

Thèse de Doctorat

Sylvain VANDERNOTTE

*Mémoire présenté en vue de l'obtention du
grade de Docteur de l'École centrale de Nantes
sous le label de l'Université de Nantes Angers Le Mans*

École doctorale : Sciences et Technologies de l'Information et Mathématiques

Discipline : Automatique, Productique et Robotique

Unité de recherche : Institut de Recherche en Communications et Cybernétique de Nantes

Soutenance prévue le 29 juin 2016

Manipulation référencée multi-capteurs d'objets manufacturés

JURY

Président :	M. Yannick Aoustin , Professeur des universités, Université de Nantes, IRCCyN, Nantes
Rapporteurs :	M^{me} Véronique PERDEREAU , Professeur des universités, Université Pierre et Marie Curie, Paris M. Nicolas ANDREFF , Professeur des universités, Université de Franche-Comté, Besançon
Invités :	M. Alexis GIRIN , Docteur, IRT Jules Verne M. Patrice RABATÉ , Ingénieur, Airbus Group
Directeur de thèse :	M. Philippe MARTINET , Professeur des universités, École Centrale de Nantes, IRCCyN, Nantes
Co-encadrants de thèse :	M. Abdelhamid CHRIETTE , Maître de Conférences, École Centrale de Nantes, IRCCyN, Nantes M. Adolfo SUAREZ-ROOS , Docteur, Airbus Group, Suresnes

Remerciements

Je remercie tout d'abord les membres du jury pour s'être intéressés à mon travail. Merci à M. Yannick Aoustin pour avoir présidé ce jury. Je remercie Mme Véronique Perdereau et M. Nicolas Andreff pour avoir accepté la charge de rapporteur. Leurs remarques m'ont permis d'identifier les points à approfondir et d'améliorer mon travail d'une manière globale. J'adresse une pensée chaleureuse à Mme Perdereau qui a participé à mon comité de suivi de thèse et suivi mon parcours tout au long de ces trois années. J'adresse cette même pensée à M. Aoustin qui m'a orienté vers le Master ASP qui m'a conduit à cette thèse.

Je remercie tout spécialement M. Philippe Martinet pour avoir dirigé mes travaux. Je te remercie pour m'avoir fait confiance et permis d'avancer toujours avec la rigueur scientifique qui me faisait parfois défaut. Un grand merci à M. Abdelhamid Chriette pour avoir accepté la charge d'encadrant et pour m'avoir épaulé surtout dans cette dernière période.

Je remercie les membres de l'équipe d'Airbus Group Innovations pour m'avoir fait confiance et pour m'avoir soutenu durant toutes les phases du projet. Je remercie M. Adolfo Suarez Roos pour avoir accepté d'être co-encadrant pour cette thèse. Un grand merci à M. Patrice Rabaté pour m'avoir soutenu. Merci à M. Alexandre Rivière qui a su me transmettre la rigueur nécessaire pour une programmation de qualité. Merci enfin à M. Yann-Henri Laudrain et M. Perig Le-Henaff pour avoir partagé leur enthousiasme à chaque rencontre.

Même si ma participation à la construction de l'IRT a été limitée, j'y ai croisé beaucoup de personnes que je n'oublierai pas. Un Merci tout spécial aux cinq qui ont vécu l'aventure d'une thèse à l'IRT : David G., Khalid, Thomas, Lorenzo et Rémy. J'espère que cette expérience a été aussi bénéfique pour vous que pour moi. J'adresse une pensée toute spéciale aux autres Doctorants IRT qui ont aussi partagé cette aventure et à Serge pour nous avoir fait prendre un peu de recul.

Merci aux membres de l'équipe robotique de l'IRT : Alexis, Olivier K., David MG., Laurianne, Philip, Paul, Iñigo, Étienne. Merci également à Olivier F., Xavier, Sébastien G., Arnaud, Olivier D. le Fabmanager, Anne-Sophie, Marie, Herveline, Linda, Sébastien L.. Grâce à vous, je me suis découvert un talent pour le babyfoot.

J'adresse un grand Merci à ma famille sans qui je n'y serais pas arrivé. Un merci tout spécial à ma chère et tendre future femme qui a toujours été là pour moi. Merci à mon père et à Joël pour être venu à ma soutenance. Une pensée pour ma mère qui n'a pas pu y assister.

Table des matières

Remerciements	i
Table des matières	v
Table des figures	viii
Notations	ix
1 Introduction	1
1.1 Opérateurs et industrie manufacturière	1
1.2 Détail d’une opération d’assemblage	4
1.2.1 Point de vue opérateur	4
1.2.2 Point de vue robotique	6
1.3 Assemblage et robots	10
1.4 Problématique	14
1.5 Approche proposée	17
1.6 Organisation du manuscrit	19
2 Positionnement sans contact	21
2.1 Commande référencée capteurs : état de l’art	21
2.1.1 Modélisation classique de l’information capteur	22
2.1.2 Commande cinématique à partir d’informations visuelles	23
2.1.3 Exemple de la construction d’une tâche d’asservissement visuel	28
2.1.4 Caractéristiques géométriques pour l’assemblage	30
2.2 Commande référencée multi-capteurs et multi-tâches	35
2.2.1 Sources multiples	35
2.2.2 Tâches multiples	36
2.3 Schéma de commande générique référencé multi-capteurs	40
2.3.1 Choix des caractéristiques et définition de la relation spatiale	40
2.3.2 Correspondance expectations et observations	44
2.3.3 Définition des tâches robotiques	45
2.3.4 Application et surveillance de l’exécution	47
2.3.5 Critique	47
2.4 Application pour le positionnement d’une pièce	48
2.4.1 Contexte	48
2.4.2 Définition des caractéristiques de distance	49
2.4.3 Construction de l’opération de positionnement	51
2.4.4 Preuve de concept par simulation	53

2.5	Conclusions	57
3	Commande dynamique référencée multi-capteurs	59
3.1	État de l'art : commandes dynamiques	60
3.1.1	Généralités	60
3.1.2	Commandes classiques	60
3.1.3	Commande dynamique multi-tâches	65
3.1.4	Asservissement visuel cinématique par vision rapide et découplage dynamique	65
3.1.5	Conclusions	66
3.2	Commande dynamique référencée capteurs	67
3.2.1	Obtention du modèle cinématique d'ordre 2	67
3.2.2	Matrices d'interaction pour le modèle du second ordre	67
3.2.3	Commande	72
3.2.4	Critique	76
3.3	Validation de la commande par simulation	77
3.3.1	Suivi de trajectoire dans l'espace capteur	77
3.3.2	Robustesse au bruit de mesure	80
3.3.3	Singularités du KUKA LWR4+	83
3.3.4	Erreur de modélisation géométrique et dynamique du robot	85
3.4	Conclusions	89
4	Commandes hybrides	91
4.1	État de l'art	91
4.1.1	Commande en impédance	91
4.1.2	Commande hybride parallèle	93
4.1.3	Commande hybride externe	96
4.1.4	Critiques	97
4.2	Commande dynamique référencée capteurs hybride	98
4.2.1	Développements	98
4.2.2	Remarques préliminaires	101
4.3	Une étude de cas : positionnement d'une pièce encollée	101
4.3.1	Mise en place de la simulation	102
4.3.2	Marqueur sur structure et contrôle d'effort direct	102
4.3.3	Contrôle en effort indirect : commande hybride externe	106
4.4	Conclusions	107
5	Conclusion générale	111
	Annexes	115
A	Matrices d'interaction pour le modèle du second ordre	117
A.1	Rappel de mécanique générale	117
A.1.1	Dérivation composée	117
A.1.2	Loi de composition des vitesses	119
A.1.3	Loi de composition des accélérations	119
A.2	Ligne 3D	120
A.2.1	Représentation	120

A.2.2	Double dérivée des coordonnées de Plücker	121
A.3	Plan 3D	124
A.3.1	Représentation	124
A.3.2	Double dérivée de la représentation point vecteur	125
A.4	Point 2D	127
A.4.1	Représentation	127
A.4.2	Modèle du second ordre pour le point 2D	127
A.5	Ligne 2D	129
A.5.1	Représentation	129
A.5.2	Matrice d'interaction pour le modèle du premier ordre	130
A.5.3	Hessien pour le modèle du second ordre	131
B	Étude des erreurs de saisie	133
B.1	Description de l'expérimentation	133
B.2	Quatre phases de l'opération	136
B.3	Résultats de l'étude	136
C	Développement des briques logicielles	141
	Bibliographie	143

Table des figures

1.1	Lignes d'assemblage en 1917 et 2008	3
1.2	Lignes robotisées : palettisation et soudage	7
1.3	Étapes d'une manipulation robotique classique en milieu industriel	8
1.4	Séquence d'assemblage [Stein et al., 2011] et expérimentation Schoen and Rus [2013]	11
1.5	Illustration d'assemblage par un système double bras manipulateur You et al. [2012]	12
1.6	Diagramme pour la génération d'une trajectoire dans une expérience classique de prise et de dépose d'une pièce (<i>pick and place</i>) dans un cadre d'automatisation .	16
1.7	Nouveau cadre de programmation robotique se basant sur l'inclusion des caractéristiques du monde réel dans la loi de commande du robot	18
2.1	Schéma d'illustration pour un asservissement visuel 2D avec quatre points sur une cible.	29
2.2	Exemple d'un asservissement visuel 2D typique avec une caméra embarquée sur l'effecteur du robot [Kermorgant, 2011].	29
2.3	Exemple d'assemblage dans une maquette numérique	41
2.4	Explosion des modèles polyédriques	43
2.5	Exemple du mécanisme d'association de mesures	45
2.6	Illustration du cas d'étude pour le positionnement d'une équerre pour le renfort d'une étagère.	49
2.7	Définition des mesures de distance	50
2.8	Sélection des contraintes pour l'opération de positionnement	52
2.9	Simulation du positionnement précis	53
2.10	Résultats de simulation pour le positionnement précis avec le logiciel Gazebo . .	55
2.11	Utilisation de la redondance du LWR4+ pour s'éloigner de la structure	56
3.1	Schéma bloc illustrant une commande en boucle fermée	61
3.2	Schéma bloc illustrant la commande par couple calculé	63
3.3	Schéma bloc illustrant la commande par couples calculés dans l'espace cartésien .	64
3.4	Schéma bloc illustrant la commande référencée capteurs dynamique.	75
3.5	Suivi de trajectoire dans l'espace capteur : trajectoire désirée	78
3.6	Suivi de trajectoire dans l'espace capteur : image et erreurs	79
3.7	Suivi de trajectoire en cercle sans bruit de mesure : trajectoires désirées	81
3.8	Suivi de trajectoire en cercle avec bruit de mesure : erreurs	82
3.9	Suivi de trajectoire en cercle avec bruit de mesure : trajectoires cartésiennes . . .	82
3.10	Exemples de configurations singulières du KUKA LWR4+ correspondant aux cas où le déterminant s'annule.	84

3.11	Comportement de la commande dynamique référencée capteurs au voisinage d'une singularité robot	85
3.12	Comportement de la commande dynamique référencée capteurs au voisinage de la singularité robot : évolutions temporelles	86
3.13	Évolution temporelle des termes d'excitation de l'équation différentielle du système en boucle fermée et l'impact correspondant sur l'erreur des caractéristiques.	88
4.1	Schéma bloc illustrant la commande en impédance sans retour d'information d'effort [Khalil and Dombre, 2002].	93
4.2	Schéma bloc illustrant la commande hybride parallèle utilisant la commande par couple calculé en tant qu'asservissement spatial [Khatib, 1987].	95
4.3	Schéma bloc illustrant la commande hybride externe utilisant la commande par couple calculé avec un régulateur type proportionnel dérivée en position pour la boucle interne et un régulateur proportionnel intégral pour la consigne d'effort [Khalil and Dombre, 2002].	97
4.4	Schéma bloc illustrant la commande hybride externe utilisant une commande référencée capteurs basée image en tant que boucle d'asservissement interne [Prats et al., 2007].	97
4.5	Prépositionnement et positionnement au contact d'une pièce encollée en se basant sur un marqueur présent sur la structure comportant quatre points	103
4.6	Commande hybride parallèle : évolutions temporelles des mesures pour l'asservissement spatial et l'asservissement en effort au cours de la troisième phase . . .	105
4.7	Commande hybride parallèle : comportement du robot	105
4.8	Commande hybride externe : évolutions temporelles des mesures pour l'asservissement spatial et l'asservissement en effort au cours de la troisième phase	107
4.9	Commande hybride externe : comportement du robot	108
B.1	Vue d'ensemble de l'expérience pour la mesure de la répétabilité d'une saisie. . .	134
B.2	Configurations du doigt gauche du préhenseur et champs de pression	135
B.3	Prises de vue depuis la caméra déportée avec les repères de l'objet et du préhenseur en superposition	137

Notations

Règles générales

x	: scalaire
$\mathbf{x}$	: vecteur
$\mathbf{A}$	: matrice
O	: étiquette d'un objet dans l'espace (par exemple : un point, un repère)
$\vec{u}$	: vecteur
$\overrightarrow{OA}$	: vecteur du point O vers le point A
$\overrightarrow{OA} _F$	: vecteur de O vers A , exprimé dans la base (repère) F

Algèbre

$\mathbb{R}^n$	: espace vectoriel de dimension n composé de nombres réels
$SO(3)$	: groupe spécial orthogonal des rotations dans $\mathbb{R}^3$
$SE(3)$	: groupe spécial euclidien des transformations affines dans $\mathbb{R}^3$
$\mathfrak{so}(3), \mathfrak{se}(3)$	: algèbre de Lie associé au groupe $SO(3)$, resp. $SE(3)$
$\dot{\mathbf{x}}, \ddot{\mathbf{x}}$	: dérivée de $\mathbf{x}$, dérivée seconde de $\mathbf{x}$
$\hat{x}, \hat{\mathbf{x}}, \hat{\mathbf{A}}$	: estimé de x , resp. de $\mathbf{x}$, de $\mathbf{A}$
$\mathbf{x}^\top, \mathbf{A}^\top$	: transposé de $\mathbf{x}$, resp. $\mathbf{A}$
$[\mathbf{x}]_\times$	: matrice de préproduit vectoriel associé à $\mathbf{x}$
$\mathbf{A}^{-1}$	: matrice inverse de $\mathbf{A}$
$\mathbf{A}^{(-1)}$	: matrice inverse généralisée de $\mathbf{A}$
$\mathbf{A}^+$	: matrice pseudo-inverse de Moore-Penrose de $\mathbf{A}$
$\det(\mathbf{A})$	: déterminant de la matrice $\mathbf{A}$
$\exp(\mathbf{A})$	: matrice exponentielle de $\mathbf{A}$
$\mathbf{0}_m, \mathbf{0}_{m \times n}$	: matrice nulle de dimensions $m \times m$, resp. de dimension $m \times n$
$\mathbf{I}_m, \mathbf{I}_{m \times n}$	: matrice identité de dimensions $n \times n$, resp. de dimension $m \times n$

Géométrie euclidienne et projective

N	: repère attaché au dernier corps d'un robot (effecteur)
S, S_i	: repère attaché à un capteur quelconque, resp. au i^{ime} capteur
W	: repère fixe attaché au monde
${}^S\mathbf{x}$	: vecteur à valeurs dans le repère S , définie par rapport au repère S
${}^N\mathbf{t}_S$	: vecteur de translation de S par rapport à N , de dimension 3
${}^N\mathbf{R}_S$	: matrice de rotation du repère S par rapport au repère N
${}^N\mathbf{T}_S$	: matrice de transformation homogène du repère S par rapport au repère N
${}^N\mathbf{T}_S$	: matrice de transformation spatiale d'un torseur cinématique défini dans S vers N

Robotique

n	: nombre de degrés de liberté du robot (i.e. nombre d'articulations)
$\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$	: positions, vitesses et accélérations articulaires, de dimension n
τ	: efforts (forces ou couples) articulaires
$\mathbf{v}_N = [v_N^T \omega_N^T]^T$	: torseur cinématique attaché au repère N
$\mathbf{J}$	: matrice cinématique du robot
$\mathbf{A}$	: matrice d'inertie du robot
$\mathbf{H}$	: vecteur regroupant les efforts de Coriolis, des forces centrifuges et l'action de gravité
$\mathbf{f}$	: torseur d'efforts

Commande référencée capteurs

m	: nombre de degrés de liberté de ou des tâches
$\mathbf{s}, \dot{\mathbf{s}}, \ddot{\mathbf{s}}$	: vecteur de caractéristiques capteurs, resp. sa variation et son accélération, de dimension m
$\mathbf{L}_s$	: matrice d'interaction associée à $\mathbf{s}$ ($\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v}_s$)
$\mathbb{J}$	: jacobienne de tâche associée à $\mathbf{s}$ ($\dot{\mathbf{s}} = \mathbb{J} \dot{\mathbf{q}}$)
$\mathbf{H}_s$	: hessien d'interaction associée à $\mathbf{s}$ ($\ddot{\mathbf{s}} = \mathbf{L}_s \dot{\mathbf{v}}_s + \mathbf{H}_s \mathbf{v}$)
$\mathbf{s}^d$	: référence pour le vecteur $\mathbf{s}$
$\mathbf{e}$	: erreur de la tâche ($\mathbf{e} = \mathbf{s} - \mathbf{s}^d$)

Remarque : une tâche est composée du tuple $(\mathbf{s}, \mathbf{s}^d, \mathbf{L}_s, {}^N\mathbf{T}_S)$.

Introduction

1.1 Opérateurs et industrie manufacturière

L'industrie définit un ensemble d'activités économiques tournées vers la production de biens. Une entreprise de taille industrielle produit des biens en grande quantité à partir de la transformation et la mise en œuvre de matières premières. La production en grande quantité entraîne le plus souvent un besoin important en terme d'équipements et de moyens. L'industrie manufacturière se focalise dans la fabrication d'objets ou d'ensembles plus complexes au sein de domaines d'application variés. Les domaines concernés sont nombreux : on pourra citer comme exemple l'industrie automobile, l'industrie aéronautique ou encore l'industrie navale pour ne citer que les domaines relatifs au moyen de transport.

La notion d'industrie est basée sur plusieurs concepts liés à la nature du travail et à son organisation. Parmi eux, la division du travail et l'utilisation de machines-outils spécialisées sont deux concepts qui ont joué des rôles importants dans la définition des industries telles que nous les connaissons aujourd'hui. L'établissement de la notion de la division du travail au XVIII^{ième} siècle [Adams, 1776], la révolution industrielle au XIX^{ième} siècle, puis la généralisation des lignes de montages pour des produits de consommation courante au début du XX^{ième} siècle (notamment pendant la période des *30 glorieuses*) ont posé les bases de notre tissu industriel et de nos moyens modernes de production .

Les produits manufacturés sont des produits issus des lignes de montage ou d'assemblage. On définit un produit manufacturé par l'association de plusieurs pièces fixées ensemble. Ces pièces peuvent être elles-mêmes des ensembles manufacturés provenant d'une autre ligne d'assemblage. De nombreuses opérations sont réalisées pour obtenir l'ensemble final. Le perçage et le ponçage par exemple sont des opérations qui sont effectuées lors de la préparation des pièces. La soudure ou le collage sont des moyens de fixations utilisés pour maintenir les pièces ensemble. Enfin, la peinture et le vernissage sont des opérations de finition. Ces opérations sont réalisées le plus souvent de manière manuelle et requiert le savoir-faire d'un opérateur.

La voiture est un exemple de produit manufacturé complexe issu de lignes d'assemblage. La voiture peut être vue comme une association d'un châssis, d'un moteur et de roues. Le moteur

à combustion interne est lui même issu d'un chaîne de montage qui lui est propre. Il en va de même pour le châssis (assemblage soudé de plusieurs pièces de métal formé) et la roue (assemblage d'une jante et d'un pneu), etc. Il est possible d'éclater l'assemblage de manière très fine en y listant toutes les opérations et tous les composants. Cet éclatement s'arrête généralement aux premières étapes de la transformation de la matière première. On peut aussi qualifier les bateaux d'ensembles manufacturés, ainsi que les avions de ligne.

L'action d'associer deux ensembles en utilisant un moyen de fixation définit une opération d'assemblage. Plus précisément, une opération d'assemblage est définie par une succession d'étapes élémentaires telles que la saisie de la pièce, son acheminement vers son emplacement final, son positionnement sur l'ensemble parent et sa fixation. A l'heure actuelle, ces étapes restent majoritairement réalisées par des opérateurs.

D'un point de vue humain, toutes ces étapes sont effectuées de manière quasi inconsciente. L'action de repérer un objet dans l'espace, de le prendre et de le poser dans un autre endroit sont des actions très naturelles qui sont apprises dès le tout premier âge. Cela justifie pleinement la présence des opérateurs pour ce genre de tâches. D'un point de vue travail, cela implique un certain nombre de conséquences à la fois sur les performances de la ligne de production dans sa globalité et sur la manière de travailler des opérateurs.

Depuis le début du XX^{ième} siècle, les conditions de travail pour les opérateurs ont évoluées notamment par l'avènement d'outillage de plus en plus spécialisé (cf. figure 1.1b). Le but étant toujours d'améliorer les performances des lignes de production et plus généralement la productivité de l'entreprise. Même si l'outillage spécialisé a apporté un gain significatif dans les conditions de travail, il a aussi spécialisé la place et les mouvements des opérateurs en les rendant répétitifs. En conséquence, l'opérateur a perdu sa polyvalence et de nouveaux problèmes de santé tels que les troubles musculo-squelettiques, sont apparus.

En l'état actuel des choses, il est possible d'établir le constat que l'amélioration des conditions de travail des opérateurs et l'augmentation continue des performances des unités de production sont deux points antagonistes. La volonté de produire plus, fait que les industries atteignent progressivement une limite qu'elles ne pourront pas dépasser sans remettre en question les conditions et la qualité de travail des opérateurs. Fort de ce constat, certaines entreprises sont à la recherche d'une nouvelle manière de travailler afin de satisfaire leurs objectifs tant au niveau humain qu'économique.

En rapport aux techniques de production et aux industries manufacturières, la Robotique est une discipline relativement jeune. Elle apporte de nouvelles méthodes de production et propose de nouveaux modèles. La coopération opérateurs-robot (aussi nommée la *cobotique*) est un moyen avancé pour moderniser les lignes de montages et d'apporter une réponse aux besoins actuels de l'industrie manufacturière.

Au delà du contexte général du travail et des différents acteurs, le contexte de ce projet de thèse est fortement lié à la conjoncture nationale française. Le projet prend place au sein d'une nouvelle institution créée dans le but de soutenir la recherche et l'innovation au sein des industries françaises. L'Institut de Recherche et de Technologique Jules Verne (IRT Jules Verne) est une fondation de recherche multi-thématiques interdisciplinaire affichant la volonté de mutualiser les compétences des laboratoires de recherche universitaire et celles des industriels. Son but est de favoriser la collaboration de l'ensemble de ses acteurs à travers des projets collaboratifs.

Le projet *Assistant for industrial structure and systems and manufacturing optimization va-*



FIGURE 1.1 – (a) Ligne d’assemblage à l’arsenal de Woolwich de Londres (1917), D. Holt. CC BY-SA 2.0. (b) Ligne d’assemblage automobile, Ford Motor Co (2008). CC BY 2.0 via Wikimedia Commons.

lue (ASIMOV), dans lequel s’inscrit cette thèse, a pour objectif de réduire les coûts et les temps de cycle pour la production d’un avion. Les efforts se focalisent principalement sur les opérations d’assemblages et d’installations des systèmes sur la ligne d’assemblage finale (FAL ou *final assembly line*), où le souhait est d’automatiser certaines tâches à faible valeur ajoutée.

Le but du projet est de concevoir un robot manipulateur mobile coopératif évoluant dans un milieu industriel où a lieu l’assemblage des éléments structurels et l’installation des systèmes de l’ensemble pointes avant des avions Airbus, ainsi que leurs tronçons centraux. Les missions de ce robot sont de naviguer de manière autonome dans un environnement non contrôlé et/ou peu ou partiellement modélisé, pour rejoindre le lieu d’intervention et de réaliser sa tâche de manipulation en collaboration avec un opérateur.

Les principaux résultats attendus dans ce projet peuvent être résumés à travers deux axes majeurs de développement.

Le premier est de développer une plateforme mobile omnidirectionnelle capable d’évoluer en intérieur dans un environnement encombré et de porter un bras manipulateur. Cet axe comprend aussi le développement d’algorithmes avancés de navigation permettant à la plateforme mobile de se mouvoir de manière sûre, à l’intérieur d’une structure aéronautique en cours d’assemblage avec des opérateurs à proximité.

Le deuxième est le développement d’algorithmes pour la manipulation d’objets et de pièces manufacturées utilisés dans les assemblages aéronautiques. Il comporte plusieurs sous-aspects tels que la détection des pièces, le calcul de saisie stable pour un préhenseur donné (pince parallèle et/ou main robotique) et le contrôle du bras manipulateur en ayant un objectif de positionnement de la pièce au contact de la structure.

La dernière étape est l’intégration de l’ensemble des briques technologiques développées dans les axes sus-cités, le but étant de créer un robot collaboratif industriel robuste capable d’effectuer en toute sécurité les tâches d’assemblage en présence d’opérateurs à proximité.

1.2 Détail d'une opération d'assemblage

1.2.1 Point de vue opérateur

Parmi les opérations manuelles à effectuer pour réaliser un produit manufacturé, l'assemblage est l'opération la plus basique. De fait, c'est l'opération la plus difficile à définir. Il n'y a pas de savoir-faire, ni d'outils, à proprement parler. De plus, les actions d'une opération d'assemblage se retrouvent dans d'autres opérations telles que le ponçage, le perçage, le soudage, le collage, la peinture, le vernissage, etc.

On se focalisera sur un cas d'assemblage simple où une seule des deux parties structurelles sera mobile. La majeure partie des cas d'assemblage peut être définie par "ajouter une pièce à un ensemble rigide non mobile", sans tenir compte du procédé de fixation. On appelle structure l'ensemble rigide non mobile qui reçoit la pièce à assembler. Le poste de travail de l'opérateur se compose d'un magasin de pièces, d'outils et de la documentation nécessaire à l'opération. Il se trouve à proximité de la structure. On désigne par la zone d'emport la surface où la pièce est disposée initialement, sans attention particulière sur la manière dont celle-ci est positionnée. Dans une situation réelle, la pièce peut ne pas être directement accessible, rangée dans un tiroir dans le magasin ou mise dans un bac de convoyage par exemple. Ce dernier cas de figure est exclu du cadre de l'étude.

La position finale dénomme la position de la pièce dans la structure conformément aux instructions données sur la documentation une fois l'opération d'assemblage terminée. Elle est décrite en ayant recours à des points de références géométriques ou structurelles (plans, arêtes, coins, etc.) à la fois sur la structure et sur la pièce à assembler. Le lien entre une référence de la structure et une référence de la pièce est déterminé par une relation géométrique (plans coïncident par exemple) et/ou une cotation métrique.

L'opérateur a la capacité d'apprécier son environnement. Il est possible de dire qu'il établit une cartographie en trois dimensions de son environnement proche en s'appuyant sur ses yeux, son oreille interne et tous les autres organes impliqués dans son système de perception spatial. Il peut ainsi estimer l'attitude générale de la pièce sur la zone d'emport et celle de la structure. Avec la documentation, il repère aussi l'endroit potentiel où se trouve la position finale de la pièce tout en jugeant l'encombrement qu'il y a autour. Cependant, l'opérateur ne peut qu'estimer une position approximative : il n'est pas en capacité d'apprécier une position métrique exacte d'un simple coup d'œil.

Il est aussi capable de saisir, de manipuler la pièce et d'aller positionner celle-ci à l'endroit désiré, à l'aide de ses bras et de ses mains. Ces derniers sont des outils pour la manipulation d'objets qui n'ont pas d'équivalent technologique à ce jour. Comme énoncé précédemment, la saisie et la dépose d'objet sont des actions qui ont été apprises dès le tout premier âge, et perfectionnées tout au long de sa vie.

Pour permettre la comparaison entre une opération d'assemblage réalisée par un opérateur et par un système robotique, on dresse dans un premier temps une liste des tâches que l'opérateur effectue pour cette opération. Cette liste d'étapes peut paraître absurde car nous (*i.e.* êtres humains) ne les faisons plus consciemment.

La première de ces étapes est d'apprécier grossièrement l'environnement et la situation de tous les composants. De manière très automatique, sera jugée la capacité d'atteindre tel ou tel élément (pièce à poser ou position finale sur la structure) pour déterminer si l'ensemble de son corps doit

être déplacé pour réaliser l'opération.

Si la pièce à prendre est à portée de saisie, alors en fonction de l'habitude et de la répétabilité du geste, l'opérateur peut diriger son bras dans la direction de la pièce pour aller la saisir. Ici deux cas de figure sont possibles : soit l'opérateur décide de focaliser son attention sur sa main et la pièce pour permettre une saisie correcte du premier coup ; soit il décide de conserver son attention sur sa tâche précédente, laisse sa main entrer en collision avec la pièce, tâtonne pour connaître la situation de la pièce et décide d'une saisie.

Une fois la pièce saisie, l'opérateur l'achemine vers la position désirée sur la structure. Il évalue la trajectoire de sa main qui porte la pièce instantanément, que l'environnement soit complètement libre ou très encombré. Le chemin trouvé peut ne pas être faisable, en particulier si la position finale est difficilement accessible à cause d'un environnement encombré et/ou de la complexité de la structure. Dans ce cas, une collision est possible mais celle-ci reste sans gravité grâce aux efforts ressentis à travers sa main et son bras. En cas de collisions, il évalue directement un nouveau chemin à prendre avec une nouvelle manière de positionner son bras.

L'avant dernière étape est l'approche de la pièce vers sa position finale. Celle-ci peut être précédée d'une phase de préparation pour marquer l'empreinte de la pièce sur la structure en s'aidant de la documentation et d'un système de mesure. L'opérateur trace les références géométriques de la pièce par rapport aux références de la structure autour de l'emplacement final (références locales), conformément aux indications disponibles sur le plan de l'opération. Cette fois-ci, l'attention de l'opérateur est complètement focalisée sur cette approche et fait coïncider les marques avec les références de la pièce. C'est une phase de précision où l'opérateur peut mettre en contact la pièce avec la structure et faire glisser celle-ci jusqu'à son emplacement.

Une fois la position atteinte, l'opérateur applique alors un effort important pour éviter que la pièce ne perde sa position précise avant l'intervention du procédé de fixation (vissage par exemple). Le procédé de fixation est mis œuvre soit par sa deuxième main – cas qui nécessite de dévier momentanément son attention de la pièce en contact – ou par une deuxième personne – cas qui nécessite de libérer l'espace de travail au maximum tout en maintenant la pièce au contact.

L'action de fixation marque la fin de l'opération d'assemblage. Toutes ces opérations peuvent être effectuées en se basant sur la capacité cognitive spatiale que nous possédons tous. La qualité et la réussite de l'opération sont dépendantes de nombreux facteurs qui, pour certains, peuvent être difficilement contrôlables. Par exemple, la précision de l'assemblage dépend entre autres de la qualité de réalisation de la phase de marquage. En effet, celle-ci ne peut pas être correctement effectuée : mauvaise mise en œuvre du système de mesure, tracé du crayon de marquage trop épais, défaut d'attention de l'opérateur, mauvaise lecture du plan et de la documentation, etc. La qualité finale de l'opération dépend en effet fortement du facteur humain.

Toutes ces étapes, aussi basiques et naturelles soient-elles, sont classées sous une étiquette que l'on peut appeler macro-manipulation. En effet par abus de langage, la manipulation est habituellement associée aux différentes techniques de saisie et de manipulation dextre. La manipulation dextre peut être définie par l'action de mouvoir la pièce dans sa main après que celle-ci ait été saisie et soulevée. En Robotique, la manipulation dextre et la recherche de saisie stable sont des domaines de recherche à part entière.

Cependant, une opération d'assemblage ne peut pas être réalisée en ne regardant que la main et sa capacité à saisir les objets. Nous avons mentionné que le succès dépendait à la fois fortement de la perception spatiale de l'opérateur de son environnement, sa capacité à acheminer l'objet vers sa position désirée et à le positionner. Le terme macro-manipulation intervient dans une volonté d'élargir le cadre autour de la simple saisie – simplicité toute relative – en y intégrant le contexte

environnemental proche ainsi que les outils pour permettre le succès total de l'opération.

Ici, le facteur humain prend une place prépondérante dans le succès d'un assemblage. L'ensemble des opérations sont réalisées du fait de la capacité de l'opérateur à apprécier son environnement. Or, notre cerveau ne raisonne pas en mètres, ni en degrés. Le recours à des systèmes tiers (outils de mesures par exemple) pour garantir la précision est nécessaire, voire indispensable. Cela entraîne plusieurs conséquences : d'une part l'intervention d'un système tiers nécessite de nouvelles manipulations qui peuvent à nouveau introduire des biais pendant la réalisation de l'opération, d'autre part le système tiers peut donner une mesure erronée qui peut entraîner de nouveau l'injonction de biais dans l'opération. En d'autres termes, même si l'opérateur est «outil technologique» actuel le plus naturel pour réaliser une opération d'assemblage de par ses capacités naturelles, la garantie qu'il y arrive avec une performance acceptable est toute relative.

Par ailleurs, garantir que l'assemblage est correctement effectué reste un exercice tout aussi difficile. Par exemple, il sera toujours possible de contrôler la précision d'un ensemble à chaque ajout d'une nouvelle pièce, en prenant le risque de multiplier des opérations de contrôle de plus en plus lourdes. Les entreprises manufacturières ont apporté des solutions pour limiter l'adjonction de biais dans ces opérations. Certaines de ces solutions sont en lien étroit avec le développement de machine-outils de plus en plus spécialisées et perfectionnées. Très simplement, l'utilisation d'un gabarit lors d'une opération de perçage évite à l'opérateur l'étape de prises de mesure et de marquage et supprime les biais qui vont avec. D'autres solutions intègrent directement dès la conception de la pièce et de la structure, si les outils le permettent, des facilités pour limiter le temps passé à positionner précisément certaines pièces (ergots et trous par exemple). D'autres solutions encore, impriment l'empreinte de la pièce sur la structure au moyen d'une machine spécialisée, si l'état de surface le permet. Ainsi la garantie de précision du marquage dépend de la capacité de la machine. La méthode n'empêche pas néanmoins que l'opérateur positionne l'objet de travers.

Ces solutions ont leurs avantages et leurs inconvénients. Certaines ne peuvent tout simplement pas être mise en œuvre dans tous les cas de figure. D'autres apportent parfois plus d'inconvénients que d'avantages mais permettent d'atteindre le niveau de garantie souhaitée. *In fine*, l'opérateur avec sa polyvalence et sa flexibilité dans l'environnement de travail reste le meilleur compromis pour la réalisation d'opérations d'assemblage.

Après avoir identifié quelle était la place d'une opération d'assemblage dans le milieu industriel et dans les entreprises manufacturières, à partir des premiers concepts issus de la révolution industrielle jusqu'à l'apparition d'outillages mécaniques de plus en plus spécialisés, nous nous posons la question de l'utilisation d'un robot pour réaliser une tâche d'assemblage, d'identifier les points bloquants et les efforts déjà entrepris pour y parvenir. L'objectif de ce chapitre, après avoir esquissé la problématique d'un assemblage du point de vue d'un opérateur, est d'aborder en détail notre approche en posant les éléments théoriques et les éléments d'architecture logicielle nécessaires.

1.2.2 Point de vue robotique

Dans cette thèse, on s'intéresse à la classe des robots manipulateurs capable de mouvoir l'effecteur dans l'espace. Ces robots sériels se composent d'un bras mécanique poly-articulé portant un système de préhension et plusieurs capteurs. Ces robots sont largement répandus dans le milieu industriel pour des tâches hautement répétitives à faible valeur ajoutée. On y distingue les robots de peintures, de soudage, d'inspection, de transfert de marchandises entre lignes d'assemblage ou

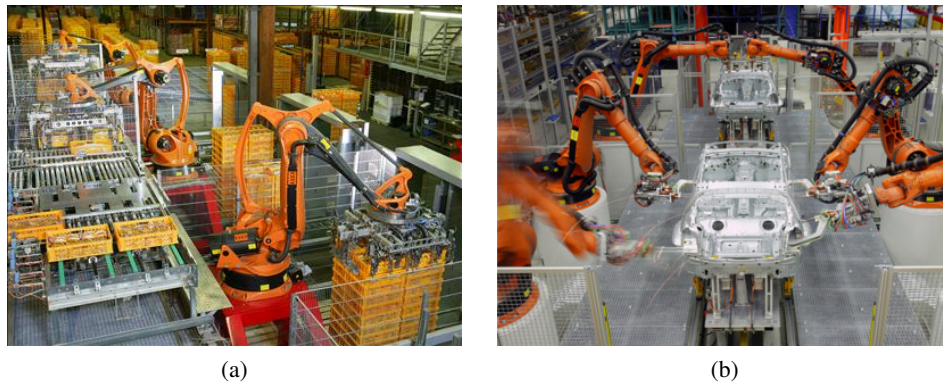


FIGURE 1.2 – (a) Unité de palettisation robotisée, KUKA Roboter GmbH, 2005. Domaine public. (b) Ligne de soudage robotisée, KUKA Systems GmbH, 2013. CC BY-SA 3.0.

de montage (palettisation, cf. figure 1.2a), etc. Ils sont de tailles différentes en fonction des tâches qui leurs sont assignées.

Ils ont tous en commun une caractéristique commune quelle que soit leur taille : la répétabilité. La répétabilité est une notion différente de la précision absolue d'un robot. La répétabilité se définit par l'erreur maximale de positionnement répété de l'outil en tout point de son espace de travail. La répétabilité moyenne d'un robot manipulateur est de l'ordre d'un dixième de millimètre. Plusieurs paramètres structurels peuvent influencer celle-ci : la rigidité des corps mécanique par exemple. C'est en partie la raison pour laquelle plus les robots sont grands, plus ils sont lourds et imposants. Ainsi, les concepteurs garantissent sa répétabilité en rendant les corps du robot plus solides et donc moins flexibles.

A contrario, la précision est mesurée par l'erreur de positionnement de l'organe terminal par rapport à un point fixe de son environnement. Cette mesure est impactée par de nombreux facteurs internes au robot. Parmi ceux-ci, on cite les erreurs de modèles géométriques, la précision de la mesure de l'environnement, la méthode de représentation des modèles virtuels (tessellation, représentations géométriques approximatives), etc. La précision peut être améliorée en aidant le robot à percevoir son environnement par l'intermédiaire de capteurs extéroceptifs tels que des caméras visuelles, des caméras à profondeur de champ ou tout autre capteur de mesure spatiale. La précision absolue est un paramètre que nous ne pouvons négliger dans une opération d'assemblage. Mais ce n'est pas le seul facteur à prendre en compte.

En partant d'un cas simple de manipulation ayant pour but l'assemblage d'une pièce manufacturée sur un ensemble plus grand, il est possible de faire une liste non exhaustive des compétences nécessaires à la réalisation de la tâche. Les étapes d'une manipulation ont été identifiées dans la section précédente d'un point de vue opérateur. L'exercice est identique mais cette fois-ci en plaçant le robot à la place de l'opérateur. Le cas se veut le plus générique, sans la moindre difficulté : la pièce à assembler est posée de manière à ce que le robot puisse la saisir, de même que l'emplacement final de la pièce est considéré comme accessible. Le robot manipulateur est équipé d'un système de préhension (pince parallèle ou main robotique bio-inspirée) capable d'attraper la pièce, ainsi que d'un ensemble de capteurs lui donnant la capacité à reconnaître et évaluer son environnement.

La première étape est d'avoir connaissance de l'opération à effectuer. La méthode peut varier en fonction des caractéristiques des entrées du système, ainsi que d'une stratégie haut niveau qui

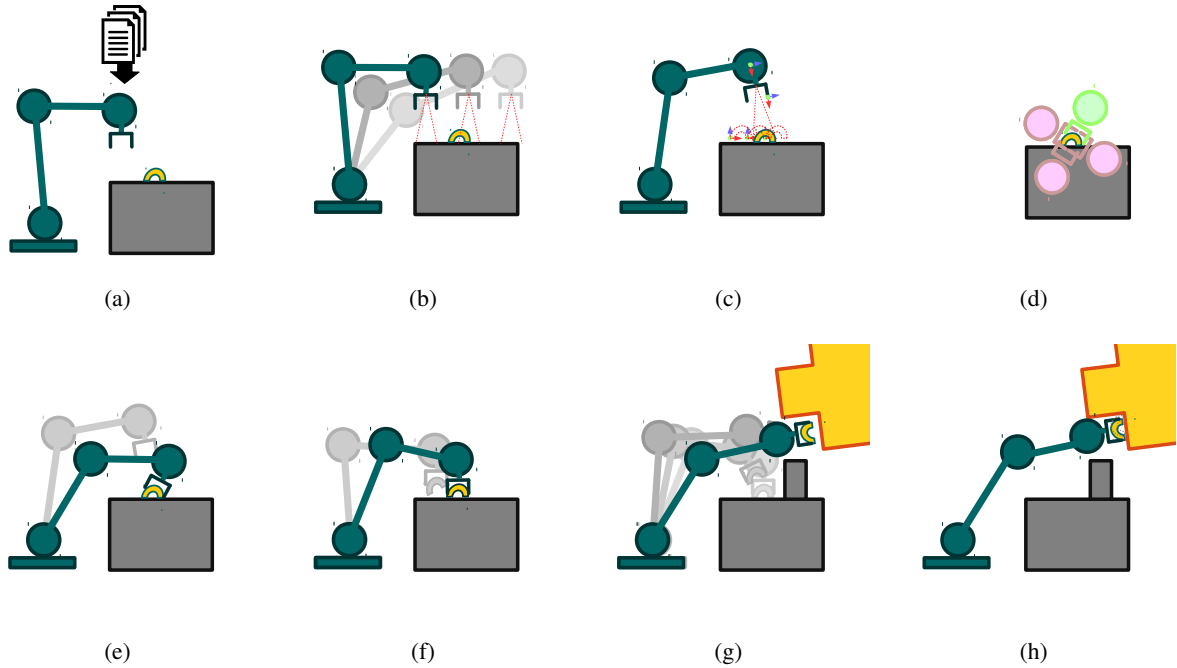


FIGURE 1.3 – Étapes d’une pour une manipulation robotique classique en milieu industriel : (a) réception des entrées et données pour la réalisation de l’opération ; (b) recherche de la pièce à saisir (balayage) ; (c) estimation de la pose de la pièce ; (d) génération et sélection d’une configuration de saisie valide ; (e) mouvement de saisie ; (f) mouvement de recalage ; (g) acheminement vers destination finale ; (h) positionnement fin sur destination finale.

permet à notre système de s’intégrer dans un plan de plus grande envergure. Les entrées nécessaires à notre système sont l’identification de la pièce à saisir et où celle-ci se place sur la structure. La recherche documentaire peut être identifiée comme une étape à part entière. En effet, l’opérateur doit être en capacité d’identifier l’emplacement de la pièce à poser sur la structure en s’appuyant sur des plans ou des maquettes numériques. La recherche documentaire pour un robot peut être automatisée par l’accès du système informatisé à une base de données qui recense l’ensemble des informations nécessaires à l’opération (une maquette numérique par exemple). Là aussi, l’accès aux informations via la base de données nécessite un système d’information robuste dont l’architecture est en lien fort avec la définition de l’opération et la stratégie d’intégration haut-niveau du système.

Ayant obtenu l’ensemble des informations nécessaires à l’opération d’assemblage, il est nécessaire de savoir où se trouve la pièce pour la saisir. En comparaison avec un opérateur humain, un robot possède des capacités de perception et de préhension restreintes (*i.e.* il n’a ni œil, ni main, ni de système de cognitif dédié à la perception spatiale sur-entraînée). Il n’est qu’en possession du modèle virtuel de la pièce. Il faut donc une solution ad-hoc pour rechercher la pièce et déterminer sa pose dans l’espace. La recherche de la pièce peut être facilitée par la mise en place d’une zone ou d’un volume à l’intérieur de l’espace de travail qui sera balayé par le robot. Le robot effectue une passe sur l’ensemble de cette zone pour permettre aux capteurs extéroceptifs (caméras visuelles, caméras à profondeur de champ, télémètres, etc.) de détecter l’objet recherché. Une fois la position détectée grossièrement, un focus peut être effectué pour procéder à une estimation de pose plus fine (par exemple, par recalage de modèles 3D mesuré réel/connu virtuel).

Ensuite, l'étape suivante est de déterminer quelle serait la meilleure façon de saisir l'objet. On définit une saisie réussie par l'action de positionner le préhenseur dans une configuration qui maintiennent l'objet saisi de manière ferme et sans glissement. On parle alors d'une saisie stable. N'ayant la capacité de manipulation dextre (*i.e.* modifier la configuration de la saisie tout en garantissant une saisie stable alors que l'objet est tenu), on suppose donc que le robot ne peut pas mouvoir la pièce dans le préhenseur une fois saisie. Il est donc important de déterminer la manière dont le robot saisit la pièce compte tenu de toutes les informations qu'il possède *a priori*. De nombreuses contraintes entrent en jeu pour déterminer la meilleure saisie. Il y a les contraintes dures : la pose de la pièce avant saisie, la pose de la pièce dans son emplacement final sur la structure, l'encombrement des corps du robot, l'encombrement de la structure, la capacité de préhension du robot, la qualité de la saisie (mesure qui garantit que la pièce ne glissera pas du préhenseur après avoir été saisie), la capacité du robot à mouvoir son organe terminal, etc. Il y a aussi des contraintes plus souples : l'accessibilité de l'opérateur aux zones de fixation de la pièce avec un outil (défini par un volume virtuel que le robot ne doit pas occuper une fois avoir positionné la pièce) ou encore l'occupation de la zone de travail par les corps du robot.

La recherche d'une saisie stable compte tenu des caractéristiques d'un préhenseur est un domaine de recherche en robotique encore très actif. Une solution possible est une recherche de type force-brute qui calcule l'ensemble des hypothèses possibles pour ensuite sélectionner celle qui respecte toutes les contraintes. La complexité de recherche de ces algorithmes augmente avec le nombre de degrés de liberté du robot ainsi que ceux du préhenseur, le problème devenant très rapidement insolvable car trop consommateur de ressources (ressources calculatoires et/ou temps de calcul).

La combinaison des mouvements de saisie peut avoir une incidence négative sur la qualité de la saisie. Du fait des écarts des modèles et/ou des interactions de contact capricieuses (surfaces glissantes, dilatées, etc.), la pose obtenue ne correspond pas à celle désirée, sélectionnée auparavant. En effet, nous avons mené une étude qui montre que la pose obtenue est toujours différente de celle désirée : la pièce glisse très légèrement avant d'être immobilisée par les doigts du préhenseur (voir l'annexe B). Pour régler ce problème, on procède alors à un recalage le plus souvent physique qui consiste à repositionner la pièce dans le préhenseur en s'aidant de l'environnement proche. Le robot, tout en ayant saisi la pièce, vient pousser celle-ci sur plusieurs surfaces connues, garantissant ainsi sa pose dans le référentiel du robot. Dans d'autres cas, le recalage est effectué par des systèmes annexes qui placent les objets à saisir dans une position connue et ainsi évite des mouvements supplémentaires au bras manipulateur.

Une fois l'objet saisi et sa pose connue, il est nécessaire de repérer l'emplacement final où l'objet doit être positionné. Encore une fois, la localisation de la structure n'est pas aussi triviale pour un robot que pour un opérateur. On appelle localisation de la structure le fait d'estimer la pose de la structure compte tenu de son modèle virtuel, de sa situation réelle ainsi que des capteurs à sa disposition pour estimer cette pose.

Là aussi, la perception d'un robot à travers des mesures extéroceptives est un domaine de recherche actif en robotique. La perception en règle générale, fait intervenir plusieurs problématiques tels que le champ de vision restreint des capteurs. De même, le recalage de modèle virtuel sur un objet réel dépend de nombreux facteurs qui peuvent entraîner une mauvaise estimation de la pose de ce dernier. Plusieurs cas de figure sont possibles. Si le modèle n'est pas assez fin ou le processus de modélisation entraîne trop d'erreurs dues aux approximations (parfois nécessaires), la pose obtenue à l'issue de ce processus d'estimation intégrera ces erreurs et ces approximations. A l'inverse, si la structure comporte des flexibilités et que celle-ci a changé de forme après sa réalisation, il y a une non-correspondance entre la structure à l'instant t et son modèle virtuel à sa conception. De même, cette non-correspondance peut entraîner des erreurs sur le processus d'estimation. Enfin,

la nature des données de sortie des capteurs peut aussi avoir une incidence sur ce processus. Par exemple, une séquence d'images prises par une caméra qui observe la scène contient relativement peu d'informations qualitatives pour estimer la situation de celle-ci par rapport à son environnement. Il est nécessaire de post-traiter ces informations brutes par une série d'autres processus pour faire ressortir les informations géométriques qui nous intéressent. Tous ces processus comportent eux aussi des erreurs et des approximations qui apportent une incertitude sur la qualité de l'opération de positionnement.

Au final, la localisation et plus généralement la perception d'un robot, est dépendante de nombreuses approximations et autres erreurs qui apparaissent à différentes étapes du processus. Le but est donc de garantir que la pose ne comporte pas trop d'erreurs et de limiter celles-ci tout au long de l'étape d'estimation. Le plus souvent, la précision atteinte n'est pas suffisante pour satisfaire la qualité d'un positionnement de la pièce pour l'assemblage. D'autant plus qu'il faut mettre en relation cette notion de précision avec celle de la précision absolue du robot.

La figure 1.3 résume l'ensemble de ces étapes de manière schématique.

Toutes ou parties de ces étapes peuvent être solutionnées par des moyens sur mesure (par exemple, les systèmes parallèles pour l'alignement de pièces). Néanmoins, ces solutions sont spécifiques et ne s'appliquent pas à toutes les situations. Toutes ces étapes n'ont pas encore trouvé de solutions génériques, robustes et efficaces. En allant plus loin, il est possible de dire que les robots manipulateurs ne sont pas encore prêts pour des tâches d'assemblage dans un environnement complexe. La section suivante fait l'inventaire des travaux de recherche concernés par cette problématique d'assemblage.

1.3 Assemblage et robots

Cette section fournit un bref aperçu des travaux de la communauté scientifique concernant les activités d'un robot autour d'une problématique d'assemblage. Il n'y a pas de champ de recherche spécifique sur la question d'assemblage : certaines équipes redéfinissent le cadre de spécification d'une tâche d'assemblage par un robot tandis que d'autres ont une approche beaucoup plus classique en revisitant les outils connus de la communauté.

En premier lieu, de nombreux travaux se concentrent sur la collaboration d'une équipe de robots assembleurs pour acheminer et positionner les pièces pour construire une structure désirée.

[Werfel and Nagpal \[2008\]](#) se positionnent sur le segment d'assemblage d'une structure à partir de pièces modulaires. L'effort est principalement concentré sur la séquence d'assemblage des différentes pièces pour atteindre la forme finale de la structure désirée. Une fois la séquence déterminée, les robots sont mis en action pour transporter les pièces de leur emplacement courant jusqu'à leur prochaine destination en se basant sur une planification de trajectoire sans collision avec les autres robots. Ces travaux ont été validés par simulation.

[Schoen and Rus \[2013\]](#) proposent un algorithme dont le but est de coordonner les différents éléments (pièces) et acteurs (robots) nécessaires à la construction d'une structure. L'algorithme détermine la séquence d'assemblage nécessaire pour cette construction, en tenant compte des contraintes temporelles d'acheminement des pièces [[Schoen and Rus, 2013](#)] et des contraintes physiques de chacune des pièces (masse, forme, équilibre) [[Stein et al., 2011](#)] pour réaliser celles-ci. L'équipe de robots est gérée de manière à optimiser le temps de construction en parallélisant l'acheminement et la dépose des pièces dans le bon ordre. Plus particulièrement dans [[Schoen and Rus, 2013](#)], l'équipe de robots est séparée en deux : une partie est responsable de l'acheminement

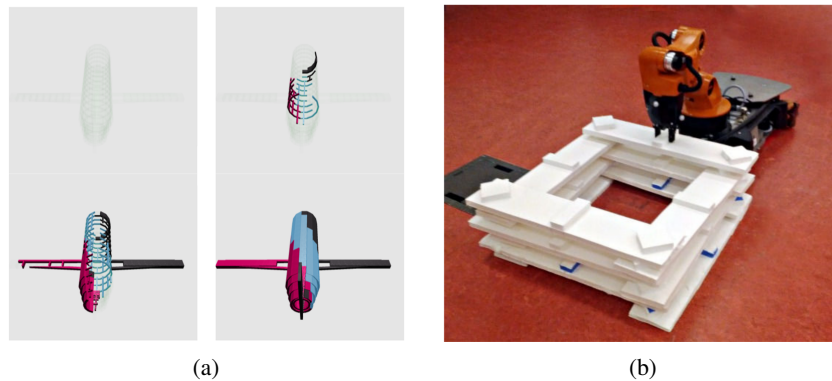


FIGURE 1.4 – (a) Séquence d’assemblage d’un modèle d’avion [Stein et al., 2011]. Les quatre vignettes représentent l’avancement de l’assemblage à quatre instants différents, les couleurs correspondent aux différents robots impliqués dans l’opération. (b) Expérimentation réelle d’un assemblage d’une structure à l’aide d’un manipulateur mobile [Schoen and Rus, 2013].

de la pièce sur le site de construction, tandis que l’autre est responsable de l’assemblage lui-même. Cependant, les auteurs indiquent que l’expérimentation réelle (cf. figure 1.4b) n’est pas répétable et que celle-ci peut échouer même dans les conditions les plus faciles. Une des causes identifiées est que toutes les manipulations sont effectuées de manière aveugle sans recours à des capteurs extéroceptifs. Par conséquent, les robots ne sont pas capables de réagir à des écarts entre le modèle de l’environnement et le monde réel.

Dans un autre registre, la communauté s’intéresse à la mise en place de cadre de spécifications pour simplifier la définition d’une tâche d’assemblage et de son processus.

Parmi eux, Horváth et al. [2002] partent du constat initial que le processus d’assemblage robotique ne tient compte que du modèle géométrique de la pièce à assembler. Par conséquent, les processus d’assemblage sont très rigides et la cellule robotique ne peut en accepter de nouveaux sans avoir à repenser tout son contexte. Cela s’explique par le fait que les trajectoires sont générées hors ligne, en amont du processus, par la spécification d’une pose initiale pour la préhension de la pièce, d’une pose finale à atteindre et d’un chemin possible entre les deux. Ces poses ont été identifiées manuellement à l’aide d’un logiciel CAO. La contribution de Horváth et al. [2002, 2003] a été de proposer un générateur de processus d’assemblage qui génère les poses initiales et finales ainsi que les trajectoires en se basant sur les caractéristiques de forme géométrique commune aux pièces à assembler. Ainsi, le processus peut être alimenté avec des pièces de formes générales différentes dont certaines caractéristiques possèdent des formes géométriques similaires. Leur approche apporte une flexibilité non négligeable par rapport à une approche classique.

Wang et al. [2008, 2012] proposent un cadre de spécification fonctionnelle d’un processus industriel pour simplifier la planification des tâches en amont du processus. La différence par rapport à Horváth et al. [2002] est que leur spécification fonctionnelle s’applique aussi pendant l’exécution du processus. Cela leur permet d’ajouter de nouvelles contraintes extérieures à la simple action de manipulation et d’appliquer des décisions qui en découlent. Le cadre peut prendre en compte plusieurs unités d’assemblage à la fois en y intégrant des contraintes telles que des outils cassés, l’engorgement de la cellule ou encore des retards dans l’approvisionnement. Ils ont appliqué ce cadre de spécification fonctionnelle pour un assemblage de pièces manufacturées effectué par un robot manipulateur. Ils y décrivent un algorithme de gestion automatisé d’assemblage haut niveau.

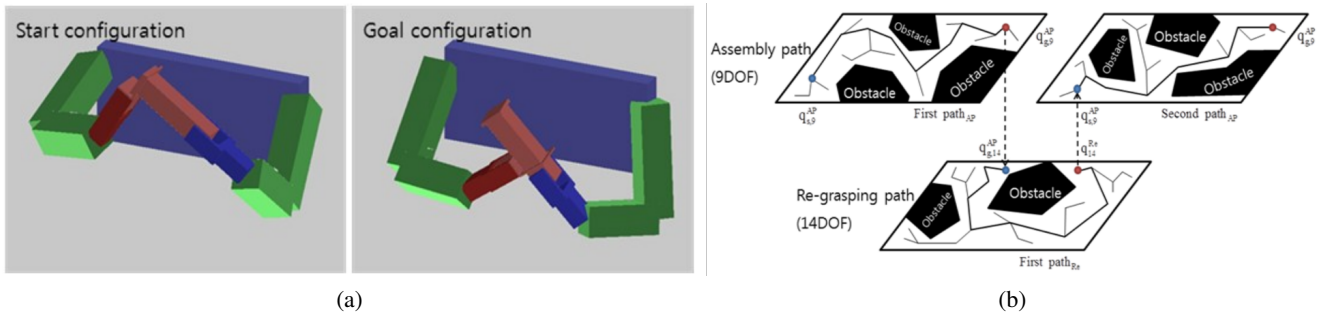


FIGURE 1.5 – Illustration d’assemblage par un système double bras manipulateur [You et al., 2012] (a) Configuration initiale et finale de l’ensemble du système bras et pièces à assembler (b) Graphe représentant les multiples espaces de recherche pour l’obtention de configurations articulaires à l’aide de la méthode *RRT*.

Les opérations de manipulation sont classifiées en six groupes de blocs fonctionnels qui représentent des manipulations basiques (insertion, alignement de plan, etc.). Ces blocs fonctionnels sont responsables des fonctions de planification de trajectoire et de contrôle du robot.

Ces cadres de spécification du processus d’assemblage effectué par un robot apporte de la flexibilité dans la définition des tâches. Néanmoins, toutes les manipulations sont encore une fois effectuées sans tenir compte des écarts entre le monde réel et sa modélisation. La flexibilité se trouve donc limitée à la définition de la tâche mais les contributions ne garantissent pas le succès des assemblages.

D’autres approches se focalisent sur des outils qui peuvent être appliqués pour solutionner une tâche d’assemblage.

You et al. [2012] proposent un algorithme pour la planification de tâches d’assemblage effectuées à l’aide d’un système manipulateur à deux bras (cf. figure 1.5). L’effort se concentre sur l’apport de solutions de configurations articulaires valides qui respectent les contraintes de manipulabilité des bras manipulateurs et de l’assemblage à effectuer, des contraintes de collision et des configurations de saisies des pièces. Les solutions sont trouvées grâce à une méthode de recherche aléatoire probabiliste *RRT* (*Rapid exploring Random Tree*). Plus spécifiquement, l’outil est utilisé sur deux types de mouvement élémentaire : le mouvement des pièces pour réaliser l’assemblage et le changement de configuration de saisie si le premier n’a plus de solution. La planification de trajectoire en milieu contraint est un aspect très largement traité dans le domaine robotique : Siciliano et al. [2010, chapitre 4] revisitent les méthodes d’interpolation dans l’espace articulaire ou cartésien ; Kavraki and LaValle [2008] posent plusieurs méthodes de modélisation et de recherche pour la planification de trajectoire avec ou sans contraintes ; Diankov [2010] propose un cadre logiciel *OpenRave* pour la simulation, la résolution de redondance et la planification de trajectoire de robots manipulateurs et/ou mobiles ; enfin, OMPL [Şucan et al., 2012] (en anglais, *Open Motion Planning Library*) offre une suite logicielle robuste pour l’ensemble des problèmes de planification de trajectoire de corps quelconque dans l’espace.

Chen and Liu [2012] proposent une loi de commande compliant robuste sur la base des lois de commande en impédance largement discutées dans le domaine ([Khalil and Dombre, 2002, chapitre 15], [Chung et al., 2008] ou [Siciliano et al., 2010, chapitre 8]). L’application visée est l’insertion d’un connecteur électronique dans une embase fixée sur une carte électronique à l’aide d’un robot type SCARA. L’accent est mis sur la robustesse face aux erreurs d’identification du

modèle dynamique du robot. Ott et al. [2015b] proposent un schéma de commande hybride qui mixe une commande en impédance et une commande en admittance pour permettre un contrôle du robot avec des interactions désirées ou non. Le lecteur pourra trouver dans Villani and De Schutter [2008] une présentation exhaustive des schémas de commande en effort passive ou active pour la gestion des interactions du robot avec son environnement.

La saisie constitue aussi un aspect important dans des tâches d'assemblage effectuées par un robot. Prattichizzo and Trinkle [2008] y résument les principaux enjeux de ce domaine spécifique tels que la détermination des configurations de saisie sur deux systèmes de préhension comportant beaucoup de degrés de liberté, ainsi que l'évaluation de la qualité de la saisie. Borst et al. [1999, 2003, 2005] proposent une méthode de recherche déterministe qui consiste à proposer plusieurs centaines de configurations de saisie pour un objet et un préhenseur. Ces hypothèses sont alors ordonnées en se basant sur le test de fermeture des forces [Ferrari and Canny, 1992] pour sélectionner l'hypothèse de configuration la meilleure. De même en parallèle, Miller and Allen [2004] ont proposé un logiciel de synthèse pour la génération de saisie en trois dimensions. Plus récemment, le projet HANDLE [Perdereau, 2013] s'est concentré sur la compréhension de la saisie et des mouvements humains afin de pouvoir les répliquer sur une cellule robotique artificielle anthropomorphique et ainsi rendre la cellule plus autonome et ses mouvements plus naturels.

Brock et al. [2008] résument la manipulation robotique en tenant compte des schémas de commande et de la planification des mouvements pour un assemblage. Le point de vue se veut haut-niveau et fait une bonne introduction à cette problématique.

Peternel et al. [2015] posent le problème de l'adaptabilité d'un bras manipulateur robotique à son environnement, spécifiquement si le robot vient à entrer en collision avec celui-ci. En effet, les bras robotiques sont très raides et peuvent entraîner des dégâts non négligeables si celui-ci venait percuter des éléments de son entourage. En partant du principe que le bras humain est capable de modifier son impédance à la demande, leur but est de faire apprendre au robot cette capacité. Le moyen utilisé est un système de téléopération où l'organe de contrôle est une interface haptique qui détermine la position du bras de l'opérateur, ainsi qu'une commande pour contrôler l'impédance du bras manipulateur. Ils proposent une nouvelle interface de commande en impédance où l'opérateur a la possibilité de moduler la raideur du robot en temps réel. Cette valeur est ensuite utilisée pour consolider un système d'apprentissage en tenant compte de la proximité de l'effecteur par rapport à son environnement proche.

Critique Les paragraphes précédents donnent un aperçu des différents travaux qui ont été entrepris pour utiliser un robot dans une tâche d'assemblage. Beaucoup d'outils sont disponibles et peuvent être utiles.

Cependant, aucun de ces outils n'apportent une garantie de succès de l'opération. Peternel et al. [2015] ne proposent que la garantie que le robot ne détruise pas son environnement de manière autonome. De la même manière dans la construction par une équipe de robots, Schoen and Rus [2013] établissent que la remarque que l'intégrité de la structure ne peut pas être vérifiée et que leurs expériences ne sont pas répétables car les robots sont aveugles. Chen and Liu [2012] ou Ott et al. [2015b] permettent au robot d'être souple et de ne pas endommager l'environnement en cas de collisions non désirées (non attendues). Les cadres de spécification d'une opération d'assemblage ont la capacité intrinsèque de prendre en compte les informations relatives à l'environnement réel du robot, mais leurs propositions n'intègrent pas les problèmes d'écart entre le monde réel et son modèle virtuel.

En règle générale, la communauté met en avant les schémas de commande hybrides qui

tiennent compte d'informations en effort en tant que solution pour la gestion des interactions avec l'environnement. Le besoin est que le robot ne doit pas endommager son environnement en cas de collisions fortuites. Néanmoins s'il existe un tel risque de collisions, celui-ci s'explique par le fait que les mouvements du robot sont calculés sur la base d'un modèle virtuel le plus souvent erroné et ne représentant pas d'une bonne manière la réalité de l'environnement. Nous considérons que la prise en compte du monde réel lors de l'exécution des mouvements d'un robot est un aspect déterminant dans le succès d'un assemblage. Ce point nous servira de base pour la construction de la problématique de ce travail de thèse.

1.4 Problématique

Cette section pose les bases du problème de manipulation de pièces manufacturées et présente la problématique de ce travail de thèse.

Les sections précédentes ont présenté d'une manière générale quelle était la mission d'un opérateur dans une usine manufacturière. Il a été fait aussi le détail d'une opération de manipulation et de positionnement d'une pièce d'un point de vue opérateur ainsi que d'un point de vue robot. Nous nous sommes concentrés sur l'aspect de macro-manipulation qui englobe les étapes de localisation de la pièce, de saisie, d'acheminement et de dépose de celle-ci sur une structure fixe. Dans cette optique, nous avons fait l'inventaire de toutes les étapes nécessaires si le robot devait réaliser un assemblage de la même manière qu'un opérateur.

Il a été identifié dans l'état de l'art que le monde réel n'était pas pris en compte pendant l'exécution des mouvements du robot. De fait, avec des outils robotiques classiques, le succès d'une opération d'assemblage ne peut être garanti. De même, aucun outil n'a été proposé pour évaluer si l'opération était réussie ou non. Ainsi à la fin du mouvement sans information spécifique, l'utilisateur n'apporte qu'une hypothèse sur le succès de l'opération. L'état de l'art a aussi démontré que les travaux se reposaient majoritairement sur l'utilisation de modèles. La planification du mouvement des bras telle qu'effectuée dans [Chen and Liu, 2012] est fortement liée aux modèles volumiques de la pièce et du robot. La transposition du travail vers une cellule réelle peut être problématique, voire impossible si les écarts entre la réalité et les modèles sont trop importants.

