



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Prédiction du mouvement humain pour la robotique collaborative : du geste accompagné au mouvement corps entier.

THÈSE

présentée et soutenue publiquement le 17 décembre 2018

pour l'obtention du

Doctorat de l'Université de Lorraine
(mention informatique)

par

Oriane Dermey

Composition du jury

<i>Rapporteurs :</i>	Rachid Alami	Directeur de recherche CNRS, LAAS
	David Daney	Chargé de recherche INRIA, Bordeaux
<i>Examinatrice :</i>	Catherine Pelachaud	Directrice de recherche, CNRS, Sorbonne Université
<i>Encadrants :</i>	Serena Ivaldi	Chargée de recherche INRIA, LORIA
	François Charpillet	Directeur de recherche INRIA, LORIA

Remerciements

Je souhaite tout d'abord remercier les membres du Jury. Merci à David Daney et Rachid Alami d'avoir accepté d'être les rapporteurs de cette thèse. Merci également à Catherine Pelachaud d'avoir accepté de participer à mon jury.

Je remercie aussi mes directeurs de thèse, pour leurs conseils et leur aide, tout au long de ma thèse. Merci à Serena pour sa présence constante et pour m'avoir poussé quand il le fallait et François pour ses conseils et discussions.

Merci à tous les membres de l'équipe Larsen, pour leur soutien, leur aide scientifique et leurs discussions enrichissantes. Je souhaite particulièrement remercier Dorian, pour m'avoir aidée de nombreuses fois à réparer le robot iCub et Olivier pour m'avoir aidée dans mes manipulations. De même, un grand merci à Francis, Iñaki, Pauline, Vladislav et Lucien et pour leurs remarques et corrections pertinentes. Dans cette équipe, je souhaite finalement remercier Véronique Constant qui m'a aidée administrativement pendant ces trois années de thèse.

Merci à tous ceux qui m'ont permis d'effectuer cette thèse dans les meilleures conditions, que ce soit Sebastien et ses collègues grâce à qui je ne suis pas morte de chaud l'été dans ma salle ; Caroline qui m'a tenue éveillée grâce à ses cafés et sa personnalité si vivante ; Isabelle qui m'a reboostée chaque midi grâce à sa gentillesse, sa bonne humeur et ses pâtisseries. Merci à elles et à Véronique d'avoir accepté d'être les stars dans les vidéos illustrant ma thèse. Et merci à tout le personnel du restaurant, de l'entretien et aux secrétaires pour leur aide permanente et leur gentillesse.

Finalement, merci à mes proches qui m'ont aidé au cours de cette thèse : ma petite maman, pour sa relecture et son intérêt pour mon mémoire ; Manu, qui m'a soutenu et aidé tout au long de cette thèse ; et Ludovic de m'avoir aidé pendant ma thèse et m'avoir permis de prendre du recul, ainsi qu'à tous ceux qui m'ont permis de me sortir de ma bulle.

English speakers

I would like to thank people from the IIT, especially Julien Jenvrin, Daniele Pucci and Silvio Traversaro for their help with the iCub robot.

I would also like to thank the people from Darmstadt, especially Alexandros Paraschos who helped me with the ProMPs, but also Marco Ewerton : I really appreciate you taking time to share your knowledge, despite the fact I was so bad in English.

Finally, thank you to all Larsen Team people, who helped me and discussed with me. A big thanks to Valerio, who reassured me at the beginning of my thesis and Debaleena who allows me to have good conversations and who supports me.

Sommaire

Introduction	1
1 Contexte	1
2 Objectif à long terme	3
3 Thématiques de recherche	4
4 Problématiques	5
4.1 Avantages	5
5 Contribution	5
6 Projets européens CoDyCo et AnDy	8
6.1 CoDyCo	8
6.2 AnDy	9
7 Organisation de la thèse	9
Bibliographie	13
8 Présentations diverses	13
Chapitre 1	
État de l'art	
1.1 Intention et intelligibilité	16
1.1.1 Théories sur l'intention	17
1.1.2 L'intentionnalité en sciences humaines	19
1.1.3 L'intentionnalité en robotique	20
1.2 Prédiction Inférence et Anticipation	21
1.2.1 Définitions	22
1.2.2 L'anticipation en robotique	22
1.2.3 La prédiction en robotique	23
1.3 Interaction physique	24
1.3.1 Techniques robotiques permettant l'interaction physique entre l'humain et le robot	24
1.3.2 Techniques d'interaction physique avec le robot iCub	25

1.4	Acceptabilité du point de vue de l'utilisateur et rôle du robot	26
1.4.1	Rôle du robot et type d'autonomie	27
1.5	Comprendre l'intention par le regard	28
1.5.1	L'attention conjointe chez l'humain	28
1.5.2	La direction du regard en robotique	29
1.6	Perceptions multimodales	30
1.6.1	Multimodalité chez l'humain	31
1.6.2	Multimodalité en robotique	31
1.7	Primitives de Mouvements	33
1.7.1	Modélisation	33
1.7.2	Justification du choix de modélisation	40
1.7.3	Durée des Mouvements	43
1.7.4	Logiciels Open Source Connexes	45
1.8	Compression dans un espace latent	45
1.8.1	Données indépendantes du temps	45
1.8.2	Données avec dépendance temporelle	48

Chapitre 2

Matériels

2.1	Robot humanoïde iCub	49
2.2	Logiciels utilisés du <i>iCub</i>	50
2.3	Autres logiciels utilisés	51
2.4	Xsens	52
2.5	Logiciels réalisés	52
2.5.1	Logiciel de prédiction de mouvements et d'exécution de gestes robotiques	52
2.5.2	Logiciel de prédiction des mouvements humains corps-complet	53

Chapitre 3

Méthodes

3.1	Hypothèses d'étude	55
3.2	Notations	56
3.3	Primitives de mouvements probabilistes (ProMPs)	58
3.3.1	Modèle appliqué à une trajectoire simple	58
3.3.2	Apprentissage d'une ProMP	59
3.3.3	Prédiction de la poursuite d'un mouvement	60
3.3.4	Prédiction de la durée de la trajectoire souhaitée	61
3.3.5	Reconnaître une primitive de mouvement parmi celles apprises	65

3.4	ProMPs appliquées à des données à grande dimension	67
3.4.1	Compression des données à l'aide d'un auto-encodeur classique	67
3.4.2	Compression de dimension à l'aide de la méthode <i>VTSFE</i>	68
3.4.3	Prédiction de la poursuite de trajectoires compressées	70

Chapitre 4

Compréhension de l'interaction physique humain-robot (Étude 1)

4.1	Cadre théorique adapté à l'apprentissage de trois primitives multidimensionnelles	75
4.1.1	Apprentissage de Primitives de Mouvement	75
4.1.2	Prédiction de l'évolution de la trajectoire à partir d'observations initiales .	76
4.2	Application à l'aide de la simulation du robot	76
4.3	Application sur le robot <i>iCub</i> réel	78
4.3.1	Apprentissage de trois ProMPs avec informations sur les forces/couples . .	80
4.3.2	Tri collaboratif d'objets	88
4.4	Améliorations envisagées et étude complémentaire	89
4.5	Conclusions	91

Chapitre 5

Multimodalité dans l'interaction humain-robot (Étude 2)

5.1	Résumé	95
5.2	Introduction	96
5.3	Formulation du Problème	97
5.4	Cadre théorique adapté à l'application : apprentissage de primitives multimodales	99
5.5	Application au robot réel	99
5.5.1	Apprentissage des ProMPs multimodales	100
5.5.2	Activation des primitives à l'aide du regard	101
5.5.3	Prédiction de la trajectoire désirée à l'aide de mesures cinématiques . . .	103
5.5.4	Prédiction de la trajectoire désirée à l'aide de mesures multimodales . . .	104
5.6	Conclusions	106

Chapitre 6

Compréhension de mouvements corps-complet humains (Étude 3)

6.1	Résumé	110
6.2	Introduction	110
6.3	Formulation du problème	111
6.3.1	ProMPs-seuls	112
6.3.2	AE-ProMPs	113

6.3.3	VTSFE-ProMPs	114
6.4	Résultats et Discussion	114
6.4.1	Prédiction du futur d'un mouvement, à l'aide de ProMPs seulement	115
6.4.2	Prédiction du mouvement future, à l'aide de <i>AE-ProMPs</i>	116
6.4.3	Prédiction du mouvement futur à l'aide de <i>VTSFE-ProMPs</i>	118
6.4.4	Précision vs temps de calcul	121
6.5	Conclusion	121

Chapitre 7 Conclusions

7.1	Discussion	125
7.2	Perspectives	129

Annexes

Annexe A Tutoriel d'utilisation du logiciel
--

A.1	Présentation du logiciel de prédiction de mouvements	131
A.1.1	Configurations du robot simulé	131
A.1.2	Acquisition de données	132
A.2	Exemple fourni par le logiciel : apprentissage d'une primitive à une dimension	133
A.3	Enregistrement de données avec l'iCub simulé	136
A.4	Tutoriel application sur l'iCub simulé	137
A.4.1	Apprentissage de trois ProMPs	137
A.4.2	Prédiction du mouvement désiré	138
A.5	Application sur l'iCub réel	139
A.5.1	Trois actions simples, avec informations sur les forces et les couples	139

Annexe B Informations concernant la matrice de fonctions de base radiale

Annexe C Informations concernant les formules de prédiction
--

Bibliographie	145
----------------------	------------

Table des figures

1	Avancée dans la robotique	1
2	Exemples de robots humanoïdes développés dans le monde.	2
3	Exemples d’application de la problématique de cette thèse.	4
1.1	Différence entre prédiction et anticipation.	23
1.2	Progression développementale du suivi de regard, inspiré de Scassellati 2002. . .	28
1.3	Exemple d’un simple et schématique réseau de neurones.	34
1.4	Exemple simplifié d’un graphe HMM	36
1.5	Différence entre les modèles GP et ProMP.	42
1.6	Différentes méthodes basées sur les autoencodeurs.	47
2.1	Robot <i>iCubNancy01</i> sur lequel les expériences ont été effectuées.	50
2.2	<i>iCub</i> simulé dans Gazebo.	50
2.3	Représentation du iCub dans <i>iCubGui</i>	51
2.4	Détection de l’orientation du regard à l’aide du logiciel <i>Intraface</i>	51
2.5	Représentation de la combinaison Xsens.	52
2.6	Architecture du logiciel.	53
2.7	Architecture du 2nd logiciel.	54
3.1	Exemple d’une Primitive de Mouvement Probabiliste	60
3.2	ProMP et fonctions de base radiale correspondantes	61
3.3	Exemples de prédiction d’une trajectoire	62
3.4	Erreur de prédiction de la durée d’une trajectoire.	63
3.5	Prédictions d’une trajectoire à partir de différentes techniques.	66
3.6	Relation entre les méthodes AEs et VTSFE.	69
3.7	Schéma de comparaison entre les méthodes VTSFE et VAE-DMP.	70
3.8	Concept de la méthode de prédiction de trajectoires à grande dimension	71
3.9	Schéma du concept des méthodes AE-ProMPs ou VTSFE-ProMPs.	72
4.1	Communication physique naturelle entre deux personnes	73
4.2	Concept d’utilisation des ProMPs.	74
4.3	Étude en simulation - Scénario.	77
4.4	Étude en simulation - 3 ProMPs apprises.	78
4.5	Étude en simulation - Résultats.	79
4.6	1e étude robot réel - Scénario.	80
4.7	Représentation du iCub dans <i>iCubGui</i>	81
4.8	1e étude robot réel - Représentation des ProMPs apprises.	82
4.9	1e étude robot réel - Exemple de prédictions.	83

4.10	1e étude robot réel - Exemple de prédictions prenant en compte les forces.	84
4.11	1e étude robot réel - Résultat avec ou sans l'information des forces.	86
4.12	2eme expérience robot réel - Scénario.	88
4.13	2eme expérience robot réel - Prédictions.	93
5.1	Communication visuelle naturelle entre humains.	95
5.2	2eme étude - Scénario	97
5.3	2eme étude - Représentation conceptuelle.	98
5.4	ProMPs multimodales.	100
5.5	Exemple de prédiction depuis l'information visuelle.	101
5.6	Schéma représentant l'intérêt des primitives visuelles.	101
5.7	Analyses du guidage visuel.	102
5.8	Exemple de prédiction d'une trajectoire à partir de guidage physique.	103
5.9	Analyses du guidage physique.	104
5.10	Exemple de prédiction multimodale.	105
5.11	Erreur de prédiction de la position cartésienne.	105
5.12	Erreur de prédiction selon la modalité utilisée.	106
6.1	Scénario représentant la capacité de prédiction humaine.	109
6.2	Concept de la 3e étude	112
6.3	3e étude - Scénario.	113
6.4	ProMPs-seulement	115
6.5	AE-ProMPs - Images résultats.	116
6.6	AE-ProMPs - Distribution de l'espace latent 5D.	116
6.7	AE-ProMPs - Précision de l'inférence.	117
6.8	AE-ProMPs - Statistiques générales	119
6.9	VTSFE - Résultats.	120
6.10	VTSFE-ProMP - Statistiques générales.	123
A.1	Présentation des boutons du Geomagic Touch.	132

Introduction

1 Contexte



FIGURE 1 – *Avancée dans la robotique – Du bras robotique industriel isolé (gauche), aux robots compliants (milieu), aux robots collaboratifs, capables de comprendre la volonté de leur utilisateur (droite).*

Que ce soit dans l'industrie, dans les milieux hospitaliers ou dans les maisons de retraite, de nombreuses tâches sont répétitives, peu ergonomiques (port de charges lourdes, mauvais positionnement du corps, etc.) voire quelques fois exténuantes. Les troubles induits sont responsables de nombreux arrêts de travail, qui sont parfois permanents et ont des conséquences médicales, sociales et économiques très néfastes.

La robotisation devrait permettre à terme d'améliorer la qualité de travail des individus, que ce soit en améliorant l'ergonomie des postes de travail, ou en soulageant les travailleurs des tâches désagréables. Aujourd'hui, les bras robotiques font une entrée massive dans beaucoup de secteurs de l'industrie (*c.f.*, première image de la Figure 1). Cependant, alors que les robots devraient s'adapter à leurs utilisateurs dans le but de les aider, actuellement c'est plutôt l'inverse qui se produit. En effet, lorsque l'on regarde comment les robots et automates sont utilisés dans les entreprises, on peut voir que ce sont les êtres humains qui s'adaptent et suivent la cadence des robots et non l'inverse. C'est par exemple le cas dans les chaînes de montage, comme le montre le film de Chaplin [45], ou encore des véhicules robotisés dans les entrepôts, comme le présente l'enquête récente de cash-investigation [73].

De plus, les robots actuels doivent le plus souvent fonctionner en environnement contrôlé. De ce fait, ils n'effectuent pas de tâches coopératives avec les opérateurs humains : les humains et les robots travaillent chacun de leur côté. On pourrait même considérer qu'ils sont en compétition. Une raison de cet état de fait est liée aux impératifs de sécurité : un robot est *a priori* **dangereux** s'il ne peut pas comprendre et anticiper l'activité humaine qui a lieu dans son environnement. Si






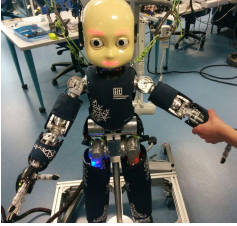



<p>HONDA</p>  <p>Asimo 1m20 / 50kg</p>	<p>Boston Dynamics</p>  <p>Atlas 1m50 / 75kg</p>	<p>Institut national de sciences & tech. Industrielles Avancées du Japon</p>   <p>HRP-4C Uroa HRP-2 1m58 / 43kg 154cm / 58kg</p>	
<p>PAL Robotics</p>  <p>REEM-C 1m70 / 100kg</p>	<p>Istituto Italiano di Tecnologia</p>  <p>iCub 1m04 / 22kg</p>	<p>SoftBank Robotics</p>    <p>Pepper NAO ROMEO 1m20 / 28kg 58cm / 5.3kg 1m40</p>	

FIGURE 2 – Exemples de robots humanoïdes développés dans le monde.

aujourd'hui de gros progrès ont été réalisés grâce à l'apparition de robots compliants, beaucoup reste à faire pour **garantir la sécurité** des travailleurs dans des espaces humains-robots partagés. Pour améliorer cette situation, de nombreuses recherches sont actuellement menées. Elles visent à doter les robots de capacités perceptives et cognitives leur permettant de comprendre l'activité humaine, y compris par le contact physique. Avec de telles capacités, les robots pourront anticiper les gestes et mouvements des personnes qu'ils côtoient, afin d'adapter leur comportement à l'activité humaine.

Le partage d'espaces de travail sécurisés entre les humains et les robots ouvre la possibilité de mettre en place des processus de coopération humain-robot (**Cobotique**¹ [49]) qui peuvent réellement soulager l'être humain dans la réalisation de tâches difficiles. La coopération humain-robot est un des challenges les plus difficiles de la robotique moderne et cela reste un domaine très ouvert (*c.f.*, deuxième image de la Figure 1). Cette thèse, centrée sur la reconnaissance des intentions et du mouvement de l'utilisateur, est une contribution à ce domaine.

La cobotique est une discipline par nature pluridisciplinaire relevant de la robotique, des sciences sociales, des neurosciences et de l'intelligence artificielle.

La robotique s'intéresse à la conception de robots de plus en plus précis, rapides et dont les structures mécaniques sont toujours plus sophistiquées et adaptées à l'environnement et aux humains. C'est ainsi que beaucoup de robots humanoïdes ont été créés, comme présenté dans la Figure 2. Nos expériences seront quant à elles effectuées sur le robot humanoïde iCub.

Les sciences sociales abordent la compréhension des mécanismes (comme la structure des robots ou leur comportement) qui permettent aux robots d'être acceptés et utilisés par les

1. Provient du mot anglais "cobot" qui est une contraction de "collaborative robot". Il s'agit du domaine de la robotique qui s'intéresse à la création de robots passifs qui assistent les humains, sans autonomie globale, en ce sens qu'ils restent dépendant des intentions, gestes et comportements de leurs utilisateurs.

humains (par exemple, à l'aide d'évaluations de type NARS²). Nous verrons notamment dans notre contribution [M.E] que les humains ont un besoin de contrôler les mouvements de leurs robots, ce qui a été l'un des enjeux de cette thèse.

Les neurosciences identifient les processus neurobiologiques permettant aux humains et aux animaux de construire une représentation correcte de l'environnement à partir de données multi-sensorielles, de planifier et d'exécuter des actions de manière autonome et adaptative et d'interagir entre eux. Ces capacités multi-sensorielles d'inspiration biologique ont guidé la deuxième étude de ce mémoire, étude dans laquelle le robot développe de telles capacités d'interaction multimodales.

Finalement, *l'intelligence artificielle (I.A.)* élabore des techniques permettant aux robots d'apprendre et, de ce fait, d'être adaptatifs et autonomes. Tout comme la robotique, l'I.A. s'inspire des neurosciences et des sciences sociales afin de permettre aux robots d'être les plus efficaces et adaptés aux humains et à leur besoins. C'est dans ce domaine que se situe principalement cette thèse, tout en étant inspirée par les sciences présentées précédemment.

Toutes ces sciences cherchent à rendre l'interaction physique et sociale entre les êtres humains et les robots efficace et naturelle. Pour ce faire, différents enjeux ressortent des études correspondant aux notions d'adaptation, d'anticipation, de prédiction, de communication, d'acceptabilité ou encore de contrôle des forces extérieures subies par le robot. Ces différentes notions seront abordées dans ce mémoire.

Cette thèse a été effectuée au sein de l'équipe Larsen³ (INRIA, Nancy). Cette équipe possède un robot humanoïde iCub [214], pesant 22kg et ayant la taille d'un enfant de 4 ans et demi (104cm). Nous nous concentrons donc sur le développement des capacités de ce robot.

2 Objectif à long terme

Lorsqu'on imagine un robot idéal, on l'imagine agir et interagir comme un humain, en s'adaptant aux mouvements des personnes qui l'entourent et en réalisant les tâches qui lui sont assignées ou qui sont bénéfiques aux utilisateurs. Bien que la robotique actuelle soit encore loin de créer de tels robots, beaucoup de recherches et de progrès ont lieu dans ce domaine, que ce soit le domaine du contrôle, de l'acceptabilité, ou encore du raisonnement (planification, géométrie, etc.). Les capacités d'autonomie robotique ne cessent de s'améliorer, avec notamment les robots de *Boston Dynamics*, qui sont capables de marcher, de courir, de sauter, de déplacer des objets, etc. Cependant, en ce qui concerne l'interaction et la collaboration humain-robot, les avancées scientifiques demeurent timides. C'est dans cette thématique que se situe cette thèse.

L'interaction humaine met en jeu un nombre important de capacités, telles que la prise en compte de l'intention de l'autre, la faculté à se faire comprendre et à s'adapter. Ces capacités étant complexes et nombreuses, cette thèse se focalise sur la compréhension et l'exécution de l'intention de l'autre. De plus, cette capacité est elle-même composée d'un ensemble de compétences humaines complexes. Par exemple, elle nécessite de comprendre différents types d'indices directs, qu'ils soient sonores, visuels ou gestuels ; ou indirects nécessitant alors des capacités de raisonnement, provenant par exemple de l'environnement, de la situation, du comportement de l'individu ou encore de son but à long terme. De surcroît, il existe différentes façons de répondre à l'intention de l'autre, que ce soit par l'exécution d'un mouvement et/ou d'une tâche, par le dialogue, ou encore par une exécution intellectuelle. Nous nous intéressons dans cette thèse à la capacité du

2. *Negative Attitudes towards Robots Scale*

3. *Lifelong Autonomy and interaction skills for Robots in a Sensing ENvironment.*

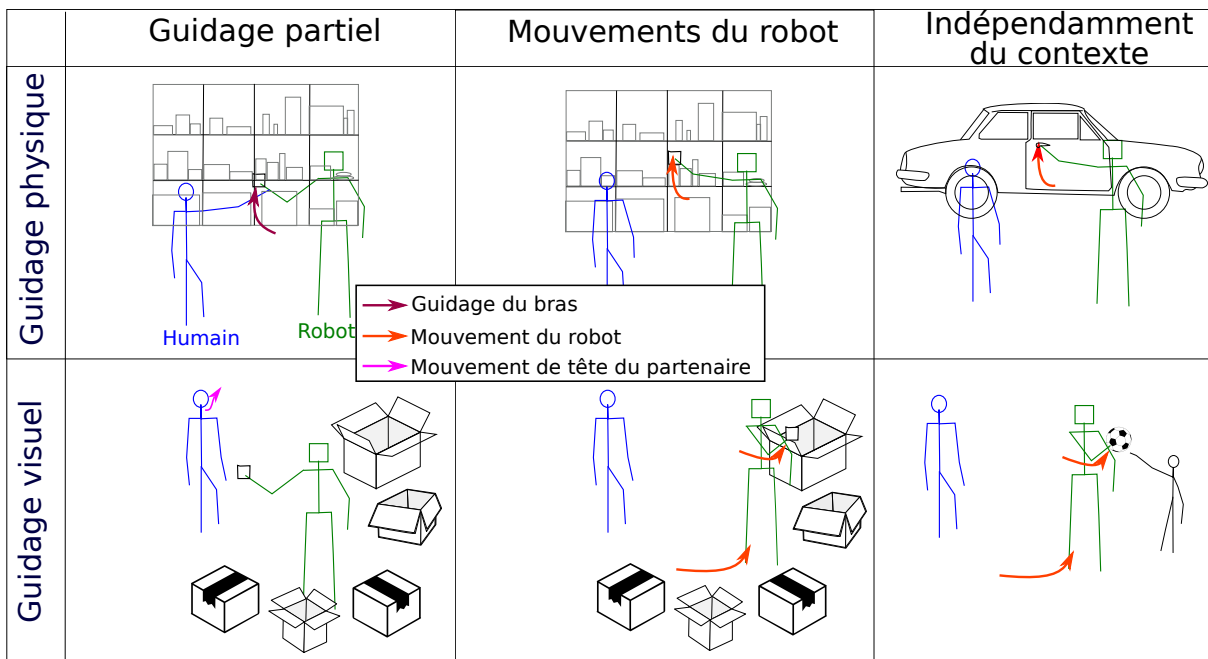


FIGURE 3 – Exemple de cas réel où l'adaptation du mouvement du robot en fonction de la volonté de l'utilisateur est nécessaire.

robot à comprendre et à exécuter les mouvements et tâches attendus par son partenaire par le biais d'indices provenant de différentes modalités.

Cette thèse se concentre alors sur la reconnaissance d'intention, effectuée à l'aide d'indices fournis par le partenaire lors d'interaction physique ou visuelle avec le robot, ou sans indices, par le mouvement naturel du corps-complet de l'utilisateur. Le but à long terme est alors de favoriser l'interaction humain-robot, en permettant au robot de mieux comprendre les intentions des humains avec lesquels il interagit. Cette thèse se focalise sur les signaux non-verbaux, sachant que nous sommes convaincus qu'une interaction naturelle ne peut être obtenue qu'en comprenant l'ensemble des signaux émis par le partenaire du robot. La Figure 3 présente le type de scénario visé par cette problématique de thèse. Il est intéressant de noter qu'avec de telles capacités, le robot pourra adapter ses connaissances aux différents contextes, à l'aide de sa compréhension de l'intention du partenaire.

3 Thématiques de recherche

Cette thèse est donc une contribution au développement de la robotique humanoïde et de l'apprentissage automatique, dans le domaine de la cobotique⁴. Elle se focalise sur les interactions non-verbales humain-robot, en particulier sur l'interaction gestuelle, à partir de différentes modalités. La prédiction de l'intention, la compréhension et la génération de gestes, sont donc les questions centrales de cette thèse.

4. Cobotique : néologisme provenant de "robotique" et "collaboration". Il s'agit donc de robotique collaborative humain-robot.

4 Problématiques

Les interactions humaines réussissent lorsque les différents acteurs participant à ces interactions sont capables de comprendre les indices que révèlent l'intention des autres. Le but de cette thèse est de formaliser cette capacité humaine afin d'améliorer les interactions humain-robot. Nous verrons comment un robot peut comprendre ces indices intentionnels, au travers d'un guidage partiel, guidage pouvant provenir de différentes modalités non-verbales.

Différents enjeux ressortent de cette problématique de thèse. Tout d'abord, le robot doit pouvoir **apprendre des gestes** que l'utilisateur souhaite qu'il exécute. Il s'agira d'**apprentissage par démonstrations kinesthésiques**, afin que l'utilisateur n'ait pas besoin d'écrire de programme informatique et qu'il puisse l'utiliser facilement. Puis, il doit pouvoir **généraliser ces gestes**, afin d'adapter ses mouvements en fonction de l'intention des personnes. De plus, il s'agira de permettre à l'utilisateur de transmettre son intention au robot à partir de **différentes modalités**. Finalement, il s'agira de permettre au robot d'**anticiper les mouvements corps complet**⁵ **des utilisateurs**, afin d'**adapter** ses propres mouvements. Les utilisateurs n'auront alors **plus besoin de s'occuper du robot**. Concernant ce dernier point, la difficulté sera alors de **prédire ces mouvements corps complet rapidement et de manière robuste**, malgré la grande dimension de tels mouvements.

4.1 Avantages

En permettant aux utilisateurs d'interagir avec le robot à l'aide de différentes modalités, l'interaction avec le robot est rendue plus agréable et intuitive pour l'utilisateur, et cela devrait diminuer sa charge cognitive.

De plus, ces choix permettent de respecter les critères principaux d'acceptabilité en robotique puisque le comportement du robot correspondra aux souhaits de l'utilisateur (*c.f.*, Section 1.4).

Finalement, de tels robots pourraient être plus facilement utilisés au quotidien, puisqu'ils peuvent **adapter leurs mouvements à différents contextes**, à l'aide des informations contenues dans le guidage partiel du robot.

5 Contribution

Afin de permettre aux robots d'acquérir ces capacités d'interaction, il est nécessaire de développer différents niveaux de cognition du robot, avec notamment des techniques de contrôle, d'adaptation et d'apprentissage.

En ce qui concerne le contrôle robotique, beaucoup d'études poussées relèvent déjà ce défi [202], en prenant notamment en compte l'interaction physique entre l'être humain et le robot. C'est pourquoi, nous intégrons dans nos études des contrôleurs déjà développés pour le robot *iCub*, basés sur les bibliothèques iKin [174] [173] (cinématique du robot) et iDyn [117] (dynamique du robot).

L'apport de cette thèse se situe à un niveau de développement cognitif intermédiaire, situé entre le contrôle automatique des mouvements du robots et les capacités de raisonnement permettant la planification de trajectoires. Notre thèse se consacre à la **prédiction de l'intention** du partenaire humain au travers de mouvements, et plus précisément par la prédiction de la poursuite d'un mouvement initié. De plus, nous nous sommes intéressés à **diminuer le coût physique et cognitif** que l'utilisateur doit fournir afin de contrôler le robot. Pour cela, nous avons permis

5. Il s'agit des mouvements de tous les membres de l'utilisateur humain.

à l'humain d'**interagir physiquement et/ou visuellement** avec le robot. Nous sommes allés jusqu'à permettre à l'humain de faire des **mouvements naturels** avec l'ensemble de son corps, **sans s'occuper du robot**, qui doit alors prédire le futur de ces mouvements.

Pour ce faire, différentes modalités perceptives ont été explorées. À l'aide de capteurs proprioceptifs, il interprète le mouvement de son propre bras, initié manuellement par l'utilisateur. À l'aide de capteurs visuels (caméra) et d'un logiciel calculant l'orientation de la tête de l'utilisateur, il interprète ses mouvements de tête. Finalement, à l'aide de capteurs externes portés par l'utilisateur, il interprète ses mouvements corps-complet.

Pour pouvoir répondre à ces enjeux, il est nécessaire d'utiliser une méthode qui permette au robot de reconnaître les différents types de mouvements. Pour cela, nous utilisons l'approche des primitives de mouvements. De plus, Le robot doit pouvoir **généraliser** ses connaissances aux différents contextes et prédire la poursuite de trajectoires initiées. C'est pourquoi, nous avons utilisé la méthode ProMP (*Probabilistic Movement Primitives*), qui modélise les primitives de mouvement sous forme de distribution. Nous avons par la suite adapté cette méthode afin de répondre à nos problématiques de thèse.

En ce qui concerne la prédiction de mouvement corps complet de l'utilisateur du robot, nous avons créé une méthode qui combine les ProMPs avec un auto-encodeur, afin de **compresser les trajectoires dans un espace réduit**. À l'aide de cette méthode, notre robot peut alors calculer en **temps réel mou**⁶ la prédiction de la trajectoire future de l'utilisateur, alors même qu'il s'agit de trajectoires corps-complet. De plus, cette méthode offre un avantage supplémentaire : celui de permettre au robot de généraliser d'avantage les mouvements appris. En effet, cette compression des données capture l'information présente dans toutes les démonstrations d'un même type de mouvement, ce qui revient à enregistrer les caractéristiques principales de ce type de mouvement, en s'extrayant des autres informations.

