. Cette stratégie basée sur une projection sphérique a notamment influencé certains aspects de ces travaux de thèse.

Asservissement visuel End-to-End

Les systèmes d’asservissement visuel End-to-End apprennent directement les lois de contrôle grâce à une phase d’entraînement, qui peut être réalisée de manière hors-ligne ou en ligne. Il n’y a pas vraiment de contrôleur, mais les lois de commande sont implicitement intégrées dans un réseau neuronal. L’objectif de ces systèmes est de générer par apprentissage supervisé la correspondance entre les images d’entrée et les commandes de contrôle en sortie, le tout étant considéré comme une tâche unique et indivisible à apprendre de bout en bout. Par exemple, dans [Kocić et al., 2019], un réseau neuronal profond pour un véhicule autonome est proposé en l’alimentant d’images de la route et de commandes de direction effectuées par un conducteur pendant la conduite de la voiture. Un autre exemple est décrit dans [Muller et al., 2005], où un réseau de neurones convolutif (CNN) prend des images stéréo en entrée et produit des angles de direction possibles pour une tâche d’évitement d’obstacles. Si la grande majorité des travaux se concentre pour le moment sur des tâches de basses dimensions d’ensemble de sortie des réseaux, des méthodes permettant d’appliquer la même stratégie à des bras manipulateurs ont également été proposées. [James et al., 2017] permet d’exécuter une tâche de prise et dépose sur des cubes à l’aide d’un bras

robotique à six liaisons articulaires, mais les auteurs mentionnent des difficultés potentielles si des mouvements de préhension plus complexes sont à étudier. Ces stratégies basées sur des réseaux de neurones profonds présentent cependant des inconvénients, notamment des réseaux lourds qui requièrent une puissance de calcul élevée.

Il faut également noter qu'il n'est question ici que de méthode rigoureusement end-to-end. En effet, les algorithmes basés sur des réseaux de neurones sont déjà largement utilisés de manière modulaire en asservissement visuel. Ils peuvent être intégrés à tous les niveaux, notamment dans l'étape de perception, grâce à des performances presque inégalées en termes de reconnaissance de forme, de segmentation d'image, etc. Mais ces techniques d'apprentissage sont également utilisées pour l'estimation directe de matrices nécessaires au calcul de la commande [Chaumette and Hutchinson, 2007], ou indirectement pour l'estimation d'information 3D comme la profondeur. On les retrouve également dans le module de contrôle uniquement en utilisant des données déjà traitées, des indices visuels 2D ou 3D complexes qui facilitent la mise en correspondance avec les commandes en sortie. Dans [Wang et al., 2020], la pose de l'objet est estimée à l'aide d'un module de perception et donnée en entrée d'un réseau de neurone qui fait office de contrôleur. Cette méthode est utilisée à défaut d'une stratégie End-to-End principalement parce que les techniques *sim2real* existantes présentent encore des performances limitées pour les algorithmes d'apprentissage par renforcement liés à la vision, en raison du rendu visuel peu réaliste des moteurs physiques actuels.

D'autres stratégies End-to-End utilisent donc l'apprentissage par renforcement. Le réseau s'entraîne sans base de données labellisée à travers un processus d'essais et d'échecs, et d'un système de récompense. Un exemple en est donné dans [Sadeghi, 2019], où un réseau basé sur le Q-learning est utilisé pour contrôler les mouvements d'un robot mobile. Un autre exemple est décrit dans [Lampe and Riedmiller, 2013], où un bras manipulateur équipé d'une caméra intégrée est conçu pour résoudre des tâches de manipulation et de préhension. Le réseau s'entraîne par essais et erreurs et ne nécessite aucune information préalable. Ces stratégies ne nécessitent pas de base de données connue a priori mais peuvent être *model-based* lorsque des tâches complexes sont considérées [Kober et al., 2013]. Bien que des stratégies *model-free* existent, elles restent souvent limitées : par exemple [Lampe and Riedmiller, 2013] se limite à la production de commande en translation de l'organe terminal. Une liste plus exhaustive de travaux relatifs à l'asservissement visuel par apprentissage supervisé ou par renforcement est disponible dans [Machkour et al., 2022].

1.2.3 Conclusion

Cette classification non-exhaustive permet d'appréhender le domaine de l'asservissement visuel, de distinguer les différents niveaux dont il est formé, et de comprendre ses enjeux. Mais comme toute classification, elle a tendance à dessiner des frontières entre ses éléments. Or, il est important d'insister sur le fait que les concepts de toutes ces stratégies d'asservissement visuel s'entremêlent en de nombreux points.

Il est ainsi souvent impossible de positionner un travail dans une case bien précise. En effet, le DVS n'est souvent qu'une extension du PBVS sans information 3D issue de capteurs. Il peut également être un module d'estimation dans l'IBVS. Les stratégies basées sur des réseaux de neurones semblent de premier abord conceptuellement différentes des autres, mais nécessitent souvent l'utilisation d'une méthode classique pour créer la base de donnée d'entraînement. Les stratégies utilisant l'apprentissage par renforcement ont besoin de modèles afin d'être capables de réaliser des tâches complexes, et ses modèles proviennent généralement des recherches de ce même domaine, apportant l'expertise nécessaire à la réduction de la difficulté de la tâche d'apprentissage. Ainsi, toutes ces méthodes se complètent et ne doivent pas nécessairement être mises en compétition.

1.2.4 Inscription du travail de thèse dans ce contexte

La démarche suivie jusqu'à maintenant insiste sur l'absence d'étape de localisation dans un modèle interne de l'environnement, pour offrir une meilleure polyvalence et une adaptabilité vis-à-vis des tâches de plus en plus complexes. L'étape de reconstruction 3D du PBVS pur, proche de ce concept, s'oppose donc à cette démarche. L'objectif des travaux de thèse est d'utiliser des primitives au plus proche des informations capteurs pour les raisons déjà explicitées. Bien qu'il ne s'agisse pas de refuser complètement l'intervention de la 3D dans la commande, l'objectif est de la réduire et surtout de minimiser sa sensibilité. Les asservissements 2D ou 2D 1/2 apparaissent donc comme des stratégies adaptées à la problématique considérées.

1.3 Asservissement visuel prédictif

1.3.1 De l'asservissement visuel à l'asservissement visuel prédictif

Même si les structures de commande référencée vision semblent se compléter, elles présentent toutes l'inconvénient de ne pas proposer de gestion directe de contraintes. De plus, dans un contexte d'asservissement visuel, il devient rapidement nécessaire d'assurer la visibilité par exemple. Des solutions ont été proposées pour ajouter certaines contraintes [Kermorgant and Chaumette, 2013] mais celles-ci sont difficilement généralisables à des contraintes arbitraires et nécessitent généralement une étude théorique spécifique. Une autre manière de gérer les contraintes consiste à combiner la planification de trajectoire et son suivi par asservissement visuel [Kazemi et al., 2009]. Cette solution permet alors à la fois de garantir une trajectoire optimale de la caméra dans l'espace cartésien et les contraintes considérées. Cependant, ces stratégies s'éloignent des propriétés de localité et de réactivité recherchées dans ces travaux de thèse.

Ensuite, il est bien connu que l'IBVS peut présenter des problèmes de stabilité et de convergence [Chaumette, 2007], et qu'il est compliqué de s'en prévenir. En effet, le premier point problématique provient des singularités. Ces singularités peuvent être

liées à la jacobienne de l'image, i.e. la matrice d'interaction, pour certaines configurations spécifiques. L'exemple classique est la rotation de 180° autour de l'axe optique [Chaumette, 2007]. Bien que ces configurations puissent souvent être gérées par l'utilisation d'indices visuels plus adéquats, aucune garantie ne peut être donnée puisqu'un découplage parfait des degrés de liberté de tâche ne peut pas être réalisé. Des singularités peuvent aussi évidemment provenir de la jacobienne du robot, celles-ci sont donc cinématiques. Un autre problème qui peut survenir concerne la possibilité d'atteindre des minima locaux. En fait, ces potentiels minima locaux résultent de mouvements d'image qui ne peuvent pas être réalisés. En d'autres termes, si la caméra atteint une position dans laquelle l'erreur se trouve dans le noyau de la jacobienne de l'image, les lois de contrôle classiques restent piégées dans un minimum local. À nouveau, des stratégies existent pour gérer ces cas difficiles [Chaumette, 2007] mais un contrôleur générique pour des larges et complexes déplacements reste très difficile à mettre en œuvre. Ces méthodes sont généralement efficaces pour des manipulateurs et des petits mouvements, mais semblent compliquées à appliquer à des manipulateurs mobiles. Ces problèmes ne sont pas nécessairement présents dans un schéma PBVS, mais d'autres difficultés relatives à la stabilité et à la convergence émanent des algorithmes d'estimation et de reconstruction 3D [Chaumette, 2007].

Enfin, un dernier point peut être mentionné : il s'agit de l'incapacité intrinsèque des méthodes classiques d'asservissement visuel à gérer des problèmes non-convexes. En effet, aucune gestion des objectifs à long terme n'est réalisée et la plupart des solutions proposées pour répondre à ces enjeux s'appuie sur plusieurs contrôleurs et l'utilisation d'un commutateur [Leca, 2021] [Fleurmond, 2015]. Des solutions au niveau du contrôleur ont été proposées. [Cherubini and Chaumette, 2011] permet d'éviter des obstacles de manière réactive, mais nécessite généralement une refonte importante de la stratégie de commande et il est difficile d'imaginer une généralisation et une généricité de la solution.

Ces difficultés relatives à l'asservissement visuel, et plus particulièrement à l'IBVS, sont expliquées plus en détail dans [Allibert et al., 2010b]. C'est dans ce contexte qu'une intégration de l'asservissement visuel dans le formalisme de la commande prédictive est proposée dans [Allibert et al., 2010a]. Cette reformulation des stratégies d'asservissement visuel en un problème de contrôle optimal permet alors de prendre en compte toute sorte de contraintes, et d'éviter les effets singuliers et les problèmes liés à la stabilité de l'IBVS. Le schéma alors obtenu est souvent appelé asservissement visuel prédictif (AVP, *Visual Predictive Control*, ou VPC). Bien que cette solution permette d'adapter l'asservissement visuel à des tâches robotiques bien plus complexes, des points sont encore incertains, notamment sa stabilité. Cependant, des théorèmes permettent de l'assurer. La stabilité d'un tel schéma est étudiée dans ces travaux, et une solution adaptée au contexte des manipulateurs mobiles est proposée.

La commande prédictive est basée sur quatre idées principales [Allibert et al., 2010b] : une référence, un modèle du processus dynamique du système considéré, une fonction de coût, et une méthode d'optimisation. Premièrement, il est nécessaire de prédire l'évolution du système sur un horizon défini à partir de modèles

de prédiction. Les prédictions sont ensuite utilisées pour définir une fonction de coût grâce à des caractéristiques visuelles judicieusement choisies. Cette fonction de coût doit présenter un unique point d'équilibre correspondant à la réalisation de la tâche : en l'occurrence le bon positionnement de la cible dans l'image pour une tâche d'asservissement visuel. Des contraintes sont aussi formulées et un solveur est finalement utilisé pour résoudre le problème d'optimisation et trouver la séquence de commandes optimales.

1.3.2 Stratégies d'asservissement visuel prédictif (AVP, ou VPC) pour le contrôle de manipulateurs mobiles

Stratégies d'asservissement visuel prédictif

Premièrement, il est important de souligner qu'il n'existe à notre connaissance que très peu de travaux étudiant rigoureusement le VPC appliqué à des problématiques de manipulateurs mobiles. Le positionnement de ce travail se fait donc par rapport aux stratégies portant plus largement sur des contrôleurs VPC appliqués à des bras robotiques uniquement, des commandes prédictives s'appuyant sur du PBVS, des méthodes de commande de manipulateurs mobiles par asservissement visuel, ou même encore plus généralement à des commandes classiques sur des robots redondants. En effet, les contributions du travail de thèse se retrouvent au croisement de plusieurs disciplines de recherche, et à travers de subtiles différences qui sont parfois difficiles à distinguer.

Au cours de la dernière décennie, l'intérêt pour les contrôleurs VPC a augmenté et de nombreux schémas ont été conçus pour contrôler différents systèmes robotiques : une caméra montée sur un bras robotique [Copot et al., 2012] [Fusco et al., 2020] [Mohamed et al., 2021]. [Allibert et al., 2010a] présente une structure de base de contrôleur et construit une commande purement réactive, s'appuyant sur un modèle de prédiction local en utilisant les matrices d'interaction. Mais les déplacements de caméra considérés restent très faibles : 0.6m environ en translation, et une rotation majoritairement orientée autour de l'axe optique. [Mohamed et al., 2021] améliore ces résultats en proposant l'utilisation de modèles de prédiction de second degré ainsi plus précis, mais les déplacements sont toujours du même ordre de grandeur. Dans [Copot et al., 2012], l'utilisation des moments d'image est proposée, permettant d'obtenir d'excellentes propriétés de découplage par rapport aux six degrés de liberté des tâches de positionnement. Néanmoins, la stratégie présentée utilise une trajectoire de référence, nécessitant une étape de planification.

Dans [Heshmati-Alamdari et al., 2014], [Mohammad Hossein Fallah and Janabi-Sharifi, 2021] et [Mcfadyen et al., 2014], un contrôleur VPC est appliqué au cas d'une caméra volante et les questions de stabilité et de robustesse de ces stratégies sont étudiés seulement dans les deux premiers travaux mentionnés sur des déplacements encore faibles. Une commande prédictive (MPC) multi-capteur dérivée du VPC est utilisée pour garer un véhicule autonome dans [Pérez-Morales et al., 2021]. Il permet d'observer de nombreux avantages du VPC pour réaliser des tâches non convexes.

En effet, la tâche de parking nécessite généralement plusieurs manœuvres. Le déplacement étudié est également beaucoup plus important, mais la tâche n'est pas directement définie dans l'image et le problème peut être réduit à deux dimensions. Un VPC s'appuyant sur un schéma IBVS est appliqué à un bras robotique continu piloté par tendon dans [Norouzi-Ghazbi et al., 2021]. Dans [Wu et al., 2022], un contrôleur hybride est développé en associant un contrôleur à asservissement visuel par réseau neuronal avec un contrôleur VPC : le premier est chargé d'amener le robot mobile proche de la situation désirée dans l'image et le contrôleur VPC se charge du positionnement précis. Ce n'est que récemment que des travaux comme [Zhang et al., 2021] émergent. Un contrôleur VPC appliqué à un drone réalise une trajectoire de plus de 3m. Le modèle utilisé prend en compte la dynamique du système et la stabilité de l'ensemble est analysé sous l'influence de perturbations. De plus, le contrôleur ne réalise pas uniquement un suivi de trajectoire, mais calcule à chaque itération une nouvelle trajectoire. L'horizon de prédiction n'est cependant pas grand, ce qui rend compliqué la prise en compte d'objectifs à long terme.

Cependant, les stratégies basées sur la commande prédictive visant à contrôler un manipulateur mobile ne sont généralement pas définies dans l'espace image. Par exemple, dans [Pankert and Hutter, 2020], la fonction de coût représente l'erreur de pose de l'effecteur final et est ensuite minimisée à l'aide d'un MPC. Une autre approche consiste à utiliser les coordonnées généralisées du système pour exprimer la fonction de coût, comme dans [Gifftthaler et al., 2017], [Avanzini et al., 2018] et [Colombo et al., 2019]. Les travaux [Le Flecher, 2020] synthétisent une stratégie de commande référencée vision permettant de contrôler les mouvements du système multi-bras et proposent une gestion des contraintes relatives au partage de l'espace de travail. Les caméras sont parfois utilisées comme principal capteur pour contrôler les manipulateurs mobiles, mais la tâche n'est pas définie dans l'espace image. Par exemple, [Martínez et al., 2013] et [Logothetis et al., 2018] utilisent respectivement une caméra temps de vol et une caméra RGB-D pour récupérer la pose de l'objet, puis s'appuient tous deux sur une fonction de coût basée sur la pose pour accomplir la tâche. Dans [Logothetis et al., 2018], l'estimation de la pose 3D de l'objet est réalisée à l'aide de méthode de Monte Carlo. Un VPC s'appuyant sur un schéma PBVS est également utilisé pour contrôler un véhicule sous-marin autonome dans [Heshmati-Alamdari et al., 2020]. À notre connaissance, le travail présenté dans [Gao et al., 2021] est le seul à considérer une stratégie de VPC pour contrôler un manipulateur mobile. Il s'appuie sur un MPC hiérarchique utilisant des points d'intérêt comme indices visuels pour contrôler un véhicule manipulateur sous-marin.

Il est également important de faire la différence entre un MPC gérant l'intégralité de la partie de commande et un MPC qui ne réalise que le suivi de trajectoire. Dans [Colombo et al., 2019] et [Minniti et al., 2019] par exemple, une trajectoire de référence est donnée en entrée. C'est également le cas pour [Prakash and Behera, 2023], où un réseau de neurone est entraîné à partir d'exemples humains et planifie ensuite les mouvements. Un contrôleur de commande optimale est utilisé par la suite pour effectuer le suivi de trajectoire.

Stabilité

Les systèmes robotiques autonomes présentent un défi en ce qui concerne la stabilité en boucle fermée. Tout d'abord, notons que dans ces travaux de thèse, la notion de stabilité signifie plus précisément une stabilité locale asymptotique au sens de Lyapunov. C'est cette stabilité qui est majoritairement étudiée dans le domaine du MPC. Même si la stabilité globale est rarement prouvée, l'avantage des méthodes basées la commande prédictive est de proposer de larges domaines de convergence lorsque les fonctions de coût sont correctement définies. Ces domaines ne sont malheureusement pas souvent quantifiables. L'idée essentielle de l'étude de la stabilité des MPC réside sur deux points clefs : la faisabilité et la fonction de valeur [Michalska and Mayne, 1993]. La faisabilité traduit l'existence d'une solution au problème d'optimisation, et la fonction de valeur est définie comme le coût optimal du problème d'optimisation [Allgower et al., 2004] et sert souvent de candidat pour une fonction de Lyapunov. Les approches d'analyse de la stabilité peuvent être résumées ainsi : si le problème est faisable et que la fonction de valeur est strictement décroissante, le système converge théoriquement vers son unique point d'équilibre.

Les manipulateurs mobiles doivent généralement effectuer de grands déplacements dans des environnements éventuellement encombrés d'obstacles a priori inconnus et l'une des questions essentielles est de savoir si une stratégie MPC avec un horizon de prédiction fini garantit ou non la stabilité en boucle fermée. Il existe différentes approches pour garantir la stabilité en boucle fermée avec des horizons finis, toutes globalement basées sur l'approximation de la partie finale du coût au-delà de l'horizon de prédiction. Seulement deux d'entre elles, principalement utilisées, sont présentées ici. La première consiste à ajouter une contrainte terminale à la fin de l'horizon de prédiction [Michalska and Mayne, 1993] [Allgower et al., 2004]. Ceci garantit la stabilité de la boucle fermée si le problème a une solution au départ du contrôle. Tout comme dans le cas de l'horizon infini, la récursivité de la faisabilité est prouvée et l'hypothèse d'optimalité implique une diminution de la fonction de coût. L'inconvénient de ce type de contrainte est que le système doit d'atteindre l'origine en temps fini. Cela entraîne des problèmes de faisabilité pour de faibles longueurs d'horizon de prédiction. La deuxième méthode habituellement utilisée est une stratégie de MPC à coût terminal qui consiste à relaxer cette contrainte terminale à une région plus large autour de l'objectif et à ajouter un coût terminal défini dans cette région de manière à surestimer le coût de l'horizon infini. Il est alors montré [Allgower et al., 2004] que la stabilité est assurée si le coût ajouté est Lyapunov sur cette région terminale.

Les preuves théoriques de la stabilité des MPC s'appuient sur l'analyse de la fonction de valeur, et supposent une optimalité de la solution renvoyée par les solveurs. Cependant, les manipulateurs mobiles comportent de nombreux degrés de liberté, impliquant un espace de recherche considérablement grand et contraint. Ils doivent également être contrôlés à des fréquences élevées, la résolution du problème d'optimisation devrait généralement être réalisée en quelques dizaines de millisecondes. Les solveurs utilisés sont donc généralement des solveurs locaux. Même si des techniques existent pour faciliter ou accélérer la résolution afin de se rapprocher

d'une solution optimale en très peu de temps, l'optimalité de la solution ne peut quasiment jamais être assurée. La théorie de la stabilité de Lyapunov établit des conditions pour la stabilité asymptotique. Cependant, il est nécessaire d'examiner plus attentivement les conditions sous lesquelles le contrôle sous-optimal assure la stabilité. Dans [Scokaert et al., 1999], un résultat similaire au théorème de stabilité de Lyapunov est démontré en autorisant une sous-optimalité de la loi de commande. Ce résultat est prouvé pour les deux méthodes expliquées précédemment : le mode dual et la contrainte terminale.

De tels théorèmes sont utilisés par exemple dans [Heshmati-Alamdari et al., 2014] pour assurer la stabilité et étudier la robustesse d'un contrôleur VPC à une caméra volante. Une méthode contournant l'utilisation de la contrainte terminale basée sur des profils de pondération en croissance monotone est démontrée dans [Alamir, 2018], mais la stratégie se base sur les mêmes théorèmes fondamentaux de Lyapunov. Cette stratégie consiste à pondérer bien plus la partie finale que la première partie de l'horizon de prédiction, mais la preuve suppose des résolutions optimales et une faisabilité initiale du problème d'optimisation.

Coordination base mobile – bras manipulateur

Les manipulateurs mobiles sont des systèmes composés d'une base mobile et d'un bras robotique. Une question importante concerne la coordination des mouvements de ces deux sous-systèmes. Généralement, ils sont utilisés de manière indépendante : la base se dirige vers la zone de manipulation, s'arrête et la tâche de manipulation s'effectue avec la base immobile. Mais il peut être intéressant d'utiliser les deux sous-systèmes en même temps, de manière coordonnée. Premièrement, il convient de définir le terme de coordination. Cet aspect intervient d'abord et obligatoirement sur un système redondant. La redondance est un concept relatif à la tâche considérée. En robotique, celle-ci correspond généralement au positionnement de l'organe terminal. Un robot est alors considéré redondant lorsque son nombre de degrés de liberté est supérieur au nombre nécessaire pour effectuer la tâche [Siciliano et al., 2008] [Sciavicco and Siciliano, 2012]. Il y a alors dégénérescence de l'inversion du modèle géométrique et cinématique du robot puisqu'une infinité de configurations permet le positionnement de l'organe terminal à une pose précise. Le domaine de la coordination propose ainsi d'utiliser la redondance afin de réaliser des tâches supplémentaires. Ensuite, il est important de préciser que la notion de redondance n'a de sens qu'au travers d'une relation entre une ou plusieurs tâches principales et une ou plusieurs tâches secondaires.

En robotique, le terme de tâche principale désigne l'objectif central à accomplir, tandis que le terme de tâche secondaire est réservé aux objectifs nécessaires au bon déroulement de la tâche principale. La littérature scientifique étudie globalement deux tâches principales : la stabilisation en un point de l'espace et le suivi de trajectoire. La stabilisation en un point est plus généralement appelé régulation en automatique. Les manipulateurs mobiles sont donc généralement des systèmes redondants et des tâches secondaires peuvent être ajoutées. De nombreuses tâches secondaires sont étudiées : évitement d'obstacle, gestion des butées articulaires, ma-

nipulabilité du manipulateur ou du manipulateur mobile, minimisation de l'énergie utilisée, etc.

Comme expliqué précédemment, la redondance signifie que le noyau de la jacobienne de la tâche est non nul. Trois classes principales de méthode de résolution existent alors. La première est basée sur cette jacobienne et permet de choisir une solution parmi l'infinité existante à partir d'un critère. L'utilisation de la pseudo-inverse par exemple permet de minimiser la norme du vecteur de commande. Cette solution est donc adéquate lorsqu'une tâche secondaire de minimisation d'énergie est considérée. La deuxième classe de méthode est basée sur le noyau de la jacobienne justement [Bayle et al., 2002]. Un terme est ajouté à la solution qui n'affecte pas la réalisation de la trajectoire relative à la tâche principale. La tâche secondaire est donc projetée sur le noyau de la jacobienne et est alors réalisée au mieux, mais il n'existe aucune garantie à sa réalisation. Enfin, la dernière classe d'approche consiste à ajouter des tâches afin de réduire ou de supprimer la redondance. Il est à noter que certaines méthodes établissent une hiérarchie entre les tâches : la tâche principale est réalisée en priorité et la tâche secondaire est exécutée au mieux et sans aucune garantie. D'autres méthodes peuvent créer un compromis. C'est par exemple le cas en contrôle optimal lorsque les tâches sont tout simplement ajoutées à la fonction de coût pour créer une tâche composée. Une tâche considérée comme prioritaire est ajoutée sous la forme d'une contrainte. Une classification plus détaillée est disponible dans [Picard, 2022].

Stratégies multi-camera

Un des enjeux majeurs de l'IBVS concerne son domaine de convergence. Même si l'utilisation de moments augmente ce dernier, il reste une limite intrinsèque infranchissable. En effet, les caméras traditionnelles basées sur la projection perspective ont un champ de vision très limité par construction. Le domaine de convergence d'une méthode d'asservissement visuel utilisant une telle caméra reste donc très restreint. C'est pourquoi la majeure partie des travaux de la littérature étudie des faibles mouvements en ce qui concerne les rotations par exemple. Puisque dans le cas d'une rotation autour des axes non-optiques de 90° , la cible n'est souvent plus dans le champ de vision de la caméra.

Une première solution à ce problème est l'utilisation de caméras omnidirectionnelles, aussi appelées catadioptriques. Elles consistent en l'association d'une caméra classique avec un miroir courbé, augmentant ainsi considérablement le champ de vision. Cette association se décompose alors en deux étapes : une projection sphérique, correspondant au miroir et une projection perspective. Le modèle général correspondant à ce type de caméra est appelé modèle de projection centrale unifié [Khomutenko et al., 2015]. Des méthodes d'asservissement visuel s'appuyant sur ces caméras ont déjà largement été proposées : elles sont majoritairement appliquées à des robots mobiles [Marie et al., 2019], [Aliakbarpour et al., 2014] mais des travaux portant cette méthode sur des bras robotiques existent également : [Abdelkader et al., 2005], [Mezouar et al., 2004]. Ces caméras s'intègrent également facilement dans les stratégies plus récentes d'asservissement visuel. En effet, [Caron et al., 2013]

propose par exemple un asservissement visuel photométrique et de bons résultats sont obtenus sur un robot à coordonnées cartésiennes et sur un robot mobile. Enfin, [Allibert et al., 2008] utilise ces caméras omnidirectionnelles dans un schéma d’asservissement visuel prédictif sur un manipulateur plan à trois degrés de liberté.

Cependant, il est possible que cette augmentation du champ de vision ne soit pas suffisante dans un contexte général de manipulation mobile. L’objectif de ce travail de thèse est également de faire en sorte que les mouvements du bras ne soient pas nécessairement contraints par la visibilité. C’est ainsi qu’il semble pertinent d’introduire les notions de déportation des caméras. En effet, une deuxième manière bien connue de pallier ces problèmes de visibilité est d’utiliser une ou plusieurs caméras déportées, aussi mentionnées comme étant en configuration *Eye-to-Hand* (ou EtoH), qui s’oppose avec la traditionnelle et plus courante configuration *Eye-in-Hand* (ou EinH). S’il semble évident qu’une caméra en EinH offre des informations plus précises, mais partielle de l’environnement, une caméra en EtoH permet d’obtenir des informations complémentaires, i.e. globales, mais moins précise. Dans un contexte d’IBVS par exemple, il semble dommage de perdre l’avantage d’une caméra EtoH. Cette complémentarité a été étudiée et appliquée à bon nombre de systèmes robotiques, et forme le domaine de la coopération de caméras EinH et EtoH. [Flandin et al., 2000] utilise par exemple l’image globale pour contrôler les degrés de liberté de translation, et l’image locale pour contrôler les rotations. [Dune et al., 2007] utilise la complicité des deux configurations pour appliquer la théorie de la géométrie épipolaire et contrôler un bras robotique six axes. [Kermorgant and Chaumette, 2011] fait de même et généralise une méthode de commande à un système robotique multi-caméra en définissant une matrice d’interaction moyenne. Plusieurs stratégies émergent alors : certaines solutions proposent d’utiliser toutes les informations des caméras lorsque d’autres privilégient l’utilisation de la caméra la mieux adaptée à la tâche en question. Par exemple, [Chang and Shao, 2010] utilise la caméra en EtoH pour estimer la pose de la cible et seule la caméra en EinH est utilisée dans la partie contrôle. [Cuevas-Velasquez et al., 2018] propose un contrôle hybride d’un bras robotique utilisant plusieurs caméras déportées et une caméra en EinH est utilisée pour le contrôle. Le système commence alors initialement par utiliser les informations de la caméra EtoH et bascule en mode EinH lorsque la distance entre le robot et la cible est inférieure à un seuil défini. Enfin, plus récemment, une stratégie similaire est appliquée à un bras robotique continuum dans [AlBeladi et al., 2022].

1.3.3 InSCRIPTION du travail de thèse dans ce contexte

Compte tenu de l’application de la stratégie choisie à des manipulateurs mobiles, présentant donc intrinsèquement des déplacements larges, contraints et complexes, il semble intéressant d’intégrer l’asservissement visuel 2D et 2D 1/2 au sein d’un schéma de commande prédictive et de s’appuyer sur plusieurs caméras.

Ainsi, les travaux de thèse portent tout particulièrement sur l’exploration de stratégies de commande visuelle prédictive multi-caméra pour le contrôle de manipulateurs mobiles. Nos contributions consistent donc en la synthèse d’un schéma

VPC permettant de répondre aux enjeux soulevés. Ces derniers concernent notamment la définition d'une fonction de coût adaptée, la garantie de la stabilité en boucle fermée, l'aptitude à réaliser des tâches variées, la coordination entre la base et le bras et une implémentation adéquate. Après avoir modélisé le système robotique dans le chapitre 2 suivant, plusieurs schémas de commande sont proposés. Le chapitre 3 met en place une première stratégie de commande mono-caméra permettant d'approfondir la problématique étudiée et d'identifier ses enjeux. Ensuite, deux approches multi-caméra plus avancées sont présentées dans le chapitre 4 et répondent aux défis soulevés. Pour chaque schéma, des résultats de simulation et expérimentaux sont exposés et analysés, mettant en avant la pertinence et l'efficacité des stratégies de commande visuelle prédictive dans le contexte de manipulation mobile étudié.

Chapitre 2

Modélisation de la plateforme robotique

L’objectif de ce chapitre est de définir les éléments nécessaires à la construction du contrôleur VPC complet. La plateforme robotique utilisée est ainsi présentée, puis modélisée afin d’obtenir les modèles de prédiction essentiels au contrôle prédictif.

2.1 Présentation de la plateforme

Le système robotique considéré est le robot TIAGo de PAL Robotics (voir Figure 2.1a). Il comprend un haut du corps composé d’un torse à un degré de liberté non-utilisé ici, d’un bras à 7 degrés de liberté et d’une tête de 2 degrés de liberté montés sur une base mobile de type différentielle. Une première caméra est attachée au poignet du bras, seuls 5 degrés de liberté de ce dernier sont donc utilisés pour la contrôler. Pour la tête, seule la rotation autour de l’axe vertical est commandée dans ce travail. Une deuxième caméra est positionnée au niveau de la tête du robot, offrant un point de vue complémentaire, permettant d’effectuer des missions plus générales. L’objectif principal du contrôleur est de positionner cette caméra, ou plus précisément le repère attaché à cette caméra, à une situation donnée relativement à une cible.

Plusieurs capteurs sont déjà présents sur le robot. Il y a des capteurs proprioceptifs : des encodeurs ainsi qu’une IMU 6 axes. Les informations de ces capteurs ne sont utilisées que dans les contrôles bas-niveau et donc à aucun moment dans les schémas de contrôle présentés dans ces travaux. Ils permettent cependant le calcul d’odométrie utilisé pour afficher les résultats dans un repère global afin de faciliter la compréhension. Des capteurs extéroceptifs sont également montés : un LIDAR de 180° à l’avant et trois SONARs à l’arrière. Une caméra Orbbec Astra RGB-D est présente dans la tête. En plus de ces capteurs déjà intégrés, une caméra *realsense D435i* a donc été ajoutée au poignet du bras. Finalement, les seuls capteurs utiles pour notre contrôleur VPC sont les deux caméras.

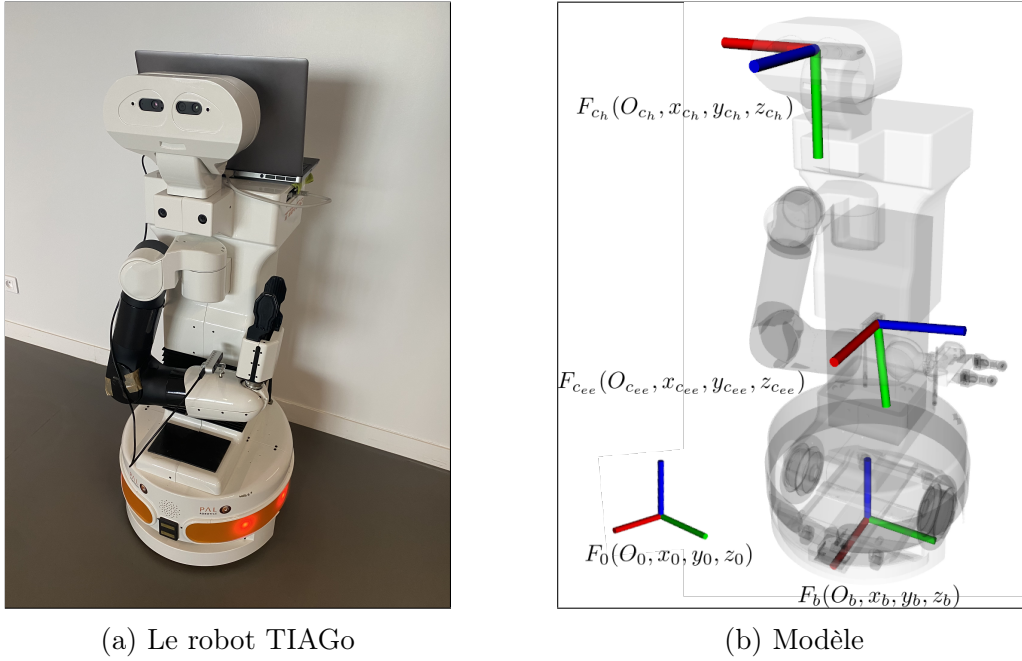


FIGURE 2.1 – Plateforme robotique

2.2 Modélisation du robot

Afin de modéliser le robot, les repères suivants sont introduits : $F_0(O_0, x_0, y_0, z_0)$, $F_b(O_b, x_b, y_b, z_b)$, $F_{c_h}(O_{c_h}, x_{c_h}, y_{c_h}, z_{c_h})$ et $F_{c_{ee}}(O_{c_{ee}}, x_{c_{ee}}, y_{c_{ee}}, z_{c_{ee}})$ correspondant respectivement au repère global, repère de la base mobile, repère de la caméra de la tête et repère de la caméra du poignet (voir Figure 2.1b). Le repère global n'est introduit que pour la modélisation et pour analyser les résultats globaux, mais il n'est jamais utilisé dans le schéma de contrôle. Les prédictions sont réalisées dans le repère de la base à un instant donné.

La pose de la base mobile χ_b et son vecteur de contrôle u_b sont définis comme suit :

$$\chi_b = [X, Y, \theta]^T, u_b = [v, \omega]^T \quad (2.2.0.1)$$

où X , Y et θ sont respectivement les coordonnées de la base et son orientation autour de l'axe vertical dans le repère initial de l'horizon de prédiction. v et ω sont les vitesses linéaire et angulaire le long de x_b et autour de z_b . La configuration du bras et son vecteur de commande sont exprimés comme suit :

$$\chi_a = [q_1, q_2, q_3, q_4, q_5]^T, u_a = [\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4, \dot{q}_5]^T \quad (2.2.0.2)$$

où q_i représente le i^{eme} angle articulaire et \dot{q}_i la i^{eme} vitesse articulaire.

Le même raisonnement est suivi pour la configuration de la tête χ_h et son vecteur de commande u_h :

$$\chi_h = [h_1]^T, u_h = [\dot{h}_1]^T \quad (2.2.0.3)$$

Finalement, la configuration totale χ_{tot} du système robotique et son vecteur de commande u_{tot} sont définis comme ceci :

$$\chi_{tot} = [\chi_b^T, \chi_a^T, \chi_h^T]^T, u_{tot} = [u_b^T, u_a^T, u_h^T]^T \quad (2.2.0.4)$$

Il convient également de définir la configuration χ_{mm} et le vecteur de commande u_{mm} du manipulateur mobile, c'est-à-dire du système sans la partie correspondant à la tête :

$$\chi_{mm} = [\chi_b^T, \chi_a^T]^T, u_{mm} = [u_b^T, u_a^T]^T \quad (2.2.0.5)$$

Les parties suivantes présentent la modélisation cinématique du manipulateur mobile. En effet, l'élément fondamental du contrôle prédictif est évidemment son modèle de prédiction, également appelé dynamique du problème de contrôle optimal. Dans ce travail, compte tenu de la problématique étudiée, les vitesses de déplacement considérées ne sont pas extrêmement grandes et les effets dynamiques sont négligés. Les prédictions sont alors réalisées à partir d'une intégration du niveau cinématique. Deux méthodes sont habituellement utilisées : globale et locale [Allibert et al., 2010a]. Dans le deuxième cas, la jacobienne du robot est nécessaire. Même si le modèle de prédiction considéré par la suite est global, la jacobienne reste centrale dans la commande. Elle est par exemple essentielle au calcul de manipulabilité, ou à la méthode de hiérarchisation de tâche.

2.3 Modélisation cinématique de la base mobile

La base mobile du robot *TIAGo* est de type différentielle sur la plateforme robotique considérée. La relation entre la variation de son vecteur d'état et son vecteur de commande peut alors être exprimée de la sorte :

$$\dot{\chi}_b = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cos(\theta) \\ v \sin(\theta) \\ \omega \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.3.0.1)$$

De plus, le schéma MPC utilisé dans ces travaux utilise un horizon de prédiction discret avec une période de prédiction notée T_s en considérant la commande constante entre deux prédictions. Une relation de récurrence est alors recherchée entre les temps t_k et $t_{k+1} = t_k + T_s$. Les notations sont allégées en ne mentionnant que la valeur de k pour signifier l'instant t_k . Dès lors, plusieurs méthodes d'intégration existent. La méthode la plus utilisée est la classique méthode d'Euler explicite, mais une approximation est alors réalisée. Le modèle de prédiction étant au cœur des lois de commande, une méthode exacte est privilégiée. En effet, l'intégration peut être calculée analytiquement comme expliqué dans [Petiteville and Cadenat, 2014]. Ces deux méthodes sont détaillées ci-dessous :

— Méthode Euler explicite

$$\chi_b(k+1) = \chi_b(k) + T_s \dot{\chi}_b(k) \quad (2.3.0.2)$$

— Méthode analytique

$$\chi_b(k+1) = \chi_b(k) + \int_{t_k}^{t_{k+1}} \dot{\chi}_b(t) dt \quad (2.3.0.3)$$

En séparant les variables d'état, on obtient alors :

$$\begin{cases} X(k+1) = X(k) + \int_{t_k}^{t_{k+1}} v(k) \cos(\theta(k) + \omega(k)t) dt \\ Y(k+1) = Y(k) + \int_{t_k}^{t_{k+1}} v(k) \sin(\theta(k) + \omega(k)t) dt \\ \theta(k+1) = \theta(k) + \int_{t_k}^{t_{k+1}} \omega(k) dt \end{cases} \quad (2.3.0.4)$$

Il faut alors distinguer deux cas pour continuer l'intégration : le cas $\omega(k) = 0$ et le cas $\omega(k) \neq 0$.

$$\omega(k) = 0 \implies \begin{cases} X(k+1) = X(k) + v(k) \cos(\theta(k)) T_s \\ Y(k+1) = Y(k) + v(k) \sin(\theta(k)) T_s \\ \theta(k+1) = \theta(k) \end{cases} \quad (2.3.0.5)$$

$$\omega(k) \neq 0 \implies \begin{cases} X(k+1) = X(k) + \frac{v(k)}{\omega(k)} \left[\sin(\theta(k) + \omega(k) T_s) - \sin(\theta(k)) \right] \\ Y(k+1) = Y(k) - \frac{v(k)}{\omega(k)} \left[\cos(\theta(k) + \omega(k) T_s) - \cos(\theta(k)) \right] \\ \theta(k+1) = \theta(k) + \omega(k) T_s \end{cases} \quad (2.3.0.6)$$

La méthode permet donc de calculer l'intégration de manière exacte. Cependant, elle crée une singularité du modèle de prédiction en $\omega(k) = 0$. Cette singularité peut devenir gênante au niveau des calculs analytiques ou symboliques des gradients de ce modèle. En effet, elle produit une discontinuité de la dérivée de X et Y par rapport à ω en $\omega = 0$. Évidemment, il s'agit d'un système physique réel et il semble logique que cette dérivée soit égale à droite et à gauche, i.e. en $\omega = 0^+$ et $\omega = 0^-$. L'objectif est donc d'estimer cette dérivée et de modéliser l'équation 2.3.0.11 par une expression linéaire de même pente. Pour cela, les développements de Taylor sont nécessaires. Premièrement, il peut être remarqué qu'une forme de cosinus et sinus cardinaux apparaissent dans les expressions de X et Y . Il convient tout d'abord d'étudier le terme suivant A identifiable dans $X(k+1)$ en 0. Pour tout $a, b \in \mathbb{R}$ et $x \in \mathbb{R}^*$:

$$A(x) = \frac{1}{x} [\sin(a + bx) - \sin(a)] = \frac{1}{x} [\sin(a)\cos(bx) + \cos(a)\sin(bx) - \sin(a)] \quad (2.3.0.7)$$

$$\underset{x \rightarrow 0}{=} \frac{1}{x} \left[\sin(a) \left[1 - \frac{(bx)^2}{2} + o(x^2) \right] + \cos(a) [bx + o(x^2)] - \sin(a) \right] \quad (2.3.0.8)$$

$$= -\frac{b^2}{2} \sin(a) x + b \cos(a) \quad (2.3.0.9)$$

Pour assurer la continuité du gradient, la pente du modèle en $\omega(k) = 0$ doit donc être égale à :

$$\frac{dA}{dx} = -\frac{b^2}{2} \sin(a) \quad (2.3.0.10)$$

En appliquant la même méthode à l'expression présente dans $Y(k+1)$, le modèle en $\omega(k) = 0$ devient :

$$\omega(k) = 0 \implies \begin{cases} X(k+1) = X(k) + v(k) \left[\cos(\theta(k)) T_s - \frac{T_s^2}{2} \sin(\theta(k)) \omega(k) \right] \\ Y(k+1) = Y(k) + v(k) \left[\sin(\theta(k)) T_s + \frac{T_s^2}{2} \cos(\theta(k)) \omega(k) \right] \\ \theta(k+1) = \theta(k) \end{cases} \quad (2.3.0.11)$$

Finalement, le modèle peut également être représenté sous la forme de matrice de passage homogène afin de l'intégrer plus facilement avec le modèle géométrique du bras. La définition d'un point $\mathcal{X} = [X, Y, Z]^T$ en coordonnées homogènes est notée $\bar{\mathcal{X}}$ avec $\bar{\mathcal{X}} = [X, Y, Z, 1]^T$

Ainsi, si \mathcal{X}_{b_k} est un point défini dans le repère de la base mobile à l'instant t_k tel que $\mathcal{X}_{b_k} = [X(k), Y(k), Z(k)]^T$, la relation devient :

$$\begin{aligned} \bar{\mathcal{X}}_{b_k} &= \begin{bmatrix} \cos(\theta(k+1) - \theta(k)) & -\sin(\theta(k+1) - \theta(k)) & 0 & X(k+1) - X(k) \\ \sin(\theta(k+1) - \theta(k)) & \cos(\theta(k+1) - \theta(k)) & 0 & Y(k+1) - Y(k) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \bar{\mathcal{X}}_{b_{k+1}} \\ &= {}^{b_k}H_{b_{k+1}} \bar{\mathcal{X}}_{b_{k+1}} \end{aligned} \quad (2.3.0.12)$$

2.4 Modélisation cinématique du bras

Pour le bras, la relation entre la variation de son vecteur d'état et son vecteur de commande est donnée $\forall i \in \llbracket 1, 5 \rrbracket$ par :

$$q_i(k+1) = q_i(k) + \dot{q}_i(k) T_s \quad (2.4.0.1)$$

Pour passer de l'espace articulaire à l'espace cartésien, il est nécessaire d'utiliser le modèle géométrique direct. Ce modèle peut être représenté sous la forme d'une matrice de passage homogène convertissant un point $\mathcal{X}_{cee} = [X_{cee}, Y_{cee}, Z_{cee}]^T$ exprimé dans le repère de l'organe terminal et un point $\mathcal{X}_b = [X_b, Y_b, Z_b]^T$ exprimé dans le repère de la base mobile :

$$\bar{\mathcal{X}}_b = {}^b H_{cee}(\chi_a) \bar{\mathcal{X}}_{cee} \quad (2.4.0.2)$$

Pour définir la matrice jacobienne du bras, il est nécessaire d'introduire la notation du torseur cinématique en un point. Dans ces travaux, le torseur cinématique d'un point P appartenant à un solide défini par un repère F_1 en mouvement par rapport à un deuxième solide défini par un repère F_0 exprimé dans un repère F_2 est noté comme suit :

$${}^{F_2} T_{P \in F_1 / F_0} = \left[\begin{array}{c} V_{P \in F_1 / F_0} \\ \Omega_{F_1 / F_0} \end{array} \right]_{F_2} \quad (2.4.0.3)$$

La matrice jacobienne du bras J_a est obtenue en dérivant la matrice de passage ${}^b H_{cee}$. Elle permet alors de mettre en relation le vecteur de commande u_a et le torseur cinématique de la caméra :

$${}^{F_{cee}} T_{O_{cee} \in F_{cee} / F_b} = \left[\begin{array}{c} V_{O_{cee} \in F_{cee} / F_b} \\ \Omega_{F_{cee} / F_b} \end{array} \right]_{F_{cee}} = {}^{F_{cee}} J_{O_{cee}}(\chi_a) \cdot u_a = J_a(\chi_a) \cdot u_a \quad (2.4.0.4)$$

Pour obtenir ce modèle géométrique direct et cette jacobienne, plusieurs solutions sont envisageables. La première est le calcul analytique, souvent laborieux et dépendant de la plateforme considérée. La deuxième solution est numérique à travers l'utilisation d'une librairie d'algorithmes de dynamique des corps rigides, comme *Pinocchio* [Carpentier et al., 2021], basée sur une modélisation standardisée, ici en l'occurrence un modèle *URDF* (Unified Robot Description Format). Le désavantage des solutions numériques est de ne pas permettre de dérivation analytique, résultant généralement à l'utilisation de méthodes d'estimation numérique de gradient réalisant une approximation, et nécessitant surtout un coût de calcul significatif. Pour résoudre ce problème, les librairies comme *Pinocchio* implémentées par *template* permettent l'utilisation de variables symboliques permettant d'éviter ces écueils.

2.5 Modélisation cinématique de la tête

Pour la tête, la méthode de modélisation est identique à celle du bras. La relation entre la variation de son vecteur d'état et son vecteur de commande est donnée par :

$$h_1(k+1) = h_1(k) + \dot{h}_1(k) T_s \quad (2.5.0.1)$$

Ainsi, de la même manière, le modèle géométrique direct peut être utilisé pour obtenir la matrice de passage homogène convertissant un point $\mathcal{X}_{c_h} = [X_{c_h}, Y_{c_h}, Z_{c_h}]^T$ exprimé dans le repère de l'organe terminal et un point $\mathcal{X}_b = [X_b, Y_b, Z_b]^T$ exprimé dans le repère de la base mobile :

$$\bar{\mathcal{X}}_b = {}^b H_{c_h}(\chi_h) \bar{\mathcal{X}}_{c_h} \quad (2.5.0.2)$$

2.6 Modélisation cinématique du manipulateur mobile

Les enjeux d'une réelle coordination entre ces sous-systèmes ont déjà été énumérés, et ils sont pris en compte dès l'étape de la modélisation. En effet, toutes les méthodes de gestion de redondance basées sur la jacobienne d'un bras robotique peuvent être généralisées à un manipulateur mobile. Mais pour cela, il est nécessaire de la définir et de la calculer.

L'objectif de cette partie est donc d'établir la jacobienne ${}^{F_{cee}} \bar{J}_{c_{ee}}$ complète du manipulateur mobile définie telle que :

$${}^{F_{cee}} T_{O_{c_{ee}} \in F_{c_{ee}}/F_0} = \begin{bmatrix} V_{O_{c_{ee}} \in F_{c_{ee}}/F_0} \\ \Omega_{F_{c_{ee}}/F_0} \end{bmatrix}_{F_{c_{ee}}} = {}^{F_{cee}} \bar{J}_{O_{c_{ee}}} \cdot u_{mm} = \bar{J}_{mm} \cdot u_{mm} \quad (2.6.0.1)$$

Il faut noter que \bar{J}_{mm} n'est théoriquement pas une jacobienne puisque $u_{mm} \neq \dot{\chi}_{mm}$ mais cet amalgame est fait dans ces travaux par souci de simplicité. Premièrement, il faut décomposer le mouvement selon les sous-systèmes :

$$T_{O_{c_{ee}} \in F_{c_{ee}}/F_0} = T_{O_{c_{ee}} \in F_{c_{ee}}/F_b} + T_{O_{c_{ee}} \in F_b/F_0} \quad (2.6.0.2)$$

2.6.1 Terme relatif au mouvement du bras

Le premier terme de 2.6.0.2 correspond à la contribution du bras par rapport à la base mobile. D'après les calculs détaillés précédemment qui sont déjà réalisés dans le repère adéquat, il vient :

$${}^{F_{cee}} T_{O_{c_{ee}} \in F_{c_{ee}}/F_0} = \begin{bmatrix} V_{O_{c_{ee}} \in F_{c_{ee}}/F_b} \\ \Omega_{F_{c_{ee}}/F_b} \end{bmatrix}_{F_{c_{ee}}} = [0_{6 \times 2} \quad J_a] \cdot u_{mm} = \bar{J}_a \cdot u_{mm} \quad (2.6.1.1)$$

en posant $\bar{J}_a = [0_{6 \times 2} \quad J_a]$ et $0_{m \times n}$ la matrice nulle de taille m par n .