Le concept de modélisation est un principe aussi vieux que les sciences fondamentales comme les mathématiques ou la physique. On peut définir un modèle par la traduction d'une observation d'un phénomène quel qu'il soit. Ce modèle nous permet d'analyser celui-ci et de prévoir des résultats par l'application d'une théorie. Toutes les observations du monde physique peuvent être rattachées à un modèle particulier. Par exemple, il est possible de décrire de manière très simple la chute d'une pierre accélérée par la gravité par une équation mathématique. Cette équation qui constitue le modèle, peut être utilisée pour déterminer le temps de la chute. Le choix du modèle est souvent lié à ce que nous souhaitons évaluer, déterminer et prédire. Un aspect très important d'un modèle est que celui-ci n'est qu'une description approximative d'une réalité très complexe : il n'est jamais vraiment parfait et n'est pas totalement représentatif de la réalité. Toutes les disciplines utilisent des modèles. Outre la finalité de celui-ci, l'identification du modèle est basée sur un compromis entre la finesse de la description de l'objet ou du phénomène réel, les ressources disponibles (existence d'un environnement mathématique, disponibilité des ressources informatiques, etc.) et les performances recherchées (temps raisonnable de calcul d'une prédiction, etc.).

L'écart d'un modèle par rapport à la réalité constitue un risque non négligeable dans le choix d'une représentation. Par cette action, l'utilisateur introduit délibérément des approximations dans

son système. Ces approximations peuvent être assumées lorsque le choix du modèle est contraint ou que les ressources à sa disposition sont limitées (cadre mathématique incomplet, mesures incomplètes, capacités de traitement limitées, etc.). D'autres approximations peuvent s'ajouter au cours du temps à cause de l'évolution du monde réel (vieillesse, déplacement des éléments, fatigue des matériaux, etc.). Les conséquences pour un robot manipulateur sont directement visibles sur leurs performances : ces écarts de modèles peuvent entraîner une erreur de positionnement de l'effecteur dans son propre référentiel ou encore peuvent entraîner un risque de collision. La majeure partie des mouvements d'un robot industriel est basée sur les modèles de ces trajectoires elles-mêmes. Lorsque celui-ci effectue une trajectoire, il n'apporte – sauf en de rares exceptions – aucune garantie à l'utilisateur que celui-ci a correctement exécuté son chemin. Il revient à l'utilisateur grâce à son expérience, d'estimer si l'hypothèse de réussite est validée ou non.

La conception d'une cellule robotique, de son implantation à sa validation par simulation, repose entièrement sur des modèles approximatifs. Alors même que le robot évolue dans le monde réel, celui-ci n'est pris en compte nul part au moment de l'exécution du mouvement demandé. Les approximations de modèles sont alors répercutées dans ce mouvement et entraînent un comportement qui n'est pas celui attendu par le concepteur de la cellule.

La figure 1.6 montre un diagramme décrivant le cheminement des informations pour la génération et l'exécution d'une trajectoire lors d'une opération classique de saisie et de dépose (en anglais, *pick and place*) telle qu'elle est appliquée dans le milieu industriel. À partir de la description d'une opération d'assemblage et de l'injonction des informations connues *a priori* sur le contexte de l'opération (modèles), on en déduit une trajectoire que le robot réalisera. Une fois la trajectoire jouée, l'opération se termine : le schéma de commande du robot garantit alors que son effecteur se trouve dans la position finale désirée. En aucun cas cette garantie ne s'applique à la pose de la pièce, alors que celle-ci est au cœur de l'opération. On note qu'à aucun moment du cheminement, les caractéristiques réelles de la pièce ou de la structure ne sont prises en compte. Par ce diagramme, nous souhaitons montrer que l'opération s'effectue de manière aveugle et qu'il n'y a aucun lien direct entre la définition de l'opération et le résultat obtenu après que le mouvement ait été effectué. Ce lien est établi par l'utilisateur qui estime si la position finale du robot réalise l'opération demandée ou non. Dans de nombreuses cellules d'automatisation, des mouvements supplémentaires de recalage ou des systèmes annexes de positionnement sont utilisés pour contenir l'incertitude de la pose de la pièce, en amont ou en aval de l'action du robot. Ces méthodes viennent remédier aux erreurs des robots qui sont principalement dues à la génération de trajectoire basée sur de mauvais modèles. Ces solutions ne sont pas génériques et ne peuvent être considérées comme pérennes.

À l'heure actuelle, les méthodes utilisées empêchent la réalisation d'une opération d'assemblage. Le cadre actuel ne permet pas au robot d'inclure des caractéristiques du monde réel et retire par définition toute capacité d'adaptation. Il n'apporte aucune garantie intrinsèque sur le succès de l'opération : celle-ci est soumise au jugement de l'opérateur et/ou du concepteur de la cellule robotique, ce qui rend *de facto* l'opération fortement dépendante du facteur humain.

L'introduction des caractéristiques du monde réel pendant les phases d'exécution du mouvement dans une opération robotique est une condition qui ne peut être négligée si les robots doivent gagner en autonomie. Ils doivent non seulement s'adapter aux conditions dans lesquelles ils évoluent réellement (les robots n'évoluent pas dans leur monde virtuel), mais aussi apporter des garanties sur le déroulement d'une opération. Néanmoins, nous savons maintenant que le cadre ne s'y prête pas. Et que le concepteur, l'utilisateur, l'opérateur, etc., porte la responsabilité de dire si

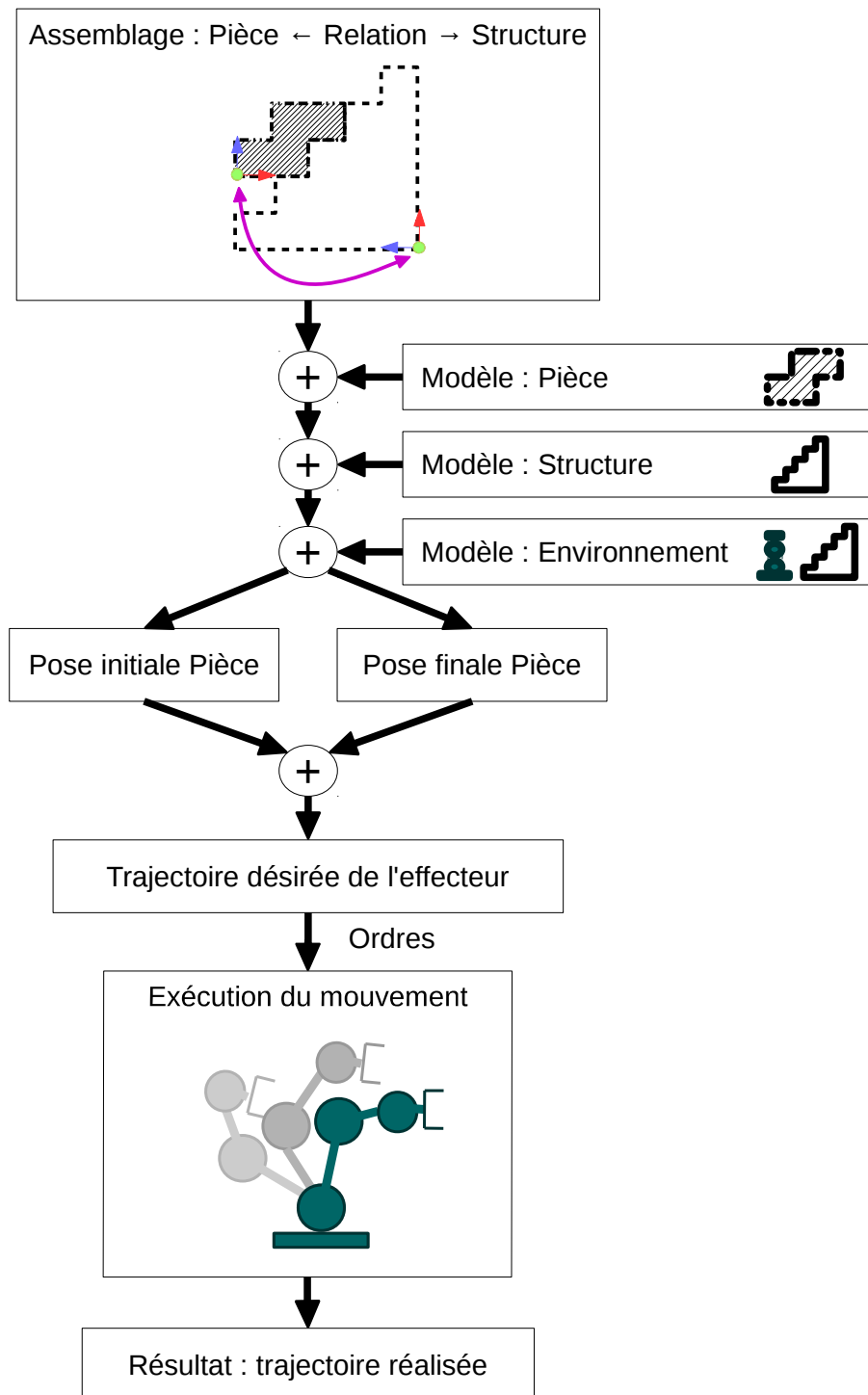


FIGURE 1.6 – Diagramme pour la génération d'une trajectoire dans une expérience classique de prise et de dépôt d'une pièce (*pick and place*) dans un cadre d'automatisation

le robot a correctement effectué le travail qui lui a été demandé.

Ce travail de thèse se concentre sur l'inclusion du monde réel au sein de la loi de commande en prenant pour base les outils d'asservissements visuels. Un nouveau cadre pour les opérations d'assemblage va être décrit dans la section suivante.

1.5 Approche proposée

Au regard des éléments avancés, on pose le constat suivant : le cadre de programmation robotique tel qu'il est appliqué à l'heure actuelle ne permet pas au robot de faire des tâches telles qu'un assemblage. D'une part, les modèles utilisés entraînent trop d'approximations qui impactent directement la précision absolue du robot. D'autre part, il s'avère que les robots industriels sont aveugles et ne sont pas contrôlés avec les éléments de l'espace dans lequel ils évoluent réellement, ce qui les rend intrinsèquement dangereux.

L'approche que nous proposons tend à redéfinir un cadre pour la programmation robotique en généralisant le recours aux techniques d'asservissement visuel et de commande référencée capteurs. Ces techniques permettent entre autres de créer des asservissements en se basant sur des observations du monde réel. Les robots sont au cœur de ces boucles fermées : leur mouvement permet la réalisation de l'objectif d'asservissement en alignant les-dites observations à des valeurs désirées. Le lecteur pourra trouver une définition plus précise de la commande référencée capteurs dans la section 2.1 qui lui est dédiée.

L'approche apporte plusieurs bénéfices. En premier lieu, le fait de contrôler le robot en tenant compte des informations du monde dans lequel il évolue réellement est un aspect important de celle-ci. Outre le fait que les robots soient des machines capables d'une excellente répétabilité, il sera toujours compliqué d'accorder notre confiance à une machine aveugle qui ne sait pas s'adapter. De plus, comme le montre le diagramme de la figure 1.7, le mouvement n'est plus défini exclusivement sur la base d'informations connues *a priori* en se basant sur des modèles, ce qui a pour conséquence directe de restreindre la propagation des erreurs d'approximations. D'autres sources d'erreurs sont quant à elles, intégrées au sein de la boucle de régulation, ce qui permet – en principe – de les corriger pendant l'exécution de l'asservissement.

La figure 1.7 montre un diagramme qui résume le nouveau cadre de programmation robotique pour les tâches d'assemblage. À partir de la même matière première de la spécification de l'opération qui établit un lien relationnel entre une pièce et la structure, la nouvelle approche redéfinit ce lien en le décomposant en relations élémentaires. Les relations élémentaires décrivent des liens entre les caractéristiques élémentaires qui peuvent être des points d'intérêt topologiques sur la pièce (par exemple, un plan, une arête ou un coin). Ces relations élémentaires sont alors confrontées aux relations réelles qui ont été mesurées par des capteurs extéroceptifs. À la fin de l'opération, le cadre garantit la correspondance entre les relations élémentaires réelles et les relations élémentaires désirées. Implicitement, l'approche garantit que l'opération s'est bien déroulée, puisque le but du mouvement du robot est de réaliser la relation pièce-structure définie dans l'opération. Le lien entre l'état final du robot et la définition de la tâche est direct, chose qui était laissée au jugement de l'opérateur dans l'approche précédente.

Les outils qui ont servi de base à ce travail proviennent du domaine de recherche lié aux lois de commande par capteurs extéroceptifs, autrement nommées commandes référencées capteurs [Chaumette \[1990\]](#). Issue des approches du type "regarde et avance" (en anglais, *look and move*) [[Hutchinson et al., 1996](#)], l'expérience la plus célèbre est sans doute l'asservissement vi-

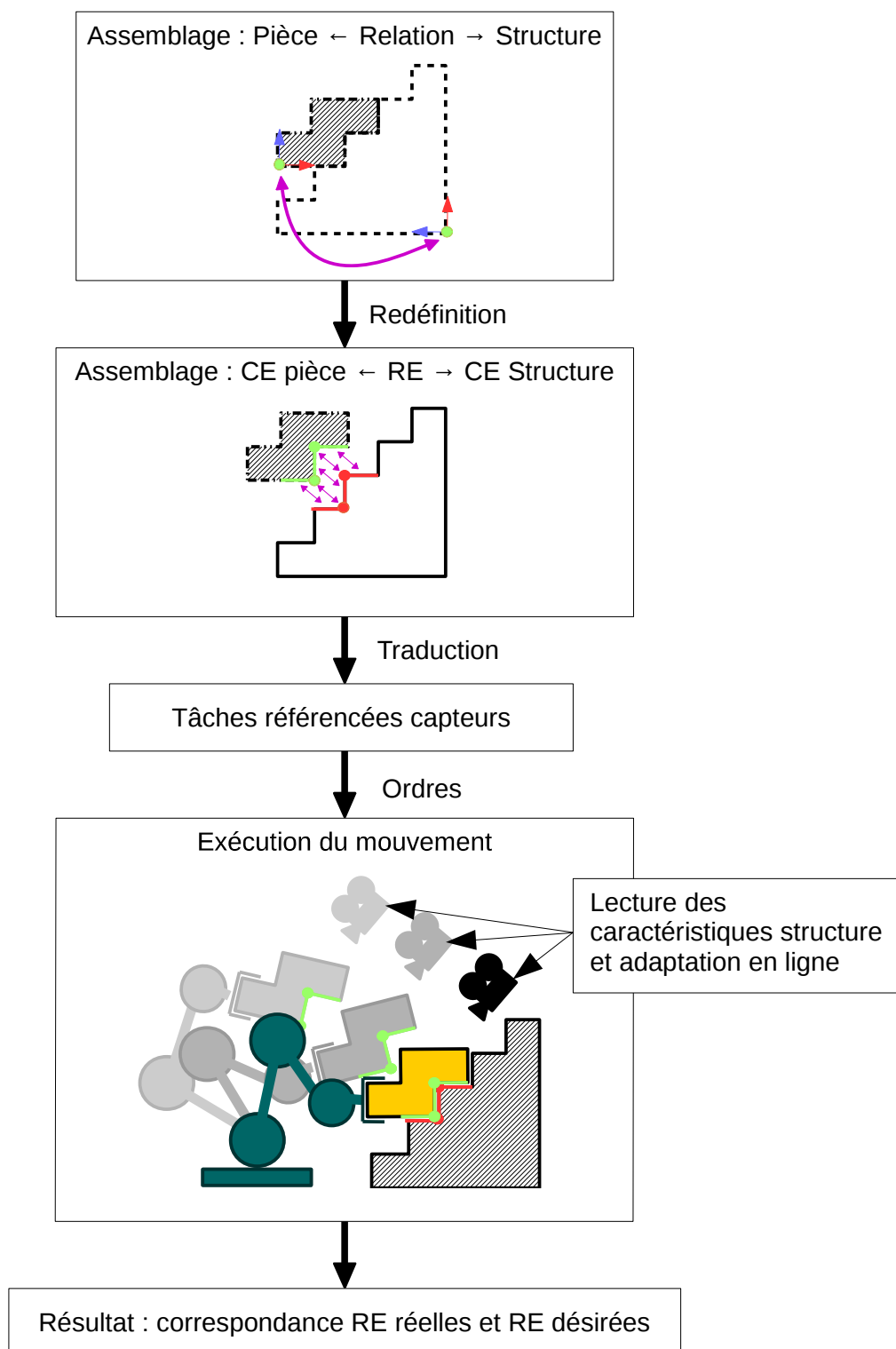


FIGURE 1.7 – Nouveau cadre de programmation robotique se basant sur l’inclusion des caractéristiques du monde réel dans la loi de commande du robot. Les étiquettes *CE* et *RE* sont les acronymes respectifs pour Caractéristiques Élémentaires et Relation Élémentaire.

suel [Chaumette and Hutchinson, 2006; Chaumette et al., 2007; Chaumette and Hutchinson, 2008] où le but est de contrôler un robot par l'intermédiaire d'une caméra fixée à son effecteur.

Néanmoins, ces schémas de commande ne sont pas suffisamment armés pour la définition d'un nouveau cadre de programmation robot. D'une part, la plupart de ces schémas de commande sont essentiellement des schémas de commande cinématique où l'information délivrée est une information de vitesse (vitesse cartésienne et/ou vitesses articulaires). D'autre part, pour apporter une solution complète aux problèmes d'assemblage, il est nécessaire de prendre en compte de nouveaux types d'information pour les contraintes d'effort et de force. Ces contraintes doivent pouvoir être intégrées aisément dans un schéma d'ordonnancement de tâche.

En addition au cadre précédemment proposé et du contrôle en effort, nous souhaitons prendre en compte d'autres classes d'application telles que la peinture ou la gravure. Ces classes ont des besoins identiques à l'assemblage (positionnement du robot par rapport à l'environnement). Mais elles ont aussi besoin d'un contrôle plus fin du comportement du robot, notamment en termes de vitesse de déplacement. C'est pourquoi nous proposons une nouvelle formulation de la commande référencée capteurs qui prend en compte les modèles de projection des caractéristiques du second ordre ainsi qu'un découplage dynamique. Cette nouvelle formulation nous apporte la possibilité de spécifier des contrôleurs sur l'erreur en position mais aussi sur l'erreur en vitesse. En outre, cette formulation reste compatible avec les schémas de commande hybrides pour la gestion des interactions physiques.

Le but global recherché de l'approche est de simplifier la programmation robotique pour les tâches d'assemblage et pour d'autres applications. En effet, la loi de commande se base principalement sur des mesures de caractéristiques qui font partie du monde réel et *de facto*, rend moins obscure le fonctionnement d'un robot du point de vue opérateur. L'approche se veut aussi compatible avec un point de vue industriel. En effet, les relations élémentaires introduites ci-dessus peuvent être produites à partir d'une maquette numérique, outil de conception indispensable de nos jours dans l'assemblage de grands ensembles tels que des voitures ou des avions. Outre la vision axée sur l'opération d'assemblage, l'approche est taillée pour la prise en compte du contexte général du robot (contraintes robotiques, évitement de l'opérateur collaborateur, etc.), ce qui permettra une nouvelle forme de travail collaboratif dans de nouvelles cellules robotiques sécurisées.

1.6 Organisation du manuscrit

Le document est organisé en quatre chapitres.

Le chapitre 1 a introduit et mis en place le contexte dans lequel ce travail de thèse s'inscrit. Il présente l'aspect historique du travail manufacturier ainsi que le contexte général de ce projet de thèse.

Les différents aspects du travail effectué par un ouvrier ont été présentés. Nous avons présenté aussi les détails d'une transposition possible de ce travail vers une cellule robotique automatisée. L'état de l'art se concentre sur les efforts de la communauté scientifique dans le champ de recherche spécifique lié aux opérations d'assemblage avec un ou plusieurs robots et sur des parties plus spécifiques telles que la saisie ou la génération de trajectoire. On présente aussi la problématique de ce travail de thèse et l'approche que nous proposons pour solutionner une opération d'assemblage avec un robot, ainsi que les pistes qui sont envisagées.

Le chapitre 2 se concentre plus particulièrement sur la proposition d'un nouveau cadre de

programmation robotique en généralisant le recours à la commande référencée capteurs et aux outils d'ordonnancement de tâches. On y fera une présentation détaillée de la commande référencée capteur par un état de l'art. Puis, nous détaillerons les éléments du nouveau cadre et comment les opérations seront prises en compte selon les capteurs disponibles et la situation des éléments de la scène. Cette nouvelle approche a été validée par la simulation d'un positionnement précis sans contact d'une pièce à placer sur une structure.

À partir des observations issues de l'application de ce nouveau cadre de programmation, le chapitre 3 définit de nouvelles formulations de la commande référencée capteurs en se basant sur le modèle cinématique du second ordre et les équations de découplage dynamique. Nous avons effectué plusieurs simulations qui viendront valider cette nouvelle formulation.

Le chapitre 4 se concentre sur l'intégration des informations d'effort à travers la définition de nouvelles commandes hybrides avec la commande dynamique référencée capteurs. Nous avons effectué une étude de cas typique s'inspirant d'une opération d'assemblage réel.

Positionnement sans contact

La section 1.5 pose les bases du travail de thèse. Il a été décrit une nouvelle approche pour la programmation robotique dans le but d'apporter une solution aux problématiques d'opérations d'assemblage réalisées par un robot. Le but recherché est d'intégrer les caractéristiques du monde réel au sein de la boucle de contrôle du bras manipulateur. La piste envisagée pour cela s'est concentrée sur la généralisation des outils d'asservissement visuel et plus généralement, ceux pour la commande référencée capteur et multi-capteurs.

Ce chapitre propose un état de l'art sur les concepts de commande référencée capteurs et en fait une brève présentation théorique. Puis, la nouvelle approche sera présentée dans le détail et celle-ci sera accompagnée de résultats de simulations pour validation.

2.1 Commande référencée capteurs : état de l'art

La commande référencée capteurs permet de construire une commande en boucle fermée pour l'asservissement de l'effecteur d'un robot. On définit un robot par une chaîne de corps rigides articulés capables de se mouvoir dans un espace opérationnel moyennant une commande particulière sur ses actionneurs. Cette chaîne de corps possède un ensemble de capteurs proprioceptifs et/ou extéroceptifs de manière à lui donner la capacité de percevoir son état interne (proprioceptif) et/ou sa situation par rapport à son environnement (extéroceptif).

Pour positionner l'effecteur du robot dans son espace opérationnel, la commande classique s'appuie sur l'utilisation intensive des capteurs proprioceptifs et la connaissance fine du modèle du robot. La commande proprioceptive est sensible aux erreurs d'identification du modèle et à toutes les imprécisions géométriques cumulées tout au long de la chaîne de corps.

La commande référencée capteur est une méthode qui se base sur l'utilisation d'un capteur, quelle que soit sa nature, et est particulièrement intéressante dans le cas de capteurs extéroceptifs. Il est fréquent de parler de tâche : une tâche dans la commande référencée capteurs est définie par la différence entre l'état actuel d'une ou plusieurs caractéristiques du signal qui émane du capteur et un état désiré défini par l'utilisateur. La réalisation de cette tâche est possible par le contrôle d'un

robot pour conduire le capteur vers l'état désiré. Par exemple, la tâche dans un asservissement visuel est de superposer quatre points observés dans l'image sur leurs positions désirées définies par l'utilisateur. Le robot déplace la caméra de sorte que les points mesurés atteignent leurs positions désirées. Sans perte de généralité, quatre éléments sont nécessaires pour définir une tâche pour une commande référencée capteur : la sélection des caractéristiques pour assurer la commande, la position générale du capteur par rapport à l'effecteur du robot, la mesure des caractéristiques extraites du signal capteur et leurs valeurs désirées définies dans le domaine des observations du capteur. Cette définition de la tâche se veut générique : il est possible de définir une tâche en utilisant n'importe quel type de système de mesure dès lors qu'il est possible de rassembler ces quatre éléments listés ci-dessus.

2.1.1 Modélisation classique de l'information capteur

La commande référencée capteurs a pour but de contrôler les degrés de liberté de l'effecteur d'un robot quelconque à travers des mesures effectuées par un capteur. Le plus populaire des exemples de cette commande est l'asservissement visuel (en anglais, *visual servoing*). [Chaumette and Hutchinson \[2006\]](#); [Chaumette et al. \[2007\]](#); [Chaumette and Hutchinson \[2008\]](#) donnent un aperçu sur les basiques de l'asservissement par un retour d'informations visuelles dont nous résumons les grandes lignes dans le paragraphe suivant. Cette commande s'appuie sur le formalisme de la fonction de tâche [[Samson et al., 1991](#)] où le but est d'amener l'erreur entre une valeur désirée et la valeur courante à zéro.

Le but est d'amener la chaîne poly-articulée à atteindre la mesure désirée en intégrant le signal capteur dans la boucle de contrôle. Tenant compte des deux approches possibles : soit le capteur est embarqué et mesure donc une information de son environnement ; soit le capteur est fixe et mesure une information liée à la pose de l'effecteur. Dans tous les cas, la mesure désirée correspond à une nouvelle pose à atteindre par l'effecteur dans l'espace opérationnel. Pour lier ces mesures aux déplacements des degrés de liberté internes du robot, on utilise son modèle cinématique.

Soit $\mathbf{s}$ un vecteur de dimension m représentant une mesure d'une caractéristique fixe de l'environnement.

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v} \quad (2.1)$$

où,

- $\mathbf{v} = [\mathbf{v}^\top \ \omega^\top]^\top$ est le torseur cinématique du capteur,
- $\mathbf{L}_s \in \mathbb{R}^{m \times 6}$ est la matrice d'interaction liée à la caractéristique mesurée.

L'expression analytique est déterminée par la nature du capteur : un exemple de matrice d'interaction est donnée dans [[Chaumette and Hutchinson, 2006](#)] pour un point 2D issu d'une mesure caméra.

Il est possible de lier le torseur $\mathbf{v}$ à celui de l'effecteur du robot, connaissant la position et l'orientation du repère du capteur par rapport à celui de l'effecteur. Soit ${}^S\mathbf{T}_N \in \mathbb{R}^{6 \times 6}$ la matrice de transformation spatiale cinématique du repère de l'effecteur N par rapport au repère du capteur S et $\mathbf{v}_N$ le torseur cinématique de l'effecteur du robot, on écrit :

$$\mathbf{v} = {}^S\mathbf{T}_N \mathbf{v}_N \quad (2.2)$$

où la matrice de transformation cinématique est définie par :

$${}^S\mathbb{T}_N = \begin{bmatrix} {}^S\mathbf{R}_N & [{}^S\mathbf{t}_N]_{\times} {}^S\mathbf{R}_N \\ \mathbf{0}_3 & {}^S\mathbf{R}_N \end{bmatrix} \quad (2.3)$$

L'équation ci-dessus comporte les éléments suivants :

- ${}^S\mathbf{R}_N \in SO(3)$ est la matrice de rotation dans l'espace entre les deux repères, exprimée par rapport au repère capteur,
- ${}^S\mathbf{t}_N \in \mathbb{R}^3$ est la quantité de translation entre l'origine du repère attaché au capteur et celle du repère attaché à l'effecteur, du point de vue du repère capteur.

Enfin, le torseur cinématique de l'effecteur $\mathbf{v}_N$ est déduit à partir modèle cinématique du robot en utilisant ${}^N\mathbf{J}_N$ la matrice cinématique exprimée dans le repère de l'effecteur. Il est possible de trouver une définition de la matrice cinématique d'une chaîne articulée et de son modèle cinématique dans [Khalil and Dombre, 2002]. Tenant compte des équations (2.1), (2.2) et du modèle cinématique direct du robot, on obtient :

$$\dot{\mathbf{s}} = \mathbf{L}_s {}^S\mathbb{T}_N {}^N\mathbf{J}_N \dot{\mathbf{q}} \quad (2.4)$$

Le produit $\mathbf{L}_s {}^S\mathbb{T}_N {}^N\mathbf{J}_N$ représente alors la jacobienne de tâche associée à la caractéristique $\mathbf{s}$. Pour simplifier la notation, la matrice cinématique du robot sera notée $\mathbf{J}$ dans la suite du document.

2.1.2 Commande cinématique à partir d'informations visuelles

Nous avons vu comment construire une tâche pour la commande référencée capteurs à travers un exemple typique d'asservissement visuel sur la base de point 2D, les quatre éléments à rassembler étant la pose du capteur par rapport à l'organe mobile du robot, l'encodage de la caractéristique lié au modèle de projection du capteur, la matrice d'interaction qui lie l'évolution de l'objet observé à sa celle de sa mesure et le vecteur de caractéristique désiré à atteindre.

Pour pouvoir commander le robot en tenant compte des informations visuelles sur la base de l'équation (2.4), il faut pouvoir déterminer le vecteur de vitesses articulaires $\dot{\mathbf{q}}$. Or l'inversion de cette équation n'est pas directe car non-linéaire. Les vitesses articulaires sont alors déterminées en utilisant une méthode d'optimisation de type Gauss-Newton. Ces outils et ces méthodes sont identiques à celles utilisées dans des problèmes d'odométrie visuelle (*Structure From Motion* ou SFM) et de localisation et cartographie simultanée (*Simultaneous Localization And Mapping* ou SLAM). Elles renvoient à la base vers les travaux de Lucas et Kanade (revisités par Baker and Matthews [2004]; Szeliski [2006]) sur l'ajustement de rayons, rapporté par Strasdat et al. [2010]; Kümmerle et al. [2011]; Strasdat [2012]. Malis and Chaumette [2000] effectuent un travail similaire.

On présente ensuite un bref rappel des différentes méthodes d'optimisation pour entrevoir l'intérêt de la méthode de Gauss Newton, puis comment celle-ci s'articule avec la commande référencée capteur. Enfin, on présentera l'équation de commande tirée de l'équation normale de la méthode d'optimisation de Gauss-Newton.

2.1.2.1 Méthodes d'optimisation

Comme mentionné ci-dessus, la commande référencée capteurs est basée sur les méthodes d'optimisation. Ce paragraphe effectue un bref rappel sur les méthodes itératives de la descente de gradient, celle de Newton, de Gauss Newton, puis de Levenberg-Marquardt.

Soit $F : \mathbb{R}^n \rightarrow \mathbb{R}$ un champ de scalaires, les problèmes d'optimisation classiques ont pour objectif de trouver le minimum de F . Le problème est noté de la manière suivante :

$$\min_{\mathbf{x} \in \mathbb{R}^n} F(\mathbf{x}) \quad (2.5)$$

Le plus souvent, le but est de trouver une solution locale au voisinage d'une solution initiale $\mathbf{x}_0$. D'autres méthodes garantissent de trouver une solution globale pour un problème donné mais celles-ci ne seront pas traitées ici.

Pour que le couple $(\bar{\mathbf{x}}, F(\bar{\mathbf{x}}))$ puisse être considéré comme un minimum local, deux conditions doivent être posées. La condition nécessaire est que le gradient de la fonction doit être nul :

$$\nabla F(\bar{\mathbf{x}}) = 0 \quad (2.6)$$

L'autre condition suffisante est que, si le hessien $\mathbf{H}_F(\bar{\mathbf{x}})$ est défini positif, alors

$$\forall \mathbf{y} \in \mathbb{R}^n \setminus \{0\} \quad \mathbf{y}^\top \mathbf{H}_F(\bar{\mathbf{x}}) \mathbf{y} > 0 \quad (2.7)$$

Descente de gradient Une première méthode consiste à suivre le gradient de la fonction pour déterminer le minimum local. En assumant que la fonction est dérivable, à partir d'une solution initiale proposée par l'utilisateur $\mathbf{x}_0$, la recherche de la solution se fait de manière itérative en calculant plusieurs pas $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k, \dots$, selon la direction négative du gradient. La valeur de la prochaine itération est déterminée à partir de la fonction suivante :

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla F(\mathbf{x}_k) \quad (2.8)$$

Le facteur α_k , strictement positif, est choisi de manière à imposer que la valeur de $F(\mathbf{x}_{k+1})$ soit plus petite que $F(\mathbf{x}_k)$. Ainsi, la convergence vers la solution où $\nabla F(\bar{\mathbf{x}})$ s'annule, est garantie. Plusieurs méthodes existent pour la sélection du facteur α_k : la recherche linéaire avec retour sur trace (en anglais, *backtracking line search*) ou encore la méthode du recuit simulé. L'inconvénient de la descente de gradient – outre le fait qu'elle nécessite une fonction dérivable – est que la convergence vers le minimum est lente, surtout au voisinage de celui-ci.

Méthode de Newton Une autre méthode locale est celle appelée méthode de Newton. Celle-ci possède un meilleur taux de convergence que la descente de gradient. Cependant, cette méthode ne peut distinguer les minima, maxima ou cols qui peuvent induire l'algorithme en erreur. Elle requiert que la fonction F soit dérivable deux fois et que le hessien soit défini positif dans le voisinage de la solution recherchée.

De la même manière que précédemment, à partir d'une solution initiale proposée $\mathbf{x}_0$ proche de la solution recherchée, on détermine la relation incrémentale pour chaque pas en se basant sur l'approximation du gradient $\nabla F(\bar{\mathbf{x}})$ en utilisant les séries de Taylor :

$$\nabla F(\bar{\mathbf{x}}) \approx \nabla F(\mathbf{x}_0) + \mathbf{H}_F(\mathbf{x}_0)(\bar{\mathbf{x}} - \mathbf{x}_0) \quad (2.9)$$

Étant donné que $\nabla F(\bar{\mathbf{x}}) = 0$, on obtient $\bar{\mathbf{x}} \approx \mathbf{x}_0 - \mathbf{H}_F^{-1}(\mathbf{x}_0)\nabla F(\mathbf{x}_0)$. On en déduit la formule récursive :

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_F^{-1}(\mathbf{x}_k)\nabla F(\mathbf{x}_k) \quad (2.10)$$

Il est possible d'appliquer la méthode de Newton par la résolution d'un système linéaire : soit la mise à jour incrémentale δ telle que $\delta = \mathbf{x}_{k+1} - \mathbf{x}_k$, le système linéaire est :

$$\mathbf{H}_F(\mathbf{x}_k)\delta = -\nabla F(\mathbf{x}_k) \quad (2.11)$$

Et la valeur de $\mathbf{x}$ à la prochaine itération $k + 1$ est :

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \delta \quad (2.12)$$

Contrairement à la descente de gradient, la méthode de Newton converge rapidement au voisinage de la solution. De plus, si F représente une fonction quadratique, cette méthode garantit de trouver la solution en une seule itération.

Méthode de Gauss-Newton La méthode de Gauss-Newton offre une variante de la méthode de Newton qui permet de s'affranchir du calcul du hessien de la fonction en question, souvent difficile à obtenir. Cette méthode s'adresse à une large classe de problèmes d'optimisation et est utilisable sur des fonctions de type champ de scalaires $F : \mathbb{R}^n \rightarrow \mathbb{R}$ qui peuvent s'exprimer ainsi :

$$F(\mathbf{x}) = \alpha \mathbf{d}(\mathbf{x})^\top \Lambda \mathbf{d}(\mathbf{x}) \quad (2.13)$$

où $\alpha > 0$, $\mathbf{d} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ est un vecteur de scalaires, et Λ une matrice symétrique définie positive. La fonction de coût $\mathbf{d}(\mathbf{x})$ peut prendre n'importe quelle forme. Cette méthode peut être utilisée pour estimer un modèle paramétrique $\mathbf{x}$ en minimisant le coût quadratique $\mathbf{d}_i(\mathbf{x}) = (s_i(\mathbf{x}) - s_i^d)^2$, par exemple.

En posant $\alpha = \frac{1}{2}$ et sachant que Λ est symétrique, la première dérivée de F est telle que :

$$\nabla F = \frac{1}{2} (\mathbf{d}(\mathbf{x})^\top \Lambda \mathbf{J}_d(\mathbf{x}))^\top + \frac{1}{2} (\mathbf{J}_d(\mathbf{x})^\top \Lambda \mathbf{d}(\mathbf{x})) = \mathbf{J}_d(\mathbf{x})^\top \Lambda \mathbf{d}(\mathbf{x}) \quad (2.14)$$

en posant $\mathbf{J}_d(\mathbf{x}) = \frac{\partial \mathbf{d}(\mathbf{x})}{\partial \mathbf{x}}$. De la même manière en utilisant le théorème de dérivation des fonctions composées (en anglais, *chain rule*), le hessien peut s'exprimer tel que :

$$\mathbf{H}_F(\mathbf{x}) = \mathbf{J}_d(\mathbf{x})^\top \Lambda \mathbf{J}_d(\mathbf{x}) + \mathbf{H}_d(\mathbf{x}) \Lambda \mathbf{d}(\mathbf{x}) \quad (2.15)$$

où $\mathbf{H}_d(\mathbf{x})$ représente le hessien du tenseur $\mathbf{d}(\mathbf{x})$. Au voisinage de la solution, on fait l'approximation $\mathbf{d}(\mathbf{x})|_{\mathbf{x} \rightarrow \bar{\mathbf{x}}} \approx \mathbf{0}$. De fait, le hessien de la fonction F devient :

$$\mathbf{H}_F(\mathbf{x})|_{\mathbf{x} \rightarrow \bar{\mathbf{x}}} \approx \mathbf{J}_d(\mathbf{x})^\top \Lambda \mathbf{J}_d(\mathbf{x}) \quad (2.16)$$

Cette approximation est possible pour de faibles valeurs de $\mathbf{d}(\mathbf{x})$ car le second terme de l'équation (2.15) peut être négligé. Cette propriété reste vraie lorsque la solution se trouve au voisinage du vrai minimum de F où la méthode de Newton atteint une convergence quadratique. De la même manière que dans le cas de la méthode de Newton, l'application de la méthode de Gauss-Newton peut passer par la résolution d'un système linéaire. L'équation (2.11) peut être réécrite sous la forme de l'équation normale suivante :

$$(\mathbf{J}_d^\top \Lambda \mathbf{J}_d) \delta = -\mathbf{J}_d^\top \Lambda \mathbf{d} \quad (2.17)$$

Méthode de Levenberg-Marquardt Cette dernière méthode peut être considérée comme un mélange de la méthode de la descente de gradient et celle de Gauss-Newton. En effet, la première converge vers des minima locaux et n'est pas affectée par des points cols ou des maxima locaux. Cependant, son taux de convergence est particulièrement mauvais surtout lorsque la méthode s'approche de la solution recherchée. Au contraire, les méthodes de type Newton ont un meilleur comportement autour du voisinage de la solution et converge plus rapidement que la descente de gradient.

La méthode de Levenberg-Marquardt propose une équation de mise à jour qui permet à l'aide d'un facteur μ strictement positif, de changer le comportement de l'équation normale :

$$(\mathbf{J}_d^T \Lambda \mathbf{J}_d + \mu \mathbf{I}) \delta = -\mathbf{J}_d^T \Lambda \mathbf{d} \quad (2.18)$$

Si μ tend vers zéro, le comportement de l'algorithme s'approche de celui de la méthode de Gauss-Newton. A l'inverse si μ tend vers l'infini, on obtient $\mathbf{J}_d^T \Lambda \mathbf{J}_d \ll \mu \mathbf{I}$ et l'équation normale (2.18) peut s'écrire sous la forme :

$$\delta = -\alpha \mathbf{J}_d^T \Lambda \mathbf{d} \quad (2.19)$$

où $\alpha \rightarrow 0$, ce qui a pour effet de réduire l'erreur par un petit pas infinitésimal si le minimum n'est pas atteint.

L'algorithme est appliqué ainsi : le pas $\mathbf{x}_k + \delta$ n'est accepté que si il réduit l'erreur, *i.e.* $F(\mathbf{x}_k + \delta) \ll F(\mathbf{x}_k)$. Si le pas est accepté, on assume que nous nous rapprochons de la solution, ainsi μ est réduit pour que l'algorithme se comporte un peu plus selon la méthode de Gauss-Newton. A l'inverse, si le pas est rejeté et n'a pas un effet positif sur l'erreur, μ est choisi plus large de manière à ce que la direction de recherche soit celle où la pente est la plus forte.

2.1.2.2 Méthode de Gauss-Newton, groupe de Lie et commande référencée capteurs

L'asservissement visuel, et plus généralement la commande référencée capteurs, est un moyen de commander le mouvement d'un système dans l'espace. Ce système peut être une caméra embarquée sur l'effecteur d'un robot, ou une cible embarquée observée par une caméra fixe. La section 2.1.3 montre la construction d'une tâche d'asservissement visuel où le but, à travers l'observation des points 2D dans l'image, est de faire coïncider la pose du repère optique courant de la caméra embarquée avec celui désiré. Autrement dit, il est nécessaire de déterminer le chemin que prendra le repère optique de la caméra pour parvenir à faire coïncider les stimuli visuels avec ceux désirés. La figure 2.1 montre un schéma résumant la situation.

Pour ce faire, il est nécessaire d'introduire le cadre mathématique pour décrire l'espace de configuration des corps rigides et des repères. Le cadre en question est le groupe de Lie SE3 (en anglais, *Special Euclidian group*) qui est le groupe des transformations rigides (translations et rotations) dans l'espace à trois dimensions. La pose est définie par l'association d'une rotation notée $\mathbf{R} \in \text{SO3}$ et d'une translation notée $\mathbf{t} \in \mathbb{R}^3$. SO3 représente le groupe des rotations (en anglais, *Special Orthogonal group*). En considérant un point $\mathbf{x}$ dans l'espace, la relation de passage entre la définition de ce point dans le repère A et celle dans le repère B est définie par :

$${}^A\mathbf{x} = {}^A\mathbf{R}_B {}^B\mathbf{x} + {}^A\mathbf{t}_B \quad (2.20)$$

L'indice supérieur à gauche indique dans quel repère est définie la caractéristique géométrique. On lit ${}^A\mathbf{R}_B$ comme étant la matrice de rotation du repère B, exprimée par rapport au repère A. Cette

relation peut être écrite sous une forme matricielle à l'aide des coordonnées homogènes telle que :

$$\begin{bmatrix} {}^A\mathbf{x} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^A\mathbf{R}_B & {}^A\mathbf{t}_B \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^B\mathbf{x} \\ 1 \end{bmatrix} = {}^A\mathbf{T}_B \begin{bmatrix} {}^B\mathbf{x} \\ 1 \end{bmatrix} \quad (2.21)$$

Grâce à la notation matricielle homogène, le passage récursif entre plusieurs repères est effectué par une simple multiplication matricielle. Ci-après, est définie l'équation de passage du repère C vers le repère A en passant par B :

$${}^A\mathbf{x} = {}^A\mathbf{T}_B {}^B\mathbf{T}_C {}^C\mathbf{x} \quad (2.22)$$

On note que la notation classique est confondue avec la notation homogène dans l'équation précédente. Sauf mention contraire et explicite, celles-ci seront confondues dans le reste du document.

On définit $\mathfrak{se}3$ l'espace tangent à SE3 à l'identité, représentant l'algèbre de Lie isomorphe au groupe de Lie des champs équiprojectifs de l'espace euclidien à dimension 3. Autrement dit, tout élément appartenant à l'algèbre de Lie $\mathfrak{se}3$ peut être assimilé au torseur cinématique $\mathbf{v}$ [Chaumette, 1990]. De plus, il est possible de définir l'application exponentielle (*exponential map*) en tant qu'opérateur de passage entre l'algèbre de Lie $\mathfrak{se}3$ et l'espace SE3 tel que :

$$\text{expm}(\mathbf{v}_{\mathfrak{se}3}) = \text{expm} \left(\begin{bmatrix} v \\ \omega \end{bmatrix}_{\mathfrak{se}3} \right) = \begin{bmatrix} \text{expm}([\omega]_{\times}) & Vv \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \in \text{SE3} \quad (2.23)$$

avec

$$\text{expm}([\omega]_{\times}) = \begin{cases} \mathbf{I} + [\omega]_{\times} + \frac{1}{2}[\omega]_{\times}^2 & \text{quand } \theta \rightarrow 0 \\ \mathbf{I} + \frac{\sin(\theta)}{\theta}[\omega]_{\times} + \frac{1-\cos(\theta)}{\theta^2}[\omega]_{\times}^2 & \text{sinon} \end{cases} \quad \text{avec } \theta = \|\omega\|_2 \quad (2.24)$$

et

$$V = \begin{cases} \mathbf{I} + \frac{1}{2}[\omega]_{\times} + \frac{1}{6}[\omega]_{\times}^2 & \text{quand } \theta \rightarrow 0 \\ \mathbf{I} + \frac{1-\cos(\theta)}{\theta^2}[\omega]_{\times} + \frac{\theta-\sin(\theta)}{\theta^3}[\omega]_{\times}^2 & \text{sinon} \end{cases} \quad \text{avec } \theta = \|\omega\|_2 \quad (2.25)$$

De même que pour le passage entre $\mathfrak{se}3$ et SE3, l'équation (2.24) représente l'opérateur de passage entre $\mathfrak{so}3$ et SO3. Celle-ci est plus communément appelée la *formule de Rodriguez*.

Soit ${}^W\mathbf{T}_S(t)$ le chemin parcouru par le repère du capteur S exprimé par rapport à un repère quelconque W, il est alors possible d'écrire l'évolution de la pose du repère W telle que :

$${}^W\mathbf{T}_S(t) = {}^W\mathbf{T}_S(t_0) \text{expm}(\mathbf{v}(t)) \quad (2.26)$$

Le problème d'optimisation de style Gauss-Newton est formulé de la manière suivante : en prenant pour objectif une minimisation de distance entre la valeur mesurée $s_i ({}^W\mathbf{T}_S(t))$ et la valeur désirée s_i^d d'une caractéristique i , le problème d'optimisation est défini par :

$$\min_{\mathbf{v} \in \mathbb{R}^6} \sum_i s_i ({}^W\mathbf{T}_S \text{expm}(\mathbf{v})) - s_i^d \quad (2.27)$$

En considérant la méthode de Gauss-Newton présentée précédemment, l'équation normale devient :

$$(\mathbf{J}_s^T \mathbf{J}_s) \mathbf{v} = -\mathbf{J}_s^T \lambda (\mathbf{s} - \mathbf{s}^d) \quad (2.28)$$

où $\mathbf{J}_s$ définie par :

$$\mathbf{J}_s = \frac{\partial s_i ({}^W\mathbf{T}_S \text{expm}(\mathbf{v}))}{\partial \mathbf{v}} \quad (2.29)$$

n'est autre que la matrice d'interaction $\mathbf{L}_s$ définie dans l'équation (2.1).

En supposant le produit de matrice $\mathbf{J}_s^T \mathbf{J}_s$ de rang plein, la loi de commande référencée capteur peut s'écrire sous la forme suivante :

$$\mathbf{v} = -\lambda (\mathbf{J}_s^T \mathbf{J}_s)^{-1} \mathbf{J}_s^T (\mathbf{s} - \mathbf{s}^d) \quad (2.30)$$

On reconnaît ici le produit $(\mathbf{J}_s^T \mathbf{J}_s)^{-1} \mathbf{J}_s^T$ comme étant la pseudo-inverse de Moore-Penrose à droite que l'on note $\mathbf{J}_s^+$. Ainsi, l'équation de commande devient :

$$\mathbf{v} = -\lambda \mathbf{J}_s^+ (\mathbf{s} - \mathbf{s}^d) \quad (2.31)$$

En tenant compte du modèle cinématique direct du robot ainsi que de l'équation (2.2), il est possible d'écrire l'équation suivante :

$$\dot{\mathbf{q}} = -\lambda \mathbf{J}^{-1N} \mathbb{T}_S \mathbf{J}_s^+ (\mathbf{s} - \mathbf{s}^d) \quad (2.32)$$

En posant $\mathbf{J}_s = \mathbf{L}_s$, on obtient l'écriture définitive de l'équation de la commande référencée capteurs telle que :

$$\dot{\mathbf{q}} = -\lambda (\mathbf{L}_s^S \mathbb{T}_N \mathbf{J})^+ (\mathbf{s} - \mathbf{s}^d) \quad (2.33)$$

En considérant cette dernière équation, en posant $\mathbf{e} = (\mathbf{s} - \mathbf{s}^d)$ l'erreur exprimée dans l'espace capteur, il est possible de dire que l'évolution de l'erreur est fonction plus ou moins directement des vitesses articulaires telle que $\dot{\mathbf{e}} = f(\dot{\mathbf{q}})$. Par contraction, le système précédent peut s'écrire sous la forme suivante :

$$\dot{\mathbf{e}} \equiv -\lambda \mathbf{e} \quad (2.34)$$

Par conséquent, le système robotique appliquant la loi commande (2.33) devient équivalent à un système du second ordre où, dans le cas idéal, la solution pour $\mathbf{e}$ assure une décroissance exponentielle de l'ensemble de ses composantes. Le scalaire λ joue le rôle ici d'un gain agissant sur la vitesse de convergence du système vers zéro.

On note que la méthode de Gauss-Newton s'applique sur un très large ensemble de problèmes dont la commande référencée capteurs. Cela s'explique par le fait que le hessien de la fonction d'objectif est approximé par le produit des matrices $\mathbf{J}_s^T \mathbf{J}_s$. En asservissement visuel, [Malis \[2004\]](#) apporte la preuve que la demi-somme des matrices d'interaction à l'état courant et à l'état d'équilibre offre une meilleure approximation du hessien.

2.1.3 Exemple de la construction d'une tâche d'asservissement visuel

L'objectif ici est d'illustrer l'application de la modélisation classique par un exemple d'asservissement visuel avec une caméra embarquée. La figure 2.2 montre un équipement pour la réalisation d'un asservissement visuel basé image (en anglais, *image-based visual servoing* ou IBVS).

Les caractéristiques utilisées sont quatre points à deux dimensions relevés dans l'image. La matrice de passage entre le repère de la caméra S et le repère de l'effecteur N est connue, calibrée au préalable. Le lecteur pourra trouver la méthode de [Tsai and Lenz \[1989\]](#) pour obtenir cette matrice. De la même manière, la caméra est calibrée et les paramètres intrinsèques (distance focale, taille des pixels, etc.) sont connus.

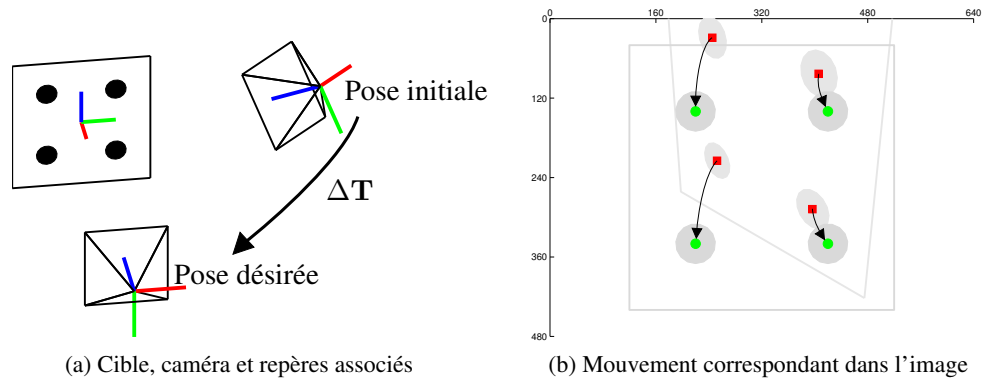


FIGURE 2.1 – Schéma d'illustration pour un asservissement visuel 2D avec quatre points sur une cible.

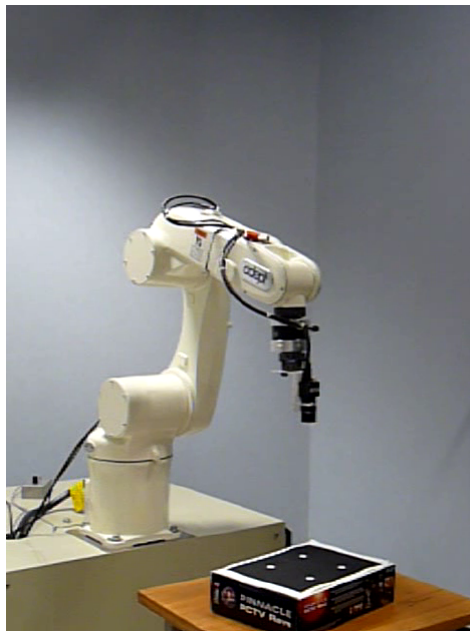


FIGURE 2.2 – Exemple d'un asservissement visuel 2D typique avec une caméra embarquée sur l'effecteur du robot [Kermorgant, 2011].

L'encodage mathématique de la caractéristique du point 2D est nécessaire pour déterminer la matrice d'interaction $\mathbf{L}_s$. Celui-ci est fortement lié au modèle mathématique du capteur, ici une caméra. Un des modèles possibles pour réaliser une tâche d'asservissement visuel est le modèle de la caméra sténopé (en anglais, *pinhole camera*). Il simplifie l'ensemble caméra par une chambre noire avec un trou d'épingle sans lentille. Ce modèle permet d'écrire la relation entre un point 3D du monde réel observé par la caméra et sa projection sur le plan image.

Soit $\mathbf{x}_i = [X_i \ Y_i \ Z_i]^\top$ le point 3D observé exprimé par rapport au repère de la caméra et $\mathbf{x}_{n,i} = [x_i \ y_i]^\top$ la coordonnée normalisée sur le plan focal de la caméra, la relation est définie telle que :

$$x_i = \frac{S X_i}{S Z_i} ; \quad y_i = \frac{S Y_i}{S Z_i} \quad (2.35)$$

Le point projeté sur l'image est défini par $\mathbf{x}_{p,i} = [u_i \ v_i]^\top$ tel que :

$$u_i = u_0 + F_u x_i ; \quad v_i = v_0 + F_v y_i \quad (2.36)$$

où F_u, F_v, u_0 et v_0 représentent les paramètres intrinsèques de la caméra.

A ce stade, en tenant compte de la connaissance des paramètres intrinsèques, la caractéristique des points observés peut être exprimée à la fois dans l'espace du plan focal de la caméra ou dans l'espace de l'image. Le choix est arbitraire. La matrice d'interaction pour un point 2D dans l'espace du plan focal est définie ci-après [Chaumette and Hutchinson, 2006] :

$$\mathbf{L}_{s,i} = \begin{bmatrix} -\frac{1}{Z_i} & 0 & \frac{x_i}{Z_i} & x_i y_i & -1 - x_i^2 & y_i \\ 0 & -\frac{1}{Z_i} & \frac{y_i}{Z_i} & 1 + y_i^2 & -x_i y_i & -x_i \end{bmatrix} \quad (2.37)$$

Quatre points 2D sont utilisés pour réaliser la tâche. L'équation de tâche (2.4) se réécrit sous la forme suivante en empilant les matrices d'interaction pour les quatre points :

$$\dot{\mathbf{s}} = \begin{bmatrix} \dot{s}_1 \\ \vdots \\ \dot{s}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{s,1} \\ \vdots \\ \mathbf{L}_{s,4} \end{bmatrix} {}^S \mathbb{T}_N \mathbf{J} \dot{\mathbf{q}} \quad (2.38)$$

Le dernier élément à définir, pour terminer cette construction de tâche référencée capteurs, est le vecteur de référence à atteindre. Celui-ci peut être déterminé en tenant compte de la pose désirée de la caméra et de la fonction de projection directe (2.35) ou tout simplement apprise par l'enregistrement de la position des points quand la caméra est dans sa position désirée.

L'asservissement visuel est réalisé par l'application de la loi de commande (2.33). Le système converge alors de manière à ce que l'erreur suive une décroissance exponentielle jusqu'à sa stabilisation autour de zéro.

2.1.4 Caractéristiques géométriques pour l'assemblage

Il existe plusieurs types de caractéristiques mesurables qui peuvent être utilisées dans un cadre de commande référencée capteurs. Deux conditions sont nécessaires pour déterminer si une caractéristique peut être utilisée : il faut connaître la fonction de projection de la caractéristique (celle-ci dépend de la nature du capteur) et être capable d'extraire la-dite information du flux de données brutes. L'extraction de l'information utile peut nécessiter une certaine performance de l'algorithme d'extraction. On juge de la performance nécessaire en fonction de la dynamique du système à commander : plus le système bouge rapidement, plus la fréquence d'acquisition nécessaire augmente (théorème de Shannon).

Il est possible de catégoriser les différentes caractéristiques utilisées dans les expérimentations d'asservissement référencée capteur en deux grandes familles : celles qui sont exprimées dans un espace à trois dimensions et celles, dans un espace à deux dimensions. Les paragraphes suivants donnent un aperçu de ce que l'on peut trouver dans ces deux familles.

2.1.4.1 Caractéristiques de type 3D

La première caractéristique la plus évidente que l'on pourrait citer est le point trois dimensions (3D). De nombreux capteurs peuvent être utilisés pour mesurer un point 3D saillant dans la scène observée. Les capteurs habituellement utilisés sont les caméras stéréoscopiques, les caméras à profondeur de champ ou les télémètres de type LiDAR (*Light Detection And Ranging*). Chacun de ces capteurs fournit un flux de donnée 3D où il est possible de reconnaître et sélectionner un point saillant de la scène.

On peut considérer alors une fonction de projection du capteur telle que définie ci-dessous. Soit $\mathbf{x} = [x \ y \ z]^T$ un point 3D mesuré par le capteur et donc défini par rapport au repère capteur, l'expression de la mesure est donnée par la fonction de projection $\text{proj}(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$:

$$\text{proj}(\mathbf{x}) = \mathbf{x} \quad (2.39)$$

La matrice d'interaction entre l'évolution de la mesure du point 3D et l'évolution du repère du capteur (*i.e.* son torseur cinématique) est donnée par l'équation suivante :

$$\mathbf{L}_{\mathbf{x}} = \begin{bmatrix} -\mathbf{I}_3 & [\mathbf{x}]_{\times} \end{bmatrix} \quad (2.40)$$

Il est aussi possible d'utiliser des vecteurs poses (translation et orientation) en tant que caractéristiques à part entière. Celles-ci peuvent être obtenues à l'aide d'un estimateur de pose par caméra visuelle ou par alignement de nuages de points 3D. La mesure est composée du couple $(\mathbf{x}, \mathbf{y})$ composé respectivement de la translation et de l'orientation du repère mesuré. La matrice d'interaction pour une mesure de pose exprimée dans le repère capteur, où l'orientation est exprimée telle que $\mathbf{y} = \mathbf{u} \sin(\theta)$, est définie par la relation suivante :

$$\mathbf{L}_{(\mathbf{x}, \mathbf{y})} = \begin{bmatrix} -\mathbf{I}_3 & [\mathbf{x}]_{\times} \\ \mathbf{0}_3 & B(\mathbf{y}) \end{bmatrix} \quad (2.41)$$

avec

$$B(\mathbf{y}) = \cos(\theta) \mathbf{I}_3 - \frac{\sin(\theta)}{2} [\mathbf{u}]_{\times} + \frac{\cos(\theta) - 1}{2} [\mathbf{u}]_{\times}^2 \quad (2.42)$$

Un plan dans l'espace 3D peut servir de caractéristique dès lors qu'il est possible d'extraire son équation à partir du flux capteur. L'extraction se passe généralement comme suit : à partir d'une image 3D (*i.e.* un nuage de points) de la scène prise à l'aide d'une caméra à profondeur de champ par exemple, les points 3D sont regroupés par appartenance à une surface plane. La méthode d'extraction est généralement réalisée avec un algorithme de type RANSAC (*Random Sample Consensus*) qui permet d'évaluer si un certain groupe de points appartient à un plan d'équation $\mathcal{P}$ (*inliers*) et d'en écarter les autres (*outliers*). Soit un point 3D $\mathbf{x} = [x \ y \ z]^T$ quelconque dans l'espace, celui-ci appartient au plan $\mathcal{P} = [a \ b \ c \ d]^T$ si et seulement si :

$$ax + by + cz + d = 0 \quad (2.43)$$

Soit $\mathbf{n} = [a \ b \ c]^T$ le vecteur normal au plan. En tenant compte de la notation homogène du point, l'équation peut aussi s'exprimer sous la forme :

$$\begin{bmatrix} \mathbf{n}^T & d \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = \mathbf{n} \cdot \mathbf{x} + d = 0 \quad (2.44)$$

Connaissant un point $\mathbf{x}_0$ appartenant effectivement au plan, il est possible d'en déduire la coordonnée d de $\mathcal{P}$ telle que :

$$d = -\mathbf{n} \cdot \mathbf{x}_0 \quad (2.45)$$

En s'inspirant d'[Espiau and Rives \[1987\]](#), compte tenu d'une mesure d'un plan 3D définie par $\mathcal{P} = [\mathbf{n}^\top \ -\mathbf{n} \cdot \mathbf{x}_0]^\top$, extraite d'un nuage de points 3D pris d'un capteur, la matrice d'interaction associée au plan $\mathcal{P}$ est définie ci-dessous

$$\mathbf{L}_{\mathcal{P}} = \begin{bmatrix} \mathbf{0}_3 & [\mathbf{n}]_{\times} \\ \mathbf{n}^\top & \mathbf{0}_{1 \times 3} \end{bmatrix} \quad (2.46)$$

A partir des équations (2.45) et (2.46), il est possible de déduire la matrice d'interaction pour un vecteur. Soit une mesure d'un vecteur $\mathbf{n} = [a \ b \ c]^\top$ issue d'un flux capteur à trois dimensions, sa matrice d'interaction associée est définie par :

$$\mathbf{L}_{\mathbf{n}} = \begin{bmatrix} \mathbf{0}_3 & [\mathbf{n}]_{\times} \end{bmatrix} \quad (2.47)$$

Les droites peuvent être elles-aussi utilisées en tant que caractéristiques pour un asservissement visuel 3D. [Andreff et al. \[2002\]](#) modélisent une droite observée par un capteur avec les coordonnées de Plücker bi-normalisées. La mesure de la ligne est exprimée par $\mathcal{L} = [\mathbf{u}^\top \ \mathbf{h}^\top \ h]^\top$ où les composantes représentent respectivement le vecteur unitaire orienté le long de la droite en question, le vecteur unitaire perpendiculaire au plan formé par le vecteur $\mathbf{u}$ et l'origine du repère du capteur et h la distance normale de la droite jusqu'à l'origine du repère. La matrice d'interaction associée est définie par la relation suivante :

$$\mathbf{L}_{\mathcal{L}} = \begin{bmatrix} -\frac{1}{h}(\mathbf{u} \times \mathbf{h}) \cdot \mathbf{h} & -[\mathbf{h}]_{\times} \\ \mathbf{0}_3 & -[\mathbf{u}]_{\times} \\ (\mathbf{u} \times \mathbf{h})^\top & \mathbf{0}_{1 \times 3} \end{bmatrix} \quad (2.48)$$

De nombreuses autres caractéristiques de la famille de type 3D peuvent être utilisées. Par exemple, [Teuliere and Marchand \[2012\]](#) utilisent une image de disparité provenant d'une caméra à profondeur de champ de la marque *Microsoft Kinect*. [Pagès et al. \[2004, 2005\]](#) se basent sur la lecture de projections laser sur une surface avec une caméra pour se positionner par rapport à cette surface. Plus loin dans ce document est présentée une méthode de positionnement précise à l'aide d'informations de distance d'un point à un plan selon la normale au plan [[Vandernotte et al., 2014](#)].

Chacune de ces caractéristiques peut être sélectionnée en fonction de son comportement et combinée pour réaliser une commande référencée capteurs. Ainsi, il est possible d'asservir l'effecteur d'un robot manipulateur sur la base d'une mesure d'une ou plusieurs références de son environnement. Par exemple, l'utilisation d'un point 3D ne fixera que les trois degrés de liberté de translation de l'effecteur alors qu'une caractéristique de plan fixera un degré de translation et deux d'orientation. De ce point de vue, les caractéristiques 3D sont très similaires aux contraintes mécaniques utilisées (coïncidence de plan, de segment, etc.) dans les logiciels de conception assistée par ordinateur pour la réalisation d'assemblage d'ensembles virtuels.

2.1.4.2 Caractéristiques de type 2D

En opposition aux caractéristiques de type 3D, il est possible de classer celles mesurées à partir de capteurs 2D telles que des caméras visuelles. De nombreuses caractéristiques 2D ont été utilisées

pour réaliser une expérience d'asservissement visuel.

Une des premières caractéristiques 2D que l'on peut citer est le point 2D. Dès lors qu'un point dans une image peut être détecté et poursuivi d'une image à l'autre, on peut l'utiliser pour un asservissement visuel. Il existe plusieurs méthodes pour détecter un point saillant dans l'image : le plus répandu est le détecteur de Harris qui s'occupe de reconnaître les points en se basant sur l'information de contours. La poursuite d'un point est faisable en s'appuyant sur un descripteur tel que le descripteur SIFT (*Scale Invariant Feature Transform*) [Lowe, 1999], qui permet de générer un identifiant unique d'un point en fonction des valeurs des pixels (leur couleur) de son voisinage immédiat. D'autres descripteurs existent : SURF (*Speed Up Robust Feature*) [Bay et al., 2006, 2008] met l'accent sur la performance ; le descripteur BRIEF (*Binary Robust Independent Element Feature*) [Calonder et al., 2010] utilise une séquence binaire en tant que signature et ainsi gagne aussi en termes de performance.