Ainsi, les enjeux principaux de cette thèse sont présentés dans le tableau 1 et nos contributions sont les suivantes :

Dans la première étude, présentée dans la Section 4 et dans le papier [D.P], le robot apprend à reconnaître et générer les mouvements permettant de réaliser les différentes tâches que son partenaire souhaite qu'il effectue. Pour cela, l'utilisateur apprend au robot à réaliser des gestes par démonstration. Dans cette étude, nous avons notamment montré qu'en apprenant en plus du mouvement cartésien, le mouvement des forces et couples exercés sur le bras robotique, cela a l'avantage de permettre une meilleure inférence de la durée totale de la trajectoire ainsi que de fournir au robot une capacité de prédire les forces d'interaction qu'il est supposé recevoir. Cependant, nous verrons que si le robot utilise ce mouvement de force pour prédire le mouvement cartésien, alors la qualité de la prédiction cartésienne diminue. Ainsi, nous montrerons que pour avoir les meilleurs résultats, la prédiction du mouvement et l'inférence de sa durée doivent être effectuées à l'aide de l'information des forces, mais que cette information ne doit plus être utilisée lorsque le robot génère le mouvement correspondant au mouvement initié par l'utilisateur. De plus, nous fournissons dans cette étude un logiciel avec tutoriel et code source du logiciel, afin de permettre la réutilisation de notre application et l'avancée de nos travaux de recherches. L'utilisateur pourra notamment utiliser une sous-partie de ce logiciel afin d'apprendre n'importe quel type de trajectoires, pour n'importe quel type d'application [soft.Pred].

Afin de diminuer le coût moteur de l'utilisateur et d'améliorer son interaction avec le robot, une seconde étude, présentée dans la Section 5 et dans le papier [D.M], permet au partenaire de transmettre son intention à l'aide d'un guidage physique et/ou d'un guidage visuel, par le

6. Les systèmes temps réel mous ou souples autorisent exceptionnellement le dépassement des contraintes temporelles, celui-ci pourra être compensé à court terme.

Étude	Robot	Humain	Coût humain
Études 1,2,3 : Apprentissage de gestes lors de pHRI ⁷	Il reproduit, génère et généralise des mouvements à partir des gestes appris. Il peut notamment adapter ces mouvements à des contraintes (<i>par ex.</i> , obstacles) ou à différents contextes.	Professeur durant la phase d'apprentissage, il prend le robot par la main pour lui faire apprendre différents gestes et il effectue ces démonstrations plusieurs fois pour un même geste.	mental et physique
Étude 1 : Prédiction du mouvement lors de pHRI	Il comprend le geste initié physiquement et finit le mouvement par lui-même, quelque soit la situation donnée.	Il initie le geste en guidant physiquement la main du robot : c'est lui qui adapte le guidage à la situation donnée.	mental et physique
Étude 2 : Prédiction du mouvement à l'aide de plusieurs multi-modalités	Idem, mais il prédit la fin du mouvement à partir de la (des) modalité(s) choisie(s) par l'humain.	Idem, mais à l'aide de guidage physique et ou visuel (à l'aide d'un mouvement de tête), selon la situation donnée et sa volonté.	mental
Étude 3 : Prédiction du mouvement de l'humain, sans indices de sa part	Il prédit les mouvements du corps entier de l'utilisateur, à l'aide de capteurs que celui-ci porte.	Il ne s'occupe plus du robot.	aucun

TABLEAU 1 – Enjeux généraux de la thèse.

biais de mouvements de la tête. Ainsi, l'utilisateur peut choisir comment il souhaite transmettre son intention au robot, ce qui lui évite notamment de toucher le robot si cela le dérange ou de donner rapidement des informations au robot (mouvement de tête) afin de lui demander d'effectuer sa tâche. Nous verrons dans cette étude qu'en guidant le robot visuellement à l'aide de mouvements de tête, la qualité de l'inférence du type de mouvement désiré augmente. De plus, il s'agit d'un guidage rapide, sans coût physique pour l'utilisateur et nous verrons qu'il permet de désambiguïser différents types de trajectoires qui commencent de la même manière. Nous y verrons en contrepartie que lorsque le type de mouvement est reconnu, le guidage physique permet une meilleure prédiction de la continuité du mouvement. Par la suite, nous analyserons comment le robot peut combiner au mieux ces informations, afin de prédire au mieux la trajectoire désirée future.

Finalement, dans la dernière étude présentée dans la Section 6 et dans le papier [D.W], à l'aide d'une méthode que nous avons créée, notre robot est capable de comprendre l'intention de son partenaire et de prédire ses mouvements corps complet futurs, sans que celui-ci n'ait besoin de donner volontairement d'indices au robot. Pour cela, dans notre application l'utilisateur porte une combinaison (*c.f.*, Xsens MVN 2.4) composée de capteurs mesurant les positions

articulaires de celui qui la porte. Cette technique peut aussi être appliquée à des squelettes calculés à partir d'images de caméras. La difficulté de cette dernière étude est que, puisqu'il s'agit ici de mouvements corps complet, beaucoup d'informations doivent être traitées. Afin d'éviter que cette quantité de données ralentisse la prédiction du mouvement futur, la méthode développée permet de compresser les données et de prédire la poursuite du mouvement. Nous comparerons notamment différentes méthodes permettant de compresser les données, que nous combinons avec les ProMPs. Les codes sources de cette étude sont disponibles dans [soft.WB.Pred].

Deux contributions secondaires ont aussi été réalisées lors de cette thèse. Tout d'abord, dans l'étude [M.E], avec Marichal *et al.* nous avons étudié l'acceptabilité d'utilisateurs non-experts lors de l'utilisation d'un bras robotique. Cette étude nous a notamment fait comprendre à quel point il est important pour les utilisateurs de conserver le contrôle sur les robots. Puis, pendant cette thèse, j'ai réalisé un programme permettant de manipuler le robot à partir du dispositif haptique *Geomagic Touch*, présenté dans la Section 2.5. Ce programme a été réalisé afin d'effectuer des simulations, afin de pouvoir guider manuellement le bras du robot simulé. Ce programme a notamment été utilisé dans une étude internationale à laquelle j'ai contribué [CodyCo], présentée dans la Section 1.3.2.

Les différentes problématiques soulevées par cette thèse et les résultats obtenus ont notamment été utilisés dans les projets européens CoDyCo et AnDy.

6 Projets européens CoDyCo et AnDy

6.1 CoDyCo

Le but du projet CoDyCo⁸ est d'améliorer les méthodes qui concernent les mouvements de robots humanoïdes, en présence de plusieurs contacts. Ces méthodes prennent en compte l'environnement constitué notamment d'humains. Pour améliorer les approches traditionnelles, ce projet se concentre sur des méthodes permettant d'effectuer des tâches d'interaction de manière coordonnée qui combinent la planification de ces mouvements et la compliance⁹ des membres du robot (c'est-à-dire, adaptation de la rigidité de ses articulations), afin de contrôler les événements et les contacts prédictibles ou non, ces avancées théoriques devant être validées sur des scénarios d'interaction réels.



Dans ce projet, le premier objectif concerne la mobilité du robot, ce qui n'a pas été abordé dans ce mémoire. Le second objectif nous concerne davantage. Il s'agit de maîtriser les aspects perceptuel et moteur de l'interaction et d'exploiter les forces de contacts. Au début de ce projet, les avancées dans la compliance des mouvements et le sens du toucher ont permis aux robots d'interagir physiquement de manière sécurisée et robuste avec les contacts inattendus, provenant notamment d'humains. Dans ce projet, le but est d'aller plus loin, en ne se contentant pas de prendre en compte les contacts imprévisibles, mais aussi en exploitant les contacts prédictibles pour qu'ils participent à l'achèvement de l'objectif du robot.

Finalement, le dernier objectif consiste à valider des scénarios dans le monde réel avec le robot *iCub* effectuant des tâches utilisant son corps en entier. Les évaluations des différentes expériences de ce projet permettent de montrer que le robot *iCub* peut à la fois exploiter les forces rigides

8. *whole-body Compliant Dynamical COntacts in cognitive humanoids.*

9. Il s'agit de la capacité d'un manipulateur à avoir un comportement souple, à s'adapter à son environnement[231].

qu'il subit en contact avec son corps en tant que support et compenser les forces indésirables qu'il subit.

6.2 AnDy

Le but du projet AnDy¹⁰ [183] est d'améliorer la capacité des robots à coopérer avec les êtres humains dans des environnements industriels et domestiques. Pour cela, les robots doivent interpréter les demandes provenant d'interactions physiques afin de réduire les risques de blessures. Ce projet concerne des robots "collaboratifs" qui facilitent le travail des humains. En effet, ceux-ci soutiennent les actions humaines complexes et assistent les gestes techniques ou répétitifs afin que ces gestes soient faits de manière plus ergonomique, réduisant ainsi le risque d'accidents musculo-squelettiques.



Ces robots seraient par exemple capables d'aider les ouvriers à monter des pièces lourdes dans des industries automobiles, de déplacer des meubles, ou encore de soutenir des personnes âgées lors de leurs déplacements. Pour cela, nous utilisons une combinaison permettant de mesurer des informations sur les déplacements et les forces exercées par l'utilisateur qui la porte (*c.f.*, Section 2.4). À partir de ces mesures, l'idée est de développer un modèle des mouvements ergonomiques que devrait effectuer l'utilisateur et qui prédira les mouvements dynamiques que l'utilisateur envisage de faire. Basé sur ce modèle, il s'agira finalement de créer un contrôle physique prédictif, permettant à des robots d'aider l'utilisateur afin que celui-ci effectue des mouvements ergonomiques. Pour cela, trois types de robots seront utilisés : les "cobots", des robots industriels homologués pour travailler en collaboration étroite avec des êtres humains; les exosquelettes, capables de soutenir et de guider le corps des personnes dans leur travail; et les robots "compagnons", des humanoïdes avancés qui anticiperont les actions de leur utilisateur humain, afin de les aider dans leurs tâches. Cette dernière étape nous concerne particulièrement puisque cette thèse consiste à permettre au robot humanoïde *iCub*, d'agir de manière autonome et adaptative, en s'adaptant à l'intention de son partenaire.

7 Organisation de la thèse

Ce mémoire est organisé en sept chapitres. Les quatre premiers fournissent le contexte dans lequel se sont réalisées les différentes études, avec un état de l'art permettant de situer scientifiquement cette thèse, puis les matériels et les méthodes utilisées. Finalement, les trois derniers chapitres détaillent les expériences réalisées et les différents résultats obtenus.

Dans le Chapitre 1, nous avons présenté les différents éléments de l'état de l'art selon différents points de vue (sciences cognitives, robotique, I.A., etc.) afin de permettre au lecteur de situer et de contextualiser cette thèse au sein des connaissances scientifiques actuelles. Cependant, sachant que nous ne sommes pas expert dans tous ces domaines, nous ne fournissons pas un état de l'art complet dans chacun de ces domaines et nous mettons l'accent sur les domaines de l'intelligence artificielle et de la robotique, en lien avec les problématiques et les différentes méthodes utilisées dans cette thèse.

Ce chapitre commence par l'explication des notions abordées, telles que l'*intention* (Section 1.1), l'*anticipation* et la *prédiction* (Section 1.2). Ces explications permettent notamment de mettre en évidence certains mécanismes pris en compte lors de la conception de nos applications.

10. *Advancing anticipatory behaviors in dyadic human-robot collaboration.*

Puis, la Section 1.3 fournit un récapitulatif concernant les études et les différentes techniques existantes permettant l'*interaction physique* entre humains et robots, afin d'introduire et positionner la première étude de ce mémoire dans cet ensemble de connaissances.

Nous soulevons alors, dans la Section 1.4, une des limites de telles interactions physiques, en présentant différentes études sur l'*acceptabilité*, comprenant également le paradigme maître/esclave (Section 1.4.1). En effet, cette section met en avant que l'interaction physique et le manque de contrôle sur les robots agissant de manière autonome dérangent certaines personnes. Cela permet de mettre en avant l'un des avantages de notre seconde étude, qui consiste à permettre à des utilisateurs de guider leur robot sans interagir physiquement avec celui-ci, en le guidant visuellement, à l'aide de mouvements de tête. Cette idée s'inspire notamment de différentes études concernant la *communication par le regard*, études qui sont présentées dans la Section 1.5.

Le but principal de cette seconde étude ne se limite pas au besoin de guidage visuel. Celle-ci s'inspire des études présentées dans la Section 1.6 sur les **perceptions multimodales**, qui montrent que les êtres humains et les robots utilisant plusieurs modalités améliorent leurs perceptions de l'environnement. Le but est alors de permettre au robot d'améliorer sa capacité à comprendre les mouvements de l'utilisateur et à prédire leur poursuite.

Jusqu'à maintenant, ces différentes sections présentent les notions, motivations et problématiques de cette thèse. Les prochaines sections quant à elles présentent les différentes techniques de robotique et d'intelligence artificielle permettant de résoudre ces problématiques.

Tout d'abord, nous présentons dans la Section 1.7 les principales méthodes qui correspondent à l'approche des **primitives de mouvements**, qui permettent aux robots d'apprendre différents types de mouvements. Nous expliquons notamment notre choix de méthode de modélisation des mouvements du robot, à l'aide d'une comparaison des techniques les plus adaptées pour nos applications. En effet, nous montrons que la méthode de **Primitives de Mouvements Probabilistes** (ProMP) a l'avantage de contenir la variabilité des trajectoires de démonstration, trajectoires qui permettent au robot d'apprendre ces primitives de mouvement. Ainsi, plutôt que d'apprendre une simple trajectoire représentative d'un type de mouvement, le robot apprend une distribution de ces trajectoires. Cela lui permet notamment de générer différents mouvements ; de reconnaître si un mouvement initié par l'utilisateur appartient à cette distribution ; ainsi que de prédire la poursuite d'un mouvement initié.

Finalement, l'une des problématiques consistant à adapter notre application à des trajectoires représentées par de nombreuses données, cela nécessite de simplifier les calculs afin d'éviter qu'ils ne durent trop longtemps. Pour cela, la Section 1.8 présente un état de l'art sur les différentes familles de méthodes permettant la compression-décompression de données. De plus, nous présentons dans cette Section d'où proviennent les méthodes AE-ProMPs et VTSFE-ProMPs, créées dans l'étude 6, en présentant l'origine des deux sous-méthodes qui les composent, nommées AE et VTSFE.

Le Chapitre 2 présente les matériels utilisés, avec une description de notre robot iCub ; une description des logiciels et des appareils utilisés permettant d'effectuer les tests en simulation ; ou encore une description de la combinaison utilisée afin de récupérer les mouvements du corps entier des utilisateurs.

Le Chapitre 3 détaille les méthodes utilisées. Premièrement, la méthode ProMPs est présentée dans la Section 3.3, avec les différentes améliorations que nous lui avons apportées. Cette méthode permet donc de modéliser un ensemble de trajectoires qu'effectue le robot ou l'utilisateur, à l'aide d'une distribution de trajectoires. Elle est dans un premier temps décrite dans une forme simplifiée (où toutes les trajectoires sont supposées avoir la même durée et appartenir à la même

primitive) puis, progressivement, elle est détaillée en ajoutant de la complexité. Cette méthode sera par la suite adaptée dans les différentes études.

Deuxièmement, les méthodes AE-ProMPs et VTSFE-ProMPs sont présentées dans la Section 3.4. Celles-ci permettent en premier lieu de compresser les trajectoires de l'utilisateur dans un espace latent réduit à l'aide de différents types d'auto-encodeur, puis de modéliser ces trajectoires compressées en primitives de mouvement à l'aide de la méthode ProMPs. Pour expliquer ces méthodes, nous présentons d'abord le fonctionnement d'un auto-encodeur classique, puis la méthode VTSFE qui permet de compresser des données provenant de trajectoires de manière dynamiquement plus cohérente. Nous présentons finalement les méthodes combinant les ProMPs avec ces types d'auto-encodeurs.

Les trois derniers chapitres présentent les trois études dans lesquelles le robot prédit l'intention de son partenaire et agit en conséquence. Les contributions de ces études étant précisées en amont (*c.f.*, Section 5), nous nous contentons de rappeler ici les thématiques de ces études.

La première étude, présentée dans le Chapitre 4, permet au robot d'interpréter l'interaction physique initiée par son partenaire avec lui-même, afin d'effectuer un mouvement précis et une tâche correspondante.

La seconde étude, présentée dans le Chapitre 5, ajoute à l'étude précédente la capacité du robot à utiliser l'information visuelle afin de reconnaître la trajectoire de son partenaire. Ainsi, l'utilisateur a maintenant le choix entre interagir physiquement avec le robot et/ou en effectuant simplement des mouvements de tête, afin d'indiquer au robot le mouvement et la tâche à effectuer. De plus, cette étude permet d'évaluer si en combinant ces deux perceptions, le robot est capable d'améliorer ses prédictions.

La dernière étude, présentée dans le Chapitre 6, permet de prédire la poursuite du mouvement du corps entier de l'utilisateur du robot, ce qui requiert l'utilisation d'une méthode de compression de données afin d'effectuer des calculs en temps réel mou.

Finalement, le Chapitre 7 permet de conclure de mémoire. Pour cela, la Section 7.1 rappelle les principaux résultats obtenus ainsi que leurs limitations. Cela permet de faire ressortir le fil directeur de cette thèse et ses avantages, puisque chaque étude contribue à répondre aux problématiques des études précédentes. Puis, la Section 7.2 présente les perspectives, en présentant les prochaines études que nous aimerions réaliser ainsi qu'en situant nos contributions dans un cadre plus général, au sein des objectifs à long terme qui nous avons présentés dans la Section 2.

Bibliographie

Papiers scientifiques

- [D.W] Oriane Dermý, Maxime Chaveróche, Francis Colas, François Charpillet, and Serena Ivaldi. Prediction of human’s whole-body movements. In *Humanoids*, 2018.
- [D.M] Oriane Dermý, François Charpillet, and Serena Ivaldi. Multi-modal intention prediction with probabilistic movement primitives. In *Human-Friendly Robotics*, 2017.
- [D.P] Oriane Dermý, Alexandros Paraschos, Marco Ewerton, Jan Peters, François Charpillet, and Serena Ivaldi. Prediction of intention during interaction with icub with probabilistic movement primitives. In *Front. Robotics and AI*, 2017.
- [M.E] Sebastian Marichal, Adrien Malaisé, Valerio Modugno, Oriane Dermý, François Charpillet, and Serena Ivaldi. One-shot Evaluation of the Control Interface of a Robotic Arm by Non-Experts. In *International Conference on Social Robotics*, Kansas City, United States, 2016.
- [CodyCo] Francesco Romano, Gabriele Nava, Morteza Azad, Jernej Čamernik, Stefano Dafarra, Oriane Dermý, Claudia Latella, Maria Lazzaroni, Ryan Lober, Marta Lorenzini, et al. The codyco project achievements and beyond : Toward human aware whole-body controllers for physical human robot interaction. In *IEEE Robotics and Automation Letters*, 3(1) :516–523, 2018.

8 Présentations diverses

- [D.JNRH] Oriane Dermý, Francois Charpillet, and Serena Ivaldi. Prediction of Intention : from gesture to whole-Body movement. *Présentation lors des Journées Nationales de la Robotique Humanoïde*, France, 2018.
- [D.MLSS] Oriane Dermý. Robots that learn to control the physical interaction with humans. *Poster présenté lors de Machine Learning Summer School*, Espagne, 2016.

Logiciels créés

- [soft.Pred] Oriane Dermý. Logiciel de modélisation, génération, inférence et prédiction de mouvements, à l’aide de ProMPs. <https://github.com/inria-larsen/icubLearningTrajectories/tree/master/MatlabProgram>, 2017.
- [soft.Pred.Mult] Oriane Dermý. Logiciel utilisant des ProMPs multimodales afin de permettre au robot iCub d’ exécuter des gestes. <https://github.com/inria-larsen/icubLearningTrajectories> , 2017.

[soft.ProMP.Mult] Oriane Dermy. Logiciel de modélisation, génération, inférence et prédiction de mouvements multimodaux. <https://github.com/misaki43/Multimodal-prediction-of-intention-with-ProMPs>, 2017.

[soft.WB.Pred] Oriane Dermy. Codes sources permettant la prédiction de mouvements à grande dimension, à l'aide de ProMPs. <https://github.com/misaki43/AE-ProMPS7>, 2018.

Vidéos

Interactions humaines (inspiration)

[HHI] Oriane Dermy. Vidéo représentant deux personnes en train d'interagir non verbalement avec des gestes de tête ou par interaction physique. <https://drive.google.com/file/d/0B9sXstBzN0iuNktQLTVfN25GMjQ/view?usp=sharing>, 2017.

[WB-HHI] Oriane Dermy and Isabelle Legrand. Vidéo d'interaction entre humains avec prédiction de l'intention à partir d'un mouvement de corps complet. <https://youtu.be/hzTsNNshn00>, 2018.

Interactions humain-robot

[PredReel] Oriane Dermy. Vidéo où l'iCub prédit la fin d'un mouvement à l'aide de mesures cinématiques. <https://www.youtube.com/watch?v=bLiZSZBMEDA&feature=youtu.be>, 2017.

[PredVisu] Oriane Dermy. Vidéo où iCub prédit la fin d'un mouvement à l'aide de mesures visuelles https://www.youtube.com/watch?v=0DnK93_5Lsk&t=0s&index=10&list=UUsNHuacpA3YnSbI6cbjBa4A, 2017.

[MultiDim] Oriane Dermy. Vidéo représentant la prédiction de mouvements corps entier à l'aide de ProMPs. <https://youtu.be/07PYTWN6egs>, 2018.

[VTSFE-ProMPs] Oriane Dermy. Vidéo représentant la prédiction de mouvements corps entier à l'aide de la méthode VTSFE-ProMPs. https://www.youtube.com/watch?v=fShju_8cVEs&list=UUsNHuacpA3YnSbI6cbjBa4A&index=3, 2018.

[AE-ProMPs] Oriane Dermy. Vidéo représentant la prédiction de mouvements corps entier à l'aide de la méthode AE-ProMPs. <https://youtu.be/xqH-Tt7quxg>, 2018.

[humanoids] Oriane Dermy and Isabelle Legrand. Vidéo des résultats des méthodes de prédiction corps complet présentées à *Humanoids* <https://youtu.be/hzTsNNshn00>, 2018.

Tutoriels

[Tutoriel] Oriane Dermy. Vidéo tutoriel d'utilisation du *Geomagic Touch* (dispositif haptique). <https://www.youtube.com/watch?v=ZydXf6wrXgY&feature=youtu.be>, 2017.

1

État de l'art

Afin de permettre aux robots de développer des compétences toujours plus complexes et avec davantage d'autonomie, il est inspirant de comprendre le fonctionnement du cerveau animal et plus précisément du cerveau de l'humain. En effet, l'environnement du robot étant variable, les méthodes de planification et de contrôle basées sur des modélisations physiques et géométriques de l'environnement et du robot deviennent imprécises et souvent inadaptées lors d'interactions avec les humains. Il faut alors permettre au robot d'apprendre comment ses comportements sont impactés par l'environnement et comment ses actions doivent alors s'adapter. Or, il s'agit typiquement de capacités humaines. C'est pourquoi, dans cette présentation de l'état de l'art, un effort a été fourni afin de résumer en amont quelques connaissances dans les domaines des sciences humaines et neurosciences liées aux problématiques de cette thèse. Ces connaissances ont plus ou moins inspiré les avancées en robotique afin de permettre aux robots d'être toujours plus autonomes et intelligents. Certaines de ces techniques modélisent directement le fonctionnement du cerveau humain à l'aide de réseau de neurones artificiels, d'autres ont une forme hybride avec une modélisation physique de l'environnement, utilisée par le robot comme un a priori qu'il adapte à la suite par apprentissage.

Dans le cadre de cette thèse, il s'agit de développer des méthodes utilisables en pratique avec le robot *iCub*, afin de lui permettre d'agir selon l'intention de son partenaire. Pour cela, différentes approches sont possibles, que ce soit par l'utilisation de réseaux de neurones ou différents types de modélisations physiques. Dans cette thèse, nous utilisons une approche qui permet au robot de modéliser ses mouvements en ayant à la fois très peu de contraintes physiques tout en étant plus fiable que par modélisation par réseaux de neurones.

Sommaire

1.1 Intention et intelligibilité	16
1.1.1 Théories sur l'intention	17
1.1.2 L'intentionnalité en sciences humaines	19
1.1.3 L'intentionnalité en robotique	20
1.2 Prédiction Inférence et Anticipation	21
1.2.1 Définitions	22
1.2.2 L'anticipation en robotique	22
1.2.3 La prédiction en robotique	23
1.3 Interaction physique	24
1.3.1 Techniques robotiques permettant l'interaction physique entre l'humain et le robot	24
1.3.2 Techniques d'interaction physique avec le robot iCub	25
1.4 Acceptabilité du point de vue de l'utilisateur et rôle du robot . . .	26
1.4.1 Rôle du robot et type d'autonomie	27
1.5 Comprendre l'intention par le regard	28
1.5.1 L'attention conjointe chez l'humain	28
1.5.2 La direction du regard en robotique	29
1.6 Perceptions multimodales	30
1.6.1 Multimodalité chez l'humain	31
1.6.2 Multimodalité en robotique	31
1.7 Primitives de Mouvements	33
1.7.1 Modélisation	33
1.7.2 Justification du choix de modélisation	40
1.7.3 Durée des Mouvements	43
1.7.4 Logiciels Open Source Connexes	45
1.8 Compression dans un espace latent	45
1.8.1 Données indépendantes du temps	45
1.8.2 Données avec dépendance temporelle	48

1.1 Intention et intelligibilité

Lors de la collaboration entre un humain et un robot, la compréhension mutuelle de ces deux agents est primordiale pour assurer la réussite de leur tâche commune. La compréhension mutuelle signifie que chaque agent comprend l'action en cours de l'autre (c'est-à-dire, son statut, son but et toutes les informations qu'il peut estimer ou prédire). Nous verrons dans la Section ci-dessous que la reconnaissance de l'intention n'est qu'une des facettes de ce problème, mais qu'elle joue un rôle crucial dans l'acquisition de compétences anticipatives.

C'est pourquoi la Section suivante explicitera ce qu'est l'intention et montrera son importance dans la théorie de l'action. Pour cela, nous présenterons d'abord les théories classiques de l'action qui ne prennent pas en compte cette notion d'intention. Puis, nous rappellerons les limites de telles théories et nous finirons par présenter les théories plus complètes, qui prennent en compte l'intention. Par la suite, nous verrons comment se développent les capacités de compréhension de l'intention et des comportements associés chez les humains, afin de comprendre comment cela a pu inspirer la recherche en robotique. Finalement, nous présenterons les recherches en robotique qui se basent sur la compréhension de l'intention de l'utilisateur.

De ces connaissances, nous posons la problématique de cette thèse, qui consiste à permettre au robot d'estimer l'intention de l'utilisateur, afin de prédire le mouvement et l'action qu'il doit effectuer et donc de respecter les attentes de l'utilisateur.

1.1.1 Théories sur l'intention

Une intention peut-être définie, de manière simplifiée, comme :

« L'envie ou la résolution prise par une personne d'agir et d'atteindre un ou des objectifs fixés à l'avance, sa décision de passer à l'action étant quasiment définitive. »
[Internaute]

Cette étape intéresse de nombreux domaines, que ce soit pour comprendre les mécanismes du cerveau qui permettent l'action (*par ex.*, modèles neuronaux, théorie de l'action) ; pour comprendre comment l'humain interagit avec l'environnement (sciences sociales) ; pour comprendre certains de ses troubles ; ou encore, dans notre cas, pour créer des applications interactives.

Bratman théorise dans son livre [25] la planification de l'intention. Selon lui, les intentions sont nécessaires à l'élaboration de plans d'action, qui nous permettent de nous organiser et agir au quotidien. Il démontre dans ce livre le lien entre l'intention et l'action. Nous allons voir maintenant que la théorie classique de l'action a été améliorée par la prise en compte de l'intentionnalité des individus.

Dans [167], Pacherie *et al.* reprennent les différentes théories existantes de l'action. Ces théories ne sont pas directement le sujet de cette thèse, mais parce qu'elles permettent de comprendre ce qu'est l'intentionnalité, nous les vulgarisons ici, en nous basant sur ce document. D'après la théorie causale classique de l'action, un agent accomplit une action si et seulement si un certain état psychologique de l'agent cause cette action. Les premières versions de cette théorie [56, 95] considéraient cet état psychologique comme un complexe de croyances et de désirs, sans qu'intervienne la notion d'intention. Ce complexe était composé d'un but ou désir D , d'une croyance d'orientation S (c'est-à-dire, situation dans laquelle croit se trouver l'agent) et d'une croyance instrumentale, qui correspond à la croyance qu'une action A dans la situation S amène au but voulu D)¹¹. Mais cette théorie a soulevé beaucoup de difficultés présentées dans [167]. Par exemple, ce modèle ne prend pas en compte la distinction entre *avoir l'intention de faire A* et simplement *désirer faire A*. Dans son livre [25], Bratman explique cette distinction par

Les intentions sont des pro-attitudes [(c'est-à-dire, attitude positive, désir)] qui contrôlent notre conduite, ce ne sont pas les désirs ordinaires. Les désirs ordinaires ont simplement une influence potentielle sur l'action.

Il explique alors que les comportements rationnels ne peuvent pas être analysés en considérant uniquement les croyances et les désirs d'un agent, mais nécessite aussi la considération de l'intention de cet agent. Il précise alors que l'agent commence par peser ses désirs et croyances, puis, à un moment, il sélectionne l'un de ces désirs et agit intentionnellement pour effectuer les actions nécessaires à son accomplissement.¹² Beaucoup d'autres chercheurs partent de cette même analyse, cependant, leurs analyses sur les fonctions de l'intention varient. Pour Bratman, les intentions peuvent être caractérisées par trois fonctions :

11. En informatique, on peut considérer que les différents modèles qui permettent aux robots d'agir (boucles de contrôle, réseaux neuronaux, HMMs, etc.) peuvent modéliser cette description de l'action. En effet, les capteurs du robot estiment la situation S ; les robots ont un but D ; et ils font une estimation des actions à effectuer A (*c.f.*, 1.7).

12. Pour comparer à nouveau avec les algorithmes informatiques, cela correspond à l'étape de sélection de l'action, par exemple lorsque la fonction d'évaluation d'une action (c'est-à-dire, une fonction qui estime si l'action permet d'atteindre le but de manière pertinente) dépasse un certain seuil.

- Elles mettent un terme au raisonnement pratique dans lequel les différentes croyances et désirs ont été évalués, afin de permettre l'action ;
- Elles déclenchent un autre raisonnement pratique qui consiste à planifier comment accomplir l'action choisie ;
- Elles coordonnent les activités de l'agent dans le temps avec les activités des autres agents.

Notons que ces trois fonctions correspondent à une période de formation initiale de l'intention et au déclenchement de l'action. Pour Brand et Mele [24, 151], les intentions peuvent être caractérisées par quatre fonctions :

- Elles déclenchent l'action ;
- Elles soutiennent l'action jusqu'à son achèvement ;
- Elles guident dans la production de l'action ;
- Elles ont un rôle de suivi et de contrôle de l'action.