2.6.2 Terme relatif au mouvement de la base mobile

Le deuxième terme de 2.6.0.2 traduit la contribution de la base mobile par rapport au repère global. Ce mouvement s'exprime facilement dans le repère F_b au point O_b :

$$T_{O_b \in F_b / F_0} = \begin{bmatrix} V_{O_b \in F_b / F_0} \\ \Omega_{F_b / F_0} \end{bmatrix}_{F_b} = \begin{bmatrix} v \cdot \vec{x}_b \\ \omega \cdot \vec{z}_b \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & 0_{4 \times 7} & & & \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot u_{mm} = {}^{F_b} \bar{J}_b \cdot u_{mm} \quad (2.6.2.1)$$

Il faut maintenant encore projeter cette jacobienne dans le repère F_{cee} et passer du point O_b au point O_{cee} . La matrice de transformation de torseur cinématique permettant d'effectuer ce changement est notée ${}^{F_{cee}} X_{F_b}$. En supposant que ${}^{F_{cee}} R_{F_b}$ soit la matrice de rotation entre les repères F_{cee} et F_b , cette matrice est définie telle que :

$${}^{F_{cee}} X_{F_b} = \begin{bmatrix} {}^{F_{cee}} R_{F_b} & -{}^{F_{cee}} R_{F_b} \cdot \overrightarrow{O_b O_{cee} \times} \\ 0_{3 \times 3} & {}^{F_{cee}} R_{F_b} \end{bmatrix} \quad (2.6.2.2)$$

où la notation $\vec{u} \times$ définit la matrice de pré-produit vectoriel par un vecteur $u = [u_x \ u_y \ u_z]^T$ tel que le produit vectoriel $u \times v$ avec un vecteur quelconque v s'écrit :

$$u \times v = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} v \quad (2.6.2.3)$$

Une nouvelle fois, les bibliothèques de dynamique des corps rigides proposent des implémentations pour obtenir ces matrices à partir du modèle du robot. Et finalement :

$${}^{F_{cee}} T_{O_{cee} \in F_b / F_0} = {}^{F_{cee}} X_{F_b} \cdot {}^{F_b} T_{O_b \in F_b / F_0} = {}^{F_{cee}} X_{F_b} \cdot {}^{F_b} \bar{J}_b \cdot u_{mm} = {}^{F_{cee}} \bar{J}_b \cdot u_{mm} = \bar{J}_b \cdot u_{mm} \quad (2.6.2.4)$$

2.6.3 Jacobienne du manipulateur mobile

En reprenant l'équation 2.6.0.2 et en y injectant les termes calculés, l'expression de la jacobienne du système entier devient :

$$\bar{J}_{mm} = \bar{J}_b + \bar{J}_a \quad (2.6.3.1)$$

2.7 Modélisation des caméras

Le dernier maillon du système considéré doit encore être modélisé : les caméras. Le fonctionnement des caméras est traduit mathématiquement grâce au modèle sténopé, aussi appelé modèle *pinhole*. Ce modèle suppose que le système optique de la caméra respecte les conditions de Gauss et modélise le processus de formation des images au sein d'une caméra. Tout se passe comme si les rayons ne passaient que par

un seul point appelé centre optique (voir figure 2.2) : dans notre cas ce dernier se trouve être l'origine du repère caméra : $O_{c_{ee}}$ ou O_{c_h} . En utilisant encore une fois la notation matricielle des coordonnées homogènes, il est possible de décrire un premier processus par une projection perspective ${}^i H_c$ transformant un point exprimé dans le repère caméra $\mathcal{X}_c = [X_c \ Y_c \ Z_c]^T$ en un point image $\mathbf{x}_i = [x \ y \ Z_c]^T$ dans le plan $Z_c = 1$:

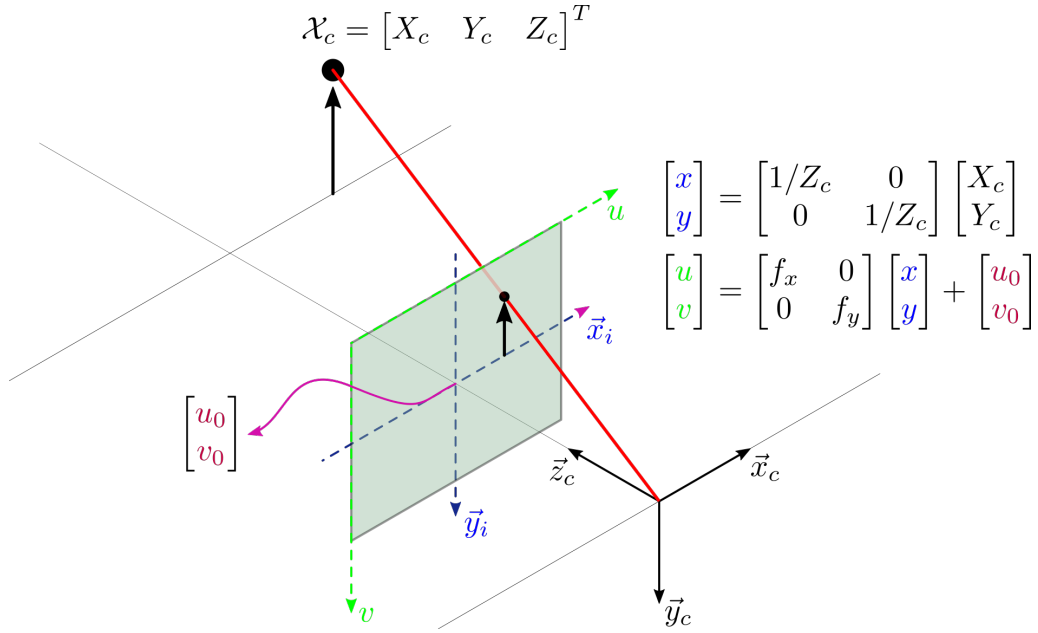


FIGURE 2.2 – Modèle sténopé

$$\bar{\mathbf{x}}_i = \begin{bmatrix} x \\ y \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} 1/Z_c & 0 & 0 & 0 \\ 0 & 1/Z_c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = {}^i H_c \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = {}^i H_c \bar{\mathcal{X}}_c \quad (2.7.0.1)$$

Ces coordonnées dans le plan image sont ensuite converties en pixels, correspondant aux deux premières coordonnées du vecteur homogène $[u \ v \ Z_c \ 1]^T$ à partir de f_x et f_y , prenant en compte les distances focales ainsi que les facteurs d'agrandissement, et coordonnées en pixels $[u_0 \ v_0]^T$ du centre de l'image :

$$\begin{bmatrix} u \\ v \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & 0 & u_0 \\ 0 & f_y & 0 & v_0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ Z_c \\ 1 \end{bmatrix} \quad (2.7.0.2)$$

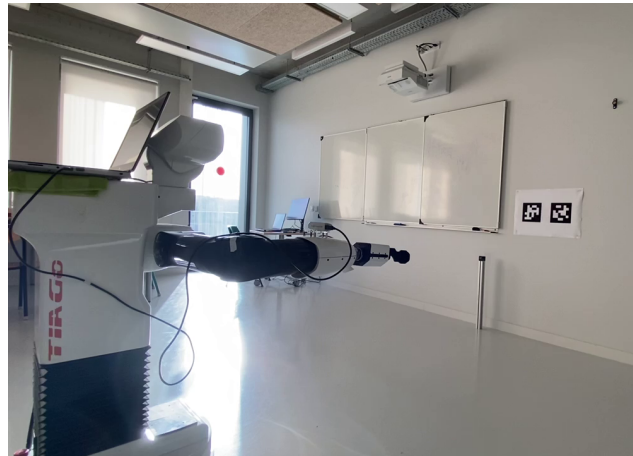
Chapitre 3

Asservissement visuel prédictif mono-caméra pour manipulateur mobile

L'objectif de ce chapitre est d'établir un contrôleur VPC permettant de piloter le système robotique modélisé. Il servira essentiellement de première base de travail permettant de valider la pertinence d'un tel schéma et de mieux comprendre les enjeux de la problématique afin d'améliorer la stratégie de commande dans les chapitres suivants.



(a) Configuration initiale



(b) Environnement expérimental

FIGURE 3.1 – Problématique étudiée dans ce chapitre

La finalité du travail de ce chapitre est de piloter le manipulateur mobile à partir d'une configuration initiale similaire à celle présentée en figure 3.1a de manière à positionner le repère de la caméra de l'organe terminal dans une situation de référence par rapport à une cible. Seule la caméra de l'organe terminal est utilisée dans les travaux de ce chapitre. Le robot commence alors avec le bras déployé afin de permettre à l'unique caméra de détecter la cible. Le robot doit se déplacer de 2 à

3m dans son environnement pour atteindre la situation de référence. La figure 3.1b présente l'environnement expérimental en question.

Après une brève mais nécessaire introduction de la cible utilisée, le chapitre débute par une étude du choix des indices visuels utilisés au cœur de la fonction de coût du problème de contrôle optimal. Celle-ci récapitule les avancées réalisées dans le domaine de l'asservissement visuel sur le choix de ces indices et analyse en quoi ces résultats sont également intéressants au sein d'une problématique de contrôle VPC de manipulateurs mobiles. Le chapitre présente ensuite le formalisme du schéma de contrôle VPC : la fonction de coût, les modèles de prédiction, ainsi que les contraintes du problème sont introduits. La stabilité du schéma, qui est rarement abordée dans la littérature sur les schémas VPC, est étudiée et assurée, constituant ainsi une contribution notable de ces travaux. Enfin, une section de résultats clôture ce chapitre. Celle-ci compile des résultats de simulation et expérimentaux permettant de valider les choix théoriques réalisés et de montrer la pertinence d'une stratégie d'asservissement visuel prédictif pour piloter un manipulateur mobile.

3.1 Présentation de la cible

Comme expliqué précédemment, les contributions de ce travail se situent principalement au niveau du contrôle. La partie perception utilise comme support des *aprilTags* [Wang and Olson, 2016] (Figure 3.2) et leurs algorithmes de détection sont très performants. Les parties de contrôle et de perception sont dépendantes, comme l'illustrent bien par les stratégies de DVS, et il est nécessaire de réduire l'impact des choix posés pour la seconde sur la commande. Premièrement, puisque la stratégie d'asservissement visuel retenue est basée sur des primitives géométriques seulement, le contenu et le contraste de la cible ne sont pas prépondérants. Deuxièmement, le point clef de cette phase de perception est l'extraction des points d'intérêt dans l'image. En effet, il faut que cette extraction se réalise suffisamment vite pour ne pas perturber la boucle de commande fonctionnant généralement à des fréquences supérieures à 5-10Hz. La détection d'aprilTag est rapide (de l'ordre de la milliseconde) mais ces performances sont évidemment éloignées des cas réels où l'environnement n'est pas instrumenté. Cependant, les avancées spectaculaires dans ces domaines, notamment grâce aux réseaux de neurones profonds, à l'apprentissage supervisé et à d'immenses bases de données, offrent aujourd'hui des performances incomparables. De plus, le déploiement de matériel informatique dédié permet de réduire le temps de calcul drastiquement et aujourd'hui, des réseaux de neurones publics et gratuits comme YOLO [Redmon et al., 2016] permettent de spécialiser un réseau et d'atteindre des temps de reconnaissance inférieurs à 20ms. Enfin, il est important de tenir compte que le modèle de cette cible est plan, et que cette particularité n'est également pas souvent réaliste. Il est donc nécessaire d'étudier l'influence de cette hypothèse et des précisions sont avancées dans les sections suivantes.

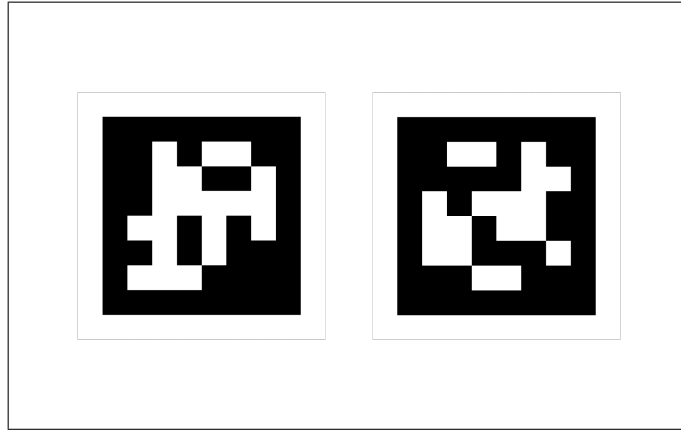


FIGURE 3.2 – Cible utilisée

3.2 Choix des indices visuels

Dans un contexte d’asservissement visuel, l’objectif est de faire converger un vecteur d’indices visuels S vers son vecteur de référence S^* . Les erreurs au cœur de la fonction de coût sont alors définis par $S - S^*$ et occupent une place centrale dans le schéma VPC. En effet, ce sont eux qui définissent la forme de l’espace de recherche et s’il est connu que le choix des indices visuels a un impact important sur les performances d’un asservissement visuel, il en va de même pour un schéma VPC.

Comme expliqué dans l’introduction, le choix de ces indices visuels s’est historiquement penché sur des primitives géométriques simples comme des points, droites [Espiau et al., 1992] [Chaumette, 1990]. Cependant, un premier point négatif de ces primitives est l’absence de généralité. En effet, une représentation générique pour tout objet, de formes diverses et variées, permettrait de construire un schéma de commande lui aussi générique. Ensuite, ces caractéristiques sont souvent à l’origine des problèmes déjà mentionnés de singularités, de minima locaux et de faible domaine de convergence [Chaumette, 2007]. Ces points peuvent être abordés en équilibrant la dépendance des primitives selon les différents degrés de liberté de la tâche considérée. Un moyen d’assurer cet équilibre est d’utiliser des caractéristiques découplées, c’est-à-dire de trouver des indices visuels qui ne traduisent chacun qu’un seul degré de liberté de la tâche considérée. Une manière de quantifier ce découplage est d’étudier les formes des matrices d’interaction [Tahri, 2004]. La forme idéale de cette matrice est alors diagonale. Malheureusement, cet objectif est certainement irréalisable, mais la définition d’une forme proche de cet idéal est possible grâce à l’utilisation des moments 2D de l’image [Chaumette, 2004] [Tahri and Chaumette, 2003]. En fait, l’objectif est de retrouver les propriétés de l’asservissement visuel basé position, mais en utilisant des caractéristiques visuelles extraites uniquement de l’image afin d’éviter toute estimation de la pose dans le schéma de contrôle.

C’est donc dans ce contexte que les moments de l’image et la théorie des invariants ont été introduits dans les schémas d’asservissement visuel. Dans cette section,

l'analyse et les propriétés relatives à ces primitives sont rappelées.

3.2.1 Matrice d'interaction

La matrice d'interaction est définie comme la jacobienne de l'image et elle est centrale dans le choix d'indices visuels. Introduite dans [Espiau et al., 1992] et devenue fondamentale dans le domaine de l'asservissement visuel, elle rend premièrement possible la complétion de la chaîne de jacobienne d'une modélisation robotique. C'est-à-dire qu'elle permet d'établir la relation complète entre le vecteur de commande et la variation des indices visuels dans l'image, et de se ramener ainsi à un problème de régulation. Dans cette chaîne, la matrice d'interaction L_s fait le lien entre le torseur cinématique de la caméra ${}^{F_c}T_c$ et la variation d'un indice visuel s :

$$\dot{s} = L_s \cdot {}^{F_c}T_c \quad (3.2.1.1)$$

Elle permet donc également d'analyser ce couplage mentionné entre un indice visuel S et les degrés de liberté de la tâche. L'idéal serait de trouver des matrices L_S avec un seul élément non nul, i.e. obtenir des indices visuels découplés, et d'éviter au maximum les non-linéarités sur cet élément.

Ainsi, si cette caméra est en bout de chaîne cinématique d'un système robotique de jacobienne J et de vecteur de commande u , il est possible de compléter la relation par :

$$\dot{s} = L_s \cdot \bar{J}_{mm} \cdot u_{mm} \quad (3.2.1.2)$$

Cette relation est fondamentale dans le domaine de l'asservissement visuel puisqu'il devient possible de réaliser de l'asservissement visuel 2D (IBVS). En effet, les problèmes peuvent être définis au plus proche des informations capteur et d'éviter les étapes de reconstruction nécessaires à l'asservissement visuel 3D (PBVS).

3.2.2 Projection perspective

La projection perspective est la méthode de projection la plus utilisée puisqu'elle modélise le fonctionnement des caméras classiques. À partir de cette méthode, une image plane est obtenue et de nombreux indices visuels peuvent être construits. Dans ce travail de thèse, deux catégories de primitives sont étudiées : les points d'intérêt directement, et les moments 2D. Les résiduels relatifs à la tâche de positionnement peuvent alors être définis.

— Points

Afin de contrôler les six degrés de liberté de la tâche de positionnement, les indices visuels classiquement utilisés sont les quatre coins de la cible et le vecteur S est composé de leurs coordonnées dans l'image :

$$S_p^p = [x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4]^T \quad (3.2.2.1)$$

La matrice d'interaction d'un point $\mathbf{x}_i = [x_i, y_i]^T$ projection du point 3D $\mathcal{X}_c = [X_c, Y_c, Z_c]^T$ défini dans F_c est classiquement donnée par les travaux fondateurs [Espiau et al., 1992] :

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = L_{\mathbf{x}_i} \cdot {}^{F_c}T_{O_c \in F_c/F} = \begin{bmatrix} -1/Z_c & 0 & x_i/Z_c & x_i y_i & -1 - x_i^2 & y_i \\ 0 & -1/Z_c & y_i/Z_c & 1 + y_i & -x_i y_i & -x_i \end{bmatrix} \cdot {}^{F_c}T_{O_c \in F_c/F} \quad (3.2.2.2)$$

Cette équation correspond à un seul point d'intérêt, la matrice d'interaction totale est alors donnée par :

$$L_{S_p^p} = \begin{bmatrix} L_{\mathbf{x}_{i1}} \\ L_{\mathbf{x}_{i2}} \\ L_{\mathbf{x}_{i3}} \\ L_{\mathbf{x}_{i4}} \end{bmatrix} \quad (3.2.2.3)$$

Premièrement, cette matrice est de dimension 8 par 6, une redondance d'information est donc déjà présente. De plus, l'utilisation de seulement trois points d'intérêt provoque l'apparition de minima locaux [Chaumette, 2007], même si ce problème est toujours présent avec l'utilisation de plus de points, quoique plus rare. Des configurations comme l'exemple classique des trois points alignés existent aussi et provoquent une chute du rang de cette matrice, traduisant un échec de commandabilité des six degrés de liberté de la tâche.

— Moments 2D

Comme déjà expliqué, l'utilisation du vecteur S_p^p présentent de nombreux inconvénients et l'objectif est de sélectionner des indices visuels adéquats pour contrôler chaque translation et rotation. En d'autres termes, des caractéristiques qui permettent de découpler et de linéariser autant que possible la relation entre l'espace de la tâche et l'espace de l'image sont recherchées. [Tahri, 2004] a montré que l'utilisation des moments permet de s'approcher de telles propriétés. En effet, leurs caractéristiques d'invariance à certaines transformations permet de les combiner pour réduire les dépendances avec les degrés de liberté de la tâche.

Les moments 2D dans l'image peuvent être définis de deux façons : continue ou discrète. Dans le cas continu et en admettant une fonction de densité binaire, si O est l'objet observé et O_p sa projection dans l'image, les moments d'ordre $i + j$ sont définis de la sorte :

$$m_{ij} = \iint_{O_p} x^i y^j dx dy \quad (3.2.2.4)$$

De premières propriétés géométriques de l'objet peuvent être déduites à partir des moments d'ordre 0 et 1 :

1. La surface a de l'objet correspond au moment d'ordre 0.

$$a = m_{00} \quad (3.2.2.5)$$

2. Le centre de gravité $\mathbf{x}_{i_g} = [x_g, y_g]^T$ se calcule à l'aide des moments de degré 0 et 1 :

$$x_g = m_{10}/m_{00} \quad (3.2.2.6)$$

$$y_g = m_{01}/m_{00} \quad (3.2.2.7)$$

La surface a semble intuitivement constituer un bon indice visuel pour contrôler la translation suivant l'axe optique z_c puisqu'elle est invariante localement aux autres mouvements. En effet, lorsque l'objet est centré et parallèle au plan image, les éléments de sa matrice d'interaction correspondant aux trois rotations et aux deux translations selon le plan image sont nuls [Tahri, 2004]. Les coordonnées du centre de gravité sont eux invariants à la rotation autour de l'axe optique et dépendent évidemment des translations selon les axes non-optiques. Ils peuvent ainsi être adéquats à traduire ces mouvements de translations selon x_c et y_c . Une autre famille de moment dérive du centre de gravité : les moments centrés. Ces derniers présentent l'important avantage d'être invariants par définition aux mouvements de translation dans l'image, i.e. selon les axes x_c et y_c . Les moments centrés d'ordre $i + j$ sont définis comme suit :

$$\mu_{ij} = \iint_{O_p} (x - x_g)^i (y - y_g)^j dx dy \quad (3.2.2.8)$$

Les moments centrés d'ordre 2 sont appelés moments d'inertie de l'objet. Dans l'image, ils permettent de construire une ellipse d'inertie équivalente à l'objet O_p considéré et donc de calculer ses axes principaux. A partir de ces derniers, il est possible de revenir à l'orientation de l'objet suivant l'axe optique z_c tel que :

$$\alpha = \frac{1}{2} \arctan \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \quad (3.2.2.9)$$

Ces quatre premiers indices visuels sont illustrés sur la figure 3.3 dans le cas de la cible considérée dans ces travaux.

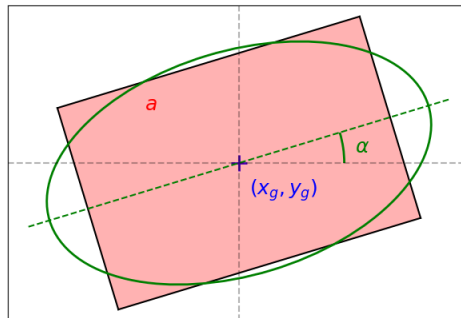


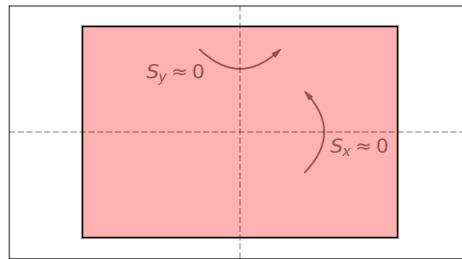
FIGURE 3.3 – Illustration des indices a , x_g , y_g et α

Les degrés de liberté restants, i.e. les rotations autour des axes non-optiques, sont historiquement les plus complexes à contrôler en asservissement visuel 2D. Mais il

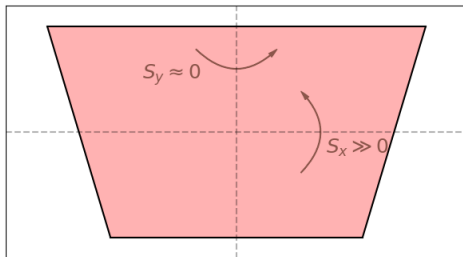
semble également assez intuitif qu'une quantification de ces rotations se retrouve dans l'analyse des asymétries de l'objet (voir figure 3.4). Cette notion d'asymétrie apparaît dans l'étude des moments d'ordre 3. En reconnaissance des formes, les coefficients d'asymétrie S_x et S_y (pour *skewness*) sont généralement utilisés et définis comme suit [Prokop and Reeves, 1992] :

$$S_x = \mu_{30}/\mu_{20}^{3/2} \quad (3.2.2.10)$$

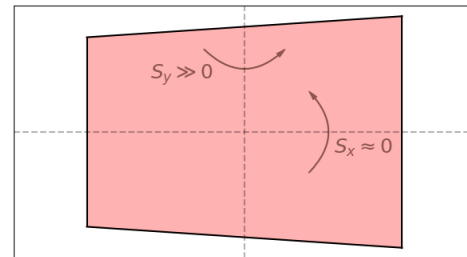
$$S_y = \mu_{03}/\mu_{02}^{3/2} \quad (3.2.2.11)$$



(a) Référence



(b) Rotation autour de x_c



(c) Rotation autour de y_c

FIGURE 3.4 – Illustration des indices S_x et S_y

Ces propositions forment donc une solution solide au problème de découplage. Mais les travaux menés dans [Tahri, 2004] améliorent encore ces résultats en proposant une nouvelle méthode de calcul d'invariants. En partant de ces six indices visuels intuitifs, ils proposent un ensemble de caractéristiques permettant d'obtenir ce qui reste encore actuellement un découplage proche de l'idéal. Ces résultats réduisent au maximum les non-linéarités relatives à la surface grâce à une étape de normalisation, ainsi que les dépendances des indices quantifiant les rotations autour des axes non-optiques. Les indices visuels, utilisés dans ce travail, sont les suivants :

$$S_p^m = [x_n, y_n, a_n, s_x, s_y, \alpha]^T \quad (3.2.2.12)$$

Ils sont définis comme suit :

- La grandeur a_n , a pour l'objectif de représenter la translation suivant l'axe optique z_c . Pour une valeur de profondeur Z^* et de surface a^* désirées, a_n s'écrit :

$$a_n = Z^* \sqrt{\frac{a^*}{a}} \quad (3.2.2.13)$$

- Les coordonnées normalisées x_n et y_n du centre de gravité visent à contrôler les translations suivant les axes non-optiques x_c et y_c , respectivement. Elles sont définies par :

$$x_n = a_n x_g \quad (3.2.2.14)$$

$$y_n = a_n y_g \quad (3.2.2.15)$$

- L'orientation α reste inchangée et permet donc de contrôler la rotation autour de l'axe optique z_c .

$$\alpha = \frac{1}{2} \arctan \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \quad (3.2.2.16)$$

- Les indices d'asymétrie s_x et s_y permettent de gérer le cas d'objets symétriques comme la cible considérée dans ce travail. Dans les cas d'objets non-symétriques, d'autres indices sont certainement plus performants [Chaumette, 2004]. Ces grandeurs sont des combinaisons modifiées des invariants de Hu [Hu, 1962] :

$$s_x = (c_2 c_3 + s_2 s_3) / K \quad (3.2.2.17)$$

$$s_y = (c_2 s_3 - s_2 c_3) / K \quad (3.2.2.18)$$

avec $c_1 = \mu_{20} - \mu_{02}$, $s_1 = 2\mu_{11}$, $c_2 = \mu_{03} - 3\mu_{21}$, $s_2 = \mu_{30} - 3\mu_{12}$, $c_3 = c_1^2 - s_1^2$, $s_3 = 2s_1 c_1$, $I_1 = c_1^2 + s_1^2$, $I_3 = \mu_{20} + \mu_{02}$, et $K = I_1 I_3^{3/2} / \sqrt{a}$

Ces choix ont été dirigés par l'analyse des matrices d'interaction correspondant à ces indices. Leurs expressions analytiques sont données dans [Tahri, 2004]. Cependant, il est important de revenir sur un point de ces calculs qui est l'hypothèse d'un objet observé plan ou présentant une surface de limbe plane [Chaumette, 2004]. Cette approximation se justifie souvent en réalité grâce à la robustesse des schémas d'asservissement visuel. Aussi, si les matrices d'interaction sont utilisées dans le modèle de prédiction du schéma VPC, le cas général d'un objet non plan augmenterait grandement les temps de calcul puisque leurs expressions sont très complexes. L'approximation d'une cible plane permet de simplifier les calculs et de réaliser un bon compromis. Ceci rejoint également la justification du choix de cible fait dans ces travaux.

Cette hypothèse étant précisée, la forme analytique de la matrice d'interaction du vecteur S_p^m à la pose désirée peut être calculée à partir des travaux [Chaumette, 2004]. Après simplification, cette matrice devient :

$$L_{S_p^m} = \begin{bmatrix} C_1 & 0 & 0 & 0 & C_2 & 0 \\ 0 & C_1 & 0 & C_3 & 0 & 0 \\ 0 & 0 & C_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & C_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & C_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & C_6 \end{bmatrix} \quad (3.2.2.19)$$

où les C_i sont des valeurs numériques de même ordre de grandeur. Une application numérique a par exemple été réalisée dans [Chaumette, 2004].

Le résultat de découplage ainsi obtenu n'est pas parfait, mais proche de l'idéal. De plus, ce découplage n'est que local autour de la position désirée. D'après les expressions analytiques des matrices d'interaction, ce découplage persiste tant que la caméra fait face à la cible, c'est-à-dire tant que le plan objet et le plan image restent quasi-parallèles. Une hypothèse qui a également du sens puisque cette condition est nécessaire pour percevoir l'objet. Cette condition peut cependant être remise en cause si l'orientation entre la caméra de l'organe terminal et la cible est grande.

De plus, les résultats évoqués se généralisent au cas discret et ces derniers peuvent également être utilisés. Certaines informations sont perdues comme la surface par exemple, mais des solutions existent pour régler ce problème [Tahri, 2004]. L'utilisation des moments discrets est intéressante lorsque les calculs analytiques des moments continus sont trop complexes. Pour le cas étudié, les travaux [Steger, 1996] offrent des expressions génériques permettant de les calculer dans le cas considéré. C'est pourquoi les moments 2D continus ont été privilégiés.

3.3 Synthèse du contrôleur

La section précédente a permis de définir des informations au plus proche des données des caméras permettant de contrôler la caméra du robot. Grâce à ces résultats, il est donc possible de construire une fonction de coût basée sur l'erreur relative au vecteur d'indice visuel et donc le schéma de contrôle VPC.

Pour rappel, le contrôle prédictif nécessite quatre éléments principaux : une référence, un modèle de prédiction, une fonction de coût et une méthode de résolution. Grâce à ces éléments, le problème de contrôle optimal peut être défini, puis résolu. Cette section présente dans un premier temps un succinct état de l'art sur les techniques de résolution numérique en contrôle optimal. La fonction de coût, les contraintes, ainsi que les modèles de prédiction sont ensuite présentées. Les contributions relatives à la stabilité sont également détaillées. Pour finir, l'effort d'implémentation permettant la mise en temps réel, les bonnes performances et la portabilité du schéma présenté est explicité. Ce dernier aspect peut être considéré comme une contribution du travail.

3.3.1 Méthodes de résolution en contrôle optimal

Les méthodes de résolution des problèmes de contrôle optimal peuvent être divisées en trois groupes [Diehl and Gros, 2011] : les méthodes basées sur l'équation de Hamilton-Jacobi-Bellman, les méthodes indirectes, et les méthodes directes. Les deux premiers groupes permettent une résolution d'un problème de contrôle optimal défini en temps continu (*continuous time optimal control*). Les méthodes directes transforment quant à elles le problème original de contrôle optimal en temps continu en temps discret pour le résoudre.

La première classe de méthode souffre de la "malédiction" de la dimension. Ce concept, formulé par Richard Bellman, fait référence aux défis et complications qui surgissent lorsque le nombre de dimensions d'un problème augmente. Ces méthodes restent ainsi limitées à des dimensions d'état de système très faible (≤ 6) et ne permettent pas de gérer toutes sortes de contraintes. La classe des méthodes indirectes utilise des conditions d'optimalité pour éliminer les commandes du problème et se ramener à un problème de valeur limite. Sous réserve que cette conversion soit possible, ce qui ne l'est pas toujours, deux principaux inconvénients sont présents : les équations obtenues restent souvent difficiles à résoudre et nécessitent généralement des techniques de résolution spécialisées selon les contraintes actives.

En raison de ces inconvénients, ces classes de méthodes ne sont donc que rarement utilisées en contrôle optimal, et encore moins au cœur d'un schéma MPC. La dernière classe d'approche, dont le principe est souvent résumé à "d'abord discrétiser, puis optimiser", couvre largement les techniques les plus répandues et les plus utilisées dans l'état de l'art du contrôle optimal. Le problème discrétisé devient un problème d'optimisation non-linéaire classique (Non-Linear Programming Problem, ou encore *NLP*). Les excellentes avancées dans ce domaine sont alors directement utilisables, et la prise en compte de toutes sortes de contraintes devient alors possible. En fait, ce sont très certainement les méthodes directes et l'utilisation des outils d'optimisation non-linéaire qui ont permis la nouvelle émergence et la popularisation des techniques de contrôle prédictif. Les travaux de cette thèse s'inscrivent ainsi également dans cette démarche, et le problème de contrôle optimal défini ci-après est donc discrétisé.

Enfin, si toutes les méthodes directes sont basées sur une paramétrisation finie de la trajectoire prédite, elles diffèrent dans la manière dont la trajectoire d'état est traitée. Ceci implique à nouveau plusieurs techniques de résolution au sein des méthodes directes. Ces dernières sont définies et étudiées dans la section 3.3.8 après que le problème d'optimisation soit entièrement défini.

3.3.2 Présentation du problème

Comme expliqué précédemment, l'asservissement visuel prédictif est le résultat de la combinaison du contrôle prédictif et de l'asservissement visuel. Il partage donc les caractéristiques de ces deux stratégies de contrôle. Comme la commande prédictive, le VPC consiste en une résolution répétée d'un problème de contrôle optimal. Cette répétition se fait à intervalles de temps réguliers définis par les itérations k de la boucle de contrôle correspondant au temps $t = kT_s$. À chaque itération de la

boucle de contrôle, il s'agit donc de trouver une séquence de commandes optimale $U^*(\cdot)$ qui minimise une fonction de coût J_{N_p} sur un horizon de N_p prédictions sous un ensemble de contraintes $C(U(\cdot))$. La séquence de commandes optimale obtenue est un vecteur de dimension N_c appelé horizon de contrôle. Ainsi, une séquence de commandes est composée de N_c vecteurs de commande u_{mm} :

$$U_k(\cdot) = [u_{mm}(k), \dots, u_{mm}(k + N_c - 1)] \quad (3.3.2.1)$$

Il est possible de définir un horizon de commande plus petit que l'horizon de prédiction : $N_p > N_c$ pour des questions de puissance de calcul. Les N_c premières prédictions sont calculées avec des commandes indépendantes, tandis que les prédictions restantes sont obtenues en utilisant la dernière commande de la séquence. Le comportement du système est anticipé grâce au modèle de prédiction et une séquence de commandes qui optimise un critère de performance est déterminée. À une itération k , le problème optimal est défini comme suit :

$$U_k^*(\cdot) = \min_{U_k(\cdot)} J(k, U_k(\cdot)) \quad (3.3.2.2)$$

avec

$$J(k, U_k(\cdot)) = J_{N_p}(S(k), Z_c(k), \chi_a(k), U_k(\cdot)) = \sum_{p=k+1}^{k+N_p} F(p) \quad (3.3.2.3)$$

et

$$F(p) = F_{vs}(p) \quad (3.3.2.4)$$

sous les contraintes :

$$\hat{S}(k) = S(k) \quad (3.3.2.5a)$$

$$\hat{Z}_c(\hat{S}(k)) = Z_c(S(k)) \quad (3.3.2.5b)$$

$$\hat{\chi}_a(k) = \chi_a(k) \quad (3.3.2.5c)$$

$$\hat{S}(p+1) = f_S(\hat{S}(p), U(p)) \quad (3.3.2.5d)$$

$$\hat{Z}_c(p+1) = f_{Z_c}(\hat{Z}_c(p), U(p)) \quad (3.3.2.5e)$$

$$\hat{\chi}_a(p+1) = f_{\chi_a}(\hat{\chi}_a(p), U(p)) \quad (3.3.2.5f)$$

$$C(U^*(\cdot)) \leq 0 \quad (3.3.2.5g)$$

Le coût $F_{vs}(p)$ traduit la tâche d'asservissement visuel à réaliser et s'exprime à l'aide des indices visuels introduits au chapitre précédent. Cet aspect est formalisé par l'équation (3.3.2.4) et l'expression quadratique de $F_{vs}(p)$ fonction de l'erreur $S - S^*$ est détaillée plus loin. L'ajout du chapeau indique que la grandeur concernée est prédite, c'est-à-dire qu'elle se différencie de sa grandeur réelle. Par exemple, \hat{S} , \hat{Z}_c et $\hat{\chi}$ correspondent respectivement aux prédictions réalisées des indices visuels, des informations 3D Z_c de ces indices visuels, et de la configuration du bras grâce aux modèles de prédiction définis par les équations (3.3.2.5d), (3.3.2.5e) et (3.3.2.5f). Le

modèle de prédiction f_S est spécifique au vecteur d'indice visuel utilisé et les différents modèles considérés ici seront détaillés par la suite. Les prédictions permettent de calculer les coûts $F(p)$ à chaque prédiction $p \in \llbracket k + 1, k + N_p \rrbracket$. Les équations (3.3.2.5a), (3.3.2.5b) et (3.3.2.5c) traduisent l'étape de mesure.

Enfin, les contraintes $C(U^*(\cdot)) \leq 0$ traduisent les contraintes du problème. Elles peuvent être de différentes sortes : nécessaires intrinsèquement à la notion d'asservissement visuel comme les contraintes de visibilité, physiques comme la prise en compte des butées articulaires ou des performances des actionneurs, ou encore relatives à la stabilité.

Les stratégies de contrôle prédictif se différencient de la planification de trajectoire par contrôle optimal par l'étape de rétroaction. Cette étape permet alors d'effectuer un contrôle en boucle fermée, permettant d'une part de gérer les perturbations ou les erreurs de modélisation. Pour intégrer la rétroaction dans le schéma, seule la première commande de la séquence optimale u_{mm}^* est appliquée au robot jusqu'à la prochaine itération de la boucle de contrôle et le processus est alors répété à $t = (k + 1)T_s$. En utilisant le nouvel état du système, l'horizon de commande et de prédiction est déplacé vers l'avant. La période de la boucle de contrôle T_s est généralement fixe et égale à la période de prédiction.

3.3.3 Analyse du schéma VPC

D'un point de vue théorique, tout comme sur le principe du MPC, l'idéal serait d'utiliser des horizons de prédiction et de contrôle infinis [Allgower et al., 2004]. Cependant, ce problème ne peut évidemment pas être résolu à cause de la puissance de calcul nécessaire pour résoudre ce problème complexe. En effet, le problème d'optimisation considéré consiste en des modèles de prédiction et des fonctions de coût hautement non-linéaires et un nombre conséquent de variables d'optimisation (80 déjà avec $N_c = 10$). La résolution de ce problème d'optimisation doit être réalisée en un temps très limité. En robotique, la fréquence de la boucle de contrôle est généralement au minimum 5Hz, le problème doit donc être résolu en moins de 200ms.

Du fait de l'utilisation d'un horizon de prédiction fini, les prédictions réalisées à toute itération k diffèrent des résultats en boucle fermée même en l'absence de perturbations. En effet, si l'horizon de prédiction est trop court pour atteindre la situation de référence dès le début de la commande, les objectifs à long terme ne peuvent pas être directement pris en compte. Ce constat est primordial pour comprendre les enjeux relatifs à la stabilité du schéma de contrôle. Il n'est donc nullement garanti qu'une minimisation répétée sur une fenêtre glissante conduise à une solution optimale pour le problème global. Évidemment, plus l'horizon utilisé est court, plus ce problème est important. Toute la théorie de la stabilité des schémas MPC est basée sur ce constat et tente justement de se ramener à un horizon de prédiction infini grâce à l'ajout de contraintes ou de propriétés judicieusement choisies. Cet élément est donc également central dans un schéma VPC et des contributions ont été apportées à ce sujet.

De plus, un schéma MPC permet l'utilisation directe de modèles non-linéaires pour la prédiction. Il s'agit d'une propriété très appréciable en comparaison de stratégies de commande nécessitant les expressions analytiques des gradients ou l'inversion des modèles directs. Le schéma a effectivement l'avantage de permettre l'utilisation des outils de contrôle optimal : il est possible d'utiliser les nombreux solveurs existants, de choisir les mieux adaptés au problème considéré et de s'affranchir d'une refonte totale du contrôleur et de lourds calculs analytiques lors d'une modification. Cette caractéristique a permis certaines contributions basées sur la modification en ligne du problème d'optimisation. Le MPC permet également de prendre explicitement en compte de nombreuses sortes de contraintes, ce qui est souvent difficile à gérer autrement. Il s'agit évidemment d'un avantage qui a déjà été évoquée de nombreuses fois, mais cela provient justement de la possibilité de profiter des outils existants du contrôle optimal.

Cependant, l'utilisation de ces solveurs ne se fait pas sans difficultés. Pour que la rétroaction soit efficace, il est nécessaire qu'une solution du problème soit trouvée en un temps limité. Cet aspect est fondamentalement différent de la planification de trajectoire qui n'a généralement pas cette contrainte de temps de résolution. Cette différence rejoint les problématiques de stabilité liées à la taille de l'horizon de prédiction.

Enfin, l'utilisation d'informations au plus proche des données capteurs permet de s'affranchir d'étapes d'estimation pouvant altérer les performances du système en régime permanent. L'expression de la tâche de positionnement dans l'image semble intéressante vis-à-vis de ce constat. De plus, l'utilisation des moments de l'image et d'une judicieuse pondération de ces derniers pour améliorer les performances du système est une contribution des travaux.

3.3.4 Définition du coût F_{vs}

Le coût F_{vs} définit la tâche principale d'asservissement visuel. Il s'agit d'une tâche de régulation et la fonction de coût correspondante doit donc proposer de bonnes propriétés de continuité, de dérivation et de minima locaux. En effet, cette fonction doit présenter un unique minimum global correspondant à $S = S^*$. La forme privilégiée de cette fonction de coût dans les problèmes de contrôle optimal est la forme quadratique pour obtenir un coût semi-défini positif. Il convient alors également de définir une matrice de pondération Q_S semi-définie positive et le coût devient :

$$F_{vs} = \left[\hat{S} - S^* \right]^T Q_S \left[\hat{S} - S^* \right] \quad (3.3.4.1)$$

Pour ce premier schéma d'asservissement visuel prédictif, l'objectif est de définir la fonction de coût entièrement dans l'image. Cette caractéristique permet d'obtenir une excellente précision après convergence grâce à l'utilisation de données au plus proche des capteurs et elle est quasiment inexistante dans la littérature pour contrôler des manipulateurs mobiles. Les rares travaux définissant la fonction de coût dans l'image n'utilisent pas nécessairement les moments de l'image, et ne profitent donc

pas des avantages énoncés précédemment. Ainsi, le coût F_{vs} peut être défini de deux manières différentes pour ce schéma de contrôle :

1. À partir des points d'intérêt :

$$F_{vs} = \left[\hat{S}_p^p - S_p^{p*} \right]^T Q_{S_p^p} \left[\hat{S}_p^p - S_p^{p*} \right] \quad (3.3.4.2)$$

2. À partir des moments 2D :

$$F_{vs} = \left[\hat{S}_p^m - S_p^{m*} \right]^T Q_{S_p^m} \left[\hat{S}_p^m - S_p^{m*} \right] \quad (3.3.4.3)$$

L'utilisation des moments permet d'avoir une capacité d'action supplémentaire sur l'espace de recherche du problème d'optimisation. En effet, pour le premier coût F_{vs} défini à partir des coordonnées des points d'intérêt, la matrice de pondération $Q_{S_p^p}$ ne permet de gérer que la courbure de cet espace puisque les pondérations relatives aux points d'intérêt sont égales. Il ne semble pas pertinent d'équilibrer cette matrice différemment puisqu'aucun découplage n'est réalisé. En général donc, $Q_{S_p^p} = \lambda I_{8 \times 8}$ avec $\lambda \in \mathbb{R}^*$.

Il est possible de procéder différemment quand les moments sont utilisés puisque cette fois, en plus de la courbure, le découplage obtenu permet également d'influencer la forme de l'espace de recherche. La trajectoire de l'organe terminal peut alors être modifiée en privilégiant certains indices visuels par rapport à d'autres. De plus, les mouvements traduits par les indices visuels qui composent S_p^m sont identifiés et il devient donc possible d'agir sur la trajectoire cartésienne obtenue grâce aux différences de pondération des six degrés de liberté de la tâche. Dans ce cas donc, la matrice $Q_{S_p^m}$ est simplement diagonale.

3.3.5 Définition des modèles de prédiction

Analyse du problème et choix des modèles

La fonction de coût étant définie, il reste à établir les modèles de prédiction permettant de les calculer. Différents types de modèles sont généralement proposés [Allibert et al., 2010b] : il peut s'agir d'un modèle local basé sur la matrice d'interaction et d'une approximation linéaire de l'intégration ou d'un modèle global non-linéaire et reposant sur les matrices de transformation homogènes. Ce choix dépend principalement de la disponibilité des données 3D, car le modèle global est peu robuste aux grandes erreurs d'estimation ou de mesure des informations 3D. En effet, comme expliqué dans [Le Flecher, 2020], la profondeur est utilisée deux fois : pour projeter les points 3D dans le repère caméra grâce à l'équation 2.7.0.1 et pour l'opération inverse. La première opération est inversement proportionnelle à la profondeur et, comme $Z_c \gg 0$, celle-ci est considérée peu sensible aux erreurs sur Z_c . Cependant, l'opération inverse est proportionnelle à la profondeur et ainsi très sensible aux erreurs sur Z_c . Pour le modèle local, la matrice d'interaction donnée

dans l'équation 3.2.2.2 est utilisée, laquelle est également inversement proportionnelle à la profondeur et une estimation ou mesure grossière peut donc suffire. Une hypothèse souvent réalisée en asservissement visuel lorsque la matrice d'interaction est utilisée est d'ailleurs l'approximation de la profondeur par sa valeur de référence. Cette hypothèse permet d'éviter une mesure d'information 3D ou une étape de reconstruction 3D en utilisant cette robustesse. Cette approximation permet d'obtenir de bonnes performances lorsque la problématique étudie des scénarios à faible déplacement pour des bras robotique par exemple. Cependant, elle peut être remise en question lors de scénarios de robotique mobile où l'approximation peut être trop grossière et entraîner des échecs de la loi de commande [Allibert et al., 2010b]. Une approximation est généralement encore réalisée via une intégration d'Euler. La prise en compte des contraintes 3D devient également indirecte, comme une formulation cartésienne d'évitement d'obstacle par exemple. En effet, l'utilisation du modèle local est plus adapté à un problème formulé entièrement dans l'image.

Une deuxième différence majeure concerne le domaine de validité de ces modèles. Le modèle de prédiction local est d'autant plus précis que le déplacement considéré pour une étape d'intégration est court (cf. intégration d'Euler). Ce déplacement est donc lié aux périodes de prédiction T_s et aux vitesses maximales des actionneurs. Ces paramètres déterminent la longueur de l'horizon de prédiction. Et il est généralement nécessaire de le maintenir suffisamment long pour des problèmes de stabilité en boucle fermée déjà évoqués. Dans ces conditions, le modèle local de prédiction peut introduire des approximations importantes pouvant engendrer des dysfonctionnements de la loi de commande.

Enfin, un dernier point à souligner est la caractéristique cumulative des erreurs de modélisation et donc de prédiction dans un problème de contrôle optimal. Ceci est évidemment dû à la récursivité au cœur de la prédiction. Les erreurs de modélisation s'additionnent et peuvent finir par produire une erreur conséquente en bout de la chaîne récursive.

Ainsi, la manipulation mobile présentant généralement de grands déplacements, l'étude de stabilité et la volonté d'une certaine modularité du contrôleur, ont conduit à choisir un modèle de prédiction global et à utiliser des caméras 3D qui permettent une initialisation correcte des informations de profondeur.

Modèle de prédiction de S_p^p

Grâce aux étapes de modélisation, les modèles de prédiction sont rapidement disponibles. Pour le vecteur basé sur les points et la projection perspective, il faut dans un premier temps retrouver les points d'intérêt dans le repère caméra de l'organe terminal $F_{ce}(p)$ avant de les convertir dans le repère de la base mobile $F_b(p)$ grâce au modèle géométrique du bras. Ensuite, le modèle d'intégration de la base mobile est utilisé pour prédire la position des points dans le repère de la base mobile de la prédiction suivante $F_b(p+1)$. Le chemin inverse est alors suivi : les informations sont renvoyées dans le repère de la caméra $F_{ce}(p+1)$ et finalement dans le plan image. Ces cinq étapes peuvent être traduites mathématiquement par l'équation suivante :

$$\bar{\mathbf{x}}_i(p+1) = {}^{i(p+1)}H_{c(p+1)} {}^{c(p+1)}H_{b(p+1)} {}^{b(p+1)}H_{b(p)} {}^{b(p)}H_{c_p} {}^{c(p)}H_{i(p)} \bar{\mathbf{x}}_i(p) \quad (3.3.5.1)$$

Pour rappel, $\bar{\mathbf{x}}_i$ désigne le point dans l'image en coordonnées homogènes et une matrice ${}^A H_B$ correspond à la matrice de transformation homogène d'un repère B à un repère A . En ne prenant en compte que les deux premières coordonnées x et y de cette équation, et en l'appliquant aux quatre points d'intérêt, le modèle de prédiction est complètement défini :

$$\hat{S}_p^p(p+1) = f_{S_p^p} \left(\hat{S}_p^p(p) \right) \quad (3.3.5.2)$$

Modèle de prédiction de S_p^m

La stratégie pour obtenir le modèle de prédiction du vecteur d'indice visuel basé sur les moments 2D est de prédire l'évolution des points grâce au modèle $f_{S_p^p}$ présenté ci-dessus, et de calculer les moments grâce aux formules du chapitre précédent. En définissant cette relation par une fonction g_p telle que $S_p^m = g_p(S_p^p)$, le modèle de prédiction est directement obtenu par :

$$\hat{S}_p^m(p+1) = g_p \circ f_{S_p^p} \left(\hat{S}_p^p(p) \right) \quad (3.3.5.3)$$

3.3.6 Définition des contraintes physiques

La gestion des contraintes a toujours été un axe central de recherche dans le domaine de l'asservissement visuel. Les contraintes de visibilité par exemple sont très souvent étudiées puisque le non-respect de celles-ci entraîne inévitablement un échec de la loi de commande. La prise en compte des contraintes physiques des limites des actionneurs et des articulations est également un point souvent adressé dans la littérature. Enfin, l'intégration de contraintes 3D telles que l'évitement d'obstacle peut aussi s'avérer très utile. Comme déjà expliqué, ces contraintes sont difficiles à intégrer dans des stratégies d'asservissement visuel classiques. Grâce aux moments dans l'image, il est par exemple possible d'influencer la trajectoire de la caméra [Mahony et al., 2002], et de respecter les contraintes de visibilité implicitement. Cependant, cet aspect interroge sur les performances de stabilité, qui sont encore plus délicates à assurer dans un contexte de manipulation mobile. L'asservissement visuel prédictif permet d'intégrer explicitement ces contraintes dans un formalisme de contrôle optimal.