Ci-après est décrit le modèle du sténopé : la projection d'un point 3D $\mathbf{x}$ exprimé dans le repère de la caméra sur le plan focal est définie par $\text{proj}_{3D \rightarrow 2D}(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ telle que :

$$\text{proj}_{3D \rightarrow 2D}(\mathbf{x}) = \frac{1}{z} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \mathbf{x}_n \quad (2.49)$$

Le résultat de cette fonction de projection est communément appelé coordonnées normalisées, notées $\mathbf{x}_n$. Le passage vers l'image utilise les paramètres intrinsèques de la caméra, celui-ci est défini par les équations suivantes :

$$u = u_0 + F_u x_n \quad ; \quad v = v_0 + F_v y_n \quad (2.50)$$

La notation homogène permet d'écrire cette relation sous forme matricielle :

$$\mathbf{x}_p = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} F_u & 0 & u_0 \\ 0 & F_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}_n = \mathbf{K} \mathbf{x}_n \quad (2.51)$$

On trouve dans ces équations (u, v) les coordonnées en pixels du point dans l'image, (F_u, F_v) les distances focales en pixel.m^{-1} et (u_0, v_0) le point principal de l'image qui représente l'intersection de l'axe optique de la caméra avec le plan focal. Ces paramètres peuvent être déduits en suivant une procédure de calibration. Au final, la fonction de projection complète d'un point 3D appartenant à un référentiel absolu $\mathbf{W}$ peut être résumé par la forme suivante :

$$\mathbf{x}_p = \mathbf{K} \text{proj}_{3D \rightarrow 2D} ({}^C\mathbf{T}_W^W \mathbf{x}) \quad (2.52)$$

La matrice d'interaction pour les coordonnées normalisées d'un point sur le plan focal est donnée par l'équation (2.37) et est rappelée ci-dessous :

$$\mathbf{L}_{\mathbf{x}_n} = \begin{bmatrix} -\frac{1}{z} & 0 & \frac{x_n}{z} & x_n y_n & -1 - x_n^2 & y_n \\ 0 & -\frac{1}{z} & \frac{y_n}{z} & 1 + y_n^2 & -x_n y_n & -x_n \end{bmatrix}$$

Il est tout à fait possible d'exprimer cette matrice d'interaction en considérant les coordonnées en pixel $\mathbf{x}_p$, en déterminant la jacobienne $\frac{\partial \mathbf{K} \text{proj}_{3D \rightarrow 2D}(\mathbf{x}_n)}{\partial \mathbf{x}_n}$ et en s'appuyant sur la matrice précédente.

D'autres caractéristiques géométriques extraites d'une séquence d'images ont fait l'objet d'un travail de modélisation pour les utiliser dans un asservissement visuel. Parmi elles, il est possible

de citer les moments dans l'image [Chaumette, 2004; Long et al., 2014b]. Les moments dans l'image représentent des moyennes pondérées, mesures quantitatives mathématiques, fonctions de l'intensité de pixels de l'image en question ou d'autres moments d'ordres inférieurs. En vision par ordinateur, ils servent à décrire les éléments segmentés en déterminant par exemple leur centre de gravité et/ou l'aire de l'objet.

L'homographie a aussi fait l'objet d'une telle étude [Benhimane and Malis, 2006]. L'homographie est une relation mathématique entre deux sets de points 2D observant la même surface plane. Cette relation est utile pour de nombreuses tâches en vision par ordinateur : redressement d'image, superposition de deux prises de vue successives ou encore estimation du mouvement de la caméra entre deux points de vue.

D'autres équipes ont affiché le souhait de s'affranchir d'une opération d'extraction des caractéristiques. En effet, certaines méthodes peuvent être coûteuses en termes de temps et difficilement applicables avec un asservissement temps réel. Pour cela, une expérience d'asservissement visuel basée sur la simple intensité des pixels de l'image a été développée, appelée asservissement visuel direct [Silveira and Malis, 2012; Silveira et al., 2013; Silveira, 2014]. L'approche a été revisitée en utilisant d'autres sources capteurs comme une image de disparité [Teuliere and Marchand, 2012] provenant d'une caméra à profondeur de champ ou encore un sonar à ultrason [Nadeau and Krupa, 2011; Nadeau et al., 2015].

Certains travaux ont cherché à améliorer les performances de l'asservissement visuel 2D en mixant des caractéristiques 2D et 3D [Malis and Chaumette, 2000]. L'ajout de l'information de profondeur de la scène par rapport au repère de la caméra aux caractéristiques a permis au système d'atteindre des performances équivalentes aux caractéristiques 3D [Cervera and Martinet, 1999]. De même, Cervera et al. [2002] ont proposé une amélioration de l'asservissement visuel 2D pour obtenir un meilleur comportement de l'effecteur dans l'espace cartésien [Cervera et al., 2002, 2003].

2.1.4.3 Assemblage

Il existe de nombreux outils pour mettre en place une expérience d'asservissement référencé capteur, et ainsi être capable de contrôler la position de l'effecteur d'un robot en tenant compte d'objets ou de points remarquables dans le monde réel. Il a déjà été annoncé auparavant comment se construisait une commande référencée capteurs. Pour rappel, il faut réunir quatre paramètres (cf. paragraphe 2.1.3) : choisir la caractéristique selon la nature du capteur utilisé, établir sa matrice d'interaction, calibrer la liaison entre le repère capteur et le repère mobile du robot et déterminer les valeurs désirées à atteindre dans l'espace des mesures du capteur. Comme le montre les paragraphes précédents, les deux premiers points sont déjà largement couverts et la communauté propose une palette de caractéristiques sur étagère à choisir en fonction de ses ressources et ses besoins.

Néanmoins, une tâche peut entraîner un comportement de l'effecteur différent en fonction des caractéristiques utilisées pendant la phase de convergence du système. Par exemple, un asservissement visuel 2D en utilisant quatre points dans l'image produit, pendant la convergence vers sa position d'équilibre, une trajectoire de l'effecteur qui n'est pas une ligne droite dans l'espace cartésien. La forme générale de la trajectoire n'est pas prévisible. Si la distance à couvrir est importante, le mouvement engendré peut entraîner la sortie des points d'intérêt en dehors du champ de vision de la caméra, ce qui constitue une défaillance grave dans le processus de contrôle. Au contraire, un asservissement utilisant une mesure de pose induit un mouvement en ligne droite dans l'espace cartésien.

L'obtention des valeurs à atteindre dans l'espace capteur, le dernier des quatre éléments, est un point non négligeable mais trop souvent négligé dans les travaux de la communauté. Même si de nombreuses caractéristiques pour la commande référencée capteurs sont disponibles pour contrôler le robot, certaines se révèlent peu commodes à utiliser sur des problèmes robotiques réels. Notamment car les caractéristiques à l'état désiré sont difficiles à prédire. Par exemple, la méthode proposée par [Teulière and Marchand \[2012\]](#) repose sur une image de disparité issue du capteur *Microsoft Kinect*. L'image désirée est obtenue par une prise de vue du capteur dans la bonne position. Compte tenu de la difficulté à modéliser fidèlement une scène, de la difficulté à déterminer et simuler la fonction de projection du capteur (dont une bonne partie a été estimée sur la base d'un travail de rétro-ingénierie par la communauté du logiciel libre), l'image désirée est quasi impossible à prédire. Si la scène évolue, l'opérateur aura la charge de prendre une nouvelle prise de vue. De fait, il est assez difficile d'imaginer l'utilisation d'une telle méthode pour réaliser une opération d'assemblage alors que la scène est constamment modifiée.

Pour la réalisation d'une opération d'assemblage par un bras robotique, il est possible de tirer parti des outils de conception assisté par ordinateur (CAO) utilisés lors des phases de conception de produits manufacturés. En effet, les équipes de bureaux d'études ont la capacité de réaliser l'assemblage d'un ensemble complexe virtuellement afin de prévenir très en amont les défauts de conception et ce, pour éviter un surcoût éventuel pendant la phase de fabrication du produit. Dès lors, la maquette numérique possède déjà l'ensemble des informations nécessaires pour établir les valeurs à l'équilibre pour des caractéristiques de type lignes, plans ou encore points à trois dimensions. De plus, les logiciels CAO proposent une palette de contraintes d'assemblage pour fixer les pièces entre elles qui indiquent, d'un point de vue commande référencée capteurs, la nature des caractéristiques à choisir et leurs valeurs à l'équilibre. Les caractéristiques de type 3D sont donc les meilleures candidates à la réalisation d'une opération d'assemblage en utilisant une commande référencée capteurs.

2.2 Commande référencée multi-capteurs et multi-tâches

Les concepts présentés dans la section précédente peuvent être étendus aux tâches qui nécessitent plusieurs capteurs et/ou plusieurs sous-tâches organisées selon un mode particulier.

Cette section fait une courte présentation de l'état de l'art sur l'asservissement en tenant compte d'informations provenant de plusieurs sources capteurs. La présentation d'outils pour la mise en place d'une ou plusieurs tâches exécutées en même temps selon un ordre de priorité sera également présentée.

2.2.1 Sources multiples

Pour rappel, la commande référencée capteur s'appuie sur la régulation à zéro d'une erreur entre une quantité mesurée s par rapport à une valeur à l'état d'équilibre s^d , où la forme de s dépend de la nature de la caractéristique choisie. Comme le montre le paragraphe 2.1.3, une tâche peut être construite par empilement de plusieurs caractéristiques provenant du même capteur.

On considère maintenant un système composé de k capteurs où chaque capteur S_i observe une ou plusieurs caractéristiques dont les mesures sont rassemblées sous le vecteur s_i de dimension m . Chacun de ses capteurs possède sa propre matrice de transformation cinématique établissant la relation entre leurs torseurs cinématiques respectifs et celui de l'effecteur, notées ${}^{S_i}\mathbb{T}_N$. L'équation

(2.4) se réécrit sous la forme suivante.

$$\dot{\mathbf{s}} = \mathbf{L}^T \mathbf{J} \dot{\mathbf{q}} = \begin{bmatrix} \mathbf{L}_{s,1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{L}_{s,k} \end{bmatrix} \begin{bmatrix} \mathbf{S}_i^T \mathbb{T}_N \\ \vdots \\ \mathbf{S}_k^T \mathbb{T}_N \end{bmatrix} \mathbf{J} \dot{\mathbf{q}} \quad (2.53)$$

$\mathbf{L} \in \mathbb{R}^{m \times 6k}$ et $\mathbb{T} \in \mathbb{R}^{k \times 6}$ regroupent, respectivement, les matrices d'interaction associées aux caractéristiques du $i^{\text{ème}}$ capteur et les matrices de transformations cinématiques entre le repère du $i^{\text{ème}}$ capteur et le repère de l'effecteur N. Cette tâche peut être utilisée au sein de la loi de commande définie par l'équation (2.33). De la même manière, l'application de la loi de commande avec k capteurs permet aussi une décroissance exponentielle de l'ensemble des composantes de l'erreur $\mathbf{e} = (\mathbf{s} - \mathbf{s}^d)$.

2.2.2 Tâches multiples

Il existe d'autres manières de prendre en compte plusieurs sources de capteurs au sein d'une commande référencée capteur. Parmi elles, on peut citer la commande partitionnée, les approches hiérarchiques, la commande commutative et les méthodes hybrides. Ces méthodes se concentrent plus sur l'aspect multi-tâches et l'ordonnancement entre elles. Néanmoins, les tâches peuvent être définies sur des mesures provenant de plusieurs sources de capteurs.

Pour faciliter la compréhension de cette partie, on définit la vitesse de variation désirée $\dot{\mathbf{e}}^*$ d'une tâche telle que $\dot{\mathbf{e}}^* = -\lambda \mathbf{e} = -\lambda (\mathbf{s} - \mathbf{s}^d)$ [Kermorgant, 2011], aussi nommée comportement de référence par Mansard [2006]. De même, on pose $\mathbb{J}$ comme étant la jacobienne d'une tâche telle que $\mathbb{J} = \mathbf{L}_s^T \mathbb{T}_N \mathbf{J}$.

2.2.2.1 Commande partitionnée

La commande partitionnée permet d'appliquer deux tâches différentes, basées sur des capteurs différents ou non, simultanément sur un même système robotique. En considérant deux tâches, on suppose ici que ces deux tâches contrôlent chacune une partie des degrés de liberté du robot. La commande s'écrit alors :

$$\dot{\mathbf{q}} = \mathbb{J}_1^+ \dot{\mathbf{e}}_1^* + \mathbb{J}_2^+ \dot{\mathbf{e}}_2^* \quad (2.54)$$

sous la condition que $\mathbb{J}_1^+ \mathbb{J}_2 = \mathbb{J}_2^+ \mathbb{J}_1 = 0$, condition qui exprime l'indépendance des tâches. Cette méthode est utilisée notamment pour appliquer des commandes parallèles de force et de position sur un système [Khalil and Dombre, 2002; Prats, 2009]. Dans [Nadeau and Krupa, 2011] où le robot porte une sonde échographique, les auteurs contrôlent cinq des six degrés de liberté de l'effecteur grâce aux mesures de la sonde tandis que l'axe de translation en direction du patient est contrôlé par une mesure d'effort.

L'utilisation de la loi de commande partitionnée est intrinsèquement liée à l'application finale. Elle nécessite en effet de déterminer par avance quelle mesure contrôlera quel degré de liberté de l'effecteur ou quelle articulation. Les couplages ou conflits entre les tâches sont tout simplement ignorés.

La prise en compte d'informations issues de capteurs de force et de couple sera traitée dans le chapitre suivant.

2.2.2.2 Commande hiérarchique

La commande hiérarchique permet d'imposer un ordre de priorité entre l'application de telle ou telle tâche. Ces approches se basent sur la méthode du gradient projeté [Siciliano and Slotine, 1991; Samson et al., 1991]. Les conditions nécessaires pour la mise en œuvre d'une commande hiérarchique requiert que les tâches ne commandent pas l'ensemble du système ($m < n$). Deux cas de figure peuvent alors se présenter : soit le robot est redondant, c'est-à-dire qu'il possède plus d'articulations que nécessaire pour imposer un mouvement à l'effecteur ($n > 6$); ou alors la tâche choisie ne contrôle pas l'ensemble des degrés de liberté du repère à contrôler ($m < 6$). On parle alors de redondance.

La méthode en question permet de calculer le sous-espace des mouvements disponibles laissés par la tâche de priorité supérieure. Le but des commandes hiérarchiques est donc d'exploiter ces sous-espaces pour contrôler les parties du système qui ne le sont pas, ce qui entraîne une certaine flexibilité dans la définition de la stratégie de contrôle des robots en général. Le lecteur pourra trouver un travail complet sur l'ensemble des commandes hiérarchique dans [Marey, 2010; Abou Moughlbay, 2013].

L'expression la plus simple et la plus répandue de la commande hiérarchique, en considérant une tâche principale qui ne contrôle pas l'ensemble du système, est telle que [Chiaverini et al., 2008] :

$$\dot{\mathbf{q}} = \mathbb{J}^+ \dot{\mathbf{e}}^* + (\mathbf{I} - \mathbb{J}^+ \mathbb{J}) \mathbf{z}_{\dot{\mathbf{q}}'} \quad (2.55)$$

où le composant $(\mathbf{I} - \mathbb{J}^+ \mathbb{J})$ représente l'opérateur de projection sur le noyau de la jacobienne de tâche de la tâche principale, noté $\mathbf{P}$. En utilisant cette projection, les composantes de vitesses articulaires $\mathbf{z}_{\dot{\mathbf{q}}'}$ ne seront ajoutées au vecteur des vitesses articulaires que si la matrice $\mathbf{P}$ le permet. Le vecteur $\mathbf{z}_{\dot{\mathbf{q}}'}$ peut être choisi de manière arbitraire et représente le plus souvent la contribution d'une tâche annexe secondaire telle que $\mathbf{z}_{\dot{\mathbf{q}}'} = \mathbb{J}_2^+ \dot{\mathbf{e}}_2^*$.

La généralisation à trois ou k tâches ne peut pas être réalisée avec l'équation (2.55). Par exemple, la commande pour 3 tâches prend la forme suivante :

$$\dot{\mathbf{q}} = \mathbb{J}_1^+ \dot{\mathbf{e}}_1^* + \mathbf{P}_1 (\mathbb{J}_2^+ \dot{\mathbf{e}}_2^* + \mathbf{P}_2 \mathbf{z}_{\dot{\mathbf{q}}_3}) = \mathbb{J}_1^+ \dot{\mathbf{e}}_1^* + \mathbf{P}_1 \mathbb{J}_2^+ \dot{\mathbf{e}}_2^* + \mathbf{P}_1 \mathbf{P}_2 \mathbf{z}_{\dot{\mathbf{q}}_3} \quad (2.56)$$

Les opérateurs de projection sur le noyau ne sont pas commutatifs. La première tâche ne sera pas perturbée par la seconde et la troisième. Or, il n'y a pas de garantie que la seconde ne sera pas perturbée par la troisième. En effet, le produit $\mathbf{P}_1 \mathbf{P}_2$ ne correspond pas au projecteur sur le noyau de la seconde tâche. De fait l'ordre des priorités n'est pas respecté en totalité [Antonelli, 2009].

Une amélioration de cette approche, pour éviter le problème mentionné ci-dessus, est de prendre en compte l'action des composantes de la seconde tâche moins prioritaire. Par exemple, pour deux tâches ([Chiaverini et al., 2008, équation 11.4.3]) :

$$\dot{\mathbf{q}} = \mathbb{J}_1^+ \dot{\mathbf{e}}_1^* + (\mathbb{J}_2 \mathbf{P})^+ (\dot{\mathbf{e}}_2^* - \mathbb{J}_2 \mathbb{J}_1^+ \dot{\mathbf{q}}_1) \quad (2.57)$$

Le concept peut être généralisé à k tâches et peut être formulé selon une équation récursive. Cette équation récursive est nommée le plus souvent : le schéma de séquençement de tâches efficient (*efficient task sequencing scheme*), initialement formulé par Siciliano and Slotine [1991].

$$\begin{aligned} \dot{\mathbf{q}}_i &= \dot{\mathbf{q}}_{i-1} + (\mathbb{J}_i \mathbb{P}_{i-1})^+ (\dot{\mathbf{e}}_i^* - \mathbb{J}_i \dot{\mathbf{q}}_{i-1}) \\ \mathbb{P}_i &= \mathbb{P}_{i-1} + (\mathbb{J}_i \mathbb{P}_{i-1})^+ (\mathbb{J}_i \mathbb{P}_{i-1}) \end{aligned} \quad (2.58)$$

pour $i = 1, \dots, k$. $\mathbb{P}_i$ représente le projecteur orthogonal sur le noyau de la matrice jacobienne augmentée $[\mathbb{J}_1 \dots \mathbb{J}_i]^T$. Les conditions initiales sont $\dot{\mathbf{q}}_1 = \mathbb{J}_1^+ \dot{\mathbf{e}}_1^*$ et $\mathbb{P}_1 = \mathbf{P} = (\mathbf{I} - \mathbb{J}_1^+ \mathbb{J}_1)$.

La méthode du gradient projeté est largement utilisée dans la communauté et plus particulièrement lorsque le système dispose d'un grand nombre d'articulations comme les robots humanoïdes. [Baerlocher and Boulic \[2004\]](#) utilisent cette méthode pour déterminer le mouvement d'un robot virtuel avec un grand nombre de contraintes hiérarchisées. [Park \[2006\]](#) et [Sentis \[2007\]](#) appliquent le projecteur pour ordonner les comportements d'un robot humanoïde et adresser le problème de commande en deux sous problèmes qui se focalisent respectivement sur la posture du buste du robot et les effecteurs de celui-ci (mains, tête et pieds). [Sentis \[2007\]](#) intègre aussi au sein de la loi de commande les différents points de contact pour l'équilibre dynamique du robot. Dans [\[Mansard et al., 2009a,b; Kanoun et al., 2011\]](#), des contraintes d'inégalité, dites unilatérales, sont formalisées au sein de ce cadre. [Mansard et al. \[2009b\]](#) proposent un logiciel baptisé *Pile de tâches* (en anglais, *Stack of tasks*) qui se base de manière systématique sur cette méthode et proposent un nouvel opérateur d'inversion baptisé *inverse continue* destiné à remplacer la méthode pseudo-inverse. [Kanoun et al. \[2011\]](#) proposent une implémentation différente qui ne se base pas sur le calcul de la jacobienne. En effet, les auteurs se basent sur la définition d'un problème d'optimisation où le but affiché est la hiérarchisation entre les tâches.

2.2.2.3 Commande commutative

La commande commutative applique une stratégie différente de la commande hiérarchisée présentée ci-dessus. En effet tandis que le projecteur orthogonal permet de calculer automatiquement la disponibilité des degrés de liberté du système pour appliquer plusieurs tâches en même temps, la commande commutative permet de passer d'une tâche à une autre de manière atomique, sous condition. Pour deux tâches avec une condition de passage, la forme de la commande s'écrit comme suit :

$$\dot{\mathbf{q}} = \begin{cases} \mathbb{J}_1^+ \dot{\mathbf{e}}_1^* & \text{si la condition est vérifiée,} \\ \mathbb{J}_2^+ \dot{\mathbf{e}}_2^* & \text{sinon} \end{cases} \quad (2.59)$$

La condition est spécifiée par l'utilisateur. Cette loi de commande entraîne un phénomène d'hystérésis où le système oscille entre les deux positions d'équilibre des deux tâches respectives.

[Chesi et al. \[2004\]](#); [Gans and Hutchinson \[2007\]](#) proposent une loi de commande basée à la fois sur une tâche d'asservissement visuel 2D basé image (*IBVS*) et une autre 3D basé position (*PBVS*). Le premier applique un asservissement basé position mais celui-ci n'est appliqué que partiellement tant que l'objet reste dans le champ de vision de la caméra. Le second applique les deux tâches tour à tour selon si l'une d'entre elles observe une erreur trop grande. Ainsi, le système garantit une erreur minimale dans les deux tâches. Ce type de commande est applicable aussi aux robots mobiles. De la même manière que dans [\[Chesi et al., 2004\]](#), [Folio and Cadenat \[2008\]](#) utilisent une commande commutative pour appréhender la perte de visibilité d'un objet : si celui-ci n'est plus visible, l'asservissement visuel du robot mobile est partiellement appliqué et prédit l'évolution des caractéristiques observées pour retrouver la visibilité de l'objet.

Outre le fait que la commande commutative est sensible aux phénomènes d'hystérésis, la définition des conditions pour se permettre de passer d'une tâche à l'autre de manière propre peut être un exercice difficile. Une condition mal définie ou qui ne correspond pas avec la situation réelle du système à contrôler peut entraîner de fortes oscillations entre les deux points d'équilibre des tâches considérées.

2.2.2.4 Commandes pondérées

Les commandes pondérées représentent un mélange entre la commande hiérarchique et la commande commutative. En effet, les différentes tâches sont appliquées en même temps tenant compte d'une pondération particulière, qui permet d'appliquer tout ou partiellement telle ou telle tâche. Ce type de commande peut prendre plusieurs formes. De nombreux travaux se basent sur cette approche pour mettre en place un objectif de commande à plusieurs tâches [Nakamura et al. \[1987\]](#); [Nakamura \[1990\]](#); [Chan and Dubey \[1995\]](#); [De Luca et al. \[2006\]](#); [Hafez and Jawahar \[2007\]](#); [Kermorgant \[2011\]](#).

Nous retiendrons la formule basée sur la jacobienne de tâche étendue où l'ensemble des tâches sont empilées pour n'en former qu'une seule. La différence se situe au niveau de l'opérateur d'inversion qui comprend une matrice définie positive $\mathbf{H}$ comportant les poids associés à chacune des composantes de chacune des tâches. La commande a la forme suivante :

$$\dot{\mathbf{q}} = (\mathbf{H}\mathbb{J})^+ \mathbf{H}\dot{\mathbf{e}}^* \quad (2.60)$$

La matrice de pondération $\mathbf{H}$ peut être construite de plusieurs manières. Sa définition la plus simple est d'établir une matrice diagonale qui, pour chaque terme, supporte un poids particulier variable selon le besoin. Par exemple,

$$\mathbf{H} = \begin{bmatrix} h_1 \mathbf{I}_{m_1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & h_k \mathbf{I}_{m_k} \end{bmatrix} \quad (2.61)$$

où $\sum_i^k h_i = 1$. Même si la forme de la loi de commande n'est pas tout à fait identique entre les différents travaux cités ci-dessus, la contribution de chacune des tâches est réglée par le poids associé à chacune d'elles. L'esprit reste le même, même si la pondération est appliquée sur les valeurs articulaires comme dans [\[Chan and Dubey, 1995\]](#) ou sur les composantes s des tâches capteurs considérées à l'image de [Kermorgant \[2011\]](#).

[Kermorgant \[2011\]](#) se base sur l'équation (2.60) et propose un schéma de commande générique pour la prise en compte de n'importe quelles mesures capteurs au sein d'un asservissement visuel. La modulation des poids est gérée par une méthode *ad-hoc* où les poids varient de 0 à $+\infty$ si la tâche doit ne pas être du tout prise en compte ou à l'inverse, doit avoir une priorité maximale selon la situation du système.

L'opérateur d'inversion comme proposé dans l'équation (2.60) peut parfois entraîner des discontinuités dans la loi de commande, en particulier si la modulation des poids associés aux tâches n'est pas rigoureuse. [Mansard et al. \[2009a\]](#) proposent un cadre comprenant un nouvel opérateur d'inversion baptisé *inverse continue* qui permet de s'affranchir de ces problèmes de discontinuité. Les auteurs utilisent cet opérateur entre autres pour spécifier de nouvelles tâches dites contraintes unilatérales où le poids associé à la contrainte est nul si celle-ci n'est pas activée. Cette approche permet la construction d'un schéma de commande générique basé sur le formalisme de la jacobienne de tâche, capable de gérer à la fois des contraintes bilatérales (par exemple, asservissement de l'effecteur par rapport à une référence locale) et unilatérales (par exemple, évitement de butées articulaires).

La commande pondérée où les poids peuvent prendre une valeur nulle est considérée parfois comme une classe de schéma de commande à part entière nommée *tâches à dimensions variables*.

2.3 Schéma de commande générique référencé multi-capteurs

Les sections précédentes ont montré l'étendue des travaux scientifiques réalisés autour de la commande de bras manipulateurs et autres robots, en utilisant des capteurs extéroceptifs tels que des caméras ou tout autre type de capteurs. L'état de l'art s'est attardé sur comment était construit une tâche pour une commande référencée capteur ainsi que sur la fusion de plusieurs commandes entre elles tenant compte d'un ordonnancement entre les tâches, définies selon les besoins de l'utilisateur.

Pour rappel, la section 1.5 a posé le souhait de généraliser le recours à ces outils pour proposer un nouveau cadre pour la programmation robotique en tenant compte de l'environnement dans lequel il évolue. Le but recherché ici est de rendre le bras manipulateur contrôlable à partir de références appartenant à son environnement réel et ainsi le rendre intrinsèquement plus précis. Plus largement, le cadre permet de lui donner la capacité de réaliser de nouvelles opérations haut-niveau qui n'étaient pas réalisées jusqu'alors telles que des opérations d'assemblage, ou des opérations de saisie et de dépose (*pick and place*) dans un environnement non connu *a priori*.

Cette section pose les bases du cadre de programmation robotique énoncé dans la section 1.5. Le concept se base sur le fait qu'une mesure provenant d'un capteur peut contraindre un ou plusieurs degrés de liberté de l'effecteur d'un bras manipulateur. La finalité du cadre est d'automatiser la construction d'une loi de commande référencée capteurs tenant compte à la fois d'informations issues de modèles virtuels numériques ainsi que de l'équipement du système robotique (capteurs, préhenseur, etc.), afin de garantir que la position attendue de la pièce par rapport à son contexte réel (par rapport à une structure) corresponde à celle attendue dans la maquette numérique.

La présentation se veut générique car le cadre a la prétention de s'appliquer sur n'importe quel système robotique capable de réaliser des opérations de type saisie et dépose ou assemblage. Nous verrons comment les dites caractéristiques seront choisies et sélectionnées pour construire des tâches d'asservissement de l'effecteur en utilisant les principes de la commande référencée capteur, multi-capteurs et multi-tâches. Au delà de la définition des tâches, nous aborderons aussi l'avantage d'utiliser un tel cadre pour la remontée d'informations concernant l'opération qui est effectuée par le manipulateur.

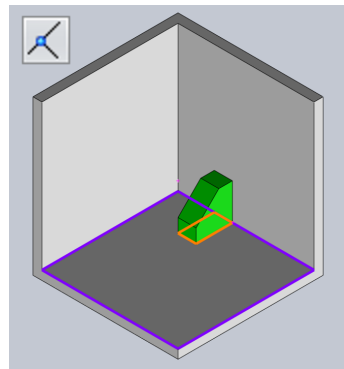
2.3.1 Choix des caractéristiques et définition de la relation spatiale

La section 2.1.4.3 établit une première discussion sur quel type de caractéristiques pourrait être utilisé dans le cadre d'un assemblage. Cette section se veut plus précise quant à la manière de les choisir et de les intégrer au processus.

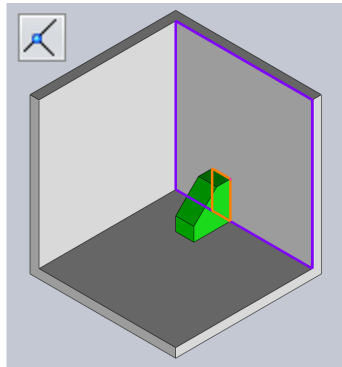
La première étape est donc de choisir les caractéristiques candidates pour la création d'une tâche robotique référencée capteur, dans le modèle virtuel représentant les différents composants d'un assemblage. L'assemblage est composé entre autres de la pièce à positionner ainsi que de la structure fixe qui supportera la pièce après fixation. La maquette numérique y décrit la relation de positionnement relatif entre la pièce et la structure à travers la spécification d'une ou plusieurs contraintes géométriques qui permettent de fixer tout ou partie des degrés de liberté de la pièce (cf. figure 2.3).

On définit $\mathcal{R}$ comme étant un ensemble de relations spatiales entre deux objets quelconques. Dans la maquette numérique, comme exprimé plus haut, la relation $\mathcal{R}$ est définie telle que

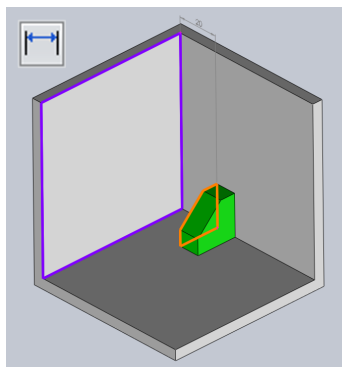
$$\mathcal{R} = \{ \mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k \} \quad (2.62)$$



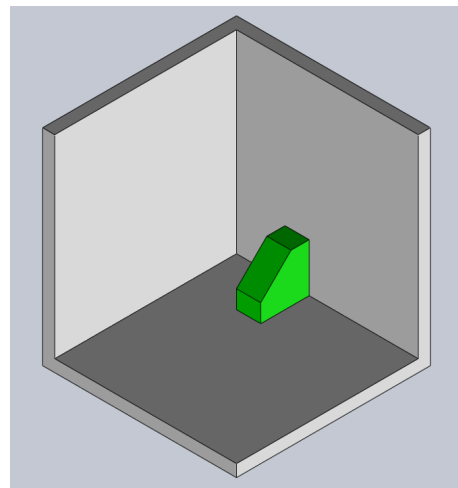
(a) C1 : plans coïncidents



(b) C2 : plans coïncidents



(c) C3 : distance entre plans



(d) Pièce totalement contrainte par rapport à la structure

FIGURE 2.3 – Exemple d'assemblage dans une maquette numérique : à gauche sont représentées les contraintes géométriques de la pièce. L'ensemble des contraintes permettent de fixer totalement les degrés de liberté de la pièce dans le repère de la structure.

où la relation spatiale $\mathcal{C}_i$ – aussi nommée contrainte – prend la forme :

$$\mathcal{C}_i = \{ M_i, N_i \mid f_i(M_i, N_i) \} \quad (2.63)$$

M_i et N_i sont des éléments géométriques élémentaires du modèle volumique qui appartiennent respectivement à la pièce et à la structure. Les logiciels CAO utilisent le modèle polyédrique pour la représentation d’objet volumique en trois dimensions. Les objets sont formés par l’assemblage de faces planes délimitées par des polygones [Benchmann and Bernard, 2007]. En se basant sur cette représentation, on définit les éléments géométriques élémentaires comme étant des plans (qui sont le pendant des faces du modèle polyédrique), des arêtes et des points, qui composent les polygones. f_i est la fonction décrivant la relation qui, dans le cas d’un logiciel CAO se restreint à quelques éléments dont une liste non exhaustive est présentée ci-dessous :

- Coïncidence : les deux éléments sont confondus, superposés après application de la contrainte.
- Parallèle : les deux éléments seront placés parallèlement entre eux, après application de la contrainte
- Perpendiculaire : idem, mais placés perpendiculairement.
- Tangente : les deux éléments seront tangent.
- Coaxial : les deux éléments comportant des axes de révolution verront ces axes superposés.
- Etc.

Le nombre k de contraintes $\mathcal{C}$ n’est pas limité pour décrire la relation. Néanmoins, il est usuel de d’arrêter d’ajouter des contraintes dès lors que tous les degrés de liberté de la pièce à positionner sont totalement fixés. On note que Chaumette [1990] propose des motifs 2D qui permettent de créer ces liaisons virtuelles avec un asservissement visuel. Notre but ici est d’appliquer un comportement image d’une liaison à des mesures image d’une structure complexe à travers un asservissement visuel ou référencé capteurs.

Le but maintenant est de sélectionner plusieurs relations qui vont réaliser la tâche. Ces relations candidates devront être formulées sous forme de tâche référencée capteur. Deux méthodes sont présentées ci-après pour effectuer la sélection.

La première méthode est un listage complet de l’ensemble des relations spatiales possibles entre toutes les composantes géométriques élémentaires de chacune des parties (pièce et structure). Dans un premier temps, il est nécessaire d’identifier tous les éléments géométriques de chacune des parties. Il est possible de se baser sur la représentation en modèle polyédrique pour faciliter la recherche. Le principe est illustré en figure 2.4. Les modèles constituent un ensemble de plans, d’arêtes et de points qui vont être associés en tenant compte d’une description relationnelle dans la prochaine étape. Les éléments sont agencés de manière topologique, ce qui permet de déterminer une hiérarchie dans les éléments.

La définition des relations spatiales est effectuée de manière systématique sur l’ensemble généré par la combinaison des éléments de chacune des parties. Par un critère de distance dans la maquette numérique assemblée, il est possible de déterminer si les deux objets sont coïncidents, parallèles, etc. Ceci constitue une liste tenant compte de beaucoup de contraintes qui, pour certaines, sont en conflit direct entre elles.

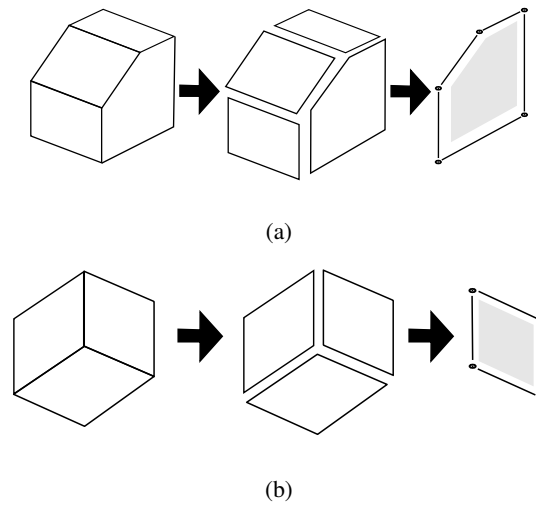


FIGURE 2.4 – Explosion des modèles polyédriques pour déterminer les composants géométriques élémentaires utilisés pour la sélection des contraintes candidates pour la construction d’une tâche.

La seconde étape logique est donc d’écarter les contraintes qui n’ont pas de signification physique. En effet, toutes les fonctions relationnelles ne peuvent définir la relation entre toutes les associations. Par exemple, il est absurde d’établir qu’un point soit parallèle à un plan. Si aucune fonction f_i ne peut exister pour décrire la relation entre les deux éléments, la combinaison est écartée et ne sera pas candidate.

La dernière étape est d’ordonner les différentes contraintes en tenant compte du nombre de degrés de liberté qu’elles fixent, en se basant sur l’ordonnancement des différents éléments.

Une seconde méthode – moins systématique – se base directement sur la lecture des contraintes qui ont été spécifiées pour la réalisation de l’assemblage au sein du logiciel CAO. L’avantage direct par rapport à la première méthode est que la recherche topologique sur l’ensemble des combinaisons possibles n’est plus nécessaire. L’utilisateur du logiciel CAO a déjà effectué ce travail.

L’inconvénient tout aussi direct est que même si l’utilisateur a pu mettre en place une contrainte dans le monde virtuel, celle-ci ne peut potentiellement être utilisée dans un schéma de commande référencée capteurs. Par exemple, celui-ci souhaite créer une contrainte parallèle entre deux plans mais l’un de ces plans n’est pas visible physiquement pour le robot en situation réelle.

Un autre inconvénient, d’ordre pratique, est qu’il est nécessaire d’aller chercher l’information au sein d’un logiciel CAO. Généralement, les logiciels sont des logiciels commerciaux dont les modifications et/ou l’extraction de données pour une utilisation externe est compliquée, voire impossible selon les stratégies commerciales des fournisseurs.

Ces deux méthodes de sélection, par le parcours de l’ensemble des combinaisons ou en s’appuyant sur les données de la maquette numérique, permettent de créer une liste ordonnée composée d’un certain nombre de contraintes qui peuvent potentiellement être transformées en tâche pour la construction d’une commande référencée capteurs.

2.3.2 Correspondance expectations et observations

Comme énoncé précédemment, la nature de la caractéristique dépend fortement de la nature du capteur utilisé. Néanmoins, l'équipement du robot n'apparaît pas dans l'assemblage virtuel. Le but de cette étape est de faire correspondre des mesures capteurs aux caractéristiques de la structure identifiées à partir de son modèle, tenant compte de la nature de ce capteur.

On note $\mathcal{Z}_j$ un ensemble qui regroupe des mesures de caractéristiques extraites à partir d'un traitement du signal propre au capteur j .

$$\mathcal{Z}_j = \{Z_{j,1}, \dots, Z_{j,l}\} \quad (2.64)$$

On suppose ici que la mesure $Z_{j,i}$ est l'image d'une caractéristique géométrique qui correspond potentiellement à une composante géométrique élémentaire de la structure M_i identifiée à travers la lecture du modèle volumique virtuel.

Établir la correspondance $(M_i, Z_{j,i})$ repose sur un critère d'association entre la mesure estimée de la caractéristique de la structure M_i tenant compte de la fonction de projection du capteur j et la mesure effective.

Il est utile de rappeler que M_i est une caractéristique appartenant à la structure quand la pièce est dans sa position désirée. Cela entraîne que la mesure estimée $Z'_{j,i}$ n'est autre que le vecteur d'état désiré s^d d'une tâche référencée capteurs.

L'obtention de la mesure estimée $Z'_{j,i} = \text{proj}(M_i)$ de la caractéristique repose sur plusieurs hypothèses. D'une part, il est nécessaire de connaître la pose du capteur par rapport à la structure, quand le système est dans sa position d'équilibre, *i.e.* à la fin du mouvement demandé. Dès lors que nous souhaitons positionner un objet ou autre partie de l'effecteur par rapport à une structure fixe, il est facile de l'intégrer à la maquette numérique et ainsi déterminer quelle serait la pose du capteur.

D'autre part, il est nécessaire que les caractéristiques de la structure dans les combinaisons candidates puissent être vues par le capteur. En effet, une caméra monoculaire simple ne peut donner une mesure d'un plan seulement en utilisant les méthodes classiques de traitement d'images.

Le critère d'association repose quant à lui sur une hypothèse de visibilité. En effet, si la caractéristique de la structure n'est pas mesurable car celle-ci n'est pas dans le champ de vision du capteur, la construction de la tâche ne pourrait se faire. Un avantage est que cette hypothèse peut être évaluée dès l'obtention de la mesure de la caractéristique dans l'état d'équilibre.

Le critère peut être un simple critère de distance entre les mesures à l'état d'équilibre et courantes. Outre sa simplicité, cette méthode apporterait aussi une condition nécessaire sur la position initiale du système avant l'application de la commande référencée capteurs : en position initiale, l'effecteur doit être proche de sa position finale. C'est ici bien entendu une limitation majeure de l'approche proposée. Il est aussi possible d'envisager un autre critère topologique basé sur l'agencement des différentes caractéristiques dans l'espace capteur. Cette approche pourrait ainsi éviter les faux-positifs.

La figure 2.5 montre un exemple d'association entre les mesures courantes des caractéristiques de la structure extraites du flux d'images et celles estimées à partir des informations du modèle virtuel quand le système est dans son état d'équilibre. La scène 3D montre le système effecteur-capteur-préhenseur-pièce et la structure. Avant l'application de la commande référencée capteur, le système se présente initialement dans une position quelconque, représenté en rouge (2.5a). Les

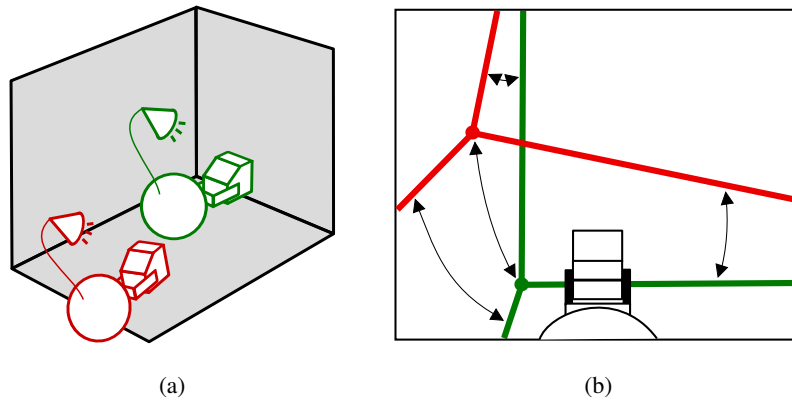


FIGURE 2.5 – Exemple du mécanisme d'association de mesures : (a) scène 3D avec la structure et l'ensemble effecteur-caméra-pièce représenté en position initiale (en rouge) et désirée (en vert); (b) point de vue caméra avec les caractéristiques de la structure : en rouge mesurées en position initiale, en vert estimées en position finale. Sont représentées par des flèches les associations possibles entre les mesures courantes et celles estimées.

caractéristiques extraites sont représentées en rouge dans l'image (2.5b). La maquette numérique et l'ensemble des informations dites «modèles» obtenues a priori permettent de calculer la pose du système dans la position désirée, représenté en vert. La fonction de projection du capteur qui est connue (cf. équation (2.52), paragraphe 2.1.4.2) nous permet d'estimer la projection des caractéristiques dans l'espace de l'image. La méthode choisie d'association est alors appliquée.

Dans cet exemple schématisé, on peut noter que le critère de distance peut entraîner une mauvaise association des deux lignes qui composent le plan situé à gauche de l'effecteur. Ce risque peut être pris en compte dans le critère d'association de sorte que si deux couples de nature identique sont trop proches, ils seront écartés et ne seront pas utilisés pour la construction d'une tâche.

Au final, à l'issue de cet étape, est disponible une liste de couples $(Z_{j,i}, Z'_{j,i} = \text{proj}_j(M_i))$ candidats pour la construction de tâches référencées capteurs. L'étape suivante est détaillée dans le paragraphe suivant.

2.3.3 Définition des tâches robotiques

Les deux paragraphes précédents nous ont permis de sélectionner des contraintes de positionnement en tenant compte à la fois d'informations issues du modèle virtuel du système et des informations issues des capteurs qui équipent la cellule robotique.

L'étape suivante est la sélection de ces contraintes en fonction de leur qualité, puis leur traduction en tâches référencées capteur. La prise en compte de tâches plus spécifique aux mouvements de bras robotiques sera également abordée pour l'ordonnancement des différentes contraintes spatiales et robotiques.

La sélection des contraintes pour le positionnement spatial peut se baser sur un critère de qualité de la mesure qui diffère selon le capteur et sa nature. Par exemple, une mesure provenant d'une caméra peut être accompagnée d'un test d'occlusion potentielle, implémenté en suivant la question : est-ce que la mesure de la caractéristique ne risque-t-elle pas de sortir du champ de vision ?

Informations des couples candidats	Composants pour une tâche référencée capteurs
Mesure $Z_{j,i}$	Vecteur d'état s_i
Projection $Z'_{j,i} = f(M_i)$	Vecteur d'état à l'équilibre s_i^d
Nature de M_i et fonction de projection capteur j	Matrice d'interaction L_{s_i}
Situation du capteur j dans la cellule	Matrice de transformation cinématique ${}^S T_N$

TABLE 2.1 – Correspondance entre les éléments qui composent les couples issus de la sélection des contraintes spatiales depuis les informations modèles et mesures capteurs et les éléments nécessaires pour la construction d'une tâche référencée capteurs.

De même, un critère lié à la résolution de la caméra peut être mis en place pour discriminer les mesures trop peu précises. En effet, la même mesure d'une caractéristique extraite d'une séquence d'images provenant d'une caméra d'une résolution de 320×240 pixels n'aura pas le même impact qu'avec une caméra de résolution 1920×1280 pixels. Pour un positionnement où l'objectif de précision est le plus ambitieux, la mesure de la deuxième caméra sera préférée à la première. Les critères liés à la résolution du capteur peuvent être étendus aux signaux capteurs provenant de caméras à profondeur de champ, télémètres 2D et 3D, etc.

Une fois cette dernière sélection passée, le couple de contraintes spatiales peut être traduit en tâche référencée capteur. Pour rappel, le couple comporte un certain nombre d'informations qui peuvent être directement liées aux informations nécessaires à la construction d'une tâche. La correspondance entre ces éléments est effectuée dans le tableau 2.1.

Outre la nécessité d'intégrer les contraintes spatiales pour réaliser une tâche de positionnement, il est aussi nécessaire de prendre en compte les capacités réelles du bras manipulateur au sein de la commande. Parmi les informations incontournables à prendre en compte, il y a les butées articulaires, la manipulabilité, les collisions potentielles entre corps, etc. Ces contraintes ne peuvent être considérées comme des contraintes spatiales. Leur prise en compte permet de garantir que le mouvement réalisé sous la commande référencée capteur qu'il se fera dans de bonnes conditions. En effet, si le robot vient à entrer en butée sur l'une de ses articulations, il perd tout simplement un degré de liberté et le mouvement engendré pourrait s'avérer être dangereux pour l'environnement proche. La construction d'une tâche pour l'évitement de butée articulaire n'est pas présenté dans ce paragraphe.

L'agencement des tâches s'effectue en ayant recours au schéma de séquençage de tâches (cf. équation (2.58)), qui est le schéma de fusion de commande le plus utilisé et le plus robuste à l'heure actuelle. Les contraintes les plus prioritaires sont celles qui garantissent que la réalisation du mouvement s'effectue dans de bonnes conditions (*i.e.* évitement de butées articulaires et auto-collision). Ensuite dans l'ordre des priorités viennent les contraintes spatiales qui fixent les degrés de liberté de l'effecteur, pour finir sur les tâches annexes sur l'occupation de l'espace par les corps et/ou la maximisation de la manipulabilité du robot. Toutes ces contraintes peuvent être appliquées sous réserve que le robot ait un nombre de degrés de liberté suffisant. Par exemple, un robot non redondant ne peut pas modifier l'espace qu'il occupe en se reconfigurant sans violer les contraintes spatiales. Les tâches appliquées sont ajustées en fonction.

2.3.4 Application et surveillance de l'exécution

L'approche décrite ici permet principalement de faciliter la création de tâches au sens commande référencée capteurs par l'agglomération d'informations issues des modèles (modèle virtuel de la maquette numérique, modèle virtuel de la cellule robotique, modèle de projection des différents capteurs, etc.) ainsi que des informations liées au monde réel via la perception de l'environnement au travers des capteurs.

Néanmoins, l'exécution du mouvement et la position atteinte à la fin de la convergence de la loi de commande apporte une certaine quantité d'informations qui peuvent être utilisées par d'autres systèmes externes.

En se plaçant dans un contexte industriel, la qualité de la tâche de positionnement est une information qui peut être directement remontée dans un système d'information ou de supervision de l'entreprise au format numérique et ainsi peut être utilisée pour consolider une maquette numérique réelle. Pour une pièce donnée, les concepteurs en bureau d'études auront l'information précise. Ils évitent ainsi de statuer sur la qualité du produit fini par les prismes des mesures de tolérance, mesures qui par définition apportent de l'incertitude.

De même si l'on considère une seule action de positionnement, la performance et la répétabilité de cette action atomique n'existent pas en tant que données dans les systèmes d'information des chaînes d'assemblage – informations pouvant servir aux processus logistiques pour optimiser le flux de matériel par exemple – ou plus généralement aux processus de gestion de ressources dans le même but d'optimiser les interactions entre les services.

L'approche qui a été décrite offre la possibilité de s'articuler avec d'autres systèmes haut-niveau dans une usine. Elle permet entre autres de fournir des informations et des données sur des opérations simples. Ces données n'existent pas à l'heure actuelle et donc ne sont pas utilisées pour le contrôle et l'estimation de la qualité du produit fini.

2.3.5 Critique

L'approche considérée se présente comme une alternative à la programmation classique robotique qui détermine la trajectoire d'un effecteur en tenant compte exclusivement des informations dites modèles d'un système. Néanmoins, il reste un long chemin à faire pour parvenir à ce résultat. Ci-après sont abordés quelques points pour estimer le chemin restant à parcourir.

D'un point de vue académique, l'ordonnancement des tâches décrivant les contraintes spatiales n'est pas optimal. Les critères de visibilité et de qualité de la mesure ne sont pas suffisants pour ordonner les contraintes spatiales afin de satisfaire l'ensemble des cas possibles que l'utilisateur pourrait rencontrer. Une solution de contournement pourrait laisser le choix à l'utilisateur en cas de problème et devient alors un problème d'implémentation.

Plus largement, le critère de visibilité pour la sélection des tâches implique que l'approche ne peut être appliquée que lorsque l'effecteur est au voisinage de sa position finale. Cela contredit l'aspiration de l'approche à remplacer le cadre de programmation, puisque celle-ci n'est pas capable de déterminer l'ensemble de la trajectoire. Dans les considérations précédentes, l'approche vient donc en complément du cadre de programmation basé modèle pour affiner la pose de l'effecteur par rapport à des références appartenant au monde réel.

L'implémentation d'un tel logiciel repose aussi sur des défis en relation directe avec les outils

numériques. La sélection des contraintes repose aussi sur l'association entre des observations issues de traitement de signaux capteurs et des prévisions calculées à partir du modèle. La méthode d'association doit prendre en compte l'ensemble des approximations qui composent l'ensemble des deux chaînes d'extraction : par exemple, un signal capteur est souvent soumis à un bruit de mesure ou encore, la prévision est calculée à partir d'un modèle de projection dont les paramètres ont été estimés à l'aide d'une procédure de calibration. Le critère d'association doit donc être robuste à ces approximations.

D'un point de vue pragmatique, l'intégration de l'approche en amont au sein de logiciels CAO utilisés par les bureaux d'études dans un contexte industriel se révèle être un challenge non négligeable. En effet, les équipes dans leur grande majorité, sont dotées de logiciels commerciaux qui ont leur propre vision d'une description d'un monde virtuel. Même si les moyens pour représenter un modèle volumique à trois dimensions sont identiques à la base, chaque logiciel utilise sa propre implémentation et ce pour différentes raisons comme par exemple le souhait d'une interopérabilité restreinte. À signaler que le problème est identique du côté robotique : en effet, chaque constructeur observe sa propre politique d'accès sur ses machines, qui permettent ou non d'y appliquer telle ou telle commande.

De nombreuses étapes restent à parcourir pour déployer un tel logiciel. Un point non négligeable aussi qui n'a pas été abordé, est la prise en compte du contact au sein du schéma de commande référencée capteurs et multi-capteurs. Le chapitre 4 lui est entièrement consacré.

2.4 Application pour le positionnement d'une pièce

Cette section démontre l'intérêt de l'approche présentée dans la section précédente à travers la simulation d'un assemblage d'une pièce. Étant donné que la question du contact n'est pas encore traitée, la simulation se limite à la première étape : un positionnement d'une pièce au voisinage très proche de sa position finale définie par rapport à la structure. Ce travail a fait l'objet d'une publication [[Vandernotte et al., 2014](#)].

La section présente d'abord le contexte de l'expérimentation ainsi que le matériel utilisé pour la simulation. Puis, l'opération de positionnement sera définie en termes de contraintes spatiales en utilisant des mesures de distances, informations géométriques appartenant aux caractéristiques de type 3D qui n'ont pas été présentées jusqu'à alors. Enfin seront présentés les résultats de simulation.

2.4.1 Contexte

Le souhait initial de ce travail était de proposer une preuve de concept de l'approche à travers la simulation d'un positionnement d'une équerre pour consolider une étagère fixée au coin d'une pièce. Le but était de montrer qu'un positionnement précis était possible en utilisant un robot quelconque où la position finale de l'objet était exprimée en termes de contraintes spatiales géométriques définies par rapport à la structure.

La scène est composée du robot placé proche de la structure. Le robot utilisé est le bras manipulateur LWR4+ (LWR pour *Light Weight Robot*) du constructeur allemand *KUKA Systems GmbH*. Le robot représente le concept de robot léger dédié à la manipulation dans un environnement non déterministe, pour être appliqué dans un contexte de travail collaboratif. Les points clés de ce ro-

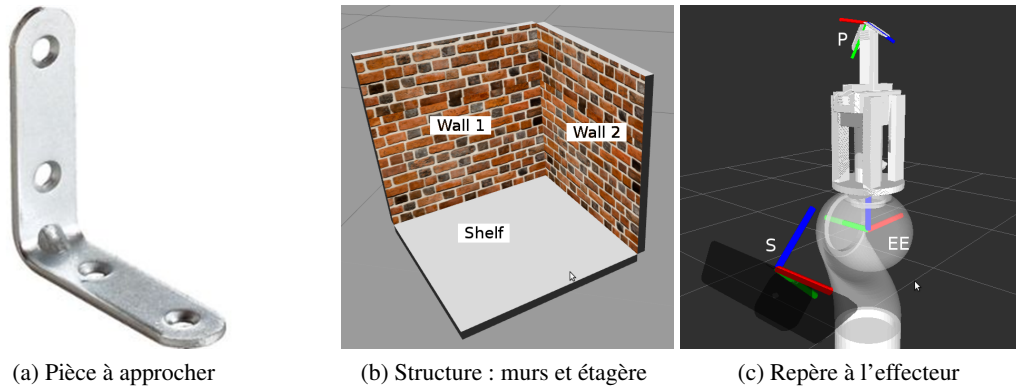


FIGURE 2.6 – Illustration du cas d'étude pour le positionnement d'une équerre pour le renfort d'une étagère.

bot sont sa conception mécanique qui empêche les pincements et autres actions contondantes et dangereuses pour un opérateur travaillant à côté; et sa loi de commande compliant ainsi que ses capteurs d'effort intégrés qui permettent une détection de collision pour déclencher un arrêt en cas de contacts non-désirés. Le LWR a été conçu par les équipes du laboratoire allemand DLR. Bischoff et al. [2010] résument les principales caractéristiques de ce robot.

L'étagère est placée dans un coin de la pièce, à une certaine hauteur. Les deux murs et l'étagère forment une collection de plans rigides orthogonaux. L'équerre (cf. figure 2.6a) est tenue par une pince parallèle de type EGN 80-160 du constructeur *SCHUNK GmbH*. Les repères des différents éléments à l'effecteur sont représentés dans la figure 2.6c. On suppose que la pièce est saisie de manière rigide et que la pose de celle-ci est connue par rapport au repère de l'effecteur. La pièce est considérée comme étant assez légère pour éviter son glissement pendant l'opération.

L'effecteur est équipé d'un capteur à profondeur de champ de type *Microsoft Kinect* qui génère un nuage de point trois dimensions dans le repère capteur. Les mesures utilisées sont des mesures de distances entre des points pertinents de la pièce et les trois plans de la structure. Pour les besoins de la simulation, l'environnement est laissé volontairement simple pour éviter toute forme d'occlusion et n'empêchera pas l'obtention des mesures nécessaires à la loi de commande présentée plus loin.

2.4.2 Définition des caractéristiques de distance

Les caractéristiques sélectionnées pour mener à bien le positionnement sont des mesures de distance estimées à partir de l'équation d'un plan P et d'un point M . Ces deux derniers éléments sont extraits du nuage de points fourni par le capteur. Le but ici est de déterminer la nature de la caractéristique pour en déterminer sa matrice d'interaction.

L'équation qui décrit la distance orthogonale entre un point M ayant pour paramètres $\mathbf{x} = [x \ y \ z]^T$ et un plan P ayant pour paramètres $\mathbf{p} = [a \ b \ c \ d]^T$ est telle que :

$$d_{PM}(\mathbf{x}, \mathbf{p}) = \frac{ax + by + cz + d}{\sqrt{a^2 + b^2 + c^2}} \quad (2.65)$$

Une autre formulation de la distance est possible : en utilisant la notation d'un plan $\mathbf{p} = [\mathbf{n}^T \ -\mathbf{x}_0 \cdot \mathbf{n}]^T$ donnée dans le paragraphe 2.1.4.1 et en utilisant la notation homogène pour $\mathbf{x}$, la relation

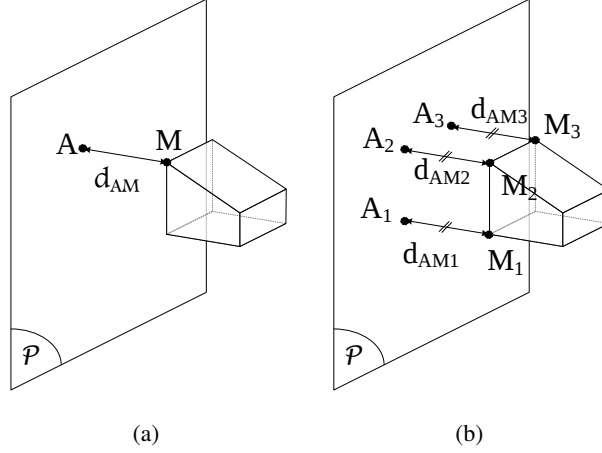


FIGURE 2.7 – Définition des mesures de distance

de la distance devient :

$$d_{PM}(\mathbf{x}, \mathbf{p}) = \frac{1}{\|\mathbf{n}\|} \mathbf{p} \cdot \mathbf{x} = \frac{1}{\|\mathbf{n}\|} [\mathbf{n}^\top - \mathbf{x}_0 \cdot \mathbf{n}] \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} \quad (2.66)$$

avec $\mathbf{x}_0 \in P$ est un point 3D quelconque appartenant au plan P et $\|\mathbf{n}\| = 1$ puisque $\mathbf{n}$ est un vecteur unitaire. La figure 2.7 introduit le point A comme étant la projection orthogonale de M sur P .

La matrice d'interaction donne la relation entre la variation de la distance d_{PM} et le torseur cinématique du capteur $\mathbf{v}$. Pour la déterminer, il est possible de s'appuyer sur la matrice d'interaction déjà connue entre la variation des composantes d'un point 3D et le torseur de la caméra. On s'attend à ce que cette relation soit de la forme suivante (ici, le point 3D utilisé est $\mathbf{x}_0$, point 3D quelconque appartenant au plan) :

$$\dot{d}_{PM} = \frac{\partial d_{PM}}{\partial \mathbf{x}_0} \mathbf{L}_{\mathbf{x}_0} \mathbf{v} = \mathbf{L}_{d_{PM}} \mathbf{v} \quad (2.67)$$

A partir de (2.66), on en déduit la dérivée partielle $\frac{\partial d_{PM}}{\partial \mathbf{x}_0}$ telle que :

$$\frac{\partial d_{PM}}{\partial \mathbf{x}_0} = -\frac{1}{\|\mathbf{n}\|} \mathbf{n}^\top = -\mathbf{n}^\top \quad (2.68)$$

En utilisant la matrice d'interaction pour un point 3D donnée en équation (2.40), la matrice d'interaction pour une mesure de distance entre un point M et un plan P est donc définie par :

$$\begin{aligned} \mathbf{L}_{d_{PM}} &= -\mathbf{n}^\top \begin{bmatrix} -\mathbf{I}_3 & [\mathbf{x}_0]_\times \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{n}^\top & (\mathbf{n}^\top \times \mathbf{x}_0) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{n}^\top & (\mathbf{n} \times \mathbf{x}_0)^\top \end{bmatrix} \end{aligned} \quad (2.69)$$

Ainsi, il est possible à partir de l'équation du plan et d'un point quelconque appartenant au plan, de déterminer la matrice d'interaction pour une mesure de distance telle que définie par l'équation

(2.66). A noter que le rang de $\mathbf{L}_{d_{PM}}$ est égal à 1, ce qui implique que la tâche référencée capteur associée pour une seule mesure ne contraindra qu'un seul degré de liberté des mouvements du repère capteur et/ou de l'effecteur.

Les mesures de distances peuvent être composées pour décrire de nouvelles contraintes spatiales moyennant très peu d'effort. Par exemple, il est possible de définir une contrainte spatiale de type distance d'un plan géométrique à un autre en définissant trois distances telles que définies ci-dessus comme illustré par la figure 2.7b. La tâche référencée capteur pour cette nouvelle contrainte est présentée ci-après :

$$\mathbf{s} = \mathbf{d} = \begin{bmatrix} d_{PM_1} & d_{PM_2} & d_{PM_3} \end{bmatrix}^T \quad (2.70a)$$

$$\mathbf{L}_d = \begin{bmatrix} \mathbf{L}_{d_{PM_1}}^T & \mathbf{L}_{d_{PM_2}}^T & \mathbf{L}_{d_{PM_3}}^T \end{bmatrix}^T \quad (2.70b)$$

$$\mathbf{s}^d = \begin{bmatrix} d_{PM_1}^d & d_{PM_2}^d & d_{PM_3}^d \end{bmatrix}^T = f \left(d^d, \mathbf{x}, \mathbf{p} \right) \quad (2.70c)$$

$${}^S\mathbf{T}_N \text{ connu} \quad (2.70d)$$

Les distances désirées sont calculées en tenant compte de l'équation du plan, de trois points non confondus appartenant à la pièce à positionner et de la distance désirée entre les deux plans. Une contrainte de distance fixe alors 3 degrés de liberté de la pièce : le degré en translation le long de l'axe normal au plan $\mathbf{n}$ et les deux degrés en rotation selon les axes orthogonaux à $\mathbf{n}$ et entre eux.

Par ailleurs, si la distance désirée est égale à zéro, cette contrainte de distance de plan à plan devient automatiquement une contrainte de type plans coïncidents. Cet exemple démontre la flexibilité de la définition des tâches pour la commande référencée capteurs et sa cohérence avec les outils de CAO et leurs contraintes d'assemblage.

2.4.3 Construction de l'opération de positionnement

On suppose que l'utilisateur a défini son assemblage en utilisant trois contraintes de distance de plan à plan dans son logiciel de conception assistée par ordinateur et que ces contraintes ont été sélectionnées pour être traduites en tâche référencée capteurs. Deux cas de figure peuvent être envisagés pour construire la commande qui réalisera l'opération. Nous verrons que ces tâches peuvent être appliquées séquentiellement pour garantir un contrôle maximal de l'utilisateur sur le comportement du bras manipulateur.

Il est nécessaire de noter que l'application brute des tâches issues des contraintes de distance de plan à plan (*i.e.* empiler l'ensemble des contraintes dans une seule et même tâche) rend la commande redondante. Certaines composantes entrent directement en conflit entre elles. En effet, les neuf contraintes de distances peuvent être combinées linéairement. L'utilisateur peut faire une lecture des besoins juste nécessaires pour fixer les degrés de liberté de la pièce. La figure 2.8 montre comment les contraintes de distances peuvent être choisies sans que celles-ci entrent en conflit. En effet, lorsque qu'une contrainte de distance plan à plan est réalisée, il reste trois degrés de liberté à fixer. Deux distances supplémentaires sont alors nécessaires pour contraindre l'éloignement par rapport au second plan ainsi que la rotation de la pièce. Lorsque ces deux contraintes sont réalisées, il ne reste plus qu'à fixer l'éloignement de la pièce par rapport au troisième.

La seconde approche est un exercice de logique et conduit à la construction de la tâche selon une approche qui n'est plus générique. En d'autres termes, la sélection automatique des contraintes peut amener à sélectionner des contraintes qui peuvent entrer en conflit. Ce type de problèmes peut

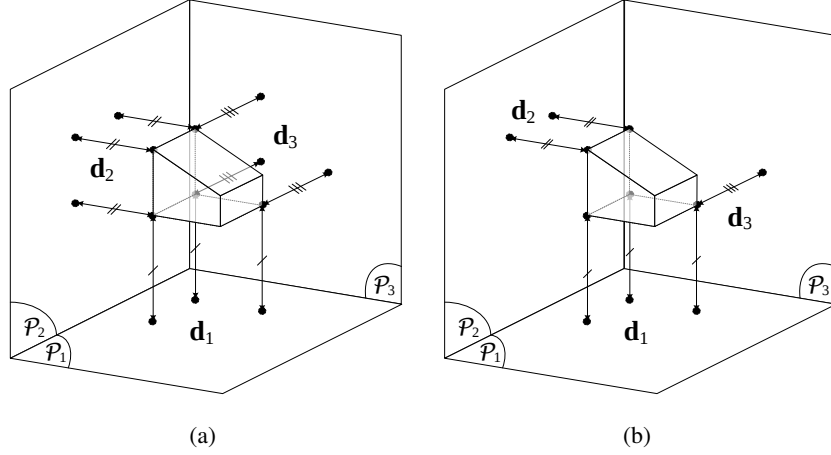


FIGURE 2.8 – Sélection des contraintes pour l'opération de positionnement : (a) Utilisation des distances pour définir de nouvelles contraintes de distance plan à plan. L'exemple est sur-contraint (le nombre de contraintes de distance est supérieur au juste nécessaire). (b) Sélection des contraintes juste nécessaires pour fixer tous les degrés de liberté de la pièce.