Cette fois-ci, ces fonctionnalités correspondent à une période de déclenchement de l'action jusqu'à son achèvement. Selon les caractérisations fonctionnelles et temporelles de l'intention faites par ces chercheurs, ceux-ci distinguent différents types d'intention, dont les définitions sont similaires. Ainsi, Bratman différencie l'intention dirigée vers le futur et l'intention dirigée vers le présent. Brand différencie les intentions prospectives (planification de l'action en avance) et immédiates. Mele différencie les intentions distales (c'est-à-dire, éloignées) et proximales (c'est-à-dire, proches). Les intentions distales mettent fin au raisonnement pratique sur les fins (c'est-à-dire, les buts) et enclenchent le raisonnement sur les moyens et les plans, avec une représentation de haut niveau des buts et plans. Les intentions proximales spécifient le plan d'action en l'ancrant dans la situation présente et elles sélectionnent les programmes moteurs appropriés. Finalement, Searle [197] distingue intentions préalables (c'est-à-dire, "je vais faire [...]") et intentions dans l'action (c'est-à-dire, "je fais [...]"). Dans [167], Pacherie caractérise trois types d'intentions selon leur contenu et leur dynamique. Il s'agit du modèle DPM (intention Distale, puis Proximale, puis Motrice). Elle différencie alors en plus des intentions dirigées vers le futur (F-intention) et vers le présent (P-intention), les intentions motrices (M-intention). Les P-intentions ont pour elle quatre fonctions :

- Elles déclenchent l'action ;
- Elles sont un soutien motivationnel de l'action en cours ;
- Elles guident et contrôlent à haut niveau¹³ la production de l'action.

Pour Mele, il y aurait alors une dynamique où les F-intentions se transformeraient en une intention d'agir maintenant, débouchant sur les P-intentions, puis ces intentions déclencheraient l'action jusqu'en son achèvement. Les M-intentions seraient alors un second niveau de guidage et de contrôle de l'action qui permettent d'assurer une transformation du sensoriel au moteur (contrôle non plus perceptif, mais sensorimoteur qui met en jeu les représentations motrices). Elle décrit alors la dynamique en amont des M-intentions comme la sélection de l'un de schémas moteurs envisagés ; et la dynamique en aval, comme les fonctions de guidage et de contrôle des M-intentions. Notons que ces fonctionnalités se reposent sur les modèles inverses qui permettent de déterminer le mouvement à accomplir pour atteindre le but (guidage) et sur les modèles prédictifs qui permettent d'anticiper et de compenser les effets sensoriels du mouvement (contrôle). En se basant sur ces théories de l'intention, on considère qu'un agent peut prédire l'action de son partenaire à partir de la compréhension de son intention, puisque l'intention d'une action se forme avant l'action elle-même et qu'elle guide et contrôle cette action durant toute sa durée.

13. Perceptif seulement.

1.1.2 L'intentionnalité en sciences humaines

L'intentionnalité est étudiée dans les sciences humaines, afin de comprendre les interactions entre les individus [154, 26].

En ce qui concerne la capacité à collaborer, les humains ont besoin d'anticiper et de s'adapter aux mouvements de leurs coéquipiers [198]. Cela nécessite la compréhension de l'intention de l'autre afin de prédire son action future. Cette capacité de compréhension repose sur le fait que l'intention d'action est définie avant le début des mouvements et que cette intention est déjà reflétée dans le premier acte moteur. Cette capacité se développe très vite chez l'humain, puisque dès 18 mois, le bébé peut reproduire correctement une action qui a été échouée par un adulte [152].

Dans [91], Gregerly *et al.* font l'hypothèse, au vu de leur résultat, que l'enfant de douze mois est capable d'attribuer des intentions causales à la personne interagissant avec lui. À la fin de cet article, ils proposent cependant une seconde interprétation, où l'enfant se représente l'intention de l'action sans pour autant se modéliser la représentation mentale de l'intention de l'agent. Ces deux hypothèses représentent les différentes théories existantes, où certains interprètent que les bébés comprennent l'intention de l'agent [98] et d'autres le but du mouvement seulement (théorie téléologique [90]). Une explication détaillée de ces différentes approches est présentée dans [96]. La *théorie téléologique* (qui consiste à penser que le bébé se représente la finalité du mouvement, sans interpréter d'intention à son partenaire) serait cependant une étape nécessaire pour que l'enfant développe sa compréhension de l'intention. Dans notre recherche, nous considérons l'approche téléologique, c'est-à-dire, que le robot doit comprendre la finalité du mouvement sans se faire une réelle représentation interne de l'intention du partenaire. Cette théorie a l'inconvénient de ne pas expliquer comment les bébés peuvent comprendre les émotions des personnes avec lesquelles ils interagissent. C'est pourquoi beaucoup de recherches se concentrent sur la seconde approche, la *théorie de la simulation* [98]. Pour expliquer la capacité de compréhension de l'intention de l'autre, l'humain serait donc capable de s'imaginer dans la position de l'autre et ainsi de produire les mêmes états mentaux (c'est-à-dire, les intentions, mais aussi les croyances et les désirs) que cette autre personne [94].

Des études controversées¹⁴ s'intéressent au rôle des neurones miroirs [185, 195], qui permettraient notamment à l'observateur de comprendre le but de l'acte moteur observé. En effet, des études montrent que la perception de l'action de l'autre est liée à l'activation du système moteur [224]. Par exemple, dans l'étude [208] Stadler *et al.* demandent à des participants de visualiser par vidéos des actions quotidiennes, dont certaines parties sont cachées par des occlusions. Les résultats de cette étude montrent qu'en perturbant le cortex pré-moteur dorsal¹⁵, les participants font plus d'erreurs de reconnaissance de l'action. Cette activation du système moteur est basée sur une correspondance entre l'acte observé et la représentation motrice de l'observateur. L'observateur serait alors capable de planifier mentalement les actions qui correspondent aux mouvements effectués par l'autre agent. On parle alors de *résonance motrice*, due à l'activation des neurones miroirs [186]. L'activation de ces neurones a donc lieu à la fois quand les individus effectuent l'acte moteur, mais aussi lorsque d'autres personnes l'effectuent. Ces théories sont présentées dans [78, 187]. Dans [84], en étudiant l'activité neuronale des singes lorsque ceux-ci effectuent des actions motrices, Fogassi *et al.* ajoutent le fait que le singe reconnaît, en plus de l'action motrice courante, la chaîne motrice qui mène au but de l'action observée. Grâce à cela, ils semblent capables d'estimer l'intention de l'individu.

14. En effet, même si beaucoup d'études ont apporté des preuves directes concernant l'activité des neurones miroirs notamment chez les singes, beaucoup d'autres utilisent la théorie des neurones miroirs sans vérifier leur activité, pour expliquer des fonctions cognitives "supérieures" comme l'empathie, langage, autisme, etc.

15. Substrat neuronal impliqué dans le contrôle de l'action.

Dans [123], l'étude montre qu'en plus de reconnaître les actions, les neurones miroirs captent l'information sur l'intention de l'agent, selon *leur préhension* d'un objet (c'est-à-dire, leur manière de tenir l'objet). Pour expliquer cette observation, Kaplan *et al.* rappellent que pour pouvoir boire, les humains saisissent le manche d'une tasse avec précision, ce qui évoque en même temps le concept d'affordance¹⁶.

Ces études en sciences humaines ont beaucoup inspiré le domaine de la robotique, comme nous allons le voir dans les Sections suivantes. Notamment, les hypothèses sur les neurones miroirs ont débouché sur un ensemble d'études en robotique développementale, où les robots apprennent à réagir suivant l'intention de leur partenaire (*c.f.*, Section 1.7.1).

1.1.3 L'intentionnalité en robotique

Différents sujets d'étude concernent la modélisation de la compréhension de l'intention.

Du point de vue de l'humain, comprendre l'intention d'un robot signifie qu'il doit décrire les mouvements ou actions du robot dirigés vers un but, de manière intuitive et non ambiguë et qu'il doit comprendre ce que le robot est en train de faire, ou ce qu'il va faire [130]. Cela nécessite que le robot fasse des mouvements *prédictibles* et *intelligibles* [69]. Un mouvement est *intelligible* si un observateur peut rapidement prédire son but, tandis qu'un mouvement est *prédictible* lorsqu'il correspond à ce qu'attendrait un observateur connaissant le but. La génération de mouvements robotiques *intelligibles* a été le sujet d'études récentes. Il s'agit d'une capacité importante permettant l'interaction entre des individus, puisqu'en faisant des mouvements intelligibles, le partenaire humain peut estimer de manière plus précise son intention et donc ce type de mouvement améliore l'efficacité de la collaboration [31, 68]. Par exemple, dans [69] Dragan *et al.* utilisent des techniques d'optimisation afin de générer des mouvements à la fois intelligibles et prédictibles. Ils proposent ainsi des modèles basés sur le principe d'action rationnelle (c'est-à-dire, des actions logiques répondant à un but) et testent leurs résultats sur des utilisateurs novices qui doivent comprendre les mouvements du robot. Dans [108], Huang *et al.* utilisent une méthode d'apprentissage par renforcement inverse à partir de comportements observés (c'est-à-dire, que l'apprentissage s'effectue à l'aide de calculs probabilistes a posteriori d'objectifs). Ils appliquent cette méthode à des voitures autonomes, non pas pour que celles-ci prédisent des mouvements, mais pour qu'elles effectuent des mouvements intelligibles par les humains, qui peuvent ainsi comprendre les objectifs de ces voitures. Dans [196], grâce à l'intelligibilité des mouvements, les humains sont même capables d'estimer le poids des objets portés par le robot. Cela suggère qu'il est possible de créer des comportements robotiques compréhensibles implicitement par des utilisateurs non experts (c'est-à-dire, des utilisateurs n'ayant pas l'habitude de programmer des robots). Dans une étude, Sciutti [195] montrent que les humains attribuent implicitement des intentions dirigées vers un but aux mouvements robotiques, ce qui leur permet notamment d'effectuer un regard d'anticipation vers l'objectif.

Du point de vue de la robotique (ou d'autres applications), une première problématique consiste à doter les robots d'intentions. Pour cela, dans [136, 137], Kozima *et al.* proposent la conception d'un robot dont le comportement est défini à plusieurs échelles, de la simple "pulsion" (déclenchée par une combinaison de comportements provenant d'un répertoire de comportements, tels que la fatigue ou l'énervement), au comportement réflexe (tel qu'attraper un objet lorsque sa main touche un objet), jusqu'au comportement plus réfléchi, où le robot utilise son système sensorimoteur afin de percevoir et exploiter les affordances entre lui et l'environnement. Le robot apprend par renforcement quels comportements permettent d'atteindre ses objectifs. Le robot

16. L'affordance est la capacité d'un objet ou d'un système à suggérer son utilisation, sa fonction.

peut ainsi choisir intentionnellement son comportement, afin d’accomplir au mieux sa tâche.

Une autre problématique consiste à comprendre l’intention des humains, par exemple dans le cadre de collaboration humain-robot [228], de robots compagnons [80], ou de navigation robotique [158]. Comprendre l’intention de l’utilisateur signifie que le robot doit exploiter l’ensemble des indices verbaux ou non verbaux naturellement générés par l’humain afin d’identifier son intention, en prenant en compte le contexte courant. Plus le robot utilise de sources d’informations¹⁷, plus l’estimation peut être précise. La reconnaissance la plus simple de l’intention correspond à estimer le but d’une action en cours, sous la condition que chaque action corresponde à un mouvement dirigé vers un but.

Du point de vue des sciences informatiques, on s’intéresse souvent à formaliser l’intention de manière détaillée. Comme l’a montré la Section 1.1.1, l’intentionnalité a différentes fonctionnalités, selon l’échelle et la période temporelle de l’action sur laquelle on se focalise. Ainsi, pour formaliser l’intention dans son ensemble, cela nécessite de fournir une représentation unique qui explique à la fois l’intention motrice (contrôle du mouvement actuel) ; l’intention à bas niveau (qui correspond aux actions dirigées vers un but, comme atteindre un objet ciblé et l’attraper) ; et l’intention à haut niveau (qui correspond aux actions complexes, abstraites et cognitives, par exemple, lorsque l’on change une ampoule au plafond, cela nécessite d’utiliser une échelle, d’y grimper, de dévisser l’ampoule, etc.).

Pour modéliser la reconnaissance de l’action et de la prédiction de l’intention, plusieurs approches existent et sont présentées dans [61]. Une première possibilité correspond aux *approches descriptives*, où l’on fait correspondre les données récoltées à des labels prédéfinis. On peut trouver un récapitulatif des méthodes correspondantes dans [54]. Une deuxième possibilité correspond aux *approches génératives*, où l’on utilise des variables latentes¹⁸ afin d’apprendre les informations utiles à l’inférence de l’action. Ces approches se basent sur l’utilisation de distributions probabilistes. Des méthodes correspondant à cette approche sont présentées dans [34]. Par exemple, la méthode *HAMMER* (*Hierarchical Attentive Multiple Models for Execution and Recognition*) [62] consiste en une organisation hiérarchique et distribuée du contrôle moteur du robot qui permet à la fois au robot d’exécuter une action, mais aussi de la reconnaître lorsqu’il perçoit visuellement un partenaire en train de l’effectuer. Pour cela, une combinaison de modèles inverses et prédictifs est utilisée, qui communiquent entre eux, ce qui permet à la fois de percevoir les tâches effectuées par le partenaire et à les exécuter. Ainsi, le robot est par exemple capable de prédire que l’utilisateur va attraper une orange ou lâcher une canette (grâce aux modèles prédictifs parallèles qui s’appuient sur les observations et les modèles inverses), et il est capable d’effectuer cette même action (grâce aux observations et la connaissance des erreurs de prédictions qu’il y a eu à l’instant précédent).

Dans la prochaine Section, nous verrons plus en détail ces méthodes permettant de prédire l’intention d’un agent.

1.2 Prédiction Inférence et Anticipation

Dans cette Section, nous présenterons en détail les notions d’*anticipation* et de *prédiction*. Mais avant cela, nous commençons ici par clarifier la différence entre la *prédiction* et l’*inférence*. La *prédiction* correspond à “*l’annonce d’événements futurs par la connaissance qu’on a de leurs causes*. [Larousse]”. Cette prédiction a lieu par exemple lorsque le robot calcule la continuation d’un mouvement à partir de ses connaissances et de ses observations. L’action d’*inférer* correspond

17. Ces informations correspondent aux signaux mesurables qui proviennent de l’utilisateur et de l’environnement.

18. Les variables latentes sont des variables non directement observables, mais qui sont prédites à l’aide d’un modèle et d’autres variables qui sont, quant à elles, directement mesurées.

à “tirer une conséquence de quelque chose, conclure, induire quelque chose de quelque chose.” [Larousse]. Notre robot infère par exemple qu'elle est la catégorie de mouvement qu'il perçoit ou encore qu'elle est la vitesse du mouvement actuel et en déduit la durée totale du mouvement.

En plus d'estimer l'intention et donc le but de l'utilisateur [229, 216], le robot doit aussi *prédire* puis *anticiper* les mouvements futurs de celui-ci, afin de planifier ses propres actions, tout en respectant son intention.

1.2.1 Définitions

Dans le langage courant, les termes *prédiction* et *anticipation* sont souvent utilisés comme des synonymes. C'est pourquoi nous commençons ici par préciser leur signification et leur différence, en définissant tout d'abord ce qu'est un système anticipatif.

Un système anticipatif est “un système contenant un modèle prédictif de lui-même et/ou de l'environnement, lui permettant de changer d'état à un instant, en accord avec les prédictions de son modèle relatif à un instant futur” [189]. Plus précisément, un système anticipatif se compose d'un système prédictif, qui représente les états futurs de l'environnement avec lequel il interagit. Ces prédictions de changements d'état permettent au système d'*anticiper* et de contrôler son interaction avec l'environnement, afin de répondre à des buts. Ainsi, ce sont des processus ou comportements qui ne dépendent pas seulement des mesures passées et présentes de l'environnement, mais aussi des mesures prédites sur l'avenir, basées sur des attentes et des croyances [33].

Dans [180], Pezzulo *et al.* différencient un système anticipatif d'un système réactif. Contrairement à un simple système réactif, les systèmes anticipatifs sont capables de *formuler des prédictions* et de les utiliser afin de répondre à des buts. Ainsi en robotique, les robots qui utilisent un *système prédictif* peuvent seulement formuler une *attente*, alors qu'en utilisant un *système anticipatif*, ils choisissent une action à effectuer, en fonction de l'*espérance* qu'ils pensent en tirer. Puis, Pezzulo *et al.* expliquent dans cette même étude les différences entre l'*anticipation* et la *prédiction*, mais aussi avec deux notions liées à ces termes : l'*attente* (*prévision*) et l'*espérance*. Pour résumer, la *prédiction* apporte une *prévision* de ce qui est susceptible de se produire. L'*anticipation* quant à elle, est une *prévision* à laquelle s'ajoute la prise en compte de la préparation et l'exécution d'un comportement que l'agent a choisi d'effectuer. Finalement, l'*espérance* est le(s) résultat(s) qu'il s'attend à obtenir. Le schéma de la Figure 1.1 représente cette différence.

D'un point de vue mathématique, ces notions peuvent être formulées avec les équations suivantes :

$$\text{Prédiction : } P(x_{t+1}|e_t, x_t, \theta)$$

$$\text{Anticipation : } P(x_t|e_t, x_t, a_t, \theta).$$

Avec x_t l'état courant du processus analysé, e_t , l'état courant de l'environnement (composé de l'ensemble des événements qui interagissent avec le processus), a_t l'action prévue et θ les paramètres qui définissent la modélisation du processus étudié, qui permet de prédire et d'anticiper le futur état du système.

1.2.2 L'anticipation en robotique

En robotique, l'*anticipation* est considérée comme la capacité du robot à choisir correctement la tâche à effectuer, selon une situation donnée [101]. Pour cela, le robot doit prédire l'effet de son action, en prenant par exemple en compte le concept d'affordance [113, 191, 121] et il doit prédire les événements et états futurs, c'est-à-dire être capable de simuler l'évolution du système

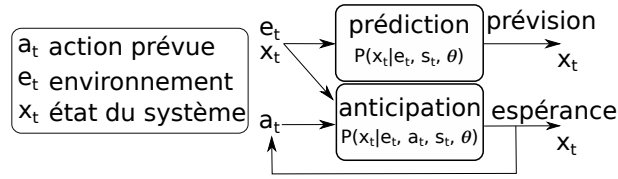


FIGURE 1.1 – Différence entre prédiction et anticipation.

couplé humain-robot, comme cela est fait dans les modèles de contrôle prédictif [241, 112] ou de planification [3, 200].

Dans [2], Alami *et al.* s'intéressent ainsi à créer un robot capable de planifier les mouvements et les tâches à effectuer, en vérifiant que le comportement du robot résultant de cette planification respecte les contraintes d'interaction avec les humains (convention sociale, sécurité, comportement de l'humain, etc.). Par exemple, l'architecture proposée permet de vérifier que les mouvements [204] et actions [2] du robot : sont compréhensibles par les humains en interaction avec lui ; sont acceptés socialement (pas trop proche, surtout d'une personne assise ; dans le champ de vision de l'utilisateur, etc.) ; diminuent l'effort d'action de la personne interactive, etc.

Dans [200], Shah *et al.* étudient les comportements humains lors de tâches collaboratives, comportements qui leur permettent de s'informer sur leurs intentions réciproques et ainsi d'anticiper et de planifier les différentes actions à effectuer. Ils ont ainsi mis en évidence que les différentes tâches à effectuer ne sont pas distribuées à l'avance (“*decision on the fly*”) ; que, plus les coéquipiers communiquent sur l'avancée de leurs actions, plus la coopération est efficace ; et finalement, que les coéquipiers prennent en compte les conséquences de leurs actions sur les autres, grâce à un modèle mental partagé de la tâche à effectuer. Shah *et al.* proposent alors un système robotique d'*exécution de plan* qui prend en compte ces différents critères afin de permettre à un robot d'*anticiper* les actions à effectuer et donc de collaborer avec ses coéquipiers humains de manière plus naturelle, plus fluide et plus rapide.

D'autres études, dont la nôtre, cherchent à doter des robots de capacités *prédictives* afin d'améliorer la collaboration humain-robot. C'est ce que nous verrons dans la Section suivante.

1.2.3 La prédiction en robotique

Pour répondre au sujet de cette thèse, le robot doit *prédire* le futur de la trajectoire qui correspond à l'intention de son partenaire, puis *anticiper* la trajectoire et la position du but de son mouvement, afin de respecter la volonté de son partenaire. Dans d'autres études, il s'agit de prédire le mouvement de l'humain, pour adapter le comportement du robot en conséquence et garantir la sécurité de l'interaction, comme dans [13], où l'algorithme prédit dans cette optique de sécurité, les zones dans lesquelles se dirigent les personnes.

Dans d'autres études, la prédiction de l'intention de l'utilisateur se limite souvent à la prédiction de la position finale. Par exemple, dans [228], Wang *et al.* proposent une méthode appelée *Modèle Dynamique Dirigé vers le But (Intention-Driven Dynamics Model)*, basée sur les *Modèles Dynamiques de Processus Gaussiens (Gaussian Process Dynamical Models)* [226]. Cette méthode permet d'inférer, lors d'un match de ping-pong, l'intention de l'adversaire humain, décrit comme la position qu'il souhaite que la balle atteigne. Cette inférence se base sur le mouvement du corps complet de l'adversaire, avant même que celui-ci frappe la balle. Dans [80], Ferrer *et al.* proposent un algorithme bayésien, nommé *Prédiction Bayésienne de l'Intention du Mouvement Humain (Bayesian Human Motion Intentionality Prediction)*, permettant de prédire l'intention du

mouvement humain. Cet algorithme consiste à calculer géométriquement le but le plus probable de l'humain, à partir de la méthode *Espérance-Maximisation* (*Expectation-maximization*) et d'un simple classificateur bayésien. Finalement, une approche classique permettant de répondre à ce problème consiste à considérer chaque mouvement comme l'instance d'une action particulière, ou d'une primitive de mouvement dirigée vers un but. L'état de l'art sur les primitives de mouvement sera présenté plus en détail dans la Section 1.7.

1.3 Interaction physique

Afin que le robot puisse prédire l'intention de son partenaire, celui-ci peut l'aider en lui fournissant des indices. Nous nous focalisons dans cette thèse sur les indices non verbaux, en commençant par les indices provenant de l'interaction physique entre l'humain et le robot.

L'intérêt porté sur l'interaction physique entre l'humain et le robot augmente de plus en plus, étant donné que l'on souhaite introduire les robots dans des environnements naturels et donc complexes, nécessitant que les robots comprennent les forces d'interaction entre eux, leur environnement et leurs partenaires humains. Il s'agit d'un sujet crucial lorsqu'il s'agit de robots d'assistance, puisque leur rôle est directement d'interagir avec des personnes non expertes, comme les robots de réadaptation, les robots domestiques ou encore les robots de soins aux personnes âgées [232].

C'est pourquoi différents projets de recherche se focalisent sur cette thématique. Par exemple, le projet *PHRIDOM* (*Physical Human-Robot Interaction in Anthropic Domains* [4]) se concentrait notamment sur la sécurité, la fiabilité (notamment des capteurs), le contrôle des architectures ainsi que la gestion des fautes et la tolérance. Les résultats de ce projet sont présentés dans [58] et concernent les critères à respecter lors de la conception d'architectures, des organes effecteurs et de contrôle. Puis, ces recherches ont été poursuivies au sein du projet *PHRIENDS* (*Physical Human-Robot Interaction : dependability and Safety*) dont le but était de développer les composants nécessaires à la création de robots interagissant physiquement avec les humains. L'état de l'art de ce projet est disponible dans [1]. Malgré ce nombre croissant d'études à ce sujet, l'utilisation de robots ne dépasse pas le cadre de la recherche, ou d'applications industrielles et médicales très spécifiques. Pouvoir les utiliser dans la vie quotidienne est en effet d'un enjeu compliqué, qui nécessite que les robots développent des capacités d'apprentissage, de prédiction, d'ajustement de leur raideur, de maintien de l'équilibre, etc.

Un récapitulatif des différentes techniques importantes répondant à ces enjeux est présenté dans la Section ci-dessous.

1.3.1 Techniques robotiques permettant l'interaction physique entre l'humain et le robot

Les problématiques nécessitant l'interaction physique entre l'humain et le robot sont multiples. Dans [135], le robot permet à l'utilisateur de **manipuler des objets lourds**, en déplaçant l'objet dans le sens de la "force intentionnelle" exercée par l'utilisateur sur l'objet. Puis, cet article présente une comparaison expérimentale de plusieurs algorithmes permettant de générer le mouvement basé sur cette force intentionnelle. Dans la même thématique, un contrôleur d'impédance [222] permet à un robot d'adapter son coefficient de viscosité de manière proportionnelle à la rigidité estimée du bras de son partenaire humain. Ce contrôleur permet de **rendre le système coopératif humain-robot facile d'utilisation et stable**. Comme exemple d'application, cette étude montre que le robot est capable d'effectuer de la calligraphie de manière coopérative avec l'utilisateur. Les mêmes auteurs proposent dans [223] un contrôleur d'impédance qui génère une force virtuelle

afin d'aider le partenaire humain à reproduire facilement une tâche de positionnement. D'autres études utilisent la capacité d'inférence du robot pour **adapter l'impédance des robots**. Ainsi, dans [99], le robot infère l'intention de l'utilisateur grâce à la mesure des forces de celui-ci et en apprenant le modèle des tâches à l'aide des *Modèles de Mixtures de Gaussiennes (GMM)*. L'adaptation de l'impédance des robots permet notamment de développer des comportements robotiques adaptés à la tâche ou à la volonté de l'utilisateur, comme présenté dans la partie 1.4. Dans [227], Wang *et al.* utilisent des *Modèles de Markov Cachés* pour apprendre à **compenser les potentielles forces physiques**. Le robot utilise alors cette méthode afin de serrer la main de son partenaire, en mesurant l'impédance de ce dernier et en reconnaissant quelle primitive de mouvement, parmi celles qu'il a apprises, il doit suivre. Dans [190], un bras robotique adapte son impédance en utilisant un algorithme basé sur des *GMM* et des mesures de forces et d'informations visuelles.

Dans le cas d'une étude de cette thèse, le robot interagit physiquement avec l'utilisateur à l'aide de l'information des forces et des couples. En effet, à partir de cette information, il estime l'intention de son partenaire et détecte la vitesse de son mouvement (*c.f.*, Section 4.3.1).

Puis, des techniques permettant de contrôler les robots à plus haut niveau ont été développées, afin notamment que les robots puissent interagir avec leur environnement. Il s'agit alors de techniques d'apprentissage qui seront présentés dans la Section 1.7.

1.3.2 Techniques d'interaction physique avec le robot iCub

Dans le cas du robot *iCub*, à partir de l'information des forces provenant des capteurs *FTS*, une technique permettant de calculer les couples et les forces externes est développée dans [116]. Cette technique consiste à améliorer un graphe qui est une représentation du modèle cinématique et dynamique du robot, à l'aide de l'information de ces capteurs. Il permet au robot de calculer la dynamique de ses membres, notamment les forces internes et externes qui y sont exercées. Cette étude a fourni la bibliothèque *iDyn* [117] open source, bibliothèque qui est utilisée dans beaucoup d'études qui concerne l'interaction physique avec le robot. Par exemple, dans l'étude [16], Berret *et al.* proposent une méthode qui module activement la compliance¹⁹ du robot, afin de gérer des situations extrêmes, telles qu'effectuer des mouvements dans un environnement où le robot subit des forces imprévisibles ou encore, dans le cas où le robot utilise des boucles de contrôles rétroactives, lorsque l'information est retardée, voir non reçue. Pour cela, des outils de contrôles stochastiques optimaux sont utilisés afin de planifier la rigidité interne de la chaîne cinématique du robot. Dans cette étude, différents modèles de compliance sont comparés, et les résultats de cette étude montrent que le modèle optimal combine un contrôle rétroactif du mouvement du robot et une loi de contrôle PD permettant d'ajuster la rigidité du robot et ce, sans avoir à utiliser un contrôle rétroactif des couples moteurs, qui a le désavantage d'être sensible aux situations extrêmes et coûteux en temps de calcul.

La bibliothèque *iDyn* a aussi permis la création d'algorithmes de contrôle du corps entier du robot [165], où les forces entre le robot et l'environnement (y compris l'humain) sont régulées afin de le stabiliser dans la position désirée. De cette expérience, une célèbre démonstration a été réalisée dans laquelle le robot fait du yoga [215]. D'autres études permettent au robot *iCub* de faire ses premiers pas ou d'effectuer des mouvements corps-complet. Par exemple, l'étude [CodyCo] consiste à permettre au robot *iCub* d'exploiter les forces de contact rigides, afin d'adapter son équilibre lorsque celui-ci se lève d'une chaise. Cette étude a notamment montré la capacité du robot à se lever avec ou sans force d'assistance provenant d'un partenaire humain.

19. c'est-à-dire, la flexibilité, l'inverse de la rigidité.

Notons que pour cette étude, il était nécessaire de tester en amont les capacités du robot en simulation afin d'éviter de l'abîmer. Or, pour pouvoir tester en simulation l'interaction physique entre l'humain et le robot, il était nécessaire de transmettre des forces d'assistance au robot simulé. Pour cela, cette étude internationale a utilisé un programme développé dans cette thèse, basé sur le dispositif haptique Geomagic Touch (*c.f.*, Section 2.5).

Ainsi, nous avons vu les différentes techniques permettant l'interaction physique en robotique. Cependant, avant de développer de telles méthodes, il est important de se demander si les potentiels utilisateurs humains acceptent le contact physique avec le robot ou de laisser de l'autonomie au robot sans contrôle complet. En effet, le risque est alors de développer des robots utiles, mais non utilisés. C'est pourquoi la Section suivante présente la notion d'acceptabilité du robot et les différents rôles que celui-ci peut prendre lors de collaboration humain-robot.

1.4 Acceptabilité du point de vue de l'utilisateur et rôle du robot

Afin que les robots soient utilisés dans la vie quotidienne, il est nécessaire de vérifier s'ils sont acceptés par les utilisateurs potentiels et que ceux-ci n'émettent pas de réticence en leur utilisation.

La notion d'acceptabilité peut être vue de plusieurs points de vue, que ce soit en termes de coût, d'utilisabilité pour l'utilisateur, ou encore d'apparence. Ici, nous nous focalisons sur l'acceptabilité du point de vue de l'utilisabilité pour l'utilisateur.

Pour évaluer cette acceptabilité²⁰, différents questionnaires et méthodes ont été créés, qui évaluent : l'attitude des humains envers les robots lorsqu'ils communiquent avec eux à l'aide du questionnaire NARS [163] (*Negative Attitude toward Robots Scale*) ; l'anxiété des humains lorsqu'ils communiquent avec des robots à l'aide du questionnaire RAS (Robot Anxiety Scale) [164] ; ou encore leur impression concernant le robot d'un point de vue anthropomorphique²¹, ou en ce qui concerne leur intelligence, leur sécurité, leur animosité, ou encore leur première impression concernant leur agréabilité, à l'aide du questionnaire Godspeed [12] (questionnaire créé afin d'aider les "créateurs" des robots à les développer) ; etc.

Dans sa thèse [177], Weistroffer rappelle aussi les différents modèles d'acceptabilité existants. Le modèle TAM (*Technology Acceptance Model*) [57] est particulièrement intéressant pour la continuation de cette thèse puisqu'il prend en compte l'utilité perçue (dans notre cas, il s'agit de savoir si l'utilisateur pense que le robot va l'aider, améliorer sa qualité de vie ou de travail) et la facilité d'utilisation perçue (dans notre cas, il s'agit de savoir si l'utilisateur a l'impression qu'utiliser le robot lui demandera de l'effort). Or, les solutions aux problématiques de cette thèse doivent justement permettre d'améliorer l'utilité et la facilité d'utilisation du robot en le dotant de capacités de prédiction.