Les sections suivantes définissent mathématiquement les contraintes intégrées au schéma de commande : visibilité, limites sur les butées articulaires, limites cinématiques des actionneurs.

Contraintes de visibilité

Les contraintes de visibilité sont certainement les contraintes qui mettent le mieux en avant l'intérêt d'intégrer l'asservissement visuel dans un schéma de contrôle

prédictif. Elles sont essentielles pour le bon fonctionnement du contrôleur et permettre de répondre d'une manière directe, générique et efficace aux problèmes de contraintes 2D dans l'espace image relatif au champ de vue des caméras, longuement étudié dans la littérature. En définissant les valeurs minimales S_p^l et maximales S_p^u des points d'intérêt dans l'image, celles-ci sont définies comme suit :

$$\begin{cases} \hat{S}_p^p(p) \leq S_p^u \\ S_p^l \leq \hat{S}_p^p(p) \end{cases} \quad \forall p \in \llbracket k+1, k+N_p \rrbracket \quad (3.3.6.1)$$

Contraintes sur les butées articulaires

Il est également nécessaire d'intégrer les butées articulaires dans les contraintes du système pour éviter le calcul de trajectoire impossible à réaliser. Ces contraintes nécessitent l'introduction des bornes minimales χ_{al} et maximales χ_{au} des angles des articulations du bras et sont alors définies par :

$$\begin{cases} \hat{\chi}_a(p) \leq \chi_{au} \\ \chi_{al} \leq \hat{\chi}_a(p) \end{cases} \quad \forall p \in \llbracket k+1, k+N_p \rrbracket \quad (3.3.6.2)$$

Contraintes sur les vitesses des actionneurs

De la même manière que pour les butées articulaires, il est nécessaire de prendre en compte les limites physiques des moteurs. Ces limites sont de deux types : cinématiques et dynamiques. Dans ce travail, les limites cinématiques sont définies de telle sorte qu'il n'est pas possible dépasser les accélérations et décélérations maximales des actionneurs. Les seules contraintes ajoutées au problème d'optimisation sont donc cinématiques. En définissant par $u_{mm l}$ et $u_{mm u}$ respectivement les vitesses minimales et maximales :

$$\begin{cases} u_{mm}(p) \leq u_{mm u} \\ u_{mm l} \leq u_{mm}(p) \end{cases} \quad \forall p \in \llbracket k, k+N_c-1 \rrbracket \quad (3.3.6.3)$$

Ces contraintes de vitesse sont importantes dans le contrôleur VPC puisqu'elles définissent la longueur physique maximale de la trajectoire prédite avec la période de prédiction T_s et le nombre de prédictions N_p . Elles sont donc aussi intimement liées à la capacité d'anticipation du système et une relaxation de ces dernières peut s'avérer très utile. Ce point est détaillé dans la section relative à l'analyse de la stabilité dans laquelle une légère modification de l'équation 3.3.6.3 est réalisée.

3.3.7 Contraintes de stabilité et de faisabilité

L'asservissement visuel par commande prédictive permet donc d'intégrer facilement de nombreuses contraintes comme celles définies ci-dessus. Cependant, cette approche souffre d'un inconvénient important : la difficulté d'assurer la stabilité du schéma. En effet, en raison de l'étape d'optimisation, la solution est numérique et

la démonstration de la stabilité est théoriquement difficile. Ce problème a donc été largement étudié dans la littérature et la théorie de la stabilité de Lyapunov établit des conditions pratiques pour assurer la stabilité asymptotique d'un schéma NMPC.

Théorème de stabilité de Lyapunov

Il convient tout d'abord de rappeler ce théorème fondamental [Sastry, 1999]. Soit un système dynamique discret défini par l'équation d'état suivante :

$$x(k+1) = f(x(k), u(k)) \quad (3.3.7.1)$$

où $x \in \mathbb{X}$ et $f : \mathbb{X} \rightarrow \mathbb{X}$ est un champ vectoriel continu. Si ce système a un point d'équilibre x_e , alors ce dernier est asymptotiquement stable en x_e s'il existe une fonction V de Lyapunov.

Une fonction $V : \mathbb{X} \rightarrow \mathbb{R}$ est une fonction de Lyapunov si :

$$\begin{cases} V(x) = 0 \iff x = x_e & (3.3.7.2a) \\ V(x) > 0 \iff x \neq x_e & (3.3.7.2b) \\ V(x(k+1)) - V(x(k)) < 0 & (3.3.7.2c) \end{cases}$$

Un candidat à cette fonction de Lyapunov alors considéré dans l'analyse de schéma NMPC est la fonction de valeur, correspondant au coût optimal J^* du problème d'optimisation 3.3.2.2. La fonction de valeur joue donc un rôle central dans l'analyse de la stabilité de la commande prédictive. Pour le problème considéré, cette fonction de valeur est alors définie par :

$$V(k) = J(k, U_k^*(\cdot)) \quad (3.3.7.3)$$

Les deux premières conditions de stabilité de Lyapunov 3.3.7.2a et 3.3.7.2b sont généralement assurées par une définition judicieuse du coût F . Le choix le plus populaire proposant ces propriétés est la forme quadratique et le coût défini précédemment respecte donc les deux premières conditions de stabilité. En effet :

$$\begin{cases} F_{vs}(S) = 0 \iff S = S^* & (3.3.7.4a) \\ F_{vs}(S) > 0 \iff S \neq S^* & (3.3.7.4b) \end{cases}$$

Malheureusement, la troisième condition 3.3.7.2c est plus compliquée à assurer et occupe ainsi la plupart des travaux portant sur la stabilité d'un schéma NMPC. La section suivante présente les principales solutions existantes permettant d'assurer que cette condition soit respectée, leurs avantages et inconvénients, et finalement la justification du choix effectué dans ces travaux de thèse.

Analyse théorique de la stabilité du schéma VPC

Les solutions trouvées dans la littérature pour des contrôleurs à horizon de prédiction fini peuvent être catégorisées dans trois classes [Grüne et al., 2017] présentées ci-dessous. Il convient de noter que ces dernières sont équivalentes d'un point de vue théorique. Seule la mise en pratique vis-à-vis de la problématique étudiée vient les distinguer. La compréhension de leurs points clefs permet alors de choisir la classe de méthode la plus adaptée. Dans un contexte de contrôle, il convient également d'introduire la notion de faisabilité. Dans un tel contexte, il est essentiel que le problème soit faisable : c'est-à-dire qu'une solution au problème d'optimisation existe bel et bien avec prise en compte des contraintes réelles. La faisabilité est intrinsèquement liée à l'étude de stabilité et une attention toute particulière est portée à ce sujet dans ces travaux.

1. Contrainte terminale

Les solutions basées sur la contrainte terminale permettent de se ramener à un horizon de prédiction infini en assurant que le dernier état prédit correspond à l'état de référence. Pour le cas VPC étudié, cette contrainte se traduit alors par :

$$\hat{S}(k + N_p) = S^* \quad (3.3.7.5)$$

En effet, sous les hypothèses d'optimalité de la solution et de modèle parfait, les prédictions d'une itération à l'autre peuvent être identifiées, c'est-à-dire avec $k' = k + 1$:

$$V(k + 1) = J(k + 1, U_{k+1}^*(\cdot)) = \sum_{p=k+2}^{k+N_p+1} F(p) = \sum_{p=k'+1}^{k'+N_p} F(p) \quad (3.3.7.6)$$

La contrainte terminale permet alors d'assurer la stricte décroissance de la fonction de valeur puisque la trajectoire à l'itération $k' = k + 1$ est théoriquement un morceau de la trajectoire à l'itération k [Colombo et al., 2019] :

$$V(k + 1) = \sum_{p=k+2}^{k+N_p+1} F(p) \quad (3.3.7.7)$$

$$= \underbrace{F(k + N_p + 1)}_{=F(S^*)=0} + \sum_{p=k+2}^{k+N_p} F(p) \quad (3.3.7.8)$$

$$= \sum_{p=k+1}^{k+N_p} F(p) - F(k + 1) \quad (3.3.7.9)$$

$$= V(k) - \underbrace{F(k + 1)}_{>0} \quad (3.3.7.10)$$

d'où

$$V(k+1) < V(k) \quad (3.3.7.11)$$

Cette contrainte permet également d'assurer que la faisabilité du problème à l'instant $k' = k + 1$ implique également la faisabilité de ce problème à l'instant k . À propos de ce point, il convient tout de même de noter qu'une autre hypothèse est réalisée ici. En effet, le problème doit être faisable et le système doit donc être capable d'atteindre l'optimum dès le début de la commande. Pour des horizons de prédiction court donc, des problèmes de faisabilité peuvent apparaître [Allgower et al., 2004]. Ce constat est d'autant plus problématique dans le contexte de la manipulation mobile étudiée dans ces travaux de thèse, puisqu'il paraît impossible d'assurer cette faisabilité en tenant compte des contraintes de puissance de calcul et physiques des actionneurs.

L'hypothèse d'optimalité réalisée peut également être remise en question puisqu'il paraît généralement impossible d'obtenir une solution globale d'un problème d'optimisation non-convexe et non-linéaire en un temps compatible avec la commande. Une interruption anticipée de l'optimisation peut alors provoquer des erreurs qui affectent la stabilité. De plus, le problème d'optimisation considéré dans un contexte de navigation réactive est variable dans le temps. L'optimisation est alors nécessairement définie dans un sens local avec les informations disponibles à une itération k donnée. Il en va de même pour l'hypothèse d'un modèle parfait et la théorie classique de la stabilité des NMPC est donc souvent inapplicable en pratique. Les travaux [Scokaert et al., 1999] proposent d'ajouter directement la condition 3.3.7.2c dans les contraintes du problème d'optimisation. L'auteur utilise et mentionne ainsi le fait que la faisabilité implique la stabilité sous les conditions mentionnées.

2. Coût terminal

La contrainte terminale est une manière intuitive et facile à mettre en œuvre de garantir la stabilité d'un schéma NMPC. Cependant, l'inconvénient majeur est que la satisfaction de la contrainte terminale peut s'avérer difficile en considérant un temps très limité ou des systèmes très complexes. Par ailleurs, un horizon de prédiction suffisamment long permettant d'atteindre le minimum global dès le début de la commande est nécessaire. De nombreuses méthodes tentent alors de réduire cette difficulté en relâchant la contrainte terminale à un ensemble \mathcal{S} plus grand contenant le point de référence et à définir un coût terminal E sur ce dernier proposant certaines propriétés. La fonction de coût devient alors :

$$J(k, U_k(\cdot)) = \sum_{p=k+1}^{k+N_p} F(p) + E(k + N_p) \quad (3.3.7.12)$$

En l'occurrence, la dernière prédiction et le coût terminal doivent respecter les conditions suivantes pour assurer la stabilité du schéma MPC [Grüne et al., 2017] :

$$\begin{cases} \hat{S}(k + N_p) \in \mathcal{S} & (3.3.7.13a) \\ E(k + N_p + 1) + F(k + N_p + 1) < E(k + N_p) & (3.3.7.13b) \end{cases}$$

Cette approche est souvent dénommée à horizon quasi-infini, car le coût terminal E consiste à réaliser une approximation de la fonction de valeur V pour un horizon infini. L'objectif est d'obtenir un ensemble \mathcal{S} autour de S^* le plus grand possible, pour que la contrainte terminale soit considérablement plus facile à assurer par la méthode d'optimisation. La difficulté réside donc en la définition de cet ensemble \mathcal{S} et du coût terminal E qui n'est pas toujours facile à établir.

La preuve de stabilité de cette méthode est similaire à celle correspondant à la contrainte terminale et nécessite les mêmes hypothèses sur l'optimalité de la solution et sur le modèle du système [Colombo et al., 2019] :

$$V(k+1) = \sum_{p=k+2}^{k+N_p+1} F(p) + E(k+N_p+1) \quad (3.3.7.14)$$

$$= F(k+N_p+1) + \sum_{p=k+2}^{k+N_p} F(p) + E(k+N_p+1) \quad (3.3.7.15)$$

$$= V(k) + F(k+N_p+1) + E(k+N_p+1) \quad (3.3.7.16)$$

$$= V(k) + \underbrace{F(k+1)}_{>0} + \underbrace{E(k+N_p+1) + F(k+N_p+1) - E(k+N_p)}_{>0} \quad (3.3.7.17)$$

d'où

$$V(k+1) < V(k) \quad (3.3.7.18)$$

3. Sans contrainte terminale

Il est également possible d'assurer la stabilité théorique d'un schéma NMPC sans ajouter de contrainte terminale, relâchée ou non. Cependant, cette méthode nécessite de fortes hypothèses sur le coût F et sa capacité de traduction de la distance à la situation de référence. En fait, cette méthode suppose que le seul coût F quantifie suffisamment bien l'erreur en tout point et en tout instant relativement à l'horizon de prédiction considéré. Les preuves théoriques de stabilité correspondant à cette méthode peuvent être trouvées dans [Grüne et al., 2017]. Ainsi, aucun coût terminal F ne doit être défini et aucune contrainte supplémentaire ne doit être ajoutée au problème d'optimisation. Les conditions théoriques de stabilité supposent cependant une connaissance parfaite a priori du problème pour établir un coût F adéquat. Si l'effort n'est plus à faire en ligne, il se situe plutôt lors de la phase hors ligne d'établissement de ce coût F qui doit alors être à l'épreuve des non-convexités ou des changements de problème d'optimisation. Pour de tels problèmes, il peut s'avérer très compliqué, voire impossible de définir un F qui garantit la stabilité. Ces hypothèses ne coïncident pas avec la problématique de commande réactive considérée dans ces travaux de thèse.

[Grüne et al., 2017] mentionne également que la stabilité peut être perdue si l'horizon de prédiction n'est pas suffisamment grand par rapport à la tâche. En effet, le contrôleur peut alors par exemple se perdre dans des minima locaux. C'est

finalement sur ce point de garantie de la faisabilité que se distingue cette méthode par rapport aux deux autres. Les schémas avec contrainte terminale ont l'avantage de garantir la faisabilité. Ainsi, ces derniers permettent généralement de détecter plus tôt un problème infaisable et de guider le système vers la situation de référence. Cette caractéristique est très intéressante, compte tenu du caractère réactif de la problématique étudiée ici puisque le schéma de contrôle s'adapte à chaque nouvelle itération à la situation actuelle.

Synthèse

Ajouter une contrainte terminale est donc nécessaire compte tenu de la problématique étudiée, où l'environnement de travail du robot n'est pas connu a priori. De plus, la capacité de cette contrainte terminale à guider le robot vers la situation de référence est très intéressante pour aider le solveur à trouver une trajectoire optimale en un temps limité ou à sortir des minima locaux. En effet, il faut encore se rappeler que malgré l'ensemble de ces résultats théoriques, la solution renvoyée par la méthode d'optimisation est généralement sous-optimale. Cet aspect est de plus accentué par les fortes non-linéarités du système considéré.

Cependant, un point important concernant les méthodes à contrainte terminale n'a pas encore été soulevé : celui de la nécessité de la commandabilité du système à la référence en un temps inférieur à l'horizon de prédiction. En fait, la tâche doit être faisable en un temps défini par la taille de l'horizon de prédiction. Mais la taille de ce dernier peut être augmentée astucieusement en relâchant par exemple des contraintes physiques des actionneurs judicieusement choisies [Durand-Petiteville and Cadenat, 2022]. Aussi, utiliser cette stratégie pour une méthode sans contrainte terminale peut s'avérer compliqué à mettre à œuvre, car cette dernière permet de guider le solveur dans l'espace de recherche complexe considéré. Les méthodes sans contrainte terminale sont donc écartées dans ces travaux de thèse.

Ensuite, la méthode consistant à relâcher la contrainte terminale présente plusieurs points de tension par rapport à la problématique. Ils sont tous globalement liés à l'aspect réactif du contrôleur considéré. Premièrement, définir a priori un ensemble relâché \mathcal{S} autour de S^* suffisamment grand pour que le gain de temps d'optimisation soit notable peut s'avérer compliqué sans une connaissance de l'environnement autour de la situation de référence. Un environnement dynamique compromet également une définition rigoureuse hors-ligne du coût terminal E respectant en tout point la condition 3.3.7.13b. De plus, la construction de ces paramètres additionnels \mathcal{S} et E est difficile pour des systèmes hautement non-linéaires et à référence dynamique. Il devient alors compliqué d'assurer une généricité et une modularité du contrôleur lorsque la contrainte terminale est relâchée.

Compte tenu de ces arguments, la méthode choisie pour garantir la stabilité du schéma VPC au cœur de ces travaux de thèse est la méthode utilisant une contrainte terminale.

Contrainte terminale

La contrainte terminale doit donc être ajoutée au problème d'optimisation. L'équation correspondante pour le schéma VPC a été donnée en équation 3.3.7.5. Cependant, une contrainte d'égalité stricte est souvent difficile à satisfaire pour les problèmes non-linéaires et non-convexes. L'algorithme de résolution est généralement incapable de trouver une solution réalisable même si une telle solution existe. La contrainte est alors convertie en inégalité en définissant un seuil δ_{TC} :

$$\hat{S}(k + N_p) - S^* < \delta_{TC} \quad (3.3.7.19)$$

Cette légère relaxation n'est absolument pas similaire à la méthode de garantie de stabilité par coût terminal décrite précédemment. Le seuil δ_{TC} est suffisamment petit pour qu'aucun problème de stabilité n'apparaisse. Aucun coût terminal E n'est donc nécessaire pour guider le système vers la situation de référence. Ce seuil est seulement introduit pour éliminer les problèmes numériques engendrés par l'utilisation d'une égalité stricte.

Contraintes des vitesses des actionneurs relâchées

Les contraintes visant à prendre en compte les limites physiques des actionneurs ont été définies précédemment. Avec le nombre d'étapes de prédiction, elles définissent la longueur maximale de la trajectoire prédite. Cependant, cette dernière doit être suffisamment grande pour que le problème soit faisable. Ceci peut représenter un défi dans un contexte de manipulation mobile en raison des grandes distances que la caméra doit parcourir pour atteindre l'objectif. Le nombre de prédictions est plafonné par la puissance de calcul maximale et ne peut donc pas s'adapter à la généralité du contrôleur recherché. Ainsi, pour permettre de garantir la contrainte terminale, il est proposé de contraindre les vitesses de la première partie des commandes en fonction des limites physiques pour effectivement tenir compte de ces limites, et de relâcher les contraintes pour la deuxième partie des commandes. Cette approche conduit à la modification des contraintes de vitesse 3.3.6.3 proposées précédemment :

$$\begin{cases} u_{mm}(p) \leq u_{mm\ u|t} & \forall p \in \llbracket k, k + N_c - N_r - 1 \rrbracket \\ u_{mm\ l|t} \leq u_{mm}(p) & \end{cases} \quad (3.3.7.20)$$

$$\begin{cases} u_{mm}(p) \leq u_{mm\ u|r} & \forall p \in \llbracket k + N_c - N_r, k + N_c - 1 \rrbracket \\ u_{mm\ l|r} \leq u_{mm}(p) & \end{cases}$$

N_r est le nombre de prédictions avec des limites relâchées, $u_{mm\ l|t}$ et $u_{mm\ u|t}$ sont respectivement les limites inférieure et supérieure strictes correspondant aux limites réelles des actionneurs, et $u_{mm\ l|r}$ et $u_{mm\ u|r}$ sont respectivement les limites inférieure et supérieure relâchées. En outre, le nombre de prédictions relâchées N_r est faible devant l'horizon de commande N_c de sorte que la commande appliquée au robot

respecte les limites des actionneurs, tout en permettant à l'horizon de prédiction d'être suffisamment grand pour garantir la contrainte terminale.

3.3.8 Résolution du problème d'optimisation

Méthodes directes

Comme expliqué précédemment, les méthodes directes convertissent le problème de contrôle optimal en temps continu, de dimension infinie, en un problème d'optimisation non-linéaire de dimension fini en discrétisant la commande du système. Après cette discrétisation, il existe plusieurs manières de résoudre le problème : les trois méthodes principales [Diehl and Gros, 2011] sont dénommées *direct single shooting*, *direct multiple shooting* et *direct collocation*.

La méthode *direct single shooting* est définie comme une approche séquentielle : c'est-à-dire que les variables d'état sont considérées comme dépendantes l'une de l'autre. L'intégration de la dynamique du système est réalisée sur l'horizon de prédiction en entier et le résultat est donné à un solveur d'optimisation non-linéaire. Dans cette méthode, seules les commandes sont alors des variables d'optimisation et le problème est résolu d'une manière dense, c'est-à-dire sans utiliser concrètement une certaine structure du problème de contrôle optimal. Souvent, c'est un solveur SQP (Sequential Quadratic Programming) qui est utilisé pour résoudre le problème. Un inconvénient majeur de cette méthode vient du fait que l'approche séquentielle propage les non-linéarités, augmentant la difficulté de résolution. C'est sur ce point que viennent appuyer les deux méthodes suivantes.

La méthode *direct multiple shooting* résout le problème avec une approche simultanée : c'est-à-dire que les morceaux de la trajectoire discrétisée sont traités de manière indépendante. L'intégration de la dynamique est uniquement réalisée sur chaque prédiction et la continuité de la trajectoire est assurée par des contraintes supplémentaires ajoutées au problème. L'idée sous-jacente de cette approche est de conserver la structure du problème de contrôle optimal : les matrices jacobiniennes et hessiennes sont alors structurées par bloc et cette propriété peut être exploitée dans la résolution. Dans cette méthode, il y a donc davantage de variables d'optimisation puisque les états le long de la trajectoire viennent s'ajouter aux commandes, mais la structure du problème est mieux utilisée dans la résolution. Pour résoudre le problème, il faut alors utiliser des outils pouvant gérer cette structure, associés généralement à un solveur basé sur la méthode de points intérieurs (Interior-Point Method ou IPM). Les bibliothèques *CasADi* et *Ipopt* proposent par exemple des outils permettant d'appliquer cette méthode.

Enfin, la méthode *direct collocation* étend cette idée d'utiliser la structure du problème de contrôle optimal. Contrairement au *multiple shooting*, la collocation n'inclut pas d'intégration de la dynamique du système et laisse ce travail au solveur. Ainsi, la dimension du problème augmente encore afin de maximiser l'efficacité d'une résolution exploitant sa structure. Cependant, cette méthode nécessite de connaître la dynamique de chaque variable introduite dans le problème. Il serait donc indispensable de reformuler les étapes de modélisation précédente pour pouvoir l'appliquer.

La dynamique des indices visuels basés sur les moments complexifie l'implémentation de cette méthode. Bien que possible puisque les matrices d'interaction relatives aux moments sont connues, cette étape n'a pas été réalisée dans ces travaux et la méthode de *direct collocation* n'a pas été testée.

Méthode de calcul des gradients

Premièrement, l'utilisation de solveur basé sur les gradients plutôt que sur ceux n'utilisant pas ces informations est quasi systématique dans la littérature. En effet, il est rare que l'utilisation d'un solveur *gradient-free* soit bénéfique et il est souvent possible que les performances ne soient pas suffisantes. En l'occurrence, pour le problème d'optimisation présenté, le solveur *DIRECT* de la librairie *NLopt* ne permet pas de trouver une solution satisfaisante avec les contraintes de temps-réel de la problématique.

Les gradients sont donc nécessaires dans l'optimisation. Il existe de nombreuses manières de calculer les dérivées : il est bien sûr théoriquement possible de le réaliser de manière analytique, mais ceci est quasiment impossible pour des fonctions encapsulant récursivement des équations complexes comme ceux au cœur de ces travaux.

D'un autre côté, la méthode des différences finies peut toujours être appliquée. Elles sont faciles à implémenter, mais le calcul du gradient d'une fonction à n variables nécessite $n + 1$ appels à cette fonction. À nouveau, pour les fonctions et codes au cœur de ces travaux, ces appels peuvent coûter très cher en temps de calcul.

La méthode des différences finies a d'abord été utilisée dans ces travaux de thèse en raison de sa rapidité d'implémentation. Cependant, le temps de calcul des gradients représentait une grande partie du temps d'optimisation ($> 80\%$ pour le problème d'optimisation présenté avec $N_p = N_c = 10$). Ce constat est particulièrement important puisque cette perte de temps liée au calcul du gradient se répercute inévitablement sur la qualité de la solution déterminée par le solveur dans le temps imparti. Il est même montré par la suite au travers de différents résultats que cette perte de temps peut s'avérer fatale pour le schéma VPC lorsque le problème d'optimisation se complexifie. Le choix de se tourner vers la différenciation algorithmique (ou différenciation automatique, ou AD) a été fait. L'idée de la différenciation algorithmique est d'utiliser les règles de dérivation en chaîne et de se ramener à la dérivation des opérations élémentaires. Des outils de différenciation algorithmique d'un code sont disponibles, comme *CppAD* en C++ par exemple. Mais dans le but d'utiliser des solveurs basés sur la structure des gradients et matrices hessiennes comme expliqué dans la section suivante, l'outil *CasADi* a été privilégié. Le calcul des gradients est alors réalisé d'une manière symbolique, pour être ensuite évalué. Ainsi, le principal avantage d'utiliser la différenciation algorithmique pour résoudre le problème d'optimisation n'est pas nécessairement lié à la rapidité de calcul du gradient, mais bien au fait qu'il peut être réalisé de manière hors-ligne. En effet, le gradient n'est alors qu'évalué lors de la commande du robot et cette méthode de calcul des gradients permet ainsi une nette amélioration des performances du schéma de contrôle. Cet aspect est illustré dans les résultats.

Choix final de méthode de résolution

Plusieurs méthodes de résolution ont été testées au cours de la thèse, associant différents solveurs intégrés à *CasADi* ou à *NLopt* avec la méthode de *single shooting* et le solveur *Ipopt* avec la méthode *multiple shooting*. Deux d’entre elles se sont démarquées dans ces travaux de thèse :

- *Direct single shooting* avec le solveur *SLSQP* de *NLopt*
- *Direct multiple shooting* avec *CasADi* et le solveur *Ipopt*

Malheureusement, seule la première combinaison permet une résolution respectant les contraintes de temps de calcul. C’est donc cette méthode qui est utilisée dans la suite de la thèse et pour l’ensemble des résultats présentés.¹

3.4 Résultats

La présentation des résultats correspondant à ce premier schéma VPC est divisée en trois parties qui incrémentent progressivement le réalisme des essais. Après une première partie introductive permettant d’introduire l’environnement de travail et d’illustrer la tâche considérée, un simulateur personnel purement théorique est utilisé afin de confirmer les choix réalisés. Ensuite, le schéma est testé dans l’environnement *Gazebo* pour obtenir des scénarios plus réalistes. Enfin, la stratégie de commande est implémentée sur une plateforme TIAGo réelle, afin de démontrer que cette stratégie s’intègre correctement sur un robot physique dans un environnement réel.

3.4.1 Introduction à la tâche étudiée et implémentation du contrôleur

Environnement de travail

Tous les algorithmes sont implémentés en utilisant le langage C++. La résolution du problème de contrôle optimal est donc abordée avec la méthode directe *multiple-shooting* et le solveur SLSQP du package NLopt [Johnson, 2007]. Les matrices relatives à la cinématique du système au cœur du modèle de prédiction sont obtenues avec Pinocchio [Carpentier et al., 2021] basé sur le modèle URDF du robot. Les gradients sont calculés symboliquement hors ligne, et seulement évalués lors du contrôle à l’aide de CasADi [Andersson et al., 2019]. Tous les tests sont effectués sur un processeur Intel Core i7-10850H. Le temps maximal de résolution est fixé à 150ms afin de permettre à la boucle de contrôle de fonctionner à une fréquence de 5 Hz. La période de prédiction est fixée à $T_s = 0.4s$ pour obtenir un horizon de prédiction suffisamment long. Enfin, les limites de la vitesse linéaire de la base mobile sont égales à $\pm 0.1m/s$, tandis que les vitesses angulaires sont fixées à

1. Il est possible que l’utilisation de la structure (sparsity) du problème n’ait pas été utilisée de manière optimale pour la méthode de *multiple shooting* et que ce choix pourrait être revu.

$\pm 0.3 \text{ rad/s}$. Les limites minimales et maximales des articulations sont données par : $\chi_{au} = [2.68, 1.02, 1.50, 2.29, 2.07]$ et $\chi_{al} = [0.07, -1.50, -3.46, -0.32, -2.07]$.

La cible est un rectangle centré en $(3, 0, 1.08625)$ et la pose initiale de la base mobile est $(0, 0, 0)$ dans F_0 comme indiqué sur la figure 4.22a. L'objectif est donc de positionner le repère de la caméra de l'organe terminal dans une situation de référence relativement à cette cible. Cette situation de référence est définie dans l'image, tout comme le coût relatif à la tâche visuelle. La caméra et la base mobile doivent alors parcourir environ $2m$ pour atteindre la cible.

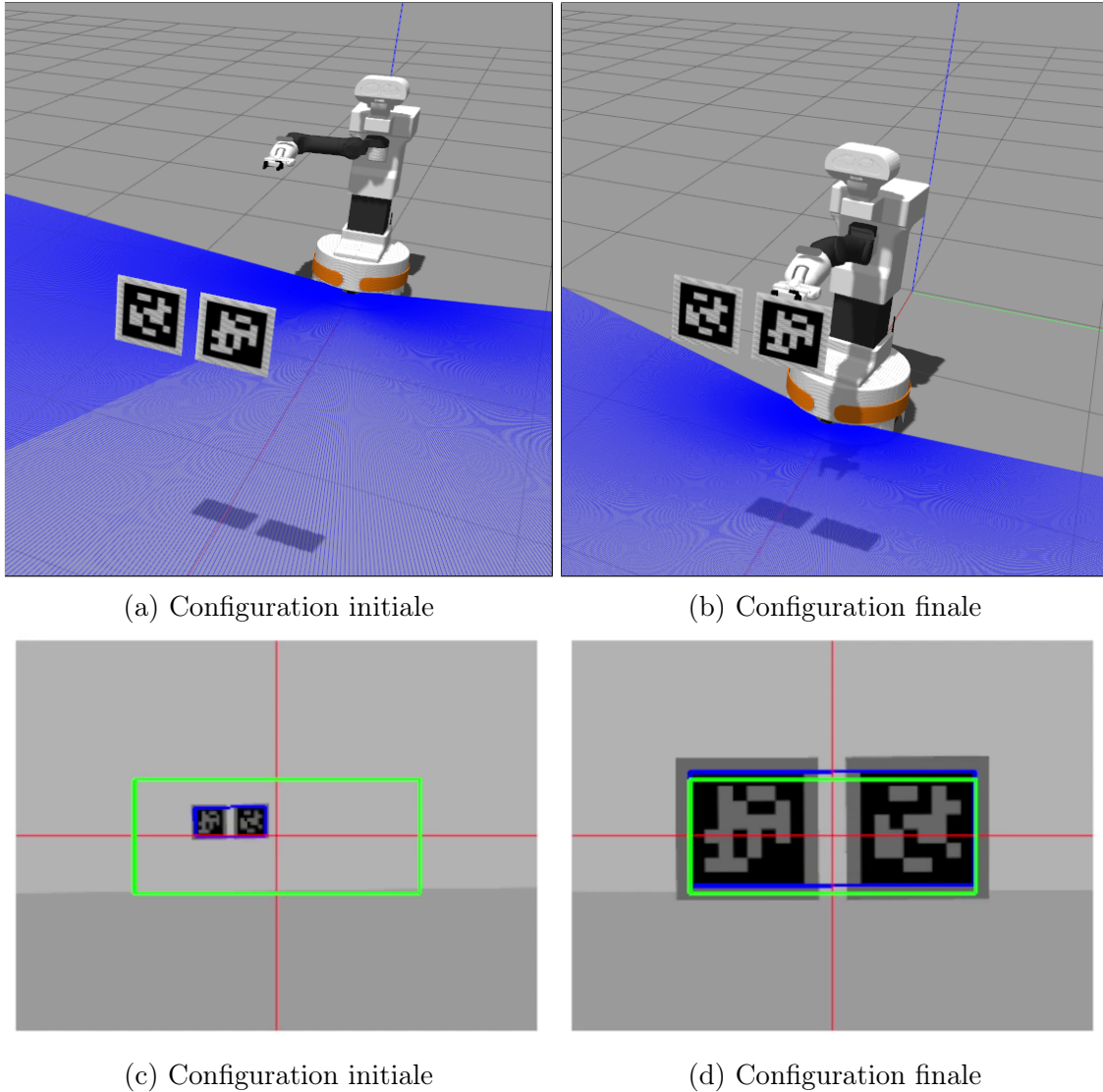


FIGURE 3.5 – Environnement *Gazebo*

Première analyse de résultats dans *Gazebo*

La figure 3.6 présente des exemples de captures d'écran de la trajectoire du robot et des images de la caméra de l'organe terminal. Elle permet d'analyser qualitative-

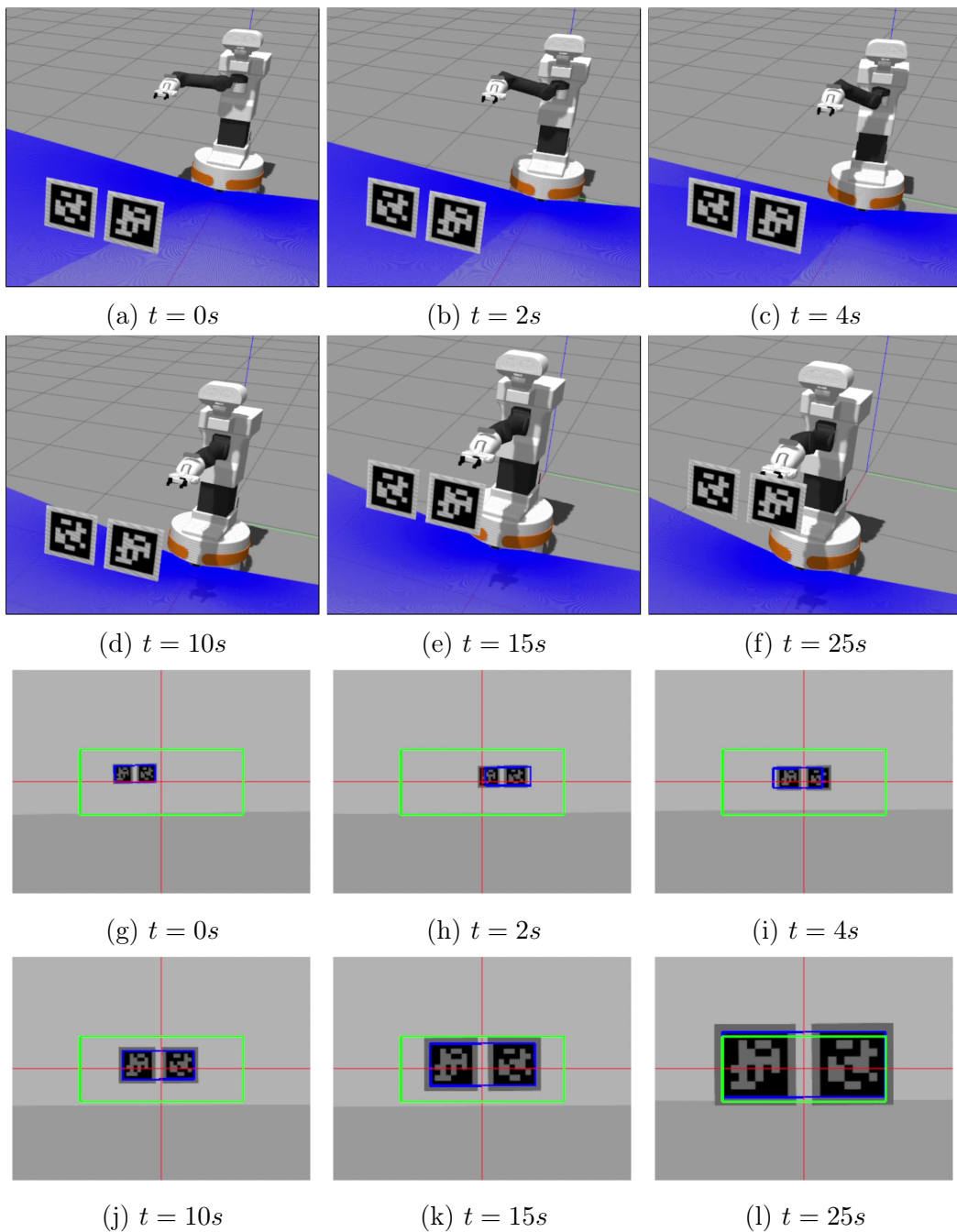


FIGURE 3.6 – Trajectoires du robot et des images obtenues

ment les résultats obtenus et la tâche d'asservissement visuel semble correctement réalisée. Seule une légère erreur finale apparaît, mais celle-ci est justifiée par la suite et n'est pas observée sur les résultats expérimentaux. Une analyse plus approfondie des trajectoires obtenues est réalisée dans la section relative aux essais expérimentaux, mais cette première évaluation souligne l'adéquation et la performance de l'approche proposée.

3.4.2 Comparaison des performances relatives au choix des indices visuels : S_p^p vs S_p^m

Cette partie a pour objectif de confirmer les propositions théoriques réalisées précédemment sur le choix des indices visuels utilisés au cœur de la fonction de coût du schéma de contrôle. En effet, il a été annoncé que la forme de l'espace de recherche a une influence sur la résolution du problème d'optimisation et que l'utilisation des moments permet de réduire la charge de calcul et d'aider le solveur à trouver une meilleure solution dans les conditions de sous-optimalité considérées.

Performances en termes d'erreur sur le positionnement final

Afin d'éliminer toute perturbation extérieure au schéma de contrôle, cette partie considère un simulateur parfait. Ce simulateur intègre uniquement les équations des modèles de prédiction présentés précédemment. La commande est donc parfaitement exécutée et aucune erreur de modélisation n'est alors possible. Cet environnement de travail permet ainsi d'isoler l'étape d'optimisation et d'observer l'influence des choix théoriques au sein de celle-ci. Il offre la possibilité d'analyser les différences de performance relatives au choix des indices visuels (S_p^p ou S_p^m), mais également d'observer l'impact de la méthode de calcul de gradients.

Cet ensemble initial de simulations vise à démontrer l'avantage de s'appuyer sur des caractéristiques visuelles basées sur les moments plutôt que sur des points pour améliorer les performances de positionnement final. En effet, comme expliqué précédemment, l'utilisation des coordonnées des points d'intérêt rend difficile la régulation des erreurs d'orientation x_c ou y_c . Pour comparer les deux approches, il est proposé d'analyser les erreurs pour chaque degré de liberté de la tâche. Pour cela, un coût à six dimensions évaluant la différence entre le repère résultant de la commande et le repère de référence est utilisé. Dans ce travail, le score ν_H est utilisé et correspond au vecteur de vitesse $[v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]^T$ qui devrait être appliquée pendant 1 seconde pour passer du repère de référence H^* au repère défini par H . Ce score est exprimé comme suit :

$$\nu_H = \log_6(H^{-1}H^*) \quad (3.4.2.1)$$

où H est le repère final obtenu avec le contrôleur VPC, H^* est le repère de référence, et \log_6 est la fonction logarithme qui permet de convertir un élément H du groupe de Lie $SE(3)$ à l'élément correspondant ν_H de son algèbre de Lie $se(3)$ [Sola et al., 2018].

Pour ce premier test, $N_p = N_c = 20$, $N_r = 0$, $Q_S = I_6$ et la contrainte terminale n'est pas incluse pour obtenir des résultats permettant de ne comparer que le choix des caractéristiques visuelles sur le critère de performance du positionnement final.

Comme le montrent les Figures 3.7a et 3.7b, le contrôleur VPC parvient à guider la caméra pour faire converger les indices visuels vers les valeurs désirées (croix vertes sur la figure) pour les deux scénarios. Cependant, la trajectoire obtenue dans l'espace image est significativement plus courte pour les caractéristiques visuelles ponctuelles

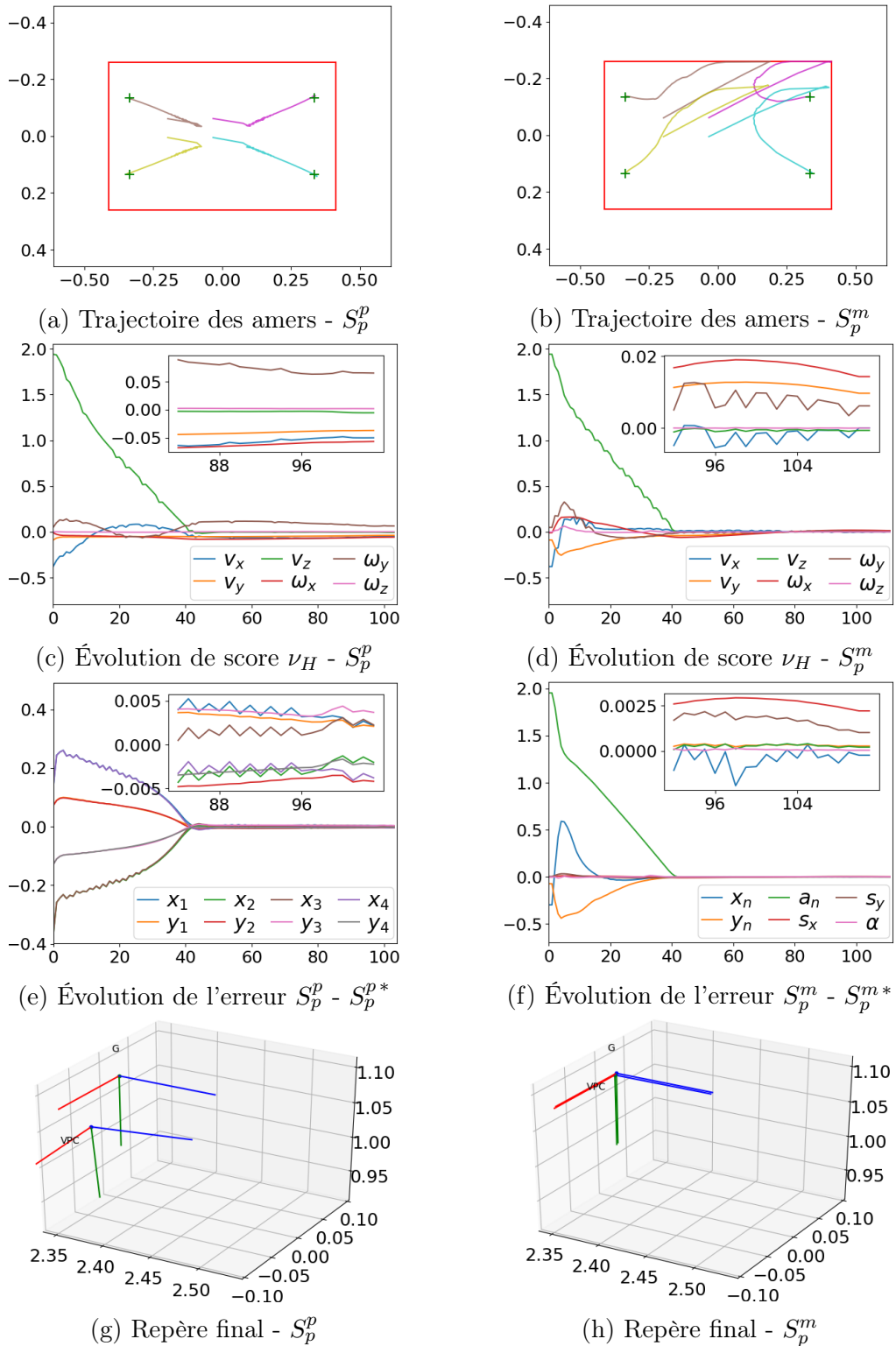


FIGURE 3.7 – Comparaison des performances relatives à la tâche visuelle pour un schéma basé sur S_p^p à gauche et S_p^m à droite

que pour les moments de l'image. Pour ce deuxième cas, la contrainte de visibilité empêche les caractéristiques visuelles de sortir du champ de vision de la caméra. En ce qui concerne la précision de la pose finale, sur les figures 3.7g et 3.7h montrent que l'utilisation des moments d'image permet d'obtenir un positionnement nettement plus précis. Cela est confirmé par les résultats présentés sur les Figures 3.7c et 3.7d, où le vecteur ν_H correspondant à S_m est significativement plus petit que celui de S_p^p , montrant l'intérêt d'utiliser les moments. Après convergence avec S_p^p , $\|\nu_H\| \approx 11e-2$, tandis qu'avec S_m , $\|\nu_H\| \approx 1e-1$. La Figure 3.7c met en évidence que la difficulté provient essentiellement de ω_x et ω_y . De plus, l'évolution des moments d'image (Figure 3.7f) suit la même tendance que ν_H (Figure 3.7d), tandis que celle des points (Figure 3.7e) est nettement différente (Figure 3.7c). Ainsi, il est facile d'ajuster la matrice de pondération Q_S pour prioriser n'importe quelle direction en utilisant les moments d'image comme caractéristiques visuelles.

L'analyse réalisée précédemment peut être généralisée à différents horizons de prédiction N_p . La figure 3.8 présente la tendance générale de $\|\nu_H\|$ en fonction de N_p . Cette dernière montre que le gain de positionnement apporté par l'utilisation des moments de l'image est globalement conservé. Il est important de noter que ces observations sont réalisées avec un temps de résolution maximal limité à $150ms$ pour respecter les contraintes imposées par la fréquence du contrôleur ($< 200ms$). Cette étude met en évidence l'importance des propriétés de découplage apportées par les moments. Ces dernières permettent d'obtenir une meilleure forme de l'espace de recherche, et ainsi d'obtenir une résolution plus rapide. En effet, d'après la figure 3.8, le schéma basé sur l'utilisation des moments avec un horizon de prédiction de longueur $N_p = 26$ permet d'obtenir des performances de positionnement final similaires au schéma basé sur les points avec un horizon de prédiction de longueur $N_p = 10$. Les performances sont donc similaires alors que le problème d'optimisation contient 2.6 fois plus de variables.

Cette conclusion est fondamentale dans ces travaux de thèse puisque cette propriété se répercute sur la résolution d'un problème de contrôle optimal plus complexe, notamment sur celui intégrant la contrainte terminale (ce point est illustré par la suite). La définition de la tâche visuelle est donc d'une importance capitale dans la conception du schéma de contrôle visuel prédictif.

Influence sur la trajectoire cartésienne

La section précédente a montré la corrélation entre les indices visuels construits à partir des moments de l'image et le vecteur vitesse ν_H représentant le mouvement de la caméra dans le domaine cartésien. Cette propriété permet d'influencer la trajectoire de la caméra. Cette section présente des résultats de simulation dans l'environnement *Gazebo* permettant de témoigner de ce constat.

Le moyen le plus simple d'observer ce phénomène est certainement de pondérer l'indice visuel a_n afin de limiter son importance dans la fonction de coût relativement aux autres indices. C'est pourquoi cette analyse étudie les deux scénarios suivants : $N_p = N_c = 10$, $N_r = 0$ avec deux matrices de pondération $Q_{S1} = I_6$ et $Q_{S2} = \text{diag}(1, 1, 0.1, 1, 1, 1)$. Comme précédemment, la contrainte terminale n'est

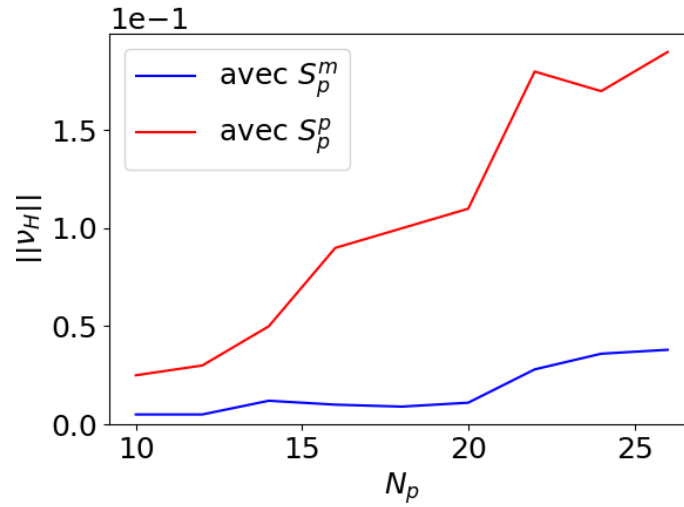


FIGURE 3.8 – Comparaison des performances de positionnement final pour différentes longueurs d’horizon de prédiction avec un temps de résolution maximal fixé à $150ms$

pas incluse.

Les figures 3.9a et 3.9b présentent les configurations obtenues au début de la trajectoire respectivement pour Q_{S1} et Q_{S2} . Celles-ci mettent en avant l’impact de la priorisation de l’avance du robot (translation en z) pour le premier cas qui entraîne une mauvaise orientation de la caméra relativement à la cible. Ce constat est illustré par les trajectoires des amers dans l’image présentées sur les figures 3.9c et 3.9d. Grâce à une pondération adéquate Q_{S2} qui privilégie une meilleure orientation de la caméra, une trajectoire dans l’image plus sûre est obtenue. Cette observation fait écho aux stratégies de prise ne compte des contraintes de visibilité dans un schéma d’asservissement visuel classique de manière implicite [Allibert et al., 2010b].

En effet, le modèle utilisé n’étant que cinématique, certaines prédictions impliquant de fortes accélérations peuvent présenter une erreur. De plus, le retard engendré par la résolution du problème d’optimisation au cœur de la boucle de commande est, lui aussi, à l’origine d’erreurs de prédiction. Sur la figure 3.9c par exemple, cet aspect est observable sur la contrainte de visibilité. Pour remédier à ce problème, cette contrainte a été configurée de manière conservatrice, permettant aux amers visuels de ne pas quitter le champ de vision de la caméra. Une trajectoire dans l’image plus adaptée, rendue possible grâce à une judicieuse pondération de la fonction de coût, associé à cette modification de contrainte, permet de s’assurer que tout problème relatif à la visibilité est évité (figure 3.9c).