être résolu par une approche spécifique à la situation. L'utilisateur perd alors l'aspect systématique et la facilité de mise en œuvre de l'approche proposée. Dans tous les cas, l'utilisateur conserve la possibilité de faire un choix entre automaticité et justesse de la réponse du système.

La commande peut être construite de deux manières différentes : soit l'utilisateur souhaite appliquer chacune des contraintes spatiales sans ordonnancement particulier, soit il souhaite maîtriser le comportement en appliquant les contraintes en même temps avec un ordre de priorité, ou encore séquentiellement l'une après l'autre dans l'ordre des priorités définies.

Dans le premier cas (cf. illustration 2.8a), la commande est construite en empilant l'ensemble des contraintes de distance plan à plan. La commande, si seules les contraintes spatiales sont considérées et si celles-ci génèrent des contraintes articulaires, est directe et identique à l'équation (2.33) rappelée ci-après :

$$\dot{\mathbf{q}} = -\lambda (\mathbf{L}_s^S \mathbb{T}_N \mathbf{J})^+ (\mathbf{s} - \mathbf{s}^d) \quad (2.33)$$

Où $\mathbf{s} = [\mathbf{d}_1^T \mathbf{d}_2^T \mathbf{d}_3^T]^T$ et $\mathbf{L}_s = [\mathbf{L}_{d_1}^T \mathbf{L}_{d_2}^T \mathbf{L}_{d_3}^T]^T$. $\mathbf{s}$ est donc composé de trois groupes de contraintes de distances. Chacun des groupes contiennent trois caractéristiques de distances : la dimension de $\mathbf{s}$ est alors telle que $m = \dim(\mathbf{s}) = 9$. Dans ce cas précis, certaines composantes auront des contributions contradictoires. La convergence du système n'est pas garantie et le comportement du robot peut s'avérer ne pas correspondre à ce que l'utilisateur attend.

Le second cas fait appel au schéma de séquençement de tâches efficient défini par l'équation (2.58). Ici, l'utilisateur établit l'ordre de l'application des tâches soit de manière arbitraire, soit de façon automatique en se basant sur des critères tels que défini dans le paragraphe 2.3.3. On suppose que l'utilisateur souhaite que la contrainte du plan P_1 soit plus prioritaire que celle du plan P_2 , et celle du plan P_3 la moins prioritaire. On utilise la notation suivante pour illustrer l'ordonnancement choisi :

$$\mathbf{d}_1 \succ \mathbf{d}_2 \succ \mathbf{d}_3 \quad (2.71)$$

L'expression ci-dessus se lit explicitement de la manière suivante : la tâche $\mathbf{d}_1$ est exécutée avant

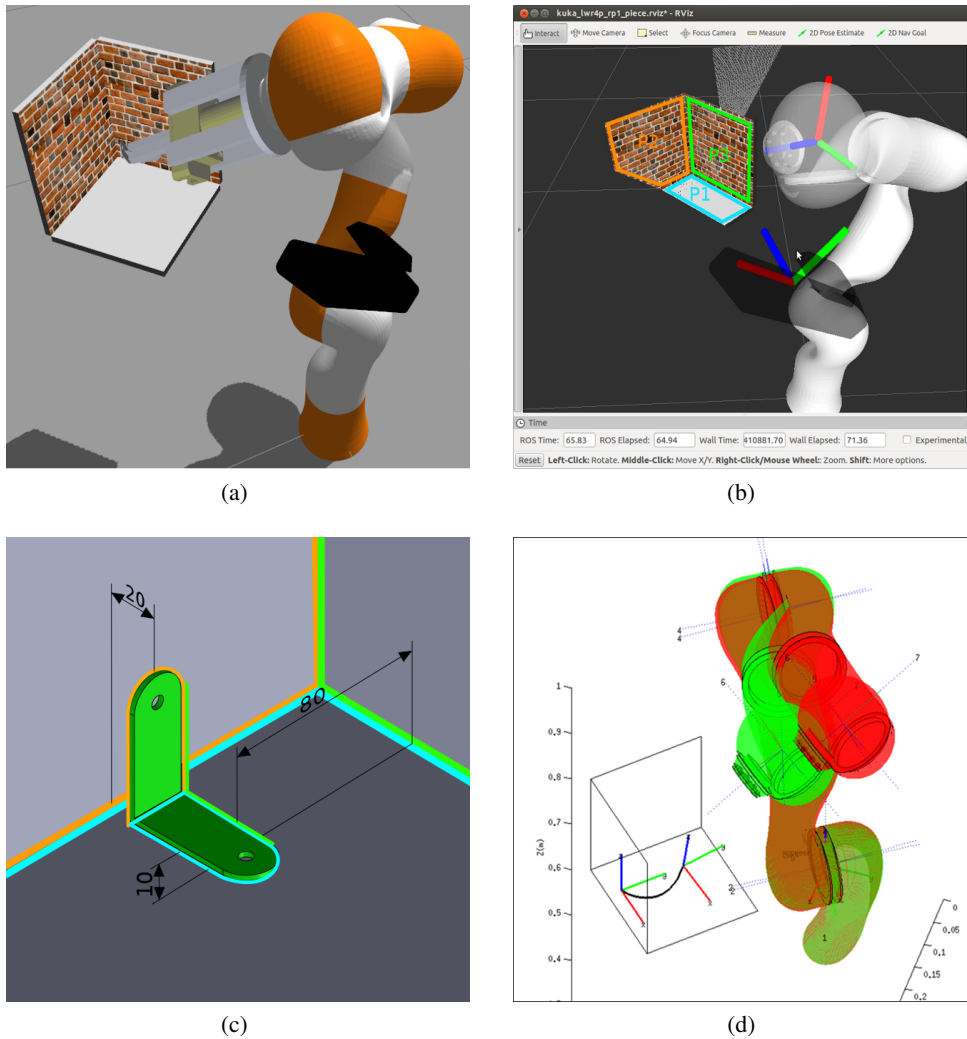


FIGURE 2.9 – Simulation du positionnement précis : (a) scène du logiciel Gazebo [Koenig et al., 2002]; (b) état du robot et du capteur selon les informations issues du simulateur; (c) définition des contraintes et leurs objectifs; (d) exécution de la commande référencée capteur.

d_2 , qui sont exécutées avant d_3 .

Ici, le problème des contributions contradictoires entre les différentes composantes des tâches est solutionné par l'utilisation du schéma de séquençage de tâches. En effet, celui-ci garantit que la tâche la moins prioritaire ne puisse accéder qu'au sous-espace des mouvements disponibles laissés par les autres.

2.4.4 Preuve de concept par simulation

Le positionnement d'une équerre pour la consolidation d'une étagère fixée sur deux murs orthogonaux a été simulé pour établir la preuve de concept. Les simulations ont été effectuées en utilisant le logiciel libre Gazebo [Koenig et al., 2002] et le cadre de programmation pour la robotique ROS (Robot Operating System) [Berger et al., 2007].

Le scénario de positionnement précis est le suivant : le robot manipulateur saisit la pièce placée auparavant sur une surface dédiée. On suppose que l'action de saisie s'est correctement déroulée et n'a pas introduit d'incertitude sur la pose entre le repère de la pièce et celui de l'effecteur. Dans un second temps, le robot déplace l'ensemble effecteur – composé du préhenseur, du capteur et de la pièce saisie – vers une pose au voisinage de la destination finale de la pièce. Étant donné que le système n'a qu'une connaissance partielle de la scène et une connaissance approximative de la pose entre son repère de référence et celui de la structure, il n'est donc pas en capacité de déterminer la pose finale de la pièce avec précision. La figure 2.9a montre le robot dans sa position initiale. Lorsque le système est au voisinage de la pose désirée, le capteur commence sa lecture de la scène et mesure les équations des plans, nécessaire pour faire fonctionner la commande comme définie dans le paragraphe précédent. La capture d'écran en figure 2.9b montre ce qu'observe le capteur. Sont illustrés les plans après l'opération de segmentation et d'estimation de leurs paramètres à partir du nuage de points. Le processus d'extraction a été réalisé en utilisant les fonctionnalités de la librairie PCL (*Point Cloud Library*) [Rusu and Cousins, 2011].

La commande utilisée est celle où on utilise le schéma de séquençement de tâche pour arbitrer les contributions de chacune des tâches. La commande a été appliquée dans les deux cas de figure décrit ci-dessus où les tâches sont exécutées séquentiellement une à une, puis toutes en même temps. Pour la simulation, on suppose que l'utilisateur a souhaité positionner la pièce telle que montré sur la figure 2.9c (les dimensions indiquées sont en millimètres). Les objectifs de distances de plan à plan sont alors tels que $d_1^d = 0.01m$, $d_2^d = 0.02m$ et $d_3^d = 0.08m$. Les gains λ_1 , λ_2 et λ_3 sont égaux, réglés de manière à ce que le temps de convergence de chacune des tâches prises séparément n'excède pas 1 seconde. Pour la simulation où les tâches sont exécutées séquentiellement, les tâches sont relâchées dans leur ordre de priorité défini par la notation (2.71), respectivement à $t_1 = 0s$, $t_2 = 2s$ et $t_3 = 6s$.

Dans les deux cas, le système converge à la position finale désirée. Les figures 2.10a et 2.10b montrent l'évolution du robot dans l'espace opérationnel pendant la convergence. Le robot dans sa configuration initiale est représenté en rouge. Cette configuration correspond à la pose incertaine obtenue par calcul en se basant exclusivement sur des informations issues des modèles virtuels. En vert est représenté le robot une fois que la commande référencée capteur a terminée sa convergence. Le repère attaché à l'effecteur représente le repère de la pièce. La trajectoire cartésienne de son origine est représentée en noir. Les évolutions des différentes composantes de distances sont montrées dans les graphiques 2.10c et 2.10d en fonction du temps. Sont représentées aussi les vitesses articulaires pour les deux cas, figures 2.10e et 2.10f. Les graphes pour l'évolution des distances montrent neuf composantes de distances rassemblées en groupes de trois conformément à la construction des tâches en utilisant des contraintes de distance plan à plan.

Premièrement, la propriété de la commande référencée capteur est réalisée : on observe que les distances parviennent à leur valeur à l'état d'équilibre en décrivant une exponentielle décroissante. On observe aussi que la vitesse de décroissance est identique pour chacune des tâches, conséquence directe du choix de gains égaux. À chaque événement, les vitesses articulaires décrivent une évolution exponentielle vers zéro jusqu'à stabilisation complète du système. La colonne de gauche montre que l'ordre de priorité est respecté. De plus, une des propriétés attendues du schéma de séquençement de tâches en équation (2.58) est réalisée : une fois que la tâche la plus prioritaire ait terminé son action, l'évolution des autres tâches ne la perturbe pas. La contrainte d_1 atteint son objectif à $t = 1s$ et ne bouge plus, même si les autres tâches s'exécutent. Lorsque toutes les tâches sont relâchées en même temps (colonne de droite), la figure 2.10d illustre bien d'une autre manière cette propriété du projecteur orthogonal qui ne laisse qu'un sous espace des mouve-

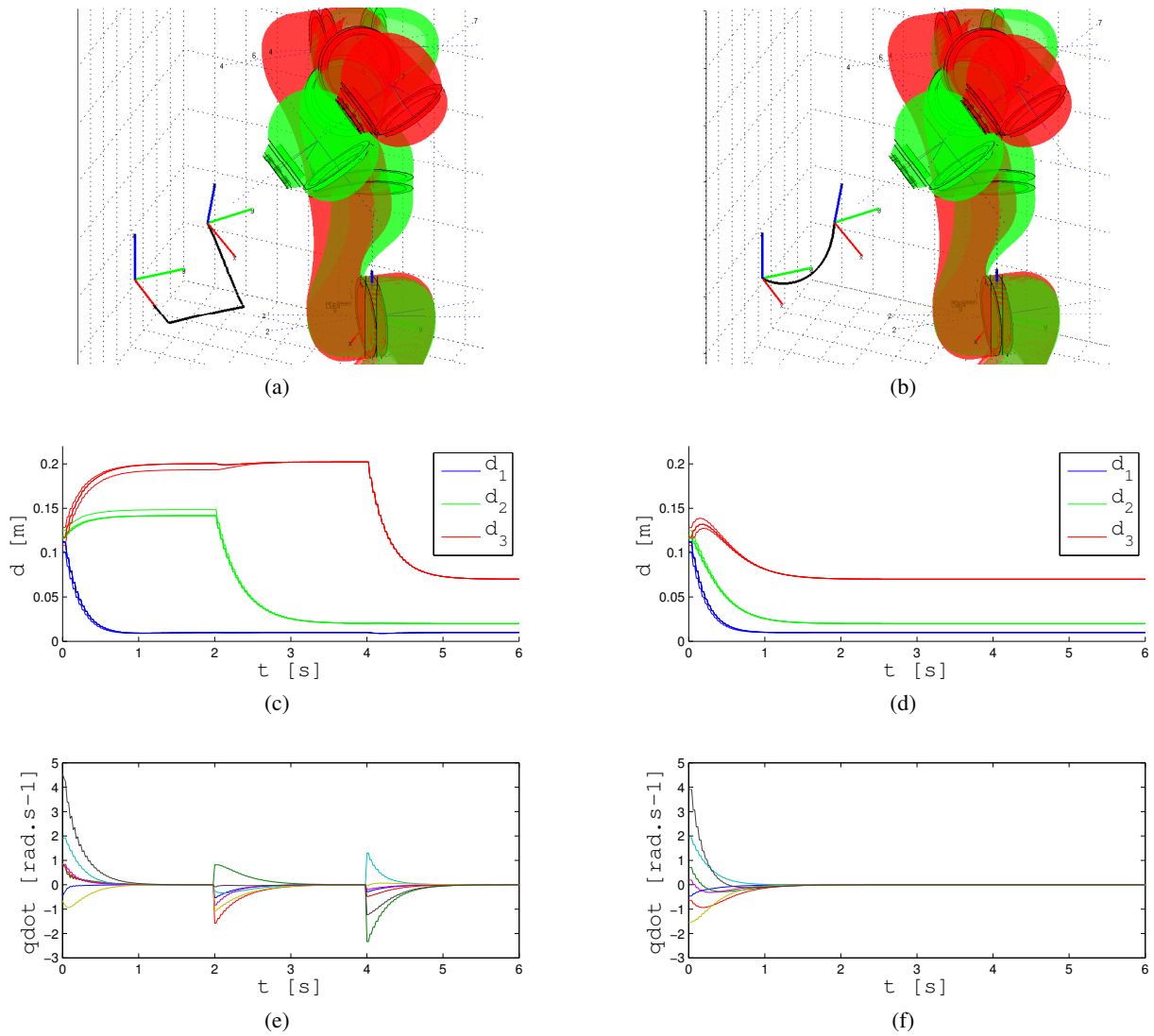


FIGURE 2.10 – Résultats de simulation pour le positionnement précis avec le logiciel Gazebo. À gauche, exécution de la commande où les tâches sont appliquées de manière séquentielle; à droite, toutes les tâches sont exécutées en même temps en respectant l'ordre des priorité.

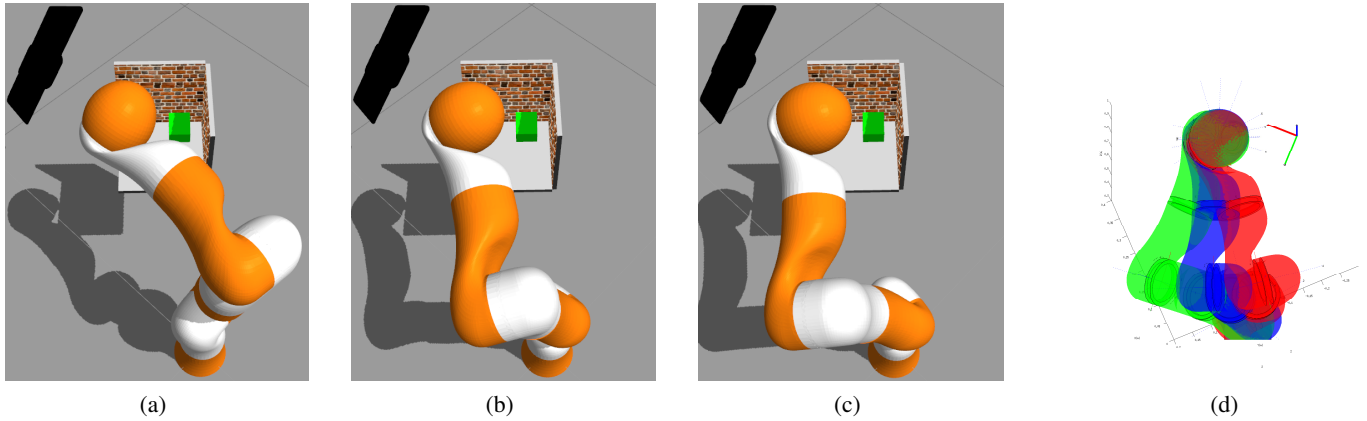


FIGURE 2.11 – (a)(b)(c) Séquence illustrant l'utilisation de la redondance du robot LWR4+ pour que celui-ci écarte ses corps de la structure en utilisant le schéma de séquençage multi-tâches ; (d) superposition des trois configurations et du repère de la pièce.

ments disponibles aux tâches les moins prioritaires. En effet, l'exécution de l'opération entraîne un léger dépassement pour la tâche d_3 puisque l'ensemble de l'espace des mouvements disponibles est réservé à la première et la seconde tâche. La troisième tâche ne commence à converger qu'à environ à $t \approx .33s$, instant où ses composantes de distances observent une réelle pente descendante.

Les simulations précédentes ne prenaient en compte que des contraintes spatiales. L'approche proposée est aussi capable de prendre en compte des contraintes robotiques et/ou environnementales. La figure 2.11 montre une nouvelle simulation où le robot doit s'écarter de la structure pour éviter une possible collision pendant la phase de dégagement, et ce tout en respectant les contraintes de positionnement de la pièce. La tâche d'écarterment a été ajoutée à la suite des autres contraintes et constitue la tâche la moins prioritaire. Les figures 2.11a, 2.11b et 2.11c montrent l'évolution de la configuration articulaire du robot LWR4+ à trois instants différents, du départ jusqu'à l'équilibre du nouveau système avec la commande à quatre tâches. Ces trois configurations articulaires ont été superposées dans la figure 2.11d afin de montrer que même pendant l'évolution de la configuration, le repère de la pièce n'a pas bougé, ce qui montre que l'ordre de priorité est bien respecté.

On note que ce mouvement de reconfiguration articulaire tout en respectant une tâche de positionnement à l'effecteur n'est rendu possible que grâce aux capacités du LWR4+. En effet, celui-ci fait partie de la classe des robots redondants qui admettent une infinité de configurations articulaires pour une pose donnée de l'effecteur, notamment grâce à son septième axe. Le schéma de séquençage de tâches et plus généralement le projecteur orthogonal sont aussi des moyens pour trouver plus facilement des configurations articulaires. En effet, certains algorithmes pour le modèle géométrique inverse utilisent ces outils afin de déterminer quelle est la meilleure configuration articulaire pour une pose donnée avec un robot redondant. Ils peuvent être utilisés autant en ligne dans le cadre d'une loi de commande, qu'hors ligne pour des algorithmes de recherche comme énoncé précédemment.

2.5 Conclusions

Dans ce chapitre, nous avons fait l'inventaire de toutes les approches existantes pour améliorer la précision absolue des bras manipulateurs et la fiabilité du travail qui leur est demandé.

Ce travail a été effectué en se reposant à la fois sur des techniques existantes telles que la commande référencée capteur et multi-capteurs, sujets qui ont fait l'objet d'un rappel de l'état de l'art courant; ainsi que par la proposition d'un nouveau cadre de programmation robotique pour faciliter la définition et la mise en place des tâches référencées capteurs.

Ce cadre de programmation est à mettre en concurrence directe avec la méthode qui est actuellement utilisée par les équipes d'intégrateurs. En effet, les robots industriels ne construisent leurs trajectoires qu'avec des données issues de modèles virtuels et n'ont pas la possibilité de s'adapter à leur environnement, ce qui les rend dangereux à la fois pour les opérateurs, ainsi que pour l'environnement lui-même.

La généralisation des outils tels que la commande référencée capteurs et le schéma de commande générique comme proposé dans la section 2.3 permet de programmer les robots avec des éléments appartenant au monde réel et ainsi gagner un comportement plus lisible par les collaborateurs travaillant dans son environnement proche.

Le cadre en question a été éprouvé dans un exercice de positionnement précis d'une équerre pour la consolidation d'une étagère. L'opération de positionnement a été construite en suivant les directives proposées par le schéma de commande générique. Celle-ci a été simulée avec des outils de simulation robotique libres (sources ouvertes) et réalistes. Les outils développés spécialement pour la simulation constituent la première base logicielle de ce cadre.

Cependant, la commande référencée capteur qui constitue la base principale de ce cadre de programmation ne peut résoudre l'ensemble des problèmes liés à la macro-manipulation. Il est possible de citer quelques points faibles qui font que le cadre en l'état ne peut pas apporter de solutions performantes pour l'ensemble des classes d'applications que l'on peut trouver dans l'industrie manufacturière.

En premier lieu, la loi de commande utilise des informations issues d'un modèle virtuel de la structure et ce modèle ne correspond pas à la réalité. Il est possible alors que l'ensemble des tâches (contraintes spatiales) ne soient pas totalement réalisées (*i.e.* leurs erreurs ne convergent pas totalement vers zéro), même si la hiérarchisation des tâches empêche les conflits. Dans ce cas, le résiduel des erreurs peut servir à améliorer la connaissance du modèle ou alors, être renvoyé dans la chaîne de conception de la structure en amont de l'opération robotique.

Deuxièmement, la loi de commande issue du cadre ne permet pas un contrôle du bras manipulateur sur sa vitesse de déplacement. En effet, la commande référencée capteurs n'apporte qu'une solution à des problèmes quasi-statiques. Ainsi, le cadre en l'état ne peut pas proposer de solution à des classes d'applications telles que la peinture, le ponçage ou encore la gravure, ou tout autre application où le contrôle en vitesse est nécessaire.

Le chapitre suivant propose une nouvelle formulation de la commande référencée capteurs qui peut être utilisée pour ces classes d'applications.

Commande dynamique référencée multi-capteurs

Le chapitre précédent a montré qu'il était possible de généraliser les outils de commande référencée capteurs pour générer des contraintes à partir d'outils tels que des logiciels de conception assistée par ordinateur.

Néanmoins, le cadre proposé précédemment est incomplet et ne peut être utilisé sur toutes les classes d'application que l'on peut trouver en industrie manufacturière.

D'une part, le comportement d'une tâche comportant plusieurs contraintes spatiales n'est pas contrôlable avec la seule commande cinématique : le chemin parcouru par l'effecteur n'est pas prédictible et de ce fait, assez peu utilisable en pratique sur de larges erreurs ou de longues distances à parcourir. De plus, la capacité d'un contrôle sur la vitesse de déplacement et la variation des caractéristiques n'est pas possible.

D'autre part, le cadre de programmation n'est pas en capacité de prendre en compte directement des tâches liées à des informations d'effort. Cela est dû au fait que la nature d'information de force et/ou de couple est fondamentalement différente d'une information spatiale. Le lien entre ces deux informations n'est pas direct et les quantités ne peuvent pas s'exprimer au même niveau dans les modèles robotiques. Il n'est donc pas possible d'associer une contrainte spatiale et une contrainte d'effort dans une seule et même tâche. La commande cinématique n'est pas en mesure d'y apporter une solution.

Ce chapitre propose une nouvelle formulation de la commande référencée capteurs pour une contrainte spatiale pour passer d'une commande cinématique (*i.e.* commande qui génère des consignes de vitesses articulaires) à une commande dynamique (*i.e.* une commande qui génère des consignes d'efforts articulaires). Cette nouvelle formulation ouvre de nouvelles opportunités par rapport à la commande cinématique comme par exemple, un contrôle fin de la variation de la caractéristique dans l'espace capteurs.

La première section comportera un rappel de l'état de l'art sur les commandes dynamiques classiques. Sera présenté ensuite le modèle du second ordre de l'information capteur s qui est la dérivation de l'équation (2.1). À partir de ce modèle, seront présentées les différentes formes que

la loi de commande référencée capteurs dynamique peut prendre. Enfin, différents cas d'étude en simulation sont proposés pour valider le concept et vérifier sa robustesse.

3.1 État de l'art : commandes dynamiques

3.1.1 Généralités

Les commandes dynamiques proposent des méthodes de contrôle en boucle fermée où le bloc régulateur génère des consignes en effort transmises au système articulé à contrôler. Le but d'une commande en boucle fermée est d'apporter la garantie que l'exécution de la trajectoire demandée au système articulé respecte une certaine marge d'erreur.

Sans perte de généralités, plusieurs objectifs à réaliser sont à prendre en compte lors de la conception d'un régulateur. Parmi ceux-ci, nous pouvons lister les quantités d'erreur statique et d'erreur de poursuite, le temps de réponse du système en boucle fermée ainsi que les quantités autorisées de dépassement. Outre la méthode choisie pour construire le bloc de régulation en question, les gains qui composent le régulateur permettent d'ajuster son comportement pour atteindre ces objectifs.

Il est possible de justifier l'intérêt de générer des commandes en effort par une raison tout à fait pratique. En effet, les systèmes électro-mécaniques à contrôler sont composés de moteurs électriques pour la plupart. Deux méthodes existent pour commander ce type de moteur : l'utilisateur peut contrôler la vitesse de rotation de ces moteurs en variant la tension aux bornes de celui-ci, ou il peut contrôler le couple généré par le moteur en variant la quantité de courant à ses bornes.

Pour un moteur à courant continu, alors que la méthode de commande en vitesse via la variation de tension est sujette à des variations de paramètres physiques internes au moteur tels que l'augmentation de chaleur ou la variation de charge sur l'axe, le couple d'un moteur est directement proportionnel au courant qu'il l'alimente et ce, dans n'importe quelles conditions. La deuxième méthode apporte donc moins d'incertitude sur le système à commander puisque ces paramètres internes ne varient pas en fonction des conditions d'utilisation. Les commandes en effort ont vocation à donner des consignes de couples moteurs de manière à limiter l'impact des variations des paramètres internes des actionneurs.

La figure 3.1 montre un schéma bloc illustrant une commande en boucle fermée classique. Chacun des blocs peuvent être modélisé par une fonction de transfert. La construction du bloc régulateur revient à déterminer sa fonction de transfert afin de respecter les objectifs définis auparavant.

3.1.2 Commandes classiques

Le problème de la commande de robot manipulateur a été largement traité dans la littérature et de nombreuses méthodes ont été proposées.

L'approche la plus utilisée dans la robotique industrielle est le correcteur proportionnel, intégral, dérivé (PID) décentralisé pour chaque degré de liberté du robot. C'est aussi le plus utilisé en règle générale pour l'ensemble des systèmes à contrôler dans l'industrie. D'autres approches non linéaires, plus récentes ont été proposées. Parmi elles, il y a la commande en couple calculé (en anglais, *computed torque control*, aussi nommée *linear resolved acceleration approach*) dont le but est de découpler les équations de mouvement du robot et ainsi linéariser son comportement (du point de vue des coordonnées généralisées).

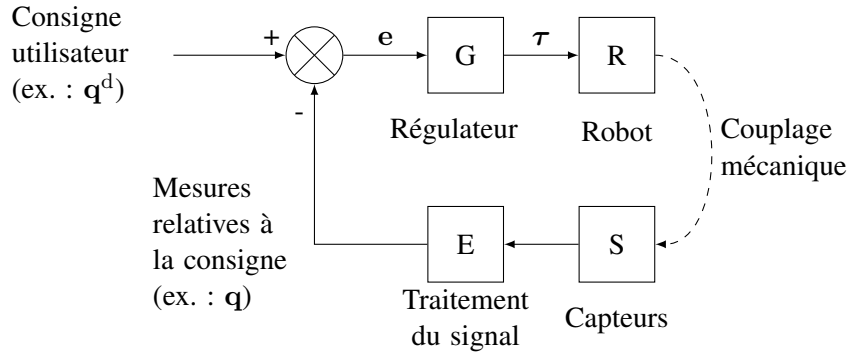


FIGURE 3.1 – Schéma bloc illustrant une commande en boucle fermée. Ici, le régulateur délivre des informations de type effort (exemple : couple articulaire). Souvent, les blocs image robot, capteurs et traitement du signal et le couplage mécanique qui les lie sont confondus dans un seul bloc.

De manière à faire face aux problèmes d'identification des paramètres internes du système à contrôler, certaines approches considèrent des techniques adaptatives pour identifier les paramètres dynamiques en ligne pendant l'exécution de la commande. Enfin, des méthodes alternatives utilisent des propriétés du modèle dynamique pour proposer des commandes basées sur Lyapunov et sur la passivité de système mécanique.

Cette section se focalise sur la présentation de l'approche de la commande par couple calculé et ses variantes. Celle-ci est en effet considérée comme la meilleure solution théorique pour le contrôle des systèmes robotiques en général et offre les meilleures performances. À partir de cette commande, d'autres approches qui présentent un intérêt dans le cadre de notre étude, seront présentées dans le chapitre 4 : à savoir la commande en impédance et les méthodes hybrides parallèles de contrôle en position et en force.

La commande des robots manipulateurs a largement été traitée dans la littérature. Le lecteur pourra trouver des études détaillées et des présentations théoriques dans [Samson et al., 1991; Khalil and Dombre, 2002; Chung et al., 2008; Siciliano et al., 2010].

3.1.2.1 Commande par couple calculé

La commande par couple calculé se base sur l'utilisation intensive du modèle dynamique inverse (*inverse dynamic model* ou IDyM) en tant qu'outil pour le découplage des équations de mouvement. Sa forme générale pour un robot sériel à n degrés de liberté est rappelée ci-après [Khalil and Dombre, 2002] :

$$\tau = A(q)\ddot{q} + C(q, \dot{q})\dot{q} + Q(q) + \text{diag}(\dot{q})F_v + \text{diag}(\text{sign}(\dot{q}))F_c + J^T f_e \quad (3.1a)$$

$$\tau = A(q)\ddot{q} + H(q, \dot{q}) \quad (3.1b)$$

où

- $\tau \in \mathbb{R}^n$ est le vecteur des couples articulaires du robot,
- $A(q) \in \mathbb{R}^{n \times n}$ est la matrice d'inertie,

- $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \in \mathbb{R}^n$ est le vecteur regroupant les effets des forces de Coriolis et des forces centrifuges,
- $\mathbf{Q}(\mathbf{q}) \in \mathbb{R}^n$ est le vecteur de couples lié à la gravité et au poids de corps du robot,
- $\mathbf{F}_c \in \mathbb{R}^n$ et $\mathbf{F}_v \in \mathbb{R}^n$ sont respectivement les coefficients de frottement respectivement sec et visqueux qui s'applique sur les articulations (c s'utilise pour identifier les frottements de Coulomb),
- $\mathbf{f}_e \in \mathbb{R}^n$ une éventuelle contribution d'effort (force et couple) s'appliquant sur l'effecteur.

L'équation du modèle dynamique inverse peut s'exprimer sous sa version courte en factorisant les forces de Coriolis et centrifuge, le vecteur de gravité, les frottements ainsi que les éventuelles forces extérieures sous l'étiquette $\mathbf{H}$.

Le couple transmet à l'articulation j à l'aide d'un moteur électrique commandé en courant (continu ou synchrone), en considérant l'hypothèse que la transmission n'introduit pas de jeu ni de flexibilité, est exprimée pour chaque articulation par :

$$\tau_j = n_j k_{aj} k_{Tj} u_j \quad (3.2)$$

où n_j est le rapport de réduction des engrenages, k_{aj} est le gain d'amplification du courant, k_{Tj} est la constante du couple moteur et u_j est l'entrée de commande du moteur relative à une quantité de courant. Comme énoncé plus haut, le but de la conception d'un contrôleur est de déterminer les couples articulaires pour en déduire ensuite les entrées de commande du moteur.

Le modèle dynamique direct décrit un système composé de n équations différentielles du second ordre couplées et non linéaires. Il est possible de remplacer le bloc G (figure 3.1) par un simple correcteur PID. Néanmoins le couplage fort entre les équations fait que les performances de la commande avec le correcteur PID sont moyennes, en particulier lors de trajectoire à haute vitesse. La commande par couple calculé permet de prendre en compte ces couplages dynamiques. Elle permet entre autres de découpler les équations de mouvement par l'annulation des non linéarités de la partie dynamique du robot. Ainsi dans des conditions parfaites, il est possible de considérer que les nouvelles entrées de commande ont un comportement linéaire et indépendant.

On décrit ci-dessous la construction de la commande. On pose l'hypothèse que les positions et vitesses articulaires sont mesurables et que le bruit de mesure est négligeable. En posant $\hat{\mathbf{A}}$ et $\hat{\mathbf{H}}$ les quantités estimées respectives de $\mathbf{A}$ et de $\mathbf{H}$, la partie de découplage est assurée par l'utilisation du modèle dynamique inverse :

$$\boldsymbol{\tau} = \hat{\mathbf{A}}(\mathbf{q})\mathbf{w}(t) + \hat{\mathbf{H}}(\mathbf{q}, \dot{\mathbf{q}}) \quad (3.3)$$

$\mathbf{w}(t)$ constitue le vecteur d'entrée découplée. L'utilisateur peut déterminer arbitrairement $\mathbf{w}$ pour mettre en place un schéma d'asservissement en position ou en vitesse. Par exemple, pour un asservissement en vitesse en utilisant les variables articulaires, le vecteur d'entrées découplées est de la forme :

$$\mathbf{w}(t) = \ddot{\mathbf{q}}^d(t) + \mathbf{K}_v(\dot{\mathbf{q}}^d(t) - \dot{\mathbf{q}}(t)) + \mathbf{K}_p(\mathbf{q}(t)^d - \mathbf{q}(t)) \quad (3.4)$$

où

- $\mathbf{q}^d(t)$, $\dot{\mathbf{q}}^d(t)$, $\ddot{\mathbf{q}}^d(t) \in \mathbb{R}^n$ constituent le profil de trajectoire en position, vitesse et accélération à réaliser en fonction du temps,

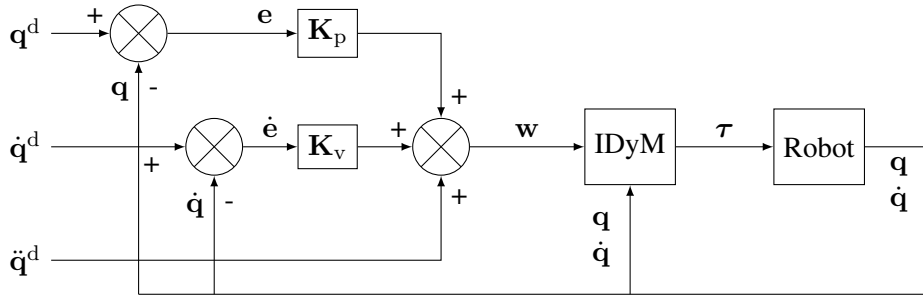


FIGURE 3.2 – Schéma bloc illustrant la commande par couple calculé, utilisant un correcteur proportionnel dérivée avec une action prédictive de l'accélération désirée en amont du modèle dynamique inverse. Le robot fournit à travers ses capteurs proprioceptifs des mesures des positions articulaires, les vitesses articulaires sont déduites par dérivation [Khalil and Dombre, 2002].

- $\mathbf{q}(t), \dot{\mathbf{q}}(t) \in \mathbb{R}^n$ sont les vecteurs images de la configuration articulaire et de ses vitesses articulaires courantes,
- $\mathbf{K}_p, \mathbf{K}_v \in \mathbb{R}^{n \times n}$ sont les matrices des gains proportionnel et dérivé, respectivement.

Le système d'équations différentielles qui décrit l'évolution de l'ensemble du système en boucle fermée est de la forme suivante :

$$\hat{\mathbf{A}} \left(\ddot{\mathbf{q}}^d + \mathbf{K}_v \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} \right) + \hat{\mathbf{H}} = \mathbf{A} \ddot{\mathbf{q}} + \mathbf{H} \quad (3.5)$$

$$\ddot{\mathbf{e}} + \mathbf{K}_v \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = \hat{\mathbf{A}}^{-1} \left((\mathbf{A} - \hat{\mathbf{A}}) \ddot{\mathbf{q}} + \mathbf{H} - \hat{\mathbf{H}} \right) \quad (3.6)$$

Si l'estimation des paramètres du modèle dynamique du système ne comporte pas d'erreur (*i.e.* $\hat{\mathbf{A}} = \mathbf{A}$, de même pour $\hat{\mathbf{H}}$), le second terme de l'équation (3.6) s'annule et le système est globalement exponentiellement stable. Les gains k_{pj} et k_{vj} sont choisis pour donner à l'articulation j le comportement désiré indépendamment de la configuration articulaire du robot. Si des erreurs sont présentes, elles constituent une excitation du système d'équations différentielles et la réponse temporelle associée convergera vers une constante différente de zéro. Cette constante peut tendre vers zéro par le réglage des gains dans la limite du domaine de stabilité.

La commande par couple calculé peut aussi s'appliquer dans l'espace cartésien pour contrôler directement la pose (translation et orientation) de l'effecteur. Cette autre formulation de la commande utilise conjointement le modèle dynamique inverse et le modèle cinématique du robot d'ordre 2. Le modèle dynamique du robot dans l'espace cartésien est de la forme :

$$\boldsymbol{\tau} = \mathbf{A} \mathbf{J}^{-1} \left(\ddot{\mathbf{x}} - \dot{\mathbf{J}} \dot{\mathbf{q}} \right) + \mathbf{H} \quad (3.7)$$

De la même manière que pour la commande dans l'espace articulaire, on écrit l'équation de découplage telle que :

$$\boldsymbol{\tau} = \hat{\mathbf{A}} \mathbf{J}^{-1} \left(\mathbf{w}(t) - \dot{\mathbf{J}} \dot{\mathbf{q}} \right) + \hat{\mathbf{H}} \quad (3.8)$$

Le vecteur d'entrées découplées est exprimé cette fois-ci dans l'espace cartésien.

$$\mathbf{w}(t) = \ddot{\mathbf{x}}^d(t) + \mathbf{K}_v (\dot{\mathbf{x}}^d(t) - \dot{\mathbf{x}}(t)) + \mathbf{K}_p (\mathbf{x}(t)^d - \mathbf{x}(t)) \quad (3.9)$$

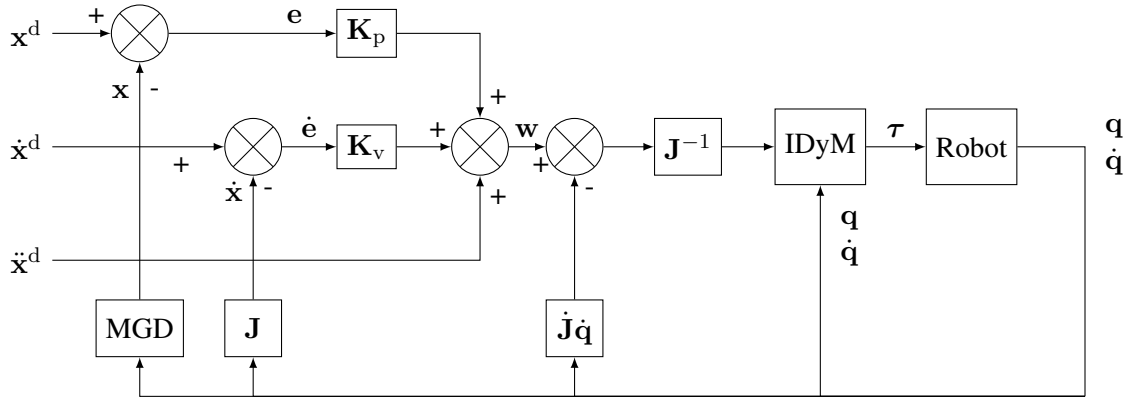


FIGURE 3.3 – Schéma bloc illustrant la commande par couples calculés dans l'espace cartésien. L'acronyme MGD représente le modèle géométrique direct.

Les propriétés de stabilité de la commande dans l'espace articulaire peuvent être transposées dans l'espace cartésien. Il est nécessaire de noter que cette commande requiert le calcul du produit $\mathbf{J}\dot{\mathbf{q}}$ où le terme dérivé $\dot{\mathbf{J}}$ n'est pas trivial à obtenir de façon analytique. [Khalil and Dombre \[2002\]](#) ont proposé une modification de l'algorithme pour le modèle dynamique direct récursif pour obtenir le produit sans dérivation de la matrice cinématique $\mathbf{J}$.

Les figures 3.2 et 3.3 montrent les différences entre les deux commandes.

Il est possible de trouver dans la littérature une troisième formulation du modèle dynamique du robot sériel. Cette formulation implique la généralisation du passage de tous les termes du modèle vers l'espace cartésien. L'équation est de la forme suivante [[Khatib, 1980, 1987](#); [Park, 2006](#); [Mansard et al., 2009a,b](#)] :

$$\mathbf{f} = \Lambda(\mathbf{q})\ddot{\mathbf{x}} + \mu(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{p}(\mathbf{q}) + \mathbf{f}_e \quad (3.10)$$

avec,

$$\Lambda(\mathbf{q}) = (\mathbf{J}\mathbf{A}^{-1}\mathbf{J}^T)^{-1} \quad (3.11a)$$

$$\mu(\mathbf{q}, \dot{\mathbf{q}}) = \Lambda \left(\mathbf{J}\mathbf{A}^{-1}\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \dot{\mathbf{J}}\dot{\mathbf{q}} \right) \quad (3.11b)$$

$$\mathbf{p}(\mathbf{q}) = \Lambda\mathbf{J}\mathbf{A}^{-1}\mathbf{Q}(\mathbf{q}) \quad (3.11c)$$

Les termes correspondent respectivement à :

- $\Lambda \in \mathbb{R}^{6 \times 6}$ est la matrice d'inertie du système exprimée dans l'espace cartésien,
- μ est le pendant du vecteur $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ qui rassemble les contributions des forces de Coriolis et centrifuge, dans l'espace cartésien,
- $\mathbf{p} \in \mathbb{R}^6$ est le vecteur du poids des corps du système dans l'espace opérationnel.

Le principe de la construction d'une commande à partir de cette formulation du modèle dynamique inverse reste identique aux formulations déjà présentées auparavant. Le modèle dynamique inverse est utilisé pour obtenir une équation de découplage.

La différence entre la commande par couple calculé et celle en espace opérationnel est que cette dernière fournit une consigne en force s'appliquant à l'effecteur alors que la première génère des consignes de couples articulaires. La formulation en espace opérationnel se veut plus générique car celle-ci permet une abstraction complète de la chaîne articulée.

3.1.3 Commande dynamique multi-tâches

Parmi les propositions dans la littérature, une loi de commande dynamique s'exprimant sous forme de tâches a été proposée dans [Mansard et al., 2009a] en ayant pour base les travaux de Khatib [1987]. Les auteurs reformulent l'équation du modèle dynamique inverse telle que :

$$\tau = J^T \Lambda \left(\ddot{x} - \dot{J}\dot{q} + J^{-1}AH \right) \quad (3.12)$$

On retrouve dans l'équation (3.13) l'expression de la matrice d'inertie du système exprimée dans l'espace opérationnel, déjà présentée dans l'équation (3.11a). L'équation constitue une nouvelle équation de découplage de la forme suivante :

$$\tau = J^T \Lambda \left(w(t) - \dot{J}\dot{q} + J^{-1}AH \right) \quad (3.13)$$

où le vecteur d'entrée découplée $w(t)$ peut être définie par l'équation (3.9) par exemple.

La principale différence par rapport à la commande classique en couples calculés (paragraphe 3.1.2.1) est que l'ensemble des composantes de l'équation s'appliquent dans l'espace opérationnel en un seul point du robot. Il est possible alors de définir un projecteur orthogonal de la même manière que le cas cinématique (voir section 2.54). L'équation de commande avec deux tâches est définie par :

$$\tau = J_1^T \Lambda_1 (w_1(t) + b_1) + N_1^T z_{\tau_2} \quad (3.14)$$

où $b_1 = -\dot{J}_1\dot{q} + J_1^{-1}AH$, $N_1^T = I - J_1^T \Lambda_1 J_1 A^{-1}$ et z_{τ_2} constitue n'importe quel autre vecteur de couples additionnels arbitraires. Du fait de ce projecteur dynamique, z_{τ_2} s'appliquera dans le noyau du produit $J_1^T \Lambda_1$ de la tâche principale, si celui-ci le permet. Ce qui de fait, applique un ordre hiérarchique entre les deux tâches, la première étant la plus prioritaire.

3.1.4 Asservissement visuel cinématique par vision rapide et découplage dynamique

Dahmouche et al. [2009, 2012]; Dahmouche [2010] ont proposé une commande qui se base sur un asservissement visuel et une équation de découplage dans l'espace cartésien telle que présentée à l'équation (3.8). La commande a été utilisée sur un robot parallèle portant une cible observée par une caméra fixe. Le découplage de la partie dynamique du robot fonctionne de la même manière que la commande en couples calculés dans l'espace cartésien, le vecteur d'entrées découplés est défini par l'équation suivante :

$$w(t) = L_v^+ \dot{s}^d + K_v L_v^+ (s^d - s) + K_p L_v^+ (s^d - s) \quad (3.15)$$

où s représente les coordonnées sur le plan focal de l'image des points saillants de la cible attachée à l'effecteur et L_v , L_v^+ des matrices d'interaction telles que définies dans le paragraphe 2.1.1.

Après étude, il s'avère que pour obtenir l'expression des matrices d'interaction L_v et L_v^+ , les auteurs se sont basés sur l'hypothèse que le mouvement du système était régi par un modèle

d'évolution dit à vitesses et à accélérations constantes. Poser un *a priori* sur l'évolution du système entraîne que les deux matrices d'interaction sont liées telles que :

$$\mathbf{L}_{\dot{\mathbf{v}}} = \Delta t \mathbf{L}_{\mathbf{v}} \quad (3.16)$$

où Δt représente l'intervalle de temps du système échantillonné. Les hypothèses prises ici font que la loi de commande devient dépendante de la fréquence d'acquisition du signal. De fait, son comportement peut changer selon la nature du capteur utilisé.

3.1.5 Conclusions

Cette section avait pour but de faire une liste non exhaustive des lois de commande utilisant des informations provenant de capteur d'effort. En premier, les lois de commande revues ont été la commande par couple calculé dans l'espace articulaire et ses différentes formulations dans l'espace cartésien puis dans l'espace opérationnel. Nous nous sommes ensuite intéressés à une nouvelle formulation de la commande par couple calculé orientée multi-tâches de manière à faire coexister plusieurs points de contrôle en utilisant un projecteur orthogonal dynamique.

À la lecture de l'état de l'art, il apparaît que seul [Dahmouche \[2010\]](#) a pris en compte des informations provenant de capteurs proprioceptifs et a utilisé celles-ci avec une équation de découplage pour en faire une commande dynamique.

À l'inverse, les autres commandes ont recours à un modèle du second ordre où les mesures de position $\mathbf{x}$ et de vitesse $\dot{\mathbf{x}}$ (informations cartésiennes par exemple) sont exploitées pour déterminer le vecteur d'entrées découplées. Cependant, ces quantités sont souvent exprimées à l'aide du modèle du robot ainsi que les informations proprioceptives.

Notre formulation de la commande dynamique référencée capteurs se présente comme une alternative à ce que propose les travaux de [\[Dahmouche, 2010\]](#) et utilise le modèle de projection du second ordre d'une caractéristique observée depuis un capteur quelconque afin d'obtenir de meilleures performances et un contrôle fin du comportement de convergence dans l'espace capteur.

3.2 Commande dynamique référencée capteurs

Cette section développe le concept de commande dynamique référencée capteurs qui utilise un modèle d'évolution du second ordre des caractéristiques observées en fonction de l'attitude générale du capteur. Le terme dynamique vient de l'idée de réutiliser les principes de la commande en couples calculés pour générer des consignes d'effort.

La première étape pose le concept du modèle du second ordre. Le modèle du second ordre entraîne la dérivation de la matrice d'interaction. Quelques exemples de calcul du modèle du second ordre concernant différentes caractéristiques géométriques seront exposés dans cette section. Puis une équation de découplage sera proposée pour linéariser le comportement des caractéristiques dans l'espace capteurs. Deux formulations sont possibles.

3.2.1 Obtention du modèle cinématique d'ordre 2

Le modèle cinématique est défini par le modèle du premier ordre qui lie l'évolution des caractéristiques observées en fonction des mouvements du capteur. On considère ici le cas où l'objet est fixe dans la scène et le capteur est mobile. Le modèle du premier ordre a déjà été défini dans le paragraphe 2.1.1, par l'équation (2.1) :

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v}$$

où les trois termes correspondent respectivement à la variation des caractéristiques, la matrice d'interaction associée et le torseur cinématique du repère de la caméra. Après dérivation, on obtient la relation suivante.

$$\ddot{\mathbf{s}} = \mathbf{L}_s \dot{\mathbf{v}} + \mathbf{H}_s \mathbf{v} \quad (3.17)$$

Le second terme de l'équation (3.17) introduit le hessien de la matrice d'interaction $\mathbf{H}_s$. On s'attend à ce que le hessien ait une forme similaire mais non égale à la dérivée de la matrice d'interaction, soit $\mathbf{H}_s \neq \dot{\mathbf{L}}_s$.

L'expression de l'accélération des caractéristiques dans l'espace capteurs en fonction des variables articulaires est aussi possible. En dérivant l'équation (2.4) $\dot{\mathbf{s}} = \mathbf{L}_s^S \mathbb{T}_N \mathbf{J}$, l'expression devient :

$$\ddot{\mathbf{s}} = \mathbf{L}_s^S \mathbb{T}_N \mathbf{J} \ddot{\mathbf{q}} + \mathbf{H}_s^S \mathbb{T}_N \mathbf{J} \dot{\mathbf{q}} + \mathbf{L}_s^S \dot{\mathbb{T}}_N \mathbf{J} \dot{\mathbf{q}} + \mathbf{L}_s^S \mathbb{T}_N \dot{\mathbf{J}} \dot{\mathbf{q}} \quad (3.18)$$

Le capteur est fixe par rapport à l'effecteur donc $\mathbb{T}_N^S = \mathbf{0}$. Néanmoins, il est tout à fait possible d'exprimer cette relation dans un cas où le capteur ne serait pas solidaire de l'effecteur. Il serait nécessaire alors de déterminer le mouvement du capteur par rapport à l'effecteur afin d'obtenir l'expression de $\mathbb{T}_N^S$.

En posant $\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{s}, \dot{\mathbf{s}}) = \mathbf{H}_s^S \mathbb{T}_N \mathbf{J} \dot{\mathbf{q}} + \mathbf{L}_s^S \mathbb{T}_N \dot{\mathbf{J}} \dot{\mathbf{q}}$, la relation qui lie l'accélération des caractéristiques observées aux accélérations articulaires est telle que :

$$\ddot{\mathbf{s}} = \mathbf{L}_s^S \mathbb{T}_N \mathbf{J} \ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{s}, \dot{\mathbf{s}}) \quad (3.19)$$

3.2.2 Matrices d'interaction pour le modèle du second ordre

On s'intéresse ici à l'obtention des matrices d'interaction lorsque le modèle du second ordre (3.17) est utilisé. La question porte principalement sur l'obtention du hessien de la matrice d'interaction $\mathbf{H}_s$ pour différentes caractéristiques.

En premier lieu seront développées la variation et sa dérivée d'un point 3D observé depuis un repère mobile. On identifiera alors quelle sera la forme de la matrice $\mathbf{H}_s$. Seront présentées ensuite les matrices d'interaction pour les caractéristiques suivantes : lignes 3D, plans 3D, points 2D et enfin lignes 2D. Les démonstrations pour ces caractéristiques sont renvoyées en annexe A.

3.2.2.1 Accélération d'un point 3D par rapport à un repère mobile

Soient P un point 3D, W et S deux repères respectivement fixe et mobile dans l'espace, le vecteur $\overrightarrow{O_W P}$ peut s'écrire dans le repère W telle que :

$$\overrightarrow{O_W P}|_W = \overrightarrow{O_W O_S}|_W + \overrightarrow{O_S P}|_W \quad (3.20)$$

Les points O_W et O_S sont respectivement les origines des repères W et S. L'équation (3.20) est dérivée une première fois, on obtient alors :

$$\frac{d}{dt} \overrightarrow{O_W P}|_W = \frac{d}{dt} \overrightarrow{O_W O_S}|_W + \frac{d}{dt} \overrightarrow{O_S P}|_W \quad (3.21)$$

Il est possible d'y faire apparaître la vitesse relative du point P par rapport à S ainsi que la vitesse d'entraînement du repère S, par l'application de la loi de composition des vitesses de la mécanique générale :

$$\begin{aligned} \frac{d}{dt} \overrightarrow{O_W P}|_W &= \frac{d}{dt} \overrightarrow{O_W O_S}|_W + \frac{d}{dt} \overrightarrow{O_S P}|_S + \Omega_{S/W}|_W \times \overrightarrow{O_S P}|_W \\ &= \frac{d}{dt} \overrightarrow{O_W O_S}|_W + \Omega_{S/W}|_W \times \overrightarrow{O_S P}|_W + \frac{d}{dt} \overrightarrow{O_S P}|_S \end{aligned} \quad (3.22)$$

De même, l'équation (3.22) est dérivée une nouvelle fois pour obtenir l'expression de l'accélération de P dans W :

$$\begin{aligned} \frac{d^2}{dt^2} \overrightarrow{O_W P}|_W &= \frac{d^2}{dt^2} \overrightarrow{O_W O_S}|_W + \frac{d}{dt} \Omega_{S/W}|_W \times \overrightarrow{O_S P}|_W \\ &\quad + \Omega_{S/W}|_W \times \frac{d}{dt} \overrightarrow{O_S P}|_W + \frac{d}{dt} \left(\frac{d}{dt} \overrightarrow{O_S P}|_S \right) \Big|_W \\ &= \frac{d^2}{dt^2} \overrightarrow{O_W O_S}|_W + \frac{d}{dt} \Omega_{S/W}|_W \times \overrightarrow{O_S P}|_W \\ &\quad + \Omega_{S/W}|_W \times \left(\frac{d}{dt} \overrightarrow{O_S P}|_S + \Omega_{S/W}|_W \times \overrightarrow{O_S P}|_W \right) \\ &\quad + \frac{d^2}{dt^2} \overrightarrow{O_S P}|_S + \Omega_{S/W}|_W \times \frac{d}{dt} \overrightarrow{O_S P}|_S \\ &= \frac{d^2}{dt^2} \overrightarrow{O_W O_S}|_W + \frac{d}{dt} \Omega_{S/W}|_W \times \overrightarrow{O_S P}|_W + \Omega_{S/W}|_W \times \left(\Omega_{S/W}|_W \times \overrightarrow{O_S P}|_W \right) \\ &\quad + \frac{d^2}{dt^2} \overrightarrow{O_S P}|_S + 2 \Omega_{S/W}|_W \times \frac{d}{dt} \overrightarrow{O_S P}|_S \end{aligned} \quad (3.23)$$

On reconnaît dans l'équation (3.23) :

- $\frac{d^2}{dt^2} \overrightarrow{O_W P}|_W$ l'accélération absolue,
- $\frac{d^2}{dt^2} \overrightarrow{O_W O_S}|_W + \frac{d}{dt} \Omega_{S/W}|_W \times \overrightarrow{O_S P}|_W + \Omega_{S/W}|_W \times \left(\Omega_{S/W}|_W \times \overrightarrow{O_S P}|_W \right)$ l'accélération d'entraînement,
- $\frac{d^2}{dt^2} \overrightarrow{O_S P}|_S$ l'accélération relative et

- $2 \Omega_{S/W}|_W \times \frac{d}{dt} \overrightarrow{O_S P}|_S$ l'accélération de Coriolis, aussi nommée accélération complémentaire.

On exprime l'accélération relative de P dans S telle que :

$$\begin{aligned} \frac{d^2}{dt^2} \overrightarrow{O_S P}|_S = & -\frac{d^2}{dt^2} \overrightarrow{O_W O_S}|_W - \frac{d}{dt} \Omega_{S/W}|_W \times \overrightarrow{O_S P}|_W \\ & - \Omega_{S/W}|_W \times \left(\Omega_{S/W}|_W \times \overrightarrow{O_S P}|_W \right) - 2 \Omega_{S/W}|_W \times \frac{d}{dt} \overrightarrow{O_S P}|_S \\ & + \frac{d^2}{dt^2} \overrightarrow{O_W P}|_W \end{aligned} \quad (3.24)$$

Les équations précédentes sont vectorielles. On s'intéresse maintenant aux valeurs de ces vecteurs : celles-ci doivent être exprimées dans le repère mobile en prémultipliant la partie droite par la matrice de rotation ${}^S R_W$. De fait,

$$\begin{aligned} \frac{d^2}{dt^2} O_S P|_S = {}^S R_W \left(-\frac{d^2}{dt^2} O_W O_S|_W - \frac{d}{dt} \Omega_{S/W}|_W \times O_S P|_W \right. \\ \left. - \Omega_{S/W}|_W \times \left(\Omega_{S/W}|_W \times O_S P|_W \right) - 2 \Omega_{S/W}|_W \times \frac{d}{dt} O_S P|_S \right. \\ \left. + \frac{d^2}{dt^2} O_W P|_W \right) \\ = -\frac{d^2}{dt^2} O_W O_S|_S - \frac{d}{dt} \Omega_{S/W}|_S \times O_S P|_S \\ - \Omega_{S/W}|_S \times \left(\Omega_{S/W}|_S \times O_S P|_S \right) - 2 \Omega_{S/W}|_S \times \frac{d}{dt} O_S P|_S \\ + \frac{d^2}{dt^2} O_W P|_S \end{aligned} \quad (3.25)$$

On reconnaît alors :

- l'accélération linéaire du repère S par rapport à W, exprimée dans S telle que ${}^S \dot{v}_{S/W} = \dot{v} = \frac{d^2}{dt^2} O_W O_S|_S$,
- l'accélération angulaire ${}^S \dot{\omega}_{S/W} = \dot{\omega} = \frac{d}{dt} \Omega_{S/W}|_S$,
- la mesure du point 3D dans le repère capteur telle que $s = O_S P|_S$,
- sa dérivée $\dot{s} = \frac{d}{dt} O_S P|_S$,
- et sa dérivée seconde $\ddot{s} = \frac{d^2}{dt^2} O_S P|_S$.

Il est possible de réécrire l'équation précédente de la manière suivante :

$$\begin{aligned} \ddot{s} = & -\dot{v} + s \times \dot{\omega} + (2 \dot{s} + \omega \times s) \times \omega + \frac{d^2}{dt^2} O_W P|_S \\ = & -\dot{v} + [s]_{\times} \dot{\omega} + [2 \dot{s} + \omega \times s]_{\times} \omega + \frac{d^2}{dt^2} O_W P|_S \end{aligned} \quad (3.26)$$

soit, en notation matricielle sachant que $\mathbf{v} = [v^\top \omega^\top]^\top$ représente le torseur cinématique du repère S exprimé dans S ,

$$\ddot{\mathbf{s}} = \begin{bmatrix} -\mathbf{I} & [\mathbf{s}]_\times \end{bmatrix} \dot{\mathbf{v}} + \begin{bmatrix} \mathbf{0} & [2\dot{\mathbf{s}} + \omega \times \mathbf{s}]_\times \end{bmatrix} \mathbf{v} + \frac{d^2}{dt^2} O_W P|_S \quad (3.27)$$

Si l'on considère que le point P est fixe dans l'espace, $\frac{d^2}{dt^2} O_W P|_S$ est alors nul. Par rapport à la relation (3.17), il est ainsi possible de reconnaître dans la relation (3.27) la matrice d'interaction et le hessien telles que :

$$\mathbf{L}_s = \begin{bmatrix} -\mathbf{I} & [\mathbf{s}]_\times \end{bmatrix} \quad (3.28a)$$

$$\mathbf{H}_s = \begin{bmatrix} \mathbf{0} & [2\dot{\mathbf{s}} + \omega \times \mathbf{s}]_\times \end{bmatrix} \quad (3.28b)$$

Conformément aux attentes exprimées dans le paragraphe précédent, on note que $\mathbf{H}_s \neq \dot{\mathbf{L}}_s$. Le hessien $\mathbf{H}_s$ est effectivement différent de la dérivée temporelle de la matrice d'interaction $\mathbf{L}_s$.

3.2.2.2 Matrice d'interaction pour les autres caractéristiques

Un travail similaire à celui présenté dans le paragraphe précédent peut être effectué pour différentes natures de caractéristiques. Ci-après sont présentées les matrices d'interaction et leur dérivées respectives pour les lignes 3D selon les coordonnées de Plücker, les plans 3D selon leurs équations cartésiennes, les points 2D ainsi que les lignes 2D sur le plan focal d'une caméra.

Pour chacune des caractéristiques, l'expression de la seconde dérivée sur le modèle du point 3D dans le paragraphe précédent peut être retrouvée en annexe A.