À l'aide de ces évaluations, d'autres études se sont intéressées à l'influence de ces attitudes sur l'interaction humain-robot, par exemple dans [163] où Nomura *et al.* montrent que les participants de son expérience ayant une attitude négative envers les robots ont un comportement différent des autres lorsqu'ils interagissent avec les robots (ils communiquent moins, parlent moins).

20. Dans les questionnaires présentés dans ce paragraphe, tous ne concernent pas directement la notion d'acceptabilité, mais y sont directement liés. Par exemple, si un utilisateur a une attitude négative envers le robot, alors il est très peu probable qu'il ait envie de l'utiliser.

21. L'anthropomorphisme est le fait d'attribuer des caractéristiques du comportement ou de la morphologie humaine à d'autres entités telles que les robots, les animaux ou encore à des objets.

De plus, l'étude [166] souligne le fait que les humains aiment moins interagir physiquement avec les robots et préfèrent à la place communiquer par des gestes ou verbalement. En outre, dans l'étude [M.E], nous avons noté que les participants de l'expérience ne souhaitaient généralement pas que le robot soit totalement autonome, qu'ils voulaient contrôler ses actions. Beaucoup de sujets de recherche s'intéressent alors à l'adaptation du comportement et au rôle du robot, afin qu'il soit accepté par leur utilisateur. Pour cela, certaines études cherchent à donner aux robots la capacité de générer des "mouvements humains" [131, 129, 40]. Par exemple dans [240], les chercheurs exploitent la redondance cinématique des bras du robot lors de l'exécution de "mouvements humains", en s'inspirant de mouvements réels effectués par des humains pour une même tâche.

D'autres scientifiques cherchent à créer des robots capables de comprendre **quand et comment ils doivent assister**. C'est ce que nous verrons dans la Section ci-dessous. Concernant cette thèse, le robot peut comprendre quand et comment il doit assister l'utilisateur à partir du guidage manuel et de la prédiction du mouvement futur. En effet, lorsque l'utilisateur lâche la main du robot après avoir initié un mouvement, celui-ci comprend qu'il doit agir et comment, en prédisant la continuation du mouvement initié.

1.4.1 Rôle du robot et type d'autonomie

Lors d'interaction physique humain-robot, le rôle du robot est généralement fixe : soit il dirige totalement (*leader*), soit il suit totalement le mouvement de son partenaire (*follower*).

Dans [206], Soh *et al.* proposent au contraire un fauteuil roulant robotique qui **apprend à varier entre ces deux rôles**. En effet, pour assister la personne handicapée de manière efficace, ce fauteuil roulant robotique distingue par apprentissage les périodes où il doit assister la personne de celles où il la laisse guider (*Learning Assistance By Demonstration*).

En plus de cette capacité à étudier **quand** le robot doit agir, d'autres études s'intéressent aussi à **comment** il le doit, en suivant la volonté de l'utilisateur. Par exemple dans [42], Carlson *et al.* proposent un autre fauteuil intelligent, qui apprend quand et comment il doit agir. Pour cela, leur méthode permet au robot de détecter l'intention de l'utilisateur de manière probabiliste, puis si cette probabilité est suffisamment forte et que la trajectoire de l'utilisateur varie des mouvements appris et que cela peut mener à une collision, alors le robot agit.

Dans [122], Jarrasse *et al.* se focalisent sur la capacité **des robots à ne pas interférer avec les forces** de l'utilisateur, en rendant les mouvements du robot *transparents*²². D'autres études s'intéressent à cette même idée, en adaptant notamment l'impédance du robot, selon le comportement qu'il doit avoir. Par exemple, dans l'étude [89], le robot différencie les forces de contact involontaires (provenant par exemple de collisions), de celles qui sont volontaires (provenant de l'utilisateur), afin d'adapter son comportement en se mettant en mode collaboratif.

Sur cette même idée, l'expérience de Baraglia *et al.* [11] montre que **les robots ne doivent pas interférer avec les actions** de l'utilisateur puisque, lors de cette expérience, les participants préfèrent les robots qui aident de manière proactive (c'est à dire dès que possible, tout en prenant en compte les actions de l'humain avec qui il collabore). Ainsi, cette étude montre que les robots ne doivent pas exécuter de tâches qui rentrent en conflit avec la tâche exécutée par l'utilisateur humain. Mörtl *et al.* complètent cette étude dans [159], où ils s'intéressent à comment les humains acceptent les changements de rôle (robot maître/esclave). En effet, cette étude montre que, même

22. C'est-à-dire que le robot suit les mouvements de l'utilisateur sans que celui-ci ne ressente pas de résistance provenant du robot.

si la tâche collaborative est effectuée de manière plus efficace d'un point de vue objectif et que cela demande moins de charges mentale et physique à l'utilisateur humain lorsque le robot change de rôle petit à petit de manière proactive, d'un point de vue subjectif, ils préfèrent lorsque le rôle du robot est constant. Ce résultat est intéressant pour le développement de la robotique futur, mais il dépend du contexte de l'expérience. En effet, il arrive que l'utilisateur ne cherche pas à garder tout le contrôle, par exemple lorsqu'il s'agit de tâches peu importantes. C'est pourquoi des études supplémentaires devront être effectuées afin de comprendre dans quels contextes ce résultat est véridique.

Ces différentes études mettent en avant des intérêts subsidiaires de cette thèse. En effet, en optant pour des robots capables de suivre la volonté de leurs utilisateurs, sans planifier eux-mêmes les actions et les mouvements à effectuer, l'utilisateur peut à la fois contrôler le robot, tout en permettant à ce dernier de s'adapter au contexte au travers de l'intention de l'utilisateur.

De plus, ils mettent en avant l'un des avantages des deux dernières études de cette thèse, où l'utilisateur peut guider le robot sans avoir à interagir physiquement avec ce dernier, interaction qui n'est donc pas toujours appréciée par les utilisateurs.

Finalement, la première étude de cette thèse pourrait permettre à l'utilisateur de faire varier le rôle du robot selon sa volonté, à l'aide de l'information des forces qu'il exerce sur le robot.

1.5 Comprendre l'intention par le regard

La Section précédente présentait donc l'utilisation de forces d'échange physique comme indice non verbal permettant à un utilisateur d'indiquer son intention à un robot. Cependant, différentes études ont permis de montrer que l'interaction physique avec un robot peut être difficile pour les humains, en particulier non experts. C'est pourquoi, dans la seconde étude de cette thèse, le robot estimera l'intention de son partenaire à partir de ses mouvements de tête, informant sur la direction du regard. Cette Section présente l'état de l'art sur la compréhension de l'intention par le regard, en commençant par l'explication du fonctionnement de cette compréhension chez l'humain.

1.5.1 L'attention conjointe chez l'humain

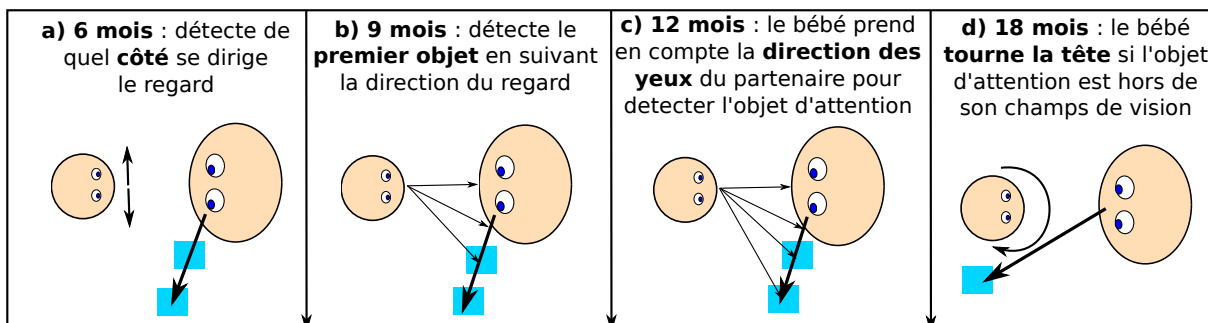


FIGURE 1.2 – Progression développementale du suivi de regard, inspiré de Scassellati 2002.

La direction du regard est un indicateur fondamental de l'intention, permettant l'interaction sociale, puisqu'elle permet l'attention conjointe.

Le scénario de la seconde étude de cette thèse (*c.f.*, Section 5) s'est inspiré de cette notion d'attention conjointe, en permettant au robot d'identifier, non pas la direction du regard, mais la direction faciale de son partenaire afin d'identifier son intention.

Dans cette Section, nous commençons donc par définir ce processus chez l'humain.

« *L'attention conjointe correspond à la capacité à partager un événement avec autrui, à attirer et à maintenir son attention vers un objet ou une personne dans le but d'obtenir une observation commune et conjointe* ». ²³.

Pour effectuer ce comportement, l'individu identifie d'abord la tête (et sa position) de son partenaire, puis il estime la direction de son regard, pour finalement repérer l'objet sur lequel se porte l'attention. Cette attention conjointe est connue pour être initiée par le regard, mais elle peut aussi avoir lieu à l'aide des mouvements de mains [239], du toucher, ou encore du son.

Cette capacité aide le développement de l'humain, par exemple pour l'apprentissage multimodal (*c.f.*, Section 1.6), ou pour l'acquisition du sens sémantique [220]. En effet, l'étude de Benitez *et al.* [15] montre que, pour apprendre le nom d'un objet, le nouveau-né lie d'abord le nom de l'objet avec la zone d'attention ; puis le nom de l'objet avec ses caractéristiques ; et pour finir les caractéristiques de l'objet avec la zone d'attention (c'est-à-dire, nom-zone \rightarrow nom-objet \rightarrow objet-zone). Après que ces liaisons soient apprises et avec de l'expérience, le nouveau-né retient alors uniquement la liaison "nom-caractéristique" et oublie la liaison "caractéristique-zone".

Le développement de l'attention conjointe chez le nourrisson peut être séparé en quatre étapes [32] représentées dans la Figure 1.2 : (a) vers *six mois*, le bébé apprend à discriminer la *direction globale* du regard de son partenaire (gauche ou droite) ; (b) vers *neuf mois*, le bébé est capable de *détecter le premier objet* présent sur la *ligne de direction* du regard de son partenaire ; (c) vers *douze mois*, le bébé est capable de prendre en compte la *direction des yeux* de son partenaire, afin de déterminer exactement l'objet d'attention du partenaire, parmi ceux présents sur la ligne de direction du regard [27] ; (d) finalement après *dix-huit mois*, le bébé est capable de *tourner la tête* vers l'objet d'attention de son partenaire, afin de reconnaître les objets d'attention non visibles dans son champ de vision. Pour avoir un aperçu plus détaillé des différentes étapes de développement de l'attention conjointe de l'enfant, voir [41] Partie "joint attention".

1.5.2 La direction du regard en robotique

La communication par le regard est souvent étudiée afin de comprendre l'interaction entre les humains. Dans [87], ils proposent un modèle graphique²⁴ Bayésien où un agent artificiel apprend à lier ses sens à ses actions (apprentissage probabiliste du modèle en utilisant la méthode du "babillage"). Grâce à cet apprentissage, l'agent artificiel est alors capable d'estimer le but de son partenaire, ainsi que de suivre son regard (c'est-à-dire, regarder dans la même direction que son partenaire).

Beaucoup d'autres études s'intéressent à cette aptitude. Nous nous focalisons ici sur les études concernant l'interaction humain-robot. Cette étude du regard permet tout d'abord d'*analyser l'Interaction Humain-Robot (HRI)* [43] [7] [110] [118] [119]. Par exemple, dans [114], un simple mécanisme d'attention conjointe permet de faire comprendre à l'utilisateur que le robot va effectuer une action (on dit qu'il le perçoit comme "engagé") et comprend globalement son comportement. D'autres étudient le regard pour *corriger le comportement du robot* [22], ou pour détecter l'attitude de l'utilisateur envers le robot [118]. En outre, cette étude peut permettre d'*assurer la sécurité de*

23. Définition de ORANOV (ORthophonie et enfants porteurs d'Autisme NON-Verbeaux)

24. Où chaque noeud de ce graphe représente une variable telle que le but G , l'action A , l'état actuel X_i , ou encore l'état final X_f et où les liens entre ces noeuds représentent les différents rapports de causalité entre ces variables.

l'interaction [221], ou encore afin d'*anticiper l'action du partenaire d'un robot*, comme dans [107], où le robot utilise cette anticipation pour effectuer une tâche de manière "proactive". Notons que cette méthode nécessite que l'utilisateur porte des lunettes *SMI Eye-Tracking*, ce qui peut limiter son utilisation (due au coût, à la gêne occasionnée, etc.). Pour le même type de but (c'est-à-dire, pour de l'anticipation), les études de cette thèse fournissent d'autres méthodes permettant à l'utilisateur d'utiliser le robot sans nécessiter le port de lunettes. Dans d'autres études, le regard permet d'*améliorer des logiciels*. Par exemple, dans [9] une interface permet la reconnaissance des gestes du bras de l'utilisateur à partir d'un modèle multimodal qui décrit la relation causale entre le regard de l'utilisateur et d'autres modalités (c'est-à-dire, des gestes de la main et actions correspondantes sur l'interface). Cette étude du regard permet aussi de *contrôler les déplacements des robots* [158] ou encore d'*adapter l'action des robots selon l'intention de leur utilisateur*. Outre cette thèse, c'est le cas de l'étude [137] où un robot enfantin comprend le comportement de l'utilisateur en apprenant à corréliser les buts de l'utilisateur (ou intention) avec ses comportements.

Ce dernier exemple correspond à notre objectif, où le robot *iCub* doit agir selon l'envie de l'utilisateur. Dans ce but, l'orientation du regard du partenaire doit tout d'abord être calculée. Pour ce faire, différentes approches existent. Par exemple, dans [10], Baluja *et al.* utilisent les Réseaux de Neurones (RNN) pour traquer le regard de l'utilisateur avec une précision de 1.5° et ce, de manière non intrusive. Dans [218], Timm *et al.* utilisent des calculs de gradients sur les images, pour localiser précisément le centre des yeux [212]. Beaucoup de méthodes robotiques utilisent des approches probabilistes, dans lesquelles le regard est souvent utilisé comme un *a priori* concernant la tâche à effectuer. Cet *a priori* permet par exemple de détecter l'objet qui semble intéresser la personne (par exemple, à l'aide de réseaux de neurones [102]), ou de prédire la position du but (c'est-à-dire, dans [44] avec des modèles dynamiques), ou encore de prédire la trajectoire à effectuer (à l'aide de primitives de mouvements, *c.f.*, Sections 1.7 et 3.3). Dans [52], un traqueur du regard de l'utilisateur est implémenté à partir de *Modèle Localement Contraint* [51], en se basant sur les textures et les formes, à l'aide d'un modèle d'apparence active. Ce modèle génère un ensemble de détecteurs de régions du visage humain, correspondant à des régions rectangulaires autour de chaque caractéristique apprise par le modèle. Cette étude montre que leur méthode est plus performante que les méthodes connues suivantes : *Modèle d'Apparence Active* [71], où le modèle permet de repérer la région entière de l'objet à reconnaître et non un ensemble de caractéristiques locales ; *Traqueur de sélection de modèle* [53] ; ou encore *Recherche optimisée de formes* [52]. Cette méthode permet cependant de détecter uniquement le visage de l'utilisateur. Dans d'autres études, le regard de l'utilisateur est directement détecté en utilisant des lunettes [110, 158, 141].

Dans notre cas, nous utilisons le logiciel Intraface 2.5 qui a l'avantage de ne pas nécessiter que l'utilisateur ait à porter des lunettes et dont le code est ouvert, mais qui est légèrement moins précis que ces lunettes.

1.6 Perceptions multimodales

Dans la seconde étude de cette thèse (*c.f.*, Section 5), l'utilisateur peut choisir de communiquer avec le robot à partir d'interaction visuelle et/ou physique. De plus, lorsque l'utilisateur utilise l'interaction physique pour guider le robot, ce dernier perçoit à la fois le mouvement de son propre bras (proprioception) ainsi que les forces exercées sur son bras (à partir notamment du toucher). Il s'agit donc de perceptions multimodales. La perception multimodale est une compétence importante chez l'animal qui permet notamment le développement et l'apprentissage de compétences plus complexes, en corrélant l'information de plusieurs modalités à la fois. Cette

Section a pour but de détailler cette notion de multimodalité.

1.6.1 Multimodalité chez l'humain

L'information multimodale est couramment utilisée chez l'humain. Lors de l'écoute d'un dialogue, l'humain utilise plusieurs modalités, principalement l'audio et le visuel. De plus, pour communiquer, McNeill montre dans [149] l'importance du lien entre le discours et les gestes, avec notamment différents gestes récurrents liés aux mots utilisés.

L'information visuelle permet notamment de compléter les informations auditives et de clarifier les confusions acoustiques [211].

Les capacités multimodales de l'humain se développent très tôt. Pour cela, l'attention conjointe, terme défini dans la Section précédente, est utilisée pour développer la "mémoire indexée spatialement" [184]. En effet, elle permet de détecter un sous-ensemble d'objets d'intérêt présents dans la zone de focalisation et ainsi de faciliter la liaison entre son et visuel [81, 213]. Confortant cette idée, des études ont montré qu'à huit mois, le nouveau-né fait le lien entre la vision et le son uniquement lorsque la localisation des événements multisensoriels est « corrélée » (dans le sens où, lorsqu'un signal provenant d'une modalité est perçu, le nouveau-né s'attend à recevoir une information de l'autre modalité dans la même zone [233, 59]). La théorie du "like me" de Meltzoff propose plus généralement que, lorsque le nouveau-né imite les personnes avec qui il interagit, il établit des relations entre ce qu'il fait et ce que ces personnes font. Il s'agit alors d'une connexion visuo proprioceptive ou audio proprioceptive. Cette théorie se base sur l'hypothèse que l'imitation est un processus inné (lié aux neurones miroirs, présentés dans la Section 1.1.2 qui permet notamment de corréliser les modalités proprioceptive et visuelle) et en déduit que la compréhension de l'autre en est une conséquence. Plus que théoriques, ces expériences étayaient ces hypothèses [153].

Les nouveau-nés sont d'ailleurs sensibles à l'effet *McGurk* [30], ce qui met en évidence l'utilisation combinée du son et du visuel. L'effet *McGurk* est une illusion multimodale [148] : lorsque l'on voit quelqu'un prononcer la syllabe "ga" alors que l'on entend la syllabe "ba", alors on perçoit la syllabe "da". De manière similaire, l'illusion de la main en caoutchouc correspond cette fois-ci à un mélange des modalités visuelle, du toucher et de la proprioception [21]. Il s'agit de créer l'illusion optique qu'une main en caoutchouc correspond à la main d'une personne, pour que celle-ci ait l'impression de ressentir ce qui se passe sur la main en caoutchouc. En étudiant le cerveau humain, les chercheurs ont découvert un réseau d'interaction intermodale, pouvant expliquer ces capacités multimodales. Ce réseau d'interactions intermodales s'activerait dès lors qu'une des modalités perçoit l'information [86].

1.6.2 Multimodalité en robotique

Cette Section montre qu'afin d'améliorer l'interaction humain-robot, différentes études se sont concentrées sur la capacité robotique d'acquérir des compétences multimodales.

Dans [106], un robot apprend quels gestes il doit effectuer avec ses mains et sa tête, en fonction des mots du discours, afin de pouvoir faire une présentation semblable à une présentation humaine. Pour cela, le robot utilise un modèle appelé *Réseau Bayésien Dynamique* qui connecte différents états latents avec d'autres états représentant les mouvements du regard et les gestes des bras, ainsi qu'avec des variables booléennes, dont chacune correspond à une caractéristique du langage parlé (*par ex.*, référence concrète, objet concret, pronom, etc.). Il s'agit d'un modèle graphique, où à chaque itération t un certain espace latent reçoit l'information de l'état latent précédent ($t - 1$) et est connecté avec les autres états (geste, regard et discours). De plus, pour chaque itération,

l'espace de discours est connecté avec les états gestes et regard. En effet, leur modélisation suppose que le discours provoque les gestes de la tête et des bras. L'apprentissage de ce modèle s'est fait à l'aide de données provenant d'une présentation orale d'un humain, catégorisées par différentes personnes (c'est-à-dire, le type de gestes et certaines caractéristiques du discours).

Dans [238] Yamazaki *et al.* rappellent l'importance des actions non verbales, telles que les mouvements de tête des partenaires, dans l'interaction humaine. Basés sur ce fait, ils permettent alors à leur robot de faire des mouvements de tête qui semblent naturels, en direction des choses qu'il présente verbalement dans un musée.

Dans la même idée, l'algorithme proposé dans [147] permet à un robot d'apprendre la connexion sémantique entre les mots et les gestes. Ainsi, le robot est capable de repérer des mots dans un discours et d'effectuer des gestes associés. Pour ce faire, la méthode consiste en un calcul mathématique par factorisation matricielle.

Plus généralement, beaucoup d'études montrent que l'information multimodale améliore nettement les performances de classification. Cela est par exemple montré dans l'étude [70], où Droniou *et al.* montrent qu'avec un apprentissage multimodal, les reconnaissances uni modales sont aussi améliorées.

Concernant le robot *iCub*, l'étude [8] propose une approche permettant à ses partenaires de ressentir une "co-présence" lors de leur interaction avec le robot. En effet, à l'aide d'informations visuelles et auditives, le robot regarde ses partenaires.

Dans la deuxième expérience de cette thèse de la Section 5, nous utiliserons l'information visuelle et l'information physique afin que le robot améliore ses compétences de reconnaissance des mouvements à effectuer. Nous verrons que dans certains cas, une seule modalité est suffisante et que dans d'autres cas, l'utilisation des deux modalités améliore les capacités de reconnaissance.

1.7 Primitives de Mouvements

Afin que le robot puisse apprendre et reconnaître des trajectoires, nous avons donc décidé de représenter les trajectoires en tant que primitives de mouvement. Cette section présente les différentes modélisations existantes de ces primitives de mouvement. Puis, le choix de modélisation utilisée dans cette thèse sera explicité. Finalement, quelques exemples de logiciels open-sources connexes qui correspondent aux différentes méthodes seront présentés, et il y sera précisé l'apport de notre logiciel parmi eux.

1.7.1 Modélisation

Pour représenter des trajectoires complexes, une approche classique consiste à raisonner en termes de primitives de mouvements. Cette approche s'inspire des théories sur le fonctionnement de l'appareil moteur biologique [55, 92]. En effet, sachant que l'appareil musculo-squelettique est composé de beaucoup d'articulations, le contrôle moteur est complexifié par les redondances possibles (c'est-à-dire, que pour atteindre une même position, différentes postures sont possibles). Une hypothèse courante est alors qu'il y ait des synergies musculaires (c'est-à-dire, une activation cohérente, dans l'espace et/ou dans le temps, d'un ensemble de muscles) qui permettent de décomplexifier ce contrôle moteur [55]. Dans l'étude [92], Gister *et al.* présentent tout d'abord ces synergies dans le cas de mouvements d'une grenouille, puis ils concluent que l'on peut considérer la moelle épinière comme possédant un petit nombre de Primitives de Mouvements.

Les sciences informatiques s'inspirent de ces phénomènes biologiques. Les primitives de mouvements artificielles permettent de générer la séquence de points d'une trajectoire, à partir d'un modèle temporel paramétrique ou par un système dynamique paramétrique.

Pour représenter ces primitives de mouvement, beaucoup de méthodes ont été proposées au fil des dernières décennies, telles que les Modèles de Mélange de Gaussiennes [37, 127], les Primitives de Mouvement Dynamique [109], les Primitives de Mouvement Dynamique Probabiliste [150]) ou encore les Primitives de Mouvement Probabiliste [170]). Nous présenterons les principales méthodes dans les Sections suivantes.

Ces différentes techniques d'apprentissage sont utilisées dans différents types de scénarios, tels que permettre à un robot à jouer au ping-pong, écrire des nombres, éviter des obstacles lors de mouvements de "*pick and place*", etc. (*c.f.*, Section 1.2.3). Souvent, il s'agit de trajectoires cinématiques qui correspondent à des gestes effectués par le robot en réaction aux actions de son partenaire. D'autres études s'intéressent à des trajectoires plus complexes, qui contiennent par exemple l'information des forces. C'est le cas de l'étude [169], où des trajectoires sont planifiées afin d'atteindre des positions spécifiques. Ces trajectoires permettent de transporter des objets, tout en déterminant le poids des objets transportés. Ce dernier types de scénarios correspond notamment aux études sur l'interaction physique entre l'humain et le robot, où le robot doit effectuer des tâches qui requièrent une collaboration avec l'utilisateur et donc des échanges de forces avec celui-ci. Les informations cinématiques résultantes de ce type de trajectoires ne peuvent pas être analysées sans prendre en compte l'échange haptique entre l'humain et le robot (c'est-à-dire, les phénomènes kinesthésiques liés au contacts physique entre l'humain et le robot) et parfois, quand ce n'est pas déterminé à l'avance, l'estimation des rôles des partenaires (c'est-à-dire, d'estimer quand est-ce que le robot doit suivre le partenaire, ou agir de manière autonome).

Dans tous ces scénarios, les utilisateurs effectuent généralement des démonstrations (c'est-à-dire, qu'ils réalisent eux-mêmes les mouvements qui correspondent aux différentes tâches) soit en guidant manuellement le robot (démonstrations kinesthésiques), soit en utilisant la téléopération. Il s'agit alors d'apprentissage par démonstration. Certaines études de ce type ont aussi été

appliquées au robot humanoïde *iCub*. Par exemple, dans [209] Stulp *et al.* utilisent des Primitives de Mouvement Dynamique afin d'adapter en ligne le mouvement dirigé vers un but, afin d'éviter les obstacles variables rencontrés par le bras robotique. Dans [172], Paraschos *et al.* utilisent les Primitives de Mouvement Probabiliste afin d'apprendre comment le robot doit incliner un râteau, à l'aide d'informations sur les forces et les couples. Pour décrire ces primitives de mouvements, différentes approches ont été étudiées.

Les Réseaux de Neurones

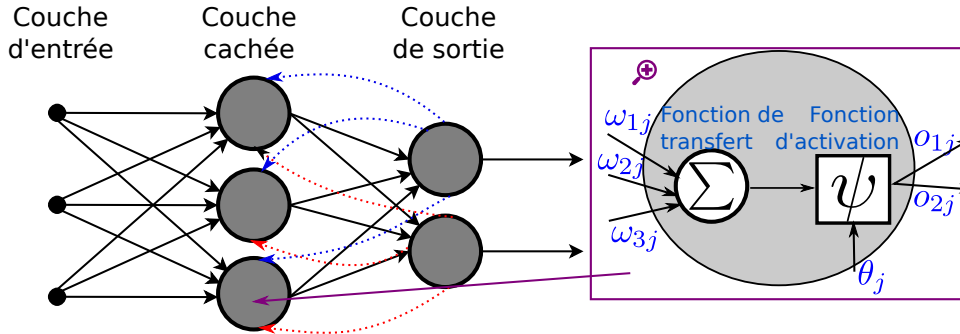


FIGURE 1.3 – Exemple d'un simple et schématique réseau de neurones.

Une première approche consiste à modéliser ces primitives de mouvements à l'aide de Réseaux de Neurones artificiels. Cette approche s'inspire de la modélisation mathématique des neurones biologiques, présentée par la Figure 1.3. Le fonctionnement est le suivant : chaque neurone artificiel est représenté par une variable, dont la valeur représente sa stimulation (c'est-à-dire, l'énergie qu'il reçoit). On a par exemple $O_{i,j}$ pour le i^e neurone de la j^e couche. De plus, les neurones interagissent entre-eux en transmettant une partie de l'énergie qu'ils reçoivent, modélisée par une fonction de transfert (ici Σ). La première couche du réseau se compose de neurones qui reçoivent l'information provenant directement de données d'entrées, par exemple de capteurs. Cette information va les stimuler plus ou moins, ce qui est modélisé par une fonction d'activation ϕ . L'activité de ces neurones va ensuite stimuler d'autres neurones qui leur sont connectés (dans le cerveau humain, cette connexion se fait par les synapses), qui vont à leur tour stimuler d'autres, jusqu'à atteindre des neurones qui vont activer des actions motrices. On dit que ces neurones composent une "couche de sortie".

Différentes modélisations de réseaux de neurones existent, en voici un exemple :

$$O_{i,j} = \psi\left(\sum_{n=1}^N \omega_{n,j} \cdot O_{n,j-1} + \theta_j\right).$$

Avec $O_{i,j}$ la valeur d'activation du i^e neurone de la j^e couche ; Σ , une simple fonction de transfert qui somme les énergies reçues ; $\omega_{i,j}$ la proportion d'énergie transmise par le neurone $O_{i,j-1}$; et ψ la fonction d'activation qui modélise l'énergie contenue dans le neurone en fonction de l'énergie qu'il reçoit, énergie auquel on ajoute une énergie constante θ_j .

Les réseaux dynamiques peuvent contenir des cycles entre les neurones. C'est le cas des Réseaux de Neurones Récurrents (RNN), souvent utilisés pour modéliser l'apprentissage de séquences temporelles. L'étude [72] fournit une explication de ce type d'approche, où chaque unité neuronale est connectée avec elle-même, ce qui lui permet d'avoir une information sur son état interne passé. Avec les RNN à Temps Continues (CTRNN), les neurones sont non seulement

activés par l'entrée synaptique courante, mais aussi par les états neuronaux passés. Grâce à cette caractéristique, l'activité des neurones change de manière continue et est donc meilleure pour produire des séquences sensorimotrices continues. Basée sur cette méthode et sur l'idée que les comportements complexes de l'humain sont créés à partir de l'intégration de plusieurs primitives motrices, l'étude de Yamashita *et al.* [237] propose un mécanisme neuronal qui segmente les informations sensorielles et motrices en primitives distinctes, pouvant être à la suite combinées et ce, même pour de nouvelles séquences. Pour illustrer leur étude, un robot humanoïde apprend cinq comportements qui correspondent à des séquences différentes de mouvement, dans lesquelles le robot déplace un objet cubique sur une table.

De même, dans [168], la méthode utilisée combine deux CTRNNs totalement connectés : un réseau "bas niveau" permet de recevoir les informations sensorielles (neurones d'entrées) et de générer les commandes moteurs correspondantes (neurones de sorties) ; et des unités neuronales externes, nommés "neurones de contrôles", permettent d'influencer les fonctions du réseau "bas niveau" afin de favoriser la génération de primitives motrices. L'étude présente alors des résultats sur une simulation du robot *Khepera*. Par exemple, dans la première expérience, le robot doit alors apprendre des comportements de type "tourner à droite" ou "tourner à gauche" dans un couloir avec des intersections (en forme de "T"), de telle sorte que le robot évite les collisions avec les murs.

Dans [17], Billard utilise une Hiérarchie de Modèles de Réseaux de Neurones, où chaque modèle représente le fonctionnement de l'une des zones du cerveau humain qui permet le contrôle visuo-moteur. Cette méthode permet à cette simulation de robot humanoïde de reproduire les gestes d'un partenaire, de manière aussi naturelle que si un être humain le faisait.