Remarque. Une légère erreur de positionnement final est observable sur les figures 3.9c et 3.9d. Cette dernière n’est pas liée au schéma de contrôle VPC mais à un problème de commande bas-niveau du bras du robot en simulation lorsque celui-ci est tendu. Ce problème n’apparaît que pour de très faibles commandes, comme celles calculées aux instants proches de la convergence. Ce phénomène n’étant pas présent sur les résultats expérimentaux, nous n’avons pas insisté plus que de raison pour le

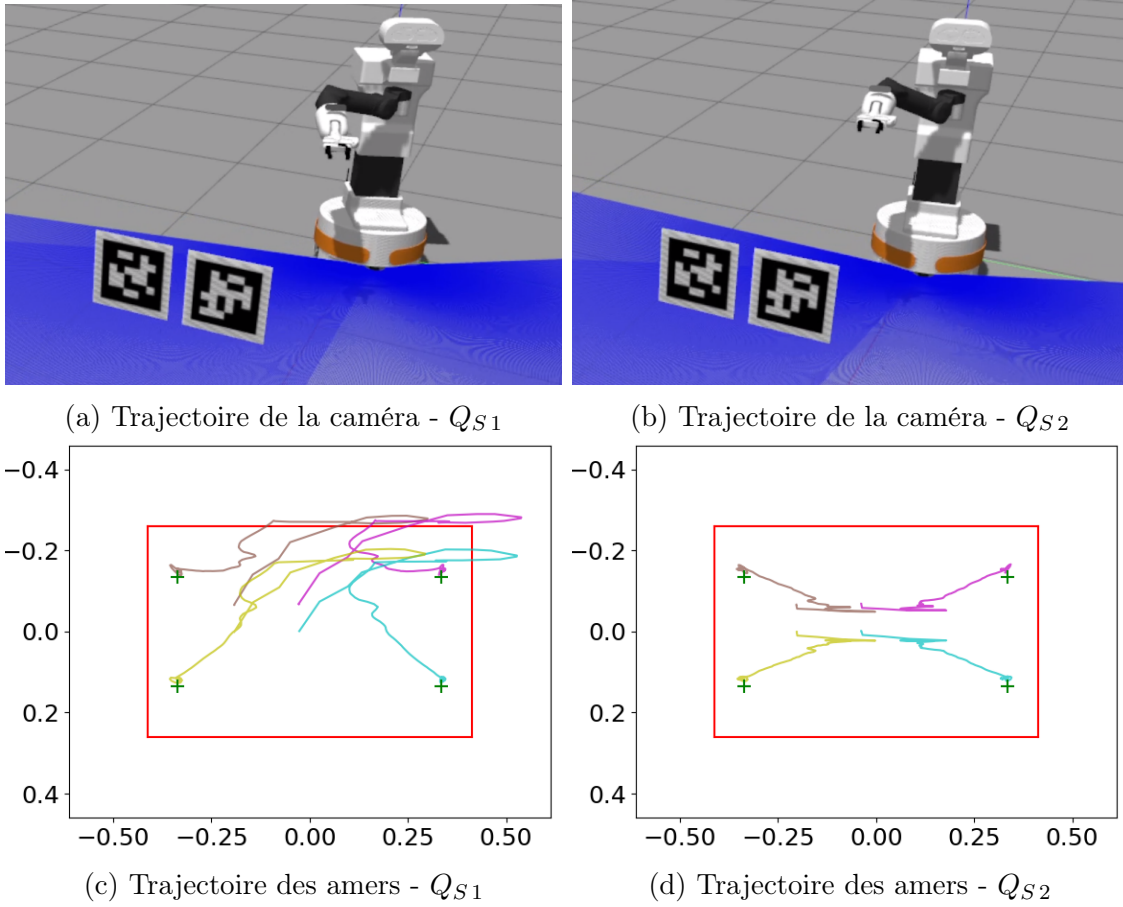


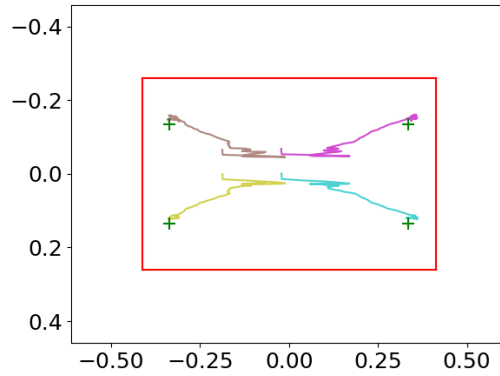
FIGURE 3.9 – Comparaison des trajectoires obtenues avec $Q_{S1} = I_6$ à gauche et $Q_{S2} = \text{diag}(1, 1, 0.1, 1, 1, 1)$ à droite

résoudre.

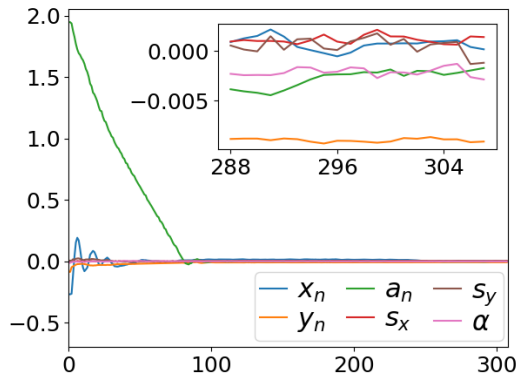
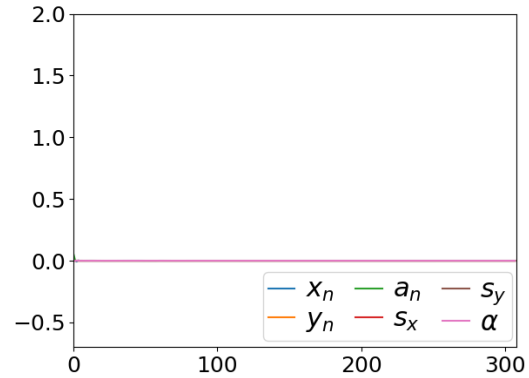
3.4.3 Résultats du schéma complet en simulation

Dans cette section, la contrainte terminale est incluse et le schéma permet donc d'assurer la stabilité en boucle fermée. Ensuite, les paramètres de prédiction sont configurés à $N_c = 10$ et $N_r = 1$ pour obtenir une trajectoire prédite suffisamment longue pour garantir la faisabilité du problème dès le début de la trajectoire. Enfin, la matrice de pondération est choisie d'après les résultats précédents $Q_S = \text{diag}(1, 1, 0.1, 1, 1, 1)$.

Comme précédemment, les figures 3.10a et 3.10b montrent que la loi de commande pilote correctement la caméra et la tâche visuelle est réalisée. Enfin, la figure 3.10c met en avant le respect de la contrainte terminale pendant toute la trajectoire, malgré la grande distance initiale du robot à la cible. Trois raisons à cela : (i) l'utilisation des moments, (ii) l'implémentation de la résolution du problème d'optimisation performante grâce au calcul symbolique des gradients, et (iii) la relaxation de la dernière séquence de commande. Ce triptyque constitue une contribution de



(a) Trajectoire des points d'intérêt


 (b) Évolution de l'erreur $S_p^m - S_p^{m*}$


(c) Évolution de l'erreur relative à la contrainte terminale

FIGURE 3.10 – Résultats du schéma complet

ces travaux de thèse : ils permettent d'adapter la notion de contrainte terminale au contexte de manipulation mobile mettant en jeu de grands déplacements. La relaxation alors mise en place complique l'optimisation et l'utilisation des moments accompagnée du calcul symbolique des gradients sont nécessaires à la résolution.

Remarque. *Le problème relatif aux très faibles commandes apparaît à nouveau et produit l'erreur de positionnement final*

3.4.4 Intérêt du calcul de gradient symbolique

L'objectif de cette partie est de comparer les performances du schéma de contrôle basé sur un calcul de gradient symbolique relativement au même schéma s'appuyant sur un calcul de gradient par différences finies. À nouveau, cette partie est réalisée dans un simulateur parfait pour éviter toute perturbation extérieure au schéma de contrôle. Pour ces tests, le schéma est paramétré de la même manière que la section précédente : $N_p = N_c = 10$, $N_r = 1$, $Q_S = \text{diag}(1, 1, 0.1, 1, 1, 1)$ et avec contrainte terminale.

Deux points sont comparés : les performances relatives à la tâche visuelle, ainsi que le respect de la contrainte terminale.

Performances de la tâche visuelle

Les figures 3.11a et 3.11b mettent une nouvelle fois en évidence la capacité du schéma de contrôle à amener les points d'intérêt à leur position de référence dans l'image. Cependant, la trajectoire du schéma basé sur les différences finies est nettement plus perturbé. De plus, les figures 3.11c, 3.11d, 3.11e et 3.11f qui présentent les évolutions du vecteur ν_H et de l'erreur $S_p^m - S_p^{m*}$ montrent que le positionnement final est plus précis grâce à l'utilisation du calcul symbolique du gradient. En effet, malgré la définition de la tâche visuelle à l'aide des moments dans l'image, la résolution du problème d'optimisation n'est pas assez rapide avec les différences finies et la solution renvoyée par le solveur est sous-optimale.

Performances relatives à la contrainte terminale

Concernant le respect de la contrainte terminale, les figures 3.12a et 3.12b révèlent que les enjeux liés à la sous-optimalité de la résolution du problème de contrôle optimal exposé précédemment est également problématique pour le respect de la contrainte terminale. Le solveur étant configuré avec un temps maximal de résolution, il est possible que celui-ci s'arrête prématurément et délivre une solution ne satisfaisant pas l'ensemble des contraintes. Ce phénomène est très marqué sur la figure 3.12a, notamment au début de la trajectoire du robot. En effet, le robot est alors très éloigné de la situation de référence et la contrainte terminale est plus compliquée à satisfaire. La stabilité du schéma en boucle fermée n'est donc manifestement pas assurée. Grâce à l'utilisation du calcul symbolique, ce problème est résolu. La figure 3.12b témoigne effectivement d'une erreur quasi nulle entre les vecteurs d'indices visuels courants et désirés tout du long de la commande.

3.4.5 Résultats expérimentaux

Dans cette section, le schéma de contrôle complet est testé sur le robot Tiago de l'équipe de recherche pour un scénario très proche de celui étudié en simulation (figure 3.13a et 3.13b). Les paramètres du schéma sont choisis d'après les résultats de simulation offrant les meilleures performances. La matrice de pondération est $Q_S = \text{diag}(1, 1, 0.1, 1, 1, 1)$ avec $N_p = N_c = 10$ et $N_r = 1$. Cette section est divisée en trois parties : la première analyse qualitativement la trajectoire du robot et de la caméra de l'organe terminal. Les deuxièmes et troisièmes parties analysent respectivement les performances de la commande par rapport à la tâche visuelle et par rapport aux contraintes de manière quantitative.

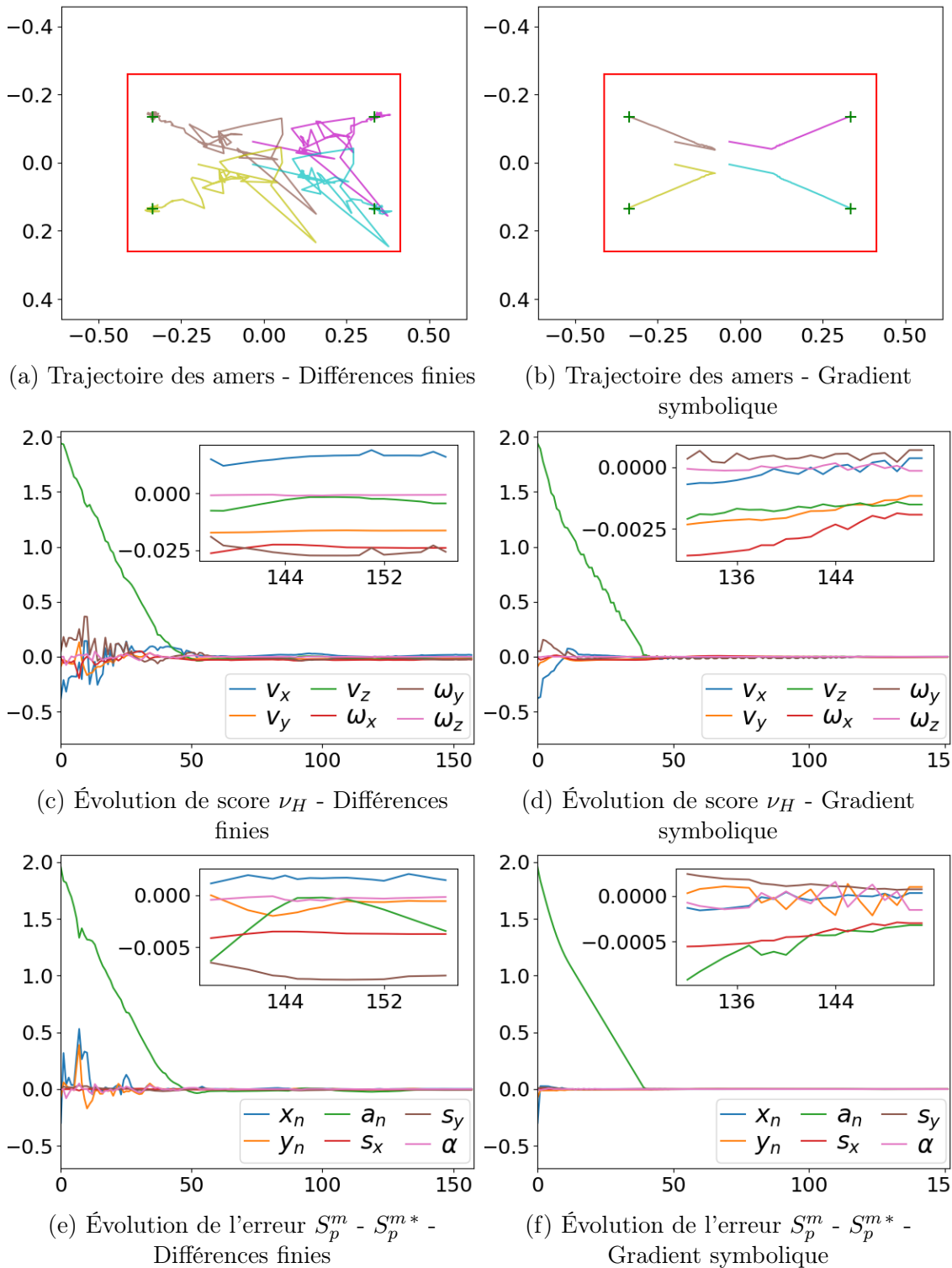


FIGURE 3.11 – Comparaison des performances relatives à la tâche visuelle pour un schéma basé sur le calcul de gradient par différences finies à gauche et sur le calcul de gradient symbolique à droite

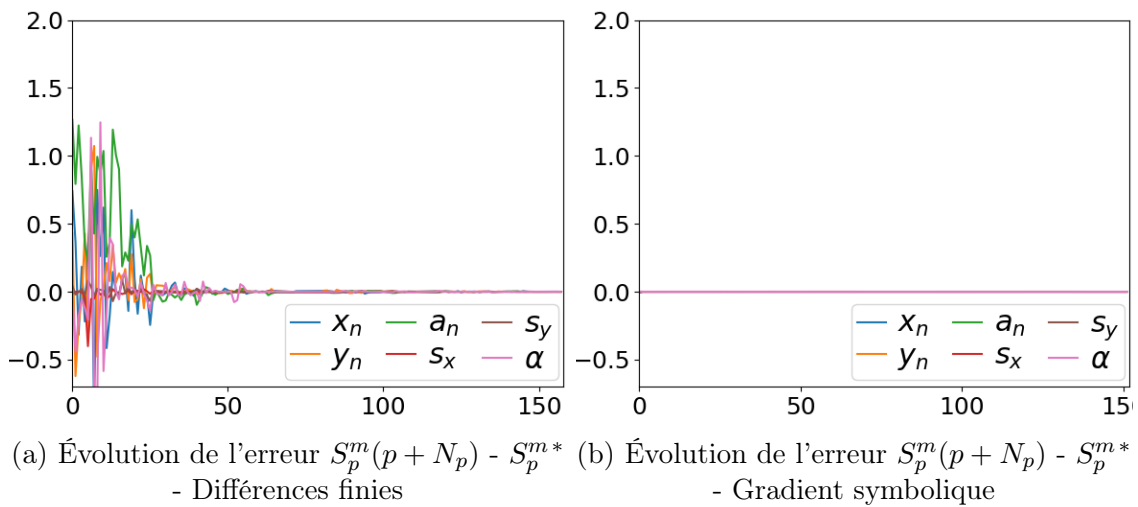
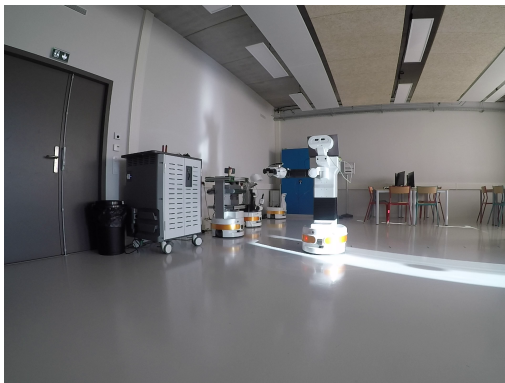
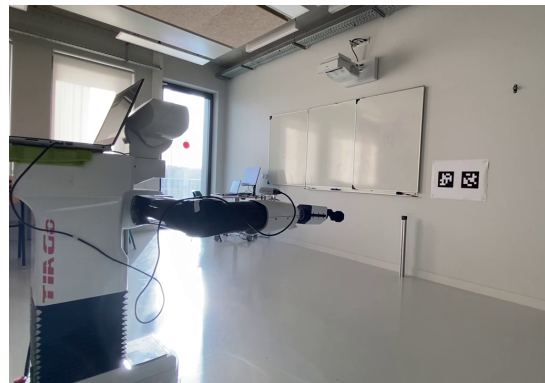


FIGURE 3.12 – Comparaison des performances relatives à la contrainte terminale pour un schéma basé sur le calcul de gradient par différences finies à gauche et sur le calcul de gradient symbolique à droite



(a) Vue 1



(b) Vue 2

FIGURE 3.13 – Scénario d'expérimentation

Analyse de la trajectoire

La figure 3.14 présente respectivement des captures d'écran de la trajectoire du robot et des images de la caméra de l'organe terminal.

Deux parties de la trajectoire du robot peuvent être différenciées sur les figures (a) – (h). La première partie (figures (a) – (d)) provoque l'avance du robot vers la cible, mais également le dépliement du bras. Bien que ce comportement soit en accord avec la réalisation de la tâche d'asservissement visuel, il n'est pas très appréciable. En effet, le robot se déplace alors le bras tendu pendant toute la suite de la trajectoire. Cette configuration est premièrement proche d'une singularité cinématique, mais a également tendance à favoriser les vibrations du système déjà très sujet à celles-ci. L'observation de ce comportement a fortement contribué à orienter les travaux de recherche de manière à éviter ce problème. C'est pourquoi ce sujet est adressé dès

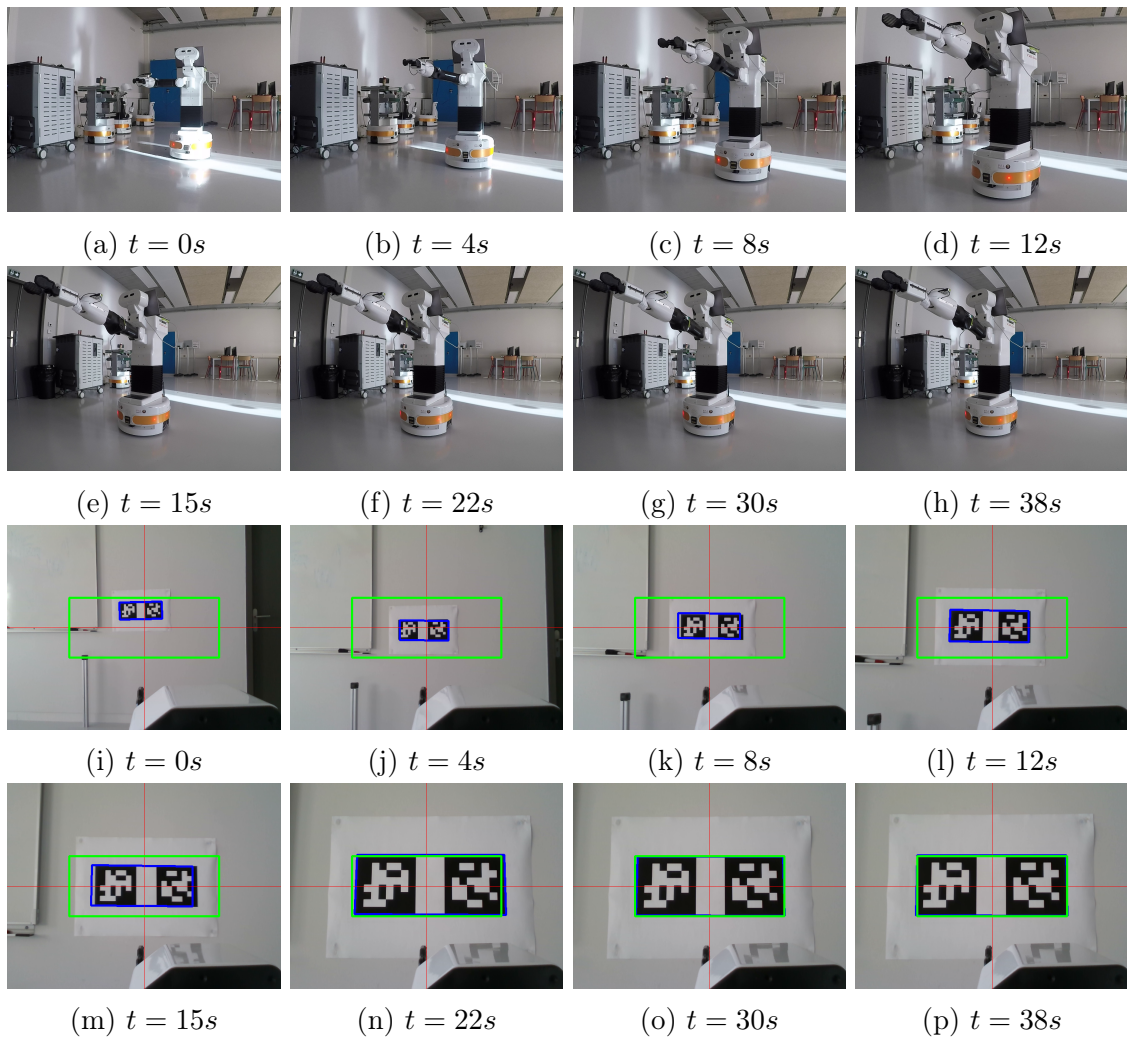


FIGURE 3.14 – Trajectoire du robot (a)-(h) et informations de la caméra (i)-(p)

le prochain chapitre.

La deuxième partie de la trajectoire (figures (e) – (h)) est caractérisée par de faibles mouvements de la base mobile. La caméra est proche de la situation de référence, mais la convergence prend beaucoup de temps. En fait, c'est le problème de sous-optimalité de la solution renvoyée par le solveur qui provoque ce comportement. En réalité, la solution utilise toujours l'ensemble des prédictions alors qu'une seule serait normalement suffisante pour atteindre l'objectif. La commande appliquée au robot est donc seulement lentement modifiée et ne permet pas à la caméra d'atteindre rapidement la situation de référence. Ce problème lié à la sous-optimalité est donc important à prendre en compte, et les travaux des chapitres précédents l'ont abordé. L'idée est d'aider et de guider le solveur pour éviter ce comportement [Durand-Petiteville and Cadenat, 2022].

Performances relatives à la tâche visuelle

Les figures 3.15a et 3.15b présentent les images perçues par la caméra respectivement à la configuration initiale et finale. Le schéma de commande permet de réaliser la tâche visuelle. Comme le montre l'évolution des indices visuels présentée en figure 3.15d. En effet, ces derniers convergent vers leurs valeurs de référence.

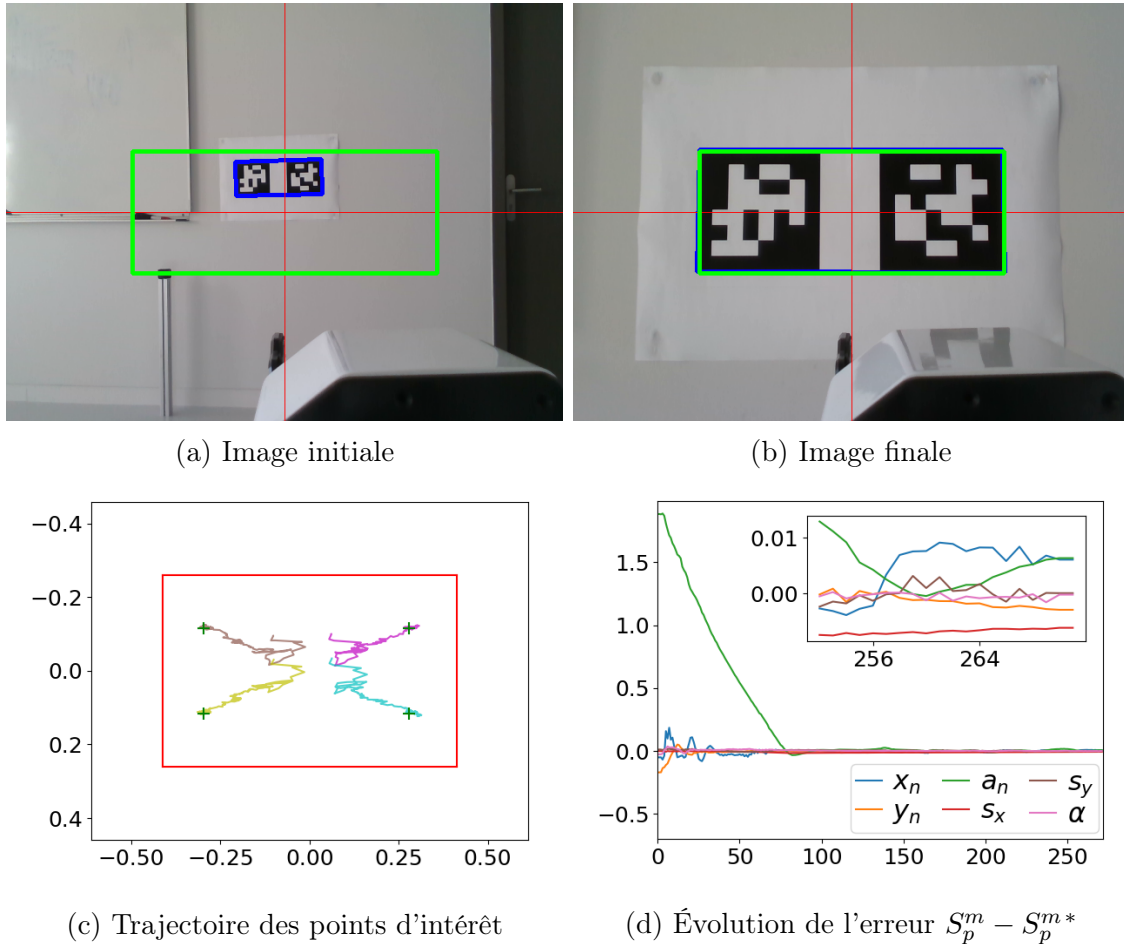


FIGURE 3.15 – Résultats du schéma complet relatifs à la tâche visuelle

Performances relatives aux contraintes

La figure 3.16 présente l'erreur relative à la contrainte terminale lors de l'expérimentation réalisée. Le contrôleur assure le respect de cette dernière quasiment parfaitement en conditions réelles. En effet, seule la contrainte terminale de la première itération de commande optimale n'est pas assurée. Celle-ci est ensuite seulement ponctuellement et faiblement non respectée dès lors que la solution initialement fournie au solveur est proche de l'optimum. La stabilité du contrôleur en boucle fermée est donc quasiment assurée tout au long de la trajectoire.

La contrainte terminale est donc vérifiée malgré une distance initiale importante à la situation de référence, grâce à la relaxation des commandes correspon-

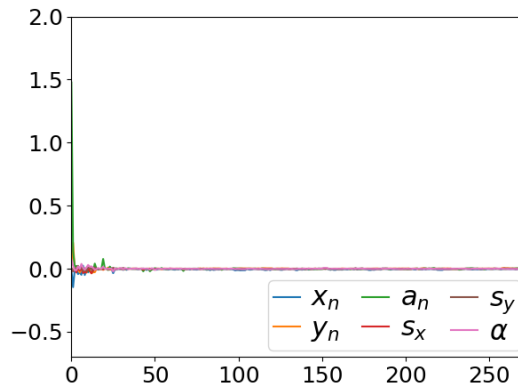


FIGURE 3.16 – Évolution de l'erreur relative à la contrainte terminale

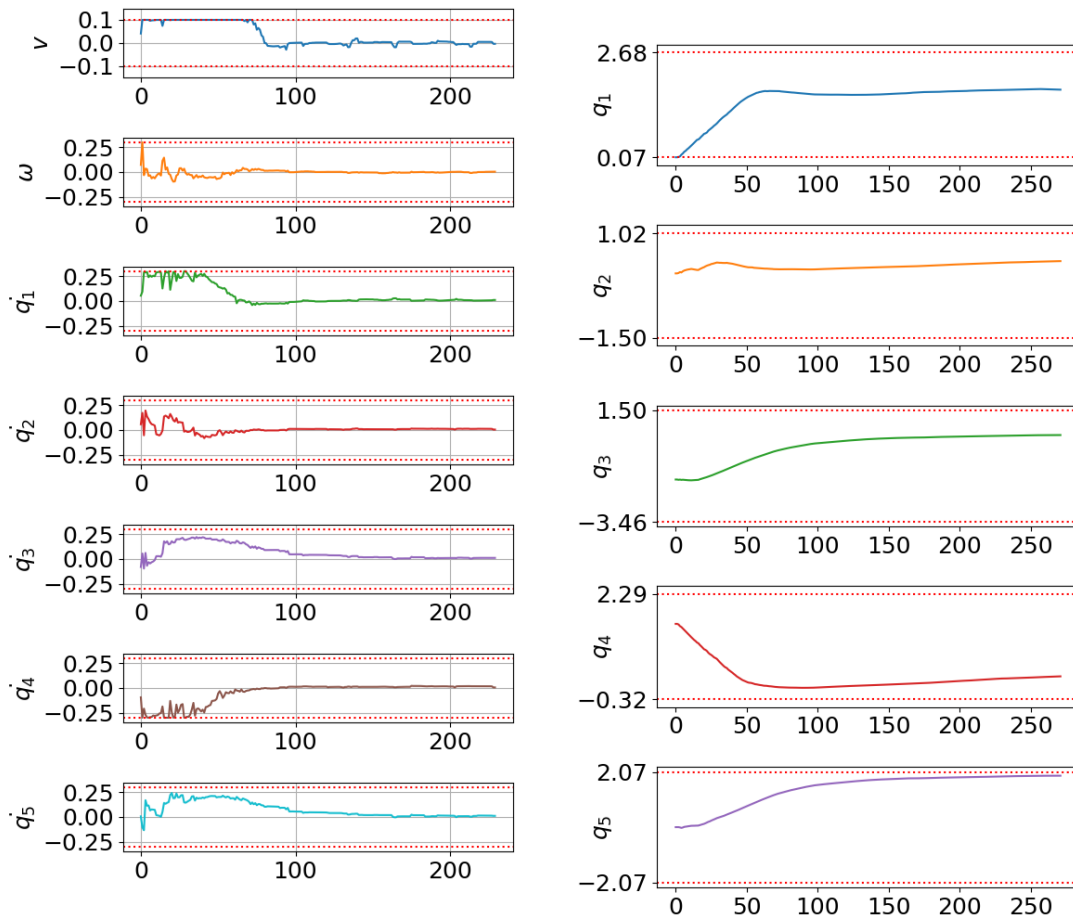
dant aux dernières prédictions. Bien évidemment, ces commandes relaxées ne sont jamais appliquées réellement au robot. En effet, la figure 3.17a, qui trace l'évolution des commandes appliquées au robot tout au long de la trajectoire, prouve que les contraintes liées aux actionneurs sont bien respectées. La figure 3.17b sur les côtés présente quant à elle l'évolution des paramètres articulaires, et confirme que les contraintes de butées articulaires sont elles aussi vérifiées.

3.5 Conclusion

Ce chapitre a permis d'établir un schéma d'asservissement visuel prédictif mono-caméra permettant de piloter le manipulateur mobile *TIAGO* et servira de base solide pour approfondir la problématique étudiée. La loi de commande établie combine donc les avantages de l'asservissement visuel 2D et de la commande prédictive. La tâche de positionnement est définie dans l'image, et les étapes de reconstruction d'information 3D sont minimales. Le choix des indices visuels au cœur de la fonction de coût a été attentivement étudié en rassemblant de nombreux éléments de la littérature. En l'occurrence, l'utilisation des moments a permis d'obtenir un positionnement final précis et permet d'influencer la trajectoire de la caméra dans le but de maximiser les performances de schéma de commande. L'intégration de la tâche visuelle au sein d'une commande prédictive a permis de prendre directement en compte toutes sortes de contraintes, chose généralement non-triviale dans un schéma d'asservissement visuel classique.

De plus, la stabilité en boucle fermée du système est assurée grâce à une adaptation astucieuse de la contrainte terminale au contexte de manipulation mobile. Cette propriété est souvent omise dans les rares travaux similaires. La relaxation de certaines limites physiques des actionneurs permet d'obtenir un horizon de prédiction suffisamment long pour respecter la contrainte terminale malgré la distance initiale importante entre la caméra et la cible.

Des résultats encourageants ont été obtenus sur la plateforme réelle. Ces expéri-



(a) Évolution de la commande appliquée au robot

(b) Évolution des paramètres articulaires

FIGURE 3.17 – Résultats du schéma complet relatifs à la tâche visuelle

mentations ont été rendues possibles grâce à l'attention toute particulière portée sur l'implémentation et la résolution du problème d'optimisation. Le calcul de gradient symbolique a largement contribué à l'obtention de bonnes solutions.

Cependant, un certain nombre de comportements non-désirés ont pu être observés. Premièrement, le robot a tendance à déplier son bras pour minimiser plus rapidement la fonction de coût. Le système se déplace alors le bras tendu, favorisant les vibrations et la proximité de singularités. De la même manière, la configuration finale n'est pas adéquate à une possible future tâche de manipulation. D'après ce constat, l'objectif d'harmoniser la trajectoire du robot a été fixé et fait l'objet du chapitre suivant.

Ensuite, malgré les efforts effectués pour améliorer la résolution du problème d'optimisation, la sous-optimalité de la solution ne permet pas une convergence assez rapide. L'idée d'aider le solveur en forçant la décroissance de la fonction de coût pour améliorer la résolution est également un objectif du prochain chapitre.

Enfin, la configuration initiale du robot est contraignante et notre volonté de

réaliser une loi de commande permettant de réaliser des tâches variées, répondant aux attentes actuelles de la robotique, nous a encouragé à repenser le schéma de commande. Il s'agit également d'une voie d'amélioration empruntée par le chapitre suivant.

Chapitre 4

Asservissement visuel prédictif multi-caméra pour manipulateur mobile

4.1 Introduction

Ce chapitre présente une version améliorée du contrôleur VPC dont les modifications ont pour objectif de répondre aux problèmes soulevés précédemment.

Le premier point concerne la tendance du robot de se déplacer le bras tendu, amplifiant les vibrations et diminuant la capacité de déplacement de l'organe terminal. La configuration finale semble également qualitativement inadéquate. L'objectif de ce chapitre est d'améliorer la coordination base-bras en utilisant notamment la redondance du système pour maximiser un critère de performance. Ce critère doit permettre de quantifier par une mesure la qualité de la configuration du système robotique. Un autre but que nous avons visé est de produire une trajectoire plus proche des mouvements humains. Ce dernier, qui semble de premier abord subjectif, rejoint en fait la problématique d'une trajectoire éloignée des singularités. Ce chapitre aborde ces problématiques en utilisant la mesure de manipulabilité qui quantifie les notions évoquées : allant d'une quantification de la distance aux singularités [Yoshikawa, 1985b] à une meilleure interaction humain-machine [Jaquier et al., 2020]. L'intégration de cette mesure dans le schéma VPC est complexe et de nombreux problèmes émergents ont dû être résolus.

Le deuxième point central de l'amélioration du schéma de commande est de permettre une plus grande autonomie du robot, ainsi qu'une extension de la gamme de tâches que le contrôleur doit pouvoir réaliser. En effet, le manipulateur mobile n'utilise pour l'instant pas sa pleine capacité, majoritairement parce que les mouvements sont très limités par les contraintes de visibilité. Ce chapitre présente une modification du contrôleur permettant de répondre à ce problème grâce à l'ajout d'une caméra et à l'utilisation judicieuse de ses informations. L'objectif est donc de piloter le robot à partir d'une configuration initiale quelconque : la seule contrainte étant que la cible doit être visible par une des caméras. Une fois de plus, cet axe

d'amélioration nécessite des efforts de mise en œuvre. La définition de la fonction en utilisant la projection perspective par exemple soulève des problèmes lorsque l'organe terminal ne fait pas face à la cible. L'utilisation d'une méthode de projection plus adéquate, la projection sphérique et le choix d'indices visuels adaptés [Tahri, 2004] sont alors nécessaires.

Enfin, la sous-optimalité de la solution ne permet pas une convergence assez rapide. La correction de cette lacune rejoint les enjeux de l'intégration de la mesure de manipulabilité. En effet, la solution est sous-optimale vis-à-vis de la tâche d'asservissement visuel. Un triptyque de contraintes variables dans le temps permettant de guider la résolution du problème d'optimisation est proposé et constitue une contribution de ces travaux de thèse.

Ce chapitre présente dans un premier temps l'adaptation du choix d'indices visuels à la projection sphérique pour résoudre les problèmes relatifs à la projection perspective classique évoqués précédemment. Après un rappel de l'état de l'art sur la manipulabilité, l'intégration de cette dernière dans la loi de commande est décrite. D'après ces changements, les modifications apportées au schéma sont ensuite indiquées et deux versions d'asservissement visuel améliorées sont proposées. Pour finir, ces deux schémas de commande sont à nouveau testés en simulation puis sur la plateforme robotique réelle et les résultats sont analysés.

4.2 Stratégie multi-caméra et choix des indices visuels

Pour permettre une plus grande autonomie du robot, la caméra de l'organe terminal doit pouvoir être orientée de manière quelconque par rapport à la cible. Cependant, des problèmes de visibilité apparaissent alors inévitablement. En effet, le principal problème des caméras "classiques" basées sur la projection perspective est leur champ de vision limité. La caméra doit rester dans une position frontale par rapport à la cible pendant l'asservissement visuel et les mouvements de celle-ci sont très contraints. Deux solutions ont été mentionnées : l'utilisation de caméras omnidirectionnelles et la coopération de caméra EinH (Eye-in-Hand, caméra de l'organe terminal) et EtoH (Eye-to-Hand, caméra déportée). Dans le contexte de la manipulation mobile, la caméra de l'organe terminal peut se retrouver initialement dans des configurations très différentes par rapport à la cible. Il semble que l'utilisation d'une caméra omnidirectionnelle en EinH ne soit pas suffisante pour régler le problème et qu'il reste difficile de garantir la visibilité durant toute la durée de la tâche. De plus, le robot *TIAGo* comporte une deuxième caméra au niveau de la tête. Son référentiel a été défini comme F_{ch} , pouvant être considérée en configuration EtoH relativement au robot. La coopération des deux caméras apparaît alors comme une solution à la problématique considérée puisque l'utilisation des informations de la caméra de la tête permet à la caméra de l'organe terminal de garder toute sa liberté de mouvement. La contrainte de visibilité étant alors sur la caméra déportée, dont les mouvements sont nettement moins contraints.

Cependant, il reste un problème souvent écarté dans la littérature : ce problème est même intrinsèque aux caméras perspectives, car la méthode de projection présente une singularité en $Z_c = 0$, mais aussi une discontinuité de la projection lors du passage de $Z_c > 0$ à $Z_c < 0$ et vice versa. En effet, les travaux considèrent généralement des positionnements relatifs des caméras EinH et EtoH cohérents permettant d’avoir tout le temps un champ de vision commun. Ces configurations s’éloignent donc de la singularité et de la discontinuité mentionnées. Toutefois, ces conditions ne peuvent pas être respectées dans le cadre général de la manipulation mobile. [Kermorgant and Chaumette, 2011] par exemple, étudie un scénario avec des déplacements faibles et un champ de vision des deux caméras globalement commun. D’autres travaux expriment la tâche dans un repère global, jusqu’à ce que la caméra EinH soit correctement bien positionnée pour que la singularité dans l’image ne soit plus présente et bascule ensuite sur un asservissement visuel classique [Chang and Shao, 2010] [Cuevas-Velasquez et al., 2018]. Ces stratégies sont bien adaptées pour un bras robotique uniquement, mais se complexifient lorsque la caméra EtoH n’est pas fixe et qui nécessite une étape de reconstruction 3D.

La stratégie proposée dans ces travaux de thèse évite ces inconvénients en cherchant à exprimer une tâche générique dans l’image permettant la réalisation de diverses trajectoires sans restriction sur le positionnement relatif des deux caméras. En fait, l’idée se trouve à l’intersection des solutions des caméras omnidirectionnelles et de la coopération de caméras EinH et EtoH. Elle consiste à utiliser virtuellement l’étape de projection sphérique des caméras catadioptriques avec les informations de la caméra EtoH. L’image virtuelle alors obtenue est sphérique et ne présente alors plus de problème de singularité ou de discontinuité. La tâche peut alors être définie dans cette image et aucune condition sur le positionnement relatif des caméras ne doit plus être respectée. Cette stratégie utilise des informations 3D provenant des capteurs, mais ces dernières n’ont pas besoin d’être précises. Les caméras du système robotique sont d’ailleurs d’une précision moyenne et la robustesse du schéma de contrôle justifie leurs utilisations. Le choix des indices visuels doit donc être repensé. La théorie des invariants introduite dans le chapitre précédent se généralise pour des moments de dimensions supérieures à 2. Les moments 3D sont très utiles lorsqu’une projection sphérique est utilisée. L’objectif est de réaliser un choix d’indices visuels d’une manière analogue au chapitre précédent.

La suite de cette section définit mathématiquement cette projection sphérique et suit une structure similaire à celle des indices visuels correspondant à la projection perspective, toujours dans le but d’obtenir le meilleur découplage possible.

4.2.1 Projection sphérique et moments 3D

— Définition

Le principe de projection sphérique est illustré sur la figure 4.1 : un point 3D $\mathcal{X}_c = [X_c, Y_c, Z_c]$ est projeté sur la sphère unité en un point $\mathbf{x}_s = [x_s, y_s, z_s]^T$.

La relation de projection est alors donnée par l’expression suivante [Tahri, 2004] :

$$\mathbf{x}_s = \mathcal{X}_c / \|\mathcal{X}_c\| \quad (4.2.1.1)$$

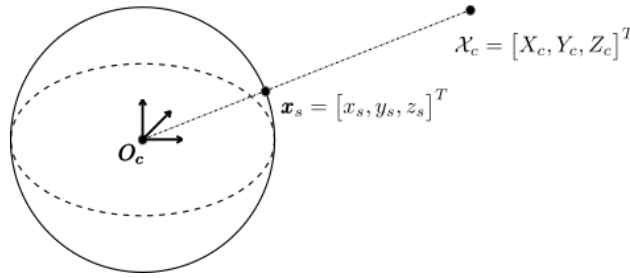


FIGURE 4.1 – Projection sphérique

En introduisant $\rho = \|\mathcal{X}_c\| = \sqrt{X_c^2 + Y_c^2 + Z_c^2}$ et la matrice de projection sphérique sH_c , cette expression devient :

$$\bar{\mathbf{x}}_s = \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} = \begin{bmatrix} 1/\rho & 0 & 0 & 0 \\ 0 & 1/\rho & 0 & 0 \\ 0 & 0 & 1/\rho & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = {}^sH_c \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = {}^sH_c \bar{\mathcal{X}}_c \quad (4.2.1.2)$$

Il faut noter que bien que surfacique, la projection sphérique n'est plus plane. Il serait tentant de vouloir se ramener à des coordonnées 2D sur la surface de la sphère, et d'y appliquer des méthodes similaires à la projection perspective. Malheureusement, la caractéristique cyclique de ces systèmes de coordonnées fait apparaître de nouvelles discontinuités [Tatsambon Fomena, 2008]. À nouveau, si l'espace de travail est limité aux cas sans discontinuité dans les coordonnées sphériques, les moments 2D pourraient à nouveau être utilisés.

— Points

Du fait de l'impossibilité d'utiliser des coordonnées en deux dimensions, il semble intuitif de s'orienter vers des informations en trois dimensions. Les points, comme les moments, sont calculés et exprimés en 3D dans cette partie. Ainsi, les points d'intérêt peuvent être utilisés directement comme indices visuels :

$$S_s^p = [x_{s1}, y_{s1}, z_{s1}, x_{s2}, y_{s2}, z_{s2}, x_{s3}, y_{s3}, z_{s3}, x_{s4}, y_{s4}, z_{s4}] \quad (4.2.1.3)$$

Le même problème relatif au couplage de ces indices apparaît, et ce dernier peut à nouveau être étudié par l'analyse de la matrice d'interaction correspondante. La matrice d'interaction d'un point du plan image sphérique est donnée dans [Tahri, 2004] :

$$\begin{bmatrix} \dot{x}_s \\ \dot{y}_s \\ \dot{z}_s \end{bmatrix} = L_{\mathbf{x}_s} \cdot {}^{F_c}T_{O_c \in F_c/F} = \begin{bmatrix} -\frac{1}{\rho} + \frac{x_s^2}{\rho} & \frac{x_s y_s}{\rho} & \frac{x_s z_s}{\rho} & 0 & -z_s & y_s \\ \frac{x_s y_s}{\rho} & -\frac{1}{\rho} + \frac{y_s^2}{\rho} & \frac{y_s z_s}{\rho} & z_s & 0 & -x_s \\ \frac{x_s z_s}{\rho} & \frac{y_s z_s}{\rho} & -\frac{1}{\rho} + \frac{z_s^2}{\rho} & -y_s & x_s & 0 \end{bmatrix} \cdot {}^{F_c}T_{O_c \in F_c/F} \quad (4.2.1.4)$$

La forme particulière de la partie correspondant aux rotations peut d'ores et déjà être remarquée. Par définition, la projection sphérique présente de bonnes propriétés par rapport aux mouvements de rotation. En effet, un mouvement de rotation de la caméra se traduit par le même mouvement sur la projection sphérique de l'objet et laisse donc sa forme inchangée. Cette propriété rend conceptuellement différente l'étude d'invariants par rapport au cas de la projection perspective : toutes les grandeurs quantifiant la forme de la projection ne paraissent plus adéquates pour contrôler les mouvements de rotation. L'étude des moments centrés est plus judicieuse dans ce cas pour contrôler les mouvements de translation.

— Moments 3D

Les objectifs de l'utilisation des moments avec une projection sphérique sont similaires à ceux relatifs à la projection perspective : une représentation générique d'objets de formes différentes et un découplage maximal cependant plus compliqué à obtenir.

Cette fois encore, deux types de moments peuvent être définis : continus et discrets. Seulement dans ce cas, le calcul analytique des moments continus correspondant à la projection sphérique est plus complexe. C'est pourquoi les moments discrets ont été utilisés dans ce travail. Si O est l'objet observé et O_s sa projection sphérique, ces derniers sont définis de la sorte [Tahri, 2004] :

$$m_{i,j,k} = \sum_{O_s} x_s^i y_s^j z_s^k \quad (4.2.1.5)$$

Cette fois-ci encore, la caractéristique géométrique du centre de gravité est très intéressante puisqu'elle permet une représentation générique tout en gardant les propriétés relatives aux mouvements de rotation. Ces coordonnées du centre de gravité \mathbf{x}_{sg} en trois dimensions sont alors définis comme ceci :

$$x_{sg} = m_{100}/m_{000} \quad (4.2.1.6)$$

$$y_{sg} = m_{010}/m_{000} \quad (4.2.1.7)$$

$$z_{sg} = m_{001}/m_{000} \quad (4.2.1.8)$$

La matrice d'interaction des coordonnées du centre de gravité peut être calculée analytiquement en faisant à nouveau l'hypothèse d'un objet plan. Son expression, donnée dans [Tahri, 2004], étudiée localement à la pose désirée, donne :

$$L_{\mathbf{x}_{sg}} = \begin{bmatrix} C_1 & 0 & 0 & 0 & -z_{sg} & 0 \\ 0 & C_2 & 0 & z_{sg} & 0 & 0 \\ 0 & 0 & C_3 & 0 & 0 & 0 \end{bmatrix} \quad (4.2.1.9)$$

où les C_i sont encore des valeurs numériques de même ordre de grandeur.

À nouveau, cette analyse n'est que locale autour de la situation désirée. Les observations des mouvements quantifiés par ces coordonnées sont analogues à celles

réalisées avec la projection perspective. Même si certaines valeurs ne sont plus exactement égales à zéro lorsqu'on s'éloigne de ce voisinage, les tendances se conservent. Cependant, il faut noter qu'il n'est plus possible d'indiquer quelle coordonnée quantifie tel mouvement sans préciser la pose de la caméra étudiée. En effet, avec des rotations d'angles comme 90° , cette forme de matrice d'interaction est retrouvée, mais le découplage est différent. Tout ce qui peut être indiqué globalement est que ces trois coordonnées du centre de gravité proposent tout de même un découplage correct localement en toute pose de caméra pour tous les mouvements, excepté la rotation autour d'un axe dirigé vers la cible.