Ligne 3D Soit une ligne dans l'espace paramétrée par les coordonnées de Plücker, le vecteur $\mathbf{s}_\mathcal{L}$ qui décrit la caractéristique est tel que :

$$\mathbf{s}_\mathcal{L} = \begin{bmatrix} \underline{\mathbf{u}} \\ \mathbf{h} \end{bmatrix} \quad (3.29)$$

où $\underline{\mathbf{u}}$ est le vecteur directeur normalisé qui supporte la droite $\mathcal{L}$, $\mathbf{h}$ le vecteur normal au plan formé par $\mathcal{L}$ et l'origine du repère du capteur. L'expression de sa matrice d'interaction peut être retrouvée sur la base des développements effectués dans [Espiau and Rives, 1987]. On la donne ici :

$$\mathbf{L}_{\mathbf{s}_\mathcal{L}} = \begin{bmatrix} [\underline{\mathbf{u}}]_\times & [\mathbf{h}]_\times \\ \mathbf{0} & [\underline{\mathbf{u}}]_\times \end{bmatrix} \quad (3.30)$$

De la même manière que pour le point 3D, la double dérivée des coordonnées de Plücker (cf section A.2.2) fait apparaître le hessien au sein du modèle du second ordre :

$$\mathbf{H}_{\mathbf{s}_\mathcal{L}} = \begin{bmatrix} [2\dot{\underline{\mathbf{u}}}]_\times & [2\dot{\mathbf{h}} + \omega \times \mathbf{h}]_\times \\ \mathbf{0} & [2\dot{\underline{\mathbf{u}}} + \omega \times \underline{\mathbf{u}}]_\times \end{bmatrix} \quad (3.31)$$

Plan 3D Soit un plan $\mathcal{P}$ défini par les paramètres de son équation cartésienne, le vecteur $\mathbf{s}_{\mathcal{P}}$ qui décrit la caractéristique est tel que :

$$\mathbf{s}_{\mathcal{P}} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} \mathbf{n} \\ d \end{bmatrix} \quad (3.32)$$

$\mathbf{n}$ constitue le vecteur normal au plan. En s'inspirant d'[Espiau and Rives \[1987\]](#), on détermine la matrice d'interaction d'un plan ainsi que le hessien (cf. section [A.3.2](#)). Les deux matrices sont définies respectivement par :

$$\mathbf{L}_{\mathcal{P}} = \begin{bmatrix} \mathbf{0} & [\mathbf{n}]_{\times} \\ \mathbf{n}^{\top} & \mathbf{0} \end{bmatrix} \quad (3.33)$$

et

$$\mathbf{H}_{\mathcal{P}} = \begin{bmatrix} \mathbf{0} & [2\dot{\mathbf{n}} + \omega \times \mathbf{n}]_{\times} \\ 2\dot{\mathbf{n}}^{\top} & (v \times \mathbf{n})^{\top} \end{bmatrix} \quad (3.34)$$

Point 2D normalisé Soit un point quelconque de l'espace d'équation $\mathbf{x} = [x \ y \ z]^{\top}$, la projection de ce point sur le plan focal d'une caméra tenant compte d'un modèle de caméra sténopé a été déjà définie dans la section [2.1.4.2](#). On rappelle ici l'équation [\(2.49\)](#) :

$$\mathbf{s}_{\mathbf{x}_n} = \text{proj}_{3D \rightarrow 2D}(\mathbf{x}) = \frac{1}{z} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix}$$

Selon la démonstration en section [A.4.2](#), on fait apparaître dans le modèle du second ordre la matrice d'interaction et le hessien correspondant. La matrice d'interaction d'un point 2D est donnée en équation [\(2.37\)](#) :

$$\mathbf{L}_{\mathbf{x}_n} = \frac{\partial \mathbf{x}_n}{\partial \mathbf{x}} \mathbf{L}_{\mathbf{x}} = \begin{bmatrix} -\frac{1}{z} & 0 & \frac{x_n}{z} & x_n y_n & -1 - x_n^2 & y_n \\ 0 & -\frac{1}{z} & \frac{y_n}{z} & 1 + y_n^2 & -x_n y_n & -x_n \end{bmatrix}$$

Les quantités $\mathbf{L}_{\mathbf{x}}$ et $\mathbf{H}_{\mathbf{x}}$ sont exprimées respectivement dans la section précédente en équations [\(3.28a\)](#) et [\(3.28b\)](#). Le hessien est explicité en utilisant les composantes du modèle du second ordre

du point 3D telle que :

$$\mathbf{H}_{\mathbf{x}_n} = \frac{d}{dt} \frac{\partial \mathbf{x}_n}{\partial \mathbf{x}} \mathbf{L}_{\mathbf{x}} + \frac{\partial \mathbf{x}_n}{\partial \mathbf{x}} \mathbf{H}_{\mathbf{x}}$$

$$\mathbf{H}_{\mathbf{x}_n} = \begin{bmatrix} \frac{\dot{z}}{z^2} & 0 & -\frac{\dot{z}}{z^2} x_n + \frac{\dot{x}_n}{z} \\ 0 & \frac{\dot{z}}{z^2} & -\frac{\dot{z}}{z^2} y_n + \frac{\dot{y}_n}{z} \end{bmatrix}$$

$$\begin{pmatrix} 2\frac{\dot{y}}{z} - y_n\frac{\dot{z}}{z} + \omega_3 x_n - \omega_1 \\ (1 + y_n^2)\frac{\dot{z}}{z} + \left(\dot{y}_n + 2\frac{\dot{y}}{z}\right) y_n + (\omega_3 y_n - \omega_2) \end{pmatrix} x_n + y_n \dot{x}_n \quad \begin{pmatrix} -(1 - x_n^2)\frac{\dot{z}}{z} + \left(-2\frac{\dot{x}}{z} - \dot{x}_n\right) x_n + (\omega_1 + \omega_3 x_n) y_n \\ \left(x_n\frac{\dot{z}}{z} - 2\frac{\dot{x}}{z} + \omega_3 y_n - \omega_2\right) y_n - x_n \dot{y}_n \end{pmatrix}$$

$$\begin{bmatrix} -y_n\frac{\dot{z}}{z} + 2\frac{\dot{y}}{z} + \omega_3 x_n - \omega_1 \\ x_n\frac{\dot{z}}{z} - \frac{2\dot{x}}{z} + \omega_3 y_n + \omega_2 \end{bmatrix} \quad (3.35)$$

Ligne 2D Soit une ligne 2D paramétrée selon les coordonnées polaires sur le plan focal, le vecteur $\mathbf{s}_{\mathcal{L}_{2D}}$ qui décrit la caractéristique est composé de la façon suivante [Chaumette \[1990\]](#) :

$$\mathbf{s}_{\mathcal{L}_{2D}} = \begin{bmatrix} \rho \\ \theta \end{bmatrix} \quad (3.36)$$

où ρ représente la distance orthogonale entre le centre du plan focal (point principal) et sa projection sur la droite et θ l'angle entre le vecteur unitaire selon x_n et la direction de la droite $\mathcal{L}_{2D}$ sur le plan focal. La relation de passage entre la représentation cartésienne du point 2D normalisé et les coordonnées polaires s'effectue grâce à l'équation suivante :

$$\rho = x_n \cos(\theta) + y_n \sin(\theta) \quad (3.37)$$

Selon la démonstration en annexe, section [A.5.3](#), il s'avère que l'expression de l'accélération des coordonnées polaires $[\ddot{\rho} \ddot{\theta}]^T$ est fonction de nombreux paramètres à la fois 2D et 3D. Cette expression ne permet pas de dégager une matrice d'interaction et son hessien correspondant d'une manière claire en évitant les dépendances inutiles. On ne peut donc pas utiliser la ligne 2D en tant que caractéristique pour une commande dynamique en utilisant cette paramétrisation.

Il existe une autre possibilité de formaliser le modèle du second ordre pour la ligne 2D en se basant cette fois des coordonnées de Plücker. Mais cette option n'a pas encore été étudiée.

3.2.3 Commande

Il existe plusieurs possibilités pour la formulation de la commande dynamique référencée capteurs. Dans un premier temps, seul le capteur sera considéré. Ensuite, la commande sera exprimée tenant compte d'un capteur fixé rigidement sur l'effecteur d'un robot.

3.2.3.1 Capteur seul

À partir de l'équation (3.17) qui lie l'accélération d'une caractéristique (image d'une caractéristique physique d'un objet fixe dans la scène) à l'accélération et à la vitesse du repère mobile

associé au capteur, il est possible d'écrire :

$$\dot{\mathbf{v}} = \mathbf{L}_s^+ (\ddot{\mathbf{s}} - \mathbf{H}_s \mathbf{v}) \quad (3.38)$$

où l'opérateur $(\cdot)^+$ dénote ici la pseudo-inverse.

On suppose qu'il est possible d'agir sur la position du capteur en utilisant les six composantes d'accélération. Dès lors en s'inspirant de la commande en couples calculés présentée dans les paragraphes précédents, on pose :

$$\dot{\mathbf{v}} = \mathbf{L}_s^+ (\mathbf{w}(t) - \mathbf{H}_s \mathbf{v}) \quad (3.39)$$

avec $\mathbf{w}(t)$ le vecteur d'entrées découplées. L'équation précédente constitue de fait une équation de découplage qui permet de passer de l'espace de l'image vers l'accélération du repère mobile. On définit le vecteur $\mathbf{w}(t)$ tel que :

$$\mathbf{w}(t) = \ddot{\mathbf{s}}^d + \mathbf{K}_{sv} (\dot{\mathbf{s}}^d - \dot{\mathbf{s}}) + \mathbf{K}_{sp} (\mathbf{s}^d - \mathbf{s}) \quad (3.40)$$

$\mathbf{K}_{sv}, \mathbf{K}_{sp} \in \mathbb{R}^{m \times m}$ sont des matrices diagonales définies positives.

On note que la mesure de la variation $\dot{\mathbf{s}}$ est requise pour réaliser ce correcteur. Ceci constitue une hypothèse majeure du travail concernant l'expression du modèle cinématique du second ordre. On considère que le capteur est en capacité de fournir la variation du signal $\dot{\mathbf{s}}$ de la même manière que $\mathbf{s}$ et ce, de façon synchronisée.

Cette affirmation n'est pas vérifiée en pratique : en effet, les capteurs d'informations spatiales se contentent que de fournir le signal utile, sa variation étant déduite après coup par dérivation numérique. Néanmoins, cette dérivation obtenue par différenciation entre deux états du signal $\mathbf{s}$ à des instants différents, ne peut être considérée que comme une approximation eulérienne de $\dot{\mathbf{s}}$, plus ou moins fidèle à la réalité et fonction de la fréquence d'acquisition du signal.

L'obtention du signal $\dot{\mathbf{s}}$ de pair avec le signal $\mathbf{s}$ doit nécessairement passer par la conception de nouvelles méthodes de mesure de phénomène physique et des nouvelles architectures capteurs. Un exemple d'architecture innovante pour un capteur de vision dynamique peut être trouvé dans les travaux suivants : [Lichtsteiner et al., 2008; Posch et al., 2011; Berner et al., 2013; Brandli et al., 2014].

Stabilité On démontre la stabilité du système commandé via les équations (3.39) et (3.40) par le système d'équations différentielles du système en boucle fermée ci-dessous. On dénote les termes marqués par $\hat{\cdot}$ les quantités estimées pour le calcul de l'équation de découplage de la loi de commande.

$$\ddot{\mathbf{e}} + \mathbf{K}_{sv} \dot{\mathbf{e}} + \mathbf{K}_{sp} \mathbf{e} = \hat{\mathbf{L}}_s \left((\mathbf{L}_s^+ - \hat{\mathbf{L}}_s^+) \ddot{\mathbf{s}} + \hat{\mathbf{L}}_s^+ \hat{\mathbf{H}}_s \hat{\mathbf{v}} - \mathbf{L}_s^+ \mathbf{H}_s \mathbf{v} \right) \quad (3.41)$$

où $\mathbf{e} = \mathbf{s}^d - \mathbf{s}$. Si les matrices estimées correspondent aux matrices réelles, le second terme à droite de l'équation s'annule. La réponse temporelle $\mathbf{e}(t)$ en boucle fermée est globalement exponentiellement stable dans l'espace capteur. Les gains $\mathbf{K}_{sv}$ et $\mathbf{K}_{sp}$ sont choisis de manière à régler le comportement de convergence du système complet (temps de réponse, dépassement, etc.). Dans le cas contraire, une excitation apparaît. Les gains peuvent alors être choisis de manière à limiter l'impact de cette excitation sur le comportement du système. Le bruit de mesure, par exemple, issu de la méthode et/ou de l'appareil de mesure, peut être à l'origine de ce genre d'excitation.

3.2.3.2 Avec un robot : formulation en couples calculés

En considérant un capteur monté sur l'effecteur du robot, deux formulations sont alors possibles. La première formulation se base sur l'équation (3.19) où les entrées de commande du robot sont ses couples articulaires. De la même manière que dans le paragraphe précédent, on inverse l'équation (3.19) pour obtenir :

$$\ddot{\mathbf{q}} = (\mathbf{L}_s^S \mathbb{T}_N \mathbf{J})^+ (\ddot{\mathbf{s}} - \mathbf{b}) \quad (3.42)$$

L'expression va être injectée dans l'équation de découplage du modèle dynamique (3.3). On obtient ainsi les couples articulaires en fonction des composantes d'accélération des caractéristiques géométriques dans l'espace des mesures du capteur. Pour une meilleure lisibilité, on pose $\mathbb{J} = \mathbf{L}_s^S \mathbb{T}_N \mathbf{J}$. La quantité $\mathbf{b}$ est définie par l'équation (3.19).

$$\boldsymbol{\tau} = \mathbf{A} \mathbb{J}^+ (\ddot{\mathbf{s}} - \mathbf{b}) + \mathbf{H} \quad (3.43)$$

L'équation nous permet d'obtenir une nouvelle équation de découplage qui permet de passer de l'espace des mesures aux couples articulaires du robot :

$$\boldsymbol{\tau} = \mathbf{A} \mathbb{J}^+ (\mathbf{w}(t) - \mathbf{b}) + \mathbf{H} \quad (3.44)$$

$\mathbf{w}(t)$ est choisi de la même manière qu'avec le capteur seul en utilisant l'équation (3.40).

Pour une comparaison moins formelle de la commande du robot, la figure 3.4 montre le schéma bloc correspondant à cette loi de commande.

Stabilité La stabilité de la commande formée par les équation (3.44) et (3.40) est démontrée ci-dessous. L'équation de mouvement qui régit le système en boucle fermée est donnée par la relation suivante :

$$\hat{\mathbf{A}} \hat{\mathbb{J}}^+ (\ddot{\mathbf{s}}^d + \mathbf{K}_{sv} \dot{\mathbf{e}} + \mathbf{K}_{sp} \mathbf{e} - \hat{\mathbf{b}}) + \hat{\mathbf{H}} = \mathbf{A} \ddot{\mathbf{q}} + \mathbf{H} \quad (3.45)$$

En tenant compte de (3.42), l'équation précédente devient :

$$\hat{\mathbf{A}} \hat{\mathbb{J}}^+ (\ddot{\mathbf{s}}^d + \mathbf{K}_{sv} \dot{\mathbf{e}} + \mathbf{K}_{sp} \mathbf{e} - \hat{\mathbf{b}}) + \hat{\mathbf{H}} = \mathbf{A} \mathbb{J}^+ (\ddot{\mathbf{s}} - \mathbf{b}) + \mathbf{H} \quad (3.46)$$

$$\ddot{\mathbf{e}} + \mathbf{K}_{sv} \dot{\mathbf{e}} + \mathbf{K}_{sp} \mathbf{e} = \hat{\mathbb{J}} \hat{\mathbf{A}}^{-1} \left((\mathbf{A} \mathbb{J}^+ - \hat{\mathbf{A}} \hat{\mathbb{J}}^+) \ddot{\mathbf{s}} - \mathbf{A} \mathbb{J}^+ \mathbf{b} + \hat{\mathbf{A}} \hat{\mathbb{J}}^+ \hat{\mathbf{b}} + \mathbf{H} - \hat{\mathbf{H}} \right) \quad (3.47)$$

Par rapport à l'équation temporelle du système en boucle fermée pour le capteur seul, équation (3.41), on ajoute les erreurs de modélisation géométriques et dynamiques. Il est possible d'établir les mêmes conclusions : s'il n'y a pas d'erreur de modélisation (i.e. $\hat{\mathbf{A}} = \mathbf{A}$, etc.), le second terme de l'équation à droite s'annule et le système est globalement exponentiellement stable.

$$\ddot{\mathbf{e}} + \mathbf{K}_{sv} \dot{\mathbf{e}} + \mathbf{K}_{sp} \mathbf{e} = \mathbf{0} \quad (3.48)$$

Dans le cas contraire, les erreurs de modélisation constituent une excitation supplémentaire. Les gains $\mathbf{K}_{sv}$ et $\mathbf{K}_{sp}$ peuvent être ajustés pour compenser cette nouvelle excitation dans la limite des performances du système.

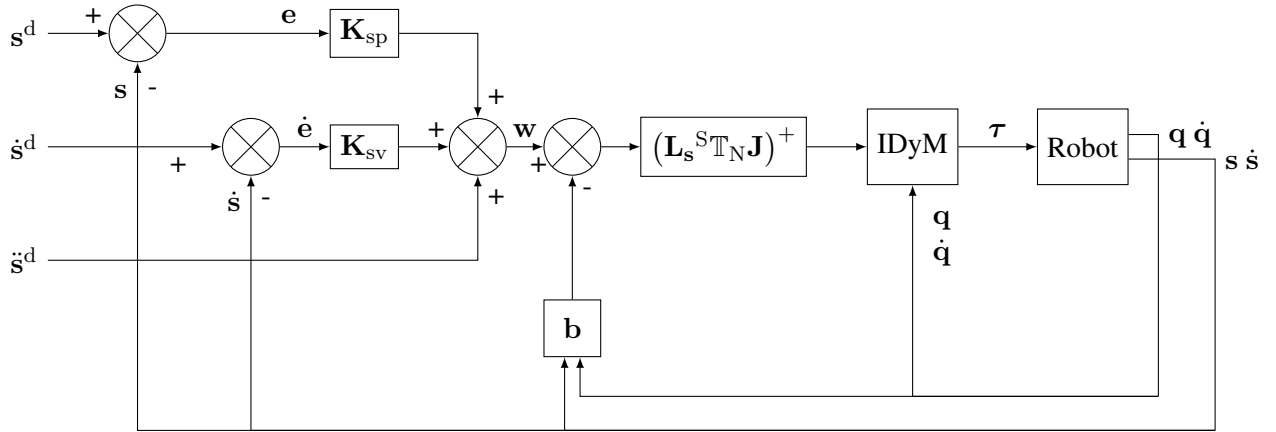


FIGURE 3.4 – Schéma bloc illustrant la commande référencée capteurs dynamique.

3.2.3.3 Avec un robot : formulation en espace opérationnel

Le modèle dynamique direct d'un robot est défini par l'équation suivante :

$$\ddot{\mathbf{q}} = \mathbf{A}^{-1}(\boldsymbol{\tau} - \mathbf{H}) \quad (3.49)$$

Les accélérations articulaires sont injectées dans l'équation (3.19), ce qui donne la nouvelle relation :

$$\ddot{\mathbf{s}} = \mathbf{L}_s^S \mathbb{T}_N \mathbf{J} \mathbf{A}^{-1}(\boldsymbol{\tau} - \mathbf{H}) + \mathbf{b} \quad (3.50)$$

$$\ddot{\mathbf{s}} = \mathbf{L}_s^S \mathbb{T}_N \mathbf{J} \mathbf{A}^{-1} \boldsymbol{\tau} - \mathbb{J} \mathbf{A}^{-1} \mathbf{H} + \mathbf{b} \quad (3.51)$$

En posant $\boldsymbol{\tau} = \mathbf{J}^T \mathbf{f}$ [Khalil and Dombre, 2002], la relation qui lie le torseur d'effort s'appliquant à l'effecteur, dans le repère de l'effecteur aux couples articulaires, on obtient :

$$\ddot{\mathbf{s}} = \mathbf{L}_s^S \mathbb{T}_N \mathbf{J} \mathbf{A}^{-1} \mathbf{J}^T \mathbf{f} - \mathbb{J} \mathbf{A}^{-1} \mathbf{H} + \mathbf{b} \quad (3.52)$$

$$\mathbf{f} = \Lambda^N \mathbb{T}_S \mathbf{L}_s^+ (\ddot{\mathbf{s}} - \mathbf{b} + \mathbb{J} \mathbf{A}^{-1} \mathbf{H}) \quad (3.53)$$

où $\Lambda = (\mathbf{J} \mathbf{A}^{-1} \mathbf{J}^T)^{-1}$ (équation (3.11a)) est la matrice d'inertie du système exprimée dans l'espace opérationnel.

$$\mathbf{f} = \Lambda^N \mathbb{T}_S \mathbf{L}_s^+ (\ddot{\mathbf{s}} - \mathbf{b} + \mathbb{J} \mathbf{A}^{-1} \mathbf{H}) \quad (3.54)$$

L'équation de découplage pour cette formulation est donc de la forme :

$$\mathbf{f} = \Lambda^N \mathbb{T}_S \mathbf{L}_s^+ (\mathbf{w}(t) - \mathbf{b} + \mathbb{J} \mathbf{A}^{-1} \mathbf{H}) \quad (3.55)$$

où $\mathbf{w}(t)$ est déjà défini en équation (3.40).

On note qu'une autre formulation est possible en exprimant la matrice d'inertie du système dans l'espace capteur et non plus dans l'espace opérationnel. Néanmoins, la matrice d'interaction doit être nécessairement d'un rang supérieur ou égal à 6 pour permettre l'inversion.

Stabilité La stabilité est équivalente à la commande en utilisant la formulation présentée dans le paragraphe précédent. Le même exercice peut être fait en tenant compte de l'équation (3.55) et (3.40). L'équation de mouvement du système en boucle fermée est présentée ci-dessous :

$$\ddot{\mathbf{e}} + \mathbf{K}_{sv}\dot{\mathbf{e}} + \mathbf{K}_{sp}\mathbf{e} = (\Lambda^N \mathbb{T}_S \mathbf{L}_s^+)^{(-1)} \left[\left(\Lambda^N \mathbb{T}_S \mathbf{L}_s^+ - \hat{\Lambda}^N \hat{\mathbb{T}}_S \hat{\mathbf{L}}_s^+ \right) \ddot{\mathbf{s}} - \Lambda^N \mathbb{T}_S \mathbf{L}_s^+ \mathbf{b} + \hat{\Lambda}^N \hat{\mathbb{T}}_S \hat{\mathbf{L}}_s^+ \hat{\mathbf{b}} + \mathbf{J}^{-\top} \mathbf{H} - \hat{\mathbf{J}}^{-\top} \hat{\mathbf{H}} \right] \quad (3.56)$$

En posant

$$\Lambda^N \mathbb{T}_S \mathbf{L}_s^+ = \mathbf{J}^{-\top} \mathbf{A} \mathbf{J}^{-1N} \mathbb{T}_S \mathbf{L}_s^+ = \mathbf{J}^{-\top} \mathbf{A} \mathbb{J}^+ \quad (3.57)$$

l'équation de mouvement en boucle fermée peut être réécrite telle que :

$$\ddot{\mathbf{e}} + \mathbf{K}_{sv}\dot{\mathbf{e}} + \mathbf{K}_{sp}\mathbf{e} = \mathbb{J} \mathbf{A}^{-1} \mathbf{J}^{\top} \left[\left(\mathbf{J}^{-\top} \mathbf{A} \mathbb{J}^+ - \hat{\mathbf{J}}^{-\top} \hat{\mathbf{A}} \hat{\mathbb{J}}^+ \right) \ddot{\mathbf{s}} - \mathbf{J}^{-\top} \mathbf{A} \mathbb{J}^+ \mathbf{b} + \hat{\mathbf{J}}^{-\top} \hat{\mathbf{A}} \hat{\mathbb{J}}^+ \hat{\mathbf{b}} + \mathbf{J}^{-\top} \mathbf{H} - \hat{\mathbf{J}}^{-\top} \hat{\mathbf{H}} \right] \quad (3.58)$$

L'équation de stabilité pour la formulation en espace opérationnel est équivalente à celle pour la formulation en couple calculé donnée par l'équation (3.48). La différence est que le terme d'excitation si des erreurs de modélisation existent s'applique dans l'espace opérationnel et non plus dans l'espace articulaire.

3.2.3.4 Approximation du hessien

La section 3.2.2 établit le modèle de projection du second ordre pour différentes caractéristiques. Or il s'avère que pour certaines d'entre elles, l'expression du modèle et plus particulièrement, l'expression du hessien n'est pas trivial ou n'a pas de solution acceptable du point de vue de la commande référencée capteurs.

Il est possible alors d'approximer le hessien par la dérivée temporelle de la matrice d'interaction.

$$\ddot{\mathbf{s}} = \mathbf{L}_s \dot{\mathbf{v}} + \mathbf{H}_s \mathbf{v} \approx \mathbf{L}_s \dot{\mathbf{v}} + \dot{\mathbf{L}}_s \mathbf{v} \quad (3.59)$$

Du fait de la construction de la commande, utiliser l'approximation ci-dessus pour calculer l'équation de découplage aura un effet limité. Il se concentrera notamment sur la quantité $\hat{\mathbf{b}} \neq \mathbf{b}$. Au regard des équations de stabilité dans les paragraphes précédents, l'approximation entraîne un comportement identique aux erreurs de modélisation, qui sont contenues par l'ajustement des gains de l'équation de commande.

3.2.4 Critique

Plusieurs formulations de la commande référencée capteurs sont possibles. Néanmoins, celles-ci sont difficilement applicables en pratique sur des systèmes réels, en particulier si un robot industriel est utilisé. Plusieurs raisons justifient cela.

Une des raisons principales est que les capteurs extéroceptifs ont une bande passante limitée. Pour la majeure partie des capteurs, cette limitation est incompatible avec les performances exigées pour le contrôle d'un robot avec une loi de commande en couples calculés. De plus, le problème est accentué dès lors qu'un traitement supplémentaire est requis pour extraire les caractéristiques désirées, nécessaires à l'application de la loi de commande référencée capteurs dynamique.

Peu de capteurs sont capables de fournir l'information de variation (vitesse) des caractéristiques. Un mauvais réflexe serait de différencier les mesures obtenues à deux instants différents du

traitement car celui-ci entraînerait de nombreuses approximations dont les conséquences peuvent être problématiques pour le système dans son ensemble. De nouvelles architectures de capteurs ont fait leur apparition récemment, avec une bande passante qui correspondrait aux performances dont la loi de commande aurait besoin.

3.3 Validation de la commande par simulation

Le but de cette section est de proposer une série de simulations pour permettre d'évaluer le potentiel de la commande dynamique référencée capteurs telle que proposée dans la section précédente.

En premier lieu, la validation commencera par la réalisation d'un suivi de trajectoire. Les simulations concerneront la robustesse de la commande par rapport au bruit de mesure. Ensuite, nous allons évaluer la robustesse par rapport aux singularités des robots et l'incidence des écarts éventuels aux modèles géométrique et dynamique sur la stabilité du système en boucle fermée.

Étant donné que la commande dynamique référencée capteur peut s'appliquer en utilisant n'importe quelles des cinq caractéristiques traitées auparavant, les tests de robustesse vont être effectués avec quatre points 2D mesurés à l'aide d'une caméra dont nous connaissons les paramètres intrinsèques. En effet, l'asservissement visuel basé image reste un classique de la commande référencée capteurs, ce qui fait que le point 2D est une caractéristique naturellement candidate pour ces essais.

3.3.1 Suivi de trajectoire dans l'espace capteur

La commande dynamique référencée capteurs apporte deux éléments non négligeables qui sont le contrôle du comportement de convergence et la réalisation de trajectoire dans l'espace capteurs. En effet, du fait de l'utilisation du modèle du second ordre et de la commande par linéarisation, l'utilisateur est dans la capacité de garantir que l'erreur statique et l'erreur de traînage s'annulent le long d'une trajectoire désirée.

Certains travaux [[Mezouar and Chaumette, 2002](#)] ont déjà statué que le couplage entre une planification de trajectoire dans l'espace capteurs et la commande référencée capteur était en mesure d'augmenter le champ d'application de la commande cinématique classique et ainsi, d'étendre ses bénéfices pour des erreurs initiales larges. La commande dynamique apporte ici la garantie supplémentaire d'annuler l'erreur de traînage tout au long de cette trajectoire et ainsi obtenir de meilleures performances en terme de suivi de trajectoire par rapport à des références extéroceptives.

Nous avons simulé le système en boucle fermée suivant une trajectoire dans l'espace capteur. Comme statué par [Mezouar and Chaumette \[2002\]](#), on construit dans un premier temps la trajectoire du repère de la caméra pour ensuite en déduire la trajectoire des caractéristiques. Puisque la caméra est embarquée et sa fixation est rigide sur l'effecteur du robot, nous avons construit la trajectoire cartésienne du repère de la caméra tenant compte d'une interpolation linéaire d'ordre cinq entre deux configurations articulaires du robot. À noter qu'il est tout à fait possible de déterminer la trajectoire du repère de la caméra de manière arbitraire de même qu'il est aussi possible de le faire avec les caractéristiques. On note que la contrainte de visibilité des caractéristiques est respectée tout au long de la trajectoire.

Le mouvement se veut sans difficultés particulières. Les configurations initiale et finale sont

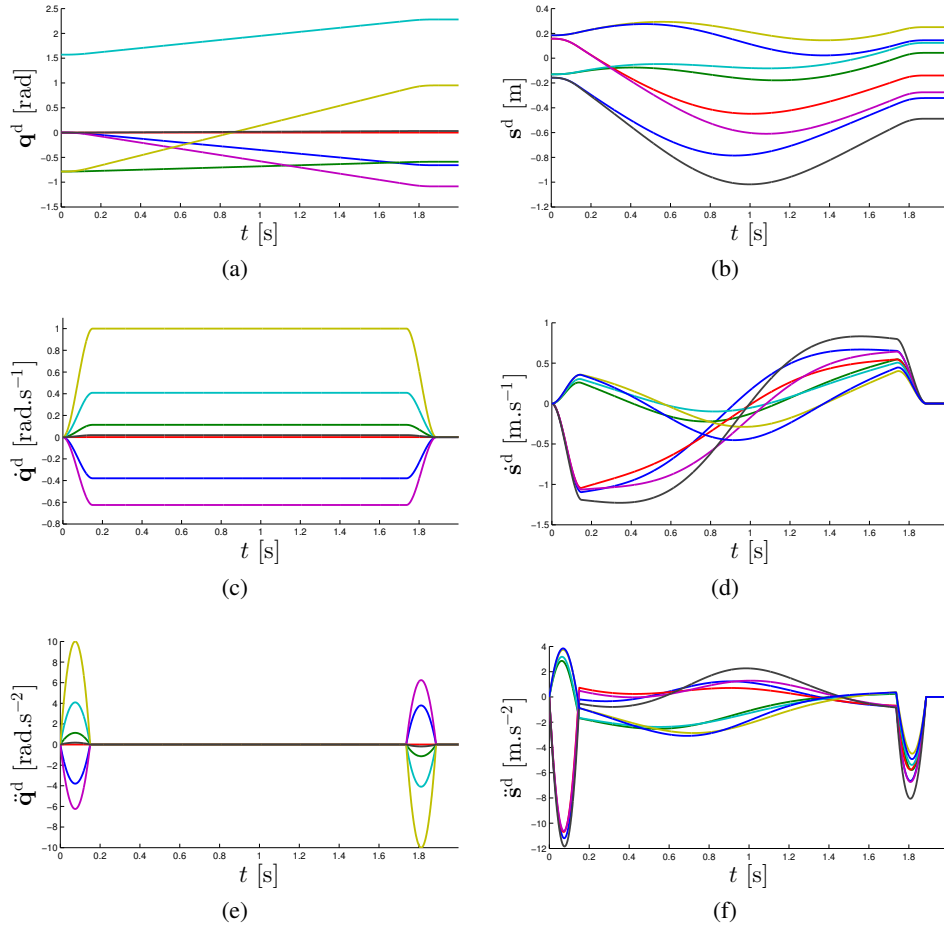


FIGURE 3.5 – Suivi de trajectoire dans l'espace capteur : (a)(c)(d) profil de trajectoire articulaire respectivement $\mathbf{q}$, $\dot{\mathbf{q}}$ et $\ddot{\mathbf{q}}$; (b)(d)(f) profil de trajectoire des caractéristiques points 2D $\mathbf{s}^d$, $\dot{\mathbf{s}}^d$ et $\ddot{\mathbf{s}}^d$, obtenues en tenant compte de la trajectoire articulaire et de la pose prédéfinie du capteur dans l'effecteur.

telles que :

$$\begin{aligned} \mathbf{q}_{\text{initiale}} &= \begin{bmatrix} 0 & -\frac{\pi}{4} & 0 & \frac{\pi}{2} & 0 & -\frac{\pi}{4} & 0 \end{bmatrix}^T \\ \mathbf{q}_{\text{finale}} &= \begin{bmatrix} -0.6583 & -0.5885 & 0 & 2.2808 & -1.0850 & 0.9512 & 0.0320 \end{bmatrix}^T \end{aligned} \quad (3.60)$$

La vitesse articulaire maximale est fixée à 1 rad.s^{-1} , l'accélération à 10 rad.s^{-2} . Les gains $\mathbf{K}_{\text{sp}}$ et $\mathbf{K}_{\text{sv}}$ sont réglés de telle sorte que le système en boucle fermée ait une réponse critique avec une constante de temps de l'ordre de 10 ms.

Les résultats de simulation sont montrés sur les figures 3.5 et 3.6. Comme mentionné précédemment, le profil de trajectoire des caractéristiques est déduit à partir du mouvement désiré du robot.

Les évolutions temporelles des erreurs dans l'espace des caractéristiques sont montrés sur les figures 3.6b et 3.6d. On distingue trois phases : les phases au début et à la fin du mouvement qui correspondent au temps d'accélération et de décélération du système ; la phase intermédiaire

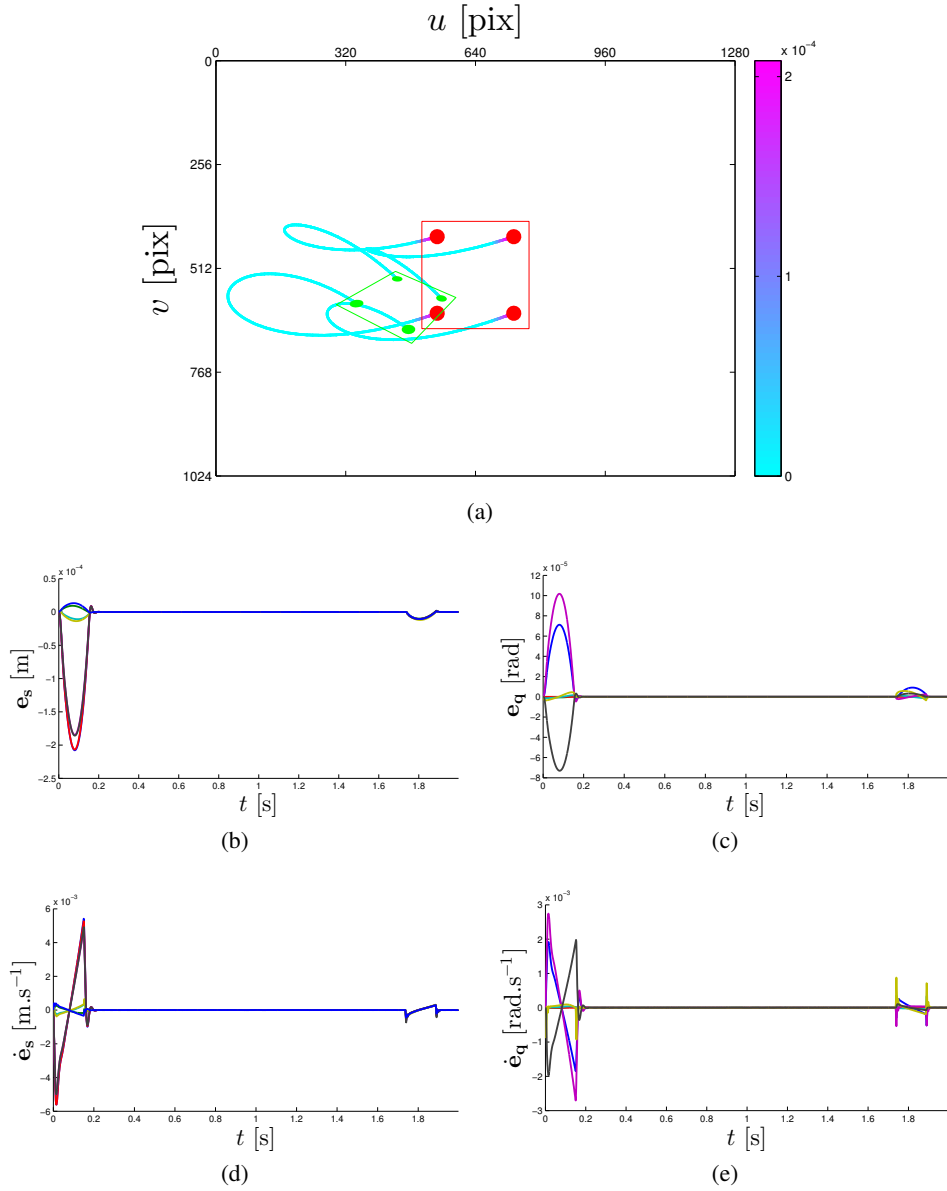


FIGURE 3.6 – Suivi de trajectoire dans l'espace capteur : (a) évolution des caractéristiques dans l'image selon le profil de trajectoire (voir figure 3.5). La couleur est fonction de la norme de l'erreur $s - s^d$. (b)(d) Évolution temporelle des erreurs dans l'espace des caractéristiques, respectivement e_s et $\dot{e}_s$. (c)(e) Évolution temporelle des erreurs dans l'espace articulaire, respectivement e_q et $\dot{e}_q$.

correspond au temps où le système évolue à vitesse articulaire constante. Lors des phases où l'accélération est non nulle, l'erreur est elle aussi non nulle. Les phases d'accélération sont trop courtes et le système en boucle fermée ne possède pas le temps nécessaire pour annuler cette erreur. En revanche, une fois que le système arrive en régime établi où la vitesse est constante, l'erreur de position et celle de traînage s'annulent. On observe aussi que les erreurs de position et de traînage en espace articulaire sont elles aussi nulles lors de la phase à vitesse constante. Le système a été simulé dans des conditions variées avec toujours les mêmes remarques de comportement, et ce quelle que soit la stratégie utilisée pour générer la trajectoire dans l'espace capteur.

En première conclusion, notre formulation pour la commande dynamique référencée capteurs permet de suivre des trajectoires définies dans l'espace capteurs avec la garantie supplémentaire d'annuler l'erreur de traînage et ainsi obtenir des performances satisfaisantes.

3.3.2 Robustesse au bruit de mesure

Une mesure est une quantité physique observée avec un instrument dédié qui mesure un signal image de l'information que l'on souhaite récupérée. Le bruit de mesure peut être défini par l'apparition de signaux parasites superposés au signal utile et qui gênent, voire empêchent totalement la compréhension de l'information que l'on souhaite extraire. Les paramètres du bruit sont très fortement liés aux caractéristiques de l'instrument, mais aussi à l'échantillonnage du signal continu, ainsi qu'à la méthode d'extraction du signal. Ce qui en fait un signal difficilement prévisible et qui lui confère le qualificatif de signal aléatoire.

Dans notre cas d'étude, le signal à asservir est constitué de quatre points 2D extraits d'une image. L'étude va évaluer la commande tenant compte d'un bruit de type uniforme d'écart type σ_{pix} allant de 0 à 3 pixels, pour une caméra ayant une résolution de 640×480 . Seuls les mouvements de la caméra sont simulés.

Le repère de la caméra effectue un cercle dans le plan XY au dessus d'une cible comportant les quatre points observés. La caméra est pointée vers le bas. L'accélération est progressive jusqu'à atteindre une vitesse tangentielle au cercle de 0.5m.s^{-1} . Les équations pour la loi de commande sont les équations (3.39) et (3.40). Les gains sont réglés pour que le système se comporte comme un système du second ordre en régime permanent avec une constante de temps de 100 ms, soit :

$$\begin{aligned} \mathbf{K}_{\text{sp}} &= 100 \\ \mathbf{K}_{\text{sv}} &= 20 \end{aligned} \tag{3.61}$$

La caméra est placée initialement dans une pose différente où elle devrait être au début de la trajectoire. Cette pose fait que la caméra est mal alignée selon ses six degrés de liberté.

Les résultats de simulation sont reportés sur les figures 3.7, 3.8 et 3.9. Le premier groupe de figures 3.7(a)-(d) montre l'évolution du système pendant sa phase de convergence sans bruit de mesure. Les trajectoires désirés de chaque caractéristique dans l'espace de l'image décrivent des cercles. À partir de la position de la cible observée depuis la position initiale de la caméra (en gris, figure 3.7a), les mesures convergent vers leurs états désirés. L'évolution temporelle des erreurs en position $\mathbf{e} = \mathbf{s} - \mathbf{s}^d$ et en vitesse $\dot{\mathbf{e}} = \dot{\mathbf{s}} - \dot{\mathbf{s}}^d$ sont représentées en figure 3.7c et 3.7d. On considère que l'erreur s'annule à $t \simeq 0.6\text{s}$. Au delà, il est possible de considérer que la caméra suit parfaitement la trajectoire demandée dans l'espace capteur. Le premier objectif du contrôleur (*i.e.* assurer un suivi de trajectoire) est rempli.

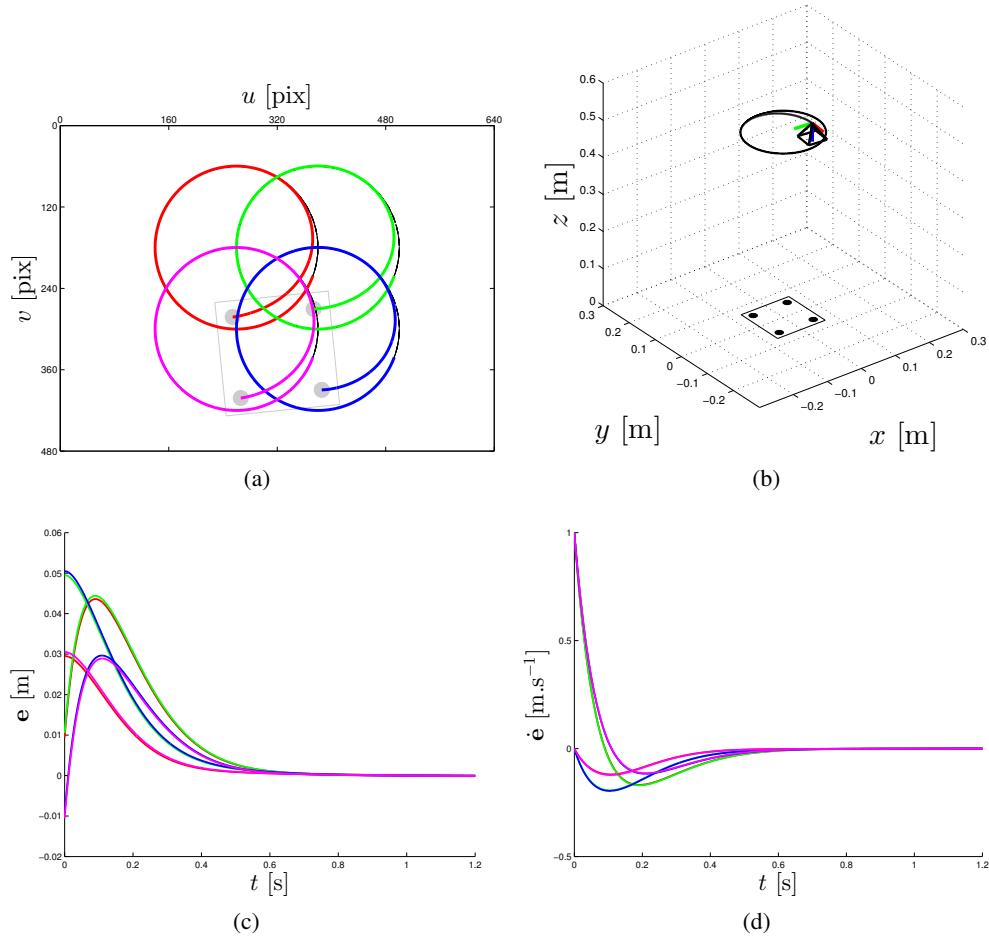


FIGURE 3.7 – Suivi de trajectoire en cercle sans bruit de mesure : (a) trajectoires désirées et courantes des caractéristiques dans l'image en pixels; (b) trajectoires désirées et courantes de l'origine du repère de la caméra; (c) et (d) évolution temporelle des erreurs en position e et vitesses $\dot{e}$ (avec $e = s - s^d$).

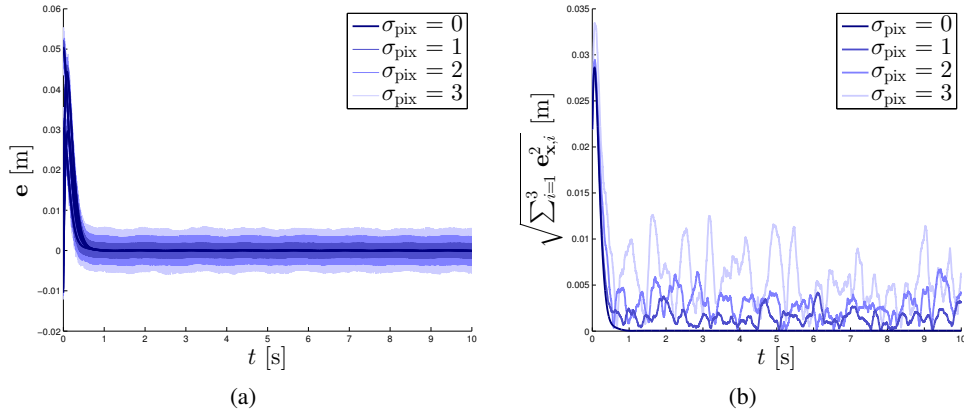


FIGURE 3.8 – Suivi de trajectoire en cercle avec bruit de mesure : (a) erreurs en position dans l'espace de l'image avec $\sigma_{\text{pix}} = \{0, 1, 2, 3\}$; (b) distance entre la position entre l'origine mesurée et l'origine désirée du repère de la caméra, selon les différentes valeurs de σ_{pix} .

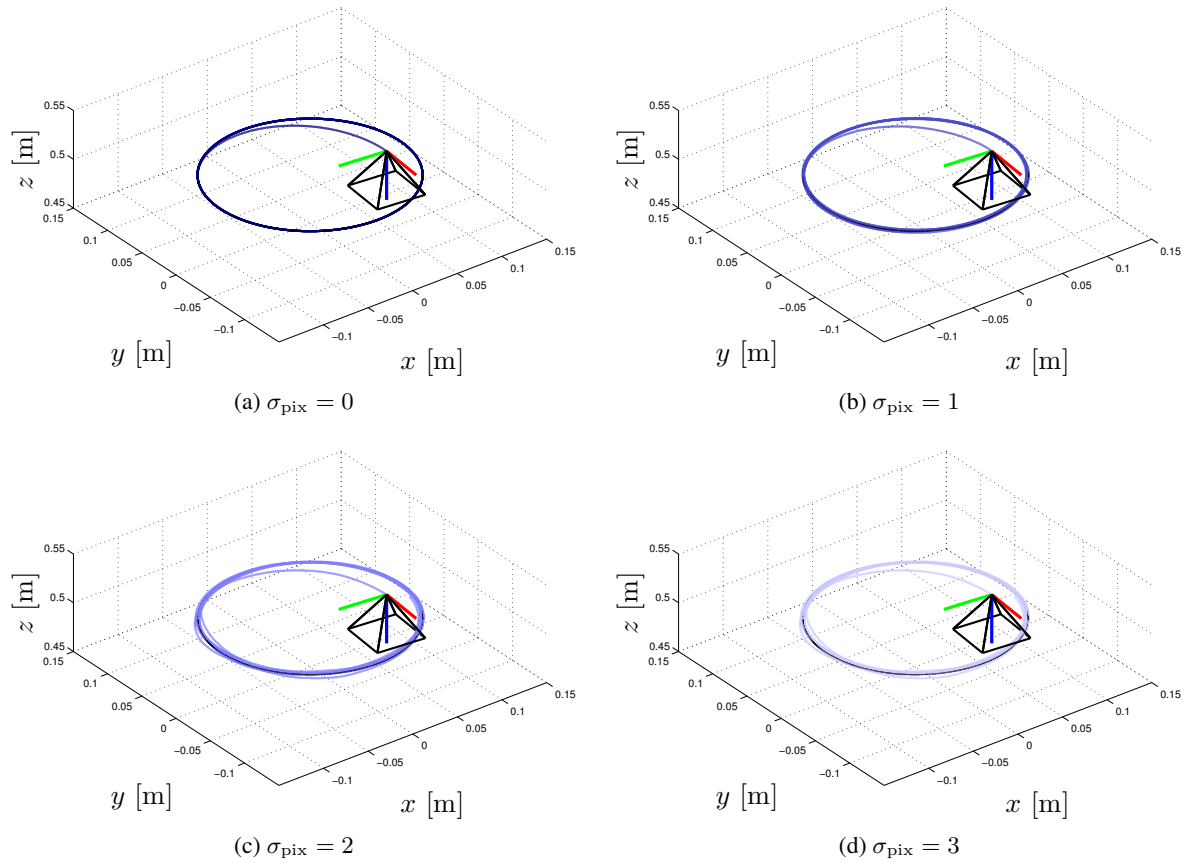


FIGURE 3.9 – Suivi de trajectoire en cercle avec bruit de mesure : trajectoires cartésiennes du repère de la caméra selon les différents écart-types σ_{pix} .

Ensuite nous avons ajouté aux mesures un signal parasite aléatoire (image d'un bruit uniforme) d'écart type σ_{pix} allant de 1 à 3 pixels reprojecté sur le plan focal, ce qui correspond à un écart type dans l'espace des caractéristiques σ_s allant de 1.7mm à 5mm. La simulation est rejouée en intégrant le signal parasite. La figure 3.8a montre l'évolution temporelle de l'erreur selon les différentes valeurs d'écart-type. On observe qu'une fois la période transitoire passée, le système converge et se stabilise. L'erreur relevée ne montre que le signal parasite et ce, quelque soit le taux de bruit utilisé. Le régulateur ici agit aussi comme un filtre passe-bas qui rejette la perturbation. Néanmoins, le bruit a quand même un impact sur la position du repère de la caméra. La figure 3.8b montre la distance relevée entre la position de l'origine du repère mesurée et celle attendue par rapport aux trajectoires des caractéristiques. Cette distance est l'image de l'erreur de position cartésienne du repère par rapport à la trajectoire 3D attendue. Les indicateurs de moyenne et de déviation standard sont reportés dans le tableau 3.1. Les courbes de la figure 3.9 montrent la trajectoire de l'origine du repère mesurée pendant la simulation.

σ_{pix} [pix]	μ_{ex} [mm]	σ_{ex} [mm]
0	4.985×10^{-6}	1.201×10^{-7}
1	1.502	0.8078
2	2.930	1.359
3	5.548	2.830

TABLE 3.1 – Moyennes et déviations standards de l'erreur de position cartésienne de l'origine du repère, selon σ_{pix}

En simulation, la loi de commande présentée dans la section précédente est peu sensible au bruit. Pour les erreurs les plus grandes, l'erreur de position est significative. Néanmoins en pratique, il est assez rare qu'un signal soit bruité de l'ordre de ± 3 pixels. Les gains peuvent être ajustés de manière à améliorer l'erreur dans l'espace cartésien, mais cela se fait au détriment de la bande passante et de la capacité du régulateur à soustraire le signal parasite.

3.3.3 Singularités du KUKA LWR4+

Le passage par les singularités d'un robot constitue une problématique majeure dans la commande robotique.

Par définition, le robot est en position singulière lorsque les degrés de libertés de l'effecteur sont plus nombreux que les degrés de libertés du robot (i.e. $n < m$). Une conséquence directe est que le rang de la matrice cinématique du robot (qui constitue son modèle cinématique) est déficient et que son inverse n'a plus de solutions. Pour les robots redondants, les degrés de liberté supplémentaires peuvent servir à éviter ces configurations, si celles-ci sont évitables.

Lorsque le robot est dans une configuration singulière, le déterminant de sa matrice cinématique s'annule. Dans le cas d'un robot redondant ($n > m$), l'expression à annuler est telle que [Khalil and Dombre, 2002] :

$$\det(\mathbf{J}\mathbf{J}^T) = 0 \quad (3.62)$$

L'analyse du déterminant de la matrice cinématique pour le KUKA LWR4+ montre qu'il existe quatre cas où le robot est en configuration singulière :

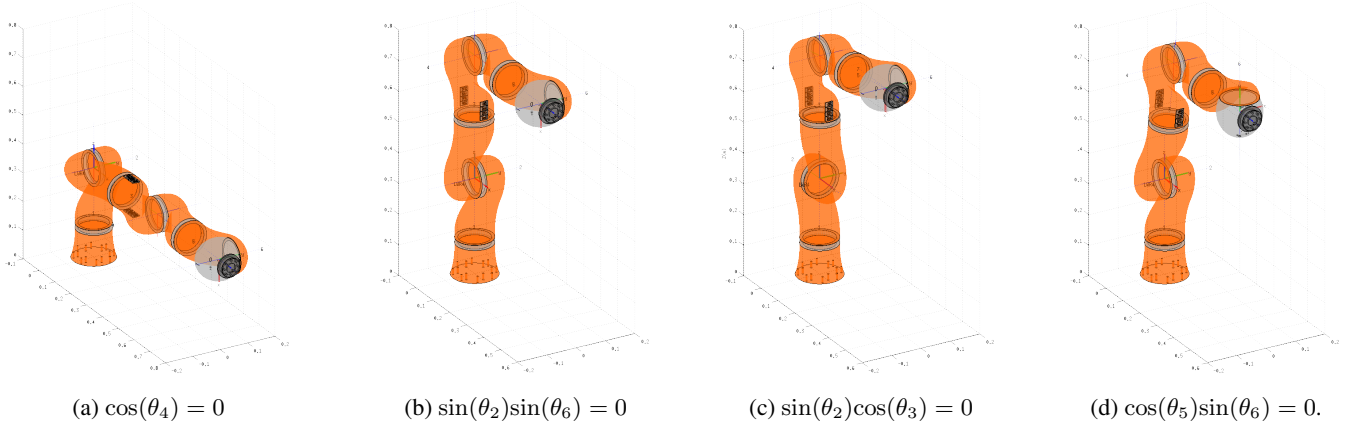


FIGURE 3.10 – Exemples de configurations singulières du KUKA LWR4+ correspondant aux cas où le déterminant s’annule.

- (a) $\cos(\theta_4) = 0$
- (b) $\sin(\theta_2)\sin(\theta_6) = 0$
- (c) $\sin(\theta_2)\cos(\theta_3) = 0$
- (d) $\cos(\theta_5)\sin(\theta_6) = 0$

Une étude similaire a été effectuée par [Bischoff et al., 2010].

Le but ici est d’observer le comportement du système asservi selon la commande dynamique référencée capteurs lorsque le robot approche une singularité. Dans un asservissement référencé capteur, les cas les plus susceptibles de se produire sont lorsque le robot passe soit dans la configuration (b) soit dans la configuration (d).

Nous avons mis en place un asservissement visuel basé image utilisant quatre points 2D. La caméra est fixée sur l’effecteur du robot et l’origine du repère correspond à l’origine du repère de l’effecteur. La cible observée est placée devant la caméra, on demande à ce que la caméra effectue une trajectoire transversale à vitesse constante de 10 cm.s^{-1} . La trajectoire de la caméra a été choisie de manière à ce que le robot s’approche à la fois des configurations singulières (b) et (d) au même instant t . Les équations pour la commande sont les équations (3.44) et (3.40). Pour la simulation, les paramètres du robot pour le calcul de loi de commande et pour le modèle dynamique direct sont identiques pour que la linéarisation soit considérée comme parfaite (*i.e.* le second terme de l’équation (3.48) s’annule).

Les résultats de simulation sont reportés en figures 3.11 et 3.12. La simulation démarre à partir d’une position initiale incertaine, la caméra observe la cible en rouge comme montrée sur la figure 3.11a. Compte tenu des gains utilisés, les caractéristiques mesurées convergent et les erreurs entre celles mesurées et celles désirées s’annulent à partir de $t \simeq 0.5 \text{ s}$. Comme dit précédemment, la trajectoire dans l’espace de l’image conduit le robot à traverser des singularités de type (b) et (d) en même temps. Le passage se produit à $t = 2 \text{ s}$, l’instant est marqué par une ligne verticale grise sur les évolutions temporelles. Dans l’image, l’observation à ce moment est représentée par une cible grise.

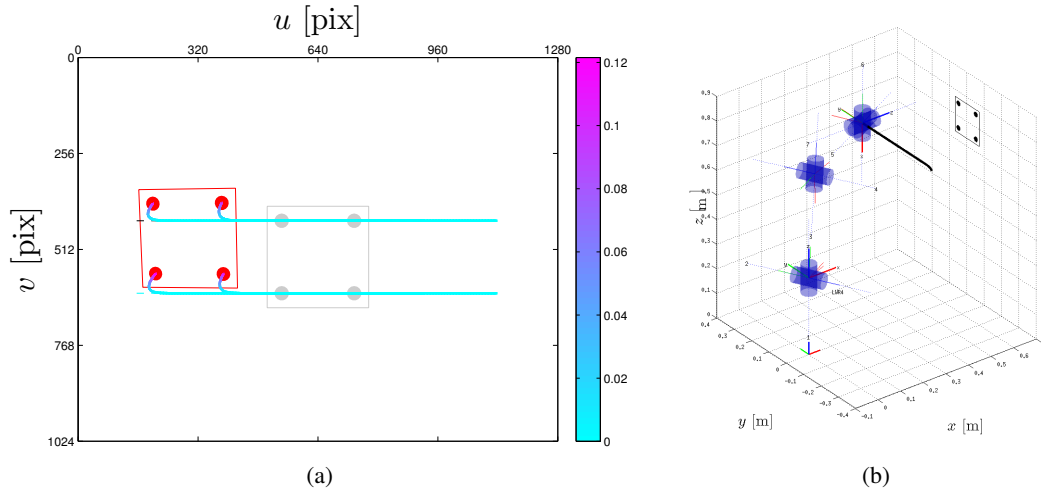


FIGURE 3.11 – Comportement de la commande dynamique référencée capteurs au voisinage d'une singularité robot. (a) Trajectoires des caractéristiques dans l'espace capteur (image) : en rouge, l'observation de la cible en position initiale ; en gris, la cible au voisinage de la singularité robot ; la couleur indique la norme de l'erreur $\|e\|$ le long de la trajectoire. (b) Trajectoire de l'origine du repère de la caméra dans l'espace cartésien. Le robot est représenté dans sa configuration à $t = 4$ s.

L'intégrateur ODE45 du logiciel Matlab avec un pas de temps variable a été utilisé pour s'approcher le plus possible de la configuration singulière. A $t \simeq 2$ s, la plus petite valeur numérique du déterminant a été relevée telle que $\det(\mathbf{J}\mathbf{J}^T) = 2.2535 \times 10^{-18}$ et la différence entre la configuration courante et la vraie configuration singulière est de l'ordre de 1×10^{-6} rad.

La trajectoire dans l'espace capteur a fait que le robot devait passer par une singularité. Or, le système ne montre pas de comportement particulier au voisinage de celle-ci. Le robot passe la singularité et continue sa poursuite de la trajectoire. On note que la méthode d'intégration explicite fait que le système s'est approché à un niveau infinitésimal de la configuration singulière.

Il est possible de dire que la commande dynamique référencée capteurs est dans ce cas précis robuste au passage de ce type de singularité. Il reste quand même nécessaire de valider cette affirmation dans des conditions réelles. Cependant, du fait des erreurs dues aux tolérances mécaniques, du traitement de signal et autres, la probabilité de s'approcher de la configuration singulière sur un système réel au même niveau que celui atteint dans la simulation, reste faible.

3.3.4 Erreur de modélisation géométrique et dynamique du robot

Les erreurs de modélisation peuvent être à l'origine d'un mauvais fonctionnement du contrôleur. En effet, pour garantir une linéarisation parfaite du système réel à commander, il est nécessaire de connaître avec finesse les paramètres du système réel. Cette connaissance des paramètres géométriques et dynamiques est nécessaire lorsque l'on souhaite utiliser une commande par couple calculé, notamment pour calculer l'équation de découplage. L'identification des paramètres d'un robot est un champ de recherche à part entière : des exemples de procédure d'identification peuvent être trouvée dans [De Luca et al., 2006; Jubien et al., 2014].

Cependant, il se peut que tous les paramètres soient mal identifiés, voire même non identi-

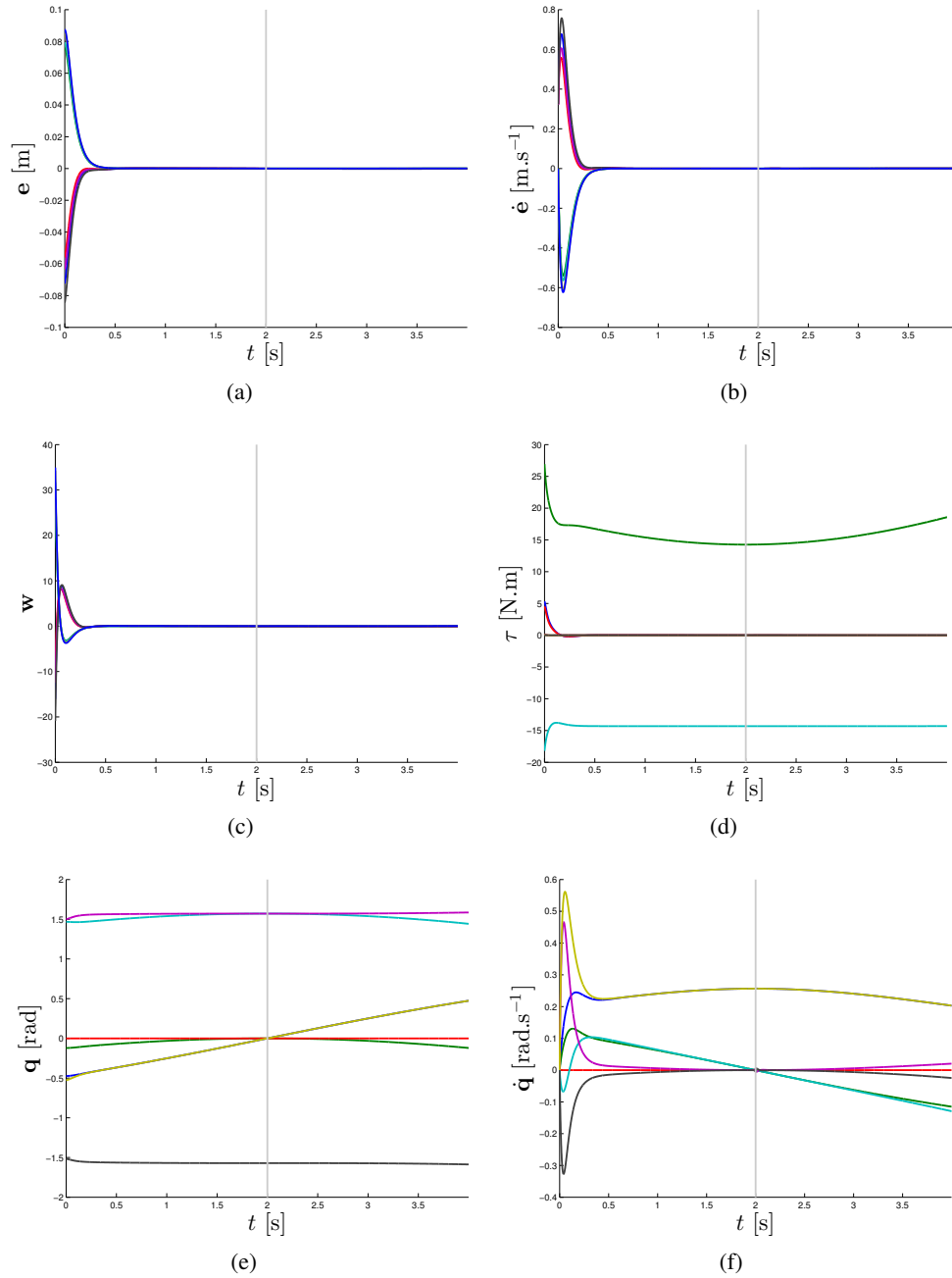


FIGURE 3.12 – Comportement de la commande dynamique référencée capteurs au voisinage de la singularité robot. Le robot passe par la singularité à $t = 2$ s, instant représenté par la ligne verticale grise.

fiables. Dans ce cas, des valeurs arbitraires sont utilisées, avec des conséquences directes sur les performances du système en boucle fermée qui se retrouvent plus particulièrement sur l'erreur statique et l'erreur de traînage lors d'un suivi de trajectoire.

Le but de l'étude sera d'observer l'impact des erreurs de modélisation qui entraînent une mauvaise linéarisation. On rappelle ici l'équation différentielle qui décrit l'évolution temporelle du système en boucle fermée lorsqu'une dynamique commande référencée capteurs est utilisée (équation (3.48)) :

$$\ddot{\mathbf{e}} + \mathbf{K}_{sv}\dot{\mathbf{e}} + \mathbf{K}_{sp}\mathbf{e} = \hat{\mathbb{J}}\hat{\mathbf{A}}^{-1} \left((\mathbf{A}\mathbb{J}^+ - \hat{\mathbf{A}}\hat{\mathbb{J}}^+)\ddot{\mathbf{s}} - \mathbf{A}\mathbb{J}^+\mathbf{b} + \hat{\mathbf{A}}\hat{\mathbb{J}}^+\hat{\mathbf{b}} + \mathbf{H} - \hat{\mathbf{H}} \right)$$

On rappelle que le système est parfaitement linéarisé lorsque l'équation ci-dessus s'annule. Une analyse rapide permet d'identifier quelles sont les conditions où la différence entre les deux jeux de paramètres utilisés a un impact :

- Lorsque le système ne bouge plus (i.e. $\mathbf{s} = \mathbf{q} = \text{cte}$ et $\dot{\mathbf{s}} = \dot{\mathbf{q}} = 0$), le second terme de l'équation ne comporte plus que $\hat{\mathbb{J}}\hat{\mathbf{A}}^{-1}(\mathbf{H} - \hat{\mathbf{H}})$ où $\mathbf{H}$ ne représente que de la gravité et des forces extérieures s'appliquant sur le robot. Il en résulte donc une erreur statique due à une mauvaise identification des paramètres de masses et de centre de gravité.
- Lorsque le système se déplace à vitesse constante, le terme $\hat{\mathbb{J}}\hat{\mathbf{A}}^{-1}(\hat{\mathbf{A}}\hat{\mathbb{J}}^+\hat{\mathbf{b}} - \mathbf{A}\mathbb{J}^+\mathbf{b})$ devient non nul, s'ajoute à l'erreur statique précédente et constitue l'erreur de traînage.
- Enfin, c'est pendant les phases de transitions où l'accélération est non nulle ($\ddot{\mathbf{s}} \neq 0$), que la quantité $\hat{\mathbb{J}}\hat{\mathbf{A}}^{-1}(\mathbf{A}\mathbb{J}^+ - \hat{\mathbf{A}}\hat{\mathbb{J}}^+)\ddot{\mathbf{s}}$ aura un impact maximal.

En simulation, le robot est contrôlé en utilisant les équations (3.40) et (3.44) avec quatre points 2D mesurés à l'aide d'une caméra dont les paramètres sont connus et parfaitement identifiés. Dans l'espace de l'image, la trajectoire des caractéristiques est définie selon une interpolation linéaire d'ordre 5 avec des vitesses et accélérations maximum. Le jeu de paramètres utilisé pour le calcul de la loi de commande s'écarte du jeu réel avec un taux d'erreur arbitraire d'1% à 2% sur l'ensemble des paramètres géométriques et dynamiques.

Les résultats de simulation sont montrés sur la figure 3.13. Sont représentés l'évolution des caractéristiques dans l'espace de l'image, l'évolution temporelle des quantités $\hat{\mathbb{J}}\hat{\mathbf{A}}^{-1}(\mathbf{A}\mathbb{J}^+ - \hat{\mathbf{A}}\hat{\mathbb{J}}^+)\ddot{\mathbf{s}}$ et $\hat{\mathbb{J}}\hat{\mathbf{A}}^{-1}(\hat{\mathbf{A}}\hat{\mathbb{J}}^+\hat{\mathbf{b}} - \mathbf{A}\mathbb{J}^+\mathbf{b})$ ainsi que l'évolution des erreurs $\mathbf{s} - \mathbf{s}^d$ correspondantes.