Une autre approche, les Réseaux de Neurones à réservoirs (ou réservoir computing) [194] consiste à modéliser des réseaux récurrents qui contiennent une couche de neurones d'entrée, un réservoir et une couche de sortie. Ce réservoir est un ensemble de neurones interconnectés de manière aléatoire. L'inconvénient de ce type d'approche est le temps de calcul nécessaire, dû aux boucles contenues dans le réservoir. Cette approche a été testée pour la reconnaissance vocale, la prédiction de séries temporelles avec des résultats plus performants qu'avec les réseaux classiques. Les deux principales méthodes de cette approche sont alors les Machines à état liquide [142] et les *Echo State Network (ESN)* [120].

Modèles de Markov Cachés (HMM)

Les modèles graphiques correspondent à une autre approche classique permettant de décrire les interactions dynamiques et de classer les actions. Ces modèles correspondent à des graphes où chaque nœud représente un état du mouvement.

Cette section présente la méthode la plus utilisée, les Modèles De Markov Cachés (HMM) qui sont des modèles stochastiques de processus markoviens²⁵ de paramètre inconnus. La Figure 1.4 présente un exemple simplifié et factice de ce type modèle. Ces modèles sont utilisés dans de nombreuses applications, telles que la reconnaissance de formes (notamment d'écrits) ; la modélisation de langage naturel [181] ; la compréhension de phénomènes physiques statistiques ; la prédiction de gènes [19] ; ou encore la création d'intelligences artificielles. Le cas d'application le plus proche du sujet de cette thèse, consiste à utiliser ces modèles afin catégoriser des mouvements, où chaque catégorie correspond alors à une primitive de mouvement. De plus, les HMMs encodent la séquence temporelle du mouvement. L'avantage des méthodes utilisées dans cette thèse est que contrairement aux HMMs, elles ne permettent pas uniquement de reconnaître le type de

25. Système stochastique dans lequel l'état futur du système ne dépend que de l'état présent et non les états passés (c'est-à-dire, "pas de mémoire").

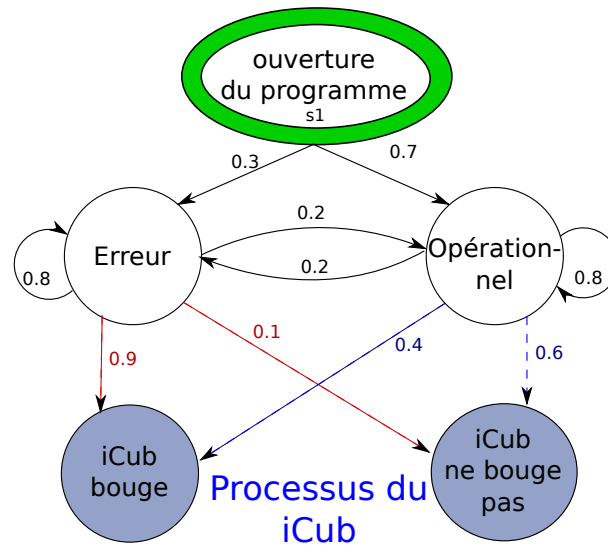


FIGURE 1.4 – Exemple simplifié et factice d'un graphe HMM, avec en vert l'état initial, en blanc deux états possibles et en bleu des états observables.

mouvement (la catégorie) mais aussi prédire précisément comment va être la poursuite du mouvement initié.

Formalisme des Modèles de Markov Cachés Les HMMs se définissent à l'aide du quadruplet $\{S, \Pi, A, B\}$, avec : des états $S = \{S_1, S_2, \dots, S_N\}$; une génération de séquences en passant d'un état à un autre $s_{i1}, s_{i2}, \dots, s_{ik}, \dots$; $P(s_{ik}|s_{i1}, s_{i2}, \dots, s_{ik-1}) = P(s_{ik}|s_{ik-1})$

À chaque étape on a :

- . S_i état i ;
- . $\pi_i = P(S_i)$ probabilité que S_i soit l'état initial;
- . a_{ij} probabilité de la transition $P(s_i|s_j)$;
- . $b_i(k) = P(k|S_i)$ probabilité d'observer k dans l'état actuel S_i .

Avec : $\sum_i \pi_i = 1$; $\forall i \sum_j a_{ij} = 1$; et $\forall i, \sum_k b_i(k) = 1$.

Beaucoup d'améliorations de ce modèle existent, qui permettent notamment d'encoder des caractéristiques spatio-temporelles, avec des algorithmes de programmation dynamique (c'est-à-dire, Viterbi²⁶). Par exemple, Brand *et al.* [23] proposent une méthode appelée Modèles Markov Cachés Couplés qui couple des HMMs avec des conditions de probabilité temporelle. Cela leur permet notamment de modéliser les mouvements de chaque bras à l'aide d'un HMM spécifique, tout en conservant une coordination entre ces bras. Dans [82], la méthode de Modèles de Markov Cachés Hiérarchiques permet d'ajouter une hiérarchie dans les états cachés, afin de modéliser les structures complexes à plusieurs échelles. Ces structures complexes existent dans des séquences naturelles (c'est-à-dire, le langage ou l'écriture). Une limite de cette étude, présentée dans [29], est que la hiérarchie des états cachés est structurée de manière arborescente, ce qui empêche le partage de sous-état parmi les états de niveau supérieur. Cette limite est visible lorsque l'on veut modéliser les comportements humains, puisqu'une même action peut être utilisée pour différentes tâches, telle que l'action "prendre un verre" qui peut être exécutée dans le but de le jeter ou de le boire). Dans cette étude, ils améliorent alors la technique précédente, en permettant le partage de sous-état entre les états de niveau supérieur.

26. Algorithme permettant de trouver la séquence d'états la plus probable d'une séquence mesurée [85]

Modèles dynamiques divers

Différentes méthodes basées sur des modélisations dynamiques ont été créées, souvent inspirées par les découvertes scientifiques en biologie.

Par exemple dans [193], Schaal *et al.* proposent une méthode afin de créer des mouvements complexes similaires à ceux humains. Pour cela, la méthode se base à la fois sur un système dynamique qui génère des motifs de mouvements moteurs afin de guider le robot (Générateurs de Motifs Moteurs) et sur un attracteur inclus dans ce modèle dynamique, qui correspond à la position désirée du robot, afin que ses mouvements se stabilisent dans la position souhaitée (Générateurs de Motifs Programmables, PPG). En mettant à la fois un PPG discret et un PPG rythmique, le robot peut alors suivre des trajectoires, qu’elles soient Cartésiennes “point à point” ou rythmiques.

De plus, dans [60], Degallier *et al.* utilisent une approche inspirée par les vertébrés, qui consiste en une génération modulaire de mouvements : les mouvements correspondent alors à des combinaisons d’un ensemble fini de primitives motrices stables, organisées au niveau de la colonne vertébrale. À l’aide de cette idée, ils permettent au robot *iCub* d’effectuer des tâches compliquées, tel que ramper comme un nourrisson. Pour cela, des primitives motrices sont générées par des systèmes dynamiques. Ces primitives motrices sont modélisées par une méthode nommée “Central Pattern Generator (CPG)”. Elles génèrent des séquences spatio-temporelles d’activation des moteurs. Les CPGs sont alors activés par le robot à l’aide de contrôle et de rétroaction sensorielles, ce qui provoque alors des séquences d’actions complexes, sans que le robot n’ait à calculer directement les commandes motrices.

Finaleme nt dans [229] Wang *et al.* proposent un modèle nommé *Modèle Dynamique Dirigé par l’Intention (Intention-Driven Dynamics Model (IDDM))* qui modélise de manière probabiliste les mouvements : il s’agit d’un modèle dynamique de processus Gaussien qui incorpore explicitement l’intention, considéré comme une simple mesure.

Primitives de Mouvements Dynamiques (DMPs)

La méthode Primitives de Mouvements Dynamiques [109, 192, 150] (*Dynamic Movement Primitives*, DMPs) est une autre méthode modélisant la dynamique des trajectoires. Elle formalise mathématiquement les primitives de mouvements comme des systèmes d’attracteurs non-linéaires (grâce à une fonction non-linéaire qui force la trajectoire à prendre une certaine forme) stables (puisque l’attracteur assure que la trajectoire atteigne au but escompté). Cette méthode est développée dans [64]. Nous détaillons davantage cette méthode que celles des Sections précédentes, afin de permettre au lecteur de comprendre la différence entre cette méthode et celle utilisée dans cette thèse.

Cette modélisation comprend donc un attracteur non-linéaire, qui permet à la trajectoire d’atteindre la position désirée et qui assure ainsi une stabilité asymptotique²⁷ ; ainsi qu’une fonction f (appelée “*forcing term*”), qui contient l’information sur la forme globale de la trajectoire. Il s’agit du modèle dynamique de second ordre suivant :

$$\ddot{y} = \alpha_y(\beta_y(g - y) - \dot{y}) + f.$$

27. Un système est dit asymptotiquement stable si lorsqu’on lui applique une entrée finie, la sortie ne va pas diverger et que si on applique un échelon en entrée du système, alors toutes oscillations seront amorties et la sortie tendra de manière asymptotique vers une valeur stationnaire finale. De plus, si on applique une impulsion de Dirac à l’entrée du système, alors les oscillations produites à la sortie vont s’atténuer rapidement et le système retrouvera son état stationnaire précédent. [Wikipédia]

Avec g l'état but à atteindre ; y l'état du système ; \dot{y} la vitesse du système ; et finalement α_y et β_y des "termes de gains" qui correspondent à des constantes qui ajustent l'importance à donner à \ddot{y} et à f .

Dans [103], Hoffmann *et al.* ajoutent un terme à la modélisation afin que les trajectoires générées puissent éviter des obstacles. La modélisation devient alors :

$$\ddot{y} = \alpha_y(\beta_y(g - y) - \dot{y}) + f + p(y, \dot{y}).$$

Où $p(y, \dot{y})$ implémente la dynamique d'évitement d'obstacle.

Lorsqu'il s'agit de contrôle moteur, les primitives de mouvement qu'apprennent les robots portent parfois le nom de Primitives de Moteur Dynamique [109]. Dans [6], le robot commence par apprendre une primitive d'interaction en regardant deux êtres humains effectuer eux-mêmes la tâche interactive, à l'aide de capture du mouvement. Cette primitive d'interaction contient les dépendances entre les deux mouvements humains. Puis, le robot utilise des primitives d'interaction afin d'adapter son comportement aux mouvements de son partenaire. Cette méthode consiste à apprendre une distribution des paramètres des Primitives de Moteur Dynamique.

Dans [160], Mulling *et al.* utilisent ces Primitives de Moteur Dynamique afin de permettre au robot d'apprendre un ensemble de mouvements élémentaires de tennis, en se basant sur un apprentissage kinesthésique. Puis, ils utilisent une Combinaison de Primitives Motrices, afin de généraliser les mouvements appris à un ensemble plus large de situations.

Une méthode intéressante [63], nommée Primitive de Mouvement Flexible (CMP), permet au robot d'apprendre à la fois la trajectoire Cartésienne qu'il doit suivre et le contrôle moteur qui lui correspond, afin que ces mouvements soient rigides (et donc très précis) en ce qui concerne les mouvements dirigés vers l'action voulue, tout en restant flexibles dans les autres directions.

Pour cela, la méthode utilise à la fois des DMPs et des Primitives de Couples (TP). Le bras robotique apprend alors tout d'abord différentes trajectoires Cartésiennes, par apprentissage kinesthésique à l'aide de la méthode DMPs.

Puis, chacune de ces trajectoires exemples est exécutée par le bras robotique, à l'aide d'un contrôleur rétroactif dit "rigide". Cette rigidité permet au robot de suivre la trajectoire de manière précise, mais nécessite que l'environnement du robot soit contrôlé pour éviter des accidents. De ces trajectoires Cartésiennes, le robot apprend alors les "trajectoires" des couples correspondantes, utilisées lors du contrôle des trajectoires Cartésiennes : c'est ainsi qu'il apprend les Primitives de Couples. À la suite, le robot rejoue ces trajectoires Cartésiennes selon des contraintes spécifiques $C_{\tau k}$, tels que des contraintes sur la vitesse des trajectoires, sur la flexibilité souhaitée du robot (et ses couples correspondants), ou encore sur le poids des objets portés par le robot. Le robot apprend alors, pour chaque Primitive de Couple, les contraintes C_{τ} sous lesquelles elles ont été apprises. Ces Primitives de Couples représentent les trajectoires de couple en tant que combinaison linéaire de fonction de base radiale. Finalement, une Primitive de Mouvement Flexible correspond à l'ensemble des données et paramètres du mouvement robotique : avec les paramètres de la DMP qui permettent d'encoder la trajectoire Cartésienne ; les paramètres de la Primitive de Couple ; la durée de la trajectoire v_k ; et la contrainte de trajectoire c_k . Ainsi, une Primitive de Mouvement Flexible correspond à :

$$H_x^{CMP} = \{param\{DMP_k\}, param\{TP_k\}, v_k, c_k\}, k = \{1, \dots, N \cdot M\}.$$

Avec " $N \cdot M$ " le nombre d'exemples. Ensuite, le robot apprend à retrouver les paramètres $\{param\{DMP_k\} ; param\{TP_k\} ; v_k\}$ en fonction de la contrainte demandée c_k , grâce à un apprentissage probabiliste. Le robot peut ainsi effectuer convenablement ses tâches, en respectant

les contraintes imposées et ce, même lorsque ces contraintes n’ont pas été apprises lors de l’apprentissage.

Finalement, l’étude présentée dans [150], appelée Primitives de mouvements probabilistes dynamiques (*Probabilistic Dynamic Movement Primitives*), permet d’ajouter aux DMPs des propriétés probabilistes. Grâce à ces propriétés probabilistes, cette méthode pourrait aussi permettre de prédire la poursuite de trajectoire initiée à l’aide de calculs probabilistes à partir des mesures des capteurs. Cependant, cette méthode comporte des inconvénients présentés dans la Section 1.7.2. C’est pourquoi, nous utilisons une autre méthode, présentée ci-dessous.

Primitives de Mouvement Probabiliste (ProMPs)

Dans les trois études de cette thèse, présentées dans les Chapitres 4, 5 et 6, nous utilisons la méthode de Primitives de Mouvement Probabiliste (*Probabilistic Movement Primitives*, ProMPs) [171, 170]). Il s’agit d’une approche qui nous permet d’effectuer de la prédiction et de permettre au robot d’adapter ses mouvements à la volonté de l’utilisateur.

Cette méthode comporte de nombreux avantages, telles que la co-activation et le couplage de primitives, ainsi que le redimensionnement temporel. Dans notre cas, nous sommes notamment intéressés par les capacités suivantes :

- Le conditionnement de trajectoires, afin qu’elles passent par des positions spécifiques. Dans nos études, cette propriété nous permet d’adapter les mouvements du robot, selon la volonté du partenaire.
- L’apprentissage des actions que le robot doit effectuer, en supplément de la trajectoire à suivre (par exemple, il pourrait apprendre à varier la rigidité de son mouvement ; ou à tourner sa tête en direction de la position finale que doit atteindre sa main ; etc.).
- La corrélation d’informations provenant des différentes modalités qui représentent les trajectoires. Dans certaines de nos études, le robot ne connaît qu’une sous partie des mesures du modèle lors de l’inférence (qui proviennent par exemple d’un seul type de capteur). Or, grâce à cette propriété, le robot est capable de calculer une estimation des autres mesures. Nous verrons que cette propriété n’est pas utilisable lorsqu’il s’agit d’information visuelle, celle-ci n’étant pas suffisamment précise pour améliorer la trajectoire Cartésienne du robot.
- L’inférence du mouvement et la reconnaissance de sa vitesse. Grâce à ces propriétés, notre robot peut notamment reconnaître un mouvement même lorsque la vitesse de ses mouvements est variable.

L’explication de cette méthode est développée dans la Section 3.3.

Les ProMPs ont déjà été utilisés lors d’étude concernant la coordination humain-robot, afin de générer des trajectoires robotiques appropriées en réponse à des trajectoires humaines initiées [144]. Dans [143] Maeda *et al.* proposent une amélioration d’un logiciel qui permet au robot d’apprendre des primitives d’interaction, en remplaçant l’utilisation de DMPs par des ProMPs. Le robot apprend alors par imitation à collaborer avec l’humain sans que celui-ci n’ait à le programmer. Les expériences montrent alors qu’avec l’utilisation de ProMPs, la collaboration est plus performante. Dans [172], Paraschos *et al.* utilisent des ProMPs pour encoder à la fois les trajectoires et les signaux de forces et de couples. Ainsi, leur robot apprend la dynamique de ses mouvements, sans avoir besoin de modèle dynamique et évite ainsi les problèmes des modèles comme les DMPs. Dans [77, 145], le robot apprend un ensemble de tâches et les primitives de mouvement correspondantes, sans avoir besoin d’étiqueter en amont ces tâches, à l’aide de Modèles de Combinaison de Gaussiennes (GMMs) et de ProMPs. Un logiciel de “Mélange de ProMPs d’Interaction” a ainsi été développé et validé pour des tâches de collaboration et d’assistance avec

l'humain.

Dans [74], la méthode développée basée sur ces ProMPs permet de répondre aux problèmes de données incomplètes (trajectoires cachées, mesures incorrectes ou corrompues, etc.) et de variabilité dans la vitesse des trajectoires, en se basant sur les propriétés des ProMPs et sur l'algorithme Expectation-Maximisation.

1.7.2 Justification du choix de modélisation

En comparaison avec les techniques précédemment présentées, la méthode des ProMPs nous semble être la plus prometteuse afin de permettre la reconnaissance de l'intention et l'anticipation de mouvement pour des scénarios de collaboration humain-robot. En effet, cette façon de modéliser les mouvements permet de capturer la variabilité des mouvements humains, lorsque ceux-ci effectuent une même trajectoire et ce, sans avoir de contrainte sur le "but" à atteindre, comme c'est le cas des DMPs. Cette variabilité permet au robot de passer par des positions spécifiques, tout en restant dans la distribution de mouvements apprise, et en conservant ainsi les caractéristiques qui définissent la trajectoire.

Comparaison ProMPs-DMPs

Les méthodes ProMPs et DMPs ont été comparées dans différentes études, notamment lors de l'apprentissage de primitives de mouvements interactives [76] [171] [143]. Nous résumons ici les similitudes et différences entre ces deux méthodes. Une propriété spécifique de la méthode DMP est qu'à la fin des mouvements, il n'y a plus que l'attracteur dynamique qui est activé, ce qui à la fois assure et oblige le robot d'atteindre la position attractive.

Il s'agit donc d'apprentissage de trajectoires dirigées vers un but.

Dans nos études, nous souhaitons au contraire que le robot puisse inférer précisément qu'elle est la position finale désirée par son partenaire, en fonction du mouvement initié par celui-ci (ainsi, cette position finale doit être variable). La position finale ne doit donc pas être une caractéristique de la trajectoire apprise. C'est pourquoi, la méthode des ProMPs correspond davantage aux études de cette thèse. Cette analyse théorique est validée par l'étude [171] qui montre que, si l'on force la trajectoire à passer par une autre position but à l'aide de la méthode ProMP, la trajectoire reste incluse dans distribution apprise, tandis qu'avec les DMPs, les nouveaux mouvements sortent de cette distribution et ont donc un mouvement non contrôlé.

Ces deux méthodes ont des propriétés communes : redimensionnement temporel ; apprentissage possible à partir d'une unique démonstration ; généralisation avec l'augmentation du nombre de démonstrations ; etc. Avec l'extension de DMPs en PDMPs (*c.f.*, Section 1.7.1), ces deux méthodes permettent aussi l'inférence de la finition d'une trajectoire initiée, grâce à l'apprentissage de distribution.

Cependant, ces PDMPs permettent moins de généraliser les mouvements, puisque le modèle utilisé force les mouvements à toujours être dirigé vers un but. De plus, en utilisant les PDMPs, les trajectoires peuvent dévier arbitrairement en dehors du champ des démonstrations. Cette limitation peut être critique dans beaucoup d'applications (notamment dans notre cas puisqu'il s'agit d'un robot humanoïde interagissant avec un être humain) puisque les incertitudes et les perturbations sur les données peuvent arriver fréquemment et provoquer un accident avec le robot (collision, déséquilibre du robot, etc.).

En outre, la méthode ProMP comporte des propriétés supplémentaires. Par exemple, elle encapsule l'information de corrélation entre les différentes entrées du modèle ou encore, elle

permet de co-activer différentes trajectoires. Ces propriétés seront utiles lors de nos études (*c.f.*, Section 1.7.1).

D'autre part, la méthode ProMPs est plus performante lorsque les données reçues par les capteurs sont bruitées ou variables, ou globalement dans le cadre d'apprentissage de primitives d'interaction et de prédiction [143]. Elle est aussi plus efficace en ce qui concerne les tâches collaboratives. En effet, dans [143], une comparaison entre ces deux méthodes montre que l'erreur quadratique moyenne des prédictions est inférieure lorsque la méthode ProMPs est utilisée.

Comparaison ProMPs-HMMs (Modèles de Markov Cachés)

Les modèles graphiques ont de bons résultats concernant la reconnaissance d'actions, mais sont moins performants en ce qui concerne la généralisation des mouvements. Par exemple dans l'étude [207], l'utilisation d'un HMM à état discret permet de reconnaître les actions, mais pas de générer les trajectoires associées correctement. Dans notre cas, nous ne pouvons nous permettre de telles anomalies, puisque l'on s'intéresse avant tout à la trajectoire qu'effectue le robot. L'étude présentée précédemment propose cependant un type de HMMs non paramétrique à états continus, ce qui peut être une piste pour l'utilisation de HMM dans l'optique de générer des mouvements. D'autres études combinent alors les méthodes HMM et DMP pour pouvoir tout d'abord reconnaître le label du mouvement, puis de générer le mouvement correspondant. Dans notre cas, l'utilisation de ProMPs nous permet d'effectuer ces deux étapes en même temps.

De plus, tout comme les réseaux de neurones, l'apprentissage de HMMs demande un ensemble de trajectoires d'apprentissage qui grossit rapidement avec l'augmentation de la dimension des données. Or, puisque notre méthode doit pouvoir se généraliser à un robot humanoïde composé de beaucoup de degrés de libertés, cette méthode n'est pas optimale. Au contraire, en utilisant les ProMPs, le robot peut déjà générer une trajectoire correcte dès la première observation. Les autres observations lui permettent alors d'affiner sa connaissance de la primitive du mouvement.

De plus, les HMMs ne permettent pas la coordination de mouvements entre diverses modalités ou entre les mouvements du partenaire et ceux du robot, comme cela a été fait dans [143, 77].

Comparaison ProMPs-BLRs (Régression Linéaires Bayésiennes classiques)

Les modèles de régressions linéaires permettent d'établir une relation linéaire entre une variable et une ou plusieurs autres. Dans notre cas, il s'agit d'expliquer la variable ξ en fonction du temps t . Ces modèles sont estimés par différentes méthodes, telles que la méthode du maximum de vraisemblance, la méthode des moindres carrés, la méthode des moments, ou encore par des méthodes Bayésiennes.

Les ProMPs correspondent à un type particulier de Régression Linéaire Bayésienne (RLB), spécifiques aux trajectoires : on ne s'intéresse plus à des valeurs instantanées, mais à un ensemble de valeurs continues au cours du temps. Pour ce faire, les ProMPs sont exprimées comme une somme pondérée de fonctions de base Gaussienne, étalées le long de la durée de la trajectoire. Les fonctions de base Gaussienne permettent en effet de représenter des variations non brutales, régulières au cours du temps, ce qui correspond à des variations de trajectoires naturelles. Dit de manière plus mathématique, dans le cas des ProMPs, on ne s'intéresse pas à la valeur de ξ en fonction du temps t (c'est-à-dire, $p(\xi(t)|t)$), mais à la forme globale de la trajectoire $\Xi = \{\xi(1), \dots, \xi(t_f)\}$ en fonction des premières itérations $\Xi^o = \{\xi_1, \dots, \xi_{n_o}\}$ (c'est-à-dire, $p(\Xi|\Xi^o)$). De plus, puisqu'il ne s'agit pas de représenter spécifiquement ξ en fonction du temps t , car les trajectoires peuvent avoir des durées différentes. C'est pourquoi, les fonctions de base Gaussienne sont étalées sur la

durée des trajectoires puis ré-échantillonnées en une taille standard à l'aide d'une variable α (c.f., 3.3 pour plus de détails).

La table 1.1 représente mathématiquement les différences principales de ces deux méthodes.

Name	RLB	ProMPs
Modèle	$\xi(t) = \Phi_t \omega + \epsilon$	$\Xi = \Phi_{\alpha[1:t_f]} \omega + \epsilon$
Apprentissage	$p(\omega t, \xi(t))$ $\forall t \in \{t_{1,1}, \dots, t_{f,N}\}$	$p(\omega \Xi_i)$ $\forall i \in \{\Xi_1, \dots, \Xi_N\}$
Mise à jour	$p(\omega t, \xi(t))$	$p(\omega \Xi^o)$
Ce que l'on cherche	$p(\xi(t) \omega, t)$	$p(\Xi \omega, \Xi^o)$

TABLEAU 1.1 – Différence entre l'utilisation des méthodes ProMPs et RLB.

Comparaison ProMPs - GP (Processus Gaussien)

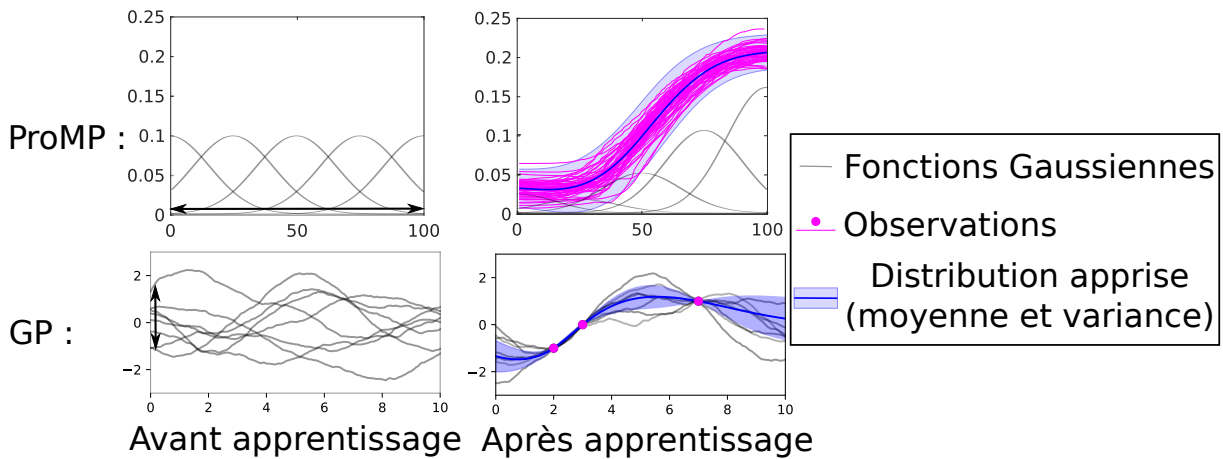


FIGURE 1.5 – Différence entre les modèles GP et ProMP.

Tout d'abord, voici un petit résumé de ce que sont les Processus Gaussien, dont la méthode est détaillée dans [182].

Les GPs sont des modèles bayésiens non paramétriques, utilisés essentiellement pour modéliser des données ou pour effectuer de la prédiction. La formule de cette méthode est : $\xi = f(t) + \epsilon$, avec $f(t) = t^\top \omega$.

Ils permettent de représenter un nombre infiniment grand de variables aléatoires en récupérant la distribution conjointe de toutes ces variables. Il s'agit alors, d'une distribution de fonctions à valeur dans un espace continu, comme le temps. Pour comprendre un peu mieux, dans le cas d'une distribution gaussienne classique, cette dernière est décrite par une moyenne de vecteur μ et une matrice de covariance Σ . Dans le cas de GPs, la distribution multivariée calculée est représentée par une moyenne de fonction $m(t)$ et une matrice de covariance de fonction $cov(f(t), f(t')) = k(t, t')$, définie par : $f(t) \sim GP(m(t), k(t, t'))$.

Pour initialiser ce modèle, une distribution *a priori* est posée sur les paramètres de f : $\omega \sim \mathcal{N}(0, \Sigma_p)$. Le graphique en bas à gauche de la Figure 1.5, représente un exemple de fonctions d'échantillons générées de manière aléatoire à partir de cette distribution *a priori*. Notons que ces fonctions sont effectivement centrées en 0.

Avec des données de démonstrations, représentées en rose sur le graphique du bas à droite, cette distribution est mise à jour. Sur cet exemple, cette mise à jour est effectuée de telle sorte que les fonctions générées à partir de cette nouvelle distribution, représentée en bleu sur ce même graphique, passent toutes par ces points. En effet, la fonction moyenne passe par tous ces points et la variance entre les fonctions générées est nulle en ces points. Cela est dû au choix de la matrice de covariance de ces fonctions.

Les différences principales entre les GPs et ProMPs sont présentées par la Figure 1.5. Dans le cas des GPs, on a vu qu'il s'agit d'une distribution de fonctions qui correspond aux données. Ainsi, à l'aide de cette distribution, on peut générer des fonctions $f(t)$ qui correspondent aux données.

Dans le cas des ProMPs, il s'agit de calculer une fonction de probabilité. Dans le cas des GPs, il s'agit de calculer une distribution de fonctions. Initialement, la distribution des fonctions des GPs est centrée en 0 avec une certaine variabilité, dans la dimension des variables non connues ξ . Dans le cas des ProMPs, il s'agit d'un ensemble de fonctions, qui sont étalées sur la dimension des variables connues et continues t .

Intéressons nous maintenant à la mise à jour. Dans le cas des GPs, pour chaque nouvelle donnée de démonstration $\{\xi_i, t_i\}$, les paramètres de la distribution de fonctions sont mis à jour, de telle sorte que les fonctions comprises dans cette distribution passent par²⁸ ces données de démonstration. Dans le cas des ProMPs, pour chaque nouvelle donnée de démonstration $\{\xi_i, t_i\}$, les paramètres de toutes les fonctions de base sont mis à jour, de telle sorte qu'ensemble, elles représentent de la meilleure façon les données.

De plus, tout comme dans le cas des RLBs, les GPs sont généralement utilisés pour connaître des valeurs instantanées et non des trajectoires. Par exemple, dans l'étude [201], Shon *et al.* proposent d'utiliser des GPs afin d'encoder et de décoder des postures dans un espace latent. La spécificité de cette étude est que les espaces d'entrée et de sortie (c'est-à-dire, avant encodage et après décodage) ne sont pas identiques : l'entrée correspond à une posture humaine, tandis que la sortie correspond à la posture du robot. De ce fait, le robot peut reproduire les postures effectuées par une personne.

Néanmoins, en combinant les *GPs* avec d'autres méthodes, l'apprentissage de mouvement a pu être amélioré. Par exemple, dans [79], Fanger *et al.* les utilisent afin de définir la fonction de non-linéarité f incluse dans le modèle DMPs afin de représenter des trajectoires. De plus, l'utilisation des GPs permet dans cette étude d'avoir une mesure de confiance à chaque phase du mouvement robotique. De même, dans [46], Chatzilygeroudis *et al.* utilisent les *GPs* afin d'approximer, au sein du modèle de la dynamique des mouvements d'un robot, les dynamiques inconnues et non attendues.