Ces coordonnées peuvent ainsi être utilisées pour contrôler les mouvements énoncés, mais aucun découplage parfait n'est possible. C'est pourquoi on les retrouve dans certains travaux comme indices visuels pour représenter les translations [Tatsambon Fomena, 2008] et dans d'autres les rotations [Tahri et al., 2008]. En fait, les travaux [Tatsambon Fomena, 2008] expliquent même qu'il paraît impossible de déterminer des invariants aux translations à partir des moments sphériques. Les travaux proposent tout de même des informations pour contrôler les rotations, mais le découplage est loin d'être réalisé. Afin d'obtenir ce découplage des translations et des rotations selon les axes non-optiques, une information 3D est utilisée ici. Le vecteur normal à la cible N_v est extrait et son orientation par rapport à l'axe optique z_c est quantifiée par l'information C_{xy} :

$$C_{xy} = \|N_v \times z_c\|^2 \quad (4.2.1.10)$$

Sa matrice d'interaction est alors intuitivement découplée et uniquement dépendante des vitesses de rotation selon les axes non-optiques. En posant $N_v = [n_x, n_y, n_z]^T$, son expression analytique peut être calculée :

$$N_v \times z_c = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} n_y \\ -n_x \\ 0 \end{bmatrix} \quad (4.2.1.11)$$

d'où $C_{xy} = n_x^2 + n_y^2$ et $\dot{C}_{xy} = 2n_x \dot{n}_x + 2n_y \dot{n}_y$. Enfin, en définissant le vecteur de vitesse de rotation de la caméra $\Omega_{F_c/F_0} = [\omega_x, \omega_y, \omega_z]^T$ et appliquant la formule classique des corps rigides au vecteur N_v , il vient :

$$\dot{N}_v = \begin{bmatrix} \dot{n}_x \\ \dot{n}_y \\ \dot{n}_z \end{bmatrix} = -\Omega_{F_c/F_0} \times N_v = \begin{bmatrix} -\omega_y n_z + \omega_z n_y \\ -\omega_z n_x + \omega_x n_z \\ -\omega_x n_y + \omega_y n_x \end{bmatrix} \quad (4.2.1.12)$$

En injectant ces résultats dans l'expression de \dot{C}_{xy} , la matrice d'interaction devient :

$$L_{C_{xy}} = [0 \quad 0 \quad 0 \quad 2n_y n_z \quad -2n_x n_z \quad 0] \quad (4.2.1.13)$$

En fait cette matrice d'interaction, grâce à l'utilisation d'une information 3D, conserve cette forme. Cette étape comprend donc une reconstruction d'information

3D, mais la commande n'utilise ces caractéristiques visuelles que lors du régime transitoire chargé de rectifier l'orientation de la caméra de l'organe terminal afin d'éviter les problèmes de reprojction. Les propriétés en régime permanent de l'asservissement visuel 2D sont donc conservées. De plus, seule une estimation grossière du vecteur normal N_v est nécessaire. En fait, les rotations étant déjà contrôlées par les coordonnées du centre de gravité, l'indice C_{xy} n'est ajouté que pour éliminer les dégénérescences de la matrice d'interaction dans certaines configurations.

Pour contrôler les translations, des invariants aux rotations sont disponibles comme expliqué précédemment [Tahri, 2004] mais leur matrice d'interaction est toujours dominée par le terme dépendant du mouvement de translation dirigé vers la cible. De tels invariants sont donc tout de même adéquats pour contrôler le mouvement d'avance vers la cible. D'après cette constatation, un seul de ces invariants, I_1 basé sur les moments sphériques d'ordre minimum, est conservé et suffit à contrôler ce degré de liberté :

$$I_1 = m_{200}m_{020} - m_{200}m_{002} - m_{020}m_{002} + m_{110}^2 + m_{101}^2 + m_{011}^2 \quad (4.2.1.14)$$

La matrice d'interaction de cet invariant est par exemple donnée localement autour de la pose désirée par :

$$L_{I_1} = [C_1 \ C_2 \ C_3 \ 0 \ 0 \ 0] \quad (4.2.1.15)$$

Avec ici, $C_3 \gg C_1$ et $C_3 \gg C_2$.

Enfin, le dernier indice visuel utilisé pour contrôler la rotation autour de l'axe optique est analogue à celui utilisé en projection perspective. Il suppose une reprojction perspective à partir de la projection sphérique pour retrouver une surface plane. Cet indice visuel est largement utilisé dans la littérature, par exemple [Tahri et al., 2008], lors de l'utilisation de caméras catadioptriques. Cette projection plane amène cependant des singularités, dont le domaine dépend de la position du centre de cette projection perspective. Dans nos travaux, comme ces étapes de projection sont virtuelles, il est judicieux de choisir un centre de projection pour lequel ce domaine de singularité est le plus faible possible. Cette condition se traduit par la définition du pôle nord de la sphère comme centre de la projection plane, il en résulte une seule singularité sur la sphère correspondant à ce centre de projection. Ainsi, seule la demi-droite $-z_c$ ne peut pas être projetée et correspond à une configuration à laquelle la cible se trouve à l'opposé du champ de visée de la caméra. Cette projection est généralement appelée projection stéréographique, et pour un point de la sphère $\mathbf{x}_s = [x_s, y_s, z_s]^T$, sa projection $\mathbf{x}_{st} = [x_{st}, y_{st}, z_{st}]^T$, tous deux exprimés dans le repère de la caméra, se détermine par la relation suivante :

$$\begin{bmatrix} x_{st} \\ y_{st} \\ z_{st} \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{1+z_s} & 0 & 0 & 0 \\ 0 & \frac{1}{1+z_s} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} = {}^{st}H_s \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{x_s}{1+z_s} \\ \frac{y_s}{1+z_s} \\ 0 \\ 1 \end{bmatrix} \quad (4.2.1.16)$$

Ou directement à partir du point 3D $\mathcal{X}_c = [X_s, Y_c, Z_c]^T$:

$$\begin{bmatrix} x_{st} \\ y_{st} \\ z_{st} \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\rho+Z_c} & 0 & 0 & 0 \\ 0 & \frac{1}{\rho+Z_c} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = {}^{st}H_c \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{X_c}{\rho+Z_c} \\ \frac{Y_c}{\rho+Z_c} \\ 0 \\ 1 \end{bmatrix} \quad (4.2.1.17)$$

Il est ensuite possible d'utiliser les moments 2D de cette projection stéréographique pour obtenir une orientation α_{st} analogue à celle obtenue en projection perspective :

$$\alpha_{st} = \frac{1}{2} \arctan \left(\frac{2\mu_{11\ st}}{\mu_{20\ st} - \mu_{02\ st}} \right) \quad (4.2.1.18)$$

où $\mu_{ij\ st}$ correspond au moment centré d'ordre $i + j$ de la projection stéréographique. Une analyse locale autour de la pose désirée de la matrice d'interaction correspondante donne :

$$L_{\alpha_{st}} = [0 \ 0 \ 0 \ 0 \ 0 \ C_1] \quad (4.2.1.19)$$

où C_1 est une valeur numérique non nulle.

Le choix d'indices visuels génériques et découplés a donc été plus complexe que pour la projection perspective, mais ceci est dû au caractère local de cette dernière. En effet, la projection perspective peut être vue comme un cas particulier du modèle de projection unifié en se limitant à des orientations relatives à la cible faibles selon les axes non-optiques. Il semble alors normal que la complexité augmente énormément. Malgré cela, six indices visuels adéquats ont été déterminés pour définir le vecteur S_s^m :

$$S_s^m = [x_{sg}, y_{sg}, z_{sg}, C_{xy}, I_1, \alpha_{st}] \quad (4.2.1.20)$$

Ce choix n'est certainement pas optimal, d'autres indices issus des travaux de [Tatsambon Fomena, 2008] auraient par exemple pu être utilisés.

4.3 Choix de la tâche secondaire - Mesure de manipulabilité

Comme expliqué précédemment, un enjeu majeur relatif au contrôle de manipulateurs mobiles est la coordination base-bras. Ce dernier peut être vu comme une tâche secondaire par rapport à l'objectif principal de régulation à la pose de référence, destinée à utiliser les degrés de mobilité interne pour optimiser un critère. Ce critère peut être de toute nature, mais un critère privilégié depuis longtemps dans le domaine de la manipulation ou manipulation mobile est la maximisation de la manipulabilité. Introduite historiquement dans les travaux [Yoshikawa, 1985b], la manipulabilité est une mesure qui quantifie généralement la capacité de l'organe terminal à effectuer des mouvements considérés. Cette grandeur est à l'origine utilisée pour calculer la distance aux singularités, mais est rapidement adaptée pour définir les configurations optimales d'un robot, dans le sens d'une dextérité de mouvement.

La mesure de manipulabilité dépend de la jacobienne du robot considéré. En l'occurrence, pour le bras robotique étudié ici, elle est définie comme ceci [Yoshikawa, 1985b] :

$$w_a = \sqrt{\det(J_a(\chi_a) J_a(\chi_a)^T)} \quad (4.3.0.1)$$

En fait, la manipulabilité correspond au volume de l'ellipsoïde de l'espace des vitesses opérationnelles. Si ce volume est nul, cela signifie qu'au moins un des demi-axes de l'ellipsoïde est nul et que le robot ne peut pas se déplacer suivant cette direction, et donc que le robot est en singularité. Une autre méthode pour calculer cette mesure consiste à étudier les demi-axes de l'ellipsoïde en utilisant la décomposition SVD de la jacobienne considérée [Yoshikawa, 1985b] :

$$w_a = \prod_{i=1}^n \sigma_i \quad (4.3.0.2)$$

où σ_i sont les valeurs singulières de la jacobienne J_a correspondant aux n vitesses opérationnelles considérées.

Depuis, cette mesure a été largement reprise et ajustée. [Klein and Blaho, 1987] propose par exemple d'utiliser le conditionnement de la jacobienne pour privilégier l'uniformité de l'ellipsoïde plutôt que son volume :

$$w_{a2} = \frac{\sigma_{max}}{\sigma_{min}} \quad (4.3.0.3)$$

où σ_{min} et σ_{max} correspondent respectivement aux valeurs singulières minimale et maximale de J_a . Une autre méthode consiste à directement utiliser σ_{min} pour ne prendre en compte que l'axe de l'ellipsoïde problématique.

La manipulabilité permet donc de quantifier la distance aux singularités cinématiques. De la même manière, il est intéressant de quantifier la distance aux limites articulaires pour en favoriser l'éloignement. Une modification des mesures de manipulabilité est possible pour prendre en compte ces limites de construction. Les

travaux de [Tsai, 1986] utilisent une fonction de pénalité P qui diminue à proximité d'une limite articulaire, tout en maintenant la manipulabilité inchangée lorsque cette limite est à une distance raisonnable. La fonction de pénalité proposée est la suivante :

$$P(\chi_a) = 1 - \exp\left(-K_p \prod_{i=0}^{n_q} \frac{(q_i - q_{i \max})(q_{i \min} - q_i)}{q_{i \max} - q_{i \min}}\right) \quad (4.3.0.4)$$

où n_q est le nombre d'articulations et K_p est un facteur permettant de gérer la vitesse de décroissance à proximité des limites articulaires. Ces travaux proposent déjà une grandeur pour assembler les deux métriques, mais une version améliorée w'_a est proposée dans [Nelson and Khosla, 1995]. Pour le bras, elle est simplement définie de la sorte :

$$w'_a = Pw_a^2 \quad (4.3.0.5)$$

La figure 4.2 illustre le principe de cette nouvelle mesure pour une seule articulation q_i : la mesure w' prend donc en compte les deux types de problème et tend vers 0 lorsque le robot s'approche des singularités cinématiques et quand il s'approche des butées articulaires.

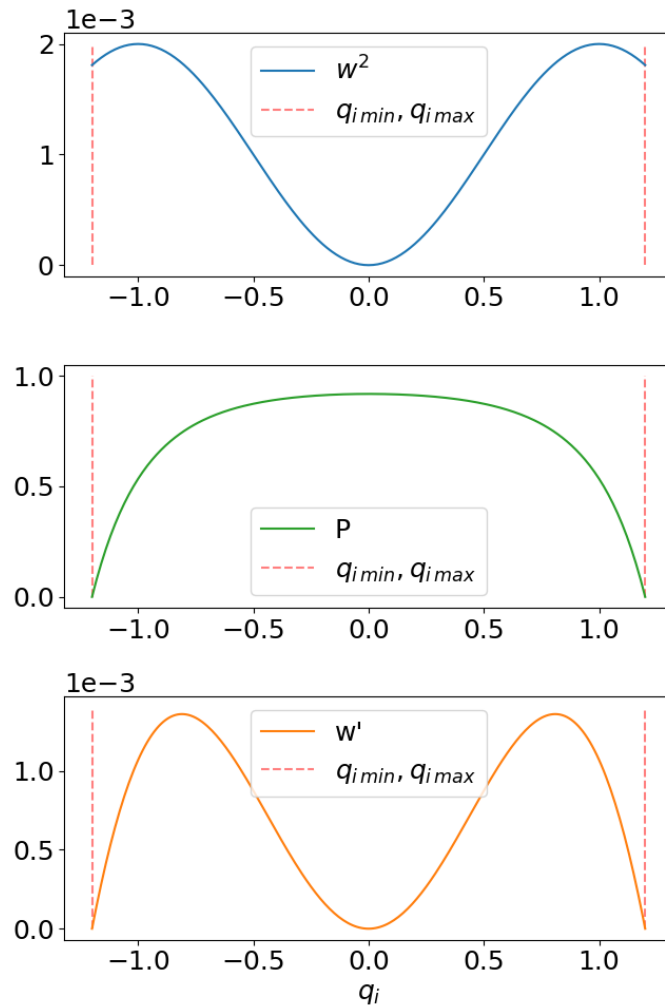
La notion de manipulabilité est également rapidement étendue de manière à prendre en compte la dynamique du système robotique en définissant la manipulabilité dynamique w_d à l'aide de la matrice d'inertie du système M [Yoshikawa, 1985a] :

$$w_d = \sqrt{\det(J_a (M^T M)^{-1} J_a^T)} \quad (4.3.0.6)$$

Elle est également étendue aux manipulateurs mobiles, même s'il est nécessaire de remarquer certaines différences puisque la matrice au cœur de la mesure n'est pas une jacobienne au sens rigoureux du terme [Bayle et al., 2001] :

$$w_{mm} = \sqrt{\det(\bar{J}_{mm}(\chi_a) \bar{J}_{mm}(\chi_a)^T)} \quad (4.3.0.7)$$

Depuis son introduction, la mesure de manipulabilité est devenue l'indice de performance privilégié des travaux de robotique. Elle permet de décrire la capacité d'un organe terminal à se déplacer dans n'importe quelle direction. Par conséquent, en maximisant cette mesure, il est par exemple possible de réduire la probabilité de défaillance du robot due à une singularité, tout en améliorant la capacité du robot à réaliser des tâches secondaires comme éviter des obstacles par exemple. D'abord utilisée pour les bras et les robots redondants, elle est également largement intégrée dans les contrôleurs pour manipulateurs mobiles. [Chen et al., 2018] l'utilise pour générer des trajectoires adéquates à la réalisation de tâches de préhension avec dextérité. [Xing et al., 2021] intègre la manipulabilité après avoir remarqué que, sans son utilisation, le contrôleur déplace le bras jusqu'aux limites de son espace de travail, et active uniquement ensuite la base mobile. Le bras a donc tendance à s'approcher de ces limites pour atteindre un étirement complet. [Colucci et al., 2022] l'exploite

FIGURE 4.2 – Illustration du concept de w'

également pour coordonner les mouvements des sous-systèmes afin de positionner le bras dans une configuration optimale pour réaliser sa tâche et le terme "dextérité" est à nouveau employé. Dans [Yang et al., 2022], la manipulabilité est utilisée à la fois comme critère pour entraîner un réseau dans une méthode d'apprentissage par renforcement et comme fonction de coût du MPC pour des interactions humain-robot. La planification de mouvements prenant en compte le confort humain est certainement un enjeu majeur du futur de la robotique, et ces travaux mettent en avant l'étude de la manipulabilité pour répondre à ce problème.

Bien que cet indice soit largement utilisé, tous les travaux présentés portant sur l'étude de manipulateurs mobiles n'utilisent que la manipulabilité du bras, et non celle du système entier. À ce propos, les travaux de [Bayle et al., 2001] et [Bayle et al., 2003] mentionnent qu'utiliser la manipulabilité du système entier seulement, peut s'avérer dommageable, car les précisions de déplacements ne sont souvent pas les mêmes pour les deux sous-systèmes, et il est souvent adéquat d'éviter une posture

en singularité du bras. Dans une telle configuration, la base mobile est contrainte de participer directement à la tâche de manipulation potentielle, ce qui demande une précision que la base mobile ne peut pas toujours fournir. Par conséquent, les travaux de [Bayle et al., 2001] et [Bayle et al., 2003] proposent de tenir compte à la fois de la manipulabilité du bras et de celle du système entier en définissant une manipulabilité w comme moyenne pondérée de ces deux mesures et en introduisant un facteur de pondération $\alpha_w \in [0, 1]$:

$$w = \alpha_w w_a + (1 - \alpha_w) w_{mm} \quad (4.3.0.8)$$

L’aspect de collaboration entre l’homme et le robot mentionné plus haut, est un grand enjeu actuel du domaine de la robotique et plus particulièrement des manipulateurs mobiles qui sont souvent amenés à travailler en présence d’humains. Plusieurs résultats indiquent que cette collaboration est améliorée lorsque le robot présente des mouvements semblables à ceux des êtres humains [Gielniak et al., 2013] [Duffy, 2003] [Fong et al., 2002] car il est alors plus simple d’interpréter et de prédire plus facilement les mouvements du robot. Ces travaux mentionnent que la prédiction des intentions motrices d’autrui pour interpréter l’objectif final est un élément fondamental de la réussite d’une action conjointe entre les êtres humains, notamment parce que des actions complémentaires peuvent plus facilement être sélectionnées. Ainsi, si les mouvements d’un robot sont plus humains, ils permettront une meilleure anticipation et prédiction des intentions, contribuant ainsi à améliorer la collaboration entre l’homme et le robot. Même dans des travaux récents [Zhao et al., 2023] [Gulletta et al., 2021], il semble qu’il n’existe pas de mesure largement acceptée pour quantifier la caractéristique humaine d’un mouvement. La plupart des travaux portant sur ce sujet ne traite que la fluidité du mouvement de l’organe terminal, mais ne considère pas la posture du robot. Les travaux qui la considèrent peuvent le faire à l’aide de mesures géométriques, d’énergie potentielle, etc. Cependant, dès les premiers travaux portant sur la manipulabilité [Yoshikawa, 1985b], l’auteur mentionne que les postures maximisant cette mesure ont tendance à ressembler à des postures humaines. En effet, les cas d’une main et d’un bras robotique sont étudiés, et les analyses des postures optimales relativement à la manipulabilité ressemblent aux postures prises par les humains. Depuis, la manipulabilité a grandement été utilisée à cet effet. [Fard, 2019] analyse la préférence des postures de marche relativement à des mesures de manipulabilité. Ces observations sont confirmées dans d’autres domaines, analysant plus particulièrement des tâches de préhension et de positionnement. De plus, les travaux de [Jaquier et al., 2020] montrent que l’analyse de la forme de l’ellipsoïde de manipulabilité traduit plus d’information sur la posture et le mouvement humain que les mesures classiques. En effet, ces travaux ont montré que la posture humaine et son ellipsoïde de manipulabilité s’adaptent à la tâche à réaliser pour l’effectuer de cette manière. Par exemple, lors du transport d’un objet lourd, l’axe majeur de l’ellipsoïde de manipulabilité s’aligne sur l’axe vertical pour mieux contrer la gravité tout en minimisant les couples articulaires. Encore selon [Jaquier et al., 2020], l’analyse de la forme de l’ellipsoïde de manipulabilité révèle également des informations sur une tâche future.

Pour conclure, les récents travaux mettent en avant des mesures permettant de quantifier la posture et le trait humain d'un mouvement. Il semble également que la manipulabilité soit un indice clef dans cette recherche de quantification, permettant de répondre à certains enjeux liés à collaboration entre l'homme et le robot. Comme souvent, la biologie fait bien les choses et les mouvements aux traits humains présentent de nombreuses caractéristiques optimales. Maximiser la manipulabilité, tout comme imiter les mouvements humains, permet de s'éloigner des singularités et de maximiser le rapport entre la force générée et la pénibilité articulaire. Il paraît donc judicieux d'intégrer la manipulabilité dans notre contrôleur. Il convient également de mener une analyse sur l'utilisation complémentaire de la manipulabilité du bras et du système entier.

4.4 Synthèse du contrôleur

4.4.1 Présentation du problème

Le problème de contrôle optimal considéré ici reste très similaire à celui présenté dans le chapitre précédent. À chaque itération de la boucle de contrôle, on cherche une séquence de commandes optimale $U^*(\cdot)$ qui minimise une fonction de coût J_{N_p} sur un horizon de N_p prédictions sous un ensemble de contraintes $C(U(\cdot))$. Premièrement, la composition de cette séquence de commande est différente puisque la tête du robot est également commandée. Ainsi, le nouveau vecteur de commande est $u_{tot} = [u_b^T, u_a^T, u_h^T]^T$ et la séquence de commande $U(\cdot)$ est alors définie comme suit :

$$U_k(\cdot) = [u_{tot}(k), \dots, u_{tot}(k + N_c - 1)] \quad (4.4.1.1)$$

De plus, la fonction de coût est, elle aussi, modifiée pour intégrer la manipulabilité dans les générations de trajectoire. À une itération k , le problème de minimisation est alors défini comme ceci :

$$U_k^*(\cdot) = \min_{U_k(\cdot)} J(k, U_k(\cdot)) \quad (4.4.1.2)$$

avec

$$J(k, U_k(\cdot)) = J_{N_p}(S(k), Z_c(k), \chi_a(k), U_k(\cdot)) = \sum_{p=k+1}^{k+N_p} F(p) \quad (4.4.1.3)$$

et

$$F(p) = F_{vs}(p) + K_w F_w(p) \quad (4.4.1.4)$$

sous les contraintes suivantes :

$$\hat{S}(k) = S(k) \quad (4.4.1.5a)$$

$$\hat{Z}_c(\hat{S}(k)) = Z_c(S(k)) \quad (4.4.1.5b)$$

$$\hat{\chi}_a(k) = \chi_a(k) \quad (4.4.1.5c)$$

$$\hat{S}(p+1) = f_S(\hat{S}(p), U(p)) \quad (4.4.1.5d)$$

$$\hat{Z}_c(p+1) = f_{Z_c}(\hat{Z}_c(p), U(p)) \quad (4.4.1.5e)$$

$$\hat{\chi}_a(p+1) = f_{\chi_a}(\hat{\chi}_a(p), U(p)) \quad (4.4.1.5f)$$

$$C(U^*(\cdot)) \leq 0 \quad (4.4.1.5g)$$

Dans 4.4.1.4, le paramètre K_w permet d'équilibrer les tâches entre elles. Il est important de souligner que l'expression de cette nouvelle fonction de coût déforme l'espace de recherche du problème. L'introduction d'une deuxième tâche ne permet plus de garantir la réalisation de la tâche de positionnement. Une contribution majeure de ce travail a été la définition d'un ensemble de trois contraintes de positionnement variant dans le temps permettant de résoudre les difficultés mentionnées. La stabilité est à nouveau assurée malgré les potentielles solutions sous-optimales calculées par le solveur. De même, le compromis entre les deux tâches inhérent à la formulation du problème est évité. Ce triptyque de contraintes est présenté dans la section 4.4.5.

4.4.2 Définition du coût de la tâche visuelle F_{vs}

Le coût F_{vs} traduit la tâche principale d'asservissement visuel de la même manière qu'au chapitre précédent :

$$F_{vs} = [\hat{S} - S^*]^T Q_S [\hat{S} - S^*] \quad (4.4.2.1)$$

Cependant, le vecteur d'indice visuel S dans cette équation dépend de la méthode de projection utilisée. La méthode de projection sphérique est utilisée lorsque la caméra de l'organe terminal est mal orientée et que des problèmes de projection perspective peuvent être rencontrés. La projection perspective peut ensuite être réutilisée lorsque ces problèmes ne sont plus présents pour que les informations utilisées ne dépendent pas d'une étape de reconstruction 3D. Ainsi, le coût F_{vs} peut cette fois être défini de deux manières différentes :

1. À partir de la projection sphérique :

$$F_{vs} = [\hat{S}_s^m - S_s^{m*}]^T Q_{S_s^m} [\hat{S}_s^m - S_s^{m*}] \quad (4.4.2.2)$$

2. Ou à partir de la projection perspective, comme dans le chapitre précédent :

$$F_{vs} = [\hat{S}_p^m - S_p^{m*}]^T Q_{S_p^m} [\hat{S}_p^m - S_p^{m*}] \quad (4.4.2.3)$$

Deux schémas de contrôle sont présentés par la suite. Un premier dont l'objectif est de commuter vers la projection perspective lorsque les soucis de projection ne peuvent plus être rencontrés. Les informations 3D ne sont alors utilisées dans la fonction de coût que pendant le régime transitoire de l'asservissement. La phase de convergence ne dépend pas d'une reconstruction 3D et un positionnement final précis est possible avec les informations 2D. Pour ces raisons, nous avons décidé d'appeler ce premier schéma : stratégie d'asservissement visuel prédictif 2D. Cette approche nécessite une stratégie de sélection de la méthode de prédiction permettant de réaliser la commutation de méthode de projection. Une contribution de ces travaux de thèse est d'utiliser la structure du problème de contrôle optimal pour réussir à l'effectuer en douceur.

Le deuxième schéma de contrôle conserve quant à lui la méthode de projection sphérique durant toute la trajectoire. Contrairement au premier, les informations 3D sont également utilisées dans le régime permanent de l'asservissement. L'impact de l'étape de reconstruction 3D sur le positionnement final de l'organe terminal est analysé. Ainsi, nous avons décidé de désigner ce deuxième schéma par l'appellation : stratégie d'asservissement visuel prédictif 2D 1/2.

4.4.3 Définition du coût de la tâche de maximisation de manipulabilité F_w

Premièrement, il convient de justifier le choix d'intégrer la tâche de maximisation de la manipulabilité dans la fonction de coût. De fait, une fonction de coût à plusieurs termes engendre bien souvent des complications : l'espace de recherche se complexifie et des minima locaux sont généralement créés. Ici, en l'occurrence, la résolution peut converger vers un compromis entre les deux tâches. Or, il y a bien une hiérarchisation de ces dernières et la convergence vers un minimum local auquel la tâche visuelle n'est pas entièrement réalisée.

Il serait tentant donc de placer cette tâche secondaire dans les contraintes du problème d'optimisation. Certains travaux imposent ainsi une manipulabilité minimale à respecter, mais cette méthode présente deux inconvénients majeurs : le premier étant que si le seuil en question est trop haut, trop de mouvements sont bloqués et le manipulateur mobile perd de la liberté de mouvement. Ensuite, si ce seuil est trop bas, la manipulabilité n'est pas maximisée et l'effet de la contrainte n'est plus visible. En fait, les contraintes sont plutôt cohérentes pour les tâches prioritaires du plus haut niveau comme les contraintes de visibilité ou d'évitement d'obstacle. La manipulabilité doit donc être intégrée dans la fonction de coût au travers de F_w . Cependant, contrairement à la tâche d'asservissement visuel, celle-ci ne doit pas nécessairement converger vers un état précis, l'objectif étant seulement de maximiser autant que possible cette mesure tout en accomplissant la tâche principale. Ainsi, la forme de F_w choisie est analogue à la mesure 4.3.0.8, mais dans le but d'effectuer une minimisation :

$$F_w = \alpha_w \frac{1}{w'_a} + (1 - \alpha_w) \frac{1}{w'_{mm}} \quad (4.4.3.1)$$

où l'indice de manipulabilité w'_{mm} est défini de manière analogue à w'_a , i.e. $w'_{mm} = Pw_{mm}^2$. Les indices de manipulabilité utilisés dans le schéma de contrôle correspondent donc à la fois aux singularités cinématiques et aux limites des butées articulaires comme expliqué au chapitre précédent. À nouveau, la distinction entre intégration par contrainte et par fonction de coût est importante. En effet, la plupart des solveurs utilisent une gestion de contrainte par principe d'activité. C'est-à-dire que la contrainte correspondante est activée uniquement lorsqu'il y a proximité avec les configurations en question. Ce comportement peut générer des instabilités si le robot se heurte à un minimum local et n'arrive plus à en sortir. L'intégration des limites de butée articulaire permet donc de diriger le robot vers des configurations éloignées de ces limites. Encore une fois, il est donc possible de définir le problème de plusieurs manières selon les gains mis en place. Le tableau 4.1 résume à nouveau les avantages et les inconvénients de chaque configuration.

	$K_w = 0$	$\alpha_w = 1$	$\alpha_w = 0$	$\alpha_w \in]0, 1[$
Posture du bras	- - -	+++	-	++
Posture du système	- - -	+	+++	++

TABLE 4.1 – Comparatif des coûts de manipulabilité

4.4.4 Définition des modèles de prédiction

Premièrement, un modèle de prédiction associé à la caméra de la tête est maintenant nécessaire puisque des contraintes, par exemple de visibilité, peuvent être définies dans cet espace image. Conceptuellement, il n'y a pas de différence entre les deux modèles. La seule différence est dans l'application puisque la transformation entre les repères de la base mobile et de la caméra correspond soit à ${}^bH_{ce}$ pour la caméra de l'organe terminal, soit à ${}^bH_{ch}$ pour la caméra de la tête.

1. Modèle de prédiction de S_s^p

Les modèles de prédiction relatifs à la projection sphérique s'obtiennent d'une manière analogue aux modèles relatifs à la projection perspective. L'idée principale est de prédire les coordonnées homogènes d'un point de l'image sphérique $\bar{\mathbf{x}}_s$ d'une prédiction à une autre. La seule différence est la matrice de projection utilisée :

$$\bar{\mathbf{x}}_s(p+1) = {}^{s(p+1)}H_{c(p+1)} {}^{c(p+1)}H_{b(p+1)} {}^{b(p+1)}H_{b(p)} {}^{b(p)}H_{c(p)} {}^{c(p)}H_{s(p)} \bar{\mathbf{x}}_s(p) \quad (4.4.4.1)$$

Pour rappel, $\bar{\mathbf{x}}_s$ désigne le point dans l'image sphérique en coordonnées homogènes et une matrice AH_B correspond à la matrice de transformation homogène d'un repère B à un repère A . Et en ne prenant en compte que les trois premières coordonnées homogènes du point, le modèle de prédiction est défini par :

$$S_s^p(p+1) = f_{S_s^p}(S_s^p(p)) \quad (4.4.4.2)$$

Par ailleurs, obtenir les coordonnées sphériques d'un point \mathbf{x}_s à partir d'un point en coordonnée perspective \mathbf{x}_i est simple. Il suffit de passer par le repère caméra et de projeter le point dans l'image correspondante :

$$\bar{\mathbf{x}}_s = {}^s H_c {}^c H_i \bar{\mathbf{x}}_i \quad (4.4.4.3)$$

2. Modèle de prédiction de S_s^m En définissant par g_s la relation permettant de calculer les indices visuels basés sur les moments 3D telle que $S_s^m = g_s(S_s^p)$, le modèle de prédiction devient :

$$S_s^m(p+1) = g_s \circ f_{S_s^p}(S_s^p(p)) \quad (4.4.4.4)$$

3. Modèle de reprojection

Comme expliqué précédemment, deux caméras sont montées respectivement sur l'organe terminal et la tête du robot. L'objectif est de positionner la première caméra par rapport à la cible. Le vecteur de caractéristiques visuelles basé sur la projection sphérique a été introduit dans ce but. Cependant, la caméra de l'effecteur terminal n'est pas omnidirectionnelle. Des problèmes de visibilité peuvent donc survenir. Pour éviter ces problèmes et permettre des mouvements avec un bras replié, la stratégie proposée est de projeter les informations visuelles fournies par la caméra de la tête dans le repère de la caméra de l'organe terminal. Cette reprojection est effectuée grâce à la matrice de transformation homogène ${}^{c_e} H_{c_h}$ qui dépend de χ_a et χ_b . Par conséquent, dans les cas où la caméra de l'effecteur est incapable de détecter la cible, il est toujours possible de déterminer les indices visuels grâce à la caméra de la tête. La loi de commande est donc alimentée en indices visuels, soit directement à partir des informations de la caméra de l'organe terminal, soit recalculées à partir des données fournies par le système de vision de la tête en utilisant le modèle de reprojection. Grâce à cette approche, les problèmes de visibilité peuvent être surmontés, et une plus grande gamme de mouvements peut être envisagée.

4.4.5 Ensemble de contraintes de positionnement variant dans le temps

Cet ensemble de trois contraintes est introduit pour plusieurs raisons. Premièrement, l'ajout d'une deuxième tâche déforme l'espace de recherche et la fonction de coût totale ne présente plus les propriétés d'unicité du point d'équilibre à la situation de référence. Sans modification du problème d'optimisation, la résolution aboutirait en un compromis entre les deux tâches. Une méthode permettant d'intégrer la priorité de la tâche visuelle est nécessaire. Deuxièmement, la difficulté de la lente convergence observée pour le premier schéma due à la sous-optimalité de la solution peut s'aggraver. Pour rappel, au voisinage de la situation de référence, la

solution optimale n'utilise théoriquement que les premières prédictions et les commandes relatives aux dernières doivent être nulles. Cependant, cette optimum n'est jamais trouvé compte tenu des conditions de résolution. Il faut donc aider la résolution en guidant le solveur pour l'atteindre.

Conceptuellement, ces deux enjeux se rejoignent : dans les deux cas, la solution est sous-optimale vis-à-vis de la tâche d'asservissement visuel. L'idée de ce triptyque de contraintes est d'utiliser la contrainte terminale pour guider la résolution. Cette dernière s'accompagne d'une contrainte de contraction de la trajectoire chargée de la faire diminuer. La trajectoire est alors forcée à réduire et à se rapprocher petit à petit vers la situation de référence. La contrainte terminale adaptée a donc une utilité double : elle permet de garantir la stabilité tout en guidant le solveur. Le compromis entre les deux tâches et les problèmes de minima locaux sont alors évités.

La relaxation des dernières commandes déjà intégrée au chapitre précédent vient compléter le triptyque. En effet, la faisabilité du problème doit toujours être assurée et cette relaxation est nécessaire dans le contexte de manipulation mobile étudié.

Pour résumer, l'ensemble de contraintes de positionnement permet donc de répondre à plusieurs enjeux :

- Garantir la faisabilité et la stabilité en boucle fermée.
- Garantir la convergence de la tâche d'asservissement visuel et donc éviter qu'un compromis entre les deux tâches ne soit réalisé.
- Empêcher que le système se bloque dans des minima locaux à cause de la sous-optimalité de la solution.

Ces trois principaux enjeux sont inter-corrélés et renvoient aux démonstrations théoriques basées sur l'évolution de la fonction de valeur $V(k) = J(k, U_k^*(\cdot))$. En effet, la nouvelle fonction de valeur du schéma de contrôle n'est plus Lyapunov puisqu'elle ne remplit plus aucune des conditions 3.3.7.2. La situation d'équilibre n'est par exemple plus unique et la décroissance de la fonction de valeur n'est plus assurée puisque la tâche de maximisation de la manipulabilité n'est pas sous la forme d'une régulation.

Il faut premièrement corriger les deux premières conditions en question en précisant que la fonction de valeur qui nous intéresse maintenant n'est plus la fonction de valeur de l'ensemble de la fonction de coût, mais bien uniquement celle de la tâche visuelle. Celle-ci est alors définie par :

$$V_{vs}(k) = \sum_{p=k+1}^{k+N_p} F_{vs}(p) \quad (4.4.5.1)$$

Cette nouvelle fonction de valeur respecte alors naturellement les deux premières conditions 3.3.7.2a et 3.3.7.2b. Cependant, rien ne garantit la troisième 3.3.7.2c, i.e. la décroissance de celle-ci. L'idée permettant de pallier ce problème provient essentiellement de la théorie de contrôle sous-optimal et plus précisément de son extension au MPC réalisée dans [Scokaert et al., 1999]. Ces travaux présentent des résultats similaires au théorème de stabilité de Lyapunov dans un cadre d'optimisation sous-optimale. Il peut être noté que l'ajout de la tâche de maximisation de la

manipulabilité peut également être vue comme une source de sous-optimalité supplémentaire. Le point clef émergeant de ces travaux est que la faisabilité implique la stabilité en modifiant le problème d'optimisation. La proposition de ces travaux peut être résumée à l'ajout de la troisième condition de Lyapunov 3.3.7.2c dans les contraintes du problème. Appliqué à notre cas, il s'agit alors simplement d'ajouter la contrainte suivante :

$$V_{vs}(k+1) < V_{vs}(k) \quad (4.4.5.2)$$

Malheureusement, intégrer cette contrainte directement dans le problème d'optimisation ne permet pas une résolution satisfaisante dans les contraintes de temps accordées au solveur. Des essais ont été menés pour tester cette approche, mais ils se terminaient généralement par un échec de la loi de commande. En effet, le problème est complexe et la limite du temps de résolution rend le respect des contraintes difficile. La relaxation des contraintes des actionneurs déséquilibre particulièrement l'impact des différentes variables d'optimisation. Cet aspect constitue un défi pour les méthodes de résolution numérique et une méthode moins directe est donc proposée dans ces travaux. Comme expliqué précédemment, celle-ci s'appuie sur le concept de la contrainte terminale qui fournit une situation de référence pour la tâche visuelle. L'ensemble de contraintes de positionnement permet ensuite de s'approcher itérativement de cette référence jusqu'à ce que tout l'horizon de prédiction soit contraint de respecter la contrainte terminale.

Pour ce faire, trois contraintes sont définies : la contrainte d'égalité entre prédiction et référence, les contraintes des vitesses des actionneurs relâchées et la contrainte de contraction de la trajectoire. Celles-ci sont introduites dans les sections suivantes et le processus global y est expliqué.

Contrainte d'égalité entre prédiction et référence

La stabilité en boucle fermée est à nouveau assurée en s'appuyant sur le concept de la contrainte terminale. Cette dernière impose que le dernier état prédit soit égal à celui désiré, forçant ainsi la trajectoire calculée à atteindre l'objectif. Dans ce schéma de contrôle, une version modifiée est proposée, à savoir la contrainte d'égalité entre la prédiction et la référence. Elle impose que certains indices visuels prédits soient égaux à ceux de référence. Cette condition est donc exprimée comme suit :

$$\hat{S}(p + I_{TC}) = S^* \quad (4.4.5.3)$$

où $I_{TC} \in [1, N_p]$ correspond à l'index de la prédiction contrainte. Son rôle et son évolution sont expliqués plus loin.

Contraintes des vitesses des actionneurs

À nouveau, pour que la contrainte d'égalité entre prédiction et référence (adaptation de la contrainte terminale) mentionnée ci-dessus puisse être respectée, il est nécessaire de fournir un horizon de prédiction suffisamment long. Pour ce faire,

les contraintes de vitesse des dernières commandes peuvent être relâchées. Cette approche conduit à l'ensemble de contraintes suivant pour les vitesses du manipulateur mobile :

$$\begin{aligned} \begin{bmatrix} u_{mm}(p) - u_{u|t} \\ u_{l|t} - u_{mm}(p) \end{bmatrix} &\leq 0, \forall p \in \llbracket k, k + N_c - N_r - 1 \rrbracket \\ \begin{bmatrix} u_{mm}(p) - u_{u|r} \\ u_{l|r} - u_{mm}(p) \end{bmatrix} &\leq 0, \forall p \in \llbracket k + N_c - N_r, k + N_c - 1 \rrbracket \end{aligned} \quad (4.4.5.4)$$

N_r est le nombre de prédictions réalisées avec des contraintes relâchées sur les vitesses, $u_{l|t}$ et $u_{u|t}$ sont respectivement les bornes inférieure et supérieure strictes correspondant aux limites réelles de l'actionneur, et $u_{l|r}$ et $u_{u|r}$ sont respectivement les bornes inférieure et supérieure relâchées.

Contrainte de contraction de la trajectoire

Enfin, une dernière contrainte est ajoutée pour contracter la trajectoire. Celle-ci impose la diminution de la transformation entre deux poses de caméra prédites successives données. Deux approches ont été étudiées dans ces travaux pour implémenter cette idée. La première est de définir la contrainte dans l'espace articulaire est d'imposer une diminution de la commande liant deux prédictions. La contrainte est alors définie de la sorte :

$$\|u_{tot}(k + I_{TC})\| < \min_{cmd} - \delta_{min}^{cmd} \quad (4.4.5.5)$$

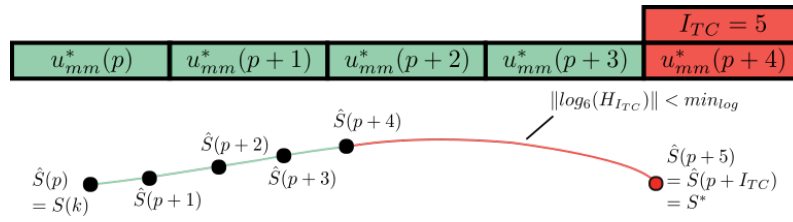
où \min_{cmd} représente la plus petite valeur de $\|u_{tot}(k + I_{TC})\|$ observée jusqu'à l'instant courant. De manière similaire à l'approche présentée dans [Scolaert et al., 1999], un terme δ_{min}^{cmd} est introduit pour forcer une diminution minimale. Il doit être suffisamment grand pour accélérer la convergence, mais suffisamment petit pour ne pas perturber les tâches.

La deuxième approche considère plutôt la contrainte dans l'espace opérationnel. Pour cela, nous définissons d'abord $H_{I_{TC}}$ comme la matrice de transformation entre la pose à l'instant prédit $p + I_{TC} - 1$ et celle à $p + I_{TC}$. Ensuite, nous nous appuyons sur l'outil logarithmique \log_6 qui permet de convertir un élément H du groupe de Lie $SE(3)$ en l'élément correspondant ν de son algèbre de Lie $se(3)$ [Sola et al., 2018].

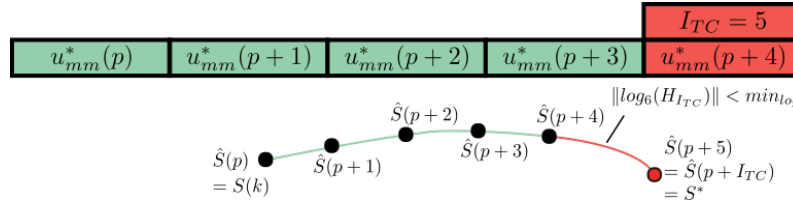
$$\nu = \log_6(H) \quad (4.4.5.6)$$

où $H = H_{I_{TC}}$ est utilisé sous sa forme de matrice de transformation homogène et ν sous sa forme de vecteur de vitesse à six dimensions. Cet outil de l'algèbre de Lie a déjà été utilisé dans le premier chapitre pour analyser le positionnement final de l'organe terminal. Ainsi, la contrainte peut être écrite comme suit :

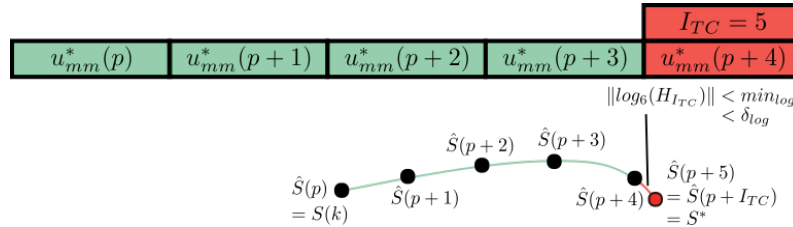
$$\|\log_6(H_{I_{TC}})\| < \min_{log} - \delta_{min}^{log} \quad (4.4.5.7)$$



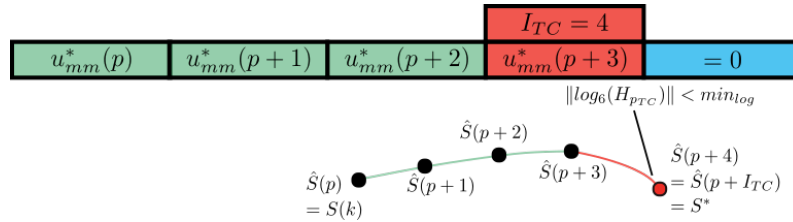
(a) Itération $k = k_0 = 0$



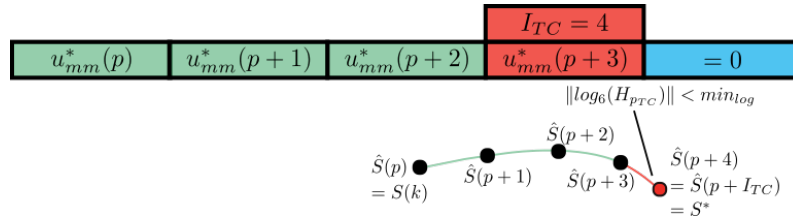
(b) Itération $k = k_1 > k_0$



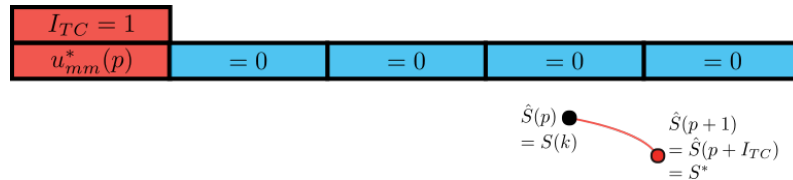
(c) Itération $k = k_2 > k_1$



(d) Itération $k = k_3 = k_2 + 1$



(e) Itération $k = k_4 > k_3$



(f) Itération $k = k_5 > k_4$

FIGURE 4.3 – Évolution de l'ensemble de contraintes variable dans le temps

où $H_{I_{TC}} = H(k + I_{TC} - 1)$, et min_{log} représente la plus petite valeur de $\|\log_6(H_{I_{TC}})\|$ observée jusqu'à l'instant courant. δ_{min}^{log} est à nouveau introduit pour forcer une diminution minimale.

Les deux approches ont été testées, et des résultats comparatifs sont présentés. L'approche finalement retenue après analyse des performances est la deuxième. En effet, une mesure dans l'espace opérationnel est plus adaptée à la quantification d'une contraction de la trajectoire cartésienne qu'une mesure dans l'espace articulaire. La contraction se fait alors plus rapidement avec la fonction log_6 . C'est pourquoi la suite des explications du fonctionnement du triptyque s'appuie uniquement sur celle-ci.

Évolution de l'ensemble de contrainte de positionnement variant dans le temps

Les trois contraintes présentées visent à garantir qu'une trajectoire plus courte atteignant la situation désirée soit calculée à chaque nouvelle itération. En d'autres termes, la fonction de valeur relative à la tâche d'asservissement visuel V_{vs} est diminuée en contractant la trajectoire.

Pour ce faire, nous utilisons l'index I_{TC} comme suit. Tout d'abord, la contrainte d'égalité entre référence et prédiction est fixée sur la dernière prédiction, et celle de contraction de la trajectoire est sur le dernier morceau de la trajectoire, c'est-à-dire $I_{TC} = N_p$. Ainsi, la trajectoire prédite atteint l'objectif, et la longueur du dernier morceau est forcée à diminuer à chaque itération. Lorsque ce dernier morceau de trajectoire n'est plus nécessaire, c'est-à-dire nul, les contraintes sont déplacées sur les prédictions précédentes, c'est-à-dire que I_{TC} est décrémenté de 1. Le processus est répété jusqu'à ce que $I_{TC} = 1$, ce qui signifie que la trajectoire prédite est composée de la seule commande $u^*(p)$ et d'une seule prédiction devant respecter la contrainte d'égalité $\hat{S}(p+1) = S^*$. La commande appliquée au robot permet alors d'amener le robot à la situation de référence et la tâche de positionnement est réalisée.

Ce processus est illustré dans la figure 4.3 où $N_p = N_c = 5$. L'itération initiale est définie par $k = k_0 = 0$, où I_{TC} est initialisé à N_p . Ainsi, la contrainte d'égalité (4.4.5.3) entre $\hat{S}(p+5)$ et S^* est respectée, et la trajectoire atteint l'état désiré (voir Fig. 4.3a). Cette contrainte peut être satisfaite à partir de l'état initial grâce à la contrainte de vitesse relaxée (4.4.5.4). Lors des itérations suivantes, par exemple pour l'itération $k_1 > k_0$ sur la figure 4.3b, le morceau de trajectoire entre $\hat{S}(p+4)$ et $\hat{S}(p+5)$ est forcé à diminuer grâce à la contrainte de contraction (4.4.5.7). Une fois que le logarithme de la transformation entre $\hat{S}(p+4)$ et $\hat{S}(p+5)$ devient plus petit que le seuil δ_{log} , c'est-à-dire à l'itération $k_2 > k_1$ sur la figure 4.3c, les prédictions $(p+4)$ et $(p+5)$ sont suffisamment proches pour être fusionnés. Dès lors, l'ensemble actuel de contraintes n'a plus d'impact sur l'optimisation, et la configuration des contraintes doit alors être mise à jour en appliquant $I_{TC} = I_{TC} - 1 = 4$, c'est-à-dire l'itération $k_3 = k_2 + 1$ dans la figure 4.3d. À partir de là, la contrainte entre prédiction et référence force $\hat{S}(p+4) = S^*$, et la contrainte de contraction agit sur $\hat{S}(p+3)$ et $\hat{S}(p+4)$ et force ce morceau de trajectoire à diminuer, c'est-à-dire l'itération $k_4 > k_3$ dans la figure 4.3e. Ce processus est répété jusqu'à ce que $I_{TC} = 1$

afin que la commande appliquée au robot lui permette effectivement d'atteindre la pose désirée, c'est-à-dire l'itération $k_5 > k_4$ dans la figure 4.3f.

4.4.6 Autres contraintes

Deux autres types de contraintes sont ajoutés au schéma. Elles sont globalement similaires à celles présentées dans le chapitre précédent : la cible doit en effet encore rester visible et les limites physiques des actionneurs doivent être respectées. De légères modifications sont effectuées pour utiliser judicieusement l'aspect multi-caméra. Celles-ci sont présentées dans les paragraphes suivants.