On observe pour chacune des phases que l'impact d'une mauvaise modélisation est peu significatif sur l'erreur dans l'espace des caractéristiques dans les conditions de la simulation. L'erreur statique maximale relevée pendant la phase à vitesse constante est de l'ordre d'environ 1.2×10^{-5} m sur le plan focal de l'image, ce qui correspond à 0.003 pixels dans l'image. De même pendant la phase d'accélération, l'erreur maximale relevée est d'environ 3×10^{-5} m, soit 0.02 pixels.

En conclusion, l'excitation évolue comme le prédit l'analyse précédente le long de la trajectoire. Néanmoins, la simulation n'avait pour but que d'observer l'impact de ces erreurs de modélisation du robot sur l'évolution du système. Les erreurs de modélisation du capteur et de sa fonction de projection ne rentrent pas dans le cadre de cette étude. Cette simulation doit être complétée par une étude identique sur un système réel pour quantifier l'impact tenant compte de l'ensemble des erreurs de modélisation (robot et capteur).

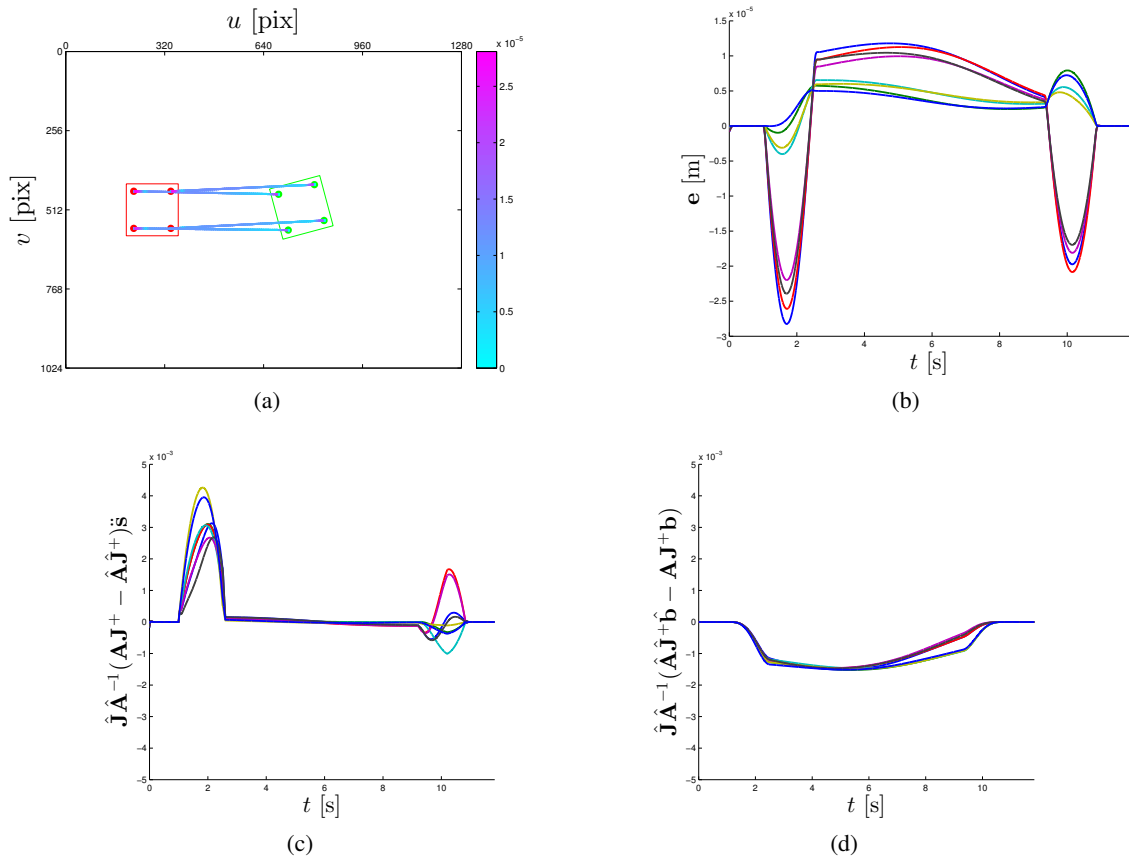


FIGURE 3.13 – Évolution temporelle des termes d'excitation de l'équation différentielle du système en boucle fermée et l'impact correspondant sur l'erreur des caractéristiques.

3.4 Conclusions

Ce chapitre s'est concentré sur la présentation d'un nouvel outil pour la commande référencée capteurs : le modèle du second ordre et son utilisation pour en faire une commande de type dynamique.

Pour rappel, plusieurs besoins ont été avancés pour le développement d'une telle loi de commande. En premier lieu, en connexion directe avec les développements du deuxième chapitre, le contrôle de la trajectoire de l'effecteur d'un robot s'aidant de références attachées au monde réel dans une application de type assemblage était un point parmi les plus importants. De plus, la commande référencée capteur classique n'offrait pas la possibilité de contrôler la variation des caractéristiques pendant la phase transitoire. Enfin, il était nécessaire d'augmenter la commande référencée capteurs pour rendre le mixage entre des informations d'efforts et de caractéristiques spatiales plus facile.

Après avoir présenté les besoins et l'état de l'art, le chapitre s'est concentré sur la nouvelle formulation de la commande et ses différentes formes. Nous nous sommes arrêtés en premier lieu sur la dérivation du modèle cinématique qui lie l'évolution des caractéristiques au torseur cinématique du capteur en faisant apparaître le hessien de la matrice d'interaction.

Ensuite, nous avons calculé les modèles du second ordre pour les caractéristiques type point 2D et 3D, ligne 2D et 3D ainsi que le plan 3D, caractéristiques qui peuvent être utilisées directement dans des opérations d'assemblage en s'appuyant sur le cadre présenté précédemment.

Enfin, différentes solutions ont été proposées pour obtenir une linéarisation du système en boucle fermée par découplage, en considérant le capteur seul ou fixé de manière rigide sur l'effecteur d'un robot.

La validation de la nouvelle formulation pour la commande dynamique référencée capteurs s'est faite au travers de différents cas d'étude spécifiques. Nous avons étudié entre autres la réalisation d'un suivi de trajectoire dans l'espace capteur, la robustesse au bruit de mesure, le passage de configuration singulière d'un robot manipulateur et l'incidence d'erreurs sur les paramètres géométriques et dynamiques du robot.

En conclusion, la nouvelle loi de commande basée sur le modèle du second ordre et la linéarisation par découplage dynamique font que le système est dorénavant capable de suivre une trajectoire dans l'espace capteurs, garantissant une erreur de traînage nulle et ce, tenant compte de n'importe quelle nature de caractéristique dans des situations variées. La nouvelle loi de commande peut être appliquée pour l'exécution d'une opération de peinture par exemple ou tout autre opération qui nécessite un contrôle fin en vitesse sans contact.

Les perspectives autour de ce travail sont potentiellement nombreuses.

En premier lieu, la loi de commande repose sur l'hypothèse forte que l'information de variation du signal $\dot{s}$ est fournie par les capteurs. Or, il n'existe pas à l'heure actuelle de capteurs capables de fournir à la fois l'information spatiale ainsi que sa variation. De nouvelles architectures de capteurs telle que *Dynamic Vision Sensor* [Lichtsteiner et al., 2008] font leur apparition et font la promesse d'une mesure de variation du signal spatial qui pourrait s'intégrer alors à notre loi de commande.

Le travail effectué pour la formalisation du hessien dans le modèle du second ordre reste aussi incomplet. En effet, il est nécessaire de limiter les dépendances du hessien à des paramètres qui n'ont pas de lien direct avec la caractéristique en question. C'est le cas particulièrement pour

les caractéristiques appartenant à la famille 2D. Par exemple, le hessien du point 2D pourrait idéalement être fonction seulement de s , de sa variation $\dot{s}$ ainsi que la profondeur du point z et de sa variation $\dot{z}$. De même, la paramétrisation $[\rho \ \theta]^\top$ pour la ligne 2D n'a pas apporté de bon résultat. Il serait possible d'obtenir le modèle du second ordre à partir de la projection des coordonnées de Plücker sur le plan focal de la caméra.

Une dernière possibilité est l'interaction de la commande dynamique référencée capteurs avec une commande en effort pour offrir un système de contrôle complet à l'utilisateur. C'est l'objet du prochain chapitre.

Commandes hybrides

Les chapitres précédents se sont focalisés sur la commande référencée capteurs et ce qu'elle pouvait apporter dans le cas où le robot réalise une opération d'assemblage et/ou une autre application où le contrôle en vitesse est nécessaire.

Le chapitre 2 a montré qu'il était possible de généraliser le recours à la commande référencée capteurs cinématique tenant compte du robot, des capteurs qui l'équipent et de l'opération d'assemblage renseignée à partir d'une maquette numérique. Le chapitre 3 donne la capacité au système de contrôler la variation des caractéristiques et apporte la garantie d'annuler l'erreur de traînage. *In fine*, la loi de commande permet d'obtenir des meilleures performances en suivi de trajectoire dans l'espace capteurs. Le robot est donc en mesure de suivre une trajectoire d'approche précise par rapport à des références rattachées à la structure et au monde réel.

Néanmoins, il reste une chose essentielle pour faire en sorte que le cadre proposé puisse apporter une solution à toutes les classes d'application dans l'industrie manufacturière : la prise en compte d'information émanant d'un capteur d'effort.

Le but de ce chapitre est d'apprécier le potentiel de la commande dynamique lorsque celle-ci est utilisée au sein d'une loi de commande hybride qui met en concurrence un asservissement spatial et un asservissement en effort. Après une première partie dédiée à l'état de l'art des commandes hybrides, les équations des schémas de commande hybride parallèle et hybride externe en tenant compte des éléments du chapitre précédent seront développées. Enfin, une étude de cas est effectué pour mettre en lumière les spécificités des nouvelles formulations des lois de commande hybride.

4.1 État de l'art

4.1.1 Commande en impédance

Les commandes basées sur le modèle dynamique sont capables d'améliorer le comportement dynamique d'une chaîne poly-articulée et ainsi de réduire les erreurs statiques et les erreurs de traînage par rapport à une trajectoire demandée. Même si les équations de découplage prennent en compte l'action d'une force extérieure, les commandes ne permettent pas le contrôle – indirect ou

direct – de la quantité d’effort appliquée par le robot sur l’environnement extérieur.

Les méthodes de type contrôle actif de la raideur ou contrôle de l’impédance permettent un contrôle indirect de la quantité d’effort que le robot applique sur son environnement. Les travaux liés à la commande en impédance sont nombreux [Khalil and Dombre, 2002; Villani and De Schutter, 2008; Chen and Liu, 2012; Ott et al., 2015a,b]. Ci-après est présentée la méthode de la commande en impédance du point de vue de Khalil and Dombre [2002].

L’idée ici est d’assigner un comportement dynamique particulier au robot lorsque celui-ci est en contact avec son environnement. Le comportement est spécifié par une impédance dynamique généralisée représentée par un système masse-ressort-amortisseur.

La vitesse de l’effecteur $\mathbf{v} = \dot{\mathbf{x}}$ et la force que le robot applique sur l’environnement $\mathbf{f}$ sont reliées par une impédance mécanique notée $\mathbf{z}$. La fonction de transfert dans le domaine de Laplace est donnée par :

$$\mathbf{F}(p) = \mathbf{Z}(p) \mathbf{V}(p) \quad (4.1a)$$

$$\mathbf{F}(p) = p \mathbf{Z}(p) \mathbf{X}(p) \quad (4.1b)$$

où $\mathbf{F}$, $\mathbf{Z}$ et $\mathbf{V}$ représentent les fonctions de transfert associées aux quantités temporelles, respectivement, de la force, du système d’impédance et de la vitesse cartésienne de l’effecteur. On note p l’opérateur de Laplace.

Par rapport au comportement que l’utilisateur souhaite appliquer à son robot, la quantité $p\mathbf{Z}(p)$ doit être l’équivalent d’un système mécanique de type masse-ressort-amortisseur. Il est possible d’explicitier la fonction de transfert d’un tel système telle que :

$$p\mathbf{Z}(p) = \Lambda s^2 + \mathbf{B}s + \mathbf{K} \quad (4.2)$$

Λ , $\mathbf{B}$ et $\mathbf{K}$ représentent respectivement l’inertie du système, son amortissement et sa raideur. Les valeurs de ces matrices sont déterminées en fonction de la tâche à effectuer :

- Des gains importants sont appliqués à Λ dans les directions où un contact avec l’effecteur est attendu. Ceci a pour effet de limiter la dynamique du robot dans ces directions.
- Des gains importants sont appliqués à $\mathbf{B}$ dans les directions où l’énergie cinétique doit être dissipée et ainsi amortir les changements de phase.
- La matrice de raideur $\mathbf{K}$ doit recevoir de faibles gains dans les directions où le contact est attendu. Ces termes contrôlent indirectement la quantité d’effort appliqué par le robot sur son environnement. Ces gains affectent aussi la précision de contrôle en position : ainsi l’utilisateur doit mettre des gains élevés dans les autres directions pour conserver une précision suffisante de la position.

Dans le domaine temporel, l’équation (4.1b) tenant compte de (4.2) devient :

$$\mathbf{f} = \Lambda(\ddot{\mathbf{x}}^d - \ddot{\mathbf{x}}) + \mathbf{B}(\dot{\mathbf{x}}^d - \dot{\mathbf{x}}) + \mathbf{K}(\mathbf{x}^d - \mathbf{x}) \quad (4.3)$$

Et ainsi,

$$\ddot{\mathbf{x}} = \ddot{\mathbf{x}}^d + \Lambda^{-1} \left(\mathbf{B}(\dot{\mathbf{x}}^d - \dot{\mathbf{x}}) + \mathbf{K}(\mathbf{x}^d - \mathbf{x}) - \mathbf{f} \right) \quad (4.4)$$

La loi de commande est construite sur la base du modèle dynamique en reprenant la méthode de la commande par couple calculé dans l’espace cartésien. En considérant l’équation (3.8), en

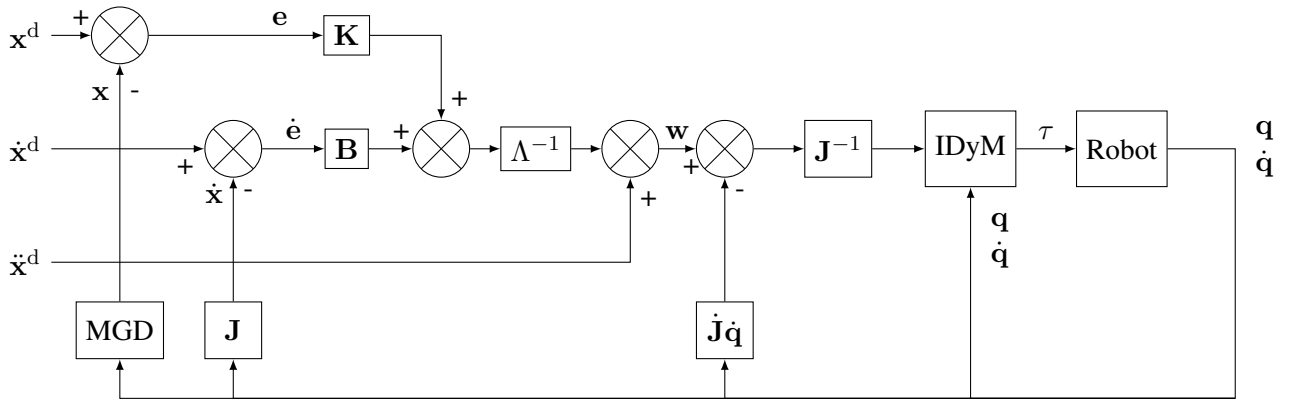


FIGURE 4.1 – Schéma bloc illustrant la commande en impédance sans retour d'information d'effort [Khalil and Dombre, 2002].

remplaçant $w(t)$ par l'équation (4.4), l'équation de la commande en impédance est définie par l'équation (4.5).

$$\tau = \hat{\mathbf{A}}\hat{\mathbf{J}}^{-1} \left(\ddot{\mathbf{x}}^d + \Lambda^{-1} (\mathbf{B}\dot{\mathbf{e}} + \mathbf{K}\mathbf{e} - \mathbf{f}) - \dot{\mathbf{J}}\dot{\mathbf{q}} \right) + \hat{\mathbf{H}} + \mathbf{J}^T \mathbf{f} \quad (4.5)$$

Il est possible d'utiliser cette loi de commande pour faire appliquer par le robot un effort déterministe. Mais le contrôle de la quantité d'effort appliquée reste indirect. En l'absence de capteur d'effort, l'utilisateur doit calculer le retour d'effort avec la raideur et l'erreur de position selon la direction du contact en tenant compte d'un modèle de ressort linéaire. La figure 4.1 montre le schéma bloc associé à cette commande lorsqu'aucune lecture de l'effort n'est disponible.

Cette loi de commande est celle qui est actuellement utilisée sur le robot LWR4+ dans son mode de commande en impédance articulaire (*joint impedance controller*) ou cartésienne (*cartesian impedance controller*) [Bischoff et al., 2010].

4.1.2 Commande hybride parallèle

Dans les commandes de type parallèle hybride position force, le robot est contrôlé par deux asservissements en boucle fermée [Raibert and Craig, 1981; Khatib, 1987]. Les deux asservissements sont complémentaires : le premier assure le positionnement du robot par des contraintes spatiales, le second assure l'application des forces via des contraintes d'effort. Les deux lois sont complètement séparées et asservissent le système en fonction des mesures de capteurs qu'ils reçoivent et leurs lois de commande respectives. La contribution de chaque boucle est fusionnée en un signal de commande global envoyé à l'équation de découplage de la loi de commande, avant d'être envoyé aux actionneurs pour faire bouger le système mécanique. Chaque degré de liberté du repère de l'effecteur est contrôlé soit par l'asservissement spatial, soit par l'asservissement en effort. Le

choix s'effectue par l'utilisation d'une matrice de sélection diagonale $\mathbf{S}$ telle que :

$$\mathbf{S} = \begin{bmatrix} s_1 & 0 & \dots & \dots & \dots & 0 \\ 0 & s_2 & 0 & \dots & \dots & \vdots \\ \vdots & 0 & s_3 & 0 & \dots & \vdots \\ \vdots & \vdots & 0 & s_4 & 0 & \vdots \\ \vdots & \vdots & \vdots & 0 & s_5 & 0 \\ 0 & \dots & \dots & \dots & 0 & s_6 \end{bmatrix} \quad (4.6)$$

où $s_j = 1$ si le $j^{\text{ème}}$ degré de liberté de l'effecteur est contrôlé via l'asservissement spatial, $s_j = 0$ si celui-ci est contrôlé par l'asservissement en force.

Trois formes de schéma de contrôle hybride peuvent être distinguées selon le signal global de commande que les deux asservissements génèrent par sommation [Khalil and Dombre, 2002] :

- La somme produit un signal équivalent à des vitesses cartésiennes.
- La somme produit un signal équivalent à des couples articulaires.
- La somme produit un signal équivalent à des efforts cartésiens (force et couple s'appliquant à l'effecteur).

On se focalise ici sur un schéma de contrôle hybride parallèle qui utilise la commande par couple calculé dans l'espace cartésien en tant qu'asservissement spatial. On ajoute à l'équation de découplage (3.8) la contribution de l'asservissement en effort :

$$\tau = \hat{\mathbf{A}}\hat{\mathbf{J}}^{-1} \left(\mathbf{w}(t) - \hat{\mathbf{J}}\dot{\mathbf{q}} \right) + \hat{\mathbf{H}} + \mathbf{J}^T \mathbf{y}(t) \quad (4.7)$$

$\mathbf{w}(t)$ et $\mathbf{y}(t)$ dans l'équation (4.7) constituent les vecteurs d'entrée pour les boucles d'asservissement respectivement spatial et d'effort. Les deux sont définis ci-après :

$$\mathbf{w}(t) = \mathbf{S} \left(\ddot{\mathbf{x}}^d + \mathbf{K}_v(\dot{\mathbf{x}}^d - \dot{\mathbf{x}}) + \mathbf{K}_p(\mathbf{x}^d - \mathbf{x}) \right) \quad (4.8)$$

$$\mathbf{y}(t) = (\mathbf{I} - \mathbf{S}) \left(\mathbf{f}^d + \mathbf{K}_{fp}(\mathbf{f}^d - \mathbf{f}) - \mathbf{K}_{fv}\dot{\mathbf{x}} + \mathbf{K}_{fi} \int_{t_0}^t (\mathbf{f}^d - \mathbf{f}) d\tau \right) \quad (4.9)$$

Le schéma bloc correspondant aux équations (4.9) et (4.8) est montré sur la figure 4.2. On observe que le vecteur d'entrée découplée de l'asservissement spatial est très similaire à l'équation (3.9). Le vecteur d'entrée pour la boucle en effort est composée des éléments suivants :

- $\mathbf{f}^d$, $\mathbf{f} \in \mathbb{R}^6$ représentent respectivement les vecteurs d'effort désiré et mesuré.
- $\mathbf{K}_{fp}$, $\mathbf{K}_{fv}$, $\mathbf{K}_{fi} \in \mathbb{R}^{6 \times 6}$ sont les gains proportionnel, dérivé et intégral du correcteur PID qui constitue l'asservissement en effort.

Il est nécessaire de noter que dû à la quantité importante de bruit habituellement observée sur les mesures d'effort, les vitesses articulaires sont préférées à la différenciation des informations d'effort pour le terme dérivé.

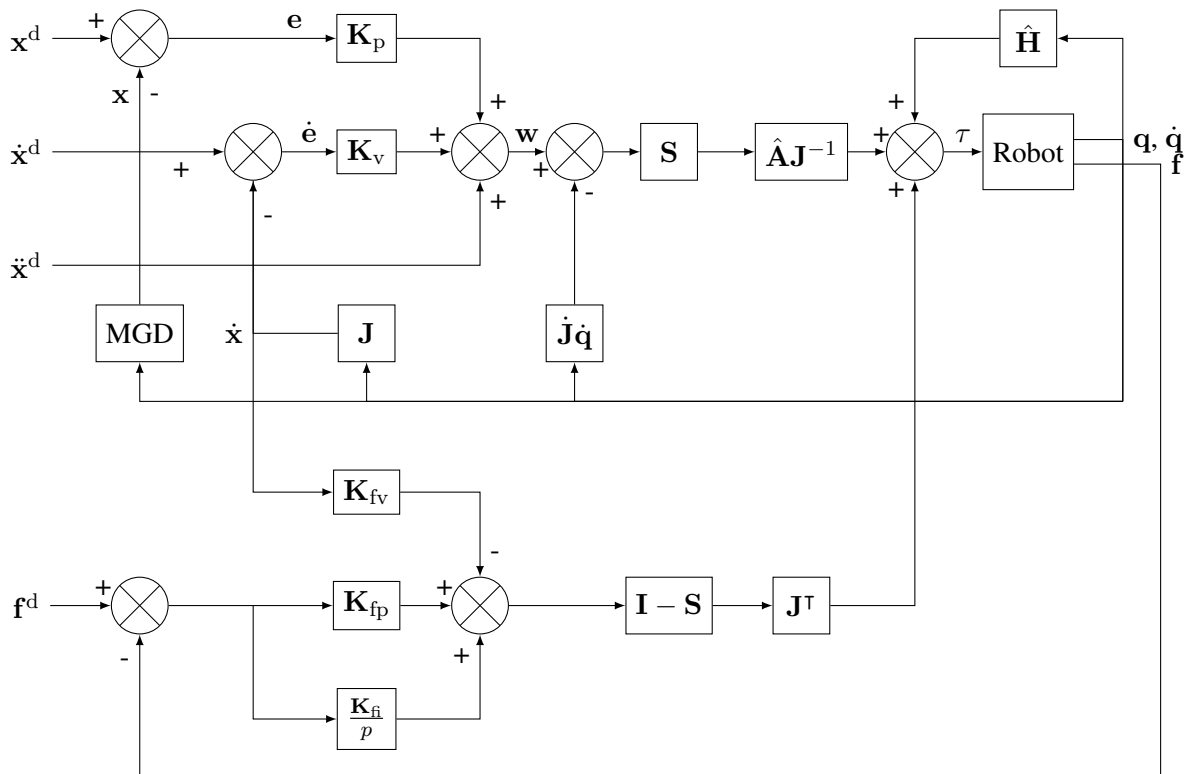


FIGURE 4.2 – Schéma bloc illustrant la commande hybride parallèle utilisant la commande par couple calculé en tant qu'asservissement spatial [Khatib, 1987].

4.1.3 Commande hybride externe

Contrairement à la commande parallèle où le schéma de commande interdit une contribution simultanée des deux asservissements spatial et d'effort sur une même direction, la commande hybride externe [De Schutter and Van Brussel, 1988a,b; Perdereau and Drouin, 1993] l'autorise.

La commande est composée d'une boucle fermée pour l'asservissement spatial enfermée à l'intérieur d'une deuxième boucle pour l'asservissement en effort. Le signal produit par la deuxième boucle externe modifie la consigne de la boucle interne. Le signal d'erreur en effort génère une nouvelle consigne d'entrée $\dot{\mathbf{x}}$ qui est ajoutée à la consigne désirée de l'asservissement en position $\mathbf{x}^d$. C'est cette somme qui constitue la nouvelle consigne $\mathbf{x}^{d'}$. Le robot reste totalement contrôlé par l'asservissement en position. Néanmoins grâce à cette méthode, l'utilisateur reste en mesure de spécifier des consignes d'effort à appliquer à l'effecteur.

Un des avantages principaux de cette approche est que la boucle d'asservissement en effort peut être ajoutée par dessus un contrôleur en position déjà existant, par exemple sur les robots industriels. Un autre avantage est que la commande permet le contrôle à la fois en effort et en position sur une même direction, ce qui constitue une différence majeure avec la commande parallèle.

En considérant une commande en couples calculés avec une régulation en position comme définie en paragraphe 3.1.2.1 en tant que boucle pour la position, la commande hybride externe se compose des équations suivantes : l'équation de découplage est réalisée par l'équation (3.8) rappelée ci-après,

$$\boldsymbol{\tau} = \hat{\mathbf{A}}\hat{\mathbf{J}}^{-1} \left(\mathbf{w}(t) - \dot{\mathbf{J}}\dot{\mathbf{q}} \right) + \hat{\mathbf{H}}$$

où l'entrée de découplage $\mathbf{w}(t)$ s'exprime par :

$$\mathbf{w}(t) = \mathbf{K}_p(\mathbf{x}^d + \mathbf{y}(t) - \mathbf{x}) - \mathbf{K}_v\dot{\mathbf{x}} \quad (4.10)$$

et

$$\mathbf{y}(t) = \mathbf{W} \left(\mathbf{f}^d + \mathbf{K}_{fp}(\mathbf{f}^d - \mathbf{f}) + \mathbf{K}_{fi} \int_{t_0}^t (\mathbf{f}^d - \mathbf{f}) d\tau \right) \quad (4.11)$$

L'équation (4.11) est similaire à (4.9), à la différence que le terme de contre réaction dérivée a été retiré. En effet, celui-ci est déjà présent en (4.10). La matrice de gains $\mathbf{W}$ est homogène à l'inverse d'une matrice de raideur. Celle-ci peut être déduite à partir d'un modèle de ressort linéaire par exemple.

Plus récemment, la commande hybride externe a été réécrite pour utiliser une commande référencée capteurs à la place de la boucle interne pour l'asservissement en position [Mezouar et al., 2007; Prats et al., 2007; Prats, 2009] et celle-ci a été utilisée avec succès lors d'une opération de découpe robotisée [Long et al., 2014a; Long, 2014].

Dans [Prats et al., 2007], les auteurs ont utilisé une commande référencée capteur tenant compte d'une caractéristique de type pose (cf. paragraphe 2.1.4.1), d'équation $\mathbf{s} = [\mathbf{t}^T \mathbf{u}^T \theta]^T$. De la même manière que précédemment, la contribution de la boucle externe responsable de l'asservissement en effort vient modifier la consigne d'entrée de l'asservissement visuel. L'équation pour la boucle interne est similaire à l'équation (2.33) et est définie par :

$$\dot{\mathbf{q}} = -\lambda \hat{\mathbf{J}}^{-1} \mathbf{N} \hat{\mathbf{T}}_S \hat{\mathbf{L}}_S^+ \left(\mathbf{s} - (\mathbf{ds} + \mathbf{s}^d) \right) \quad (4.12)$$

La contribution de la boucle d'effort est définie par l'équation (4.14). Les auteurs ont souhaité mettre en place un gain proportionnel sur l'erreur entre l'effort mesuré et celui désiré. La matrice

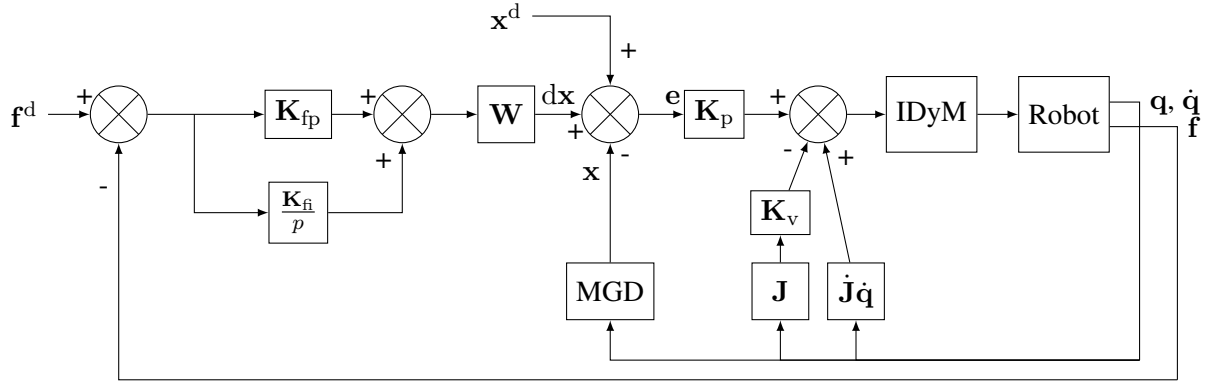


FIGURE 4.3 – Schéma bloc illustrant la commande hybride externe utilisant la commande par couple calculé avec un régulateur type proportionnel dérivée en position pour la boucle interne et un régulateur proportionnel intégral pour la consigne d'effort [Khalil and Dombre, 2002].

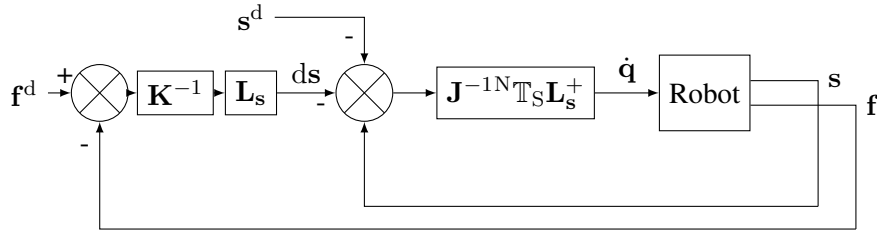


FIGURE 4.4 – Schéma bloc illustrant la commande hybride externe utilisant une commande référencée capteurs basée image en tant que boucle d'asservissement interne [Prats et al., 2007].

des gains $\mathbf{K}$ peut être assimilée à une matrice de raideur de l'environnement.

$$d\mathbf{x} = \mathbf{K}^{-1} (\mathbf{f} - \mathbf{f}^d) \quad (4.13)$$

Le déplacement $d\mathbf{x}$ déterminé par la mesure du capteur d'effort est exprimé dans l'espace capteur grâce à la matrice d'interaction estimée $\hat{\mathbf{L}}_s$ telle que :

$$d\mathbf{s} = \hat{\mathbf{L}}_s d\mathbf{x} \quad (4.14)$$

La figure 4.4 montre le schéma bloc correspondant à cette loi de commande. Il est nécessaire de noter que les équations dans les travaux de Mezouar et al. [2007] et Prats et al. [2007] comportent une matrice de passage supplémentaire $\mathbf{L}_\times$, les auteurs indiquent que celle-ci peut être apparentée à une matrice identité pour de petits déplacements de l'effecteur. Il faut aussi noter que la commande hybride externe basée vision génère des consignes de vitesses articulaires à l'instar des autres commandes en couples. Il est donc nécessaire de mettre en place une autre boucle d'asservissement supplémentaire pour le contrôle bas-niveau des moteurs électro-mécaniques.

4.1.4 Critiques

Parmi l'ensemble de ses commandes, une seule propose une commande directe sur la base d'information provenant d'un capteur d'effort : c'est la commande hybride parallèle. Cette méthode a l'inconvénient de séparer le champ d'action des deux asservissements. La commande hybride

externe permet l'action de deux asservissements sur une même direction. Néanmoins, pour éviter les conflits de contribution entre les deux asservissements spatial et en effort, la contribution de l'asservissement en effort vient modifier la consigne d'entrée de la boucle interne de l'asservissement en position. L'information en effort est donc transformée en information spatiale tenant compte d'un modèle de déplacement infinitésimal. Le robot n'est donc pas contrôlé directement par l'information en effort. La modification est d'autant plus importante si une commande référencée capteurs est utilisée en tant qu'asservissement en position. Cependant, ces deux commandes constituent à l'heure actuelle le meilleur moyen de mixer des contributions d'effort et d'asservissement référencée capteurs.

A contrario, la commande parallèle permet une application directe de la commande en effort sur le robot seulement dans certaines directions, spécifiées par l'utilisateur. L'asservissement spatial n'a considéré jusqu'à présent que des informations de pose à 6 degrés de liberté.

4.2 Commande dynamique référencée capteurs hybride

Dans une industrie manufacturière d'un point de vue opérateur, nombreux sont les mouvements qui nécessitent une attention particulière directe sur ce que l'on est en train de faire tout en exerçant un effort.

Dans une opération d'assemblage, lorsque la pièce à déposer est au contact de la structure et qu'il est nécessaire d'appliquer une quantité de force pour maintenir la pièce correctement positionnée sur la structure, le mouvement (ici, quasi-statique) peut se décomposer en deux parties : en premier lieu, il est nécessaire de maintenir la position de la pièce selon les références choisies de la structure, en second lieu il est nécessaire de maintenir la quantité d'effort voulue dans la direction souhaitée. Très vraisemblablement dans la majorité des cas, la direction où l'effort est appliqué ne rentre pas en conflit avec le maintien en position.

Dans un autre exemple, une opération de vissage voit son mouvement se décomposer en trois sous-parties : la nécessité de maintenir la position d'approche en rapport avec la position mesurée de la vis, le couple apporté autour de son axe de rotation (ce qui réalise l'action de vissage) et le mouvement de suivi de la tête de vis. En effet, l'action de tourner la vis fait que celle-ci s'enfonce dans la structure. Si l'on se place dans un cas où le robot maintient fermement la tête de vis, celui-ci doit se déplacer librement le long de l'axe de la vis.

Il est possible de lister d'autres cas de figure qui n'appartiennent pas à la catégorie des tâches quasi-statiques telles que celles énoncées ci-dessus. En effet, l'action de marquage ou de traçage sont des mouvements où l'application d'une force et le suivi d'une trajectoire particulière sont nécessaires.

Les développements effectués dans les prochains paragraphes tendent à être appliqués de manière systématique sur tous les cas de figure possibles. Un de ces cas fera l'objet d'une étude approfondie en simulation.

4.2.1 Développements

Nous allons maintenant développer les équations des commandes hybride parallèle et hybride externe où l'asservissement spatial est remplacé par une commande référencée capteurs dynamique. Pour chacune des lois de commande, nous présenterons différentes formulations en fonction du point de contrôle choisi par l'utilisateur.

4.2.1.1 Commande hybride parallèle

En s'inspirant de la formulation présentée dans l'état de l'art, l'équation (4.7) est réécrite de la manière suivante :

$$\tau = \mathbf{A}\mathbb{J}^+ (\mathbf{w}_s(t) - \mathbf{b}) + \mathbf{H} + \mathbf{J}^T \mathbf{w}_f(t) \quad (4.15)$$

L'équation (4.15) ne fait pas apparaître la matrice de sélection $\mathbf{S}$ nécessaire pour séparer la contribution de chaque asservissement dans sa direction respective. La forme de $\mathbf{S}$ est donnée par l'équation (4.6). On remarque aussi que, par rapport à l'état de l'art, les différentes quantités ne comportent pas de $(\hat{\cdot})$: on suppose dans cette section que l'ensemble des quantités pour le calcul la loi de commande sont estimées et ne correspondent vraisemblablement pas aux quantités réelles.

Étant donné que la matrice de sélection permet de sélectionner telle ou telle direction dans un seul et unique repère, il est donc nécessaire d'exprimer $\mathbf{S}$ dans la jacobienne de tâche $\mathbb{J}$. On rappelle la forme de $\mathbb{J}$ telle que $\mathbb{J} = \mathbf{L}_s^S \mathbb{T}_N \mathbf{J}$ (cf. section 2.2). Ci-après la sélection des directions où les deux asservissements vont s'appliquer s'effectue dans le repère de l'effecteur N, l'équation de découplage devient alors :

$$\tau = \mathbf{A}\mathbf{J}^{(-1)} \mathbf{S}^N \mathbb{T}_S \mathbf{L}_s^+ (\mathbf{w}_s(t) - \mathbf{b}) + \mathbf{H} + \mathbf{J}^T (\mathbf{I} - \mathbf{S}) \mathbf{w}_f(t) \quad (4.16)$$

Alors que l'inverse de la matrice d'interaction $\mathbf{L}_s$ est rendue possible en utilisant la pseudo-inverse, on utilise l'inverse généralisée pour inverser la matrice cinématique du robot $\mathbf{J}$. Connaissant la transformation rigide entre le repère de l'objet et le repère de l'effecteur, il est possible de faire intervenir la matrice de sélection directement dans le repère de l'objet, noté O ; ce qui donne pour τ l'expression suivante :

$$\tau = \mathbf{A}\mathbf{J}^{(-1)N} \mathbb{T}_O \mathbf{S}^O \mathbb{T}_N^N \mathbb{T}_S \mathbf{L}_s^+ (\mathbf{w}_s(t) - \mathbf{b}) + \mathbf{H} + \mathbf{J}^T \mathbb{F}_O (\mathbf{I} - \mathbf{S}) \mathbf{w}_f(t) \quad (4.17)$$

On note l'apparition de la matrice de transformation des efforts $\mathbb{F}_O$ du repère O vers le repère de l'effecteur N. Celle-ci, très similaire à la matrice de transformation cinématique, est définie telle que [Khalil and Dombre, 2002; Marchand et al., 2005; ViSP, 2016] :

$$\mathbb{F}_O = \begin{bmatrix} {}^N\mathbf{R}_O & \mathbf{0}_3 \\ [{}^N\mathbf{t}_O]_{\times} {}^N\mathbf{R}_O & {}^N\mathbf{R}_O \end{bmatrix} \quad (4.18)$$

L'expression de la commande référencée capteurs dynamique hybride parallèle se retrouve dans les équations (4.16) ou (4.17). La différence principale entre les deux expressions est que la matrice de sélection s'applique soit dans le repère de l'effecteur, soit dans celui de l'objet. Les deux ont leurs avantages et inconvénients.

Le principal avantage de la seconde expression est que le correcteur d'efforts associé à la quantité $\mathbf{w}_f(t)$ s'exprime dans le repère de l'objet. *I.e.* celui-ci permet de contrôler directement les efforts qui s'appliquent sur l'objet du point de vue de l'objet, ce qui paraît à première vue intéressant pour la manipulation d'un objet au contact. Néanmoins, l'objet ne comporte pas de capteur d'effort (dans le cas de robots manipulateurs, celui-ci est souvent installé au niveau du poignet et fait la jonction entre l'effecteur et le préhenseur), ce qui nécessite donc d'exprimer la quantité d'effort mesuré dans le repère de l'objet et implique par conséquent une nouvelle transformation (qui doit être mesurée). Dès lors, l'avantage sus-cité entraîne de manière quasi-automatique des inconvénients. L'information d'effort est par nature très sensible au bruit : la multiplicité des transformations entre les différents repères peut amplifier ce bruit et rendre cette partie du contrôleur

moins stable et donc, moins performante. Il en est de même pour la partie concernant l'asservissement spatial (qui, pour rappel, est réalisé par une commande référencée capteurs dynamique) dont la stabilité peut être affectée par ces multiples changements de repères. De plus, outre l'effet du bruit de mesure, ces transformations peuvent parfois comporter des erreurs. *In fine*, ce qui paraît être une bonne idée en théorie peut ne pas avoir d'effet positif en pratique.

Le lecteur peut trouver en annexe B une étude expérimentale sur l'impact d'une saisie d'un objet par un préhenseur et la qualité de cette saisie. Le but de l'étude a été de mesurer la transformation rigide entre le repère de l'objet et celui du préhenseur pendant les différentes phases d'une saisie. Il en ressort qu'une saisie entraîne une forte incertitude sur la pose de l'objet par rapport au repère du préhenseur.

4.2.1.2 Commande hybride externe

La commande hybride externe a permis d'améliorer la coexistence de deux asservissements : un spatial et l'autre en effort, en proposant un ré-agencement des deux boucles. Cette méthode a été utilisée avec succès avec la commande référencée capteurs où un asservissement visuel basé image (IBVS) a été utilisé en tant que boucle d'asservissement spatial interne (cf. paragraphe 4.1.3).

En s'aidant des équations (3.8), (4.10) et (4.11), la commande hybride externe en tenant compte de la commande référencée capteurs dynamique s'écrit de la manière suivante :

$$\tau = \mathbf{A}\mathbb{J}^+ (\mathbf{w}_s(t) - \mathbf{b}) + \mathbf{H} \quad (4.19)$$

avec,

$$\mathbf{w}_s(t) = \mathbf{K}_{sp} \left(\mathbf{s}^d - \mathbf{s} + \mathbf{y}(t) \right) - \mathbf{K}_{sv} \dot{\mathbf{s}} \quad (4.20)$$

La contribution du correcteur d'effort s'exprime à travers la quantité $\mathbf{y}(t)$. Le passage des efforts vers l'espace des caractéristiques n'est pas trivial et s'effectue de la manière suivante :

$$\mathbf{y}(t) = \mathbf{L}_s \mathbf{K}_r^{-1} \mathbf{w}_f(t) \quad (4.21)$$

La matrice $\mathbf{K}_r$ est équivalente à une matrice de raideur et peut être réglée selon le comportement désiré. À noter que l'expression ci-dessus fait que le correcteur associé à la quantité $\mathbf{w}_f(t)$ s'exprime ici dans le repère de l'effecteur. Tenant compte des mêmes remarques que pour la commande hybride parallèle, il est possible d'exprimer la contribution du correcteur d'efforts dans un autre repère (par exemple, celui de l'objet O) en utilisant l'expression suivante :

$$\mathbf{y}(t) = \mathbf{L}_s \mathbf{K}_r^{-1} {}^S\mathbb{F}_O \mathbf{w}_f(t) \quad (4.22)$$

On rappelle ici que le contrôle en effort est indirect. En effet, lorsqu'on se place dans le cas où l'asservissement spatial se trouve dans son état d'équilibre ($\mathbf{s} = \mathbf{0}$ et $\dot{\mathbf{s}} = \mathbf{0}$), l'expression du régulateur se réduit à la forme suivante :

$$\begin{aligned} \tau|_{\mathbf{e}_s=\dot{\mathbf{s}}=0} &= \mathbf{A}\mathbb{J}^+ \mathbf{L}_s \mathbf{K}_r^{-1} {}^S\mathbb{F}_O \mathbf{w}_f(t) \\ &= \mathbf{A}\mathbf{J}^{(-1)N} \mathbb{T}_S \mathbf{L}_s^+ \mathbf{L}_s \mathbf{K}_{sp} \mathbf{K}_r^{-1} {}^S\mathbb{F}_O \mathbf{w}_f(t) \\ &= \mathbf{A}\mathbf{J}^{(-1)N} \mathbb{T}_S \mathbf{K}_{sp} \mathbf{K}_r^{-1} {}^S\mathbb{F}_O \mathbf{w}_f(t) \end{aligned} \quad (4.23)$$

En plaçant les matrices des gains à l'identité et en fusionnant tous les repères, l'équation ci-dessus devient similaire à l'expression suivante :

$$\tau|_{\mathbf{e}_s=\dot{\mathbf{s}}=0} \sim \mathbf{A}\mathbf{J}^{(-1)} \mathbf{w}_f(t) \quad (4.24)$$

On s'aperçoit alors que la contribution de la correction en effort est équivalente à une accélération et non pas vue comme une contribution d'effort directe. En effet, l'expression ci-dessus ne peut être assimilée à une expression équivalente à $\tau = \mathbf{J}^T \mathbf{f}$.

4.2.2 Remarques préliminaires

Dans le cas d'une commande hybride parallèle, on peut s'attendre à ce que le choix des caractéristiques utilisées pour la correction spatiale prenne une place importante, voire prépondérante. Par exemple, l'utilisation de quatre points 2D observés depuis une caméra n'est pas susceptible de fournir de bonne performance. En effet, les contributions qui maintiennent les six degrés de liberté sont produites par un fort couplage entre les différentes caractéristiques (par exemple, une translation sur l'axe optique de la caméra correspond à un mouvement combiné et simultané des quatre points dans l'image). Lorsque l'asservissement spatial perd le contrôle d'une direction particulière pour être alloué au correcteur en effort, le problème de l'asservissement visuel devient mal posé et donc ne peut satisfaire l'objectif.

Dès lors, il est possible de dire que l'asservissement visuel par l'utilisation de la pseudo-inverse trouvera une position d'équilibre qui correspond au meilleur compromis entre les contributions de chacune des mesures. Cet argument tient si la contribution en effort n'emmène pas l'asservissement visuel trop loin de cette position d'équilibre, ou dans le pire des cas, à sa limite du domaine de stabilité. En pratique, l'utilisation de la commande hybride parallèle a déjà montrée que celle-ci pouvait provoquer des oscillations entre les deux points d'équilibre des deux asservissements [Khalil and Dombre, 2002; Villani and De Schutter, 2008].

Il est aussi possible de dire que ce problème d'oscillations est déjà réglé par la commande hybride externe. Néanmoins, comme exprimé plus haut, le fait que la contribution d'effort modifie la référence de l'asservissement spatial interne ne peut pas être assimilé à une contribution d'effort direct dans la loi de commande. Cependant, l'utilisation de la commande dynamique référencée capteurs fait que cette contribution du correcteur d'effort se rapproche le plus d'une force que lorsqu'un asservissement visuel cinématique tel que présenté dans [Prats et al., 2007] est utilisé.

4.3 Une étude de cas : positionnement d'une pièce encollée

Cette section se concentre sur l'étude de deux cas qu'il est possible de retrouver au sein d'un poste d'industrie manufacturière.

On va étudier un positionnement d'une pièce au contact après que celle-ci ait été encollée. On reconnaît ici une classe d'applications où les cas similaires sont la préparation d'une surface localisée ou le marquage en utilisant un outil électromécanique (par exemple, ponçage ou gravure). Cette classe d'applications se définit par la séquence d'opérations suivantes : prépositionner, positionner et maintenir avec une force constante selon une direction pendant un certain temps défini par l'utilisateur.

Les lois de commande proposées dans la section 4.2 résolvent le problème. L'asservissement spatial est assuré par une commande référencée capteur. Ainsi il donne la garantie que le positionnement est intrinsèquement réussi. La position de la pièce est contrainte par des références attachées à la structure et non déduite d'un modèle estimé de la réalité.

4.3.1 Mise en place de la simulation

Le robot est équipé d'un préhenseur, d'un capteur de mesure extéroceptif et d'un capteur d'effort monté sur son poignet. Les deux capteurs ont été calibrés et les poses respectives par rapport au repère de l'effecteur sont connues.

On suppose que la pièce a été saisie par le robot et que l'opération d'encollage (la dépose de colle sur la surface de la pièce en contact) a été effectuée. En suivant les recommandations issues de l'étude sur les erreurs de saisie (cf. annexe B), la pose de la pièce par rapport à l'effecteur a été mesurée après que les opérations préliminaires au positionnement au contact aient été effectuées. En connaissant la pose de la pièce par rapport à l'effecteur, on considère que l'effort mesuré est l'effort qui s'applique sur la pièce dans le repère de la pièce en utilisant l'équation (4.18).

En suivant la définition de la classe d'applications, l'opération veut que le robot prépositionne la pièce, va au contact et suit un profil d'application d'une force selon la normale de la surface utile. La première étape consiste donc à aligner la surface de la pièce avec celle de la structure. Ensuite, le robot va approcher la pièce de manière précise pour à la fin établir le contact. Une fois le contact établi, le profil d'effort est appliqué pour terminer l'opération.

Les deux premières étapes s'effectuent avec la commande dynamique référencée capteurs seule. Au moment de l'application du profil en effort, le système bascule vers la loi de commande hybride, parallèle ou externe. Les poses pièce-effecteur et caméra-effecteur sont connues. La trajectoire des caractéristiques est déterminée selon la trajectoire désirée de la pièce par rapport à la structure. La phase d'alignement est réalisée en s'appuyant sur une interpolation linéaire des caractéristiques dans l'image sans tenir compte d'aucune autre contrainte. L'hypothèse prise ici est que le mouvement ainsi généré ne comporte pas de difficulté particulière (pas de minimum local, ni de singularité à traverser). Par contre, la trajectoire des caractéristiques lors de la phase d'approche est calculée selon le mouvement désiré du repère de la pièce. On souhaite que celui-ci opère une descente verticale tout en gardant l'alignement. Le modèle de contact est un modèle de ressort linéaire où la constante de raideur est choisie telle que $\mathbf{K}_r = 2 \times 10^3 \text{ N.m}^{-1}$, ce qui correspond à la réaction de la compression de la colle entre la pièce et la structure. Il est possible de considérer que la surface encollée de la pièce est parfaitement parallèle à la surface de la structure, on peut donc dire que le problème du contact du point de vue du contrôleur peut être ramené à un contact ponctuel sans frottement dont le comportement est régi par un modèle type ressort linéaire.

4.3.2 Marqueur sur structure et contrôle d'effort direct

On se place dans un cas où la structure comporte une cible avec quatre points 2D. Les points 2D sont donc les caractéristiques choisies pour l'asservissement spatial. L'emplacement final de la pièce est relative à la position de la cible sur la structure. Les différentes étapes de l'opération sont illustrées en figure 4.5. On note que les états des caractéristiques pour la pose intermédiaire (bleu) et la pose au contact (vert), figures 4.5d et 4.5e, sont déterminés en s'appuyant sur la connaissance de la géométrie de la structure, des transformations spatiales effecteur-pièce et effecteur-capteur et de la pose désirée de la pièce sur la structure. Seul l'état initial (en rouge en 4.5d) est mesuré.

Interprétation des résultats de simulation Pendant les deux premières phases, le robot se comporte comme attendu : tenant compte des mesures de la référence attachée à la structure, le robot aligne la pièce puis descend verticalement pour entrer en contact. Seul l'asservissement spatial

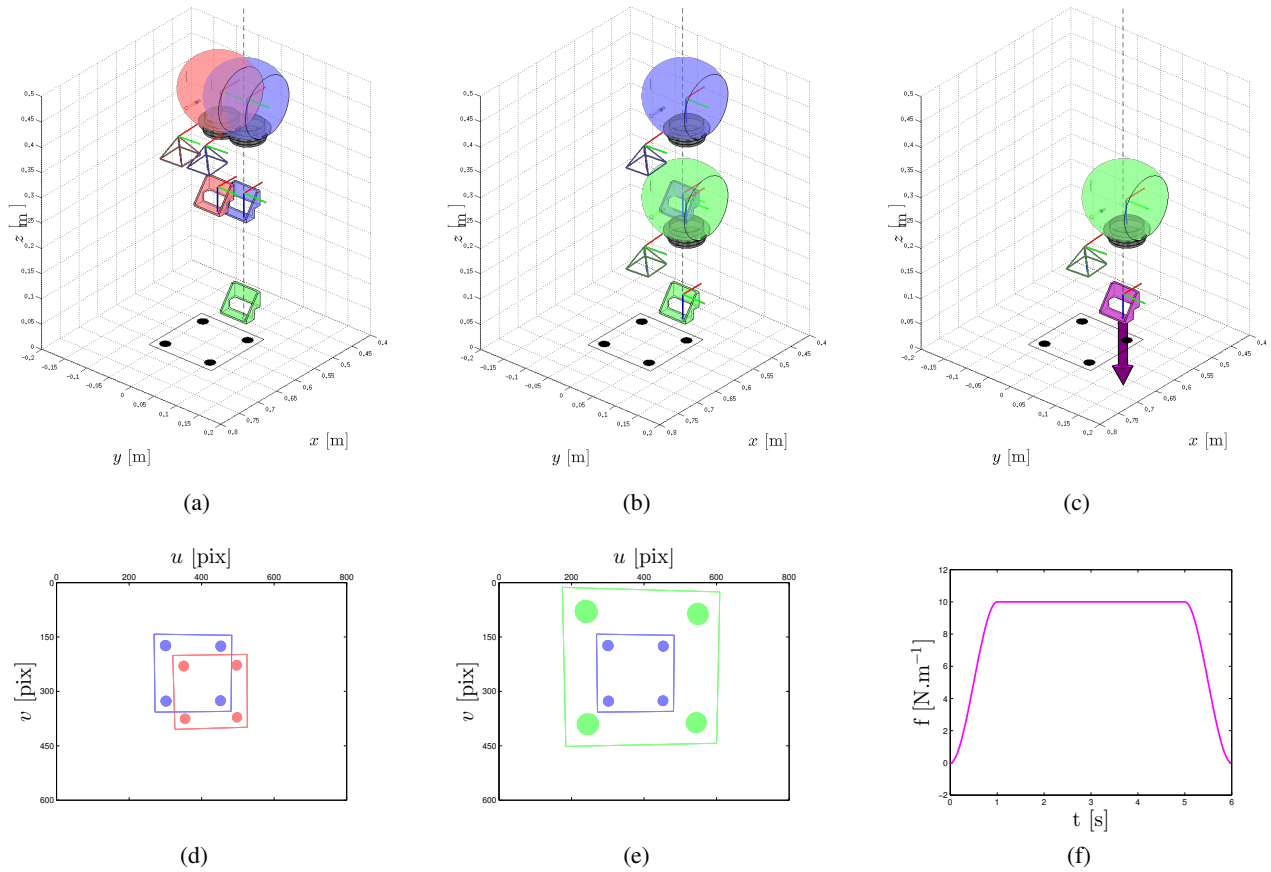


FIGURE 4.5 – Prépositionnement et positionnement au contact d'une pièce encollée en se basant sur un marqueur présent sur la structure comportant quatre points : (a) première phase : le système évolue de sa position initiale (en rouge) vers une position intermédiaire où la pièce est alignée avec la structure ; (b) seconde phase : depuis la position intermédiaire (en bleu), le système descend vers la position d'entrée en contact (en vert) ; (c) dernière phase : entrée en contact et application du profil d'effort selon la direction souhaitée. La seconde ligne illustre les stimuli capteurs utilisés par les asservissements pour chaque phase. Le corps de la caméra, son support et le système de préhension de la pièce ne sont pas représentés.

assuré par la commande référencée capteurs est activé durant ces phases et le comportement de l'asservissement est similaire aux remarques effectuées en section 3.3.

Lors de la phase de contact, le basculement entre la commande dynamique référencée capteurs et la commande hybride parallèle est atomique (*i.e.* il n'y a pas de délai pour l'application de la seconde loi de commande). Le moment qui marque la transition entre les phases 2 et 3 est $t_{p3} = 7.4$ s, indiqué par la ligne verticale grise sur les graphes d'évolution temporelle. Le profil d'effort est alors joué alors que l'asservissement spatial doit tenir la position (*i.e.* rester sur la cible verte en figure 4.5e).

Les résultats de simulation sont reportés en figures 4.6 et 4.7. Le basculement entre les deux lois de commande n'entraîne pas de comportement particulier. Sur la figure 4.6c, on observe que le profil d'effort mesuré suit globalement la référence : l'erreur est non nulle seulement pendant la période transitoire de la phase d'effort. La valeur maximale de l'erreur dans l'espace de la force est de l'ordre de 0.4 N. L'erreur est nulle lorsque la référence se stabilise à 10 N pendant 5s. On observe de petites oscillations lorsque la valeur atteint le plateau à $t \simeq 8.5$ s (figure 4.6d). Ces oscillations peuvent être contenues par l'ajustement des gains du correcteur en effort, notamment par l'augmentation de $\mathbf{K}_{vf}$ dans la limite de son domaine de stabilité.

Ici, le modèle de contact utilisé dans la simulation fait que la pièce s'enfonce de manière infinitésimale sous la surface de la structure selon l'axe z de la pièce. Le but de l'asservissement spatial ici est de maintenir la pièce dans l'axe. Mais la nature de la commande référencée capteur fait que celle-ci se stabilise autour d'un minimum local qui minimise toutes les erreurs dans l'espace capteur. Au contact, le déplacement infinitésimal dû à l'asservissement en effort entraîne une erreur dans l'espace capteur que l'asservissement visuel contraint tant bien que mal. En figure 4.7b, on reporte l'erreur de positionnement du point de contact sur la surface entre celui attendu et celui mesuré pendant l'opération. À travers ce résultat, on observe que l'asservissement visuel se stabilise en satisfaisant toutes les contraintes dans l'espace des caractéristiques autour d'un nouveau minimum local qui n'est pas celui espéré. Néanmoins, le glissement engendré est inférieur au dixième de millimètre. On peut dire que ce glissement est négligeable, car notre modèle de contact ne prend pas en compte les frottements de la pièce sur la surface. En pratique, le glissement observé en simulation sera très probablement contraint par ces efforts de frottement.

L'évolution temporelle des couples articulaires de consigne, figure 4.7a, donne une idée de la manière dont le robot se comporte pour réaliser l'action demandée. À la courbe, sont ajoutés les valeurs des couples lorsque le correcteur d'effort est encore inactif (en pointillés). En prenant l'articulation A2, on observe que le couple τ_{A2} baisse d'intensité du fait de l'action du correcteur d'effort. A2 est l'articulation – dans la configuration actuelle – qui supporte l'ensemble du poids du robot : le couple initial avant l'entrée en jeu du correcteur d'effort correspond au couple de contre-réaction généré par le poids des corps du robot (via la quantité $\mathbf{H}$ dans l'équation (4.16)). La baisse d'intensité signifie que le robot utilise son propre poids pour réaliser l'objectif du correcteur d'effort. Le phénomène observé sur l'articulation A3 est similaire.

Commentaires Nous avons simulé un pré-positionnement, puis un positionnement au contact avec un profil désiré de force à suivre pendant la phase du contact où le robot est contrôlé par une commande hybride parallèle avec un asservissement visuel dynamique et un asservissement en effort.

À la vue des hypothèses prises pour la simulation et du comportement du robot pendant celle-ci, il est possible de considérer que la loi de commande hybride parallèle avec pour asservissement

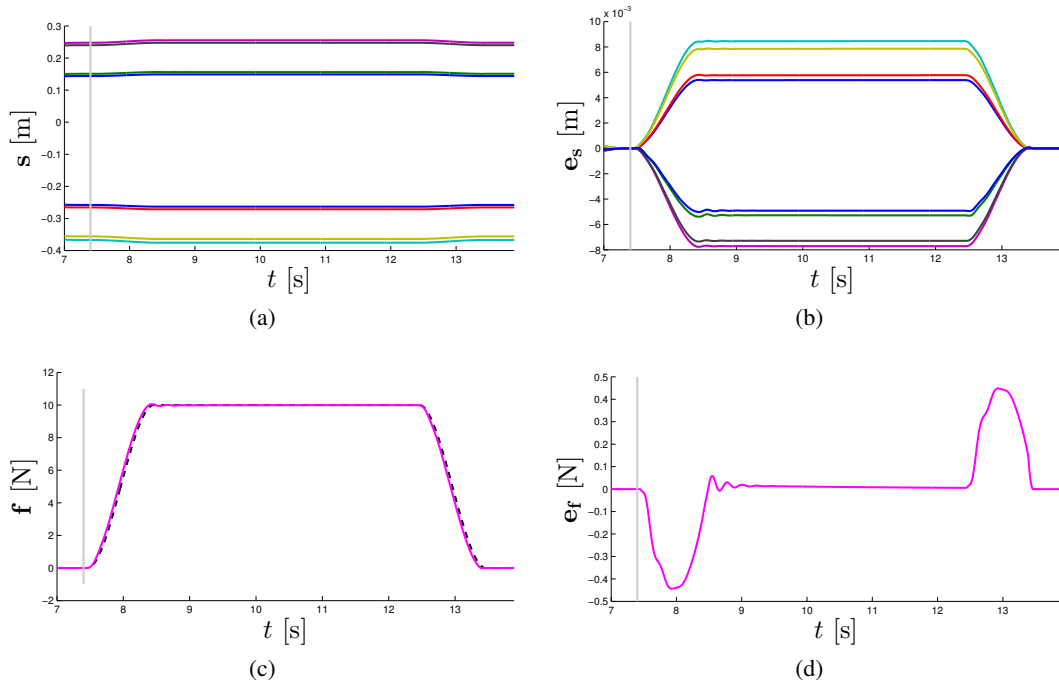


FIGURE 4.6 – Commande hybride parallèle : évolutions temporelles des mesures pour l’asservissement spatial et l’asservissement en effort au cours de la troisième phase; (a)(b) respectivement mesures des point 2D depuis la caméra et l’erreur; (c)(d) idem pour la force selon l’axe z du repère de la pièce, de la pièce qui s’applique sur la structure. La ligne verticale grise marque le début de la troisième phase.

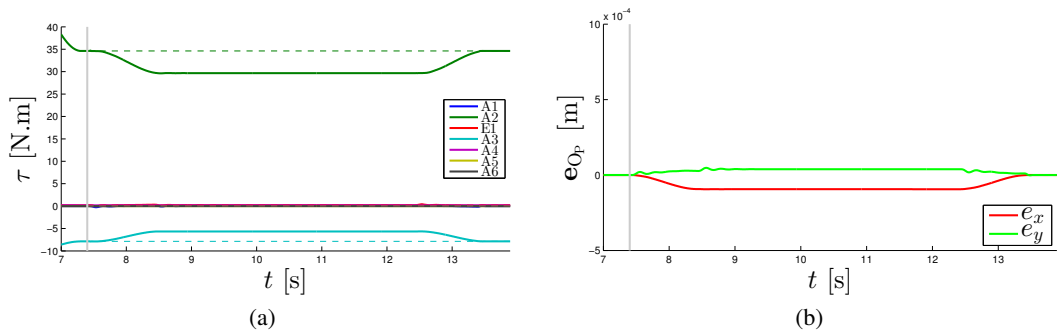


FIGURE 4.7 – Commande hybride parallèle : (a) évolution temporelle des couples articulaires de consigne; (b) l’erreur de position de l’origine du repère de la pièce entre celle attendue et celle mesurée dans le plan formé par la surface de la structure centrée au point de contact.

spatial interne une commande dynamique référencée capteurs, est suffisante pour réaliser une opération de positionnement au contact d'une pièce encollée.

Il est possible de penser que l'asservissement visuel ne peut pas s'appliquer lorsque celui-ci ne contrôle pas l'ensemble des degrés de liberté de l'effecteur/pièce. À l'inverse, la simulation montre que la boucle de régulation interne réagit plutôt bien et n'entre pas en conflit avec l'asservissement en effort.

Lorsque la pièce est au contact, on observe que celle-ci glisse sur la surface de la structure. On peut en conclure que même si la commande dynamique référencée capteurs ne rentre pas en conflit avec l'asservissement en effort, celle-ci n'est plus en mesure d'apporter la garantie que le positionnement est réalisé puisque cette garantie est valide lorsque la commande contraint l'ensemble des degrés de liberté. Néanmoins, la quantité de glissement est négligeable et le sera encore plus dans une expérimentation réelle.

4.3.3 Contrôle en effort indirect : commande hybride externe

Le cadre de l'étude est le même que précédemment. L'opération se déroule toujours en trois phases distinctes : alignement de la pièce avec la structure, approche précise et contrôlée par stimuli extéroceptifs puis l'application d'un profil de force selon une direction particulière une fois au contact. Seule la commande du robot pour la troisième phase a été modifiée. En effet, le cas de figure a été simulé cette fois en se basant sur les équations (4.19), (4.20) et (4.22). Le déroulement de l'opération est illustré en figure 4.5.

De même que dans le paragraphe précédent, la commande du robot durant les deux premières phases est assurée par une commande référencée capteurs dynamique. Il n'y a pas d'observation particulière à relever si ce n'est que la commande hybride externe aurait pu être utilisée tout au long de l'opération.

Pour rappel, la caractéristique principale de la commande hybride externe est de modifier la référence de la boucle d'asservissement spatial interne. Durant les deux premières phases, l'utilisateur n'applique pas d'effort et le robot ne rencontre pas d'obstacle. Autrement dit, l'utilisateur ne demande pas d'interagir et le robot ne voit pas d'interaction pendant sa phase de mouvement pendant l'alignement et l'approche. Il n'y a donc pas de raison pour que la référence s^d soit modifiée pendant ces phases.

Résultats de simulation Les résultats de simulation sont reportés en figures 4.8 et 4.9. Les évolutions temporelles ne montrent que le comportement du système durant la troisième phase. A $t_{P3} = 7.4$ s, un basculement de la commande dynamique référencée capteurs vers la commande hybride externe est effectué.

On observe globalement le même comportement qu'avec la commande hybride parallèle. Le profil en effort est globalement respecté et l'erreur d'effort possède la même forme qu'auparavant. Puisque le correcteur en effort vient modifier la référence de la boucle interne, on observe en figure 4.8b que les caractéristiques réagissent de manière légèrement différente qu'en figure 4.6b. De plus, nous n'observons pas d'oscillation sur le profil en effort comme en figure 4.6d.