1.7.3 Durée des Mouvements

Un problème sous-jacent à la reconnaissance de mouvement consiste en l'estimation de la durée de la trajectoire, ce qui est nécessaire afin d'aligner la trajectoire courante avec celles apprises. Dans notre cas, au début de l'interaction physique entre l'humain et le robot, ce dernier observe un geste partiel guidé par l'utilisateur. Le robot doit alors estimer l'état courant du mouvement, afin de pouvoir reconnaître quel mouvement son partenaire souhaite qu'il effectue. Ainsi, il doit pouvoir estimer la vitesse du mouvement partiel.

28. Ou se rapprochent, selon le choix de la matrice de covariance.

D'un point de vue physique, la loi de Fitt modélise la durée de trajectoires dirigées vers un but. Pour avoir une idée d'à quoi ressemble cette modélisation, voici la "formulation de Shannon" d'un mouvement humain en une dimension :

$$T = a + b \log_2 \left(1 + \frac{D}{L}\right).$$

Dans cette formule, T est le temps moyen du mouvement, a et b des paramètres déterminés empiriquement, D la distance séparant le début de la trajectoire, L la largeur de la zone (ou cible) à atteindre et S son centre. Cette modélisation suppose que la durée du mouvement dépend linéairement de la difficulté à atteindre la cible [83]. Dans [138], Langolf *et al.* montrent qu'en modifiant la taille de la cible, la forme du mouvement change. Ainsi, la loi de Fitt est imprécise lorsque la taille de la cible change. Cette idée est vérifiée dans les études [138] et [205] puisqu'ils montrent que le mouvement change selon la précision à laquelle l'humain doit atteindre la position cible.

D'un point de vue algorithmique, la méthode généralement utilisée est l'alignement temporel dynamique (Dynamics Time Warping, DTW), qui consiste à trouver la corrélation entre deux trajectoires dont la durée diffère, de manière plus robuste que la méthode de la distance euclidienne. Dans [6], Amor *et al.* adaptent cet algorithme afin de faire coïncider le mouvement partiel avec un mouvement de référence. Beaucoup d'améliorations ont été effectuées sur cette méthode. Par exemple, dans [125], Keogh *et al.* proposent une méthode robuste afin d'améliorer l'indexation. De plus, différentes méthodes ont été développées pour accélérer la vitesse de calcul, comme les méthodes *Fast Dynamic Time Wrapping*, *Lucky Time Warping* ou encore *Fast Time Wrapping* qui sont présentées et comparées dans l'étude [203]. Dans cette étude, Silva *et al.* présentent aussi leur propre méthode d'accélération de la vitesse de calcul, nommée *Couverture de Chemins Élagués*. Comme l'exprime son nom, cette méthode permet de supprimer les données improbables.

Bien que couramment utilisée, la méthode DTW a l'inconvénient de ne pas conserver la forme globale de la trajectoire.

D'autres algorithmes, plus spécifiques aux ProMPs, ont aussi été développés. Par exemple, la méthode utilisée dans l'étude [76] permet de modéliser localement la variabilité de la vitesse d'exécution des trajectoires. Pour cela, en plus de mettre à jour la ProMP à chaque itération, l'algorithme met à jour un paramètre de modulation du temps. Pour ce faire, la méthode utilise l'algorithme d'*Espérance-Maximisation* optimisant ainsi les paramètres de la ProMP et le paramètre de modulation du temps pour que la distribution corresponde au mieux à la trajectoire observée. Dans [143], Maeda *et al.* améliorent l'estimation de la durée des mouvements reconnus à l'aide de ProMPs, en utilisant une autre méthode d'alignement temporel. Cette méthode utilise un "Modèle de Mélange de Gaussiennes" afin de représenter une fonction d'alignement temporel et au lieu d'utiliser DTW, ils forcent l'alignement des deux courbes localement, sans avoir à sauter certains index. Par conséquent, ils obtiennent des trajectoires plus réalistes, plus régulières et surtout leur forme globale est préservée. Cette méthode est détaillée de manière plus précise dans [144]. Ces méthodes seront implémentées dans les futures études.

Dans notre cas, nous utiliserons l'*information des forces* reçues par le robot afin d'*estimer la vitesse de la trajectoire* (c.f., Section 3.3.4 et Section 4.3.1), puisque les forces et les vitesses sont théoriquement liées par la formule dynamique de notre robot :

$$\tau_{int} + \tau_{ext} = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + J^T(q) \cdot f_{ext}. \quad (1.1)$$

Avec τ_{int} les couples effectués par le robot. Dans notre cas, ces couples sont nuls, puisque le contrôle du robot est en mode "zéros torques"²⁹ tant que le mouvement est initié par l'utilisateur ;

29. c'est-à-dire, le robot n'exerce aucune force

τ_{ext} , les couples extérieurs, contenant donc les forces ; $M(q)$, la matrice de masse inertielle du robot ; $C(q, \dot{q})$, les forces de Coriolis et centrifuges ; $g(q)$ le vecteur de gravité ; $J(q)$ la Jacobienne et f_{ext} , les forces extérieures exercées sur les articulations du robot.

1.7.4 Logiciels Open Source Connexes

L'un des apports de cette thèse consiste en la création et l'explication d'un logiciel open source pour le robot *iCub* réel ou en simulation et qui peut être adapté à d'autres robots. Ce logiciel utilise la méthode ProMP afin de permettre au robot de reconnaître l'intention de l'utilisateur durant une tâche de collaboration et ainsi, de lui permettre d'effectuer les actions initiées de manière autonome et adaptative.

La Table 1.2 contient d'autres logiciels open source utilisés pour apprendre des primitives de mouvements. Certains d'entre eux ont notamment permis de développer des applications d'apprentissage avec le robot *iCub* (*par ex.*, [140, 209]) ou de permettre la reconnaissance de l'intention humaine. Cependant, le logiciel que nous proposons ici est différent. En effet, il fournit une implémentation de la méthode ProMPs spécifique à la reconnaissance de l'intention et la prédiction des trajectoires attendues par le partenaire. De plus, ce logiciel s'intéresse au cas spécifique d'interaction physique et/ou visuelle entre le robot et l'utilisateur. Le robot peut ainsi effectuer n'importe quel mouvement désiré par l'utilisateur, et ce, sans que l'utilisateur n'ait besoin de savoir programmer (apprentissage et manipulation du robot par interaction physique ou visuelle).

Il s'agit d'une première étape pour permettre au robot *iCub* de développer des compétences de reconnaissance de l'intention.

1.8 Compression dans un espace latent

La réduction de la dimension de données est un processus critique dans l'apprentissage automatique et le traitement des données. Il s'agit d'extraire des données un ensemble réduit de caractéristiques principales qui doivent suffire à décrire le processus.

1.8.1 Données indépendantes du temps

En général, les algorithmes développés ont pour but de compresser des données instantanées telles que des images, des lettres, etc.

Pour cela, une première méthode couramment utilisée se nomme Analyse en Composantes Principales (ACP) [219]. Il s'agit d'une méthode mathématique qui a été décrite de deux manières. Premièrement, dans [105] les ACPs sont définies comme la projection orthogonale des données sur un espace linéaire de dimension inférieur, en maximisant la variance de ces données projetées (c'est-à-dire, que le maximum d'information est conservé). Deuxièmement, dans [178], les ACPs sont définies comme la projection linéaire qui minimise l'erreur moyenne de projection (c'est-à-dire, la distance entre les données et leur projection). Cette méthode consiste tout d'abord à calculer la moyenne des données \bar{x} et leur covariance S , puis de calculer les M vecteurs propres correspondant aux M valeurs propres les plus grandes, afin d'avoir un espace latent de dimension M représentatif des données originales. Une amélioration de cette méthode, consiste en une formulation probabiliste de ces ACPs (PPCA) [219]. Dans ce cas, il s'agit cette fois de calculer les solutions du maximum de vraisemblance du modèle cette fois-ci probabiliste de variables latentes. Cette amélioration permet par exemple : de réduire le nombre de paramètres ; à l'aide

Logiciel / librairie	Méthode	Logiciel	Language	Robot	Réf.
Dynamical System Modulation for Robot Adaptive Learning via Kinesthetic Demonstrations	GMR	[100]	Matlab	Hoap3	[157]
pbdlib-matlab	HMM, GMM and others	[39]	Matlab	Baxter	[35]
DMP learning with GMR	DMP, GMR	[36]	Matlab, C	Coman	[38]
Stochastic Machine Learning Toolbox	Kernel Functions, GPs, and Bayesian Optimization	[139]	C++ Python		
pydmps	DMP	[65]	Python	Sarcos	[109]
Dynamical Systems approach to Learn Robot Motions	GMM, SEDS	[126]	Matlab	iCub	[127] [128]
Function Approximation, DMP, and Black-Box Optimization	DMP	[210]	Python C++	iCub	[140] [209]
Learning Motor Skills from Partially Observed Movements Executed at Different Speeds	ProMP	[75]	Matlab Python		[76]
icub Learning Trajectories + With multimodal information	ProMP	[soft.Pred.Mult] [soft.ProMP.Mult]	Matlab C++	iCub	[D.P] [D.M]
prediction of Whole-Body movements	AE-ProMP	[soft.WB.Pred]	Matlab Python		[D.W]

TABLEAU 1.2 – Logiciels et bibliothèques open source qui implémentent des Primitives de Mouvements et leur application. En gras, les logiciels que nous avons développés.

de l'algorithme espérance-maximisation, d'éviter de calculer la matrice de covariance à chaque étape; ou encore de fonctionner lorsque certaines données sont manquantes.

Les limites principales des ACPs sont qu'il s'agit uniquement de compression linaires des données et que ces méthodes sont coûteuses en calcul.

Une autre méthode classique correspond aux *auto-encodeurs* (AEs). Puisque la dernière étude de cette thèse s'inspire de cette méthode, nous précisons ici les variantes les plus connues de celle-ci.

Il s'agit de réseaux neuronaux qui apprennent à encoder des données dans un espace latent d'une dimension inférieure à l'espace d'entrée original, en minimisant une fonction objectif (*loss function*). Cette fonction objectif mesure la distance entre les données originales et les données reconstruites depuis cet espace latent. Le haut de la Figure 1.6 présente un tel réseau de neurones.

Basés sur les AEs, plusieurs variantes ont été créés, dont les deux plus connues sont présentées dans cette même Figure et explicitées ci-dessous.

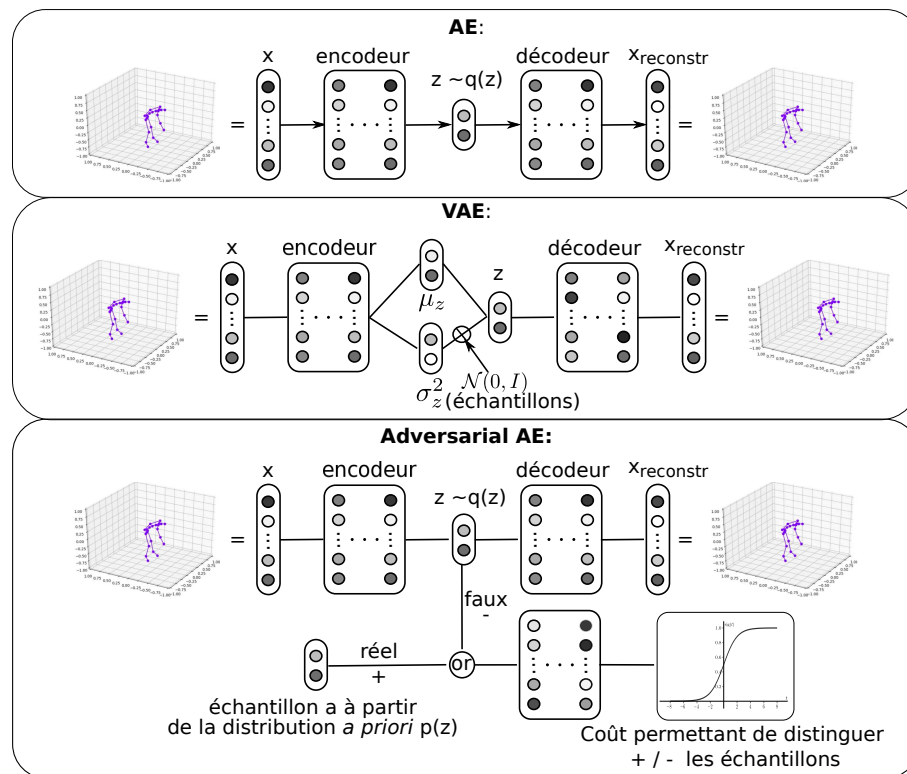


FIGURE 1.6 – Différentes méthodes basées sur les autoencodeurs.

Les réseaux auto-encodeurs adversaires (Adversarial AutoEncoders, AAE) [97] correspondent à une autre manière d'apprendre à coder/décoder des données, mais aussi de détecter lorsque des données compressées sont réelles ou générées. Il s'agit d'un ensemble de deux réseaux en compétition. Le premier, le réseau discriminant, apprend à reconnaître si les données fournies en entrées ont été créées ou si elles correspondent à des données réelles. Le second, le réseau génératif, apprend à générer des données analogues aux données de démonstration. L'apprentissage se fait en deux étapes. Pendant la phase de reconstruction, seul le premier réseau est entraîné afin de minimiser l'erreur de reconstruction (différence entre les données réelles et reconstruites). Pendant la phase de régularisation, les deux réseaux sont entraînés, sans mettre à jour le décodeur : premièrement, le réseau discriminant apprend à classifier les données qu'il reçoit en entrée (données "réelles", suivant la distribution voulue, ou générées) ; deuxièmement, l'encodeur du réseau génératif est de nouveau entraîné afin de suivre la distribution désirée en cherchant à "tromper" le réseau discriminant afin que les données générées soient considérées comme "réelles". Cette variante ne permet cependant pas d'apprendre les caractéristiques spécifiques aux données à apprendre, puisqu'il cherche uniquement à "ressembler" aux données réelles. Par exemple, si on lui fournit des données représentant différents objets, il va générer des objets qui semblent réels en général et va ainsi mélanger les caractéristiques des différents objets.

Les auto-encodeurs variationnels [133] (VAEs) répondent davantage à cette problématique. Il s'agit d'ajouter à l'auto-encodeur de l'inférence variationnelle [20]. L'inférence variationnelle permet d'approximer une fonction de densité de probabilité en optimisant un paramètre d'une fonction de densité de probabilité connue (par exemple, les distributions gaussiennes). Ces deux variantes sont d'excellents approximateurs fonctionnels et ont été utilisées pour réduire la dimension de fonctions complexes de manière non supervisée [67, 70]. Une vulgarisation de la

méthode des VAEs est fournie dans [67].

Ces méthodes permettent uniquement de représenter des données fixes, sans dépendance dans le temps. La Section suivante présente les méthodes de compression de données continues dans le temps, telles que des trajectoires.

1.8.2 Données avec dépendance temporelle

Dans cette thèse, la compression de données consistera aussi en la compression de données séquentielles, afin de représenter des trajectoires.

Dans [50], Colomé *et al.* présentent une nouvelle méthode qui réduit la dimensionnalité des ProMPs, appelée DR-ProMP, pour trouver des stratégies de marche de faible dimension pour le robot Nao. Dans cette étude, ils ont utilisé des techniques probabilistes de réduction de dimension appliquées sur un ensemble de trajectoires de démonstrations afin d'extraire les synergies inconnues entre les dimensions. Cela aboutit à une nouvelle expression de ProMPs dans laquelle une matrice de coordination permet de transiter de l'espace articulaire virtuel latent à faible dimension, à l'espace articulaire réel du robot. De plus, l'information sur la variance de chaque trajectoire à chaque instant de la trajectoire est utilisée afin de permettre de déformer plus ou moins celles-ci. Dans leur étude, la dimension de l'espace latent y est réglée manuellement (généralement égale à 4 ou 5). La taille maximale de l'espace d'origine testée dans cet article est de 15, ce qui laisse présager que le taux de compression est insuffisant pour notre application présentée dans le Chapitre 6. En effet, dans cette étude, les données de l'espace d'origine sont de dimension 69.

Alors que les AEs et les VAEs peuvent être utilisés afin d'apprendre à reconstituer des données indépendantes (par exemple, des postures humaines), ils ne peuvent pas être utilisés pour représenter des données dépendantes dans le temps (par exemple, des trajectoires posturales) au fil du temps, de manière lisse et cohérente (sans à-coups). En effet, ils n'ont pas d'information concernant la dépendance posturale au cours du temps.

Cette question a été bien expliquée par Chen *et al.* dans [48]. Ceux-ci proposent alors une méthode qui force la dépendance temporelle en apprenant des DMPs dans l'espace latent. Cette méthode se nomme VAE-DMP et utilise des Filtres Variationnels Bayésiens profonds (DVBF) [124]. Un filtrage bayésien est alors appliqué sur les variables latentes qui ont une dépendance temporelle. Pour cela, un réseau neuronal profond récurrent composé de VAEs est utilisé.

Puis, une amélioration de VAE-DMP, appelée *Extracteur de caractéristiques de séries temporelles variationnelles* (VTSFE), a été proposée par Chaveroche *et al.* dans [47] pour coder les caractéristiques de la série temporelle à des fins de classification et de prédiction.

Ces méthodes semblent intéressantes lorsque l'on souhaite mapper des trajectoires à grande dimension dans des espaces à plus petite dimension. Cependant, avec ces deux méthodes, lorsque les trajectoires sont décompressées, elles deviennent imprécises. C'est pourquoi il serait nécessaire d'appliquer les améliorations proposées à la fin de l'article [47]. Dans cette thèse, à la fin de l'étude du Chapitre 6, la méthode VTSFE est cependant combinée avec la méthode ProMP, afin de pouvoir prédire des trajectoires depuis un espace réduit. Le but est alors d'ouvrir des perspectives de recherche, car lorsque les problèmes liés au décodeur seront résolus, la méthode VTSFE-ProMP devrait avoir une meilleure qualité de compression que notre méthode AE-ProMP.

2

Matériels

Dans ce chapitre, nous présentons les différents matériels utilisés dans cette thèse. Les deux premières études se consacrant au développement des capacités du robot iCub, nous présentons tout d’abord ce robot dans la Section 2.1. Puis, la Section 2.2 présente les différents logiciels utilisés pour contrôler ce robot : il s’agit de la simulation du robot dans Gazebo, l’interface graphique iCubGui représentant son état, ou encore le logiciel YARP, qui permet la communication entre les différents logiciels.

La Section 2.3 présente ensuite les autres matériels que nous avons intégrés ou développés afin d’améliorer les capacités de ce robot. Il s’agit du logiciel Intraface, qui détecte l’orientation faciale de la personne en face de la caméra ; et du Geomagic Touch, l’appareil haptique permettant de guider “à la main” le robot simulé.

La dernière étude se consacre au développement d’une méthode permettant la prédiction de trajectoire corps-complet. Pour cela, l’utilisateur porte une combinaison, nommée XSENS, composée de capteurs afin que sa posture soit mesurée. La Section 2.4 présente cette combinaison.

Ce chapitre se conclut alors par la présentation du logiciel que nous avons créé, permettant la prédiction de trajectoires (*c.f.*, Section 2.5). Cette section donne alors un aperçu du logiciel ; précise comment sont configurés les informations du robot simulé ; puis présente comment sont récupérées les données. Pour pouvoir utiliser ce logiciel, le lecteur pourra se reporter à l’Annexe A.

2.1 Robot humanoïde iCub

Les expériences réalisées dans le cadre de cette thèse sont effectuées avec le robot humanoïde *iCub*, présenté dans la Figure 2.1. Il s’agit d’un robot de 104 cm, ce qui correspond à la taille d’un enfant de quatre ans. Sa petite taille et son poids permettent de sécuriser ses interactions avec les humains.

Ce robot a été développé lors du projet Européen *RoboCub*³⁰ et des copies existent dans plus de 20 laboratoires. Beaucoup de projets européens ont été développés sur ce robot, y compris ceux dans lequel s’inscrit cette thèse, présenté dans l’Introduction (*c.f.*, Section 6). Les articles [162, 156] présentent les différentes thématiques de recherche effectuées à l’aide de ce robot. Il s’agit majoritairement de contrôle du corps complet du robot et de robotique développementale, où la cognition du robot émerge des interactions, principalement physiques, entre lui et son environnement. Ces recherches se concentrent notamment sur l’*embodiment*

30. RoboCub est un projet financé par la Commission Européenne, Project IST-004370, dans la rubrique Strategic Objective 2.3.2.4 : Cognitive System

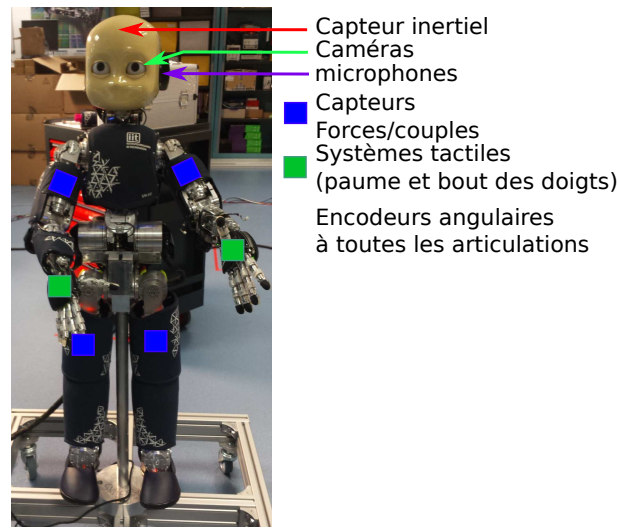


FIGURE 2.1 – Robot *iCubNancy01* sur lequel les expériences ont été effectuées.

(ou cognition incarnée), qui consiste à développer les capacités du robot à l’aide d’expériences sensorielles et de connaissance de ses positions corporelles ; sur l’exploration autonome, où le robot apprend à agir dans un environnement peu connu ; ou encore sur la capacité du robot à interagir socialement.

D’un point de vue matériel, le *iCub* est un robot humanoïde avec de riches capacités perceptuo-motrices et est composé de nombreux degrés de libertés. Le robot est effectivement composé de 53 moteurs permettant de déplacer la tête (6), les bras (7) et les mains (9), le torse (3) et les jambes. S’agissant de moteurs électriques (afin notamment de réduire la taille des moteurs), le robot est aussi composé des capteurs de couples et tactiles afin de permettre le contrôle du robot par couples impédance. De plus, le robot a un système sensoriel multimodal, tels que la vision, le son, le touché ou encore le ressenti des forces. L’ensemble de ses capteurs sont présentés dans le papier [162]. En ce qui concerne l’infrastructure logicielle du robot, elle se base sur *YARP*, présenté dans la Section 2.2, ce qui permet aux chercheurs de combiner les différents logiciels développés. Afin de permettre aux chercheurs d’étudier dans les meilleures conditions, les développements logiciels et matériels du robot sont généralement open-sources. De la même manière, les différents logiciels de simulation adaptés au robot *iCub* sont open-source, afin de permettre aux chercheurs de tester leurs méthodes sur des simulations du robot. Les programmes développés dans cette thèse seront eux aussi open-source, afin de contribuer à cette recherche partagée. La section qui suit présente ces logiciels.

2.2 Logiciels utilisés du *iCub*

Simulation du robot dans *Gazebo* *Gazebo* [134] est un simulateur de robots créé par Nate Koenig qui permet de simuler la cinématique et la dynamique de robots ou d’autres systèmes physiques. Il permet notamment de simuler les forces provenant des robots et de leur environnement. La simulation de la dynamique s’effectue à l’aide : des bibliothèques *ODE* et *DART* (*Dynamic Animation and Robotics Toolkit*), qui offrent des applications cinématiques et permettent calculer la dynamique de corps en interaction ;



FIGURE 2.2 – *iCub* simulé dans *Gazebo*.

et du modèle URDF³¹ de ces robots.

Visualisation à l'aide d'*iCubGui* *iCubGui* [5] est une interface graphique permettant à l'utilisateur de visualiser l'état du robot. Dans cette interface, le modèle cinématique du robot est affiché à l'aide des informations provenant de ses encodeurs. Elle représente aussi les mesures du capteur inertiel, et les forces exercées sur le robot. De plus, les objets visualisés par le robot peuvent être représentés dans cette interface et des objets fictifs peuvent y être ajoutés, pour par exemple donner un repère imaginaire à l'utilisateur. La Figure 2.3 représente cette interface. Sur les mains du robot, les flèches rouges représentent les forces reçues par le robot.

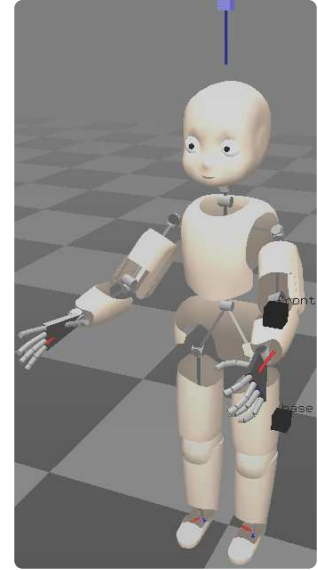


FIGURE 2.3 – Représentation du *iCub* dans *iCubGui*.

Échanges entre logiciels à l'aide de *YARP* Afin de permettre les échanges de données entre les différents programmes développés, *YARP* (*Yet Another Robot Platform*) [155] est utilisé. Il s'agit d'un *middleware* (intergiciel) open-source crée dans le but de minimiser les efforts consacrés au développement de logiciels et de maximiser la modularité des programmes.

Bien que beaucoup de chercheurs utilisent un middleware similaire nommé *ROS* (*Robot Operating System*), dans la communauté des chercheurs travaillant sur le robot *iCub*, c'est *YARP* qui est généralement utilisée.

Cinématique et dynamique du robot (*iKin* et *iDyn*) Les informations concernant la cinématique du robot sont récupérées à l'aide de la bibliothèque *iKin* [175, 173], puis calcule la cinématique inverse d'une chaîne de liens en séries, qui représente les membres et les articulations du robot. Les informations concernant la dynamique du robot sont récupérées à l'aide de la bibliothèque *iDyn* [199], qui est une extension de la librairie *iKin* qui permet de calculer les informations dynamiques de cette chaîne, telles que les forces, les moments et les couples.

2.3 Autres logiciels utilisés

Détection faciale à l'aide de *Intraface* Lors de la deuxième étude présenté dans le Chapitre 5, le robot a besoin de repérer la tête de l'utilisateur ainsi que son orientation faciale. Pour cela, nous utilisons la version d' *Intraface* [234] initialement disponible sur le site web, qui offrait le suivi du regard, l'estimation de la position et de l'orientation du regard et l'analyse d'expression. Nous avons adaptés le code source à notre étude, afin qu'à chaque détection de visage, l'orientation du regard de la face détectée soit transmise à notre logiciel à l'aide de *YARP*.



FIGURE 2.4 – Détection de l'orientation du regard à l'aide du logiciel *Intraface*.

Simulation de forces à l'aide du *Geomagic* Afin de pouvoir manipuler “à la main” le robot simulé, nous avons utilisé dans nos expériences le *Géomagic Touch* [88], un appareil haptique. Grâce à ce dispositif, l'utilisateur a ainsi pu envoyer des forces au robot simulé afin de bouger

31. *Unified Robot Description Format* qui permettent de représenter les robots simulés.

les bras du robot (*c.f.*, vidéo [Tutoriel] et Section 2.5.1). Pour cela, nous avons développé un programme en C++, inspiré du programme [176].

2.4 Xsens

Afin de récupérer la position des membres de l'utilisateur du robot, la combinaison MVN Link fournie par Xsens [188] a été utilisée (*c.f.*, Figure 2.5). Cette combinaison comporte 17 capteurs inertiels (*par ex.*, Pelvis, T8, tête) qui fournissent des mesures cinématiques inertielles de tout le corps de l'utilisateur. Ils sont pour cela composés d'accéléromètres 3D (pour calculer l'accélération), de gyroscopes 3D (pour calculer la vitesse angulaire), de magnétomètres 3D (pour calculer le champ magnétique, donc le nord magnétique de la Terre, et donc connaître l'orientation de la personne) et d'un baromètre qui mesure la pression atmosphérique.

À l'aide d'une représentation biomécanique de l'humain, le logiciel Xsens MVN récupère les données des capteurs afin d'estimer en temps réel la posture de la personne portant la combinaison. Pour cela, des algorithmes de fusion des données sont utilisés. La fréquence d'échantillonnage d'une telle combinaison est de 240Hz . Ces mesures sont ensuite récupérables en temps réel, par le biais d'un fichier de format XML. Ce fichier contient à la fois la position cartésienne des membres de l'utilisateur, ses positions angulaires articulaires, ainsi que la position de son centre de masse. C'est ce fichier que nous traitons afin de récupérer les postures de l'humain. Cette combinaison a l'avantage d'être totalement portable, ce qui permet à tout utilisateur de la porter afin de pouvoir interagir avec le robot en temps réel.

Dans l'étude présentée dans la Section 6, 23 positions en 3D sont utilisées pour représenter la posture des personnes. Il s'agit de la position de segments représentant la cinématique de l'humain calculée dans Xsens MVN.



FIGURE 2.5 – Représentation de la combinaison Xsens.

2.5 Logiciels réalisés

2.5.1 Logiciel de prédiction de mouvements et d'exécution de gestes robotiques

Dans cette thèse, nous avons développé un logiciel qui permet de modéliser des mouvements quelconques, ainsi que d'interpréter un mouvement partiel afin d'en prédire le futur. De plus, il permet d'utiliser ces prédictions afin que le robot iCub exécute des gestes.

Ce logiciel est accessible publiquement sous une licence GPL [soft.Pred.Mult]. Il se compose de deux modules, représentés dans la Figure 2.6.

Durant la phase d'apprentissage, le robot apprend les différentes ProMPs et leur tâche correspondante, pendant que le partenaire humain contrôle ses gestes. Ce contrôle est effectué à l'aide d'un guidage manuel, lorsqu'il s'agit de l'iCub réel, ou d'un dispositif haptique lors de simulation.