Contraintes de visibilité

Les seuls capteurs extéroceptifs utilisés sont les deux caméras du robot et il est donc à nouveau essentiel de conserver les amers visibles par l'une ou l'autre caméra du système. Afin de laisser une pleine liberté de mouvement au bras, les contraintes de visibilité sont intégrées à la caméra de la tête. En définissant les valeurs minimales $S_{c_h}^l$ et maximales $S_{c_h}^u$ des points d'intérêt dans l'image de la caméra de la tête, celles-ci sont définies comme suit :

$$\begin{cases} \hat{S}_{c_h}(p) \leq S_{c_h}^u \\ S_{c_h}^l \leq \hat{S}_{c_h}(p) \end{cases} \quad \forall p \in \llbracket k+1, k+N_p \rrbracket \quad (4.4.6.1)$$

Contraintes de butées articulaires

Il est également encore nécessaire d'intégrer les butées articulaires dans les contraintes du système malgré l'intégration de la mesure de manipulabilité dans la fonction de coût. En plus des contraintes relatives aux articulations du bras, il faut inclure les contraintes relatives aux butées de la tête. Avec χ_{al} et χ_{hl} respectivement les bornes minimales du bras et la tête, et χ_{au} et χ_{hu} respectivement leurs bornes maximales, les contraintes deviennent :

$$\begin{cases} \hat{\chi}_a(p) \leq \chi_{au} \\ \chi_{al} \leq \hat{\chi}_a(p) \\ \hat{\chi}_h(p) \leq \chi_{hu} \\ \chi_{hl} \leq \hat{\chi}_h(p) \end{cases} \quad \forall p \in \llbracket k+1, k+N_p \rrbracket \quad (4.4.6.2)$$

4.5 Schéma d'asservissement visuel prédictif 2D

Comme expliqué précédemment, la fonction de coût relative à la tâche visuelle peut être définie à partir de plusieurs vecteurs d'indices visuels. Différentes versions du schéma de contrôle sont alors imaginables et deux schémas de contrôle sont présentés dans ces travaux. Le premier est présenté dans cette section et son objectif est de ne s'appuyer sur la projection sphérique que durant le régime transitoire de

l'asservissement. Cette méthode de projection est donc utilisée pour amener l'organe terminal dans une situation convenable pour finalement commuter sur la projection perspective quand les problèmes mentionnés ne sont plus rencontrés. Les informations 3D ne sont alors utilisées que pendant cette phase et la fonction de coût de la fin de la trajectoire ne dépend pas d'une reconstruction 3D. Le régime permanent étant atteint avec un vecteur d'indices visuels 2D, ce premier schéma est qualifié d'asservissement visuel prédictif 2D. Il devient alors nécessaire d'élaborer une stratégie de sélection de la méthode de prédiction permettant de réaliser la commutation de méthode de projection. Dans ces travaux, une méthode utilisant la structure du problème d'optimisation permet de l'effectuer en douceur. Celle-ci est présentée dans les paragraphes suivants.

4.5.1 Stratégie de sélection de la méthode de prédiction

Comme mentionné précédemment, la stratégie proposée consiste à utiliser les informations de la caméra de la tête pour calculer les indices visuels lorsque le bras est replié, c'est-à-dire lorsque la caméra de l'organe terminal ne peut pas percevoir la cible. Les caractéristiques visuelles sont ensuite exprimées dans le repère de la caméra de l'organe terminal avec la projection sphérique, puis utilisées dans la fonction de coût pour amener l'organe terminal dans une situation où la cible peut être vue. Une fois cette configuration atteinte, l'image capturée par la caméra de l'effecteur terminal est utilisée et les caractéristiques visuelles sont à nouveau obtenues avec la méthode de projection perspective. Ces dernières permettent alors d'exprimer le coût à partir d'information 2D amenant l'effecteur terminal à la situation de référence sans étape de reconstruction 3D. Il est ainsi nécessaire de développer une méthode de commutation définissant la méthode de projection à utiliser pour calculer les caractéristiques visuelles prédites. Il est également nécessaire de réaliser cette transition en douceur afin d'éviter au maximum de brusques variations de la commande appliquée au robot. Pour cela, notre stratégie propose de changer la méthode de projection d'une prédiction à la fois. Toutes les prédictions sont initialement calculées grâce à la méthode de projection sphérique afin de garantir qu'aucun problème de projection ne se produise. Ensuite, en partant de la fin de l'horizon de prédiction, les méthodes de projection de chaque prédiction sont une-à-une basculées sur la méthode de projection perspective. Cette commutation est réalisée sur la base d'un critère quantifiant la bonne situation prédite du robot par rapport à la cible afin d'éviter les problèmes relatifs à la projection perspective.

Il est alors possible de définir un index I_{SM} correspondant à la dernière prédiction utilisant la projection sphérique et délimitant donc les prédictions s'appuyant sur l'une ou l'autre méthode. Ainsi :

- Si $p \leq k + I_{SM}$, alors le coût de la prédiction p s'appuie sur la projection sphérique :

$$F_{vs}(p) = \left[\hat{S}_s^m(p) - S_s^{m*}(p) \right]^T Q_{S_s^m}(p) \left[\hat{S}_s^m(p) - S_s^{p*}(p) \right] \quad (4.5.1.1)$$

- Si $p > k + I_{SM}$, alors le coût de la prédiction p s'appuie sur la projection perspective :

$$F_{vs}(p) = \left[\hat{S}_p^m(p) - S_p^{m*}(p) \right]^T Q_{S_p^m}(p) \left[\hat{S}_p^m(p) - S_p^{m*}(p) \right] \quad (4.5.1.2)$$

Cet index I_{SM} est donc initialisé à N_p , i.e. toutes les prédictions utilisent d'abord la projection sphérique. Plusieurs choix de critères pour faire évoluer I_{SM} sont envisageables :

- Un critère sur la visibilité des amers. En effet, si ces derniers sont visibles, aucun problème lié à la projection perspective ne devrait apparaître.
- Un critère sur le coût visuel de la dernière prédiction $F_{vs}(p + I_{SM})$. Ce concept peut également être appliqué spécifiquement à certains indices visuels. Il devient alors possible de définir une situation d'organe terminal limite, à laquelle le basculement est effectué.

Ces deux formes de critère ont été implémentées et testées. C'est la deuxième qui a été retenue pour plusieurs raisons : elle permet d'une part d'avoir une meilleure et plus grande influence sur la stratégie de changement de méthode de projection et d'une autre part de sélectionner des configurations limites plus adéquates et ainsi plus robustes. En effet, l'utilisation du critère de visibilité implique généralement une commutation dans une configuration par rapport à la cible encore à risque et la transition se fait alors ressentir sur la trajectoire du robot. Le critère du coût visuel permet de basculer légèrement plus tard et d'obtenir une trajectoire plus adaptée. A nouveau, ce choix de critère est essentiellement possible grâce à l'utilisation des moments dans l'image qui permettent de traduire les informations de position et d'orientation de l'organe terminal. De plus, cette configuration limite peut ainsi être choisie selon des moments de l'image spécifiques correspondant aux degrés de liberté importants, i.e. orientation de la caméra vers la cible par exemple. Dans la suite de ces travaux, un seuil δ_{switch} a été défini tel que si le coût visuel $F_{vs}(p + I_{SM})$ en devient inférieur, la méthode de projection de la prédiction $p + I_{SM}$ devient la projection perspective. Ainsi :

$$I_{SM} = I_{SM} - 1 \text{ si } F_{vs}(p + I_{SM}) < \delta_{switch} \quad (4.5.1.3)$$

4.5.2 Récapitulatif du processus

Le processus est illustré sur la figure 4.4 avec $N_p = N_c = 5$. La figure 4.4a correspond à l'itération initiale $k = k_0 = 0$, l'index I_{SM} est alors initialisé à $N_p = 5$ et toutes les prédictions utilisent la projection sphérique. Cette organisation est conservée jusqu'à une certaine itération $k_1 > k_0$ où le coût visuel de la 5^{ème} prédiction devient inférieur à δ_{switch} (figure 4.4b) : $F_{vs}(p+5) < \delta_{switch}$. La méthode de projection de cette prédiction devient alors la projection perspective (figure 4.4c) et $I_{SM} = 4$. La prochaine prédiction à basculer sur la projection perspective est la 4^{ème} lorsque $F_{vs}(p + 4) < \delta_{switch}$ (figures 4.4d et 4.4e). Ce procédé est répété jusqu'à ce que

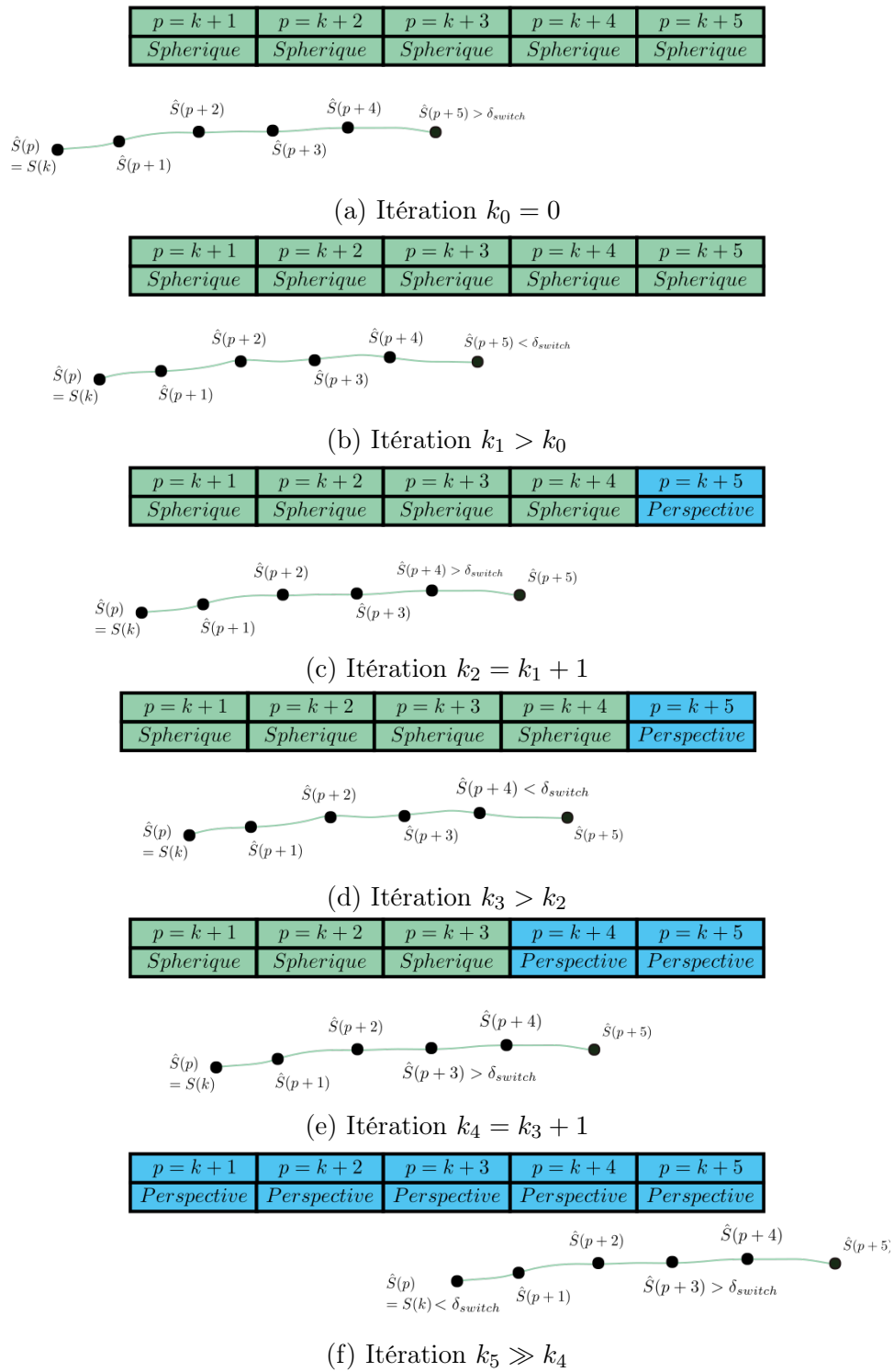


FIGURE 4.4 – Exemple simplifié du processus de commutation des méthodes de projection

$I_{SM} = 0$, i.e. toutes les prédictions sont basées sur la projection perspective et la

fonction de coût du problème de contrôle optimal est alors définie au plus proche des données capteurs pour assurer un excellent positionnement final (figure 4.4f).

4.6 Résultats obtenus avec le schéma d’asservissement visuel prédictif 2D

Ce contrôleur vise donc à étendre le travail présenté dans le premier chapitre avec une stratégie VPC multi-caméra. La tâche principale consiste toujours à positionner l’organe terminal du robot à une pose désirée définie dans l’espace image de sa caméra. L’environnement du robot est le même qu’au chapitre précédent. La différence majeure est la configuration initiale de celui-ci qui démarre maintenant avec son bras replié (voir figure 4.5a). Cette section est structurée en deux parties. Elles présentent respectivement des résultats de simulation permettant de confirmer les propositions théoriques et des résultats expérimentaux pour prouver la portabilité du schéma de commande dans un contexte réel. Le robot et la caméra de l’organe terminal doivent à nouveau parcourir environ 2m pour atteindre la situation de référence.

Tous les résultats de cette section sont obtenus avec des horizons de prédiction et de contrôle d’une taille $N_p = N_c = 10$ et seules les commandes d’une seule prédiction sont relaxées, i.e. $N_r = 1$. Le temps maximal de résolution est à nouveau fixé à 150 ms pour permettre à la boucle de contrôle de fonctionner à une fréquence de 5 Hz. La matrice de pondération est $Q_S = \text{diag}(1, 1, 1, 1, 1, 1)$ lorsque la méthode de projection est la projection sphérique, et $Q_S = \text{diag}(1, 1, 0.1, 10, 10, 1)$ avec la projection perspective. Le schéma de contrôle permet une influence pratique sur la trajectoire à l’aide de ces pondérations qui peuvent être approximatives.

De plus, seule la manipulabilité du bras est considérée pour l’obtention de ces résultats, c’est-à-dire que $\alpha_w = 1$. Il convient également de souligner que l’ensemble des résultats présentés par la suite n’auraient pas pu être obtenus sans l’approche symbolique des calculs de gradients. La complexité du problème d’optimisation et des contraintes sur le temps de résolution rendent nécessaires cette implémentation.

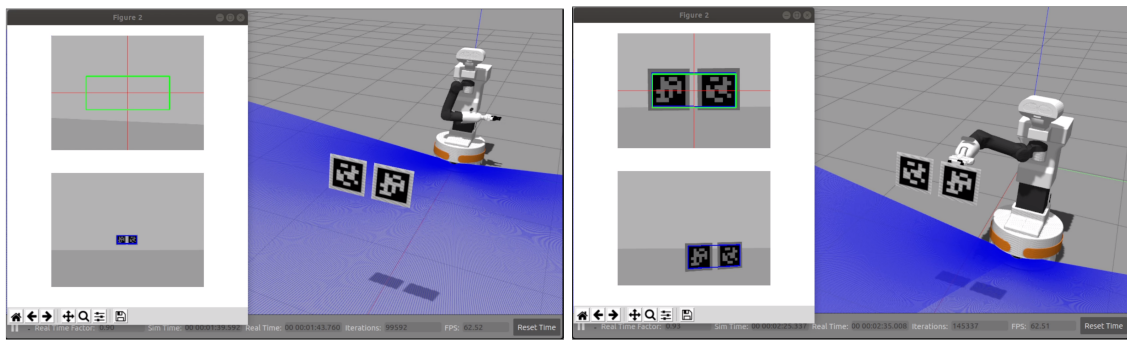
Une étude plus approfondie de l’influence du choix des paramètres de tâche de maximisation de la manipulabilité et une analyse comparative des résultats avec et sans l’utilisation des calculs de gradients symboliques seront présentées par la suite.

4.6.1 Résultats de simulation

Cette première partie de résultats a pour objectif de confirmer le bon fonctionnement du contrôleur et d’illustrer les contributions théoriques avec des résultats de simulation.

Analyse de la trajectoire du robot

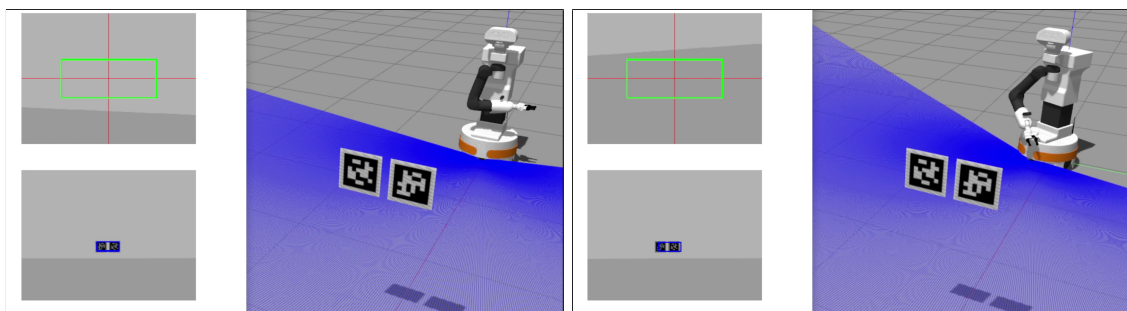
La figure 4.6 présente des captures d’écran de la trajectoire du robot et des images des deux caméras. Elle permet d’illustrer qualitativement les résultats obtenus et



(a) Configuration initiale

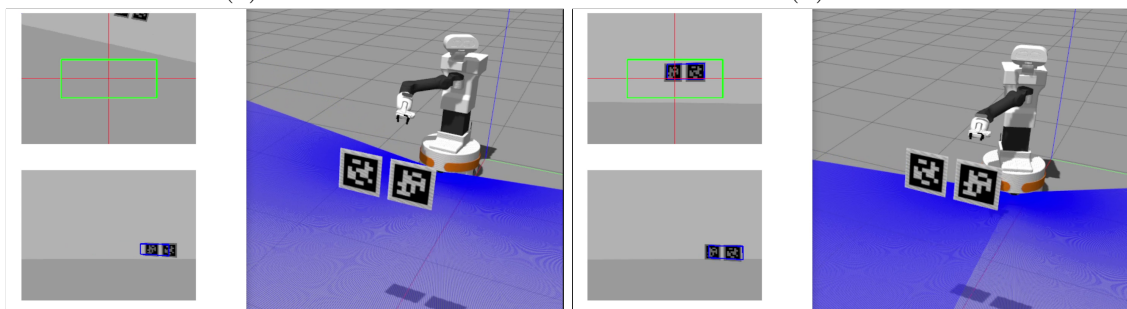
(b) Configuration finale

FIGURE 4.5 – Environnement *Gazebo*



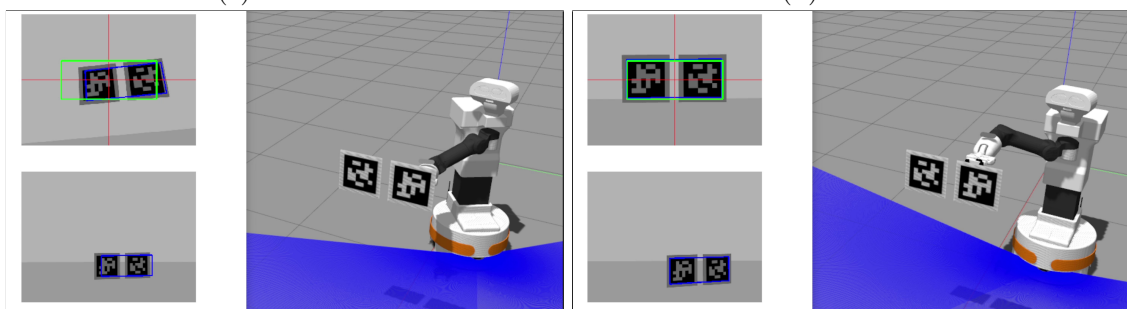
(a) $t = 0s$

(b) $t = 5s$



(c) $t = 10s$

(d) $t = 15s$



(e) $t = 25s$

(f) $t = 45s$

FIGURE 4.6 – Trajectoires du robot et des images obtenues

les deux tâches semblent correctement réalisées. En effet, le bras se déplace à des configurations éloignées des singularités et le risque de vibrations est diminué. Un compromis entre les deux tâches est également évité puisque le système converge vers la situation désirée définie par la tâche d’asservissement visuel. Les figures mettent également en évidence la capacité de coordination entre la base mobile et le bras en utilisant judicieusement la redondance du système. Enfin, ces résultats sont obtenus dans un scénario de départ pour lequel la caméra de l’organe terminal ne perçoit pas la cible, soulignant l’aptitude de la stratégie à réaliser des tâches plus variées. Une analyse plus approfondie des trajectoires obtenues est réalisée dans la section relative aux résultats expérimentaux, mais cette première évaluation met en avant la pertinence et l’efficacité de l’approche proposée.

Résultats relatifs à la tâche d’asservissement visuel

Premièrement, le processus de la sélection de méthode de projection est illustrée dans la figure 4.7. Chaque prédiction est initialement calculée en utilisant la méthode de projection sphérique (*sp*). Au fur et à mesure que les caractéristiques visuelles prédites arrivent dans une configuration adéquate pour la caméra de l’organe terminal, la méthode de projection change progressivement pour passer à la méthode de perspective (*pp*). Les différentes couleurs représentent les 10 prédictions du schéma étudié.

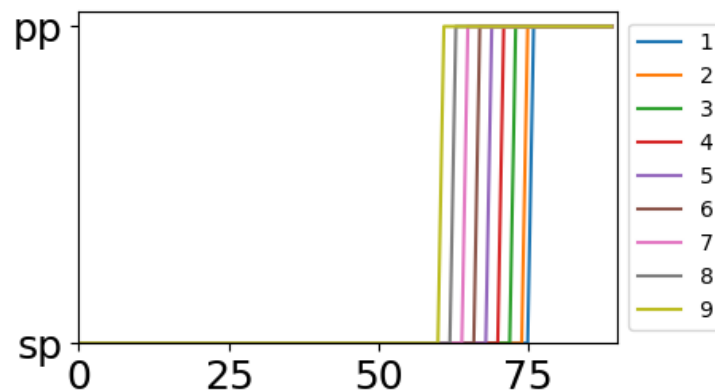


FIGURE 4.7 – Commutation de méthode de projection

Par exemple, la première transition est réalisée à la 61^{me} itération. Celle-ci correspond alors à la prédiction numéro 9, i.e. $p = k + 9$, puisque la dernière prédiction, i.e. $p = k + 10$ est la prédiction aux contraintes de vitesse des actionneurs relâchées devant respecter la contrainte terminale (ou prédiction à référence). Les figures 4.8a et 4.8b montrent alors respectivement la situation dans l’image et dans le repère global de la prédiction en question. La situation de la caméra dans le repère global confirme bien que cette première phase du contrôle amène correctement la caméra de l’effecteur en face de la cible. L’orientation de celle-ci permet alors de s’affranchir

des problèmes de la projection perspective, comme le montre également la situation de l'objet dans l'image. La transition est entièrement effectuée à partir de la 75^e itération lorsque la méthode pour la première prédiction bascule également sur la projection perspective. À partir de ce moment, il n'y a plus de risque de singularité de projection et la méthode de projection sphérique devient donc obsolète.

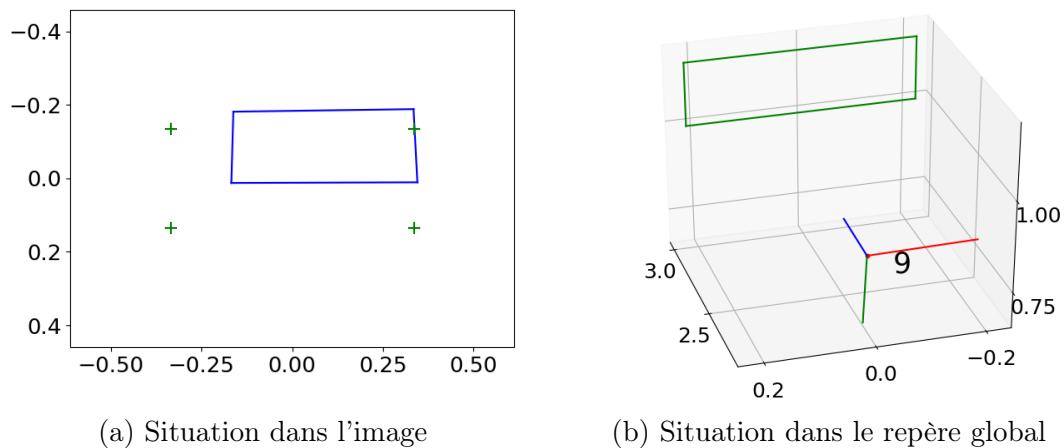


FIGURE 4.8 – Situation dans l'image et dans le repère global à l'itération de commutation de méthode de projection pour la prédiction $p = k + 9$

La commutation divise ainsi le contrôle en deux parties. Tout d'abord, la projection sphérique est utilisée sur la base des informations de la caméra de tête projetées dans le repère de la caméra de l'organe terminal pour placer le robot dans une configuration où cette dernière voit correctement la cible. Cela conduit à la deuxième partie, où la projection perspective peut être utilisée, permettant de contrôler avec précision l'organe terminal. La figure 4.9 suit cette logique et présente l'évolution de l'erreur du vecteur d'indices visuels de la première partie du contrôle (de l'itération 0 à l'itération 85) à gauche, et de la deuxième partie (de l'itération 86 à la fin du contrôle) à droite. Cette figure expose également les trajectoires des points d'intérêts dans les deux caméras : pour la caméra de la tête pendant toute la trajectoire et pour la caméra de l'effecteur uniquement lorsque toutes les prédictions sont commutées sur la projection perspective.

Sur la figure 4.9c, la caractéristique z_c se rapproche de la valeur 1, indiquant que cette partie du contrôle amène effectivement la caméra de l'organe terminal à faire face à la cible. Sur les figures 4.9b et 4.9d, on peut voir que la tâche visuelle est correctement réalisée puisque les points d'intérêt, ainsi que les moments dans l'image atteignent leurs valeurs désirées (les croix vertes pour les points d'intérêt). En effet, l'erreur entre les moments de l'image et leurs valeurs de référence s'annule.

Analyse de la tâche de maximisation de la manipulabilité

En parallèle de cette tâche de positionnement, le contrôleur doit également maximiser la manipulabilité w'_a . Les figures 4.10a et 4.10b représentent respectivement

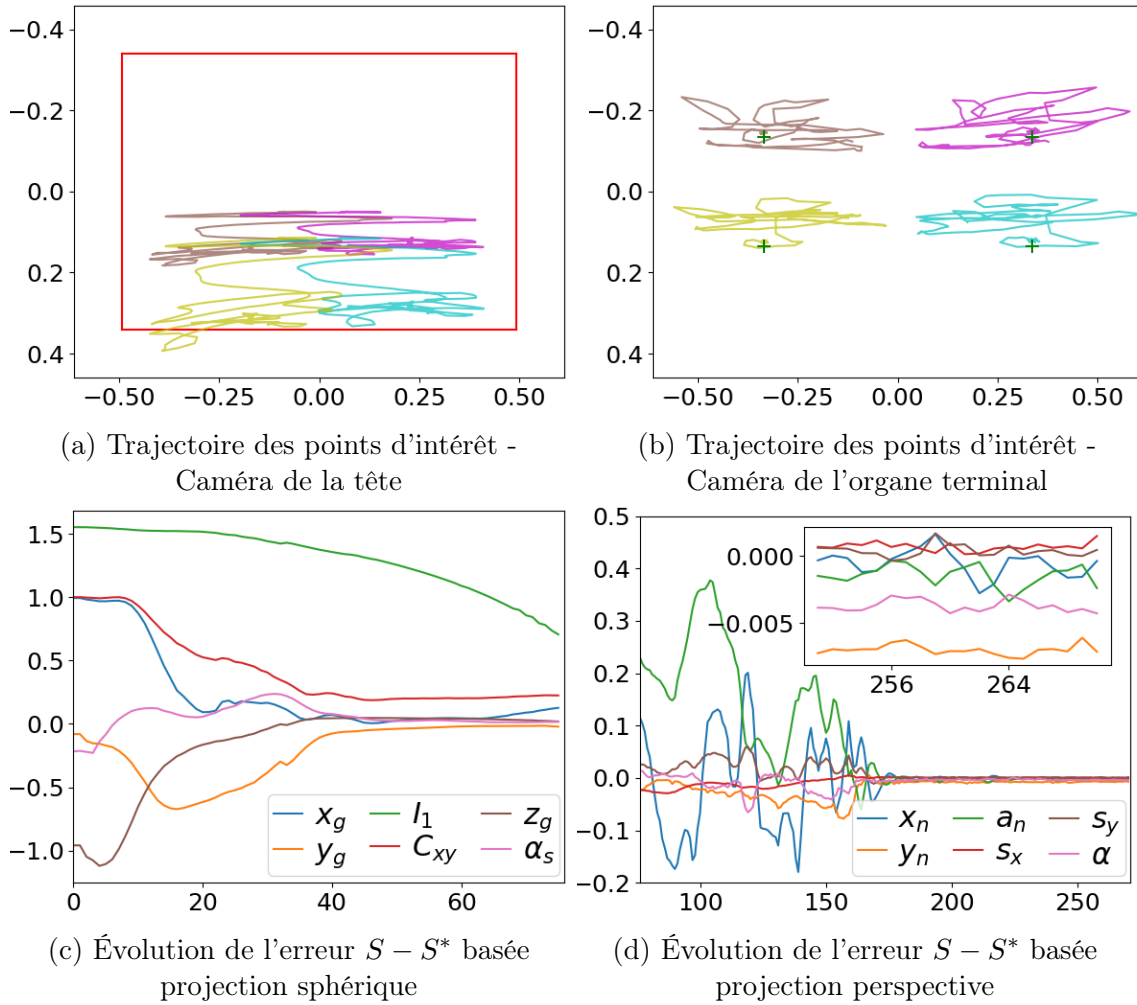


FIGURE 4.9 – Résultats relatifs à la tâche d'asservissement visuel

l'évolution de la manipulabilité w_a et de l'enveloppe des butées articulaires P avec ($K_w = 0.001$ et $\alpha_w = 1$) et sans ($K_w = 0$) la prise en compte de la tâche secondaire dans la fonction de coût. Ces figures montrent que cette tâche secondaire est également grandement réalisée : w_a est significativement plus élevée, c'est-à-dire que le robot se déplace en gardant le bras dans une configuration loin des singularités. Il termine également sa tâche de positionnement dans une configuration qui permet au robot de garder pleine liberté de mouvement pour une potentielle tâche de préhension. De plus, le robot reste très éloigné des butées articulaires, i.e. la valeur de P reste très proche de 1, contrairement au cas où cette enveloppe de limites articulaires n'est pas prise en compte et où P converge vers 0, indiquant que la configuration finale est très proche des butées.

Les figures 4.11 et 4.12 présentent les résultats obtenus sans aucune tâche secondaire à gauche et en ne prenant en compte que l'enveloppe des butées articulaires P à droite, c'est-à-dire qu'aucune mesure de distance aux singularités cinématiques n'est considérées. Elles démontrent l'utilité d'ajouter une mesure de distance aux limites articulaires dans la fonction de coût. En effet, les contraintes correspondantes

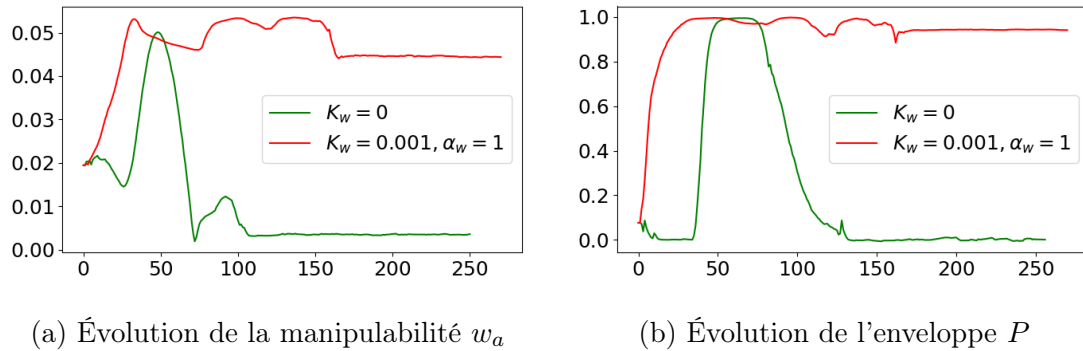
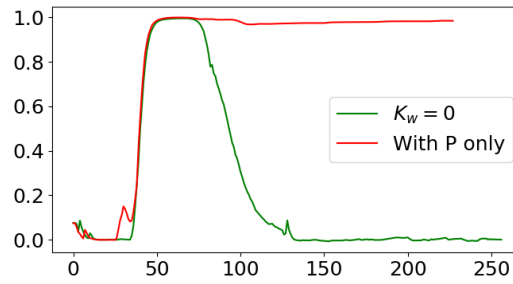
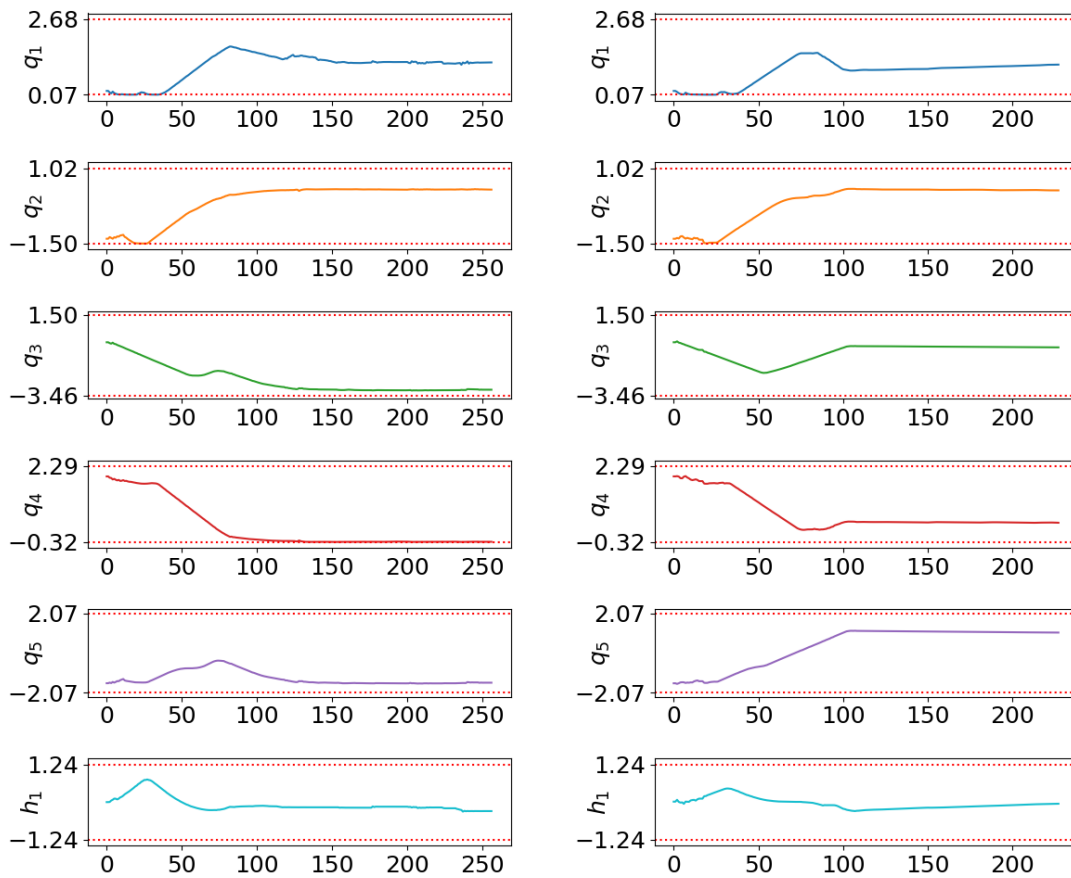


FIGURE 4.10 – Résultats relatifs à la tâche de maximisation de manipulabilité

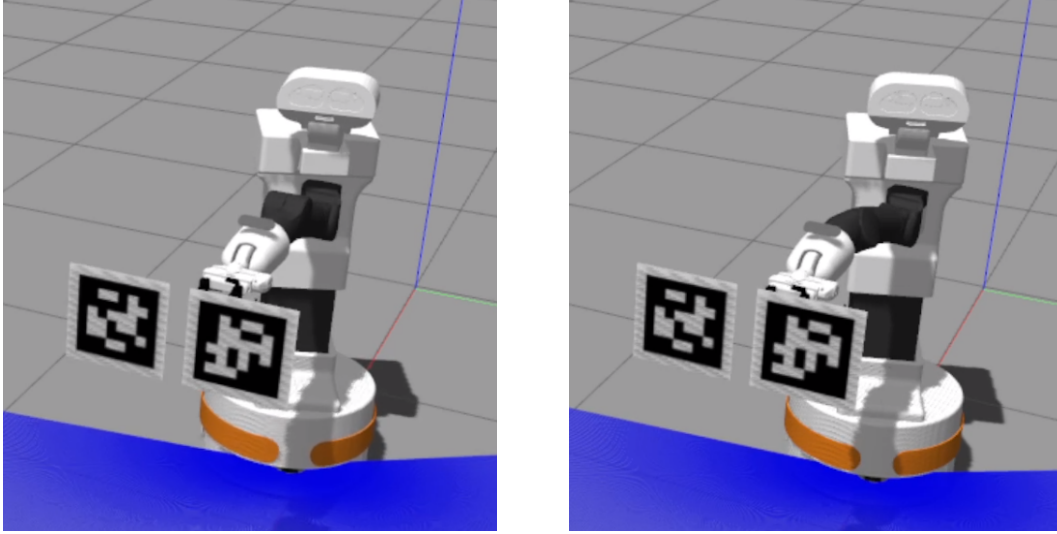
ne s'activent qu'une fois le robot bloqué dans un minimum local alors que l'enveloppe P le repousse loin de ces limites afin d'éviter d'y rentrer. La figure 4.11b confirme que le robot termine sa trajectoire avec des valeurs articulaires proches de leurs limites s'il ne tient pas compte de l'enveloppe P . En particulier ici pour les paramètres articulaires q_3 et q_4 . La mesure P permet en effet de diriger le robot lors du contrôle vers une configuration plus favorable. La figure 4.12 permet justement de comparer les deux configurations finales.

L'étude de la coordination résultante de la base mobile avec le bras et donc de la trajectoire du robot est étudiée dans la partie consacrée aux résultats expérimentaux. Comme annoncé précédemment, une analyse plus approfondie de l'impact du choix de la mesure de manipulabilité ($\alpha_w \neq 1$) est réalisée dans la section résultat du schéma d'asservissement visuel prédictif 2D 1/2.

(a) Évolution de l'enveloppe P (b) Évolution des paramètres articulaires avec $K_w = 0$ (c) Évolution des paramètres articulaires avec P uniquementFIGURE 4.11 – Comparaison des résultats avec et sans prise en compte de l'enveloppe P - Partie 1

Analyse de la convergence et de la stabilité

L'ensemble des trois contraintes de positionnement a été défini pour garantir la stabilité en boucle fermée et améliorer la convergence de la tâche visuelle tout en maximisant la manipulabilité. Cette section récapitule le fonctionnement de cet



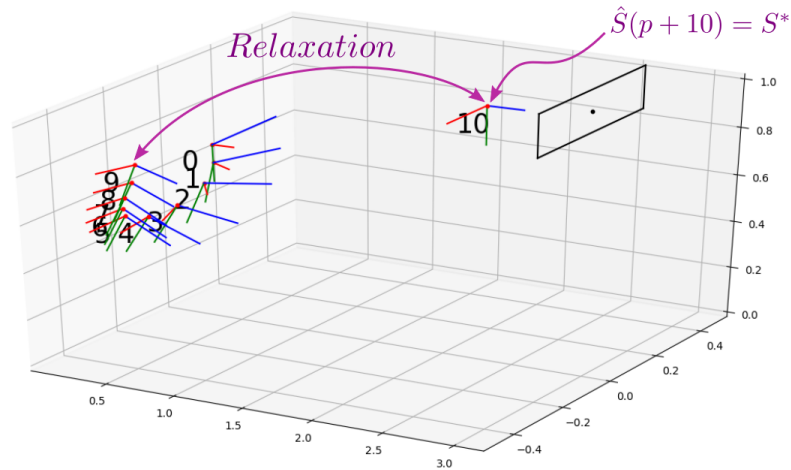
(a) Configuration finale avec $K_w = 0$ (b) Configuration finale avec P uniquement

FIGURE 4.12 – Comparaison des résultats avec et sans prise en compte de l’enveloppe P - Partie 2

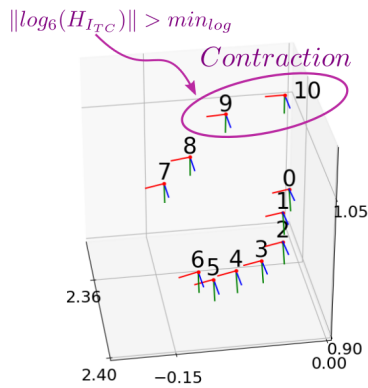
ensemble et présente les résultats qui lui sont associés. Dans un premier temps, le processus de ce triptyque de contraintes est rappelé et illustré avec des résultats de simulation. Les mesures pertinentes permettant de montrer le bon fonctionnement de ce dernier sont ensuite présentées. Pour finir, la méthode s’appuyant sur le logarithme est comparée à l’approche basée sur la commande. Cette succincte analyse permet de justifier l’utilisation de la première approche pour la suite.

Tout d’abord, rappelons le processus de l’ensemble de contraintes de positionnement en examinant la figure 4.13, qui représente le comportement simulé suivant le concept expliqué dans la figure 4.3. La prédiction contrainte I_{TC} est initialisée à $N_p = 10$. L’approche proposée permet de satisfaire la contrainte terminale dès l’itération 2 grâce au problème d’optimisation bien défini et correctement résolu. La contribution de la contrainte relaxée est mise en évidence dans la figure 4.13a. Ensuite, la contrainte (4.4.5.7) impose la transformation entre les deux dernières poses prédites à diminuer (voir figure 4.13b, 4.13c, et 4.13d) et la prédiction contrainte est maintenue à sa valeur actuelle jusqu’à ces prédictions, c’est-à-dire la 9^{me} et la 10^{me}, sont suffisamment proches pour être fusionnées. Lorsque le logarithme devient plus petit que le seuil δ_{log} (voir figure 4.13d), l’ensemble de contraintes actuel n’influence plus l’optimisation, et la configuration de la contrainte est mise à jour en décalant la contrainte terminale, c’est-à-dire $I_{TC} = 9$, comme vu sur la Fig. 4.13e.

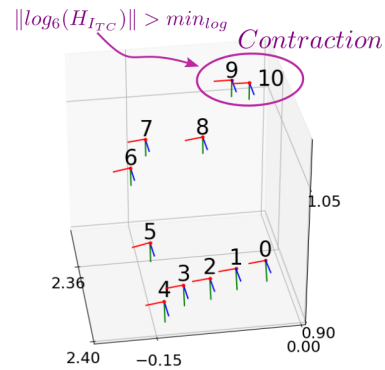
La figure 4.14a présente l’évolution de I_{TC} pour la simulation considérée : la contrainte d’égalité entre prédiction et référence est initialement mise jusqu’à la N_p^{me} prédiction sur laquelle elle reste pendant un grand nombre d’itérations en raison de la relaxation de ses commandes. Cette contrainte est décalée prédiction après prédiction au moment où la norme $\|log_6(H_{I_{TC}})\|$ est presque nulle. Cet évènement se produit grâce à la contrainte de diminution de prédiction à prédiction. Ce schéma



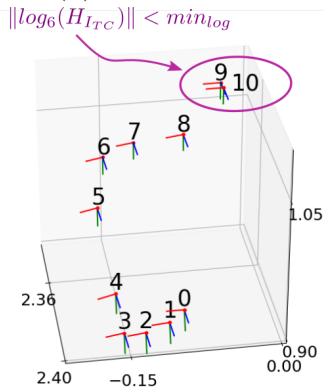
(a) Itération 2



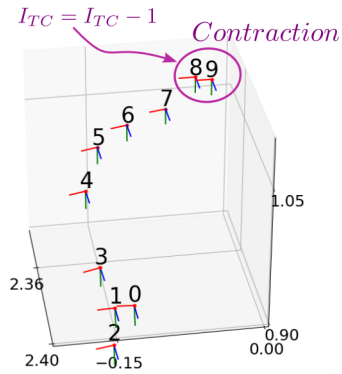
(b) Itération 92



(c) Itération 97



(d) Itération 102



(e) Itération 103

FIGURE 4.13 – Résultats de l'ensemble de contraintes de positionnement

crée alors une norme $\| \log_6(H_{I_{TC}}) \|$ avec une forme triangulaire (Figure 4.14b) synchronisée avec l'évolution de p_{TC} . La dernière figure 4.14c montre l'erreur entre les derniers moments de l'image prédits et leurs valeurs désirées, de sorte qu'une erreur proche de zéro signifie que la contrainte terminale est respectée.

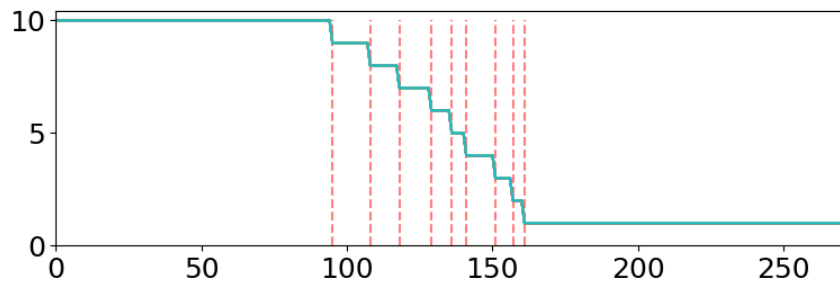
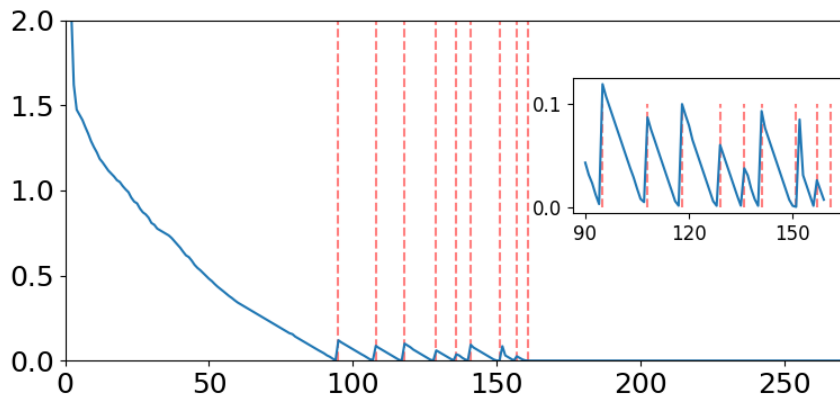
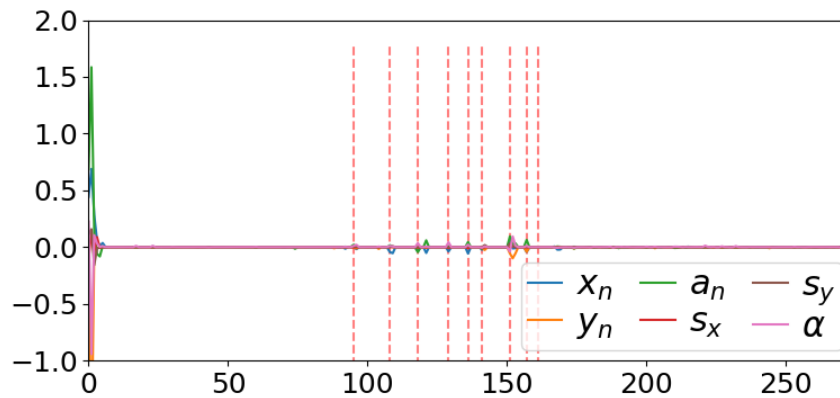
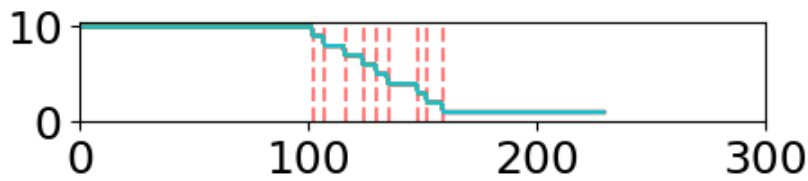
(a) Évolution de I_{TC} (b) Évolution de $\|\log_6(H_{I_{TC}})\|$ (c) Évolution de l'erreur terminale $\hat{S}(p + I_{TC}) - S^*$

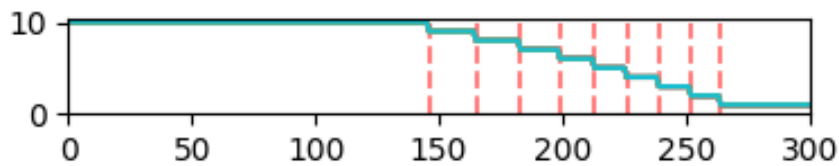
FIGURE 4.14 – Résultats relatifs à l'ensemble de contrainte de positionnement

Dans ce contexte, l'évolution de la valeur I_{TC} est intéressante car elle mesure la vitesse de convergence (voir figure 4.15). En effet, si I_{TC} atteint la valeur 1 plus rapidement, cela signifie que la tâche visuelle s'achève plus tôt. En raison de la nature distincte de la contrainte de contraction de la trajectoire basée sur le logarithme sur la figure 4.15a et de celle basée sur la commande sur la figure 4.15b, deux comporte-

ments différents sont observés : d'une part, les commutations sont commencées plus tôt, et d'autre part, l'intervalle entre deux commutations est plus court sur la figure 4.15a par rapport à la figure 4.15b. Il en résulte une durée d'asservissement plus courte dans le premier cas. Ceci souligne donc l'efficacité de la contrainte basée sur le logarithme par rapport à l'approche basée sur la commande. En effet, quantifier la distance entre deux poses est intrinsèquement plus précis lors de l'utilisation d'une mesure d'espace opérationnel que d'une mesure d'espace articulaire.



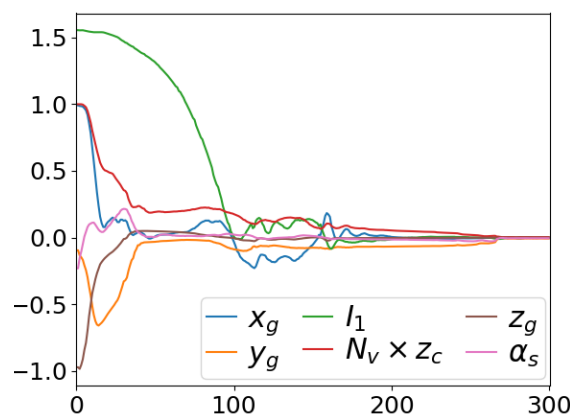
(a) Avec la contrainte de diminution basée sur le logarithme



(b) Avec la contrainte de diminution basée sur la commande

FIGURE 4.15 – Évolution de l'index I_{TC}

Cette analyse est confirmée par la comparaison de la figure 4.16 avec la figure 4.24a. En effet, les figures indiquent que l'exécution de la tâche visuelle est nettement plus lente lors de l'utilisation de la contrainte de diminution basée sur la commande.