Même si d'un point de vue capteur, le comportement du système est globalement identique à ce qu'on peut observer dans la simulation précédente, le comportement interne du robot est différent. Les couples articulaires de consigne sont reportés en figure 4.9a : on observe que le robot exploite son poids pour réaliser l'action demandée. Les différences majeures se situent pendant la période

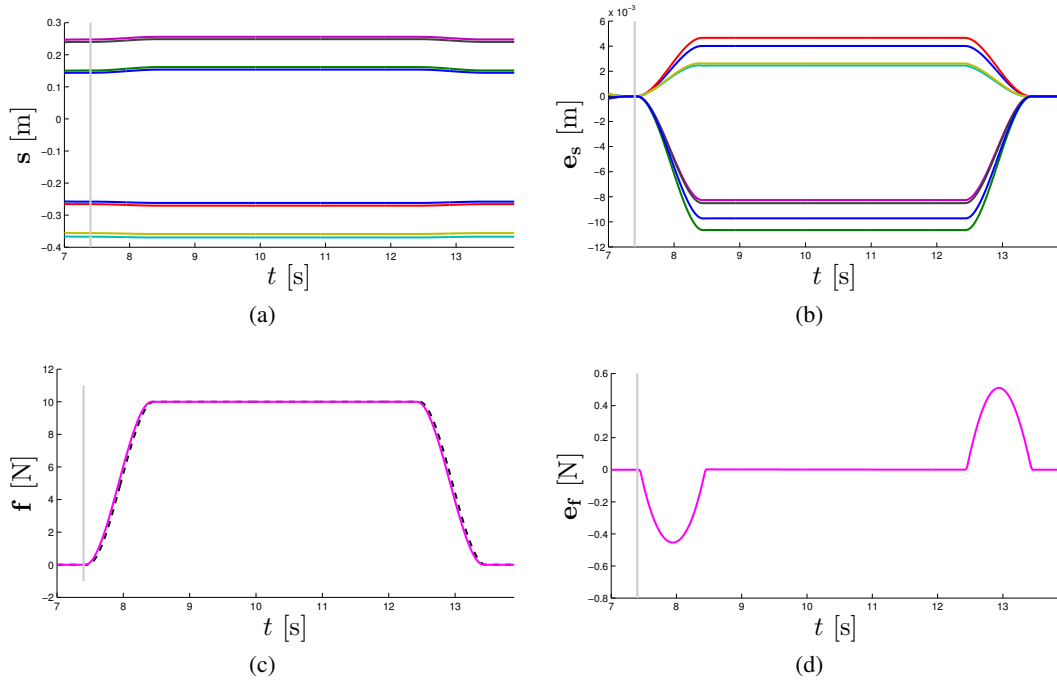


FIGURE 4.8 – Commande hybride externe : évolutions temporelles des mesures pour l’asservissement spatial et l’asservissement en effort au cours de la troisième phase ; (a)(b) respectivement mesures des point 2D depuis la caméra et l’erreur ; (c)(d) idem pour l’effort selon l’axe z du repère de la pièce. La ligne verticale grise marque le début de la troisième phase.

transitoire de l’action d’effort. On observe aussi que l’action du correcteur d’effort entraîne une contribution des articulations A2 et A3 moindre qu’avant lorsque l’effort a atteint son plateau : à $t \simeq 10$ s, τ_{A2} vaut 29.67 N.m pour la commande parallèle et vaut 31.11 N.m pour la commande externe. De plus, les articulations A1 et E1 deviennent actives.

En figure 4.9b, on s’aperçoit que le glissement de la pièce sur la structure existe toujours. Celui-ci est plus important que dans l’étude précédente mais est toujours inférieur au dixième de millimètre. De même, ce glissement est rendu possible en simulation car les frottements ne sont pas intégrés au modèle de contact. En pratique, ce glissement serait annulé par les frottements éventuels.

Commentaires Le système se comporte globalement de manière identique dans les deux cas. Les seules différences sont à relever au niveau du comportement interne du robot. L’observation est logique car la contribution du correcteur d’effort ne s’applique pas aux mêmes endroits au niveau de l’équation de découplage.

4.4 Conclusions

Le but principal du chapitre a été de fusionner la commande dynamique référencée capteurs avec des lois de commande hybride pour la gestion des interactions avec l’environnement extérieur sur la base d’une lecture des informations d’effort.

Pour rappel, la loi de commande dynamique référencée capteurs a été proposée pour mettre à

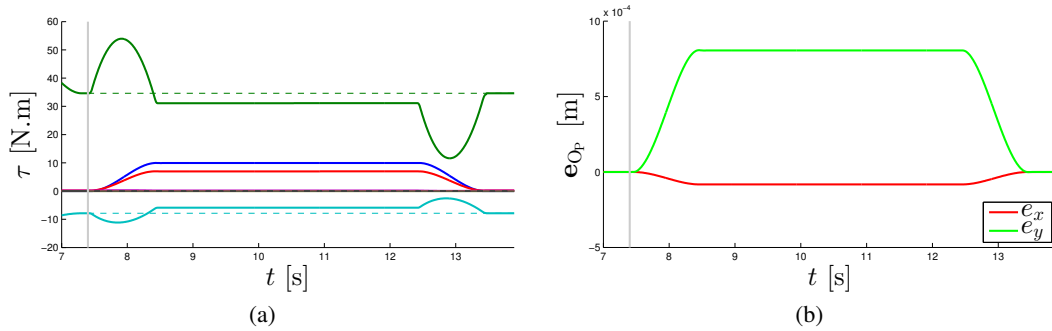


FIGURE 4.9 – Commande hybride externe : (a) évolution temporelle des couples articulaires de consigne; (b) l'erreur de position de l'origine du repère de la pièce entre celle attendue et celle mesurée dans le plan formé par la surface de la structure centrée au point de contact.

niveau la commande référencée capteur cinématique. Ainsi, notre approche a permis à l'utilisateur de contrôler la position et la vitesse des caractéristiques en boucle fermée dans l'espace capteurs. L'utilisateur peut ainsi contrôler de manière générale le comportement du robot en s'appuyant sur des références attachées au monde réel plutôt que de s'appuyer sur une estimation biaisée. Cependant, pour apporter une solution au problème de positionnement précis au contact dans le cadre d'un assemblage réalisé par un robot, cette proposition reste incomplète. Pour compléter celle-ci, le contrôle en effort à travers les commandes hybrides permet au robot une gestion de l'interaction avec son environnement.

L'état de l'art s'est concentré sur les principales lois de commande qui permettent au robot d'appliquer un effort ou de réagir en cas de contact avec l'environnement extérieur.

Deux lois de commande ont été revisitées : la loi de commande hybride parallèle et la loi de commande hybride externe. Ces deux lois ont été réécrites en tenant compte que la partie asservissement spatial était assurée par une commande dynamique référencée capteurs. De plus, différentes écritures de ces lois ont été proposées pour faire apparaître la possibilité d'exprimer telle ou telle action dans des repères choisis. On retiendra par exemple l'équation (4.17) qui rend possible d'exprimer l'action du correcteur en effort dans le repère de l'objet, ce qui la rend intéressante car le contrôle en effort s'applique directement dans le repère du corps au contact.

Les nouvelles formulations des lois de commande hybride ont fait l'objet d'une validation par simulation à travers un cas d'étude qui est représentatif d'une opération que l'on peut trouver dans une industrie manufacturière : un positionnement précis d'une pièce manufacturée au contact ayant pour objectif de maintenir la pièce sous pression pendant un certain temps. Dans les conditions du cas d'étude et les hypothèses prises, les simulations ont montrées que les deux formulations produisent des résultats comparables. Mais avant tout, celle-ci réalise l'opération souhaitée : le profil d'effort est réalisé et la pièce est maintenue en position selon les références de la structure.

En conclusion, les développements de ce chapitre démontrent la validité de la commande référencée capteurs et sa variante dynamique pour apporter une solution complète au problème du positionnement précis au contact d'une structure. En effet, la commande dynamique s'instancie bien avec les commandes hybrides et valide la possibilité du contrôle en effort. De plus, le cas étudié reste un cas témoin et il est tout à fait possible d'imaginer que les commandes hybrides développées ici peuvent apporter une solution complète à tout type d'opération de type perçage,

vissage, soudage, ponçage que l'on peut trouver dans une usine manufacturière.

Les perspectives de travail autour de la commande hybride sont nombreuses. En effet, l'étude se limite à l'application des principales commandes hybrides. Néanmoins, l'approche est aussi compatible avec une commande multi-tâches dynamique où les tâches avec des contraintes spatiales et avec des contraintes d'effort peuvent se combiner par l'utilisation d'un projecteur orthogonal. En se projetant plus loin, il serait intéressant de réfléchir à une méthode pour la combinaison des tâches en effort et des tâches spatiales sur l'exemple du projecteur orthogonal dynamique, en apportant le contrôle nécessaire à l'utilisateur pour que le robot réalise l'opération demandée.

Plus pragmatiquement, ce chapitre a montré que tous les outils sont maintenant disponibles pour réaliser une opération d'assemblage tenant compte d'objectifs de précision autant spatiale qu'en effort. Il est maintenant tout aussi important d'implémenter ces méthodes pour les appliquer à des problèmes réels.

Conclusion générale

Dans cette thèse, nous nous sommes intéressés au travail des opérateurs à travers la problématique particulière : la manipulation d'objets manufacturés référencée capteurs. Certaines tâches dans les industries manufacturières ne peuvent être encore automatisées, même dans les usines les plus technologiquement avancées, car la technologie n'apporte pas encore de solutions satisfaisantes.

La volonté de produire plus, d'augmenter la productivité et la rentabilité des entreprises entraînent des conséquences qui font que les usines doivent fournir plus d'effort et d'innovation. La spécialisation des postes de travail, l'outillage mécanisé ou l'automatisation complète sont des moyens qui tirent la productivité vers le haut. Mais ces moyens apportent leurs inconvénients : troubles musculo-squelettiques, perte de polyvalence et déficit d'intérêt de l'opérateur à son travail. L'opérateur possède de nombreuses facultés qui le rend encore aujourd'hui indispensable à la chaîne de production, en particulier dans les lignes de montage et d'assemblage. Le travail qu'il effectue n'a pas d'équivalent technologique et ne peut être remplacé par une quelconque machine.

Le projet ASIMOV, dans lequel s'inscrit cette thèse, affiche la volonté d'apporter de nouveaux outils aux opérateurs d'Airbus pour les soulager de certaines tâches dites à faible valeur ajoutée. Le placement de petites pièces de fixation (de 2 cm à 10 cm d'envergure) destinées à soutenir tout ou partie des systèmes internes d'un avion a été identifié comme faisant partie des tâches de cette catégorie. En effet, celles-ci requièrent une attention accrue de l'opérateur pour leurs positionnements (leurs positions ont été soigneusement choisies) et la multiplicité des pièces fait que l'opération devient chronophage à l'échelle d'un avion. Cette problématique de positionnement de petites pièces pour l'assemblage d'un plus grand ensemble est partagée par de nombreux acteurs liés à l'industrie manufacturière. Le constat y est identique : la technologie et les robots n'apportent pas de solutions satisfaisantes pour l'automatisation de ces tâches.

Ce travail autour de la problématique de la manipulation d'objets manufacturés s'est concentré principalement sur une tâche d'assemblage et ses principales actions. Mais les outils développés ont vocation à s'instancier avec tout autre type d'application qu'il est possible de rencontrer dans une usine manufacturière.

Le titre de la thèse fait référence à la manipulation d'objets en se basant sur l'utilisation de la commande référencée capteurs. La commande référencée capteurs est connue pour apporter une

solution au problème de positionnement absolu des bras manipulateurs en les contrôlant avec des mesures extéroceptives, images d'éléments d'intérêt de l'environnement.

Notre travail de recherche s'est limité au concept que nous avons appelé la macro-manipulation. Le terme manipulation fait souvent référence à la manipulation d'objet par un préhenseur de type main bio-inspirée (manipulation dextre) où une des difficultés principales est de déterminer comment la pièce doit être saisie et comment celle-ci peut être déplacée sans la faire tomber (*i.e.* manipuler quelque chose de la même manière que le ferait un humain avec sa main). Néanmoins, le fait de manipuler une pièce dans un environnement complexe sous objectifs et/ou contraintes, requiert de nombreuses autres informations qui n'apparaissent pas dans la définition habituelle de la manipulation. Ces tâches (aller chercher la pièce, l'amener dans sa position souhaitée, faire un positionnement précis au contact...) entrent dans un cadre étendu de la manipulation en général, d'où le terme macro-manipulation.

À partir de cette définition, nous avons proposé la définition d'un nouveau cadre de programmation robotique pour déterminer le mouvement du bras manipulateur en tenant compte de la lecture des informations de son environnement réel, de ses capteurs extéroceptifs et des maquettes numériques à disposition. Le cadre en question s'appuie très largement sur l'utilisation de la commande référencée capteurs et les lois de commandes hiérarchisées multi-tâches. Nous avons apporté une solution de positionnement précis en se basant sur la situation réelle de la structure et non plus par rapport à sa position estimée issue d'un modèle virtuel. L'approche proposée se veut systématique à tous les capteurs et toutes les situations.

Nous nous sommes concentrés ensuite sur le contrôle de la vitesse des caractéristiques dans l'espace capteurs pour apporter une solution à d'autres classes d'application que l'assemblage. En effet, la commande référencée capteur est une loi de commande qui permet un positionnement quasi-statique par rapport à une référence mesurée. Néanmoins, en l'état actuel, celle-ci ne permet pas de prédire comment le robot se comporte pendant la convergence. Nous avons donc introduit la commande dynamique référencée capteurs. Cette nouvelle loi de commande a été proposée pour combler le manque du contrôle en vitesse de la commande référencée capteurs. Grâce à cette commande, l'utilisateur est en mesure de contrôler le comportement du bras manipulateur et ainsi réaliser un contrôle en position et en vitesse dans les meilleures conditions.

Enfin, notre dernière proposition s'est focalisée sur la prise en compte des informations issues d'un capteur d'effort. Nous avons revisité les schémas de commandes hybrides parallèle et externe en montrant qu'ils étaient compatibles avec notre approche dynamique. Cette dernière contribution complète notre solution à la problématique de positionnement précis d'une pièce au contact.

Ces points ont été validés par simulation avec des environnements différents. Nous avons utilisé successivement des environnements *ad hoc* construits pour valider les différentes approches sous Matlab, Gazebo et MSC Adams. Plusieurs modélisations dynamiques pour le comportement du robot ont été utilisées et quelques premières briques du cadre de programmation proposé en premier lieu ont été développées en vue d'un déploiement d'une solution industrielle sur site. Enfin, la mise en place des lois de commande dynamique a été une des difficultés principales de ce travail de thèse. De nombreux essais ont été effectués pour appliquer une commande par couples calculés sur le KUKA LWR4+ afin de permettre une validation expérimentale du positionnement précis au contact.

La perspective principale de ce travail de thèse est bien évidemment l'implémentation de ses différentes composantes pour la création d'un logiciel professionnel. Les enjeux de ce travail d'implémentation sont tout aussi importants que la continuation du travail de recherche. La programmation robotique porte le désavantage d'être réservée aux personnes initiées ayant un fort

niveau de compétence. Un des bénéfices nets de notre approche est qu'elle évite d'avoir recours à des modèles virtuels qui sont parfois difficiles à obtenir et souvent biaisés. Le fait de déterminer le mouvement d'un robot en tenant compte d'une référence du monde réel apportera une meilleure lisibilité pour la programmation de celui-ci. Le but ici est double : il est nécessaire de fournir un ensemble de logiciels opérationnels et robustes pour les usines manufacturières et plus largement, les secteurs industriels qui souhaitent s'équiper de robots collaboratifs ; il est tout aussi nécessaire de rendre la programmation simple pour la rendre plus accessible.

La commande dynamique référencée capteurs repose sur l'hypothèse forte que le capteur est capable de renseigner le signal en même temps que sa variation associée. Une seconde perspective serait de réfléchir à une architecture de capteur capable de mesurer à la fois le signal et sa variation sans avoir recours à des méthodes de différenciation numérique et autres méthodes de filtrage numérique. Les travaux sur le capteur de vision dynamique (DVS) établissent un bon point de départ sur les nouvelles architectures électroniques pour la captation de ce genre de signaux physiques. L'inconvénient de ce capteur en particulier est que la méthode de mesure rend le signal très granulaire, ce qui rend très difficile l'extraction d'une structure de donnée quelconque. Une troisième et dernière perspective est de revoir la formalisation du hessien, en particulier pour les caractéristiques appartenant à la famille 2D. Le but ici est de trouver une forme du hessien dans le modèle du second ordre qui limiterait au maximum les dépendances à des paramètres qui n'ont pas lien directe avec la caractéristique.

En conclusion, nous estimons que les éléments scientifiques et expérimentaux qu'apportent ce travail de thèse contribuent de manière significative à la problématique du positionnement précis de pièces manufacturées par un bras manipulateur. L'approche se veut aussi facile d'utilisation en abstrayant la majeure partie de la mise en place d'une commande référencée capteur tout en garantissant que le mouvement réalisera la tâche d'assemblage demandée. L'approche rend l'utilisateur capable de contrôler le bras manipulateur directement dans son environnement réel et ainsi supprime une partie des problèmes liés aux écarts entre les modèles virtuels estimés et le monde réel.

Dans le futur, il est certain que tous les robots manipulateurs utiliseront ces techniques et n'auront plus recours aux modèles virtuels pour se mouvoir. A la manière des humains, le monde réel n'existera qu'à travers leurs yeux.

Annexes



Matrices d'interaction pour le modèle du second ordre

Cette annexe fait suite aux développements effectués dans la section 3.2.2. Dans un premier temps, on rappelle les développements de la mécanique générale pour la dérivation composée, la loi de composition des vitesses ainsi que celle pour les accélérations. On présente ensuite les méthodes pour obtenir les matrices d'interaction du premier et du second ordre pour les caractéristiques de type ligne 3D, plan 3D, point 2D et ligne 2D.

A.1 Rappel de mécanique générale

A.1.1 Dérivation composée

Cette section reprend les éléments développés par Bône et al. [1984, page 3, section 2].

Soient deux espaces vectoriels orientés E et F de dimension 3, $\vec{\phi}(t)$ un vecteur fonction du temps exprimé dans E, $L(t)$ une application dans l'ensemble des applications isométriques positives de E sur F. On note $\vec{\Psi}(t)$ un vecteur exprimé dans F définie par :

$$\vec{\Psi}(t) = L(t) \vec{\phi}(t) \quad (\text{A.1})$$

La dérivée de l'équation (A.1), selon les théorèmes de mathématiques classiques, est définie par :

$$\frac{d}{dt} \vec{\Psi}(t) = \frac{d}{dt} L(t) \vec{\phi}(t) + L(t) \frac{d}{dt} \vec{\phi}(t) \quad (\text{A.2})$$

Soit $b_E(\vec{e}_1, \vec{e}_2, \vec{e}_3)$ une base orthonormée directe de E. Quelque soit t , $L(t)$ transforme la base $b_E \in E$ en une base orthonormée directe de F :

$$B_E(t)(\vec{E}_1, \vec{E}_2, \vec{E}_3) \in F \quad (\text{A.3})$$

Si $\vec{\phi}(t)$ s'écrit

$$\phi(t) = \phi_1(t) \vec{e}_1(t) + \phi_2(t) \vec{e}_2(t) + \phi_3(t) \vec{e}_3(t) \in E \quad (\text{A.4})$$

$\vec{\Psi}(t)$ s'écrit alors,

$$\vec{\Psi}(t) = \phi_1(t)\vec{E}_1(t) + \phi_2(t)\vec{E}_2(t) + \phi_3(t)\vec{E}_3(t) \in F \quad (\text{A.5})$$

et,

$$\dot{\vec{\Psi}}(t) = \dot{\vec{\phi}}(t)\vec{E}(t) + \phi(t)\dot{\vec{E}}(t) \in F \quad (\text{A.6})$$

En comparant (A.1) et (A.6), on observe :

$$L(t)\dot{\vec{\phi}}(t) = \dot{\phi}(t)\vec{E}(t) \quad \text{et} \quad \dot{L}(t)\vec{\phi}(t) = \phi(t)\dot{\vec{E}}(t) \quad (\text{A.7})$$

On sait que $L(t)$ possède la propriété :

$$L(t)^\top L(t) = \mathbf{I} \quad (\text{A.8})$$

En dérivant,

$$\dot{L}(t)^\top L(t) + L(t)^\top \dot{L}(t) = 0 \quad (\text{A.9})$$

En prémultipliant par $L(t)$, on obtient $\dot{L}(t) = -L(t)^\top \dot{L}(t) L(t)$. Puis une nouvelle fois par $L(t)^\top$, $\dot{L}(t)^\top L(t) + L(t)^\top \dot{L}(t) = 0$. L'équation (A.9) nous indique alors que la quantité $-L(t)\dot{L}(t)^\top$ correspond à un opérateur antisymétrique sur F et peut s'écrire sous la forme $\vec{\Omega}_{E/F}(t) \times (\cdot)$. Ainsi,

$$\dot{L}(t)\vec{\phi}(t) = \vec{\Omega}_{E/F}(t) \times L(t)\vec{\phi}(t) = \vec{\Omega}_{E/F}(t) \times \vec{\Psi}(t) \quad (\text{A.10})$$

Et, de fait,

$$\dot{\vec{\Psi}}(t) = L(t)\dot{\vec{\phi}}(t) + \vec{\Omega}_{E/F}(t) \times \vec{\Psi}(t) \quad (\text{A.11})$$

Les remarques données dans l'ouvrage à ce stade sont les suivantes :

- l'application $L(t)$ permet de définir la mobilité de l'espace E relativement à F ;
- $\dot{\vec{\Psi}}(t)$ est appelée dérivée relativement à F , notée $\left. \frac{d}{dt} \vec{\Psi}(t) \right|_F$ et $L(t)\dot{\vec{\phi}}(t) = \dot{\phi}(t)\vec{E}(t)$ est appelée, par abus de langage, dérivée de $\vec{\Psi}$ relativement à E , notée $\left. \frac{d}{dt} \vec{\Psi}(t) \right|_E$.

Avec ces notations :

$$\left. \frac{d}{dt} \vec{\Psi}(t) \right|_F = \left. \frac{d}{dt} \vec{\Psi}(t) \right|_E + \vec{\Omega}_{E/F}(t) \times \vec{\Psi}(t) \Big|_F \quad (\text{A.12})$$

Commentaire L'équation ci-dessus constitue la base de la mécanique générale et la cinématique du point vue d'un repère mobile. Cette notation est reprise dans l'ensemble des ouvrages destinés à l'apprentissage de la mécanique.

L'application $L(t)$ correspond à une matrice de changement de base et plus précisément à une matrice de rotation ${}^F\mathbf{R}_E \in \text{SO}(3)$, fonction du temps.

À mon sens, la notation ci-dessus est incomplète. Cette notation est valide lorsque les espaces utilisés sont vectoriels. Celle-ci dessert l'utilisateur lorsque que celui-ci a besoin d'exprimer les valeurs dans telle ou telle base. En particulier, la notation rend implicite – autrement dit, elle cache – l'application de L dans le terme $\left. \frac{d}{dt} \vec{\Psi}(t) \right|_E$ par un abus de langage largement répandu.

Sachant que $\vec{\Psi}(t) = L(t)\vec{\phi}(t)$, la notation ne devrait pas s'acquitter de l'application de L . J'écris alors :

$$\left. \frac{d}{dt} \vec{\Psi}(t) \right|_F = L(t) \left. \frac{d}{dt} \vec{\phi}(t) \right|_E + \vec{\Omega}_{E/F}(t) \times \vec{\Psi}(t) \Big|_F \quad (\text{A.13})$$

A.1.2 Loi de composition des vitesses

Soient W un repère fixe, S un repère mobile par rapport à W et P un point mobile dans W . On exprime la vitesse du point P par rapport à W et tenant compte de la mobilité de S dans W , par :

$$\begin{aligned} \left. \frac{d}{dt} \overrightarrow{O_W P} \right|_W &= \left. \frac{d}{dt} \overrightarrow{O_W O_S} \right|_W + \left. \frac{d}{dt} \overrightarrow{O_S P} \right|_W \\ &= \left. \frac{d}{dt} \overrightarrow{O_W O_S} \right|_W + \left. \frac{d}{dt} \overrightarrow{O_S P} \right|_S + \left. \vec{\Omega}_{S/W} \right|_W \times \left. \overrightarrow{O_S P} \right|_W \end{aligned} \quad (A.14)$$

L'équation (A.14) constitue l'équation fondamentale de la composition des vitesses. On reconnaît dans l'équation (A.14) :

- $\left. \frac{d}{dt} \overrightarrow{O_W P} \right|_W$ la vitesse absolue de P dans W ;
- $\left. \frac{d}{dt} \overrightarrow{O_W O_S} \right|_W + \left. \vec{\Omega}_{S/W} \right|_W \times \left. \overrightarrow{O_S P} \right|_W$ la vitesse d'entraînement de S dans W ;
- $\left. \frac{d}{dt} \overrightarrow{O_S P} \right|_S$ la vitesse relative de P dans S .

En rapport aux remarques faites dans le paragraphe précédent, on note que le terme $\left. \frac{d}{dt} \overrightarrow{O_S P} \right|_S$ entraîne une confusion. En comparant avec l'équation (A.12), $\overrightarrow{O_S P}$ correspond à $\vec{\Psi}$ et a valeurs dans W . Une notation plus complète serait :

$$\left. \frac{d}{dt} \overrightarrow{O_W P} \right|_W = \left. \frac{d}{dt} \overrightarrow{O_W O_S} \right|_W + \left. \frac{d}{dt} \overrightarrow{O_S P} \right|_{W|_S} + \left. \vec{\Omega}_{S/W} \right|_W \times \left. \overrightarrow{O_S P} \right|_W \quad (A.15)$$

En prenant en compte les remarques dans le paragraphe précédent en faisant apparaître le terme correspondant à $\vec{\phi}$ à valeurs dans S , on a :

$$\left. \frac{d}{dt} \overrightarrow{O_W P} \right|_W = \left. \frac{d}{dt} \overrightarrow{O_W O_S} \right|_W + {}^W R_S \left. \frac{d}{dt} \overrightarrow{O_S P} \right|_S + \left. \vec{\Omega}_{S/W} \right|_W \times \left. \overrightarrow{O_S P} \right|_W \quad (A.16)$$

En asservissement visuel basé position (PBVS), les valeurs associées au vecteur $\left. \frac{d}{dt} \overrightarrow{O_S P} \right|_{S|_S}$ correspondent à la vitesse du point P dans le repère S , alors que les valeurs de $\left. \frac{d}{dt} \overrightarrow{O_S P} \right|_{W|_S}$ ne le sont pas.

A.1.3 Loi de composition des accélérations

Soient W un repère fixe, S un repère mobile par rapport à W et P un point mobile dans W . On exprime l'accélération du point P par rapport à W et tenant compte de la mobilité de S dans W , par :

$$\begin{aligned} \left. \frac{d^2}{dt^2} \overrightarrow{O_W P} \right|_W &= \left. \frac{d^2}{dt^2} \overrightarrow{O_W O_S} \right|_W + \left. \frac{d}{dt} \left(\left. \frac{d}{dt} \overrightarrow{O_S P} \right|_S \right) \right|_W + \left. \vec{\Omega}_{S/W} \right|_W \times \left. \frac{d}{dt} \overrightarrow{O_S P} \right|_S \\ &= \left. \frac{d^2}{dt^2} \overrightarrow{O_W O_S} \right|_W + \left. \frac{d^2}{dt^2} \overrightarrow{O_S P} \right|_S \\ &\quad + \left. \frac{d}{dt} \vec{\Omega}_{S/W} \right|_W \times \left. \overrightarrow{O_S P} \right|_S + 2 \left. \vec{\Omega}_{S/W} \right|_W \times \left. \frac{d}{dt} \overrightarrow{O_S P} \right|_S \\ &\quad + \left. \vec{\Omega}_{S/W} \right|_W \times \left(\left. \vec{\Omega}_{S/W} \right|_W \times \left. \overrightarrow{O_S P} \right|_S \right) \end{aligned} \quad (A.17)$$

On reconnaît dans l'équation (A.17) :

- $\frac{d^2}{dt^2} \overrightarrow{OWP} \Big|_W$ l'accélération absolue de P dans W ;
- $\frac{d^2}{dt^2} \overrightarrow{OWOs} \Big|_W + \frac{d}{dt} \vec{\Omega}_{S/W} \Big|_W \times \overrightarrow{OsP} \Big|_S + \vec{\Omega}_{S/W} \Big|_W \times \left(\vec{\Omega}_{S/W} \Big|_W \times \overrightarrow{OsP} \Big|_S \right)$ l'accélération d'entraînement de S dans W ;
- $\frac{d^2}{dt^2} \overrightarrow{OsP} \Big|_S$ l'accélération relative de P dans S ;
- $2 \vec{\Omega}_{S/W} \Big|_W \times \frac{d}{dt} \overrightarrow{OsP} \Big|_S$ l'accélération de Coriolis, aussi nommée accélération complémentaire.

A.2 Ligne 3D

A.2.1 Représentation

Les éléments ci-après sont tous exprimés dans un seul et unique repère mobile S.

Équation paramétrique d'une ligne 3D Soit un point $\mathbf{x} \in \mathbb{R}^3$ appartenant à la ligne $\mathcal{L}$ de direction $\underline{\mathbf{u}} \in \mathbb{R}^3$. Connaissant un autre point $\mathbf{x}_0$ appartenant lui aussi à cette droite, $\mathbf{x}$ est défini par :

$$\mathbf{x} = \mathbf{x}_0 + k\underline{\mathbf{u}} \quad \forall k \in \mathbb{R} \quad (\text{A.18})$$

avec $\|\underline{\mathbf{u}}\| = 1$, $\underline{\mathbf{u}}$ étant un vecteur unitaire.

Produit scalaire Deux points $\mathbf{x}$ et $\mathbf{x}_0$ appartiennent à la même droite $\mathcal{L}$ de direction $\underline{\mathbf{u}}$ si et seulement si :

$$\begin{aligned} (\mathbf{x} - \mathbf{x}_0) \times \underline{\mathbf{u}} &= 0 \\ (\overrightarrow{OX} - \overrightarrow{OX_0}) \times \underline{\mathbf{u}} &= 0 \\ \overrightarrow{XX_0} \times \underline{\mathbf{u}} &= 0 \end{aligned} \quad (\text{A.19})$$

avec $\|\underline{\mathbf{u}}\| = 1$, $\underline{\mathbf{u}}$ étant un vecteur unitaire.

Coordonnées de Plücker Ci-après une ligne $\mathcal{L}$ de direction $\underline{\mathbf{u}}$, représenté selon les coordonnées de Plücker [Espiau and Rives, 1987] :

$$\mathcal{L} = \begin{bmatrix} \underline{\mathbf{u}} \\ \underline{\mathbf{h}} \end{bmatrix} \quad (\text{A.20})$$

avec,

$$\underline{\mathbf{h}} = \mathbf{x}_0 \times \underline{\mathbf{u}} \quad (\text{A.21})$$

On note que $\underline{\mathbf{u}} \cdot \underline{\mathbf{h}} = 0$, les deux vecteurs sont perpendiculaires entre eux.

Coordonnées de Plücker bi-normalisées Ci-après une ligne $\mathcal{L}$ de direction $\mathbf{u}$, représentée selon les coordonnées de Plücker [Andreff et al., 2002] :

$$\mathcal{L} = \begin{bmatrix} \underline{\mathbf{u}} \\ \underline{\mathbf{h}} \\ h \end{bmatrix} \quad (\text{A.22})$$

avec $h = \|\mathbf{h}\|$, $\underline{\mathbf{u}}$ et $\underline{\mathbf{h}}$ étant des vecteurs unitaires.

A.2.2 Double dérivée des coordonnées de Plücker

On dérive deux fois la représentation en produit scalaire (A.19) en tenant compte que les points X et X_0 sont des points fixes dans W . On a donc :

$$\left(\frac{d}{dt} \overrightarrow{O_S X} \Big|_S - \frac{d}{dt} \overrightarrow{O_S X_0} \Big|_S \right) \times \vec{\mathbf{u}}|_S + \left(\overrightarrow{O_S X} \Big|_S - \overrightarrow{O_S X_0} \Big|_S \right) \times \frac{d}{dt} \vec{\mathbf{u}}|_S = \vec{\mathbf{0}} \quad (\text{A.23})$$

$$\begin{aligned} & \left(\frac{d^2}{dt^2} \overrightarrow{O_S X} \Big|_S - \frac{d^2}{dt^2} \overrightarrow{O_S X_0} \Big|_S \right) \times \vec{\mathbf{u}}|_S \\ & + 2 \left(\frac{d}{dt} \overrightarrow{O_S X} \Big|_S - \frac{d}{dt} \overrightarrow{O_S X_0} \Big|_S \right) \times \frac{d}{dt} \vec{\mathbf{u}}|_S + \left(\overrightarrow{O_S X} - \overrightarrow{O_S X_0} \right) \times \frac{d^2}{dt^2} \vec{\mathbf{u}}|_S = \vec{\mathbf{0}} \end{aligned} \quad (\text{A.24})$$

On se concentre sur le premier terme de l'équation (A.24). Par l'application de la loi de composition des accélérations (cf. (A.17)), on écrit :

$$\begin{aligned} & \left(\frac{d^2}{dt^2} \overrightarrow{O_S X} \Big|_S - \frac{d^2}{dt^2} \overrightarrow{O_S X_0} \Big|_S \right) \times \vec{\mathbf{u}}|_S \\ & = \left(-\frac{d^2}{dt^2} \overrightarrow{O_W O_S} \Big|_S - \frac{d}{dt} \vec{\Omega}_{S/W} \Big|_S \times \frac{d}{dt} \overrightarrow{O_S X} \Big|_S - \vec{\Omega}_{S/W} \Big|_S \times \left(\vec{\Omega}_{S/W} \Big|_S \times \overrightarrow{O_S X} \Big|_S \right) \right. \\ & \quad \left. - 2 \vec{\Omega}_{S/W} \Big|_S \times \frac{d}{dt} \overrightarrow{O_S X} \Big|_S - \frac{d^2}{dt^2} \overrightarrow{O_S X_0} \Big|_S \right) \times \vec{\mathbf{u}}|_S \\ & = \left(-\frac{d^2}{dt^2} \overrightarrow{O_W O_S} \Big|_S - \frac{d}{dt} \vec{\Omega}_{S/W} \Big|_S \times \frac{d}{dt} \left(\overrightarrow{O_S X_0} \Big|_S + k \vec{\mathbf{u}}|_S \right) \right. \\ & \quad \left. - \vec{\Omega}_{S/W} \Big|_S \times \left(\vec{\Omega}_{S/W} \Big|_S \times \left(\overrightarrow{O_S X_0} \Big|_S + k \vec{\mathbf{u}}|_S \right) \right) \right. \\ & \quad \left. - 2 \vec{\Omega}_{S/W} \Big|_S \times \frac{d}{dt} \left(\overrightarrow{O_S X_0} \Big|_S + k \vec{\mathbf{u}}|_S \right) - \frac{d^2}{dt^2} \overrightarrow{O_S X_0} \Big|_S \right) \times \vec{\mathbf{u}}|_S \\ & = \left(-\frac{d^2}{dt^2} \overrightarrow{O_W O_S} \Big|_S - \frac{d}{dt} \vec{\Omega}_{S/W} \Big|_S \times \frac{d}{dt} \overrightarrow{O_S X_0} \Big|_S - \vec{\Omega}_{S/W} \Big|_S \times \left(\vec{\Omega}_{S/W} \Big|_S \times \overrightarrow{O_S X_0} \Big|_S \right) \right. \\ & \quad \left. - 2 \vec{\Omega}_{S/W} \Big|_S \times \frac{d}{dt} \overrightarrow{O_S X_0} \Big|_S - \frac{d^2}{dt^2} \overrightarrow{O_S X_0} \Big|_S \right. \\ & \quad \left. - \frac{d}{dt} \vec{\Omega}_{S/W} \Big|_S \times k \vec{\mathbf{u}}|_S - \vec{\Omega}_{S/W} \Big|_S \times \left(\vec{\Omega}_{S/W} \Big|_S \times k \vec{\mathbf{u}}|_S \right) \right. \\ & \quad \left. - 2 \vec{\Omega}_{S/W} \Big|_S \times k \frac{d}{dt} \vec{\mathbf{u}}|_S \right) \times \vec{\mathbf{u}}|_S \end{aligned} \quad (\text{A.25})$$

On exprime le deuxième terme de (A.24) tel que :

$$\begin{aligned}
& 2 \left(\frac{d}{dt} \overrightarrow{O_S X} \Big|_S - \frac{d}{dt} \overrightarrow{O_S X_0} \Big|_S \right) \times \frac{d}{dt} \vec{u}|_S \\
&= 2 \left(-\frac{d}{dt} \overrightarrow{O_W O_S} \Big|_S - \vec{\Omega}_{S/W} \Big|_S \times \overrightarrow{O_S X} \Big|_S - \frac{d}{dt} \overrightarrow{O_S X_0} \Big|_S \right) \times \frac{d}{dt} \vec{u}|_S \\
&= 2 \left(-\frac{d}{dt} \overrightarrow{O_W O_S} \Big|_S - \vec{\Omega}_{S/W} \Big|_S \times \left(\overrightarrow{O_S X_0} \Big|_S + k \vec{u}|_S \right) \right. \\
&\quad \left. - \frac{d}{dt} \overrightarrow{O_S X_0} \Big|_S \right) \times \frac{d}{dt} \vec{u}|_S \tag{A.26} \\
&= 2 \left(\underbrace{-\frac{d}{dt} \overrightarrow{O_W O_S} \Big|_S - \vec{\Omega}_{S/W} \Big|_S \times \overrightarrow{O_S X_0} \Big|_S}_{=\frac{d}{dt} \overrightarrow{O_S X_0} \Big|_S} - \frac{d}{dt} \overrightarrow{O_S X_0} \Big|_S - \vec{\Omega}_{S/W} \Big|_S \times k \vec{u}|_S \right) \times \frac{d}{dt} \vec{u}|_S
\end{aligned}$$

Le troisième terme de (A.24) devient :

$$\begin{aligned}
& \left(\overrightarrow{O_S X} - \overrightarrow{O_S X_0} \right) \times \frac{d^2}{dt^2} \vec{u}|_S \\
&= \left(\overrightarrow{O_S X_0} \Big|_S + k \vec{u}|_S - \overrightarrow{O_S X_0} \right) \times \frac{d^2}{dt^2} \vec{u}|_S \tag{A.27} \\
&= k \vec{u}|_S \times \frac{d^2}{dt^2} \vec{u}|_S
\end{aligned}$$

En réécrivant (A.24) en considérant (A.25), (A.26) et (A.27) et en remplaçant les termes désignés, l'équation devient :

$$\begin{aligned}
& k \vec{u}|_S \times \frac{d^2}{dt^2} \vec{u}|_S - \left(\vec{\Omega}_{S/W} \Big|_S \times k \vec{u}|_S \right) \times \frac{d}{dt} \vec{u}|_S \\
&+ \left(-\frac{d}{dt} \vec{\Omega}_{S/W} \Big|_S \times k \vec{u}|_S - \vec{\Omega}_{S/W} \Big|_S \times \left(\vec{\Omega}_{S/W} \Big|_S \times k \vec{u}|_S \right) \right. \\
&\quad \left. - 2 \vec{\Omega}_{S/W} \Big|_S \times k \frac{d}{dt} \vec{u}|_S \right) \times \vec{u}|_S = \vec{0} \\
&\vec{u}|_S \times \frac{d^2}{dt^2} \vec{u}|_S - \left(\underbrace{\vec{\Omega}_{S/W} \Big|_S \times \vec{u}|_S}_{=-\frac{d}{dt} \vec{u}|_S} \right) \times \frac{d}{dt} \vec{u}|_S \\
&+ \left(-\frac{d}{dt} \vec{\Omega}_{S/W} \Big|_S \times \vec{u}|_S - \vec{\Omega}_{S/W} \Big|_S \times \left(\vec{\Omega}_{S/W} \Big|_S \times \vec{u}|_S \right) \right. \\
&\quad \left. - 2 \vec{\Omega}_{S/W} \Big|_S \times \frac{d}{dt} \vec{u}|_S \right) \times \vec{u}|_S = \vec{0} \tag{A.28}
\end{aligned}$$

Par applications multiples de l'identité de Jacobi, on obtient :

$$\begin{aligned}
& \left(-\frac{d^2}{dt^2} \vec{u}|_S - \frac{d}{dt} \vec{\Omega}_{S/W} \Big|_S \times \vec{u}|_S - \vec{\Omega}_{S/W} \Big|_S \times \left(\vec{\Omega}_{S/W} \Big|_S \times \vec{u}|_S \right) \right. \\
&\quad \left. - 2 \vec{\Omega}_{S/W} \Big|_S \times \frac{d}{dt} \vec{u}|_S \right) \times \vec{u}|_S = \vec{0} \tag{A.29}
\end{aligned}$$

On pose que $\vec{u}|_S \neq \vec{0}$, donc,

$$\begin{aligned}
 & -\frac{d^2}{dt^2} \vec{u}|_S - \frac{d}{dt} \vec{\Omega}_{S/W}|_S \times \vec{u}|_S - \vec{\Omega}_{S/W}|_S \times \left(\vec{\Omega}_{S/W}|_S \times \vec{u}|_S \right) \\
 & - 2 \vec{\Omega}_{S/W}|_S \times \frac{d}{dt} \vec{u}|_S = \vec{0} \\
 \\
 & \frac{d^2}{dt^2} \vec{u}|_S = -\frac{d}{dt} \vec{\Omega}_{S/W}|_S \times \vec{u}|_S - \vec{\Omega}_{S/W}|_S \times \left(\vec{\Omega}_{S/W}|_S \times \vec{u}|_S \right) \\
 & - 2 \vec{\Omega}_{S/W}|_S \times \frac{d}{dt} \vec{u}|_S \\
 \\
 & \frac{d^2}{dt^2} \vec{u}|_S = \vec{u}|_S \times \frac{d}{dt} \vec{\Omega}_{S/W}|_S + \left(\vec{\Omega}_{S/W}|_S \times \vec{u}|_S \right) \times \vec{\Omega}_{S/W}|_S \\
 & + 2 \frac{d}{dt} \vec{u}|_S \times \vec{\Omega}_{S/W}|_S
 \end{aligned} \tag{A.30}$$

Sous forme plus compacte en considérant les valeurs des vecteurs, on a

$$\ddot{\mathbf{u}} = \mathbf{u} \times \dot{\omega} + (2 \dot{\mathbf{u}} + \omega \times \mathbf{u}) \times \omega \tag{A.31}$$

et sous sa forme matricielle,

$$\ddot{\mathbf{u}} = \begin{bmatrix} \mathbf{0} & [\mathbf{u}]_{\times} \end{bmatrix} \dot{\mathbf{v}} + \begin{bmatrix} \mathbf{0} & [2 \dot{\mathbf{u}} + \omega \times \mathbf{u}]_{\times} \end{bmatrix} \mathbf{v} \tag{A.32}$$

L'opération est effectuée de la même manière pour la composante $\mathbf{h}$. En dérivant deux fois l'équation (A.21), on obtient :

$$\frac{d^2}{dt^2} \vec{h}|_S = \frac{d^2}{dt^2} \overrightarrow{OSX_0}|_S \times \vec{u}|_S + 2 \frac{d}{dt} \overrightarrow{OSX_0}|_S \times \frac{d}{dt} \vec{u}|_S + \overrightarrow{OSX_0}|_S \times \frac{d^2}{dt^2} \vec{u}|_S \tag{A.33}$$

En remplaçant $\frac{d^2}{dt^2} \overrightarrow{OSX_0}|_S$, $\frac{d}{dt} \overrightarrow{OSX_0}|_S$ et $\frac{d^2}{dt^2} \vec{u}|_S$ par leurs expressions à partir de leurs équations respectives (A.17), (A.14) et (A.30), et en appliquant plusieurs fois l'identité de Jacobi, l'équation précédente devient :

$$\begin{aligned}
 \frac{d^2}{dt^2} \vec{h}|_S &= \vec{u}|_S \times \frac{d^2}{dt^2} \overrightarrow{OWOS}|_S + \vec{u}|_S \times \frac{d}{dt} \vec{\Omega}_{S/W}|_S + 2 \frac{d}{dt} \vec{u}|_S \times \frac{d}{dt} \overrightarrow{OWOS}|_S \\
 &+ 2 \vec{h}|_S \times \vec{\Omega}_{S/W}|_S + \left(\vec{\Omega}_{S/W}|_S \times \vec{h}|_S \right) \times \vec{\Omega}_{S/W}|_S
 \end{aligned} \tag{A.34}$$

De la même manière que pour $\mathbf{u}$, en considérant les valeurs des vecteurs, il est possible d'écrire l'équation (A.34) sous une forme plus compacte telle que :

$$\ddot{\mathbf{h}} = \mathbf{u} \times \dot{v} + \mathbf{h} \times \dot{\omega} + 2 \dot{\mathbf{u}} \times v + (2 \dot{\mathbf{h}} + \omega \times \mathbf{h}) \times \omega \tag{A.35}$$

ou, sous sa forme matricielle,

$$\ddot{\mathbf{h}} = \begin{bmatrix} [\mathbf{u}]_{\times} & [\mathbf{h}]_{\times} \end{bmatrix} \dot{\mathbf{v}} + \begin{bmatrix} [2 \dot{\mathbf{u}}]_{\times} & [2 \dot{\mathbf{h}} + \omega \times \mathbf{h}]_{\times} \end{bmatrix} \mathbf{v} \tag{A.36}$$

Sur la base des équations (A.32) et (A.36), le modèle du second ordre pour les lignes 3D représentées par leur coordonnées de Plücker est donc de la forme :

$$\begin{bmatrix} \ddot{\mathbf{h}} \\ \ddot{\mathbf{u}} \end{bmatrix} = \mathbf{L}_{\mathcal{L}_p} \dot{\mathbf{v}} + \mathbf{H}_{\mathcal{L}_p} \mathbf{v} \quad (\text{A.37})$$

où on retrouve la matrice d'interaction du premier ordre $\mathbf{L}_{\mathcal{L}_p}$ telle que définie dans [Espiau and Rives \[1987\]](#),

$$\mathbf{L}_{\mathcal{L}_p} = \begin{bmatrix} [\mathbf{u}]_{\times} & [\mathbf{h}]_{\times} \\ \mathbf{0} & [\mathbf{u}]_{\times} \end{bmatrix} \quad (\text{A.38})$$

et le hessien $\mathbf{H}_{\mathcal{L}_p}$ tel que

$$\mathbf{H}_{\mathcal{L}_p} = \begin{bmatrix} [2\dot{\mathbf{u}}]_{\times} & [2\dot{\mathbf{h}} + \omega \times \mathbf{h}]_{\times} \\ \mathbf{0} & [2\dot{\mathbf{u}} + \omega \times \mathbf{u}]_{\times} \end{bmatrix} \quad (\text{A.39})$$

On note qu'un travail similaire peut être effectué en tenant compte des coordonnées de Plücker bi-normalisées présentées dans le paragraphe précédent.

A.3 Plan 3D

A.3.1 Représentation

Les éléments présentés ci-après sont tous exprimés dans le repère S.

Équation cartésienne Soit un vecteur $\mathbf{p} \in \mathbb{R}^4$ tel que $\mathbf{p} = [a \ b \ c \ d]^T$ qui décrit le plan P et un point $\mathbf{x} = [x \ y \ z]^T$ de l'espace. $\mathbf{x}$ appartient au plan P si et seulement si

$$\mathbf{p} \cdot \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = 0 \quad (\text{A.40})$$

$$\mathbf{p}^T \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = 0 \quad (\text{A.41})$$

$$\begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = 0 \quad (\text{A.42})$$

$$ax + by + cz + d = 0 \quad (\text{A.43})$$

Point et vecteur Le plan P peut être défini par un point $\mathbf{x}_0$ appartenant à P et un vecteur $\mathbf{n}$, $\mathbf{n}$ étant la normale au plan. Le point X décrit par $\mathbf{x}$ appartient au plan P si et seulement si

$$\overrightarrow{X_0X} \cdot \vec{\mathbf{n}} = 0 \quad (\text{A.44})$$

$$\left(\overrightarrow{OX} - \overrightarrow{OX_0} \right) \cdot \vec{\mathbf{n}} = 0 \quad (\text{A.45})$$

où O représente l'origine du repère courant.

$$\overrightarrow{OX} \cdot \vec{\mathbf{n}} - \overrightarrow{OX_0} \cdot \vec{\mathbf{n}} = 0 \quad (\text{A.46})$$

$$\mathbf{n}^\top \mathbf{x} - \mathbf{n}^\top \mathbf{x}_0 = 0 \quad (\text{A.47})$$

En considérant (A.42), on observe que $\mathbf{n} = [a \ b \ c]^\top$ et $d = \mathbf{n}^\top \mathbf{x}_0$.

Trois vecteurs Soit trois vecteurs $\mathbf{u}$, $\mathbf{v}$ et $\mathbf{w}$, ces trois vecteurs sont coplanaires (*i.e.* forment un plan P) si et seulement si leur produit mixte est nul :

$$\mathbf{w} \cdot (\mathbf{u} \times \mathbf{v}) = 0 \quad (\text{A.48})$$

On observe alors que $\mathbf{n} = \mathbf{u} \times \mathbf{v}$ et que $\mathbf{w} := \overrightarrow{X_0X}$.

Équations paramétriques Soit $\mathbf{u}$, $\mathbf{v}$ deux vecteurs coplanaires formant un plan P et X_0 un point de ce même plan, X appartient au plan si et seulement si

$$\mathbf{x} = \mathbf{x}_0 + k\mathbf{u} + l\mathbf{v} \quad \forall k \in \mathbb{R}, l \in \mathbb{R} \quad (\text{A.49})$$

A.3.2 Double dérivée de la représentation point vecteur

La représentation d'un plan 3D se compose d'un point et d'un vecteur. La double dérivée de l'expression du vecteur $\frac{d^2}{dt^2} \vec{\mathbf{n}}$ est similaire à celle effectuée pour la double dérivée des coordonnées de Plücker en section A.2.2. En effet, $\vec{\mathbf{u}}$ peut être considéré comme un vecteur unitaire quelconque. Les expressions des dérivées première et seconde restent valables pour tout vecteur unitaire quelconque, soit par application de l'équation (A.30) pour $\vec{\mathbf{n}}$,

$$\begin{aligned} \frac{d^2}{dt^2} \vec{\mathbf{n}}|_S &= \vec{\mathbf{n}}|_S \times \frac{d}{dt} \vec{\Omega}_{S/W}|_S + \left(\vec{\Omega}_{S/W}|_S \times \vec{\mathbf{n}}|_S \right) \times \vec{\Omega}_{S/W}|_S \\ &\quad + 2 \frac{d}{dt} \vec{\mathbf{n}}|_S \times \vec{\Omega}_{S/W}|_S \end{aligned} \quad (\text{A.50})$$

Il reste à déterminer la double dérivée de la coordonnée d . En dérivant deux fois l'expression

$d|_S = -\vec{n}|_S \cdot \overrightarrow{O_S X_0}|_S$, on obtient :

$$\begin{aligned}
\frac{d^2}{dt^2} d|_S &= -\frac{d^2}{dt^2} \overrightarrow{O_S X_0}|_{S(A.17)} \cdot \vec{n}|_S - 2 \frac{d}{dt} \overrightarrow{O_S X_0}|_S \cdot \frac{d}{dt} \vec{n}|_S - \overrightarrow{O_S X_0}|_S \cdot \frac{d^2}{dt^2} \vec{n}|_{S(A.50)} \\
&= -\left(-\frac{d^2}{dt^2} \overrightarrow{O_W O_S}|_S - \frac{d}{dt} \vec{\Omega}_{S/W}|_S \times \overrightarrow{O_S X_0}|_S - \vec{\Omega}_{S/W}|_S \times \left(\vec{\Omega}_{S/W}|_S \times \overrightarrow{O_S X_0}|_S \right) \right. \\
&\quad \left. - 2 \vec{\Omega}_{S/W}|_S \times \frac{d}{dt} \overrightarrow{O_S X_0}|_S \right) \cdot \vec{n}|_S - 2 \frac{d}{dt} \overrightarrow{O_S X_0}|_S \cdot \frac{d}{dt} \vec{n}|_S \\
&\quad - \overrightarrow{O_S X_0}|_S \cdot \left(\vec{n}|_S \times \frac{d}{dt} \vec{\Omega}_{S/W}|_S + \left(\vec{\Omega}_{S/W}|_S \times \vec{n}|_S \right) \times \vec{\Omega}_{S/W}|_S \right. \\
&\quad \left. + 2 \frac{d}{dt} \vec{n}|_S \times \vec{\Omega}_{S/W}|_S \right) \\
&= \frac{d^2}{dt^2} \overrightarrow{O_W O_S}|_S \cdot \vec{n}|_S \\
&\quad + \frac{\left(\frac{d}{dt} \vec{\Omega}_{S/W}|_S \times \overrightarrow{O_S X_0}|_S \right) \cdot \vec{n}|_S - \overrightarrow{O_S X_0}|_S \cdot \left(\vec{n}|_S \times \frac{d}{dt} \vec{\Omega}_{S/W}|_S \right)}{= \vec{0}} \\
&\quad - 2 \frac{\frac{d}{dt} \overrightarrow{O_S X_0}|_S \cdot \frac{d}{dt} \vec{n}|_S - \overrightarrow{O_S X_0}|_S \cdot \left(2 \frac{d}{dt} \vec{n}|_S \times \vec{\Omega}_{S/W}|_S \right)}{= 2 \frac{d}{dt} \vec{\Omega}_{S/W}|_S \cdot \frac{d}{dt} \vec{n}|_S} \\
&\quad + 2 \left(\vec{\Omega}_{S/W}|_S \times \frac{d}{dt} \overrightarrow{O_S X_0}|_S \right) \cdot \vec{n}|_S \\
&\quad + \frac{\left(\vec{\Omega}_{S/W}|_S \times \left(\vec{\Omega}_{S/W}|_S \times \overrightarrow{O_S X_0}|_S \right) \right) \cdot \vec{n}|_S}{- \overrightarrow{O_S X_0}|_S \cdot \left(\left(\vec{\Omega}_{S/W}|_S \times \vec{n}|_S \right) \times \vec{\Omega}_{S/W}|_S \right)} = 2 n|_S \cdot \left(\vec{\Omega}_{S/W}|_S \times \left(\vec{\Omega}_{S/W}|_S \times \overrightarrow{O_S X_0}|_S \right) \right) \\
&= \frac{d^2}{dt^2} \overrightarrow{O_W O_S}|_S \cdot \vec{n}|_S \\
&\quad + 2 \frac{\frac{d}{dt} \vec{\Omega}_{S/W}|_S \cdot \frac{d}{dt} \vec{n}|_S}{+ 2 \left(\vec{\Omega}_{S/W}|_S \times \frac{d}{dt} \overrightarrow{O_S X_0}|_S \right) \cdot \vec{n}|_S} \\
&\quad + 2 \frac{\left(\vec{\Omega}_{S/W}|_S \times \frac{d}{dt} \overrightarrow{O_S X_0}|_S \right) \cdot \vec{n}|_S}{+ 2 \vec{n}|_S \cdot \left(\vec{\Omega}_{S/W}|_S \times \left(\vec{\Omega}_{S/W}|_S \times \overrightarrow{O_S X_0}|_S \right) \right)} = -2 \vec{\Omega}_{S/W}|_S \cdot \left(\vec{u}|_S \times \frac{d}{dt} \overrightarrow{O_W O_S}|_S \right) \\
&= \frac{d^2}{dt^2} \overrightarrow{O_W O_S}|_S \cdot \vec{n}|_S + 2 \frac{d}{dt} \vec{\Omega}_{S/W}|_S \cdot \frac{d}{dt} \vec{n}|_S - 2 \vec{\Omega}_{S/W}|_S \cdot \left(\vec{u}|_S \times \frac{d}{dt} \overrightarrow{O_W O_S}|_S \right) \quad (A.51)
\end{aligned}$$

En utilisant une forme compacte, l'équation (A.51) devient :

$$\ddot{d} = \mathbf{n} \cdot \dot{v} + 2 \dot{\mathbf{n}} \cdot v + (v \times \mathbf{n}) \cdot \omega \quad (A.52)$$

ou, sous une forme matricielle :

$$\ddot{\mathbf{d}} = \begin{bmatrix} \mathbf{n}^\top & \mathbf{0} \end{bmatrix} \dot{\mathbf{v}} + \begin{bmatrix} 2 \dot{\mathbf{n}} & (v \times \mathbf{n})^\top \end{bmatrix} \mathbf{v} \quad (\text{A.53})$$

Sur la base des équations (A.50) et (A.53), le modèle du second ordre pour le plan 3D est de la forme :

$$\begin{bmatrix} \ddot{\mathbf{n}} \\ \ddot{\mathbf{d}} \end{bmatrix} = \mathbf{L}_{\mathcal{P}} \dot{\mathbf{v}} + \mathbf{H}_{\mathcal{P}} \mathbf{v} \quad (\text{A.54})$$

où on retrouve la matrice d'interaction du premier ordre $\mathbf{L}_{\mathcal{P}}$ telle que définie dans [Espiau and Rives \[1987\]](#),

$$\mathbf{L}_{\mathcal{P}} = \begin{bmatrix} \mathbf{0} & [\mathbf{n}]_{\times} \\ \mathbf{n}^\top & \mathbf{0} \end{bmatrix} \quad (\text{A.55})$$

et le hessien $\mathbf{H}_{\mathcal{P}}$ tel que

$$\mathbf{H}_{\mathcal{P}} = \begin{bmatrix} \mathbf{0} & [2 \dot{\mathbf{n}} + \omega \times \mathbf{n}]_{\times} \\ 2 \dot{\mathbf{n}}^\top & (v \times \mathbf{n})^\top \end{bmatrix} \quad (\text{A.56})$$

A.4 Point 2D

A.4.1 Représentation

Soit $\mathbf{x}$ un point 3D observé depuis un repère mobile $\mathcal{S}$, la projection de $\mathbf{x}$ sur le plan focal d'une image, notée $\mathbf{x}_n$ est réalisée par l'équation 2.49 rappelée ci-dessous :

$$\mathbf{x}_n = \text{proj}_{3\text{D} \rightarrow 2\text{D}}(\mathbf{x}) = \frac{1}{z} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix}$$

A.4.2 Modèle du second ordre pour le point 2D

La matrice d'interaction du point 2D normalisé peut être retrouvée en se basant sur la matrice d'interaction du point 3D. On note que :

$$\begin{aligned} \dot{\mathbf{x}}_n &= \frac{\partial \mathbf{x}_n}{\partial \mathbf{x}} \dot{\mathbf{x}} \\ &= \frac{\partial \mathbf{x}_n}{\partial \mathbf{x}} \mathbf{L}_{\mathbf{x}} \mathbf{v} \end{aligned} \quad (\text{A.57})$$

ainsi que,

$$\mathbf{L}_{\mathbf{x}_n} = \frac{\partial \mathbf{x}_n}{\partial \mathbf{x}} \mathbf{L}_{\mathbf{x}} \quad (\text{A.58})$$

où $\mathbf{L}_{\mathbf{x}}$ est donnée en équation (2.40) et retrouvée en équation (3.28a). La dérivée partielle est de la forme :

$$\frac{\partial \mathbf{x}_n}{\partial \mathbf{x}} = \begin{bmatrix} \frac{1}{z} & 0 & -\frac{x}{z^2} \\ 0 & \frac{1}{z} & -\frac{y}{z^2} \end{bmatrix} = \begin{bmatrix} \frac{1}{z} & 0 & -\frac{x_n}{z} \\ 0 & \frac{1}{z} & -\frac{y_n}{z} \end{bmatrix} \quad (\text{A.59})$$

Le modèle du second ordre est obtenu en dérivant l'équation (A.57) :

$$\begin{aligned}
 \ddot{\mathbf{x}}_n &= \frac{d}{dt} \frac{\partial \mathbf{x}_n}{\partial \mathbf{x}} \mathbf{L}_x \mathbf{v} + \frac{\partial \mathbf{x}_n}{\partial \mathbf{x}} \frac{d}{dt} (\mathbf{L}_x \mathbf{v}) \\
 &= \frac{d}{dt} \frac{\partial \mathbf{x}_n}{\partial \mathbf{x}} \mathbf{L}_x \mathbf{v} + \frac{\partial \mathbf{x}_n}{\partial \mathbf{x}} (\mathbf{L}_x \dot{\mathbf{v}} + \mathbf{H}_x \mathbf{v}) \\
 &= \frac{\partial \mathbf{x}_n}{\partial \mathbf{x}} \mathbf{L}_x \dot{\mathbf{v}} + \left(\frac{d}{dt} \frac{\partial \mathbf{x}_n}{\partial \mathbf{x}} \mathbf{L}_x + \frac{\partial \mathbf{x}_n}{\partial \mathbf{x}} \mathbf{H}_x \right) \mathbf{v}
 \end{aligned} \tag{A.60}$$

On reconnaît dans l'équation (A.60) la matrice d'interaction du point 2D initialement présentée en équation (2.37) :

$$\mathbf{L}_{\mathbf{x}_n} = \frac{\partial \mathbf{x}_n}{\partial \mathbf{x}} \mathbf{L}_x = \begin{bmatrix} -\frac{1}{z} & 0 & \frac{x_n}{z} & x_n y_n & -1 - x_n^2 & y_n \\ 0 & -\frac{1}{z} & \frac{y_n}{z} & 1 + y_n^2 & -x_n y_n & -x_n \end{bmatrix}$$

On pose le hessien du point 2D tel que :

$$\mathbf{H}_{\mathbf{x}_n} = \frac{d}{dt} \frac{\partial \mathbf{x}_n}{\partial \mathbf{x}} \mathbf{L}_x + \frac{\partial \mathbf{x}_n}{\partial \mathbf{x}} \mathbf{H}_x \tag{A.61}$$

avec la dérivée temporelle de la dérivée partielle $\frac{d}{dt} \frac{\partial \mathbf{x}_n}{\partial \mathbf{x}}$ de la forme :

$$\frac{d}{dt} \frac{\partial \mathbf{x}_n}{\partial \mathbf{x}} = \begin{bmatrix} -\frac{\dot{z}}{z^2} & 0 & \frac{\dot{z}}{z^2} x_n - \frac{\dot{x}_n}{z} \\ 0 & -\frac{\dot{z}}{z^2} & \frac{\dot{z}}{z^2} y_n - \frac{\dot{y}_n}{z} \end{bmatrix} \tag{A.62}$$

Les quantités $\mathbf{L}_x$ et $\mathbf{H}_x$ sont données respectivement en équations (3.28a) et (3.28b). On a donc

$$\frac{d}{dt} \frac{\partial \mathbf{x}_n}{\partial \mathbf{x}} \mathbf{L}_x = \begin{bmatrix} \frac{\dot{z}}{z^2} & 0 & -\frac{\dot{z}}{z^2} x_n + \frac{\dot{x}_n}{z} & y_n \dot{x}_n - y_n x_n \frac{\dot{z}}{z} & -x_n \dot{x}_n + (1 + x_n^2) \frac{\dot{z}}{z} & -y_n \frac{\dot{z}}{z} \\ 0 & \frac{\dot{z}}{z^2} & -\frac{\dot{z}}{z^2} y_n + \frac{\dot{y}_n}{z} & y_n \dot{y}_n - (1 + y_n^2) \frac{\dot{z}}{z} & -x_n \dot{y}_n + y_n x_n \frac{\dot{z}}{z} & x_n \frac{\dot{z}}{z} \end{bmatrix} \tag{A.63}$$

$$\frac{\partial \mathbf{x}_n}{\partial \mathbf{x}} \mathbf{H}_x = \begin{bmatrix} 0 & 0 & 0 & 2x_n \frac{\dot{y}}{z} + \omega_3 x_n^2 - \omega_1 x_n & -2\frac{\dot{z}}{z} - \omega_1 y_n - 2x_n \frac{\dot{x}}{z} + \omega_3 x_n y_n & 2\frac{\dot{y}}{z} + \omega_3 x_n - \omega_1 \\ 0 & 0 & 0 & 2\frac{\dot{z}}{z} - \omega_2 x_n + 2y_n \frac{\dot{y}}{z} + \omega_3 x_n y_n & -2y_n \frac{\dot{x}}{z} + \omega_3 y_n^2 - \omega_2 y_n & -\frac{2\dot{x}}{z} + \omega_3 y_n + \omega_2 \end{bmatrix} \tag{A.64}$$

En faisant la somme des équations (A.63) et (A.64), on obtient :

$$\begin{aligned}
 \mathbf{H}_{\mathbf{x}_n} &= \begin{bmatrix} \frac{\dot{z}}{z^2} & 0 & -\frac{\dot{z}}{z^2} x_n + \frac{\dot{x}_n}{z} \\ 0 & \frac{\dot{z}}{z^2} & -\frac{\dot{z}}{z^2} y_n + \frac{\dot{y}_n}{z} \end{bmatrix} \\
 &\quad \begin{bmatrix} \left(2\frac{\dot{y}}{z} - y_n \frac{\dot{z}}{z} + \omega_3 x_n - \omega_1 \right) x_n + y_n \dot{x}_n & -(1 - x_n^2) \frac{\dot{z}}{z} + (-2\frac{\dot{x}}{z} - \dot{x}_n) x_n + (\omega_1 + \omega_3 x_n) y_n \\ (1 + y_n^2) \frac{\dot{z}}{z} + \left(\dot{y}_n + 2\frac{\dot{y}}{z} \right) y_n + (\omega_3 y_n - \omega_2) x_n & \left(x_n \frac{\dot{z}}{z} - 2\frac{\dot{x}}{z} + \omega_3 y_n - \omega_2 \right) y_n - x_n \dot{y}_n \\ -y_n \frac{\dot{z}}{z} + 2\frac{\dot{y}}{z} + \omega_3 x_n - \omega_1 \\ x_n \frac{\dot{z}}{z} - \frac{2\dot{x}}{z} + \omega_3 y_n + \omega_2 \end{bmatrix}
 \end{aligned} \tag{A.65}$$

A.5 Ligne 2D

A.5.1 Représentation

Cartésienne Soit une ligne 2D paramétrée selon les composantes d'une équation affine sur le plan focal, le vecteur qui décrit la ligne est composé des éléments suivants :

$$\mathcal{L}_{2D} = \begin{bmatrix} \alpha \\ \beta \\ \zeta \end{bmatrix} \quad (\text{A.66})$$

où $\alpha x_n + \beta y_n + \zeta = 0$.