Un module *Matlab* permet de rejouer les ProMPs, ou de finir un mouvement qui a été initié par un utilisateur. Ce module permet au robot d'apprendre des ProMPs à partir des

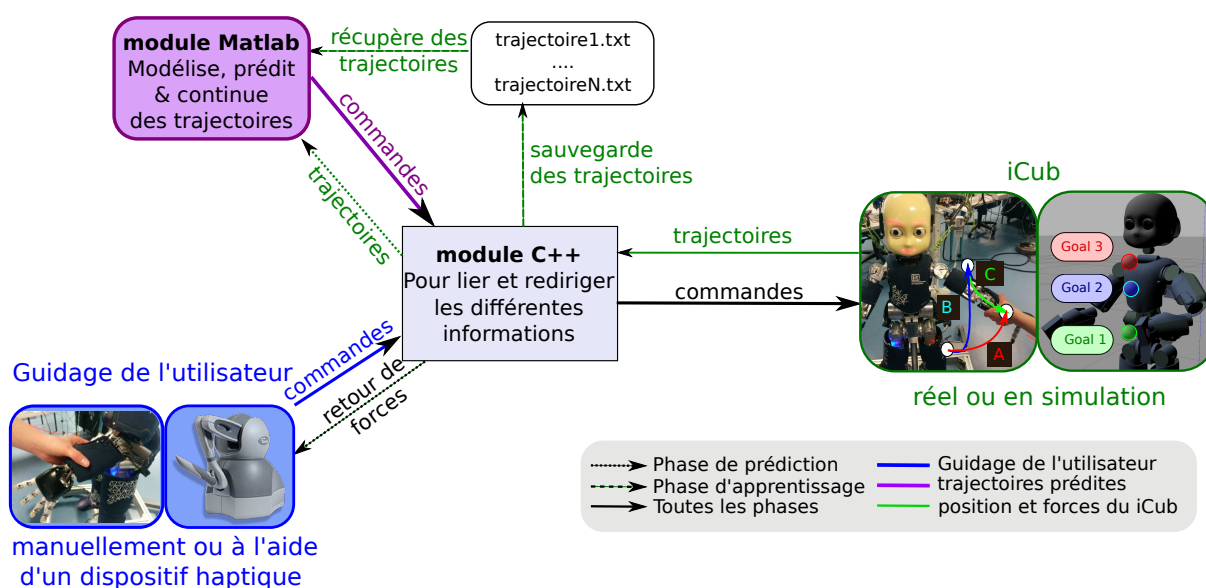


FIGURE 2.6 – *Architecture du logiciel avec représentation des échanges de données.* Le robot est soit guidé par l'utilisateur (manuellement ou à l'aide d'un dispositif haptique), ou à l'aide d'un module Matlab. Le module C++ permet de faire transiter les informations entre le robot et ces deux moyens de le guider.

trajectoires effectuées ; de rejouer ces ProMPs ; de reconnaître la ProMP qui correspond le plus à une trajectoire en cours ; et de prédire le “futur” de cette trajectoire, jusqu'à atteindre le but.

Un module C++ transmet au robot l'information de contrôle qui provient soit de l'utilisateur, soit du module Matlab. Le robot est alors capable de finaliser le mouvement initié par l'utilisateur (directement ou à l'aide d'un dispositif haptique) de manière autonome, comme présenté dans la Figure 4.2. Le module C++ est présenté dans la Section A.1.1, puis les algorithmes utilisés dans le module *Matlab* sont expliqués dans la Section 3.3. Un exemple jouet est alors présenté dans la Section A.2, suivi par un exemple plus concret dans la Section 4, où le robot simulé apprend différents types d'informations concernant les mouvements qu'il doit effectuer (c'est-à-dire, qu'il s'agit de ProMPs multi dimensionnelles). Finalement, une étude sur le robot réel est présentée dans la Section 4.3.

Dans le répertoire GIT du logiciel, un tutoriel est aussi fourni, ainsi qu'un fichier d'explication (README) et des vidéos de présentation. Le fichier d'explication décrit comment lancer des démonstrations basiques. Les vidéos présentent ces démonstrations, afin d'éclaircir le fonctionnement de cette application.

Pour plus d'information, le lecteur pourra se reporter à l'Annexe A.1.

2.5.2 Logiciel de prédiction des mouvements humains corps-complet

Le logiciel précédent n'est cependant pas capable d'exécuter des prédictions de trajectoires à grande dimension en temps réel, comme l'a mis en avant l'expérience présentée dans la Section 6.4.1. C'est pourquoi un deuxième logiciel a été créé, permettant la prédiction de trajectoire humaine corps-complet à grande dimension. Le schéma de la Figure 2.7 présente les différentes parties de ce logiciel.

Ce programme utilise le même module Matlab afin de modéliser, reconnaître et prédire des mouvements.

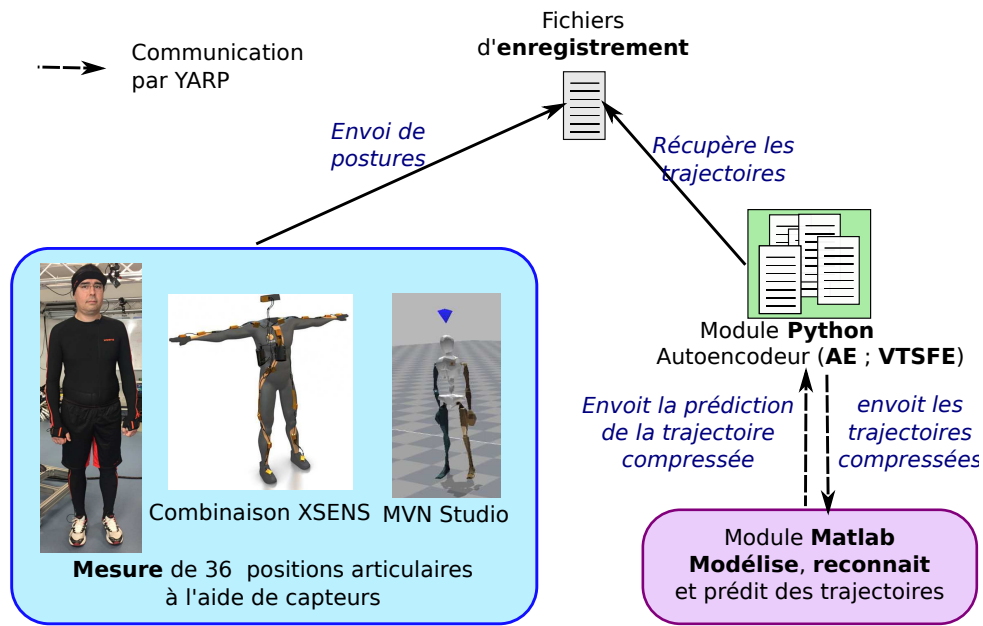


FIGURE 2.7 – *Architecture du logiciel avec représentation des échanges de données. L'utilisateur porte la combinaison Xsens et se déplace avec. Les mesures de la posture sont alors enregistrées et forment des trajectoires. Un auto-encodeur codé en Python compresse ces trajectoires, qui sont ensuite modélisées ou prédites par le module Matlab.*

La différence est que dans ce logiciel, un module Python est ajouté en amont, afin de compresser des trajectoires à grande dimension dans un espace réduit. Ces trajectoires à grande dimension proviennent des postures mesurées par les capteurs du Xsens, lorsqu'une personne porte la combinaison. Malgré cette compression, les caractéristiques principales des mouvements sont conservées (grâce l'auto-encodeur du module Python, qui apprend ces caractéristiques). Ces trajectoires compressées sont alors modélisées par le module Matlab. Cette modélisation est ensuite utilisée afin d'interpréter des mouvements compressés partiels et d'en prédire le futur. Le mouvement prédit est finalement décompressé à l'aide de l'auto-encodeur Python.

3

Méthodes

Dans cette Section, nous détaillons le cadre théorique utilisé. Il s’agit d’une description de la méthode ProMP, suivie par des méthodes de compression de données, puis de leur combinaison avec les ProMPs. Ces méthodes sont utilisées au sein de notre application afin de reconnaître les intentions humaines et de prédire la poursuite de mouvements.

En amont, la Section 3.1 précise les différentes hypothèses utilisées dans nos expériences et dans nos calculs et la Section 3.2 présente les différentes notations utilisées dans cette thèse, afin d’aider la compréhension des différentes méthodes.

Puis, la Section 3.3 présente la méthode ProMP.

Pour cela, la Section 3.3.2 formule tout d’abord le problème de l’apprentissage d’une primitive, dans un cas simplifié. L’apprentissage de cette primitive est effectué à l’aide de plusieurs trajectoires de démonstration. Ensuite, la Section 3.3.3 formule et fournit la solution au problème de prédiction de la trajectoire « future », à partir des premières observations de celle-ci. De plus, dans la Section 3.3.4, nous discutons de l’inférence de la durée globale de la trajectoire à prédire.³² Finalement, dans la Section 3.3.5 nous montrons comment reconnaître, à partir d’une portion de trajectoire, quelle compétence (chacune étant représentée par une ProMP) doit être sollicitée parmi celles apprises, puis comment prédire la poursuite de la trajectoire initiée.

La Section 3.4 présente la méthode AE-ProMP utilisée afin de prédire la poursuite de trajectoires décrites par des données de dimension importante. Pour cela, un rappel sur ce qu’est un auto-encodeur est présenté dans la Section 3.4.1, suivi par une présentation de la méthode VTSFE présentée dans la Section 3.4.2, permettant d’auto-encoder des trajectoires (c’est-à-dire, suite de données à valeur dans un espace continu (temps)). Pour finir, les méthodes permettant de combiner ces solutions d’auto-encodage (position seule ou de trajectoire) avec les ProMPs sont présentées dans la Section 3.4.3.

3.1 Hypothèses d’étude

Plusieurs hypothèses sont à considérer dans ces travaux :

- Dans les études de cette thèse, l’intention de l’utilisateur est définie comme un mouvement à effectuer dans un futur immédiat qui aboutit à une certaine position but. L’utilisateur

³². Pour certaines tâches, telles que les tâches consistant à atteindre une cible, il est raisonnable de supposer que la différence de durée des trajectoires est négligeable. Cependant, pour d’autres tâches, la durée des trajectoires peut varier, en particulier lorsque celles-ci sont effectuées par des personnes différentes.

de notre application pourra lier à cette identification de l'intention une primitive d'action correspondante, telles qu'attraper, lâcher ou encore pousser un objet.

- Lors des différentes expériences de cette thèse, nous émettons l'hypothèse que les mouvements effectués par l'utilisateur sont inclus dans les distributions apprises.
- Pour modéliser nos trajectoires, nous avons choisi d'utiliser des fonctions gaussiennes, qui se prêtent tout à fait à des mouvements naturels humains, qui varient de manière « continue », c'est-à-dire sans variation forte de position (principe du *minimum jerk* [104]).
- Afin d'apprendre une distribution de ces modélisations de trajectoires (c'est-à-dire, une ProMP), nous supposons que les paramètres de ces modélisations suivent une loi normale.
- Les ProMPs 3.3 supposent que le bruit des trajectoires s'approche du bruit blanc gaussien, ce qui s'applique à notre cas où la variation des données provient de l'imprécision des mesures.

3.2 Notations

Nous parlons de « *trajectoire entière* » (resp. « *mouvement entier* ») lorsqu'une trajectoire (resp. un mouvement) est observée jusqu'à la fin. Au contraire, une « *trajectoire partielle* » (resp. « *mouvement partiel* ») correspond à une trajectoire (resp. un mouvement) commencée par l'utilisateur et arrêtée en cours de route.

Afin de faciliter la compréhension du cadre théorique, nous synthétisons tout d'abord les notations mathématiques utilisées tout au long de cette thèse.

Trajectoires :

- $t \in \mathbb{N}^+$: la t^e mesure de la trajectoire. Par abus de langage, nous parlerons d'*instant* t , où entre chaque *instant* se déroule 0.01 secondes. De la même manière, la *durée* t_f correspondra au nombre de mesures décrivant la *trajectoire entière* (c'est-à-dire, qu'il s'agira d'une durée $t_f \times 0.01$ secondes).
- $X(t) \in \mathbb{R}^3$, $X(t) = [x(t), y(t), z(t)]^\top$: coordonnées cartésiennes de l'effecteur du robot.
- $F(t) \in \mathbb{R}^6$, $F(t) = [f_x, f_y, f_z, m_x, m_y, m_z]^\top$: forces de contact (c'est-à-dire, les forces et moments externes mesurés par le robot au niveau de son effecteur).
- $A(t) \in \mathbb{R}^3$, $A(t) = [a_1, a_2, a_3]^\top$: les angles lacet, tangage et roulis représentant l'orientation de la tête de l'utilisateur.
- $\xi(t) \in \mathbb{R}^D$: vecteur contenant les valeurs de l'état de la trajectoire à l'instant t . Ce vecteur peut être mono dimensionnel (*par ex.*, $\xi(t) = [z(t)]$), ou multidimensionnel (*par ex.*, $\xi(t) = [X(t), F(t)]^\top$), selon le type de trajectoire que l'on veut représenter à l'aide de la ProMP.
- $\Xi = \Xi_{[1:t_f]} = [\xi(1), \dots, \xi(t_f)]^\top \in \mathbb{R}^{D \cdot t_f}$: correspond à une *trajectoire entière*, composée de t_f points mesurés.
- $\Xi_{i[1:t_{fi}]}$: correspond à la i^e trajectoire de démonstration d'une tâche (c'est-à-dire, une trajectoire effectuée par l'utilisateur), de durée t_{fi} .

Primitives de Mouvement :

- $k \in [1 : K]$: k^e ProMP (c'est-à-dire, une primitive de mouvement probabiliste) d'un ensemble de K ProMPs, où chacune d'elle correspond à une tâche spécifique.
- n_k : nombre de trajectoires de démonstration utilisées pour apprendre la k^e ProMP.
- $S_k = \{\Xi_{\{k,1\}}, \dots, \Xi_{\{k,n_k\}}\}$: ensemble de n_k trajectoires de démonstration correspondant à la ProMP k .
- $\xi(t) = \Phi_t \omega + \epsilon_\xi$: modélisation de la trajectoire avec :
 - $\epsilon_\xi \sim \mathcal{N}(0, \Sigma_\xi)$: bruit blanc gaussien représentant l'erreur de précision des mesures.

- $\Phi_t \in \mathbb{R}^{D \times D \cdot M}$: fonctions de base radiale (*RBFs*) utilisées afin de modéliser les trajectoires. Elles correspondent à une matrice diagonale par bloc, avec :
 - M : nombre de *RBFs*.
 - $\psi_{ji}(t) = \frac{e^{-\frac{(t-c_i)^2}{2h}}}{\sum_{m=1}^M e^{-\frac{(t-c_m)^2}{2h}}}$: la i^e *RBF* correspondant à l'ensemble des entrées $j \in [1 : D]$. Notons que le numérateur provient d'une gaussienne $\frac{1}{\sqrt{2\pi h}} e^{-\frac{(t-c_i)^2}{2h}}$, où c_i et h sont respectivement le centre et la variance de la i^e Gaussienne. Les Gaussiennes incluses dans la *RBF* sont normalisées.
- $\omega \in \mathbb{R}^{D \cdot M}$: vecteur paramétrique indépendant du temps, utilisé pour pondérer les *RBFs*. Il s'agit des paramètres à apprendre.
- $p(\omega) \sim \mathcal{N}(\mu_\omega, \Sigma_\omega)$: distribution normale calculée à partir d'un ensemble $\{\omega_1, \dots, \omega_n\}$. Elle représente la distribution des paramètres du modèle des trajectoires. Il s'agit de la *distribution a priori*.

Modulation du temps :

- \bar{s} : le nombre d'échantillons utilisé en tant que référence : toutes les trajectoires sont redimensionnées afin qu'elles aient la même durée $t_f = \bar{s}$.
- $\Phi_{\alpha_i t} \in \mathbb{R}^{D \times D \cdot M}$: *RBFs* redimensionnées afin qu'elles soient définies sur toute la durée de la trajectoire Ξ_i .
- $\alpha_i = \frac{\bar{s}}{t_{fi}}$: paramètre de modulation du temps correspondant à la i^e trajectoire.
- $\alpha = \Psi_{\delta_{n_o}} \omega_\alpha + \epsilon_\alpha$: modélisation de la fonction de redimensionnement δ_{n_o} permettant de définir le paramètre de modulation temporel α , avec :
 - Ψ : un ensemble de *RBFs* utilisé dans la modélisation de la fonction de redimensionnement de temps entre δ_{n_o} et α ;
 - δ_{n_o} : différence entre la mesure récoltée à l'instant n_o et celle à l'instant 1. Cette différence peut correspondre à $\delta_{n_o} = \xi(n_o) - \xi(1)$ lorsque l'on considère l'ensemble des mesures décrivant la trajectoire (c'est-à-dire, la position cartésienne, les forces, etc.); $\delta_{n_o} = X(n_o) - X(1)$ lorsque l'on considère uniquement les positions cartésiennes de la trajectoire; ou encore $\delta_{n_o} = A(n_o) - A(1)$, lorsque l'on considère uniquement l'orientation de la tête du partenaire pour représenter le mouvement.
 - ω_α : vecteur paramètre pondérant les *RBFs* incluses dans la matrice Ψ .

Prédiction et inférence :

- $\Xi^o = [X^o, F^o]^\top = [\xi^o(1), \dots, \xi^o(n_o)]^\top$: observation du début d'une trajectoire, composée de n_o mesures.
- Σ_ξ^o : bruit des mesures récoltées de la trajectoire partielle.
- $\hat{\alpha}$: estimation du paramètre de modulation du temps de la trajectoire à prédire.
- $\hat{t}_f = \frac{\bar{s}}{\hat{\alpha}}$: inférence de la durée de la trajectoire à prédire.
- $\Xi^* = [\xi^o(1), \dots, \xi^o(n_o), \xi^*(n_o+1), \dots, \xi^*(t_f)]$: réalité terrain correspondant à la trajectoire que le robot doit prédire.
- $\hat{\Xi} = [\hat{X}, \hat{F}]^\top = [\xi^o(1), \dots, \xi^o(n_o), \hat{\xi}(n_o+1), \dots, \hat{\xi}(\hat{t}_f)]^\top$: trajectoire prédite.
- $p(\hat{\omega}) \sim \mathcal{N}(\hat{\mu}_\omega, \hat{\sigma}_\omega)$: distribution a posteriori du vecteur paramètre d'une ProMP, calculée à partir des observations Ξ^o .
- \hat{k} : index de la ProMP reconnue dans l'ensemble des K ProMPs apprises précédemment.

Modèles de compression des données (3^e expérience) :

- $x(t) \in \mathbb{R}^N$: données réelles, avant compression à l'instant t .
- h_i : i^e couche cachée du réseau de neurones.
- $[z_1(t), \dots, z_R(t)] \in \mathbb{R}^R$: données compressées dans l'espace latent (espace de dimension

- réduite), à l'instant t .
- x_{rec} : données reconstruites, après compression et décompression.
- α, β, τ, f_t : variables du modèle DMP (*c.f.*, Section 1.7.1) incluses dans le modèle VAE-DMP.
- $q_\phi(z|x)$: réseau de neurones qui approxime la distribution inconnue $p_\theta(z|x)$. On parle alors d'encodeur probabiliste où ϕ sont les paramètres de cette distribution (c'est-à-dire, les poids et biais de l'encodeur du réseau de neurones). Ces paramètres sont alors optimisés afin que la distribution q corresponde le plus possible à la distribution $p_\theta(z|x)$.

3.3 Primitives de mouvements probabilistes (ProMPs)

La méthode proposée dans cette thèse est une adaptation de la méthode ProMP développée par Paraschos *et al.* [170]. Cette méthode permet d'apprendre des distributions de trajectoires (c'est-à-dire, des primitives de mouvement). Cela permet de générer de nouvelles trajectoires similaires et de prédire la poursuite de mouvements initiés. Notre adaptation permet au robot *iCub* de générer des mouvements, mais aussi de reconnaître et prédire des mouvements de son bras, de la tête de son partenaire, ou encore de son corps complet. Des méthodes ont été ajoutées afin de pouvoir prédire la durée de ces trajectoires ; d'utiliser l'information des forces appliquées au robot ; d'utiliser ses différents sens afin d'améliorer ses compétences de prédiction ; ou encore de compresser des trajectoires à grande dimension afin d'accélérer les prédictions. Le logiciel qui implémente cette méthode est écrit en Matlab (le logiciel est disponible en ligne [soft.Pred.Mult]). Cette Section a pour but de détailler cette méthode.

3.3.1 Modèle appliqué à une trajectoire simple

Une ProMP est un modèle paramétrique bayésien représentant un ensemble de trajectoires observées :

$$\xi(t) = \Phi_t \boldsymbol{\omega} + \epsilon_\xi, \forall t \in [1 : \bar{s}] \quad \text{ou} \quad \Xi = \{\xi(1), \dots, \xi(\bar{s})\} = \Phi_{[1:\bar{s}]} \boldsymbol{\omega} + \epsilon_\Xi. \quad (3.1)$$

Où $\boldsymbol{\omega} \in R^M$ correspond à une distribution de paramètres ω indépendants du temps, qui pondèrent les fonctions de base radiale (RBFs) ; $\epsilon_\xi \sim \mathcal{N}(0, \Sigma_\xi)$ est le bruit blanc gaussien des trajectoires qui représente la précision des données et de la variabilité des trajectoires (*c.f.*, Section 3.1) ; \bar{s} correspond au nombre d'instant représentent l'ensemble des trajectoires (c'est-à-dire, leur durée) ; et Φ_t le vecteur contenant les M RBFs [28] évaluées à l'instant t :

$$\Phi_t = [\psi_1(t), \psi_2(t), \dots, \psi_M(t)] \in \mathbb{R}^{1 \times M}, \quad \forall t \in [1 : \bar{s}].$$

Avec :

$$\begin{cases} \psi_i(t) &= \frac{1}{\sum_{j=1}^M \psi_j(t)} e^{-\frac{1}{2h}(t/\bar{s}-c_i)^2} \in \mathbb{R}, \quad \forall t \in [1 : \bar{s}] \\ c_i &= \frac{i}{M} \\ h &= \frac{1}{M^2} \text{ (par exemple.)} \end{cases} \quad (3.2)$$

Notons que l'ensemble de ces matrices ψ sont réparties sur leur domaine de définition (c'est-à-dire, sur $[1 : \bar{s}]$) afin de représenter la trajectoire dans sa totalité.

L'Annexe B fournit davantage d'informations sur cette matrice de fonctions de base radiale.

La probabilité d'observer une trajectoire est alors :

$$p(\Xi|\boldsymbol{\omega}) = \prod_{t=1}^{\bar{s}} \mathcal{N}(\xi(t)|\Phi_t \boldsymbol{\omega}, \Sigma_\xi)$$

3.3.2 Apprentissage d'une ProMP

Supposons que le robot ait enregistré un ensemble de n_1 trajectoires : $\{\Xi_1, \dots, \Xi_{n_1}\}$, où la i^e trajectoire est définie par : $\Xi_i = \{\xi(1), \dots, \xi(t_{f_i})\}$.

$\xi(t)$ correspond au vecteur composé de l'ensemble des variables définissant la trajectoire à apprendre à l'instant t . Ce vecteur peut être à la fois mono dimensionnel (*par ex.*, $\xi(t) = [z(t)]$ correspondant à la coordonnée cartésienne d'axe z), ou multidimensionnel (*par ex.*, $\xi(t) = [X(t), F(t)]^\top$).

Remarque : chaque trajectoire i a sa propre durée t_{f_i} , c'est pourquoi un paramètre de modulation du temps α peut être appliqué pour ré-échantillonner celles-ci afin qu'elles soient décrites par un même nombre d'instants \bar{s} (c'est-à-dire, qu'elles sont de même durée, *c.f.*, Section 3.3.4). Les trajectoires ainsi normalisées temporellement sont alors utilisées pour apprendre la ProMP. Dans cette Section, on simplifie en supposant qu'elles sont déjà de même durée \bar{s} .

Pour chaque trajectoire Ξ_i , le vecteur de paramètre ω_i est alors calculé afin d'obtenir $\xi_i(t) = \Phi_t \omega_i + \epsilon_\xi, \forall t \in [1 : \bar{s}]$, avec l'erreur la plus petite. Ce calcul consiste donc à trouver les paramètres qui minimisent l'écart entre les données de la trajectoire observée $\xi_i(t)$ et leur modélisation $\Phi_t \omega_i + \epsilon_\xi$. Pour cela, la méthode des moindres carrés (*Least Mean Square*) est utilisée :

$$\omega_i = (\Phi_t^\top \Phi_t)^{-1} \Phi_t^\top \xi_i(t). \quad (3.3)$$

Sachant que la matrice $\Phi_t^\top \Phi_t$ peut ne pas être inversible, un terme sur la diagonale est ajouté. Il s'agit de la régularisation de Tikhonov :

$$\omega_i = (\Phi_t^\top \Phi_t + \lambda)^{-1} \Phi_t^\top \xi_i(t), \quad \lambda = 10^{-11} \cdot \mathbf{1}_{D \cdot M \times D \cdot M}. \quad (3.4)$$

Ce calcul est effectué pour chaque trajectoire observée, fournissant un ensemble de paramètres : $\{\omega_1, \dots, \omega_n\}$. Une distribution de cet ensemble de paramètres est alors calculée, avec l'hypothèse que cette distribution suit une loi normale :

$$p(\omega) \sim \mathcal{N}(\mu_\omega, \Sigma_\omega). \quad (3.5)$$

$$\begin{aligned} \text{Avec : } \mu_\omega &= \frac{1}{n} \sum_{i=1}^n \omega_i. \\ \text{Et : } \Sigma_\omega &= \frac{1}{n-1} \sum_{i=1}^n (\omega_i - \mu_\omega)^\top (\omega_i - \mu_\omega). \end{aligned}$$

Les paramètres probabilistes de la ProMP sont ainsi appris. Lorsque l'on souhaite générer une trajectoire, on utilise généralement la trajectoire "moyenne" de sa distribution : $\Xi = \Phi_{[1:\bar{s}]} \mu_\omega$.

Le graphe de gauche de la Figure 3.1 représente des trajectoires de démonstration. Dans cet exemple, il s'agit d'un mouvement de levée de bras du robot iCub, regardé dans l'axe vertical, correspondant à la hauteur. Le graphe de droite représente la ProMP correspondante, redimensionné afin d'avoir un nombre d'échantillons correspondant au nombre de référence $\bar{s} = 100$. Cette distribution est apprise avec $M = 5$ *RBFs*³³.

La Figure 3.2 permet quant à elle d'illustrer l'influence qu'a le nombre choisi de fonctions de base radiale sur la précision de la modélisation. On y remarque que lorsqu'il n'y a pas assez de fonctions (gauche), la modélisation ne permet pas de représenter correctement la forme des trajectoires. Au milieu, les trajectoires sont correctement représentées par les 5 fonctions. Avec

33. Cet exemple est inclus dans le logiciel Matlab sous le nom `demo_plot1DOF.m`. D'autres exemples plus compliqués sont aussi introduits dans le logiciel, sous des noms de type : `demo_plot*.m`.

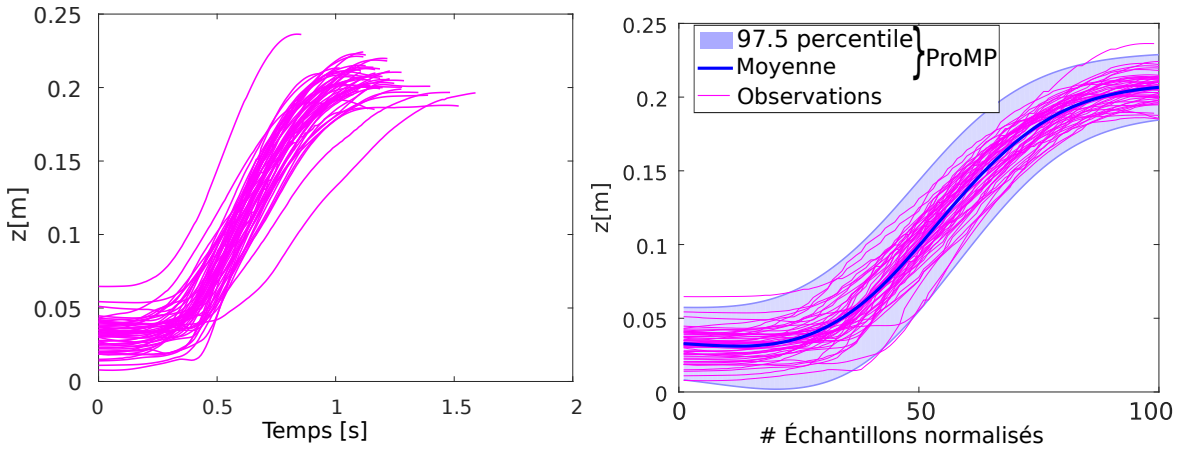


FIGURE 3.1 – *Exemple de ProMP*—Les trajectoires observées sont représentées en magenta. La ProMP est représentée en bleu. Les paramètres suivants sont utilisés : $\bar{s} = 100$ le nombre référence d'échantillons ; $M = 5$ le nombre de RBFs étalées sur la dimension temporelle ; et $h = 0.04$ ($=\frac{1}{M^2}$) la variance de ces RBFs.

plus de fonctions (droite), la variance de la distribution décroît, parce que la modélisation permet plus de précision. Cependant, augmenter arbitrairement le nombre de fonctions de base radiale n'est pas une bonne idée, puisqu'à un certain moment, ça n'améliore plus la précision du modèle et pire, ça peut provoquer du surapprentissage et cela augmente le coût en calcul. Après avoir appris la distribution des paramètres $p(\omega) \sim \mathcal{N}(\mu_\omega, \Sigma_\omega)$ qui définit la ProMP correspondant à une certaine primitive d'action³⁴, celle-ci peut être utilisée afin de prédire l'évolution d'un mouvement initié, dont on a des observations. Cette prédiction est présentée dans la Section suivante.

3.3.3 Prédiction de la poursuite d'un mouvement

Supposons que le robot mesure les n_o premières observations de la trajectoire à prédire : $\Xi^o = \{\xi^o(1), \dots, \xi^o(n_o)\}$. Celui-ci doit alors de prédire l'évolution de cette trajectoire, c'est-à-dire de trouver $\{\hat{\xi}(n_o + 1), \dots, \hat{\xi}(\hat{t}_f)\}$, où \hat{t}_f est l'estimation de la durée de la trajectoire³⁵. Il s'agit donc d'estimer la trajectoire complète $\hat{\Xi}$, sachant les n_o premières données : $\hat{\Xi} = \{\Xi^o, \hat{\xi}(n_o + 1), \dots, \hat{\xi}(\hat{t}_f)\}$.

Pour effectuer cette prédiction, le robot commence par identifier quelle est la distribution apprise qui correspond au mouvement initié³⁶, appelée “*distribution a priori*” : $p(\omega)$.

Puis, les paramètres de cette distribution sont adaptés afin de générer la trajectoire estimée $\hat{\Xi}$.

Pour cela, la distribution “*a posteriori*” des paramètres correspondant à cette trajectoire, notée $p(\hat{\omega}) \sim \mathcal{N}(\hat{\mu}_\omega, \hat{\Sigma}_\omega)$, est calculée par la mise à jour de la distribution a priori à l'aide de la formule :

$$\begin{cases} \hat{\mu}_\omega &= \mu_\omega + K(\Xi^o - \Phi_{[1:n_o]} \mu_\omega) \\ \hat{\Sigma}_\omega &= \Sigma_\omega - K(\Phi_{[1:n_o]} \Sigma_\omega) \end{cases} \quad (3.6)$$

34. c'est-à-dire, calcul de la distribution $p(\omega)$ à partir de l'ensemble de mesure $\{\omega_1, \dots, \omega_n\}$, où chaque ω_i correspond à l'estimation d'un paramètre calculé à partir des démonstrations de trajectoires.