FIGURE 4.16 – Évolution $S - S^*$ avec la contrainte de contraction basée sur la commande

4.6.2 Résultats expérimentaux

Dans cette section, le schéma d’asservissement visuel prédictif 2D est testé sur le robot TIAGo avec un scénario très proche de celui étudié en simulation. Le robot est donc placé dans une configuration initiale de démarrage tel que la caméra Realsense D435 montée sur son poignet ne perçoit pas la cible (figures 4.17a et 4.17b). La deuxième caméra permettant de mettre en œuvre le schéma multi-caméra est la caméra d’origine de la tête du TIAGo : une Orbbec Astra RGB-D. Cette dernière n’a une précision que de $\pm 10\text{cm}$ sur la mesure de profondeur, prouvant à nouveau la robustesse du schéma de contrôle proposé.

À nouveau, une première analyse qualitative de la trajectoire du robot et des images des caméras est menée et les observations sont justifiées quantitativement par la suite.



(a) Vue 1

(b) Vue 2

FIGURE 4.17 – Scénario d’expérimentation

Analyse de la trajectoire

La figure 4.18 présente des captures d’écran de la trajectoire du robot, des images de la caméra de l’effecteur final et des images de la caméra de tête.

Nous pouvons d’abord analyser la première partie de la trajectoire du robot sur les figures (a), (b), (c) et (d) pour chaque cas. Cette partie initiale vise à déplier le bras et à l’orienter face à la cible. La définition du problème d’optimisation s’appuie sur la projection sphérique afin d’éviter les problèmes de projection. En l’occurrence, la configuration initiale du robot par rapport à la cible est justement sujette à l’apparition de ces problèmes puisque l’axe optique de la caméra de l’organe terminal est quasiment perpendiculaire à la normale de la cible. Le comportement observé est produit pour deux raisons principales. Premièrement, la configuration avec le bras replié présente de faibles valeurs de mesure de manipulabilité, en raison de la proximité des limites articulaires. Deuxièmement, les caractéristiques visuelles et la matrice de pondération Q_S sont définies pour prioriser la correction de l’orientation de la caméra. L’utilisation de caractéristiques visuelles découplées nous permet d’influencer la trajectoire cartésienne grâce à la matrice Q_S , et ce comportement peut

donc être légèrement modifié. Les figures (i) à (l) et (o) à (r) présentent respectivement les trajectoires des images de l'effecteur final et de la tête pendant cette phase, et mettent en évidence l'indisponibilité initiale des informations visuelles dans l'image de la caméra de l'organe terminal. Les informations utilisées sont donc celles de la caméra de tête, qui sont alors reprojétées grâce au modèle correspondant. À la fin de cette première phase, le robot atteint des configurations proposant un compromis entre les deux tâches définies par F_{vs} et F_w . Le système conserve en effet des capacités de mouvement élevées tout en effectuant la tâche visuelle.

De plus, la situation de la caméra par rapport à la cible permet maintenant l'utilisation de la projection perspective et les méthodes de projection commencent à commuter. Les figures (e) , (m) et (s) indiquent en particulier l'instant où toutes les commutations sont effectuées, la fonction de coût est alors entièrement définie à partir d'information 2D. Les figures (e) , (f) et (g) décrivent alors la deuxième partie du mouvement où le robot s'approche de la cible tout en laissant de la place pour manœuvrer. Elle met en évidence la capacité du contrôleur à coordonner la base mobile et le bras en utilisant judicieusement la redondance du système pour accomplir au mieux les deux tâches. Lorsque l'organe terminal se rapproche de la pose désirée, le robot utilise ses degrés de liberté internes pour effectuer la tâche secondaire. On peut voir que le torse du robot tourne autour de la première articulation q_1 pour augmenter la manipulabilité.

Enfin, la dernière partie de la trajectoire illustre l'utilisation de l'ensemble de contraintes de positionnement pour prioriser la tâche visuelle par rapport à la maximisation de la manipulabilité. Seuls de petits mouvements sont générés et donc compliqué à observer sur des screenshots. La figure (h) montre la configuration finale du robot qui permet de conserver une certaine dextérité pour une potentielle tâche de préhension.

Résultats relatifs à la tâche principale d'asservissement visuel

Tout comme pour les résultats de simulation, la figure 4.19 illustre le bon déroulement du processus de sélection de la méthode de projection utilisée pour chaque prédiction. La transition est entièrement réalisée entre les itérations 56 et 71, celle-ci est donc assez rapide puisqu'il s'agit presque d'une commutation par itération.

Les figures 4.20a et 4.20b présentent respectivement l'image perçue par la caméra de la tête à la configuration initiale et celle perçue par la caméra de l'organe terminal à la configuration finale. Le schéma de commande permet donc bien de piloter le robot afin qu'il réalise la tâche visuelle. Ceci est confirmé par l'évolution des indices visuels présentée en figure 4.20d puisque ces derniers convergent vers leurs valeurs de référence. Les figures 4.20e et 4.20f suivent à nouveau la logique des deux phases de la trajectoire pour afficher l'évolution de l'erreur du vecteur d'indice visuel : avant (à gauche et donc basé sur les moments de l'image sphérique \hat{S}_s^m) et après (à droite et basé sur les moments de l'image plane \hat{S}_p^m) la commutation de la méthode de projection de toutes les prédictions.

La figure 4.20e montre que la première phase permet effectivement de placer la caméra de l'organe terminal en face de la cible (puisque z_g converge vers 1),

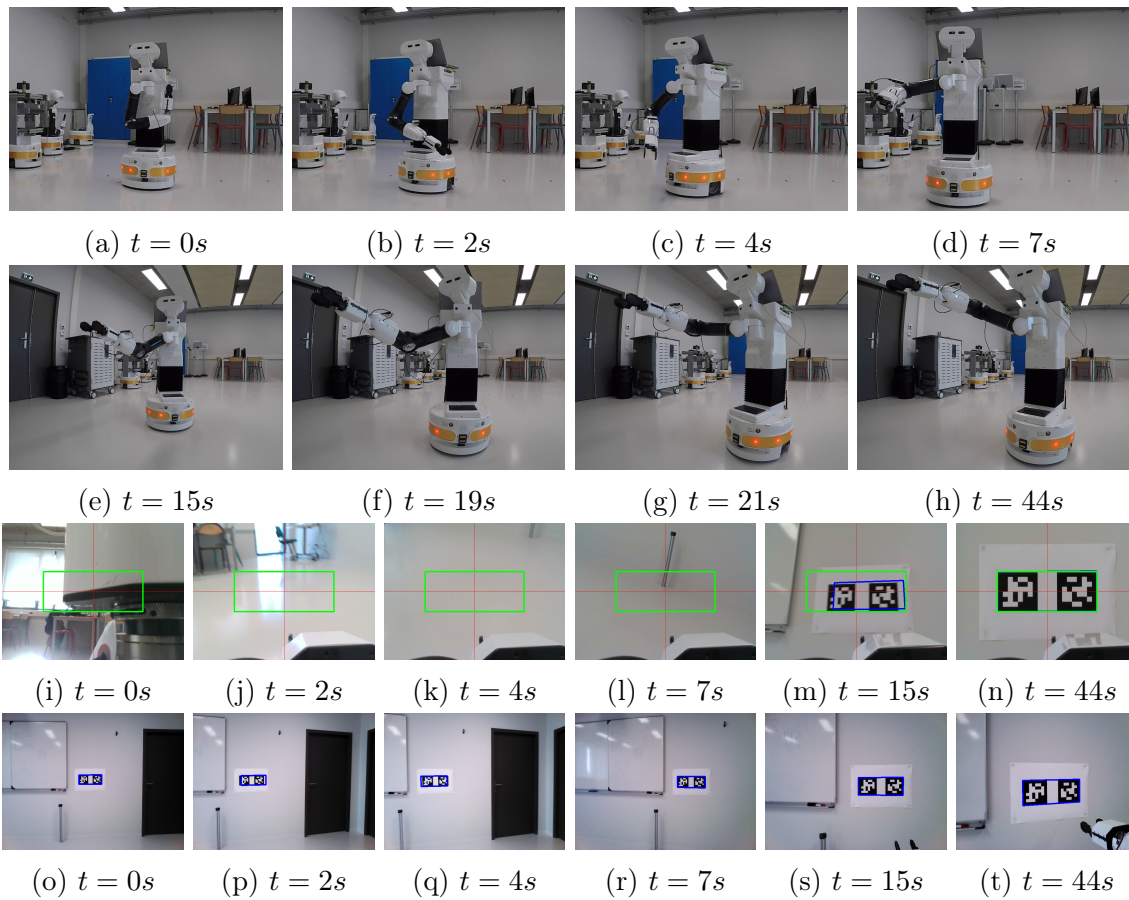


FIGURE 4.18 – Trajectoire du robot (a)-(h), informations de la caméra de l'organe terminal (i)-(n) et informations de la caméra de la tête (o)-(t)

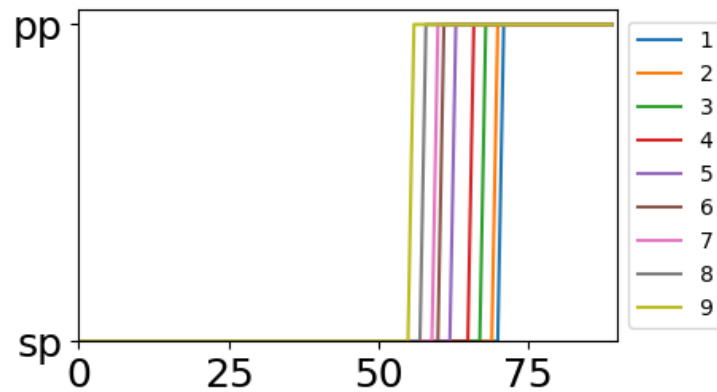


FIGURE 4.19 – Commutation de méthode de projection

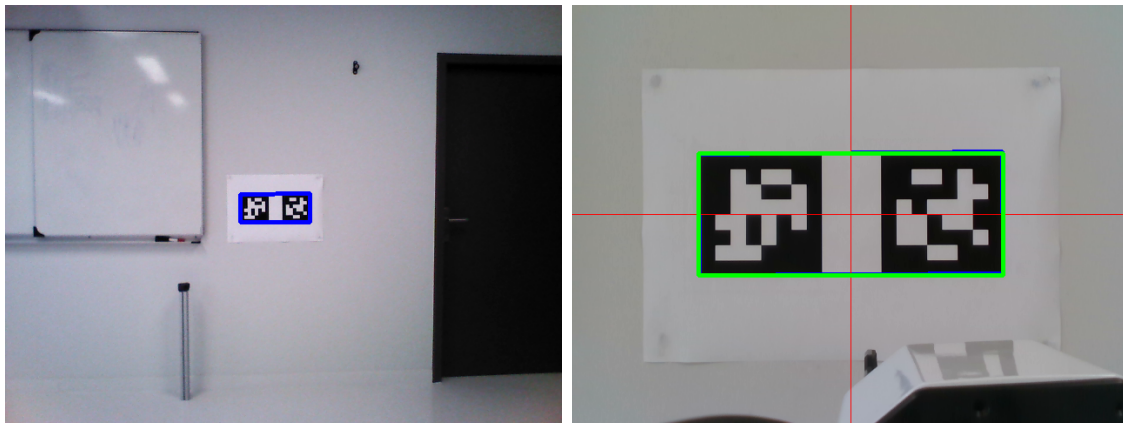
correctement orientée (puisque C_{xy} et α_S convergent également vers leurs valeurs de référence), et de manière centrée (puisque les erreurs relatives à x_g et y_g tendent également vers 0). Seule l'erreur de l'indice visuel I_1 prend un peu plus de temps puisqu'elle quantifie l'erreur de translation vers la cible. Cette figure indique donc que le robot avance bel et bien vers la cible tout en positionnant la caméra de l'organe terminal dans une situation adéquate pour commuter sur l'utilisation des informations de la caméra de l'organe terminal à l'aide de la projection perspective. Enfin, la figure 4.20f montre quant à elle le bon fonctionnement de l'ensemble de positionnement qui évite le compromis entre les deux tâches. Tous les indices visuels de \hat{S}_p^m convergent réellement vers leurs valeurs désirées.

Convergence et stabilité

La figure 4.21 présente les résultats relatifs à l'ensemble de contraintes de positionnement d'une manière analogue à ce qui a été fait en simulation. Ces résultats sont plutôt similaires à ceux de simulation et le triptyque proposé permet effectivement d'éviter un compromis entre les tâches. Les contraintes de prédiction à référence et de contraction sont ponctuellement et simultanément non-respectées (figure 4.21c) à cause du temps maximal de résolution. Ce non-respect est vite corrigé et reste généralement faible.

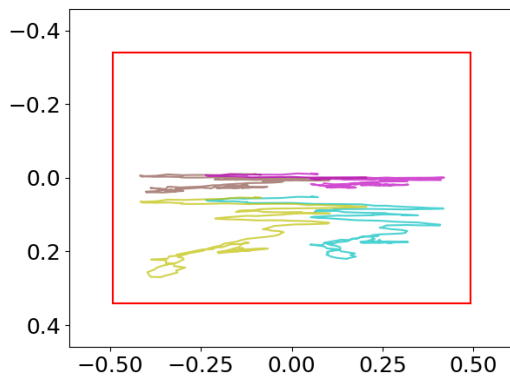
4.7 Schéma d'asservissement visuel prédictif 2D 1/2

Il est proposé d'explorer une deuxième approche. Celle-ci change légèrement la démarche suivie jusqu'à maintenant qui avait pour objectif de minimiser les étapes de reconstruction 3D et d'utiliser au maximum des informations au plus proche des données de la caméra. En effet, les travaux [Chaumette and Hutchinson, 2006] soulignent qu'une estimation approximative de ces dernières peut perturber la tra-

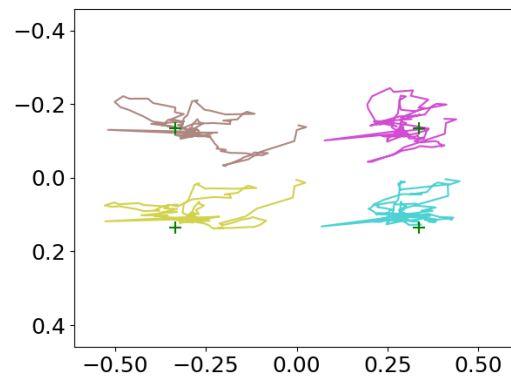


(a) Image initiale

(b) Image finale



(c) Trajectoire des points d'intérêt dans l'image de la tête



(d) Trajectoire des points d'intérêt dans l'image de l'organe terminal

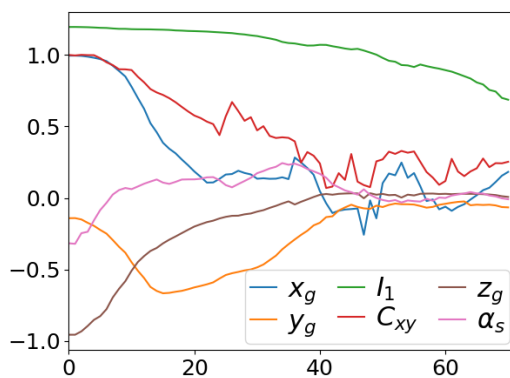
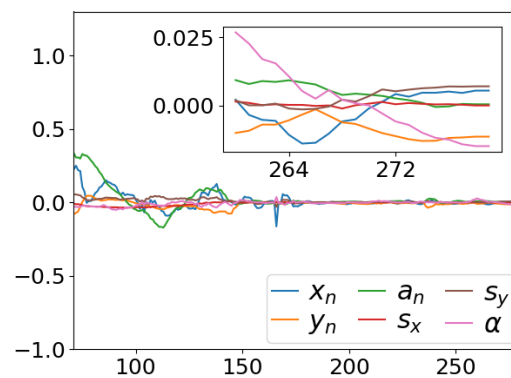

 (e) Évolution de l'erreur $\hat{S}_s^m - S_s^{m*}$

 (f) Évolution de l'erreur $\hat{S}_p^m - S_p^{m*}$

FIGURE 4.20 – Résultats du schéma complet relatifs à la tâche visuelle

jectoire effectuée par le robot pour atteindre sa situation de référence, mais n'a aucun effet sur la précision du positionnement final.

Comme la projection sphérique est utilisée virtuellement dans ces travaux, l'uti-

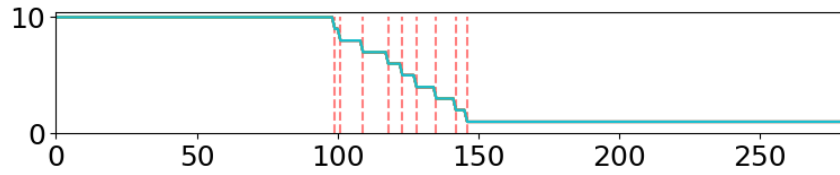
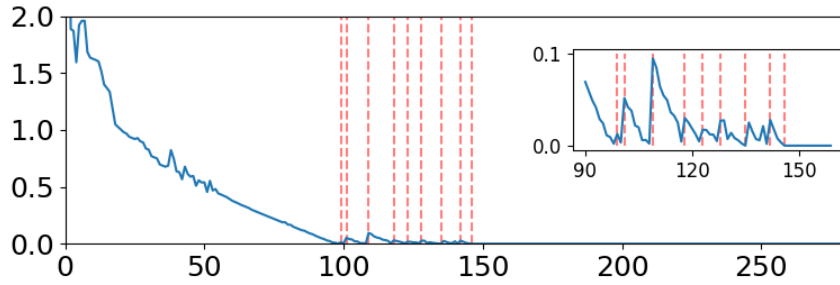
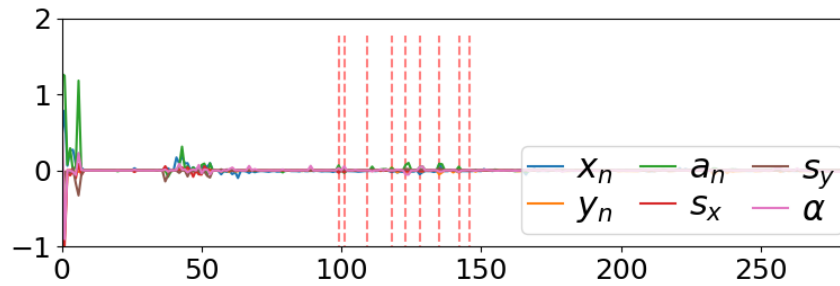
(a) Évolution de I_{TC} (b) Évolution de $\|\log_6(H_{I_{TC}})\|$ (c) Évolution de l'erreur terminale $\hat{S}(p + I_{TC}) - S^*$

FIGURE 4.21 – Résultats relatifs à l'ensemble de contrainte de positionnement

lisation d'information 3D est nécessaire pour la réaliser. De plus, certains indices visuels du vecteur S_s^m requièrent une étape de reconstruction. Une estimation correcte de ces informations est donc cruciale, car elles apparaissent à la fois dans l'erreur et dans les modèles de prédiction. Des estimations approximatives induisent donc des perturbations sur la trajectoire réalisée, mais ont également un impact sur le positionnement final. Le premier schéma proposait dans ce sens de n'utiliser la projection sphérique que pour éviter les problèmes de projection perspective et de commuter par la suite lorsque ces derniers ne peuvent plus se produire.

Cependant, un deuxième schéma de contrôle qui conserve quant à lui la méthode de projection sphérique durant toute la trajectoire a été élaboré. Dans ce schéma, les informations 3D sont donc également utilisées dans le régime permanent de l'asservissement et les étapes de commutation sont donc évitées. L'impact de l'étape de reconstruction 3D sur le positionnement final de l'organe terminal est analysé par la suite. Le vecteur d'erreur au cœur de la fonction de coût n'est donc plus 2D, mais

bien hybride. Compte tenu de cette différence conceptuelle, nous avons décidé de qualifier ce deuxième schéma de stratégie d’asservissement visuel prédictif 2D 1/2.

4.8 Résultats obtenus avec le schéma d’asservissement visuel prédictif 2D 1/2

Une nouvelle fois, des résultats de simulation sont tout d’abord présentés. Ces derniers viennent premièrement compléter ceux du schéma précédent en illustrant l’impact de l’utilisation de gradients symboliques. La prise en compte de la manipulabilité du manipulateur mobile dans la tâche secondaire est également étudiée pour mettre en évidence la meilleure combinaison des indices de manipulabilité. Ces analyses permettent ensuite de sélectionner les meilleures options pour notre schéma VPC à tester sur le robot. Plusieurs scénarios expérimentaux s’appuyant sur les combinaisons sélectionnées sont réalisés afin de mettre en avant les bonnes performances de la stratégie de commande proposée.

Tous les résultats de cette section sont obtenus avec $N_p = N_c = 10$ et $N_r = 1$. Le temps maximal de résolution est à nouveau fixé à 150ms et la boucle de contrôle fonctionne à une fréquence de 5 Hz. La matrice de pondération est $Q_S = \text{diag}(1, 1, 1, 1, 1, 1)$ mais celle-ci peut être ajustée pour modifier le comportement du robot.

4.8.1 Résultats de simulation

Dans cette première section, le schéma de contrôle introduit est exécuté dans *Gazebo*. La figure 4.22 présente un exemple de configuration de simulation initiale, ainsi que la configuration finale du robot obtenue avec le contrôleur VPC présenté. L’objectif est de comparer différentes approches pour mettre en évidence la pertinence des points clés introduits dans le schéma. Pour les sections suivantes des résultats de simulation, la pose initiale de la base mobile est $(0,0,0)$ et $\alpha_w = 1$. Cette configuration est représentative d’un cas générique.

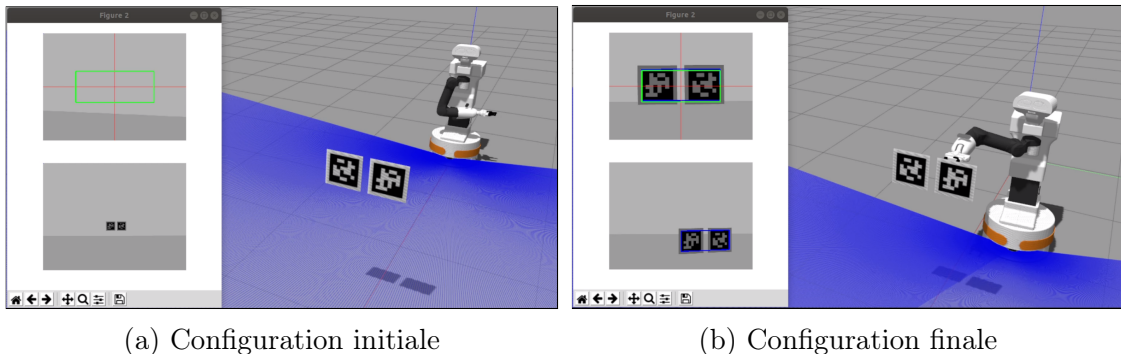


FIGURE 4.22 – Environnement *Gazebo*

Analyse de la trajectoire du robot

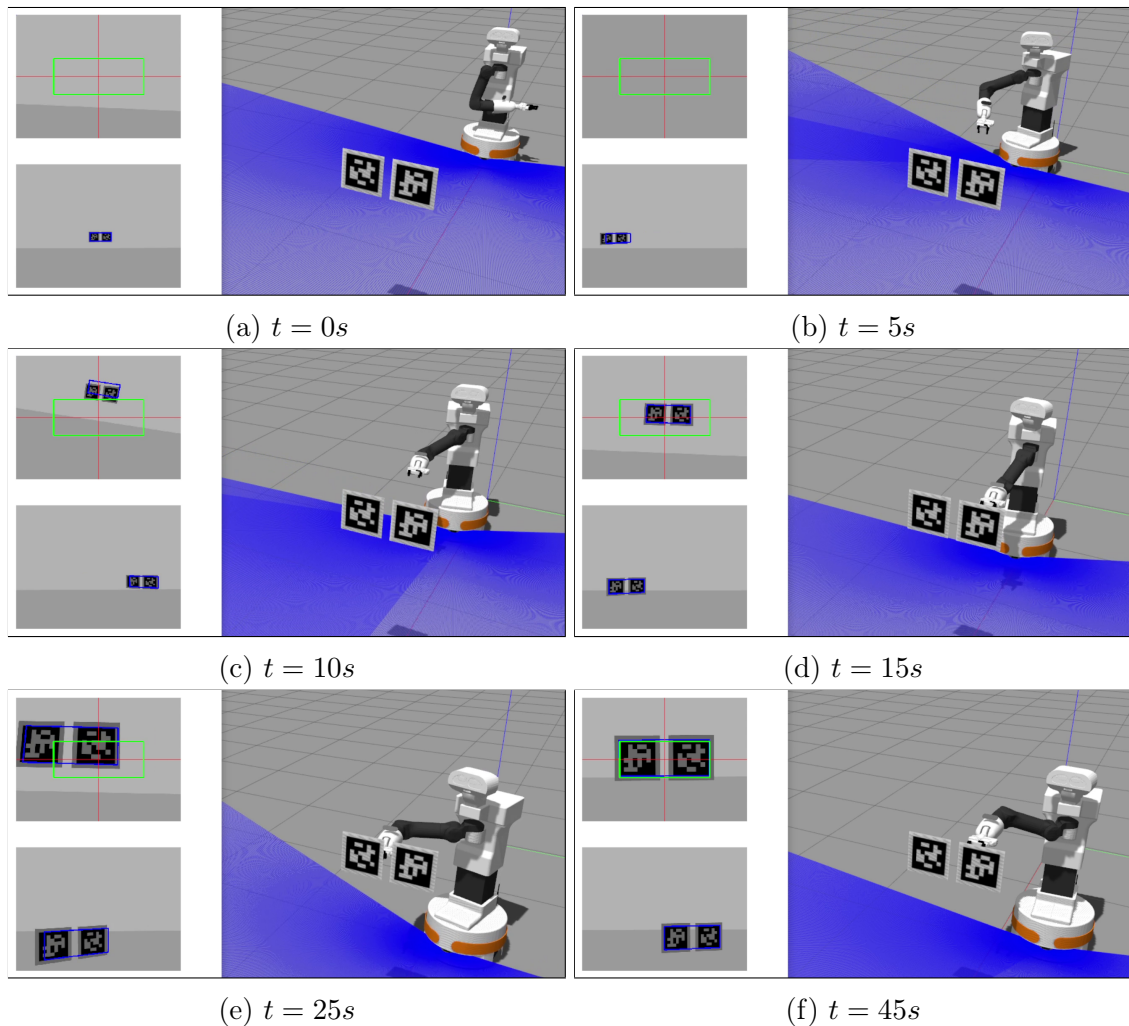


FIGURE 4.23 – Trajectoires du robot et des images obtenues

La figure 4.23 présente des captures d'écran de la trajectoire du robot et des images des deux caméras. Celles-ci sont relativement similaires aux résultats obtenus avec le schéma présenté précédemment. Les deux tâches semblent à nouveau correctement réalisées. Une analyse plus approfondie est une fois de plus réalisée dans la section des résultats expérimentaux.

Résultats relatifs à la tâche visuelle

La figure 4.24a présente l'erreur entre les moments de l'image et leurs valeurs désirées. Celle-ci montre clairement la bonne exécution de la tâche visuelle puisque les erreurs s'annulent. Afin d'étudier l'impact de l'utilisation de gradients symboliques. La figure 4.24b montre l'erreur analogue sans calcul symbolique des gradients avec CasADi lorsque celui-ci s'appuie sur les différences finies. Elle montre des er-

reurs beaucoup plus élevées, mettant ainsi en évidence l’incapacité à obtenir un positionnement précis.

Concernant la réalisation de la tâche d’asservissement visuel, cette nouvelle version de schéma de commande propose des performances similaires au schéma 2D. En effet, la convergence et l’erreur finale sont équivalentes. La loi de commande est robuste aux erreurs de mesure des informations 3D et ne perturbe pas la tâche de positionnement. En effet, le scénario étudié permet de placer la caméra de l’organe terminal proche de la cible. Les informations utilisées dans la commande proviennent ainsi de cette caméra et les erreurs de mesures sont alors faibles, permettant un positionnement précis malgré les étapes de reconstruction 3D.

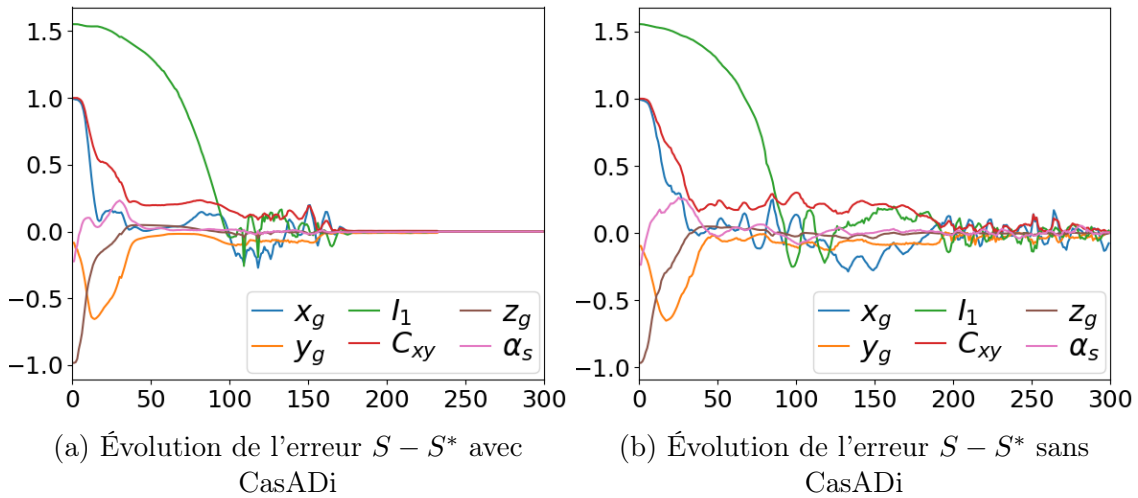


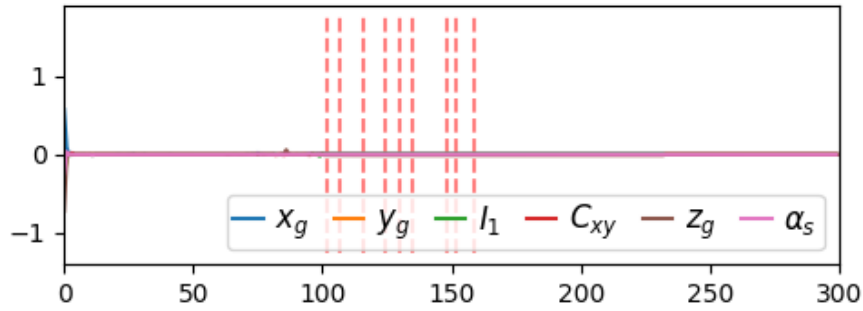
FIGURE 4.24 – Résultats relatifs à la tâche visuelle avec et sans CasADi

Résultats relatifs à la stabilité

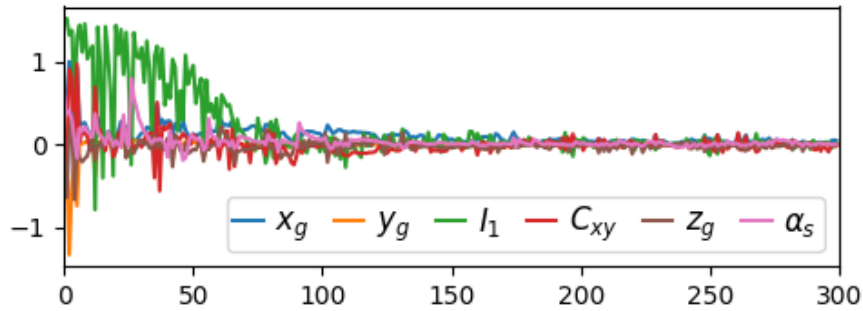
La figure 4.25a permet d’analyser le respect de la contrainte entre prédiction et référence en montrant l’évolution de l’erreur entre les I_{TC}^{eme} indices visuels prédits et leurs valeurs désirées lorsque CasADi est utilisé. Une valeur proche de zéro indique que la contrainte est respectée. De plus, les lignes verticales rouges représentent les décalages de l’ensemble de contraintes de positionnement variant dans le temps. L’erreur entre $\hat{S}(p + I_{TC})$ et S^* étant nulle ou proche de zéro sur l’ensemble du contrôle, la contrainte de prédiction-référence est respectée.

La figure 4.25b présente les résultats obtenus sans CasADi, mettant en avant le défi de respecter la contrainte de prédiction-référence. Ce scénario exige de nombreuses itérations pour que le solveur calcule une solution satisfaisant toutes les contraintes, ce qui est difficile dans une limite de temps raisonnable ($< 200ms$) sans CasADi. La valeur de l’erreur relative à la contrainte d’égalité entre prédiction et référence est élevée au début de l’asservissement lorsque le bras est replié et reste non nulle jusqu’à la fin. L’erreur est même très importante lors de la phase de déploiement du bras, car la situation de référence est très éloignée de la configuration actuelle. De nombreuses itérations de résolution sont alors nécessaires et ne sont pos-

sibles que grâce au calcul symbolique. La décroissance de la fonction de valeur n'est donc plus assurée et il est impossible de garantir la stabilité du système en boucle fermée. Ainsi, ces résultats montrent l'intérêt du calcul symbolique des gradients.



(a) Évolution de $\hat{S}(p + I_{TC}) - S^*$ avec CasADi



(b) Évolution de $\hat{S}(p + I_{TC}) - S^*$ sans CasADi

FIGURE 4.25 – Évolution de $\hat{S}(p + I_{TC}) - S^*$ avec et sans CasADi

Analyse de l'impact de choix de la mesure de manipulabilité

L'objectif de cette section est d'étudier l'impact du choix de la mesure de la manipulabilité. Quatre scénarios sont pris en compte :

- *C1* : Sans prise en compte de la manipulabilité, c'est-à-dire $K_w = 0$.
- *C2* : Prise en compte de la manipulabilité w'_a du bras uniquement, c'est-à-dire $\alpha_w = 1$.
- *C3* : Prise en compte de la manipulabilité w'_{mm} du manipulateur mobile uniquement, c'est-à-dire $\alpha_w = 0$.
- *C4* : Prise en compte des deux mesures w'_a et w'_{mm} , avec $\alpha_w = 0.1$.

Les figures 4.26a, 4.26b et 4.26c présentent respectivement l'évolution de w_a , w_{mm} et P obtenue pour chaque cas. Ces figures illustrent clairement que les cas *C2* et *C3* sont effectivement les scénarios pour lesquels w'_a et w'_{mm} sont maximisées, respectivement, comme prévu. Elles révèlent également que dans le cas *C4*, le compromis attendu entre les deux mesures de manipulabilité est atteint. De plus, il est à noter que l'évolution de w'_{mm} dans les cas *C1* et *C3* donne des résultats similaires en termes de maximisation. Néanmoins, dans la figure 4.26c, P diminue pour ces deux cas. Cela peut être attribué à l'articulation q_4 qui s'approche de sa limite. Les performances sont assez similaires pour chaque mesure concernant *C1* et *C3*. Cependant, ceci est spécifique au scénario considéré. Ces deux résultats soulignent l'importance de conserver le terme w'_a dans F_w . Par conséquent, opter pour $K_w = 0$ ou $\alpha_w = 0$ n'est pas le choix le plus approprié. Concernant maintenant les cas *C2* et *C4*, conclure est plus difficile. Les performances semblent en effet comparables dans le scénario de simulation étudié et seule une légère différence dans la mesure w'_a peut être observée. L'impact de cette différence sera mis en évidence dans la trajectoire de configuration du robot dans la section des résultats expérimentaux.

4.8.2 Résultats expérimentaux

À partir des résultats de simulation précédents, deux configurations présentant des performances équivalentes ont été mises en évidence : les cas *C2* et *C4*. Ils ont donc été mis en œuvre et testés sur notre plateforme robotique, avec une pose initiale de la base mobile $\chi_b = [0, 0, 0]^T$. De plus, deux autres scénarios avec $\alpha_w = 1$ et des poses initiales différentes de la base mobile ont été réalisés pour démontrer la robustesse et la flexibilité de la méthode proposée : $C2_l$ et $C2_r$, respectivement, avec $\chi_b = [0, +0.8, 0]^T$ et $\chi_b = [0, -0.8, 0]^T$.

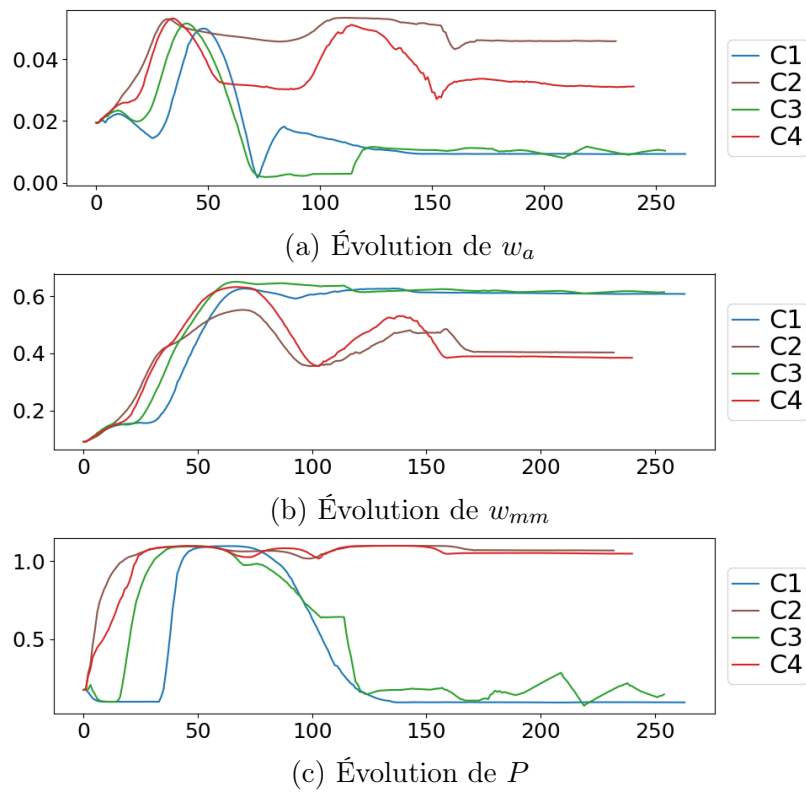
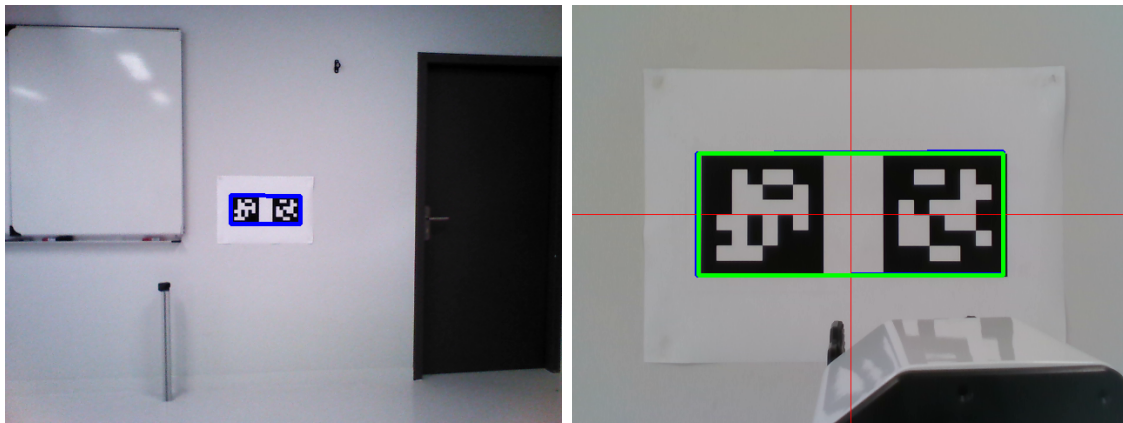


FIGURE 4.26 – Évolution des mesures de manipulabilité

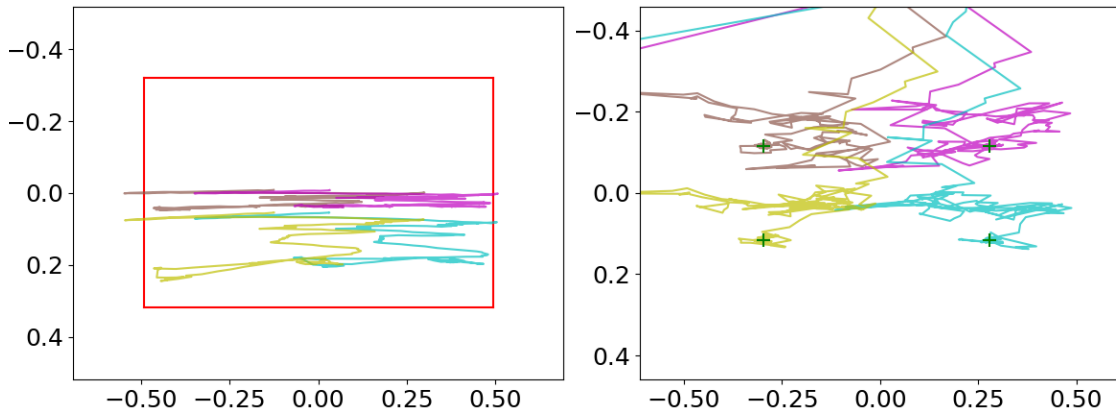
Analyse de la tâche visuelle

Dans tous les scénarios, le système accomplit avec succès la tâche principale (voir figure 4.27b). Le robot commence avec un bras replié, et la cible n'est pas détectable par la caméra de tête (voir figure 4.27a). De plus, les figures 4.27c et 4.27d affichent respectivement la trajectoire des points d'intérêt dans les images de la tête et de l'effecteur final, et confirment que la tâche visuelle est correctement réalisée.

Il convient également de noter que les indices visuels sont temporairement perdus dans l'image de l'effecteur final, mais le robot peut continuer à exécuter la tâche grâce à la caméra de tête.



(a) Image initiale de la caméra de la tête (b) Image finale de la caméra de l'organe terminal



(c) Trajectoire de la cible dans la caméra de la tête (d) Trajectoire de la cible dans la caméra de l'organe terminal

FIGURE 4.27 – Résultats relatifs à la tâche visuelle

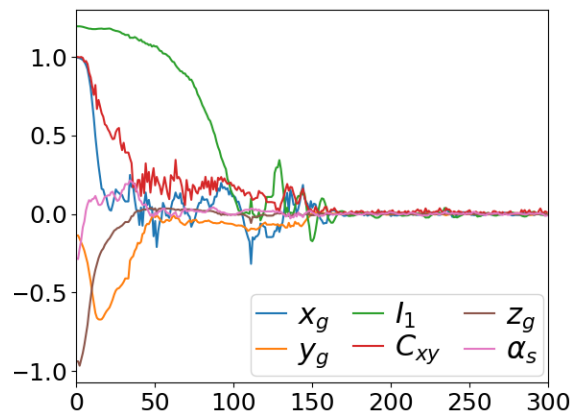


FIGURE 4.28 – $S - S^*$ evolution

Résultats relatifs à l'ensemble de contraintes de positionnement

Les performances de l'ensemble de contraintes de positionnement sont analysés dans cette section. Sur la figure 4.29a, on peut voir que l'erreur entre $S(p + I_{TC})$ et S^* est proche de zéro, ce qui indique que la contrainte terminale est généralement satisfaite, sauf dans quelques cas exceptionnels. Ce problème apparaît à nouveau, car le processus d'optimisation s'arrête parfois à cause du temps maximal de résolution fixé et fournit une solution qui n'englobe pas tout l'ensemble de contraintes en raison d'un délai de résolution maximum accordé au solveur. Les contraintes de prédiction à référence et de contraction sont ponctuellement et simultanément non-respectées. Ce non-respect est vite corrigé et reste généralement faible. Cependant, ce dernier est rapidement corrigé par le contrôleur et cela ne perturbe pas le processus de définition de l'ensemble de contraintes de positionnement. Sur la figure 4.29b, on peut voir l'impact de la contrainte de contraction de la trajectoire. En effet, l'évolution de $\|\log_6(H_{I_{TC}})\|$ montre qu'elle diminue constamment. De plus, elle présente une forme triangulaire en raison de la mise à jour de I_{TC} (cf. figure 4.29c) lorsqu'elle atteint une faible valeur.

Analyse de la trajectoire du robot

Les Figures 4.30, 4.31, 4.32, et 4.33 présentent respectivement des captures d'écran du robot, des images de la caméra de l'effecteur final et des images de la caméra de tête, pour les scénarios $C2$, $C4$, $C2_l$ et $C2_r$.

Nous pouvons d'abord analyser la première partie de la trajectoire du robot sur les figures (a), (b), (c) et (d) pour chaque cas. Pour tous les scénarios, cette partie initiale est similaire et vise à déplier le bras. Ce comportement est produit pour deux raisons principales. Premièrement, la configuration avec le bras replié présente de faibles valeurs de mesure de manipulabilité, en raison de la proximité des limites articulaires. Le choix de l'indice de manipulabilité utilisé dans le schéma de contrôle n'affecte pas significativement cette partie des mouvements. Deuxièmement, les caractéristiques visuelles et la matrice de pondération Q_S sont définies pour prioriser la correction de l'orientation de la caméra de l'effecteur final. L'utilisation de caractéristiques visuelles découplées nous permet d'influencer la trajectoire cartésienne grâce à la matrice Q_S , et ce comportement peut donc être légèrement modifié. Les figures (i) - (l) et (o) - (r) présentent respectivement les trajectoires des images de l'effecteur final et de la tête pendant cette phase, et mettent en évidence la prise en charge de la disponibilité initiale des points cibles dans la caméra de l'effecteur final en utilisant les informations de la caméra de tête. À la fin de cette première phase, le robot atteint des configurations qui proposent un compromis entre les deux tâches définies par F_{vs} et F_w . Le système conserve en effet des capacités de mouvement élevées tout en effectuant la tâche visuelle.

Les figures (e), (f) et (g) décrivent la partie médiane du mouvement. Cette dernière commence lorsque le robot se rapproche de la cible tout en laissant encore de la place pour manœuvrer. Elle met en évidence la capacité du contrôleur à utiliser judicieusement la redondance du système pour accomplir au mieux les deux tâches

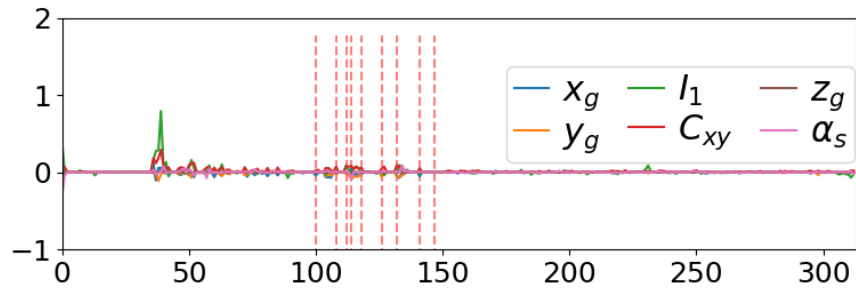
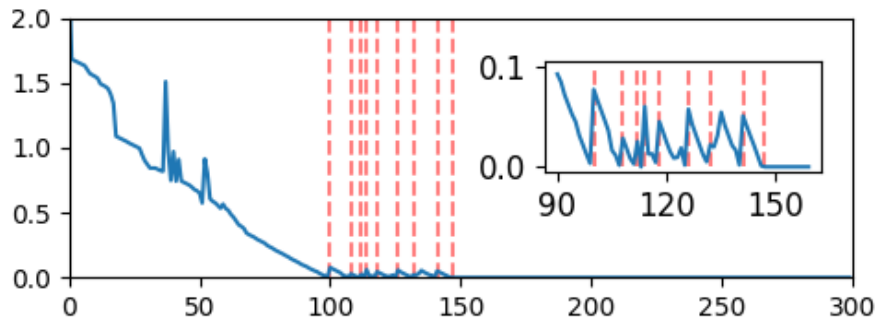
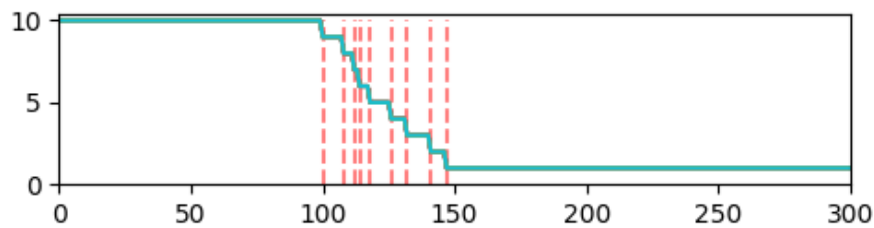
(a) Évolution l'erreur $\hat{S}(p + I_{TC}) - S^*$ (b) Évolution de la norme $\|\log_6(H_{I_{TC}})\|$ (c) Evolution de I_{TC}

FIGURE 4.29 – Résultats relatifs à l'ensemble de contraintes de positionnement

en coordonnant la base mobile et le bras. Lorsque l'effecteur final se rapproche de la pose désirée, le robot utilise ses degrés de liberté internes pour effectuer la tâche secondaire. On peut voir que le bras du robot tourne autour de la première articulation q_1 pour augmenter la manipulabilité. Cet aspect est encore plus remarquable lorsque la position initiale de la base mobile est éloignée de la position finale. Ce comportement dure en effet plus longtemps pour $C2_r$ que pour $C2$, et pour $C2$ que pour $C2_l$, justifiant principalement la durée plus longue de l'ensemble de la trajectoire. Les figures (m) et (s) présentent des exemples d'images pour les deux caméras à ce moment-là : la cible devient généralement visible par la caméra de l'effecteur final, mais le bras conserve toute sa plage de mouvement grâce à la conception de la contrainte de visibilité définie uniquement sur la caméra de tête.