Une droite dans l'espace peut être définie par l'intersection de deux plans non parallèles. Soit un point 3D $\mathbf{x} = [x \ y \ z]^\top$, $\mathbf{x}$ appartient à la droite si et seulement si :

$$\begin{cases} a_1x + b_1y + c_1z + d_1 = 0 \\ a_2x + b_2y + c_2z + d_2 = 0 \end{cases} \quad (\text{A.67})$$

où $[a_i \ b_i \ c_i \ d_i]^\top$ sont les composantes qui définissent le plan i . En projetant sur le plan focal d'une caméra, l'équation (A.67) devient :

$$\begin{cases} a_1x_n + b_1y_n + c_1 + \frac{d_1}{z} = 0 \\ a_2x_n + b_2y_n + c_2 + \frac{d_2}{z} = 0 \end{cases} \quad (\text{A.68})$$

$$\begin{cases} -\frac{1}{z} = \frac{1}{d_1}(a_1x_n + b_1y_n + c_1) \\ -\frac{1}{z} = \frac{1}{d_2}(a_2x_n + b_2y_n + c_2) \end{cases} \quad (\text{A.69})$$

Par égalité, on a alors,

$$\frac{1}{d_1}(a_1x_n + b_1y_n + c_1) - \frac{1}{d_2}(a_2x_n + b_2y_n + c_2) = 0 \quad (\text{A.70})$$

$$(a_1d_2 - a_2d_1)x_n + (b_1d_2 - b_2d_1)y_n + (c_1d_2 - c_2d_1) = 0 \quad (\text{A.71})$$

On obtient alors une équation de ligne 2D sur le plan focal en utilisant les coordonnées des deux plans 3D, telle que :

$$\alpha x_n + \beta y_n + \zeta = 0 \quad (\text{A.72})$$

où $\alpha = a_1d_2 - a_2d_1$, $\beta = b_1d_2 - b_2d_1$ et $\zeta = c_1d_2 - c_2d_1$.

On note que cette représentation n'est pas minimale car elle nécessite 8 paramètres.

Polaire Soit une ligne 2D paramétrée selon les coordonnées polaires sur le plan focal, le vecteur des coordonnées polaires qui décrit la caractéristique est composé de la façon suivante :

$$\mathcal{L}_{2D} = \begin{bmatrix} \rho \\ \theta \end{bmatrix} \quad (\text{A.73})$$

où ρ représente la distance orthogonale entre le centre du plan focal (point principal) et sa projection sur la droite et θ l'angle entre le vecteur unitaire selon x_n et la direction de la droite $\mathcal{L}_{2D}$ sur le plan focal. Le passage entre la représentation cartésienne et polaire s'effectue grâce à l'équation suivante :

$$\rho = x_n \cos(\theta) + y_n \sin(\theta) \quad (\text{A.74})$$

A.5.2 Matrice d'interaction pour le modèle du premier ordre

Cette section revisite les travaux de [Chaumette \[1990\]](#); [Espiau et al. \[1992\]](#); [Debain \[1996\]](#) où la matrice d'interaction du premier ordre pour une ligne 2D est obtenue. Les vecteurs sont à valeurs dans le repère mobile S.

On dérive l'équation (A.74) :

$$\dot{\rho} = \dot{x}_n \cos(\theta) + \dot{y}_n \sin(\theta) + \dot{\theta}(-x_n \sin(\theta) + y_n \cos(\theta)) \quad (\text{A.75})$$

En exprimant x en fonction de y par l'équation (A.74), l'équation devient :

$$\begin{aligned} \dot{\rho} + \dot{\theta} \left(\frac{\rho - y_n \sin(\theta)}{\cos(\theta)} \sin(\theta) - y_n \cos(\theta) \right) &= \dot{x}_n \cos(\theta) + \dot{y}_n \sin(\theta) \\ \dot{\rho} + \rho \tan(\theta) \dot{\theta} + \dot{\theta} \left(-\frac{y_n \sin(\theta)^2}{\cos(\theta)} - \frac{y_n \cos(\theta)^2}{\cos(\theta)} \right) &= \dot{x}_n \cos(\theta) + \dot{y}_n \sin(\theta) \\ \dot{\rho} + \rho \tan(\theta) \dot{\theta} - \frac{\dot{\theta}}{\cos(\theta)} y_n &= \dot{x}_n \cos(\theta) + \dot{y}_n \sin(\theta) \end{aligned} \quad (\text{A.76})$$

De même, on définit λ_1 et λ_2 en réécrivant x_n dans l'équation (A.77) :

$$\begin{aligned} -\frac{1}{z} &= \frac{1}{d_i} (a_i x_n + b_i y_n + c_i) \\ &= \frac{1}{d_i} \left(a_i \frac{\rho - y_n \sin(\theta)}{\cos(\theta)} + b_i y_n + c_i \right) \\ &= \frac{1}{d_i} (-a_i \tan(\theta) + b_i) y_n + \frac{1}{d_i} \left(a_i \frac{\rho}{\cos(\theta)} + c_i \right) \\ &= \lambda_1 y_n + \lambda_2 \end{aligned} \quad (\text{A.77})$$

D'après les équations (2.1) et (2.37), on explicite $x_n = \mathbf{L}_1 \mathbf{v}$ et $y_n = \mathbf{L}_2 \mathbf{v}$ où $\mathbf{L}_1$ et $\mathbf{L}_2$ sont respectivement la première et la seconde ligne de la matrice $\mathbf{L}_{\mathbf{x}_n}$ en équation (2.37). On développe le terme pour identifier deux nouvelles matrices $\mathbf{K}_1$ et $\mathbf{K}_2$.

$$\begin{aligned} \dot{x}_n \cos(\theta) + \dot{y}_n \sin(\theta) &= \mathbf{L}_1 \mathbf{v} \cos(\theta) + \mathbf{L}_2 \mathbf{v} \sin(\theta) \\ &= y_n \mathbf{K}_1 \mathbf{v} + \mathbf{K}_2 \mathbf{v} \end{aligned} \quad (\text{A.78})$$

Les matrices $\mathbf{K}_1$ et $\mathbf{K}_2$ sont telles que :

$$\mathbf{K}_1 = \begin{bmatrix} \lambda_1 \cos(\theta) & \lambda_1 \sin(\theta) & -\lambda_1 \rho & \rho & \rho \tan(\theta) & \frac{1}{\cos(\theta)} \end{bmatrix} \quad (\text{A.79})$$

$$\mathbf{K}_2 = \begin{bmatrix} \lambda_2 \cos(\theta) & \lambda_2 \sin(\theta) & -\lambda_2 \rho & \sin(\theta) & -\left(\cos(\theta) + \frac{\rho^2}{\cos(\theta)} \right) & -\rho \tan(\theta) \end{bmatrix} \quad (\text{A.80})$$

En mixant les équations (A.76) et (A.78), on obtient :

$$-\frac{\dot{\theta}}{\cos(\theta)} y_n + \dot{\rho} + \rho \tan(\theta) \dot{\theta} = \mathbf{K}_1 \mathbf{v} y_n + \mathbf{K}_2 \mathbf{v} \quad (\text{A.81})$$

Par identification,

$$\begin{cases} \dot{\theta} = -\mathbf{K}_1 \cos(\theta) \mathbf{v} \\ \dot{\rho} = (\mathbf{K}_2 + \mathbf{K}_1 \rho \sin(\theta)) \mathbf{v} \end{cases} \quad (\text{A.82})$$

À partir de l'équation précédente et de la définition des matrices $\mathbf{K}_1$ et $\mathbf{K}_2$, on en déduit la matrice d'interaction du premier ordre pour une ligne 2D $[\rho \ \theta]^\top$ telle que :

$$\begin{bmatrix} \dot{\rho} \\ \dot{\theta} \end{bmatrix} = \mathbf{L}_{\mathcal{L}_{\rho\theta}} \mathbf{v} \quad (\text{A.83})$$

$$\mathbf{L}_{\mathcal{L}_{\rho\theta}} = \begin{bmatrix} \lambda_\rho \cos(\theta) & \lambda_\rho \sin(\theta) & -\lambda_\rho & (1 + \rho^2) \sin(\theta) & -(1 + \rho^2) \cos(\theta) & 0 \\ \lambda_\theta \cos(\theta) & \lambda_\theta \sin(\theta) & -\lambda_\theta & -\rho \cos(\theta) & \rho \sin(\theta) & -1 \end{bmatrix} \quad (\text{A.84})$$

A.5.3 Hessien pour le modèle du second ordre

La méthode s'appuie fortement sur celle décrite dans la section précédente. On dérive l'équation (A.76) une nouvelle fois :

$$\begin{aligned} -\frac{\dot{\theta}}{\cos(\theta)} \dot{y}_n - \left(\frac{\ddot{\theta}}{\cos(\theta)} + \frac{\dot{\theta}^2 \tan(\theta)}{\cos(\theta)} \right) y_n + \ddot{\rho} + \dot{\rho} \tan(\theta) \dot{\theta} + \rho \tan(\theta) \ddot{\theta} = \\ \ddot{x}_n \cos(\theta) + \ddot{y}_n \sin(\theta) + \dot{\theta} (\dot{y}_n \cos(\theta) - \dot{x}_n \sin(\theta)) \end{aligned} \quad (\text{A.85})$$

En utilisant la matrice d'interaction du point 2D (2.37) et son hessien (3.35), la partie à droite de l'équation peut se réécrire comme suit :

$$\begin{aligned} \ddot{x}_n \cos(\theta) + \ddot{y}_n \sin(\theta) + \dot{\theta} (\dot{y}_n \cos(\theta) - \dot{x}_n \sin(\theta)) \\ = \dot{y}_n (\mathbf{K}_{11} \dot{\mathbf{v}} + \mathbf{K}_{12} \mathbf{v}) + y_n (\mathbf{K}_{21} \dot{\mathbf{v}} + \mathbf{K}_{22} \mathbf{v}) + \mathbf{K}_{31} \dot{\mathbf{v}} + \mathbf{K}_{32} \mathbf{v} \end{aligned} \quad (\text{A.86})$$

Les expressions des matrices $\mathbf{K}$ peuvent être retrouvées grâce à un logiciel de calcul symbolique. Elles sont reportées ci-après :

$$\mathbf{K}_{11} = \mathbf{0} \quad (\text{A.87})$$

$$\mathbf{K}_{12} = \begin{bmatrix} \lambda_1 \cos(\theta) & \lambda_1 \sin(\theta) & \frac{\sin(\theta)}{z} - \lambda_1 \rho & 2\rho + y_n \sin(\theta) & -x_n \sin(\theta) & 2\cos(\theta) \end{bmatrix} \quad (\text{A.88})$$

$$\mathbf{K}_{21} = \begin{bmatrix} \lambda_1 \cos(\theta) & \lambda_1 \sin(\theta) & \frac{\sin(\theta)}{z} - \lambda_1 \rho & \rho + y_n \sin(\theta) & -x_n \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (\text{A.89})$$

$$\begin{aligned} \mathbf{K}_{22} = & \begin{bmatrix} \dot{\lambda}_1 \cos(\theta) - \lambda_1 \dot{\theta} \sin(\theta) & \dot{\lambda}_1 \sin(\theta) - \lambda_1 \dot{\theta} \cos(\theta) \\ -\lambda_1 \dot{\rho} - \dot{\lambda}_1 \rho - \frac{\dot{z}}{z^2} \sin(\theta) - \frac{\dot{\theta}}{z} \cos(\theta) \\ \dot{\rho} + 2\frac{\dot{y}}{z} \sin(\theta) + \omega_3 x_n \sin(\theta) + \dot{\theta} y_n \cos(\theta) + \frac{\dot{x}}{z} \cos(\theta) - \omega_3 \cos(\theta) \\ \rho \omega_z - \omega_1 \cos(\theta) - (2\omega_2 + \dot{x}_n - \omega_3 y_n + \frac{\dot{x}}{z}) \sin(\theta) + \dot{\theta} x_n \cos(\theta) \\ \rho \omega_z - \frac{\dot{z}}{z} \cos(\theta) - \dot{\theta} \sin(\theta) + (2\omega_2 + \omega_1 x_n + \omega_2 y_n) \sin(\theta) \end{bmatrix} \end{aligned} \quad (\text{A.90})$$

$$\mathbf{K}_{31} = \begin{bmatrix} \lambda_2 \cos(\theta) & \lambda_2 \sin(\theta) & \lambda_1 y_n^2 \sin(\theta) - \lambda_2 x_n \cos(\theta) & (1 - y_n^2) \sin(\theta) & (1 - x_n^2) \cos(\theta) & -x_n \sin(\theta) \end{bmatrix} \quad (\text{A.91})$$

$$\begin{aligned}
\mathbf{K}_{32} = & \begin{bmatrix} \dot{\lambda}_2 \cos(\theta) - \lambda_2 \dot{\theta} \sin(\theta) & \dot{\lambda}_2 \sin(\theta) - \lambda_2 \dot{\theta} \cos(\theta) \\
\dot{\lambda}_1 y_n^2 \sin(\theta) + \dot{\lambda}_2 x_n \cos(\theta) + \lambda_1 (2y_n \dot{y}_n \sin(\theta) + \dot{\theta} y_n^2 \cos(\theta)) + \lambda_2 (\dot{x}_n \cos(\theta) + \dot{\theta} x_n \sin(\theta)) \\
\left(\dot{\theta} \cos(\theta) - \omega_3 \cos(\theta) - 2\omega_1 x_n \cos(\theta) - \omega_2 x_n \sin(\theta) - 3y_n \dot{y}_n \sin(\theta) \right. \\
+ \frac{\dot{z}}{z} \sin(\theta) + \omega_3 x_n^2 \cos(\theta) - \dot{\theta} y_n^2 \cos(\theta) - \frac{\dot{z}}{z} y_n^2 \sin(\theta) \Big) \\
\left(\dot{\theta} \sin(\theta) - \omega_3 \sin(\theta) - \omega_1 x_n \sin(\theta) - 3x_n \dot{x}_n \cos(\theta) \right. \\
- \frac{\dot{z}}{z} \cos(\theta) - \omega_3 y_n^2 \sin(\theta) + \dot{\theta} x_n^2 \sin(\theta) - \frac{\dot{z}}{z} x_n^2 \cos(\theta) \Big) \\
(2\omega_3 x_n \cos(\theta) - \omega_2 \sin(\theta) - 2\dot{x}_n \sin(\theta) - \omega_1 \sin(\theta) \\
- \dot{\theta} x_n \cos(\theta) + \omega_1 x_n^2 \cos(\theta) - \omega_2 y_n^2 \sin(\theta) - \frac{\dot{z}}{z} x_n \sin(\theta) \Big) \end{bmatrix}
\end{aligned} \tag{A.92}$$

Du fait de l'utilisation du hessien du point 2D en équation (3.35), on note la présence de paramètres 3D nécessaires à son calcul qui ne peuvent pas se simplifier, à l'instar de la section précédente. Par identification,

$$\begin{cases} \ddot{\theta} = -\mathbf{K}_{21} \cos(\theta) \dot{\mathbf{v}} - \mathbf{K}_{22} \cos(\theta) \mathbf{v} - \mathbf{K}_{12}^2 \mathbf{v}^2 \cos(\theta) \sin(\theta) \\
\ddot{\rho} = \mathbf{K}_{31} \dot{\mathbf{v}} + \mathbf{K}_{32} \mathbf{v} + \dot{\rho} \sin(\theta) \theta \mathbf{K}_{12} \mathbf{v} + \rho \sin(\theta) (\mathbf{K}_{12} \dot{\mathbf{v}} + \mathbf{K}_{22} \mathbf{v} - \mathbf{K}_{12}^2 \mathbf{v}^2) \end{cases} \tag{A.93}$$

Il est toujours possible d'explicitier $[\ddot{\rho} \ \ddot{\theta}]^\top$ sous la forme du second ordre usuelle telle que

$$\begin{bmatrix} \ddot{\rho} \\ \ddot{\theta} \end{bmatrix} = \mathbf{L}_{\mathcal{L}_{\rho\theta}} \dot{\mathbf{v}} + \mathbf{H}_{\mathcal{L}_{\rho\theta}} \mathbf{v} \tag{A.94}$$

Oltre le fait que l'expression de ces matrices n'est pas triviale, la présence de paramètres 3D en relation avec le point 2D qui ne peuvent se simplifier rend l'exercice caduque. En effet, un des intérêts majeurs des matrices d'interaction est d'explicitier celles-ci selon les termes de la représentation. Le modèle du second ordre pour une ligne 2D paramétrée selon des coordonnées polaires n'a pas de solution acceptable du point de vue de la commande référencée capteurs.



Étude des erreurs de saisie

Le fait de saisir un objet est une action qui nous semble très naturelle. Mais nous n'accordons que peu d'importance sur la façon dont nous saisissons les objets. En effet, nos mains agiles nous permettent de manipuler un objet après avoir été saisi et soulevé, ce qu'on appelle communément la manipulation dextre. Celle-ci nous évite donc de nous préoccuper de la position exacte de l'objet dans nos mains.

Cependant, le parallèle n'est pas possible lorsqu'il s'agit de saisie avec un robot. Les effecteurs sont rigides, sous-actionnés et ne permettent pas cette manipulation dextre. Si l'on souhaite réaliser une opération d'assemblage avec un objectif de précision ambitieux, il faut donc garantir que l'action de saisir un objet n'entraîne pas de mouvements indésirables qui pourraient entraîner des erreurs de positionnement à la fin de l'opération.

Dans cette section, a été évalué le comportement réel de l'action de saisie par un robot équipé d'un préhenseur. Nous avons souhaité mesurer ce qu'entraînait la fermeture des doigts sur la position de la pièce une fois immobilisée par le préhenseur, par rapport à celle que nous attendions. On développera le protocole d'expérimentation ainsi que les résultats obtenus.

B.1 Description de l'expérimentation

Le but de l'expérience est d'évaluer la répétabilité d'une opération de saisie lorsque les conditions sont favorables, en répétant l'opération un certain nombre de fois. L'opération est simple et ne présente pas de difficulté particulière. D'une manière générale, la main est configurée pour attraper la pièce comme une pince et réalise une saisie stable à chaque itération. Seulement deux des trois doigts sont utilisés. La pose initiale de la main avant la fermeture des doigts a été choisie de manière à ce qu'aucun déséquilibre lié au centre de gravité de la pièce ne viennent interférer. La figure B.1 montre une vue d'ensemble de l'équipement expérimental.

La pièce à saisir est un pavé droit de dimensions $6 \times 6 \times 9$ centimètres. L'état des surfaces est lisse et ne présente pas d'aspérité aux zones où les doigts du préhenseur seront positionnés. Le poids de la pièce est d'environ 200 grammes.

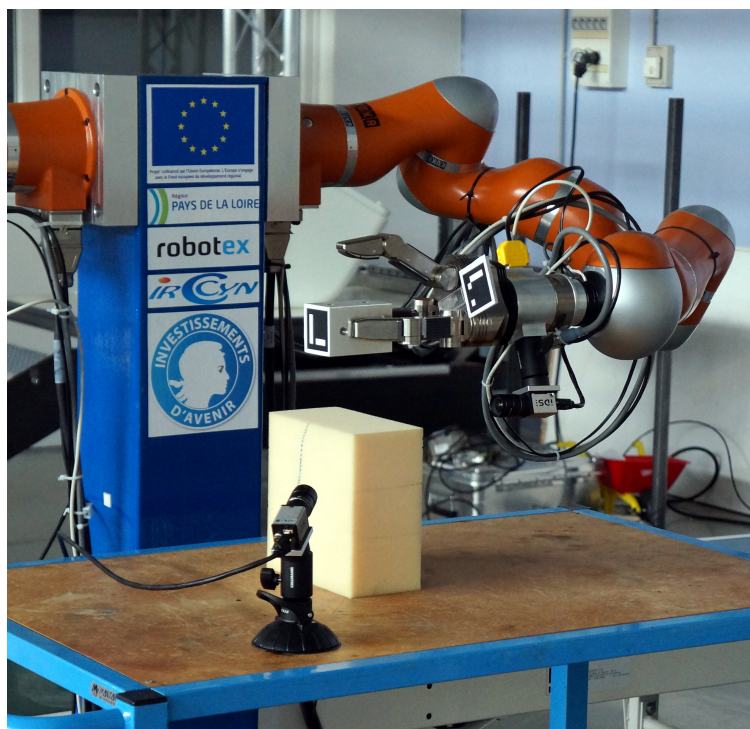


FIGURE B.1 – Vue d'ensemble de l'expérience pour la mesure de la répétabilité d'une saisie.

Pour mener cette expérience, nous avons utilisé un robot KUKA LWR4+ équipé d'une main SCHUNK SDH. Le préhenseur est utilisé de manière à former une pince avec deux de ses trois doigts. Les doigts sont équipés de capteurs tactiles où le matériau de la surface utile des doigts est en caoutchouc noir capable de se déformer localement. Pour donner une idée du coefficient de frottement de ce matériau, celui-ci est grossièrement identique à celui de l'épiderme humain.

Nous avons choisi de tester trois configurations différentes de pince, et à chaque configuration, d'appliquer trois niveaux d'effort : d'une saisie juste (la pièce glisse mais ne tombe pas) à une saisie très ferme, en fonction des capacités du préhenseur. Par ce biais, nous avons voulu simuler ce que pouvait entraîner une mauvaise répétabilité du préhenseur.

La figure B.2 montre les champs de pression des capteurs tactiles pour les neuf combinaisons. Les différentes configurations du préhenseur, nommées A, B et C, ont été choisies de manière à varier la quantité de surface des doigts en contact avec la pièce. Du fait de la géométrie des doigts, on note que la configuration A (figure B.2a) paraît très proche de la configuration B (figure B.2e). Néanmoins, la quantité de surface est très différente entre ces configurations. On observe de plus que l'épicentre des champs de pression est nettement au bord du doigt dans la configuration A, ce qui n'est pas le cas dans la configuration B.

Les mesures de position sont effectuées à l'aide d'une caméra déportée et de marqueurs placés à la fois sur le préhenseur ainsi que sur l'objet. La caméra est placée à proximité pour permettre une estimation des trois paramètres de position et des trois paramètres d'orientation (*i.e.* une pose) de chacun des repères attachés aux marqueurs respectifs.

Pour chaque combinaison de configuration et de force, l'opération de saisie est répétée une centaine de fois pour obtenir un échantillon viable non biaisé.

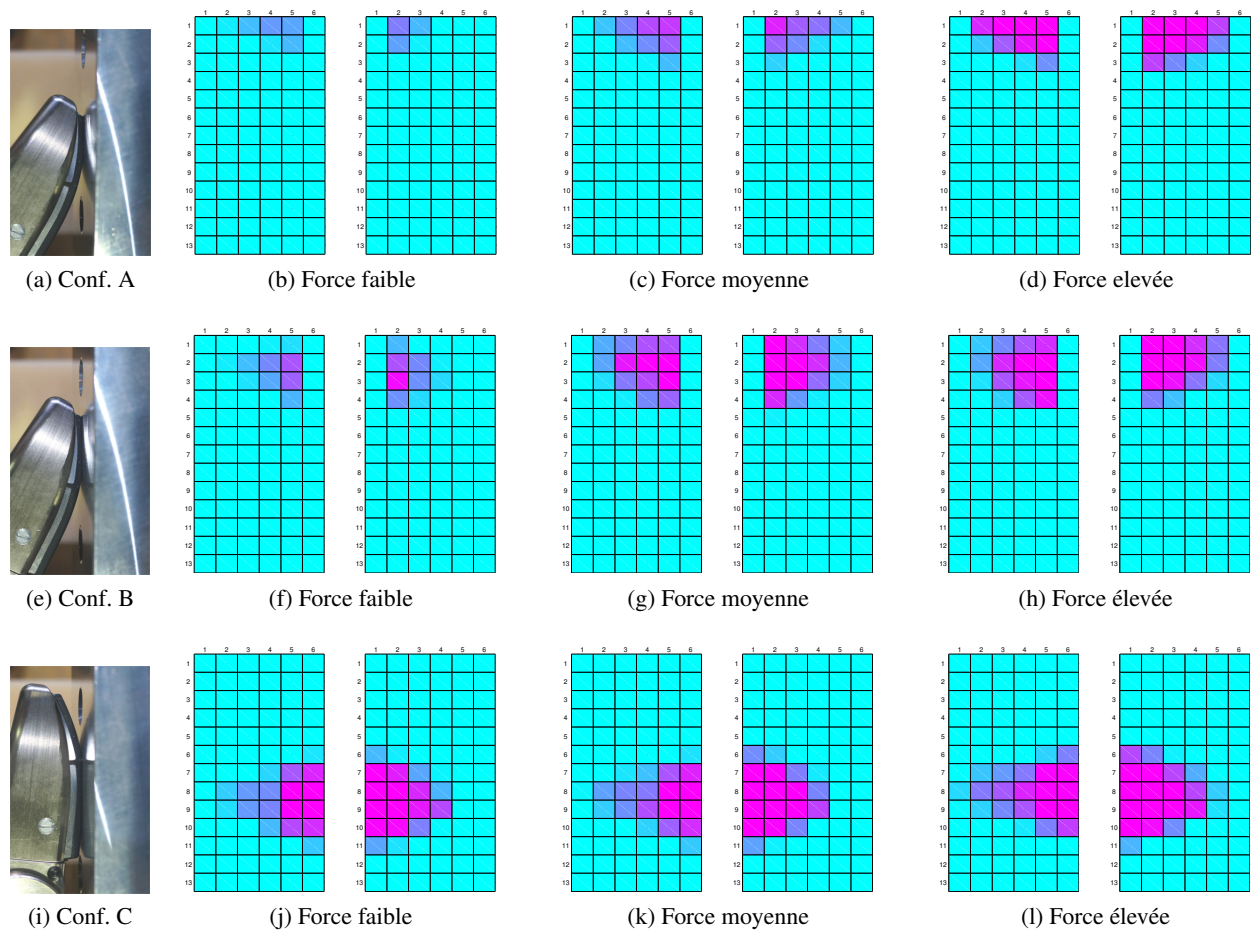


FIGURE B.2 – Colonne de gauche : photographie du doigt gauche, selon les différentes configurations. Colonnes de droite : champs de pression moyens des capteurs tactiles des doigts gauche et droit du préhenseur SCHUNK SDH des surfaces en contact lors des opérations de saisie, selon les configurations utilisées (lignes) et les niveaux d'effort utilisés (colonnes).

B.2 Quatre phases de l'opération

L'opération de saisie se décompose en quatre phases :

- L'approche : le robot effectue une trajectoire d'approche pour placer le préhenseur dans sa position de saisie.
- La fermeture des doigts sur la pièce : le préhenseur passe d'une configuration ouverte à une configuration fermée, tenant compte des différentes combinaisons évoquées ci-dessus.
- Le décollage : le robot effectue un mouvement vers le haut pour soulever la pièce saisie.
- La dépose : le robot repose la pièce en tenant compte, si il y a lieu, du mouvement interne de la pièce dans la main.

Les trois premières phases font l'objet d'un enregistrement des poses des repères de l'objet et de la main. La figure B.3 montre les quatre phases de l'opération du point de vue de la caméra déportée.

La pose utilisée pour établir la trajectoire d'approche lors de la première phase a été déterminée en fonction de la pose désirée entre le repère de la pièce et le repère du préhenseur. À la fin de cette phase d'approche, on considère alors que la pose entre l'objet et le préhenseur est celle que nous souhaitons obtenir.

La deuxième et la troisième phase sont les phases où le mouvement interne le plus susceptible de se produire. La deuxième phase observe un événement de contact entre les surfaces utiles des doigts du préhenseur et celles de la pièce. Ces contacts peuvent engendrer des erreurs de position si par exemple, l'asservissement des doigts n'est pas synchronisé, si la dynamique d'un doigt est plus rapide que l'autre (si l'un vient au contact avant l'autre) ou encore si les matériaux des deux surfaces utiles ne réagissent pas de la même manière au contact, etc.

La troisième phase observe aussi un événement de contact : la perte du contact de la pièce avec son support. Cette perte peut elle aussi engendrer un mouvement de la pièce même si celle-ci est immobilisée par les doigts du préhenseur. Une des causes probables en serait la répétabilité du robot : en effet, même si la pose de l'effecteur à la fin de l'approche a été déterminée pour obtenir la pose désirée entre l'objet et la main, cette pose a été estimée en utilisant le modèle du robot et le mouvement a été effectué sur la base de mesures proprioceptives. En partant du constat effectué en section 1.5, il n'a aucune garantie que le robot se trouve à la position absolue désirée à la fin de son mouvement.

La dernière phase ne comporte pas d'enregistrement. Les mesures de la caméra sont néanmoins utilisées pour permettre au robot de reposer la pièce au même endroit pour préparer la prochaine itération, et ce, en tenant compte du mouvement interne si celui-ci s'est produit.

B.3 Résultats de l'étude

L'étude porte sur l'évaluation de la répétabilité de l'opération de saisie par la mesure de la position de l'objet avant et après avoir été immobilisé par le préhenseur. Cette position est donnée par la mesure des poses de marqueurs situés sur les deux éléments de l'étude, à l'aide d'une caméra calibrée et d'une méthode d'estimation de pose.

Dans le cas idéal, la pose entre le marqueur de l'objet et celui de la main ne devrait pas changer entre la phase d'approche (illustrée par la figure B.3b) et celle de décollage (figure B.3d). Les

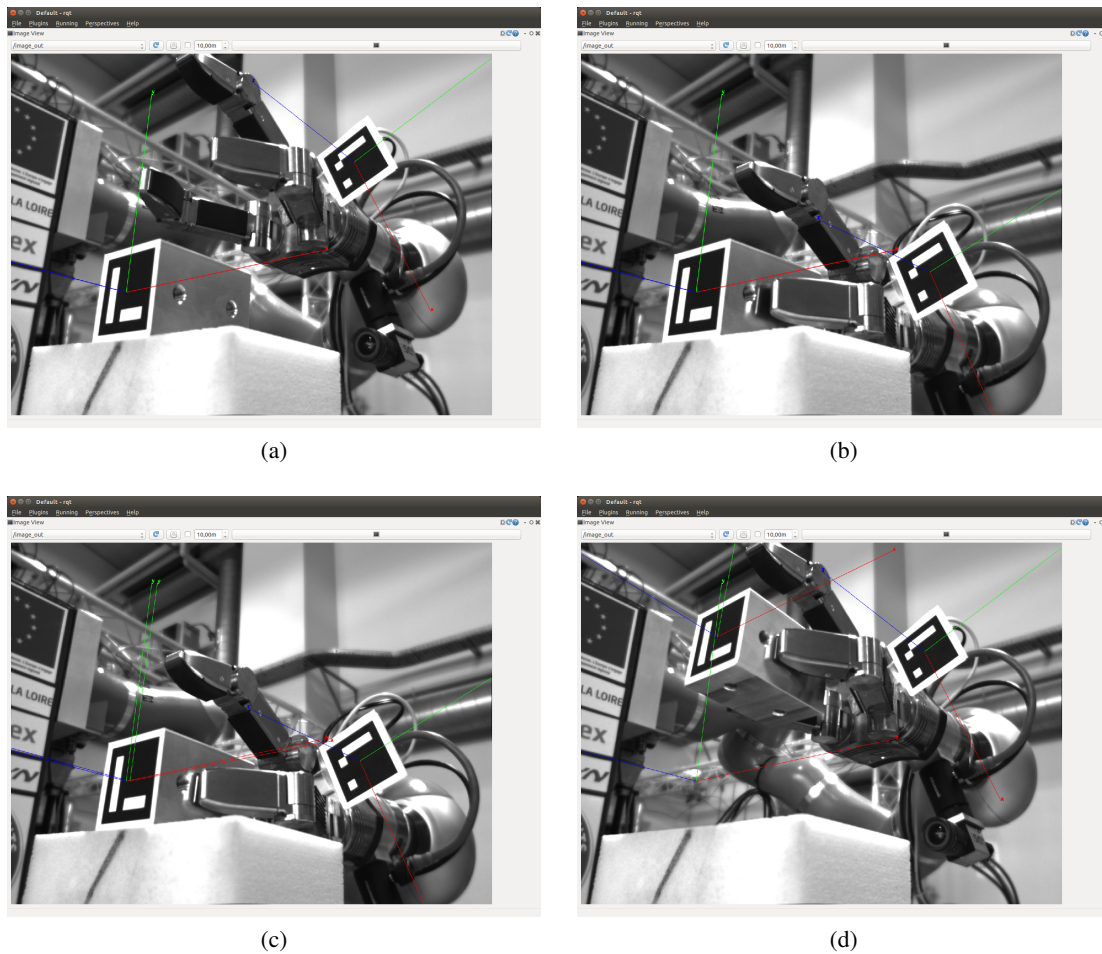


FIGURE B.3 – Prises de vue depuis la caméra déportée avec les repères de l'objet et du préhenseur en superposition. (a) Position initiale du robot au début de la phase d'approche. (b) Préhenseur en position de saisie à la fin de phase d'approche. (c) Fermeture des doigts sur la pièce : le repère de la pièce de la phase précédente a été reporté sur l'image. (d) Phase de décollage.

Force \ Configurations		A	B	C
Faible	$\bar{x}$	7.5366	4.8051	2.3540
	σ	1.2861	1.8956	0.4783
Moyenne	$\bar{x}$	7.6489	3.2683	2.8245
	σ	0.8289	0.3733	0.3796
Élevée	$\bar{x}$	0.8737	2.2998	1.9903
	σ	0.0872	0.3116	0.3461

TABLE B.1 – Moyennes et écart-types en millimètres de la distance entre l’origine du repère de l’objet lors de la phase de pré-saisie (pose désirée) et l’origine du repère de l’objet mesurée à la fin de décollage, exprimée par rapport au repère du préhenseur.

Force \ Configurations		A	B	C
Faible	$\bar{x}$	23.828	-1.3593	10.841
	σ	3.1829	7.1197	2.4178
Moyenne	$\bar{x}$	27.344	6.3663	3.3635
	σ	4.0945	2.3209	2.2216
Élevée	$\bar{x}$	0.7784	15.011	8.7049
	σ	0.6228	1.7995	1.9749

TABLE B.2 – Moyennes et écart-types en millirad de la différence de la rotation selon l’axe x entre le repère de l’objet lors de la phase de pré-saisie (pose désirée) et repère de l’objet mesurée à la fin de la phase de décollage.

mesures sont reportées dans les tableaux B.1 et B.2.

Le premier tableau montre les moyennes et les écarts-types des échantillons pour chaque combinaison configuration/force, de la distance entre le repère de l'objet lors de la phase de pré-saisie et le repère de l'objet lors de la phase de décollage, exprimée par rapport au repère de la main. Les mesures montrent une erreur statique allant de 2mm jusqu'à 7.5mm. La répétabilité, lue à partir de la déviation standard, est comprise entre 0.3mm et 2mm. La combinaison de configuration A avec une force élevée est une exception avec une erreur statique inférieure au millimètre et une variation à plus ou moins 0.09mm.

On rappelle que la configuration A entraîne une surface de contact plus faible que la configuration B, et que la configuration B entraîne une surface de contact plus faible que la configuration C. On observe que plus la surface de contact est grande, plus l'erreur statique de la distance diminue. On peut dire ainsi que la quantité de surface a une incidence positive sur l'erreur statique d'une saisie. De même, le niveau de force appliquée lors de la saisie a une incidence positive sur la variation de la distance : plus la force appliquée est élevée, plus l'écart-type diminue.

Les conclusions que l'on peut amener sur la distance entre les origines ne s'appliquent que partiellement lorsque l'on s'intéresse au second tableau. Le second tableau reporte les différences de la quantité de rotation selon l'axe x des deux repères considérés. De même que pour la distance, on observe que l'écart-type de la différence de la rotation diminue en fonction du niveau de force appliqué. Les erreurs statiques sont quant à elles erratiques et aucun schéma ne se dégage en fonction des combinaisons choisies : elles sont comprises entre 3.3 mrad et 27 mrad, avec une exception pour la combinaison configuration A/force élevée.

Au regard des erreurs statiques, une opération de saisie d'une pièce basé modèle atteint une répétabilité en dessous du centimètre, avec une évolution favorable (*i.e.* une amélioration) de l'erreur de positionnement si la surface de contact entre les doigts et la pièce à saisir augmente et que la force appliquée par les doigts augmente. Néanmoins, l'étude montre que l'action de saisie telle qu'elle a été réalisée pendant l'étude peut introduire une large incertitude si celle-ci est intégrée dans une opération d'assemblage.

Plus généralement, il sera compliqué voire impossible de garantir la précision d'un assemblage réalisé par un robot si la pose de la pièce n'est pas contrôlée après avoir été saisie. Compte tenu des propositions effectuées dans les chapitres précédents, il est aussi tout à fait possible d'exprimer la loi de commande référencée capteurs dans le repère de l'objet saisi afin d'intégrer cette incertitude à l'intérieur de la boucle de régulation, et ainsi se donner la possibilité de les rejeter.

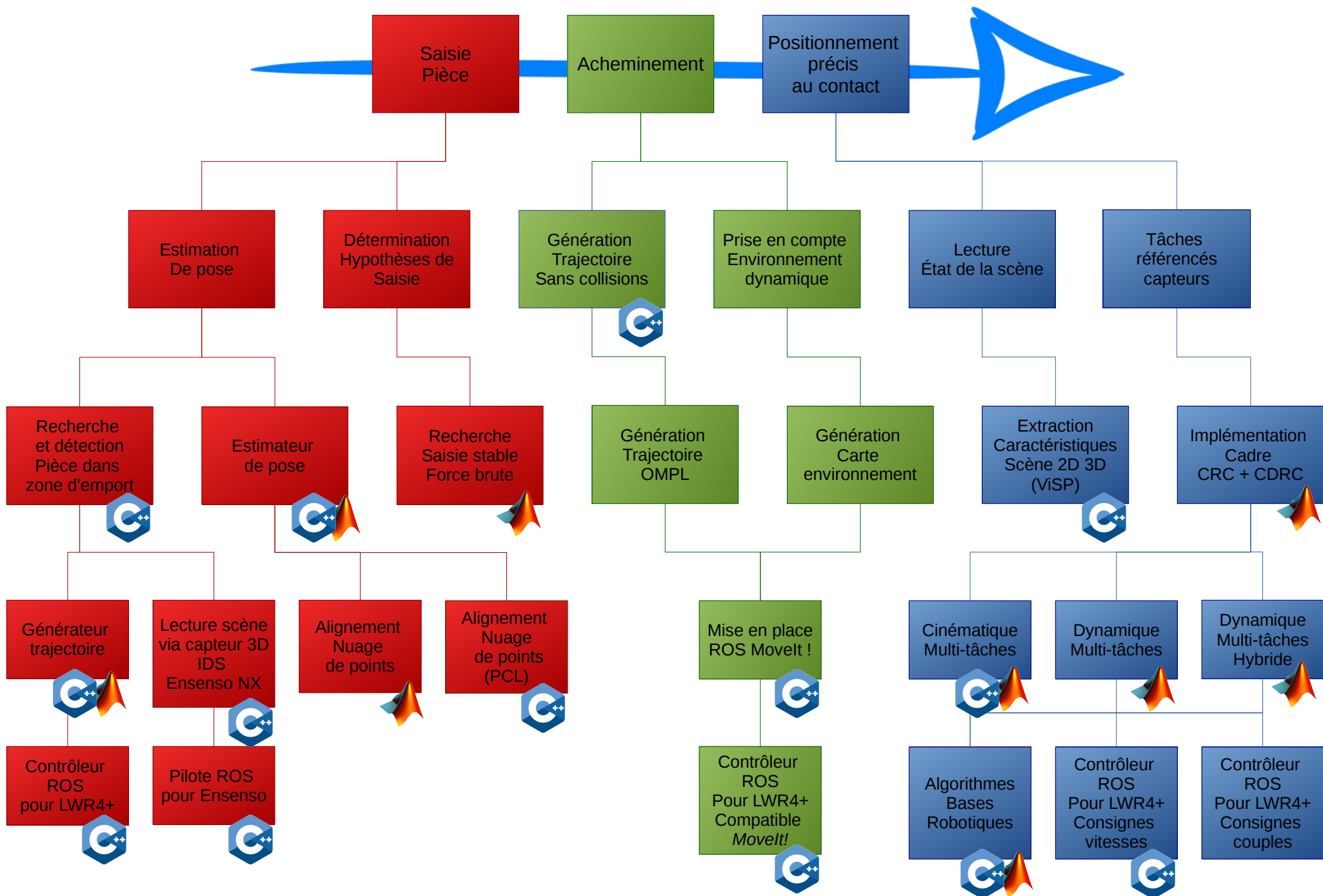


Développement des briques logicielles

Cette annexe présente un état des lieux des différentes briques logicielles qui ont été développées au cours de cette thèse. Une des volontés majeures de ce travail de thèse était de proposer un ensemble de briques logicielles à destination de l'industriel Airbus en vue d'un déploiement de la solution de positionnement sur ses sites d'assemblage d'avions.

Les différents composants se basent sur une combinaison de logiciels à sources ouvertes, de développements ad-hoc et de mise à niveau de logiciels propriétaires pour la réalisation des simulations et des expérimentations. Les simulations ont été principalement réalisées à l'aide du logiciel Matlab, MSC Adams [[Adams, 2016](#)] et Gazebo [[Koenig et al., 2002](#)]. Les simulations sous Gazebo et les expérimentations ont nécessité l'implémentation de briques en C++ en se basant sur des cadres logiciels existants telle que ROS [[Sucan and Chitta, 2016](#)], ViSP [[ViSP, 2016](#)], PCL [[Rusu and Cousins, 2011](#)] et d'autres.

Le diagramme en page suivante montre un pipeline logiciel pour une opération de manipulation complète qui englobe la saisie, l'acheminement et le positionnement d'une pièce manufacturée sur une structure. Chaque bloc possède un certain nombre de dépendances qui doivent être fonctionnelles pour réaliser l'opération.



Bibliographie

- Abou Moughlbay, A. Y. (2013). *Contributions in task sequencing and redundancy resolution, Application to humanoid robots*. Phd thesis, Ecole Centrale De Nantes. 37
- Adams (2016). Msc adams : Multibody dynamics simulation software. <http://www.mscsoftware.com/fr/product/adams>. [Online ; last access in 2016]. 141
- Adams, S. (1776). Recherches sur les causes de la richesse des nations. 1
- Andreff, N., Espiau, B., and Horaud, R. (2002). Visual servoing from lines. *The International Journal of Robotics Research*, 21(8) :679–699. 32, 121
- Antonelli, G. (2009). Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems. *IEEE Transactions on Robotics*, 25(5) :985–994. 37
- Baerlocher, P. and Boulic, R. (2004). An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. *The visual computer*, 20(6) :402–417. 38
- Baker, S. and Matthews, I. (2004). Lucas-kanade 20 years on : A unifying framework. *International journal of computer vision*, 56(3) :221–255. 23
- Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3) :346–359. 33
- Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf : Speeded up robust features. In *Computer vision (ECCV)*, pages 404–417. Springer. 33
- Benchmann, D. and Bernard, P. (2007). *Informatique graphique, modélisation géométrique et animation*. Hermes Science Publications. 42
- Benhimane, S. and Malis, E. (2006). Homography-based 2d visual servoing. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2397–2402. IEEE. 34
- Berger, E., Conley, K., Faust, J., Foote, T., Gerkey, B., Leibs, J., Quigley, M., and Wheeler, R. (2007). Robot Operating System (ROS). <http://www.ros.org/wiki/ROS/>. [Online ; last access in 2016]. 53
- Berner, R., Brandli, C., Yang, M., Liu, S.-C., and Delbruck, T. (2013). A 240x180 120db 10mw 12us-latency sparse output vision sensor for mobile applications. In *Proceedings of the International Image Sensors Workshop*, number EPFL-CONF-200451, pages 41–44. 73

- Bischoff, R., Kurth, J., Schreiber, G., Koeppe, R., Albu-Schäffer, A., Beyer, A., Eiberger, O., Haddadin, S., Stemmer, A., Grunwald, G., et al. (2010). The kuka-dlr lightweight robot arm - a new reference platform for robotics research and manufacturing. In *the 41st International Symposium on Robotics (ISR) and the 6th German Conference on Robotics (ROBOTIK)*, pages 1–8. VDE. [49](#), [84](#), [93](#)
- Bône, J.-C., Boucher, M., and Morel, J. (1984). *Mécanique générale : cours et applications*. Dunod. [117](#)
- Borst, C., Fischer, M., and Hirzinger, G. (1999). A fast and robust grasp planner for arbitrary 3d objects. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 1890–1896. IEEE. [13](#)
- Borst, C., Fischer, M., and Hirzinger, G. (2003). Grasping the dice by dicing the grasp. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 4, pages 3692–3697. IEEE. [13](#)
- Borst, C., Fischer, M., and Hirzinger, G. (2005). Efficient and precise grasp planning for real world objects. In *Multi-point interaction with real and virtual objects*, pages 91–111. Springer. [13](#)
- Brandli, C., Berner, R., Yang, M., Liu, S.-C., and Delbruck, T. (2014). A 240×180 130 db 3 μ s latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10) :2333–2341. [73](#)
- Brock, O., Kuffner, J., and Xiao, J. (2008). Motion for manipulation tasks. In *Springer Handbook of Robotics*, pages 615–645. Springer. [13](#)
- Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). Brief : Binary robust independent elementary features. *Computer Vision (ECCV)*, pages 778–792. [33](#)
- Cervera, E., Berry, F., and Martinet, P. (2002). Is 3d useful in stereo visual control ? In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1630–1635. [34](#)
- Cervera, E., Del Pobil, A., Berry, F., and Martinet, P. (2003). Improving image-based visual servoing with three-dimensional features. *The International Journal of Robotics Research (IJRR)*, 22(10-11) :821–839. [34](#)
- Cervera, E. and Martinet, P. (1999). Combining pixel and depth information in image-based visual servoing. In *Proceedings of the International Conference on Advanced Robotics*, volume 1, pages 445–450. [34](#)
- Chan, T. F. and Dubey, R. V. (1995). A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators. *IEEE Transactions on Robotics and Automation*, 11(2) :286–292. [39](#)
- Chaumette, F. (1990). *La relation vision-commande : théorie et application à des tâches robotiques*. Phd thesis, Université de Rennes I. [17](#), [27](#), [42](#), [72](#), [130](#)
- Chaumette, F. (2004). Image moments : a general and useful set of features for visual servoing. *IEEE Transactions on Robotics*, 20(4) :713–723. [34](#)

- Chaumette, F. and Hutchinson, S. (2006). Visual servo control : I. basic approaches. *IEEE Robotics & Automation Magazine*, 13(4) :82–90. [19](#), [22](#), [30](#)
- Chaumette, F. and Hutchinson, S. (2008). Visual servoing and visual tracking. In Siciliano, B. and Khatib, O., editors, *Springer Handbook of Robotics*, pages 563–583. Springer. [19](#), [22](#)
- Chaumette, F., Hutchinson, S., et al. (2007). Visual servo control, part ii : Advanced approaches. *IEEE Robotics and Automation Magazine*, 14(1) :109–118. [19](#), [22](#)
- Chen, H. and Liu, Y. (2012). Robotic assembly automation using robust compliant control. *Robotics and Computer-Integrated Manufacturing*. [12](#), [13](#), [14](#), [92](#)
- Chesi, G., Hashimoto, K., Prattichizzo, D., and Vicino, A. (2004). Keeping features in the field of view in eye-in-hand visual servoing : a switching approach. *IEEE Transactions on Robotics*, 20(5) :908–914. [38](#)
- Chiaverini, S., Oriolo, G., and Walker, I. D. (2008). Kinematically redundant manipulators. In *Springer Handbook of Robotics*, pages 245–268. Springer. [37](#)
- Chung, W., Fu, L.-C., and Hsu, S.-H. (2008). Motion control. In *Springer Handbook of Robotics*, pages 133–159. Springer. [12](#), [61](#)
- Dahmouche, R. (2010). *Contribution à l'estimation de mouvement 3D et à la commande par vision rapide : Application aux robots parallèles*. Phd thesis, Université Blaise Pascal. [65](#), [66](#)
- Dahmouche, R., Andreff, N., Mezouar, Y., Ait-Aider, O., and Martinet, P. (2012). Dynamic visual servoing from sequential regions of interest acquisition. *The International Journal of Robotics Research (IJRR)*, 31(4) :520–537. [65](#)
- Dahmouche, R., Andreff, N., Mezouar, Y., and Martinet, P. (2009). 3d pose and velocity visual tracking based on sequential region of interest acquisition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5426–5431. IEEE. [65](#)
- De Luca, A., Oriolo, G., and Giordano, P. R. (2006). Kinematic modeling and redundancy resolution for nonholonomic mobile manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1867–1873. IEEE. [39](#), [85](#)
- De Schutter, J. and Van Brussel, H. (1988a). Compliant robot motion i. a formalism for specifying compliant motion tasks. *The International Journal of Robotics Research (IJRR)*, 7(4) :3–17. [96](#)
- De Schutter, J. and Van Brussel, H. (1988b). Compliant robot motion ii. a control approach based on external control loops. *The International Journal of Robotics Research (IJRR)*, 7(4) :18–33. [96](#)
- Debain, C. (1996). *Lois de commande pour le contrôle et la mobilité de machines agricoles*. Phd thesis, Université Blaise Pascal. [130](#)
- Diankov, R. (2010). *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Carnegie Mellon University, Robotics Institute. [12](#)
- 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. [130](#)

- Espiau, B. and Rives, P. (1987). Estimation recursive de primitives 3d au moyen d'une camera mobile. Technical Report 0652, INRIA. [32](#), [70](#), [71](#), [120](#), [124](#), [127](#)
- Ferrari, C. and Canny, J. (1992). Planning optimal grasps. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2290–2295. IEEE. [13](#)
- Folio, D. and Cadenat, V. (2008). Dealing with visual features loss during a vision-based task for a mobile robot. *International Journal of Optomechatronics*, 2(3) :185–204. [38](#)
- Gans, N. R. and Hutchinson, S. A. (2007). Stable visual servoing through hybrid switched-system control. *IEEE Transactions on Robotics*, 23(3) :530–540. [38](#)
- Hafez, A. and Jawahar, C. (2007). Visual servoing by optimization of a 2d/3d hybrid objective function. In *IEEE International Conference on Robotics and Automation*, pages 1691–1696. IEEE. [39](#)
- Horváth, L., Rudas, I. J., and Bitó, J. F. (2002). Form feature based generation of robot assembly paths for product variants. In *IEEE International Conference on Industrial Technology (ICIT'02)*, volume 1, pages 181–186. IEEE. [11](#)
- Horváth, L., Rudas, I. J., and Tar, J. K. (2003). Robot assembly trajectory generation using form feature driven robot process model. In *IEEE International Symposium on Industrial Electronics (ISIE'03)*, volume 2, pages 707–711. IEEE. [11](#)
- Hutchinson, S., Hager, G. D., and Corke, P. I. (1996). A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5) :651–670. [17](#)
- Jubien, A., Gautier, M., and Janot, A. (2014). Dynamic identification of the kuka lwr robot using motor torques and joint torque sensors data. In *World Congress of the International Federation of Automatic Control (IFAC)*, pages 1–6. [85](#)
- Kanoun, O., Lamiroux, F., and Wieber, P.-B. (2011). Kinematic control of redundant manipulators : generalizing the task priority framework to inequality tasks. *IEEE Transactions on Robotics*, 27(4) :785–792. [38](#)
- Kavraki, L. E. and LaValle, S. M. (2008). Motion planning. In *Springer handbook of robotics*, pages 109–131. Springer. [12](#)
- Kermorgant, O. (2011). *Fusion d'informations multi-capteurs en asservissement visuel*. Phd thesis, Université de Rennes I. [vii](#), [29](#), [36](#), [39](#)
- Khalil, W. and Dombre, E. (2002). *Modeling, identification and control of robots*. Butterworth-Heinemann. [viii](#), [12](#), [23](#), [36](#), [61](#), [63](#), [64](#), [75](#), [83](#), [92](#), [93](#), [94](#), [97](#), [99](#), [101](#)
- Khatib, O. (1980). *Commande dynamique dans l'espace opérationnel des robots manipulateurs en présence d'obstacles*. Phd thesis, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace. [64](#)
- Khatib, O. (1987). A unified approach for motion and force control of robot manipulators : The operational space formulation. *IEEE Journal of Robotics and Automation*, 3(1) :43–53. [viii](#), [64](#), [65](#), [93](#), [95](#)

- Koenig, N., Su, J., Dolha, M., and Howard, A. (2002). Gazebo : Opensource multi robot simulator. <http://www.gazebo-sim.org/>. [Online ; last access in 2016]. 53, 141
- Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., and Burgard, W. (2011). g2o : A general framework for graph optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3607–3613. IEEE. 23
- Lichtsteiner, P., Posch, C., and Delbruck, T. (2008). A 128×128 120 db 15 us latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2) :566–576. 73, 89
- Long, P. (2014). *Contributions to the modeling and control of cooperative manipulators*. Phd thesis, École Centrale de Nantes. 96
- Long, P., Khalil, W., and Martinet, P. (2014a). Force/vision control for robotic cutting of soft materials. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4716–4721. IEEE. 96
- Long, P., Khalil, W., and Martinet, P. (2014b). Robotic cutting of soft materials using force control & image moments. In *Proceedings of the 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, pages 474–479. IEEE. 34
- Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the 7th IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157. Ieee. 33
- Malis, E. (2004). Improving vision-based control using efficient second-order minimization techniques. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1843–1848. IEEE. 28
- Malis, E. and Chaumette, F. (2000). 2 1/2 d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *International Journal of Computer Vision*, 37(1) :79–97. 23, 34
- Mansard, N. (2006). *Enchainement de tâches robotiques*. Phd thesis, Université de Rennes I. 36
- Mansard, N., Khatib, O., and Kheddar, A. (2009a). A unified approach to integrate unilateral constraints in the stack of tasks. *IEEE Transaction on Robotics*, 25(3). 38, 39, 64, 65
- Mansard, N., Stasse, O., Evrard, P., and Kheddar, A. (2009b). A versatile generalized inverted kinematics implementation for collaborative working humanoid robots : The stack of tasks. In *International Conference on Advanced Robotics (ICAR)*, pages 1–6. IEEE. 38, 64
- Marchand, E., Spindler, F., and Chaumette, F. (2005). Visp for visual servoing : a generic software platform with a wide class of robot control skills. *IEEE Robotics and Automation Magazine*, 12(4) :40–52. 99
- Marey, M. (2010). *Contributions to control modeling in visual servoing, task redundancy, and joint limits avoidance*. PhD thesis, Université Rennes I. 37
- Mezouar, Y. and Chaumette, F. (2002). Path planning for robust image-based control. *IEEE Transactions on Robotics and Automation*, 18(4) :534–549. 77

- Mezouar, Y., Prats, M., and Martinet, P. (2007). External hybrid vision/force control. In *Proceedings of the International Conference on Advanced Robotics (ICAR'07)*. 96, 97
- Miller, A. T. and Allen, P. K. (2004). Graspit! a versatile simulator for robotic grasping. *Robotics & Automation Magazine, IEEE*, 11(4) :110–122. 13
- Nadeau, C. and Krupa, A. (2011). Improving ultrasound intensity-based visual servoing : Tracking and positioning tasks with 2d and bi-plane probes. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2837–2842. 34, 36
- Nadeau, C., Ren, H., Krupa, A., and Dupont, P. (2015). Intensity-based visual servoing for instrument and tissue tracking in 3d ultrasound volumes. *IEEE Transactions on Automation Science and Engineering*, 12(1) :367–371. 34
- Nakamura, Y. (1990). *Advanced robotics : redundancy and optimization*. Addison-Wesley Longman Publishing Co., Inc. 39
- Nakamura, Y., Hanafusa, H., and Yoshikawa, T. (1987). Task-priority based redundancy control of robot manipulators. *The International Journal of Robotics Research (IJRR)*, 6(2) :3–15. 39
- Ott, C., Dietrich, A., and Albu-Schäffer, A. (2015a). Prioritized multi-task compliance control of redundant manipulators. *Automatica*, 53 :416–423. 92
- Ott, C., Mukherjee, R., and Nakamura, Y. (2015b). A hybrid system framework for unified impedance and admittance control. *Journal of Intelligent & Robotic Systems*, 78(3-4) :359–375. 13, 92
- Pages, J., Collewet, C., Chaumette, F., and Salvi, J. (2004). Plane-to-plane positioning from image-based visual servoing and structured light. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 1004–1009. 32
- Pages, J., Collewet, C., Chaumette, F., Salvi, J., et al. (2005). Visual servoing by means of structured light for plane-to-plane positioning. Technical Report 5579, INRIA. 32
- Park, J. (2006). *Control strategies for robots in contact*. Phd thesis, Stanford University. 38, 64
- Perdereau, V. (2013). Handle and beyond : Advances in in-hand manipulation. In *Research colloquia of the Department of Informatics*, King's college, London. <http://www.handle-project.eu/>. 13
- Perdereau, V. and Drouin, M. (1993). A new scheme for hybrid force-position control. *Robotica*, 11(05) :453–464. 96
- Peternel, L., Petric, T., and Babic, J. (2015). Human-in-the-loop approach for teaching robot assembly tasks using impedance control interface. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1497–1502. IEEE. 13
- Posch, C., Matolin, D., and Wohlgenannt, R. (2011). A qvga 143 db dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds. *IEEE Journal of Solid-State Circuits*, 46(1) :259–275. 73
- Prats, M. (2009). *Robot Physical Interaction through the combination of Vision, Tactile and Force Feedback*. Phd thesis, Jaume I University. 36, 96

- Prats, M., Martinet, P., del Pobil, A. P., and Lee, S. (2007). Vision force control in task-oriented grasping and manipulation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1320–1325. IEEE. [viii](#), [96](#), [97](#), [101](#)
- Prattichizzo, D. and Trinkle, J. C. (2008). Grasping. In *Springer Handbooks of Robotics*, pages 671–700. Springer. [13](#)
- Raibert, M. H. and Craig, J. J. (1981). Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement and Control*, 103(2) :126–133. [93](#)
- Rusu, R. B. and Cousins, S. (2011). 3d is here : Point cloud library (pcl). In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. [54](#), [141](#)
- Samson, C., Espiau, B., and Borgne, M. L. (1991). *Robot control : the task function approach*. Oxford University Press. [22](#), [37](#), [61](#)
- Schoen, T. R. and Rus, D. (2013). Decentralized robotic assembly with physical ordering and timing constraints. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5764–5771. [vii](#), [10](#), [11](#), [13](#)
- Sentis, L. (2007). *Synthesis and control of whole-body behaviors in humanoid systems*. Phd thesis, Stanford University. [38](#)
- Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2010). *Robotics : modelling, planning and control*. Springer Science & Business Media. [12](#), [61](#)
- Siciliano, B. and Slotine, J.-J. (1991). A general framework for managing multiple tasks in highly redundant robotic systems. In *Proceedings of the Fifth International Conference on Advanced Robotics (ICAR) : Robots in Unstructured Environments*, pages 1211–1216. [37](#)
- Silveira, G. (2014). On intensity-based nonmetric visual servoing. *IEEE Transactions on Robotics*, 30(4) :1019–1026. [34](#)
- 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. [34](#)
- Silveira, G., Mirisola, L., and Morin, P. (2013). Decoupled direct visual servoing. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 71–76. IEEE. [34](#)
- Stein, D., Schoen, T. R., and Rus, D. (2011). Constraint-aware coordinated construction of generic structures. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4803–4810. [vii](#), [10](#), [11](#)
- Strasdat, H. (2012). *Local accuracy and global consistency for efficient visual slam*. PhD thesis, Citeseer. [23](#)
- Strasdat, H., Montiel, J., and Davison, A. J. (2010). Scale drift-aware large scale monocular slam. In *Robotics : Science and Systems*, volume 2, page 5. [23](#)
- Sucan, I. A. and Chitta, S. (2016). Moveit! <http://moveit.ros.org/>. [Online ; last access in 2016]. [141](#)

- Şucan, I. A., Moll, M., and Kavraki, L. E. (2012). The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4) :72–82. <http://ompl.kavrakilab.org>. 12
- Szeliski, R. (2006). Image alignment and stitching : A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1) :1–104. 23
- Teuliere, C. and Marchand, E. (2012). Direct 3d servoing using dense depth maps. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1741–1746. IEEE. 32, 34, 35
- Tsai, R. Y. and Lenz, R. K. (1989). A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5(3) :345–358. 28
- Vandernotte, S., Chriette, A., Suarez Roos, A., and Martinet, P. (2014). Performing assembly task under constraints using 3d sensor-based control. In *Proceedings of the 13th International Conference on Intelligent Autonomous System (IAS13)*. 32, 48
- Villani, L. and De Schutter, J. (2008). Force control. In Siciliano, B. and Khatib, O., editors, *Springer Handbook of Robotics*, pages 563–583. Springer. 13, 92, 101
- ViSP (2016). Visp website. <http://visp.inria.fr/>. [Online ; last access in 2016]. 99, 141
- Wang, L., Givehchi, M., Schmidt, B., and Adamson, G. (2012). Robotic assembly planning and control with enhanced adaptability. *45th Conference on Manufacturing Systems (CIRP)*, 3 :173–178. 11
- Wang, L., Keshavarzmanesh, S., and Feng, H.-Y. (2008). Design of adaptive function blocks for dynamic assembly planning and control. *Journal of Manufacturing Systems*, 27(1) :45–51. 11
- Werfel, J. and Nagpal, R. (2008). Three-dimensional construction with mobile robots and modular blocks. *The International Journal of Robotics Research (IJRR)*, 27(3-4) :463–479. 10
- You, J.-S., Kim, D.-H., Lim, S.-J., Kang, S.-P., Lee, J. Y., and Han, C.-S. (2012). Development of manipulation planning algorithm for a dual-arm robot assembly task. In *IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1061–1066. IEEE. vii, 12

Thèse de Doctorat

Sylvain VANDERNOTTE

Manipulation référencée multi-capteurs d'objets manufacturés

Sensor-based manipulation of manufactured parts

Résumé

Cette thèse s'inscrit dans le cadre du projet ASIMOV où le but est de concevoir un robot collaboratif pour aider les opérateurs dans leurs tâches d'assemblage. La problématique se concentre sur la prise en compte de l'environnement au sein de la loi de commande par la commande référencée capteurs. Cette commande est connue pour permettre une très bonne précision absolue. Néanmoins, celle-ci reste difficile à mettre en œuvre et ne garantit pas un contrôle complet du comportement du robot. Nous proposons d'abord une approche systématique pour la génération de tâches robotiques en généralisant le recours aux méthodes de commande référencée capteurs. À partir de l'opération à réaliser, la lecture de l'environnement et de l'équipement du robot (préhenseur et capteurs), plusieurs tâches sont créées et mises en commun dans un formalisme multi-tâches pour réaliser le positionnement précis. Nous proposons ensuite une nouvelle formulation dynamique de la commande référencée capteurs pour contrôler le comportement de convergence du robot. Cette formulation est compatible avec les schémas de commandes hybrides pour la gestion des interactions. Les approches ont été validées en simulation, le but à terme étant le déploiement de la solution sur une cellule robotique composée d'un KUKA LWR4+ et d'un préhenseur.

Mots clés

Manipulation, commandes référencée capteurs, multi-capteurs, multi-tâches, commandes dynamiques, commandes hybrides

Abstract

This thesis is part of the ASIMOV project where the goal is to design a collaborative robot to help operators in their assembly tasks. The problematic focuses on the consideration of the environment inside the command law using sensor-based control methods. This methods are known to provide a good absolute precision. Nevertheless, robotic tasks are difficult to set up and do not provide full control of the robot behavior. We propose first a systematic approach for generation of the robotic tasks generalizing the use of sensor-based control methods. Based on the operation specifications, the environment state and the robot equipment (gripper and sensors), several tasks are created and mixed within a multi-task framework to realize the positioning task. Secondly, we propose a novel dynamic formulation of the sensor-based control law to control the convergence behavior of the robot. This formulation is compatible with hybrid control schemes to manage the interaction with the environment. These approaches has been validated through simulations. The aim of this work is to deploy the solution on a real robotic cell composed of a KUKA LWR4+ and a gripper.

Key Words

Manipulation, sensors-based control, multi-sensors, multi-tasks, dynamic control, hybrid control