Enfin, la dernière partie de la trajectoire illustre l'utilisation de l'ensemble de contraintes de positionnement pour prioriser la tâche visuelle par rapport à la maxi-

misation de la manipulabilité. Seuls de petits mouvements sont générés, et ils sont assez similaires pour $C2$, $C2_l$ et $C2_r$. Les figures (h) montrent la configuration finale du robot pour chaque cas. Il y a une différence notable entre les cas $C4$ et $C2$, où un déploiement de bras plus étendu est obtenu dans $C4$ car la contribution de la base mobile est prise en compte. Pendant cette partie, la caméra de l'effecteur final voit toujours la cible (Fig. (n)). Les données visuelles sont donc utilisées par le contrôleur, évitant ainsi toute erreur de reprojction et permettant donc un positionnement précis.

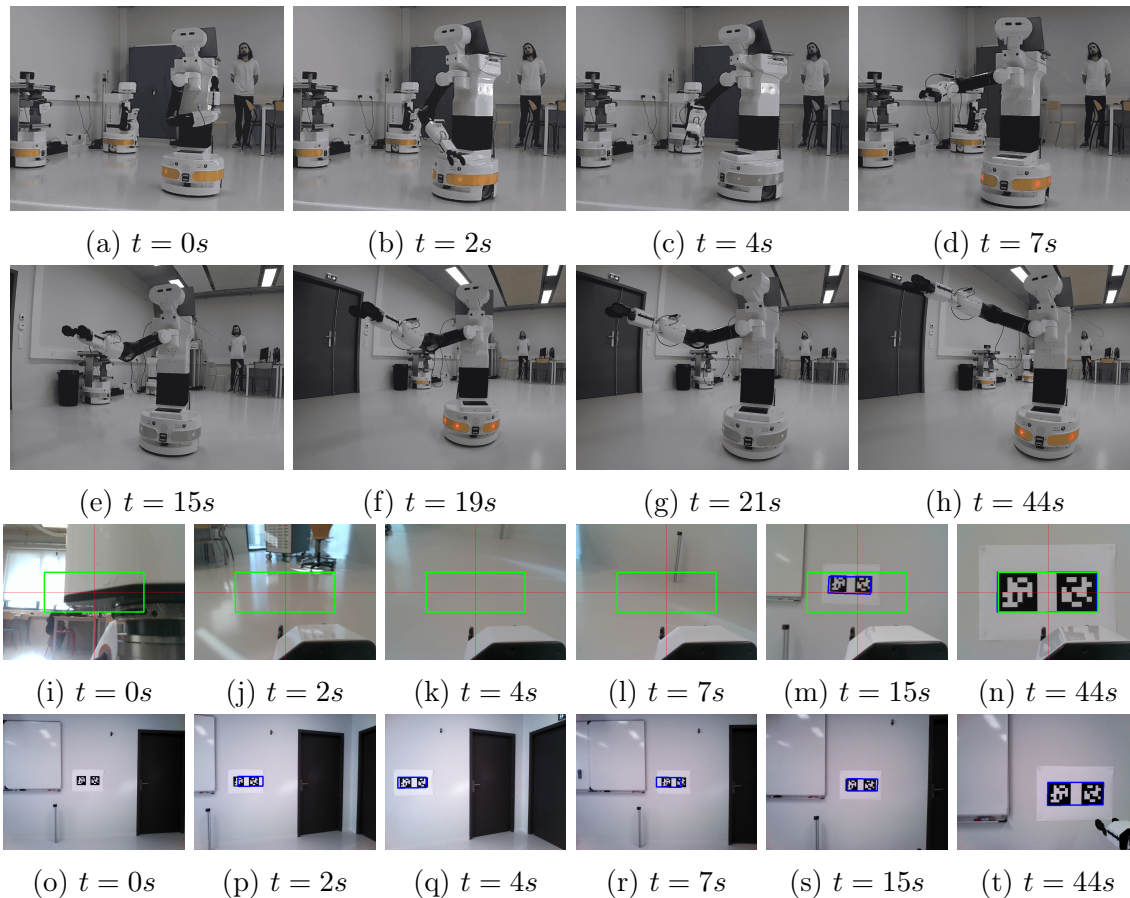


FIGURE 4.30 – Trajectoire du robot (a)-(h), informations de la caméra de l'organe terminal (i)-(n) et informations de la caméra de la tête (o)-(t)

4.9 Conclusion

Une version étendue du contrôleur VPC a été présentée dans ce chapitre. Le premier axe principal d'amélioration concernait la gamme de tâche que le robot peut effectuer. En effet, la contrainte de visibilité sur la caméra de l'organe terminale était contraignante. L'utilisation d'une seconde caméra déportée par rapport au bras a été utilisée pour aborder ce problème. Cependant, la projection perspective

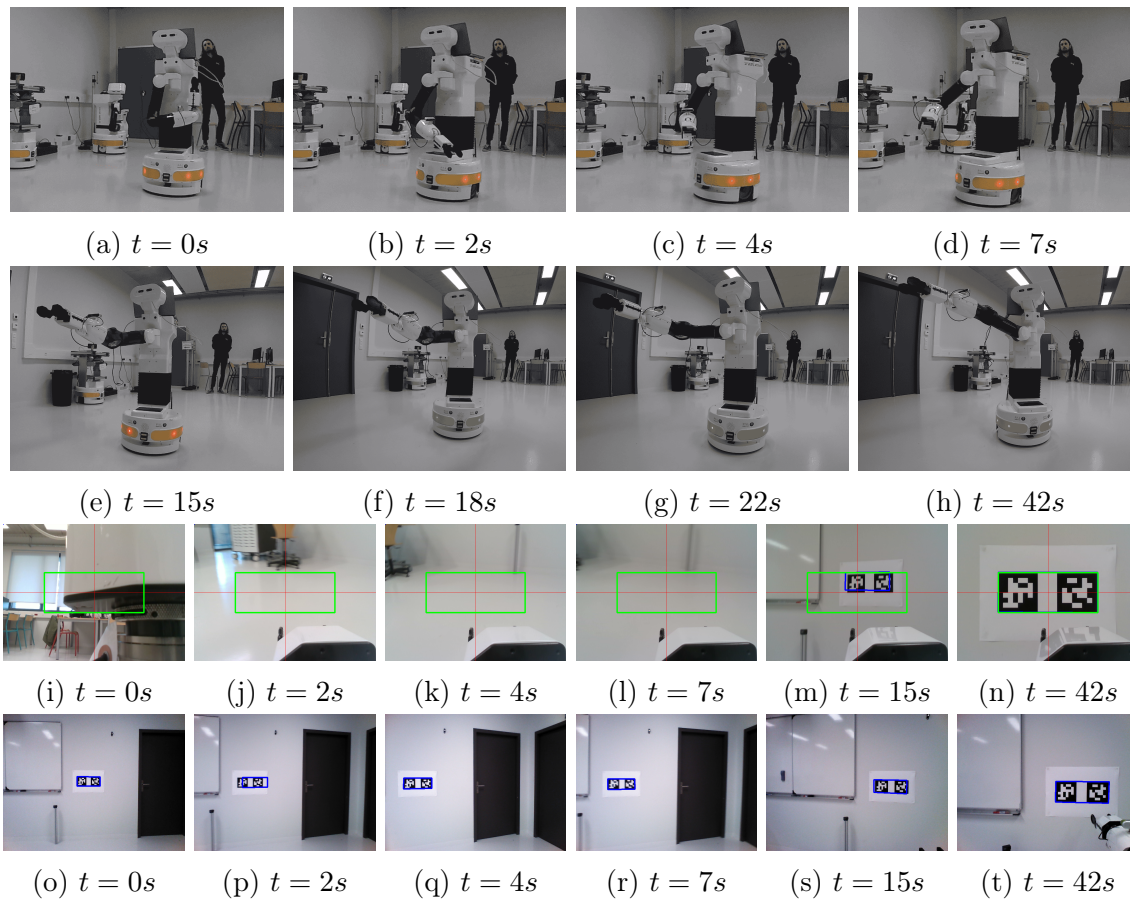


FIGURE 4.31 – Trajectoire du robot (a)-(h), informations de la caméra de l'organe terminal (i)-(n) et informations de la caméra de la tête (o)-(t)

n'est pas adaptée à des situations dans lesquelles la caméra qui ne fait pas face à la cible et des problèmes de projection peuvent apparaître. L'idée d'une projection sphérique virtuelle est mise en œuvre afin de les éviter et permet d'exprimer la tâche d'asservissement visuel dans l'espace de la caméra de l'organe terminal. Les moments 3D sont employés pour retrouver de bonnes propriétés de découplage. L'introduction d'une nouvelle définition du coût visuel a permis l'élaboration de deux schémas de contrôle. Le premier n'utilise la projection sphérique que lors du régime transitoire. Il a été qualifié d'asservissement visuel prédictif 2D et une stratégie de commutation de la méthode de projection a été proposée. Celle-ci permet de réaliser la transition vers la projection sphérique en douceur grâce à une utilisation astucieuse de la structure du problème d'optimisation. Le deuxième schéma exploite la projection sphérique pendant toute la commande. Ce dernier a été qualifié d'asservissement visuel prédictif 2D 1/2 puisque l'expression de l'erreur s'appuie sur des informations 3D. Les deux schémas ont été testés sur le robot et les résultats sont bons. Un grand nombre de scénarios a été considéré et prouve la robustesse et la répétabilité de la loi de commande.

Le deuxième axe d'amélioration était en rapport avec le comportement du robot

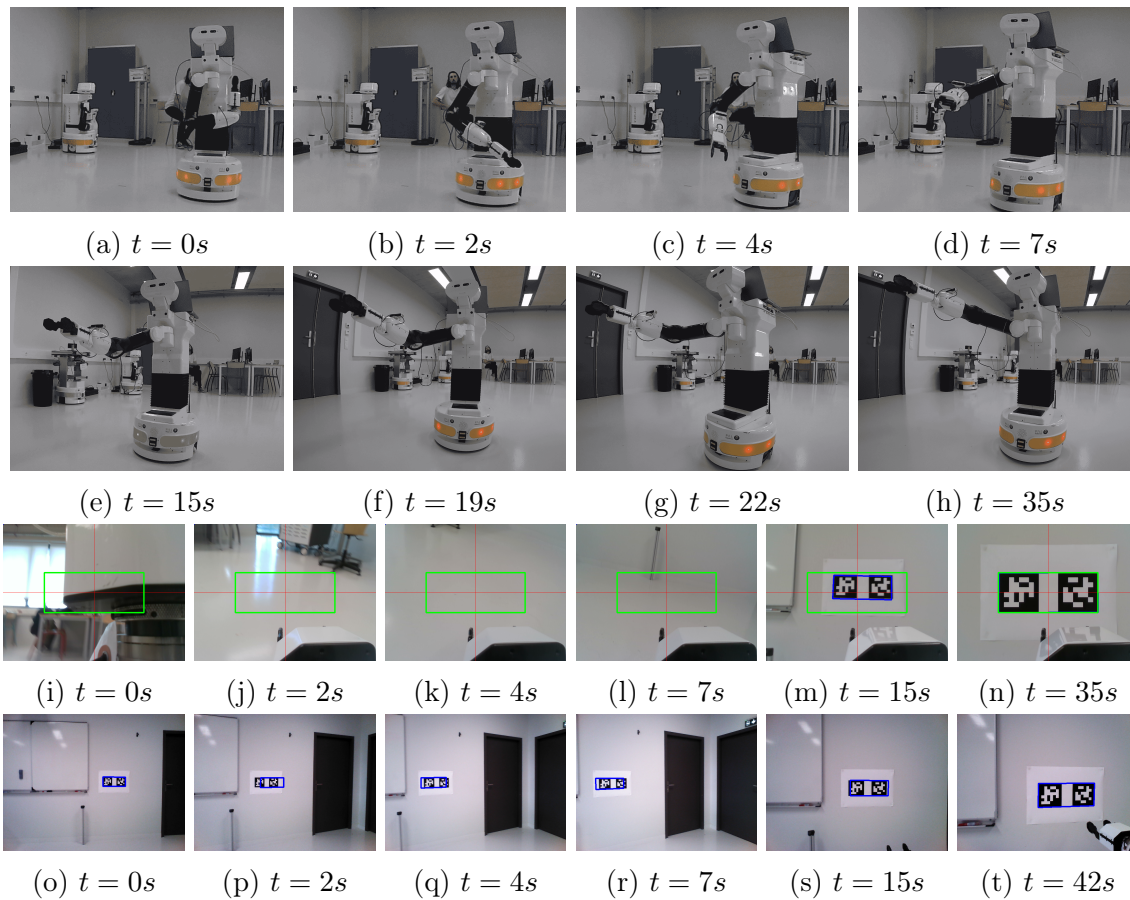


FIGURE 4.32 – Trajectoire du robot (a)-(h), informations de la caméra de l'organe terminal (i)-(n) et informations de la caméra de la tête (o)-(t)

qui se déplace essentiellement le bras tendu. L'introduction de mesures de manipulabilité a été introduite dans la fonction de coût. Cette notion permet de favoriser les configurations maximisant la capacité de mouvement de l'organe terminal. Les résultats confirment les propositions théoriques et le robot se déplace d'une manière plus adéquate. De plus, plusieurs mesures ont été introduites et les résultats ont permis de mettre en évidence les meilleures combinaisons.

Enfin, ces résultats n'ont pu être obtenus que grâce à l'introduction du triptyque de contraintes de positionnement. Ce dernier est un ensemble de contrainte variant dans le temps qui adapte la notion de contrainte terminale pour assurer la convergence de la tâche visuelle. Ce triptyque permet de répondre aux enjeux de sous-optimalité de la solution observée dans le premier chapitre, et d'éviter qu'un compromis soit réalisé entre la tâche d'asservissement visuel et la maximisation de la manipulabilité.

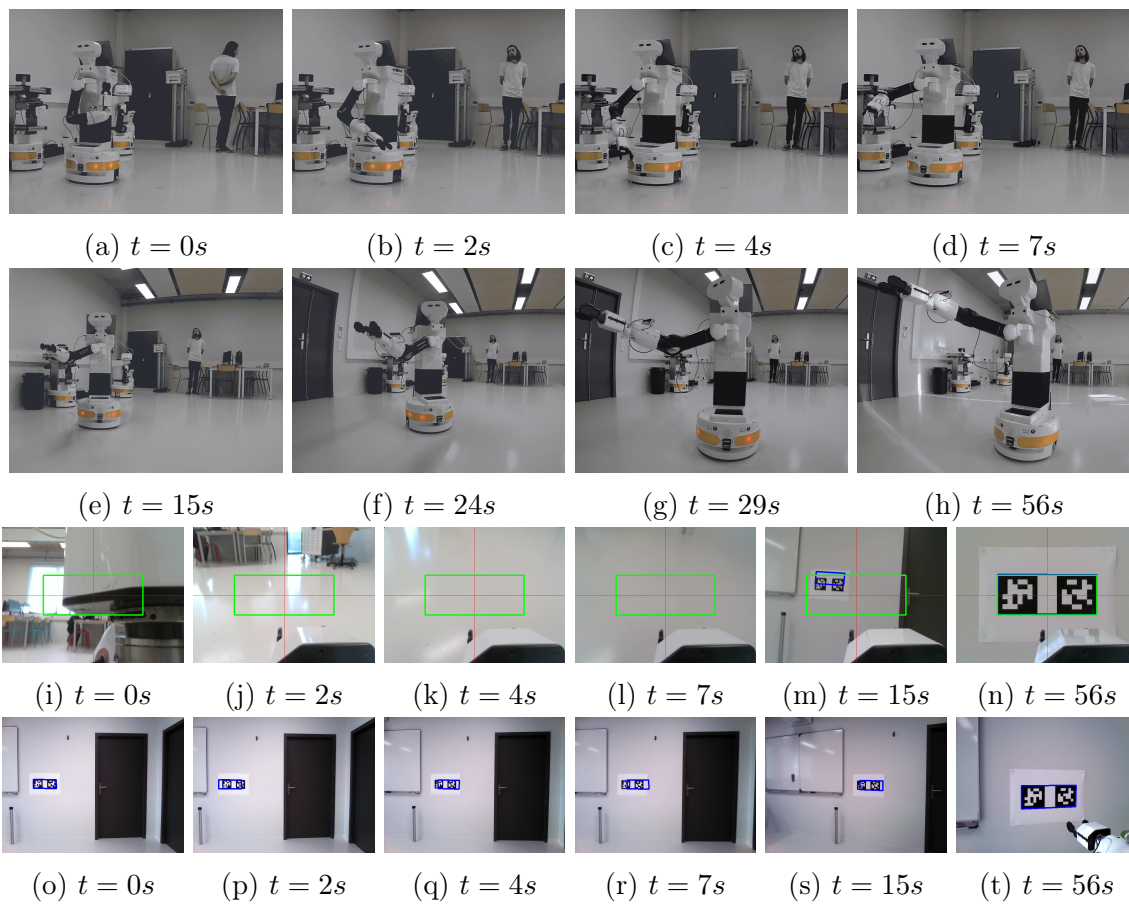


FIGURE 4.33 – Trajectoire du robot (a)-(h), informations de la caméra de l'organe terminal (i)-(n) et informations de la caméra de la tête (o)-(t)

Chapitre 5

Conclusion générale

L'essor actuel de la robotique et les défis auxquels elle est confrontée appelle au développement de systèmes robotisés capables de réaliser des tâches complexes de manière autonome. Les manipulateurs mobiles y jouent un rôle central et un état de l'art a mis en avant l'intérêt des stratégies d'asservissement visuel prédictif dans ce contexte. Ces travaux de thèse se sont intéressés à l'élaboration d'un tel schéma d'asservissement visuel prédictif pour manipulateur mobile.

Un premier schéma d'asservissement visuel prédictif mono-caméra a été présenté dans le chapitre 3 et a servi de première base pour aborder la problématique. La loi de commande combine donc les avantages de l'asservissement visuel 2D et de la commande prédictive. Tout comme l'asservissement visuel 2D, la fonction de coût est définie dans l'image et aucune localisation 3D n'est nécessaire. Le positionnement final après convergence ne dépend alors pas d'étape de reconstruction d'informations 3D. L'utilisation des moments dans l'image est centrale dans ces travaux. Elle a permis notamment de faciliter la résolution du problème d'optimisation grâce à d'excellentes propriétés de découplage et de non-linéarité. Le découplage des indices visuels a également rendu possible une certaine influence sur la trajectoire obtenue. La stabilité en boucle est assurée grâce à l'adaptation de la notion de contrainte terminale dans le contexte étudié. Les efforts d'implémentation ont permis de porter le schéma sur la plateforme robotique réelle et les résultats obtenus sont bons.

Plusieurs axes d'amélioration ont été identifiés et abordés dans un deuxième schéma de commande présenté dans le chapitre 4. Premièrement, l'ajout d'une deuxième caméra permet au robot d'opérer dans des configurations quelconques par rapport à la cible. La caméra de l'organe terminal peut alors se déplacer librement sans être contrainte par les soucis de visibilité et le schéma offre la possibilité de réaliser une gamme de tâche plus variée. Les enjeux soulevés par cette modification nous ont amenés à introduire la méthode de projection sphérique et les moments 3D. Deux versions de ce schéma ont été présentées : une stratégie d'asservissement visuel prédictif 2D et une approche par asservissement visuel prédictif 2D 1/2. Les résultats obtenus avec ces derniers ont des performances similaires et très satisfaisantes. L'utilisation de l'un ou l'autre dépend essentiellement de la disponibilité d'une bonne estimation des informations 3D. Ensuite, la notion de manipulabilité

a été introduite dans le problème d'optimisation afin de favoriser le déplacement du robot dans des configurations maximisant la capacité de déplacement de l'organe terminal. À nouveau, les problèmes soulevés par cet ajout ont été identifiés, puis résolus. Le triptyque de contrainte de positionnement variant dans le temps en constitue la principale solution.

Les différents schémas de commande proposés ont été testés sur la plateforme robotique TIAGo et les résultats obtenus ont montré la pertinence et l'efficacité des stratégies de commande prédictive visuelle pour le contrôle de manipulateurs mobiles.

Les travaux ont permis de mettre en évidence plusieurs axes de recherches futurs sur le sujet. Une première piste déjà envisagée est l'intégration de contraintes d'évitement d'obstacles. La détection et la modélisation des obstacles, mais surtout la modification en ligne du problème d'optimisation, sont des enjeux importants dans ce contexte. En effet, le triptyque de contraintes de positionnement variant dans le temps devra être modifié pour permettre la prise en compte d'obstacles. Il peut devenir nécessaire d'augmenter la longueur de la trajectoire pour les éviter, créant un conflit avec la contrainte de contraction. Un nouveau défi consiste à adapter la contrainte de contraction de la trajectoire pour garantir la stabilité en présence d'obstacles initialement inconnus.

Une deuxième modification du triptyque de contraintes de positionnement aurait pour objectif de réduire le temps de convergence. En l'état, la contraction est uniquement réalisée sur la fin de la trajectoire. Celle-ci peut certainement être également effectuée sur l'ensemble de la trajectoire. Des approches similaires aux travaux de [Durand-Petiteville and Cadenat, 2022] pourraient par exemple être utilisées.

Ensuite, même si le bras robotique réalise la tâche tout en restant loin des singularités et en utilisant des configurations maximisant la manipulabilité, celui-ci se déploie encore trop tôt. Ainsi, un objectif consisterait à initier son mouvement au dernier moment afin de maximiser les déplacements avec le bras replié et ainsi de minimiser les risques de collision. Ceci permettrait également de se rapprocher d'un mouvement humain. Cet aspect peut être abordé à partir de plusieurs points. Une première approche serait de s'appuyer sur la trajectoire prédite au cœur de la commande prédictive. Une analyse de cette dernière peut permettre de calculer l'instant optimal pour le déploiement du bras. Une deuxième idée concerne une nouvelle modification de la fonction coût. En effet, l'ajout d'une contrainte de minimisation de la commande ou de l'énergie semble également être un point de départ intéressant. Une astucieuse pondération des termes au cœur de la fonction coût permettrait certainement d'aborder ce problème.

Comme cela a souvent été souligné, la formulation du problème par commande prédictive permet l'intégration de nombreuses contraintes, et évite une importante refonte de la stratégie de commande pour des scénarios différents. L'ajout d'autres contraintes peut être envisagé. Par exemple, la commande d'une base multi-bras, afin de réaliser des tâches davantage variées, soulève la question du partage de l'espace de travail. Lorsque plusieurs bras robotiques partagent un même un espace, il devient indispensable d'éviter les collisions entre les bras. Pour cela, des contraintes

s'appuyant sur des mesures de distance minimale entre deux corps peuvent être introduites [Le Flecher, 2020].

Un défi similaire concernant la gestion des occultations de la caméra de la tête par le mouvement du bras peut également être rencontré. En effet, ce problème n'a pas été étudié dans ces travaux, mais il peut pourtant mener à la non-réalisation de la tâche dans des scénarios spécifiques. Ce problème peut également être modélisé comme un partage d'espace et une gestion des collisions. Une deuxième approche pourrait à nouveau tirer parti de l'étape de prédiction de la commande pour permettre la gestion d'occultations partielles comme cela a été fait dans [Durand-Petiteville, 2012]. Enfin, une meilleure utilisation de la complémentarité des deux caméras pourrait également permettre de répondre à cet enjeu. En effet, la cible ne doit théoriquement rester visible que par une seule caméra. L'intégration d'une contrainte de disjonction laisserait alors une plus grande liberté au système robotique. Cependant, l'utilisation de solveurs mixtes devient nécessaire et le temps de résolution maximal peut être problématique. Des solutions permettant d'éviter l'utilisation de tels solveurs existent et proposent de relaxer ces contraintes logiques [Wei et al., 2008].

Concernant la tâche de maximisation de la manipulabilité, les mesures utilisées dans ces travaux restent relativement basiques et de nombreux travaux récents proposent de s'appuyer sur des indices plus complexes comme la forme de l'ellipsoïde de manipulabilité [Jaquier et al., 2020]. Les travaux montrent que cette forme fournit plus d'informations sur le mouvement humain que les mesures de manipulabilité classiques.

De plus, la priorisation de la tâche d'asservissement visuel sur la tâche de manipulabilité pourrait être abordée avec d'autres approches. Par exemple, il serait intéressant de comparer notre méthode avec les stratégies s'appuyant sur la jacobienne de la tâche prioritaire [Bayle et al., 2002]. L'aptitude du schéma de contrôle à gérer plusieurs tâches ouvrirait la voie à de nombreuses améliorations. Le centrage de la cible dans la caméra de la tête en est un bon exemple. En effet, la tête n'est pour le moment contrôlée que dans le but de respecter la contrainte de visibilité, mais l'ajout d'une tâche de centrage, moins prioritaire, pourrait être intéressant.

Enfin, l'utilisation de modèles dynamiques devient également nécessaire si des mouvements plus rapides sont considérés. L'intégration de ces modèles pourrait permettre de réduire les vibrations et les à-coups observés. Cependant, cela représenterait une reformulation majeure du schéma de commande, aussi bien sur la modélisation des tâches, des modèles de prédictions et des contraintes.

Liste des publications

- Bildstein, H., Durand-Petiteville, A., & Cadenat, V. (2022, July). **Visual predictive control strategy for mobile manipulators**. In 2022 European Control Conference (ECC) (pp. 1672-1677). IEEE.
- Bildstein, H., Durand-Petiteville, A., & Cadenat, V. (2023, June). **Multi-camera visual predictive control strategy for mobile manipulators**. In 2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM) (pp. 476-482). IEEE.
- Bildstein, H., Durand-Petiteville, A., & Cadenat, V. (2023, September). **Enhanced Visual Predictive Control Scheme for Mobile Manipulator**. In 2023 European Conference on Mobile Robots (ECMR) (pp. 1-7). IEEE.
- Bildstein, H., Cadenat, V., & Durand-Petiteville, A. (2024, October). **Visual Predictive Control for Mobile Manipulator : Visibility, Manipulability, and Stability**. In 2024 Robotics and Autonomous Systems (RAS)

Bibliographie

- [Abdelkader et al., 2005] Abdelkader, H. H., Mezouar, Y., Andreff, N., and Martinet, P. (2005). 2 1/2 d visual servoing with central catadioptric cameras. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3572–3577. IEEE.
- [Alamir, 2018] Alamir, M. (2018). Stability proof for nonlinear mpc design using monotonically increasing weighting profiles without terminal constraints. *Automatica*, 87 :455–459.
- [AlBeladi et al., 2022] AlBeladi, A., Ripperger, E., Hutchinson, S., and Krishnan, G. (2022). Hybrid eye-in-hand/eye-to-hand image based visual servoing for soft continuum arms. *IEEE Robotics and Automation Letters*, 7(4) :11298–11305.
- [Aliakbarpour et al., 2014] Aliakbarpour, H., Tahri, O., and Araujo, H. (2014). Visual servoing of mobile robots using non-central catadioptric cameras. *Robotics and Autonomous Systems*, 62(11) :1613–1622.
- [Allgower et al., 2004] Allgower, F., Findeisen, R., Nagy, Z. K., et al. (2004). Non-linear model predictive control : From theory to application. *Journal-Chinese Institute Of Chemical Engineers*, 35(3) :299–316.
- [Allibert et al., 2010a] Allibert, G., Courtial, E., and Chaumette, F. (2010a). Predictive control for constrained image-based visual servoing. *IEEE Transactions on Robotics*, 26(5) :933–939.
- [Allibert et al., 2010b] Allibert, G., Courtial, E., and Chaumette, F. (2010b). Visual servoing via nonlinear predictive control. In *Visual Servoing via Advanced Numerical Methods*, pages 375–393. Springer.
- [Allibert et al., 2008] Allibert, G., Courtial, E., and Touré, Y. (2008). Visual predictive control for manipulators with catadioptric camera. In *2008 IEEE International Conference on Robotics and Automation*, pages 510–515. IEEE.
- [Andersson et al., 2019] Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., and Diehl, M. (2019). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1) :1–36.
- [Avanzini et al., 2018] Avanzini, G. B., Zanchettin, A. M., and Rocco, P. (2018). Constrained model predictive control for mobile robotic manipulators. *Robotica*, 36(1) :19–38.
- [Bakthavatchalam et al., 2018] Bakthavatchalam, M., Tahri, O., and Chaumette, F. (2018). A direct dense visual servoing approach using photometric moments. *IEEE Transactions on Robotics*, 34(5) :1226–1239.

- [Basri et al., 1999] Basri, R., Rivlin, E., and Shimshoni, I. (1999). Visual homing : Surfing on the epipoles. *International Journal of Computer Vision*, 33 :117–137.
- [Bayle et al., 2002] Bayle, B., Fourquet, J.-Y., Lamiroux, F., and Renaud, M. (2002). Kinematic control of wheeled mobile manipulators. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1572–1577. IEEE.
- [Bayle et al., 2001] Bayle, B., Fourquet, J.-Y., and Renaud, M. (2001). Manipulability analysis for mobile manipulators. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 2, pages 1251–1256. IEEE.
- [Bayle et al., 2003] Bayle, B., Fourquet, J.-Y., and Renaud, M. (2003). Manipulability of wheeled mobile manipulators : Application to motion generation. *The International Journal of Robotics Research*, 22(7-8) :565–581.
- [Benhimane and Malis, 2006] Benhimane, S. and Malis, E. (2006). Homography-based 2d visual servoing. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2397–2402. IEEE.
- [Caron et al., 2013] Caron, G., Marchand, E., and Mouaddib, E. M. (2013). Photometric visual servoing for omnidirectional cameras. *Autonomous Robots*, 35 :177–193.
- [Carpentier et al., 2021] Carpentier, J., Valenza, F., Mansard, N., et al. (2015–2021). Pinocchio : fast forward and inverse dynamics for poly-articulated systems. <https://stack-of-tasks.github.io/pinocchio>.
- [Chang and Shao, 2010] Chang, W.-C. and Shao, C.-K. (2010). Hybrid eye-to-hand and eye-in-hand visual servoing for autonomous robotic manipulation. In *Proceedings of SICE Annual Conference 2010*, pages 415–422. IEEE.
- [Chaumette, 1990] Chaumette, F. (1990). *La relation vision-commande : théorie et application à des taches robotiques*. PhD thesis.
- [Chaumette, 2004] Chaumette, F. (2004). Image moments : a general and useful set of features for visual servoing. *IEEE Transactions on Robotics*, 20(4) :713–723.
- [Chaumette, 2007] Chaumette, F. (2007). Potential problems of stability and convergence in image-based and position-based visual servoing. In *The confluence of vision and control*, pages 66–78. Springer.
- [Chaumette and Hutchinson, 2006] Chaumette, F. and Hutchinson, S. (2006). Visual servo control. i. basic approaches. *IEEE Robotics & Automation Magazine*, 13(4) :82–90.
- [Chaumette and Hutchinson, 2007] Chaumette, F. and Hutchinson, S. (2007). Visual servo control, part ii : Advanced approaches. *IEEE Robotics and Automation Magazine*, 14(1) :109–118.
- [Chen et al., 2018] Chen, F., Selvaggio, M., and Caldwell, D. G. (2018). Dexterous grasping by manipulability selection for mobile manipulator with visual guidance. *IEEE Transactions on Industrial Informatics*, 15(2) :1202–1210.

- [Cherubini and Chaumette, 2011] Cherubini, A. and Chaumette, F. (2011). Visual navigation with obstacle avoidance. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1593–1598. IEEE.
- [Choset et al., 2005] Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., and Burgard, W. (2005). *Principles of robot motion : theory, algorithms, and implementations*. MIT press.
- [Collewet and Marchand, 2011] Collewet, C. and Marchand, E. (2011). Photometric visual servoing. *IEEE Transactions on Robotics*, 27(4) :828–834.
- [Colombo et al., 2019] Colombo, R., Gennari, F., Annem, V., Rajendran, P., Thakkar, S., Bascetta, L., and Gupta, S. K. (2019). Parameterized model predictive control of a nonholonomic mobile manipulator : A terminal constraint-free approach. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pages 1437–1442. IEEE.
- [Colucci et al., 2022] Colucci, G., Botta, A., Tagliavini, L., Cavallone, P., Baglieri, L., and Quaglia, G. (2022). Kinematic modeling and motion planning of the mobile manipulator agri. q for precision agriculture. *Machines*, 10(5) :321.
- [Cong and Hanh, 2022] Cong, V. D. and Hanh, L. D. (2022). A new decoupled control law for image-based visual servoing control of robot manipulators. *International Journal of Intelligent Robotics and Applications*, 6(3) :576–585.
- [Copot et al., 2012] Copot, C., Lazar, C., and Burlacu, A. (2012). Predictive control of nonlinear visual servoing systems using image moments. *IET control theory & applications*, 6(10) :1486–1496.
- [Corke et al., 2000] Corke, P., Trevelyan, J., Morel, G., Liebezeit, T., Szewczyk, J., Boudet, S., and Pot, J. (2000). Explicit incorporation of 2d constraints in vision based control of robot manipulators. In *Experimental Robotics VI*, pages 99–108. Springer.
- [Corke and Hutchinson, 2001] Corke, P. I. and Hutchinson, S. A. (2001). A new partitioned approach to image-based visual servo control. *IEEE Transactions on Robotics and Automation*, 17(4) :507–515.
- [Crombez et al., 2018] Crombez, N., Mouaddib, E. M., Caron, G., and Chaumette, F. (2018). Visual servoing with photometric gaussian mixtures as dense features. *IEEE Transactions on Robotics*, 35(1) :49–63.
- [Cuevas-Velasquez et al., 2018] Cuevas-Velasquez, H., Li, N., Tylecek, R., Saval-Calvo, M., and Fisher, R. B. (2018). Hybrid multi-camera visual servoing to moving target. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1132–1137. IEEE.
- [Diehl and Gros, 2011] Diehl, M. and Gros, S. (2011). Numerical optimal control. *Optimization in Engineering Center (OPTEC)*.
- [Duffy, 2003] Duffy, B. R. (2003). Anthropomorphism and the social robot. *Robotics and autonomous systems*, 42(3-4) :177–190.

- [Dune et al., 2007] Dune, C., Marchand, E., and Leroux, C. (2007). One click focus with eye-in-hand/eye-to-hand cooperation. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 2471–2476. IEEE.
- [Durand-Petiteville, 2012] Durand-Petiteville, A. (2012). *Navigation référencée multi-capteurs d’un robot mobile en environnement encombré*. Theses, Université Paul Sabatier - Toulouse III.
- [Durand-Petiteville and Cadenat, 2022] Durand-Petiteville, A. and Cadenat, V. (2022). Advanced visual predictive control scheme for the navigation problem. *Journal of Intelligent & Robotic Systems*, 105(2) :1–21.
- [Espiau et al., 1992] Espiau, B., Chaumette, F., and Rives, P. (1992). A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3) :313–326.
- [Fard, 2019] Fard, B. M. (2019). A manipulability analysis of human walking. *Journal of biomechanics*, 83 :157–164.
- [Flandin et al., 2000] Flandin, G., Chaumette, F., and Marchand, E. (2000). Eye-in-hand/eye-to-hand cooperation for visual servoing. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 3, pages 2741–2746. IEEE.
- [Fleurmond, 2015] Fleurmond, R. (2015). *Asservissement visuel coordonné de deux bras manipulateurs*. PhD thesis, Université Toulouse III Paul Sabatier.
- [Fong et al., 2002] Fong, T., Nourbakhsh, I., and Dautenhahn, K. (2002). A survey of socially interactive robots : Concepts, design and applications.
- [Fusco et al., 2020] Fusco, F., Kermorgant, O., and Martinet, P. (2020). Integrating features acceleration in visual predictive control. *IEEE Robotics and Automation Letters*, 5(4) :5197–5204.
- [Gao et al., 2021] Gao, J., Liang, X., Chen, Y., Zhang, L., and Jia, S. (2021). Hierarchical image-based visual serving of underwater vehicle manipulator systems based on model predictive control and active disturbance rejection control. *Ocean Engineering*, 229 :108814.
- [Gielniak et al., 2013] Gielniak, M. J., Liu, C. K., and Thomaz, A. L. (2013). Generating human-like motion for robots. *The International journal of robotics research*, 32(11) :1275–1301.
- [Gifftthaler et al., 2017] Gifftthaler, M., Farshidian, F., Sandy, T., Stadelmann, L., and Buchli, J. (2017). Efficient kinematic planning for mobile manipulators with non-holonomic constraints using optimal control. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3411–3417. IEEE.
- [Grüne et al., 2017] Grüne, L., Pannek, J., Grüne, L., and Pannek, J. (2017). *Non-linear model predictive control*. Springer.
- [Gulletta et al., 2021] Gulletta, G., Silva, E. C. e., Erlhagen, W., Meulenbroek, R., Costa, M. F. P., and Bicho, E. (2021). A human-like upper-limb motion planner :

- Generating naturalistic movements for humanoid robots. *International Journal of Advanced Robotic Systems*, 18(2) :1729881421998585.
- [Heshmati-Alamdari et al., 2020] Heshmati-Alamdari, S., Eqtami, A., Karras, G. C., Dimarogonas, D. V., and Kyriakopoulos, K. J. (2020). A self-triggered position based visual servoing model predictive control scheme for underwater robotic vehicles. *Machines*, 8(2) :33.
- [Heshmati-Alamdari et al., 2014] Heshmati-Alamdari, S., Karavas, G. K., Eqtami, A., Drossakis, M., and Kyriakopoulos, K. J. (2014). Robustness analysis of model predictive control for constrained image-based visual servoing. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4469–4474. IEEE.
- [Hu, 1962] Hu, M.-K. (1962). Visual pattern recognition by moment invariants. *IRE transactions on information theory*, 8(2) :179–187.
- [James et al., 2017] James, S., Davison, A. J., and Johns, E. (2017). Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task. In *Conference on Robot Learning*, pages 334–343. PMLR.
- [Jaquier et al., 2020] Jaquier, N., Rozo, L., and Calinon, S. (2020). Analysis and transfer of human movement manipulability in industry-like activities. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11131–11138. IEEE.
- [Johnson, 2007] Johnson, S. G. (2007). The NLOpt nonlinear-optimization package. <https://github.com/stevengj/nlopt>.
- [Kazemi et al., 2009] Kazemi, M., Gupta, K., and Mehrandezh, M. (2009). Global path planning for robust visual servoing in complex environments. In *2009 IEEE international conference on robotics and automation*, pages 326–332. IEEE.
- [Kermorgant and Chaumette, 2011] Kermorgant, O. and Chaumette, F. (2011). Multi-sensor data fusion in sensor-based control : Application to multi-camera visual servoing. In *2011 IEEE International Conference on Robotics and Automation*, pages 4518–4523. IEEE.
- [Kermorgant and Chaumette, 2013] Kermorgant, O. and Chaumette, F. (2013). Dealing with constraints in sensor-based robot control. *IEEE Transactions on Robotics*, 30(1) :244–257.
- [Khomutenko et al., 2015] Khomutenko, B., Garcia, G., and Martinet, P. (2015). An enhanced unified camera model. *IEEE Robotics and Automation Letters*, 1(1) :137–144.
- [Klein and Blaho, 1987] Klein, C. A. and Blaho, B. E. (1987). Dexterity measures for the design and control of kinematically redundant manipulators. *The international journal of robotics research*, 6(2) :72–83.
- [Kober et al., 2013] Kober, J., Bagnell, J. A., and Peters, J. (2013). Reinforcement learning in robotics : A survey. *The International Journal of Robotics Research*, 32(11) :1238–1274.

- [Kocić et al., 2019] Kocić, J., Jovičić, N., and Drndarević, V. (2019). An end-to-end deep neural network for autonomous driving designed for embedded automotive platforms. *Sensors*, 19(9) :2064.
- [Koenig and Likhachev, 2005] Koenig, S. and Likhachev, M. (2005). Fast replanning for navigation in unknown terrain. *IEEE Transactions on Robotics*, 21(3) :354–363.
- [Lampe and Riedmiller, 2013] Lampe, T. and Riedmiller, M. (2013). Acquiring visual servoing reaching and grasping skills using neural reinforcement learning. In *The 2013 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE.
- [Le Flecher, 2020] Le Flecher, E. (2020). *Coordination des mouvements d’un système mobile bi-bras pour la réalisation de tâches complexes de navigation et de manipulation dans un environnement fortement dynamique*. Theses, Université Paul Sabatier - Toulouse III.
- [Leca, 2021] Leca, D. (2021). *Navigation autonome d’un robot agricole*. PhD thesis, Université Paul Sabatier-Toulouse III.
- [Logothetis et al., 2018] Logothetis, M., Karras, G. C., Heshmati-Alamdari, S., Vlantis, P., and Kyriakopoulos, K. J. (2018). A model predictive control approach for vision-based object grasping via mobile manipulator. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–6. IEEE.
- [Machkour et al., 2022] Machkour, Z., Ortiz-Arroyo, D., and Durdevic, P. (2022). Classical and deep learning based visual servoing systems : a survey on state of the art. *Journal of Intelligent & Robotic Systems*, 104(1) :11.
- [Magassouba, 2016] Magassouba, A. (2016). *Aural servo : towards an alternative approach to sound localization for robot motion control*. PhD thesis, Rennes 1.
- [Mahony et al., 2002] Mahony, R., Corke, P., and Chaumette, F. (2002). Choice of image features for depth-axis control in image based visual servo control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 390–395. IEEE.
- [Malis et al., 1999] Malis, E., Chaumette, F., and Boudet, S. (1999). 2 1/2 d visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2) :238–250.
- [Marchand, 2020] Marchand, E. (2020). Direct visual servoing in the frequency domain. *IEEE Robotics and Automation Letters*, 5(2) :620–627.
- [Marie et al., 2019] Marie, R., Said, H. B., Stéphant, J., and Labbani-Igbida, O. (2019). Visual servoing on the generalized voronoi diagram using an omnidirectional camera. *Journal of Intelligent & Robotic Systems*, 94 :793–804.
- [Martínez et al., 2013] Martínez, S. S., Ortega, J. G., Garcia, J. G., Garcia, A. S., and de la Casa Cárdenas, J. (2013). Visual predictive control of robot manipulators using a 3d tof camera. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 3657–3662. IEEE.
- [Mcfadyen et al., 2014] Mcfadyen, A., Corke, P., and Mejias, L. (2014). Visual predictive control of spiral motion. *IEEE Transactions on Robotics*, 30(6) :1441–1454.

- [Mezouar et al., 2004] Mezouar, Y., Abdelkader, H. H., Martinet, P., and Chaumette, F. (2004). Central catadioptric visual servoing from 3d straight lines. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 1, pages 343–348. IEEE.
- [Michalska and Mayne, 1993] Michalska, H. and Mayne, D. Q. (1993). Robust receding horizon control of constrained nonlinear systems. *IEEE transactions on automatic control*, 38(11) :1623–1633.
- [Minniti et al., 2019] Minniti, M. V., Farshidian, F., Grandia, R., and Hutter, M. (2019). Whole-body mpc for a dynamically stable mobile manipulator. *IEEE Robotics and Automation Letters*, 4(4) :3687–3694.
- [Mohamed et al., 2021] Mohamed, I. S., Allibert, G., and Martinet, P. (2021). Sampling-based mpc for constrained vision based control. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3753–3758. IEEE.
- [Mohammad Hossein Fallah and Janabi-Sharifi, 2021] Mohammad Hossein Fallah, M. and Janabi-Sharifi, F. (2021). Conjugated visual predictive control for constrained visual servoing. *Journal of Intelligent & Robotic Systems*, 101 :1–21.
- [Muller et al., 2005] Muller, U., Ben, J., Cosatto, E., Flepp, B., and Cun, Y. (2005). Off-road obstacle avoidance through end-to-end learning. *Advances in neural information processing systems*, 18.
- [Nelson and Khosla, 1995] Nelson, B. J. and Khosla, P. K. (1995). Strategies for increasing the tracking region of an eye-in-hand system by singularity and joint limit avoidance. *The International journal of robotics research*, 14(3) :255–269.
- [Norouzi-Ghazbi et al., 2021] Norouzi-Ghazbi, S., Mehrkish, A., Fallah, M. M., and Janabi-Sharifi, F. (2021). Constrained visual predictive control of tendon-driven continuum robots. *Robotics and Autonomous Systems*, 145 :103856.
- [Ourak et al., 2016] Ourak, M., Tamadazte, B., Lehmann, O., and Andreff, N. (2016). Wavelets-based 6 dof visual servoing. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3414–3419. IEEE.
- [Pankert and Hutter, 2020] Pankert, J. and Hutter, M. (2020). Perceptive model predictive control for continuous mobile manipulation. *IEEE Robotics and Automation Letters*, 5(4) :6177–6184.
- [Pérez-Morales et al., 2021] Pérez-Morales, D., Kermorgant, O., Domínguez-Quijada, S., and Martinet, P. (2021). Multisensor-based predictive control for autonomous parking. *IEEE Transactions on Robotics*, 38(2) :835–851.
- [Petiteville and Cadenat, 2014] Petiteville, A. D. and Cadenat, V. (2014). An anticipative reactive control strategy to deal with unforeseen obstacles during a multi-sensor-based navigation task. In *2014 European Control Conference (ECC)*, pages 2816–2821. IEEE.
- [Picard, 2022] Picard, G. (2022). *Approche générique pour la coordination des manipulateurs mobiles en milieu naturel*. PhD thesis, Université Clermont Auvergne (2021-...).

- [Prakash and Behera, 2023] Prakash, R. and Behera, L. (2023). Neural optimal control for constrained visual servoing via learning from demonstration. *IEEE Transactions on Automation Science and Engineering*.
- [Prokop and Reeves, 1992] Prokop, R. J. and Reeves, A. P. (1992). A survey of moment-based techniques for unoccluded object representation and recognition. *CVGIP : Graphical Models and Image Processing*, 54(5) :438–460.
- [Redmon et al., 2016] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once : Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788.
- [Sadeghi, 2019] Sadeghi, F. (2019). Divis : Domain invariant visual servoing for collision-free goal reaching. *arXiv preprint arXiv :1902.05947*.
- [Sastry, 1999] Sastry, S. (1999). *Nonlinear systems : Analysis, Stability, and Control*. Springer Science & Business Media.
- [Sciavicco and Siciliano, 2012] Sciavicco, L. and Siciliano, B. (2012). *Modelling and control of robot manipulators*. Springer Science & Business Media.
- [Scokaert et al., 1999] Scokaert, P. O., Mayne, D. Q., and Rawlings, J. B. (1999). Suboptimal model predictive control (feasibility implies stability). *IEEE Transactions on Automatic Control*, 44(3) :648–654.
- [Siciliano et al., 2008] Siciliano, B., Khatib, O., and Kröger, T. (2008). *Springer handbook of robotics*, volume 200. Springer.
- [Silveira and Malis, 2007] Silveira, G. and Malis, E. (2007). Direct visual servoing with respect to rigid objects. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1963–1968. IEEE.
- [Silveira and Malis, 2012] Silveira, G. and Malis, E. (2012). Direct visual servoing : Vision-based estimation and control using only nonmetric information. *IEEE Transactions on Robotics*, 28(4) :974–980.
- [Sola et al., 2018] Sola, J., Deray, J., and Atchuthan, D. (2018). A micro lie theory for state estimation in robotics. *arXiv preprint arXiv :1812.01537*.
- [Steger, 1996] Steger, C. (1996). On the calculation of arbitrary moments of polygons. *Munich Univ., Munich, Germany, Tech. Rep. FGBV-96-05*.
- [Tahri, 2004] Tahri, O. (2004). Application des moments à l’asservissement visuel et au calcul de pose. *These de doctorat, Université de Rennes*, 1.
- [Tahri and Chaumette, 2003] Tahri, O. and Chaumette, F. (2003). *Determination of moment invariants and their application to visual servoing*. PhD thesis, INRIA.
- [Tahri et al., 2008] Tahri, O., Chaumette, F., and Mezouar, Y. (2008). New decoupled visual servoing scheme based on invariants from projection onto a sphere. In *2008 IEEE International Conference on Robotics and Automation*, pages 3238–3243. IEEE.
- [Tatsambon Fomena, 2008] Tatsambon Fomena, R. (2008). *Asservissement visuel par projection sphérique*. PhD thesis, Rennes 1.

- [Tsai, 1986] Tsai, M. (1986). *Workspace Geometric Characterization of Industrial Robot*. PhD thesis, Ph. D. Thesis, Ohio State University, Department of Mechanical Engineering.
- [Wang et al., 2020] Wang, C., Zhang, Q., Tian, Q., Li, S., Wang, X., Lane, D., Petillot, Y., and Wang, S. (2020). Learning mobile manipulation through deep reinforcement learning. *Sensors*, 20(3) :939.
- [Wang and Olson, 2016] Wang, J. and Olson, E. (2016). Apriltag 2 : Efficient and robust fiducial detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4193–4198. IEEE.
- [Wei et al., 2008] Wei, S., Zefran, M., and DeCarlo, R. A. (2008). Optimal control of robotic systems with logical constraints : Application to uav path planning. In *2008 IEEE International Conference on Robotics and Automation*, pages 176–181. IEEE.
- [Wu et al., 2022] Wu, J., Jin, Z., Liu, A., Yu, L., and Yang, F. (2022). A hybrid deep-q-network and model predictive control for point stabilization of visual servoing systems. *Control Engineering Practice*, 128 :105314.
- [Xing et al., 2021] Xing, H., Torabi, A., Ding, L., Gao, H., Li, W., and Tavakoli, M. (2021). Enhancing kinematic accuracy of redundant wheeled mobile manipulators via adaptive motion planning. *Mechatronics*, 79 :102639.
- [Yang et al., 2022] Yang, W., Sundaralingam, B., Paxton, C., Akinola, I., Chao, Y.-W., Cakmak, M., and Fox, D. (2022). Model predictive control for fluid human-to-robot handovers. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6956–6962. IEEE.
- [Yoshikawa, 1985a] Yoshikawa, T. (1985a). Dynamic manipulability of robot manipulators. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 1033–1038.
- [Yoshikawa, 1985b] Yoshikawa, T. (1985b). Manipulability of robotic mechanisms. *The international journal of Robotics Research*, 4(2) :3–9.
- [Zhang et al., 2021] Zhang, K., Shi, Y., and Sheng, H. (2021). Robust nonlinear model predictive control based visual servoing of quadrotor uavs. *IEEE/ASME Transactions on Mechatronics*, 26(2) :700–708.
- [Zhao et al., 2023] Zhao, J., Wang, C., and Xie, B. (2023). Human-like motion planning of robotic arms based on human arm motion patterns. *Robotica*, 41(1) :259–276.