35. La Section 3.3.4 explicitera cette estimation de la durée.

36. Dans notre cas, le mouvement est initié par l'utilisateur qui guide le mouvement.

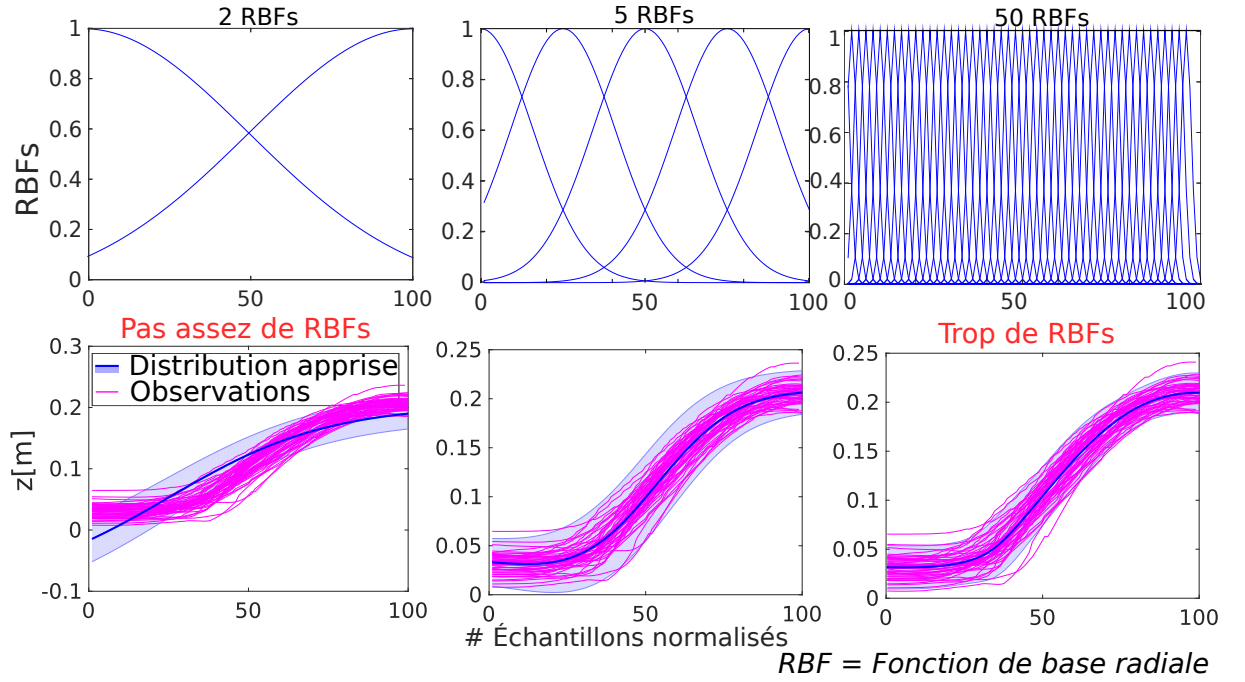


FIGURE 3.2 – Représentation d’une ProMP en fonction du nombre de fonctions de base radiale utilisé pour modéliser ces trajectoires. De gauche à droite : $M = \{2, 5, 50\}$ fonctions avant apprentissage, de variance $h = \frac{1}{M^2}$.

où K est un paramètre de gain :

$$K = \Sigma_{\omega} \Phi_{[1:n_o]}^{\top} (\Sigma_{\xi}^o + \Phi_{[1:n_o]} \Sigma_{\omega} \Phi_{[1:n_o]}^{\top})^{-1} \quad (3.7)$$

Les Équations 3.6 et 3.7 proviennent de calculs basés sur les équations de distributions marginales (c’est-à-dire, la distribution d’une des composantes de vecteurs multivariés) et conditionnelles (c’est-à-dire, la distribution d’une variable par rapport à une autre) [170, 18], comme cela est détaillé dans l’Appendix C.

La Figure 3.3 présente la trajectoire prédite, correspondant à un mouvement de levée du bras gauche du robot iCub. Les différents graphiques montrent la prédiction des trajectoires, lorsque le robot connaît $n_o = 10, 30, 50, 80\%$ des données de la trajectoire entière.³⁷ Cette Figure permet de montrer que la prédiction de la trajectoire est de plus en plus précise, avec l’augmentation des observations de la trajectoire. Une vidéo exemple présente la prédiction de trajectoires qui correspondent à cette tâche de levée de bras, à partir de la simulation *Gazebo* (c.f., la deuxième et troisième vidéo du dossier “vidéos” du dépôt GIT [soft.Pred.Mult]).

3.3.4 Prédiction de la durée de la trajectoire souhaitée

La Section précédente présentait une formulation générale des ProMPs, en faisant l’hypothèse que toutes les trajectoires observées ont la même “durée” et donc, qu’elles étaient représentées par le même nombre d’échantillons.³⁸ C’est pourquoi, la durée des trajectoires générées par les *RBFs* y est fixe et égale à t_f .

37. Cet exemple est aussi disponible dans notre logiciel, sous le nom `demo_plot1DOF.m`.

38. Rappelons que nous appelons ici durée le nombre total de mesures représentant la trajectoire entière.

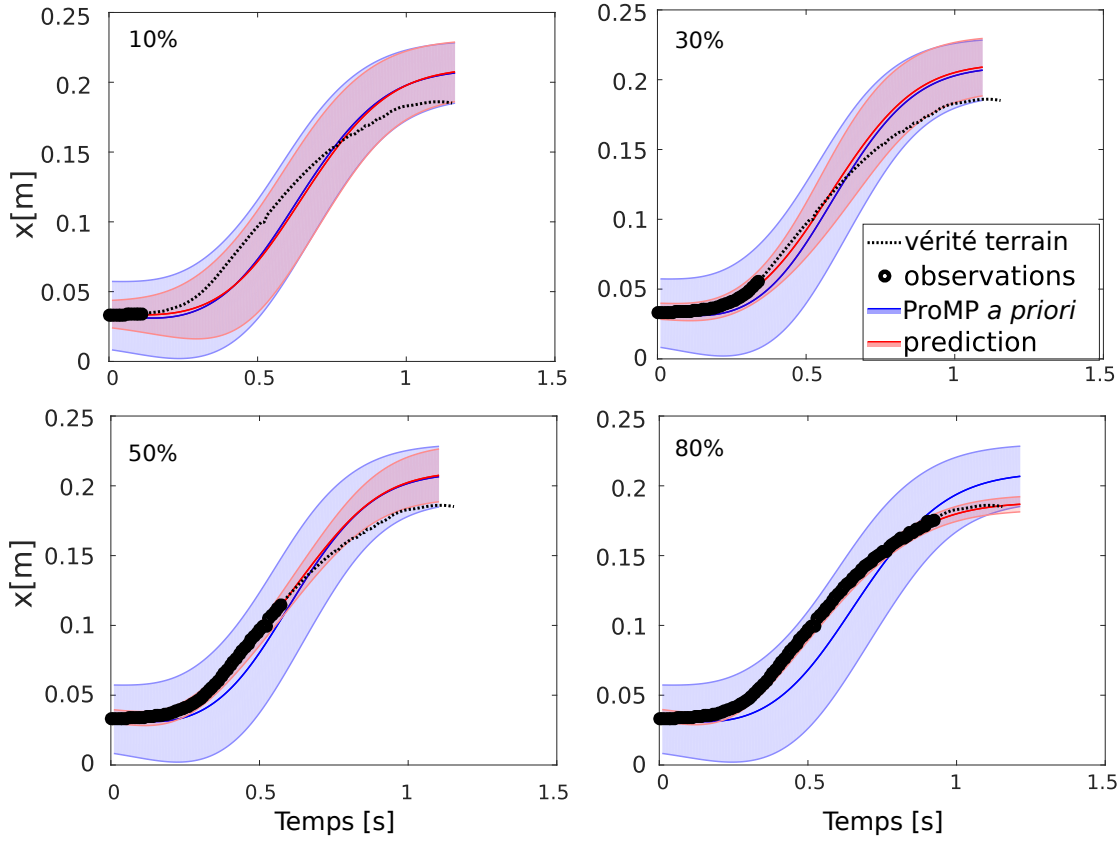


FIGURE 3.3 – *Prédiction d'une trajectoire* – Prédiction de la poursuite de la trajectoire testée, à partir de mesures partielles de celle-ci et de la ProMP apprise en amont (c.f., Figure 3.1). Les graphes représentent la prédiction après l'observation de 10%, 30%, 50% et 80% des données de la trajectoire entière.

Afin de générer des trajectoires réelles, on considère maintenant que la durée des trajectoires de démonstration varie. Pour pouvoir apprendre une distribution de ces trajectoires, on ajoute alors un paramètre de modulation du temps $\alpha = \bar{s}/t_f$ dans la modélisation des trajectoires. Ce paramètre permet de redimensionner la trajectoire actuelle afin qu'elle ne soit plus composée de t_f , mais de \bar{s} observations.

Remarques :

- La durée normalisée \bar{s} peut être choisie arbitrairement. Par exemple, elle peut correspondre à la moyenne des durées des trajectoires ($\bar{s} = \frac{1}{k} \sum_{i=1}^k t_{fi}$).
- Dans la littérature, ce terme α est parfois appelé la *phase* [170, 171]. En effet, il permet de changer la phase des fonctions de base radiale afin que celles-ci soient étalées temporellement sur l'ensemble de la durée de la trajectoire.

Le paramètre de modulation du temps de la i^e trajectoire Ξ_i est calculé avec : $\alpha_i = \frac{\bar{s}}{t_{fi}}$. Ainsi, on obtient : $\alpha t \in [1 : \bar{s}]$. Le modèle ProMP amélioré devient alors :

$$\xi_t = \Phi_{\alpha t} \boldsymbol{\omega} + \epsilon_t, \quad (3.8)$$

où $\Phi_{\alpha t}$ est la matrice de *RBFs* évaluée à l'instant αt . L'ensemble des M fonctions gaussiennes, qui correspondent aux *RBFs*, sont ainsi étalées afin de représenter le même nombre d'échantillons

\bar{s} :

$$\Phi_{\alpha t} = [\psi_1(\alpha t), \psi_2(\alpha t), \dots, \psi_M(\alpha t)].$$

Lors de la phase d'apprentissage, l'ensemble de paramètres α calculés pour chaque trajectoire de démonstration est récupéré : $S_\alpha = \{\alpha_1, \dots, \alpha_n\}$. Puis, en utilisant cet ensemble, la ProMP apprise peut être rejouée à différentes vitesses³⁹. Lorsque $\alpha = 1$, la vitesse de la trajectoire permet de finir le mouvement en \bar{s} instants.

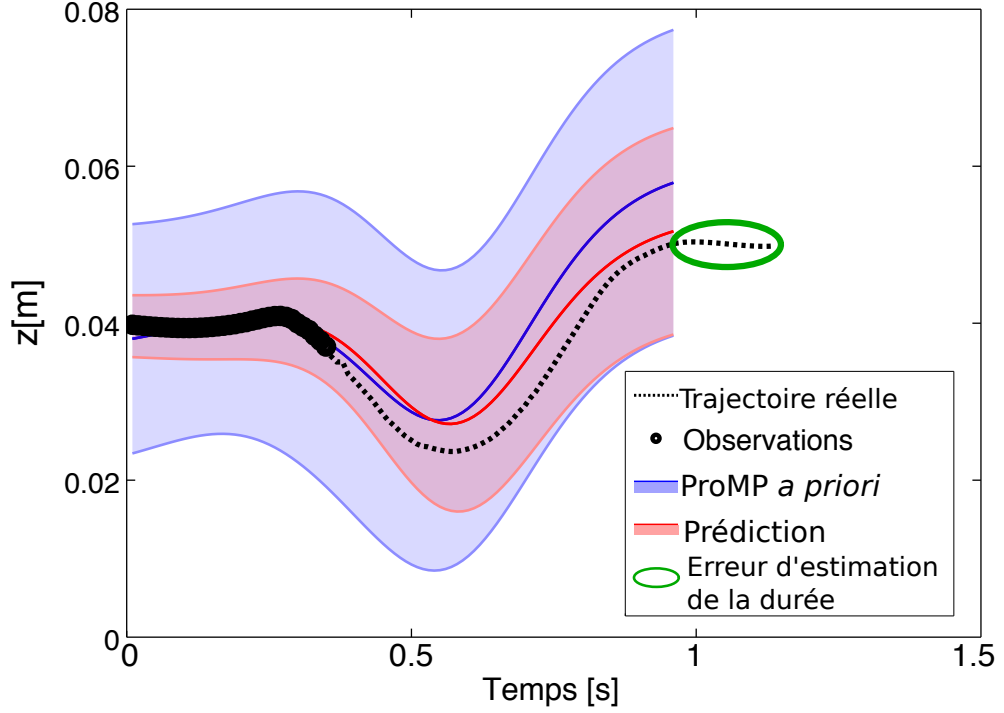


FIGURE 3.4 – *Erreur de prédiction de la durée d'une trajectoire après l'observation partielle de celle-ci.* – La distribution a priori (en bleu) est mise à jour à l'aide des observations (gros points de couleur noire) afin d'obtenir la distribution a posteriori (en rouge) alignée avec les observations. Le calcul de la distribution a posteriori est effectué avec l'hypothèse que la vitesse de la trajectoire correspond à la moyenne des vitesses des trajectoires de démonstrations utilisées lors de la phase d'apprentissage (c'est-à-dire, qu'elle est calculée à partir de la moyenne α , de l'ensemble $\alpha_1, \dots, \alpha_K$ correspondant aux K trajectoires de démonstrations). Sans inférence du paramètre de modulation du temps de la trajectoire observée et donc en utilisant la moyenne α , la trajectoire prédite a une durée visiblement différente de celle réelle, représentée par des petits points de couleur noire.

Lors de la phase de prédiction, le paramètre de modulation du temps α de la trajectoire partiellement observée est inconnu et doit être estimé. Cette estimation est essentielle pour assurer une bonne reconnaissance, comme cela est illustré dans la Figure 3.4. Dans cette Figure, les observations du début de la trajectoire sont représentées en noir, la vérité terrain (c'est-à-dire, la trajectoire désirée) par des petits points de couleur noire, la distribution a priori en bleu et la distribution a posteriori en rouge. Sur ce graphique, le cercle vert met en avant le fait que la trajectoire prédite a une durée plus courte que la trajectoire réelle attendue. Cette différence

³⁹. En effet, basé sur l'hypothèse qu'entre deux instants successifs d'une trajectoire, la même durée s'est écoulée, le nombre d'instants représentant la trajectoire contient l'information de sa vitesse.

est due à l'erreur d'estimation du paramètre de modulation du temps. Cette estimation $\hat{\alpha}$ est calculée dans cet exemple par la moyenne des paramètres contenus dans l'ensemble S_α récupérés lors de la phase d'apprentissage :

$$\hat{\alpha} = \frac{1}{n_k} \sum_{k=1}^{n_k} \alpha_k. \quad (3.9)$$

Or, cette estimation basée sur la moyenne n'est suffisante que lorsque la primitive de mouvement représente des mouvements très réguliers, ou plus généralement lorsqu'il est raisonnable de supposer que les différences de vitesse entre les trajectoires peuvent être négligées. Ainsi, cette estimation n'est pas adaptée pour de nombreuses applications.

C'est pourquoi d'autres méthodes de calcul du paramètre de modulation du temps ont été implémentées et comparées, afin d'estimer de manière plus précise la durée des trajectoires.

- La première méthode, présentée précédemment, consiste donc en un calcul de la **moyenne des α_k** (*c.f.*, Équation 3.9).
- La seconde consiste en un calcul du **maximum de vraisemblance** (ou plus exactement du log de vraisemblance, pour simplifier les calculs) :

$$\hat{\alpha} = \operatorname{argmax}_{\alpha \in S_{\alpha k}} \{\ln(L(\Xi^o | \mu_\omega, \sigma_\omega, \alpha))\}. \quad (3.10)$$

Avec :

$$\begin{aligned} \ln(L(\Xi^o | \mu_\omega, \sigma_\omega, \alpha)) = & -\frac{D \cdot t_{fi}}{2} \ln(2\pi) - \frac{D}{2} \ln(\det(\Phi_{\alpha t} \Sigma_\omega)) \\ & - \frac{1}{2} \sum_{t=1}^{t_{fi}} (\Xi^o(t) - \Phi_{\alpha t} \mu_\omega)^\top (\Phi_{\alpha t} \Sigma_\omega)^{-1} (\Xi^o(t) - \Phi_{\alpha t} \mu_\omega) \end{aligned}$$

- La troisième méthode consiste en un critère de **distance minimum**. Il s'agit de la recherche du $\hat{\alpha}$ qui minimise le plus la distance entre la trajectoire observée Ξ_t^o et celle correspondant à la moyenne de la ProMP, pour les n_o premières itérations :

$$\hat{\alpha} = \operatorname{argmin}_{\alpha \in S_{\alpha k}} \left\{ \sum_{t=1}^{n_o} |\Xi_t^o - \Phi_{\alpha t} \mu_\omega| \right\}. \quad (3.11)$$

- La quatrième méthode consiste en la création d'un **modèle**, qui se base sur l'hypothèse suivante :

Il y a une corrélation entre la valeur du paramètre α et la variation de la valeur des données de la trajectoire entre les instants $t = 0$ et $t = n_o$ (notée δ_{n_o}). Cette hypothèse est appropriée puisque les trajectoires de notre application ont généralement une croissance monotone.

En effet, le paramètre α représente la vitesse de mouvement et cette vitesse peut être approximée par $\dot{X} = \frac{\delta X}{t_f}$ (ou plus généralement $\dot{\Xi} = \frac{\delta \Xi}{t_f}$).

Ainsi on a par exemple : $\alpha \propto \delta_{n_o} = \Xi(n_o) - \Xi(1)$.

Ce paramètre peut aussi être approximé par une sous-partie des observations, par exemple par la position cartésienne du robot : $\delta_{n_o} = X(n_o) - X(1)$.

Nous modélisons l'échantillonnage entre δ_{n_o} et α avec :

$$\alpha = \Psi(\delta_{n_o})^\top \omega_\alpha + \epsilon_\alpha. \quad (3.12)$$

Où Ψ correspond à la matrice de *RBFs*, et ϵ_α à du bruit blanc gaussien.

	#observations	$\alpha = \frac{\bar{s}}{\text{Iterations}}, \bar{s} = 100$	Durée [s]
Minimum	83	1.2048	0.83
Maximum	115	0.8696	1.15
Moyenne	100	1	0.99
Écart type	9	11.1111	0.09

TABLEAU 3.1 – Information concernant la durée des trajectoires présentées dans la Figure 3.1.

Lors de la phase d’apprentissage, le paramètre ω_α est calculé en utilisant l’Équation 3.3.

Lors de la phase de prédiction, le paramètre $\hat{\alpha} = \Psi(\delta_{n_o})^\top \omega_\alpha$ est calculé, puis intégré dans la modélisation de la trajectoire.

La Figure 3.5 représente les différentes trajectoires prédites après avoir observé une trajectoire partielle, en fonction de la méthode utilisée pour estimer le paramètre de modulation du temps. Cette trajectoire partielle correspond à $n_o = 40\%$ de la trajectoire désirée. Dans ce test, il y a peu de variation de durée entre les trajectoires, comme présenté dans le Tableau 3.1, le meilleur résultat est accompli avec la méthode **moyenne des** α_k (utilisées lors de la phase d’apprentissage). Par contre, nous verrons dans la Section 3.3.5 que, lorsqu’il s’agit de trajectoires plus complexes où la durée des trajectoires varie plus fortement, les autres méthodes peuvent être préférables. De plus, dans une expérience avec l’iCub simulé, présentée dans la Section 4.2, une comparaison de ces quatre méthodes d’estimation du paramètre α est fournie.

Dans la littérature, il existe d’autres méthodes permettant de calculer ce paramètre α (*c.f.*, Section 1.7.3).

3.3.5 Reconnaître une primitive de mouvement parmi celles apprises

Afin d’améliorer les performances des robots, ceux-ci doivent être capables d’adopter différents comportements afin d’effectuer des tâches distinctes. Dans la méthode que nous proposons, chaque ProMP représente une tâche. Ainsi, il est important que le robot soit capable d’apprendre différentes ProMPs, disons un nombre K , puis d’être capable de reconnaître laquelle de ces trajectoires est initiée par l’utilisateur du robot.

Durant la phase d’apprentissage d’une ProMP $k \in [1 : K]$, le robot observe différentes trajectoires $S_k = \{\Xi_1, \dots, \Xi_n\}$. Pour chaque ProMP, il apprend alors une distribution à partir de l’ensemble des vecteurs paramètres $p(\omega_k) \sim \mathcal{N}(\mu_{\omega_k}, \Sigma_{\omega_k})$, en se basant sur l’Équation 3.3. De plus, le robot stocke la durée des trajectoires observées dans un ensemble : $S_{\alpha k} = \{\alpha_{1k}, \dots, \alpha_{nk}\}$.

Après avoir appris ces K ProMPs, le robot peut utiliser les informations apprises afin d’exécuter une trajectoire et la tâche correspondante. Pour cela, le robot doit d’abord inférer, à partir des observations partielles de la trajectoire initiée par son partenaire, quels sont le mouvement et la tâche attendus. Puis, à partir de cette inférence, il doit estimer la poursuite du mouvement attendu par l’humain⁴⁰.

Soit $\Xi^o = [\Xi_1 \dots \Xi_{n_o}]^\top$ les observations partielles de la trajectoire initiée. À partir de ces observations partielles, le robot peut reconnaître la ProMP $\hat{k} \in [1 : K]$ correspondant la mieux à la trajectoire partielle, en suivant plusieurs étapes suivantes.

40. Attendu consciemment ou non, dans le sens où l’utilisateur en a émis l’intention.

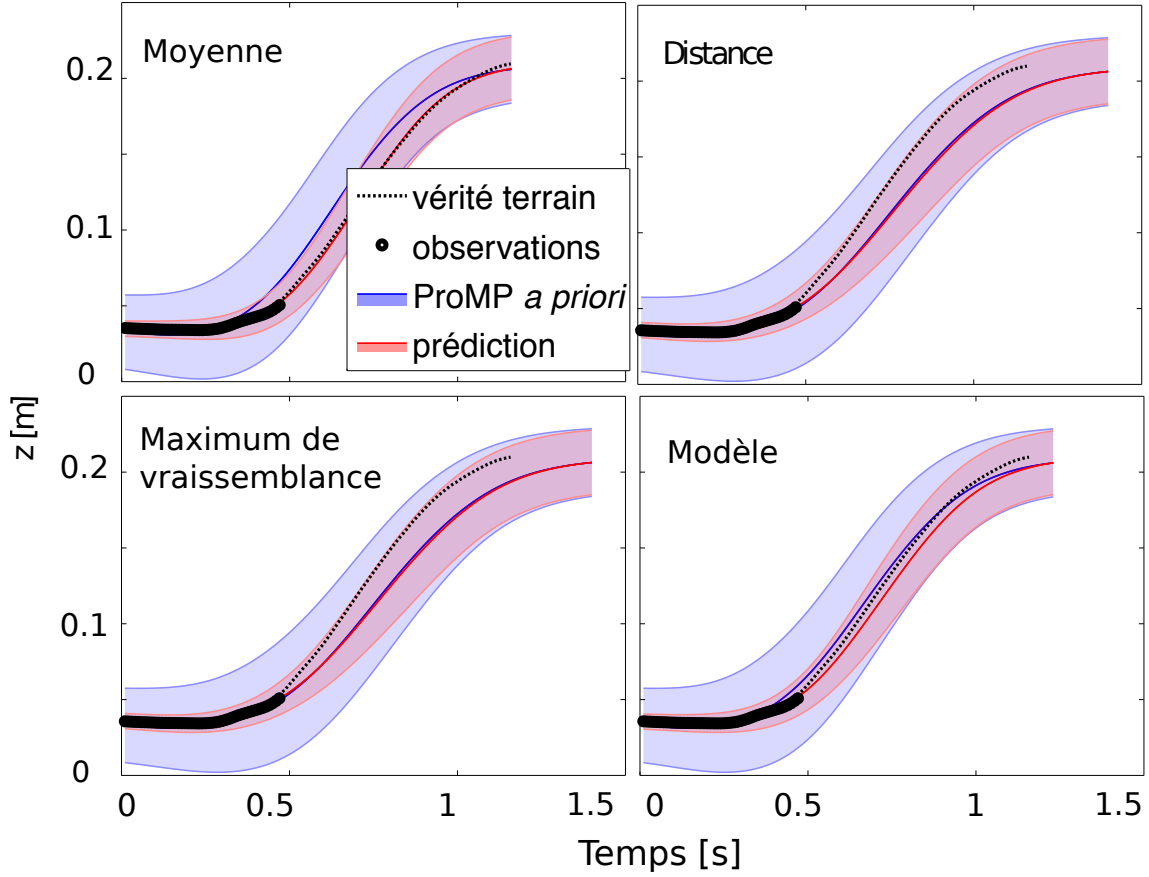


FIGURE 3.5 – *Prédiction de la trajectoire future après l’observation de $n_o = 40\%$ d’une trajectoire test, à l’aide de la ProMP apprise (Figure 3.1)– Les graphiques représentent la trajectoire prédite en fonction du critère utilisé afin d’estimer le paramètre de modulation du temps : moyenne des α_k (Équation 3.9) ; distance minimum (Équation 3.11) ; maximum de vraisemblance (Équation 3.10) ; ou modélisation du paramètre α , en fonction de la variation des données entre deux instants (Équation 3.12).*

Tout d’abord, pour chaque ProMP $k \in [1 : K]$, le robot commence par calculer le paramètre de modulation du temps $\hat{\alpha}_k$ le plus probable (*c.f.*, Section 3.3.4), afin d’obtenir un ensemble contenant les ProMPs couplées avec leur durée la plus probable : $S_{[\mu_{\omega_k}, \hat{\alpha}_k]} = \{(\mu_{\omega_1}, \hat{\alpha}_1), \dots, (\mu_{\omega_K}, \hat{\alpha}_K)\}$

Puis, le robot calcule la ProMP \hat{k} la plus probable depuis l’ensemble : $S_{[\mu_{\omega_k}, \hat{\alpha}_k]}$, en adaptant les critères présentés dans la Section précédente.

Par exemple, le robot peut minimiser la distance entre les observations partielles et les premières données de la trajectoire représentant la moyenne de la ProMP, en calculant :

$$\hat{k} = \arg \min_{k \in [1:K]} \left[\frac{1}{n_o} \sum_{t=1}^{n_o} |\Xi_t - \Phi_{\hat{\alpha}_k t} \mu_{\omega_k}| \right]. \quad (3.13)$$

Dans cette équation, pour chaque ProMP $k \in [1 : K]$, la distance moyenne est calculée entre la trajectoire initiée Ξ_t et la moyenne de la ProMP $\Phi_{\hat{\alpha}_k t} \mu_{\omega_k}$, avec $t = [1 : n_o]$. La ProMP qui correspond la mieux au mouvement initié est alors la ProMP \hat{k} qui a la distance la plus petite.

Lorsque cette \hat{k}^e ProMP est identifiée, la distribution *a posteriori* de celle-ci est calculée afin de prendre en compte le début de trajectoire observée. Pour cela, l'Équation 3.6 est adaptée :

$$\begin{cases} \hat{\mu}_{\omega_{\hat{k}}} &= \mu_{\omega_{\hat{k}}} + K(\Xi^o - \Phi_{\hat{\alpha}_{\hat{k}}[1:n_o]}\mu_{\omega_{\hat{k}}}) \\ \hat{\Sigma}_{\omega_{\hat{k}}} &= \Sigma_{\omega_{\hat{k}}} - K(\Phi_{\hat{\alpha}_{\hat{k}}[1:n_o]}\Sigma_{\omega_{\hat{k}}}) \\ K &= \Sigma_{\omega_{\hat{k}}}\Phi_{\hat{\alpha}_{\hat{k}}[1:n_o]}^\top(\Sigma_{\xi^o} + \Phi_{\hat{\alpha}_{\hat{k}}[1:n_o]}\Sigma_{\omega_{\hat{k}}}\Phi_{\hat{\alpha}_{\hat{k}}[1:n_o]}^\top)^{-1}. \end{cases} \quad (3.14)$$

Avec $\hat{\alpha}_{\hat{k}}[1:n_o]$, le vecteur $\hat{\alpha}_{\hat{k}} t$ où $t = [1:n_o]^\top$.

Finalement, la trajectoire prédite correspond à :

$$\forall t \in [1:\hat{t}_f], \hat{\xi}(t) = \Phi_{\hat{\alpha}_{\hat{k}}t}\hat{\mu}_{\omega_{\hat{k}}}.$$

Avec une durée estimée de $\hat{t}_f = \hat{\alpha}_k \bar{s}$. Le robot peut alors finir le mouvement en exécutant la trajectoire “future” la plus probable : $\hat{\Xi} = [\hat{\xi}_{n_o+1} \dots \hat{\xi}_{\hat{t}_f}]^\top$.

Dans cette section, nous avons présenté le fonctionnement de la méthode ProMP, qui permet de modéliser des distributions de trajectoires afin de pouvoir générer, inférer et prédire des trajectoires incluses dans cette distribution.

Une limitation de cette méthode concerne le temps de calcul nécessaire à la prédiction de mouvements à grande dimension. Or, dans la dernière étude de cette thèse, le robot doit modéliser et prédire des mouvements corps-complet de l'utilisateur. Nous verrons alors dans la Section 6.4.1 que le robot n'est pas capable d'exécuter ces prédictions en temps réel mou, ce qui l'empêche d'interagir correctement avec l'utilisateur.

C'est pourquoi nous proposons dans la prochaine Section, une nouvelle méthode qui combine compression de données et primitives de mouvements probabilistes, inspirée des méthodes de compression de données présentées dans la Section 1.8.2.

3.4 ProMPs appliquées à des données à grande dimension

Dans cette Section, nous présentons en détail les méthodes AE-ProMPs et VTSFE-ProMPs, qui combinent les méthodes de compression de données (VTSFE et AE) avec la méthode ProMP afin de prédire la poursuite de trajectoires à grande dimension. Pour cela, nous introduisons tout d'abord les méthodes AE et VTSFE permettant de compresser et de décompresser les données.

3.4.1 Compression des données à l'aide d'un auto-encodeur classique

Dans leur forme la plus simple, les AEs sont des perceptrons multicouches dont la couche de sortie, de la même dimension que la couche d'entrée, est entraînée afin de reconstruire l'entrée. Leur structure est souvent symétrique et est divisée en une partie encodeur et décodeur.

D'un côté, l'encodeur transforme l'entrée x_t de dimension N (dans le cas de notre étude [D.W], Section 6, cette entrée correspond à l'information cinématique du corps entier) en une valeur z_t dans l'espace latent de dimension R , avec $R \ll N$.

De l'autre côté, le décodeur fait la transformation inverse de l'espace latent vers l'espace d'origine reconstruit (dans notre exemple, l'espace cinématique).

Ces deux parties sont généralement composées d'un certain nombre de couches cachées et de fonctions d'activation non linéaires, afin de construire un espace latent comprimé non linéaire. Le

papier revu [14] résume les différentes manières de représenter les données incluant les AEs. Un schéma simplifié de cette méthode est présenté par la partie gauche de la Figure 3.6.

Un auto-encodeur est composé de :

- Une couche x d’entrée dont les unités correspondent aux données à compresser.
- Une couche compressée (espace latent) z dont le nombre d’unités est choisi et correspond à la dimension voulue de compression.
- Une couche de sortie x_{rec} , dont le nombre d’unités correspond à celui de la couche d’entrée. Après apprentissage, les valeurs des données d’entrée et de sortie doivent être identiques.
- Des couches cachées, notées h_1 , peuvent être ajoutées entre la couche d’entrée et celle de compression et d’autres, notées h_2 , entre cette même couche de compression et la couche de sortie.

Avant apprentissage, les poids de ces couches sont initialisés à l’aide de différentes méthodes, telles que l’initialisation de Xavier [93], l’initialisation aléatoire uniforme, valeur “moyenne”, etc.

De plus, une fonction d’activation doit être définie, qui active les unités en fonction de la valeur reçue en entrée. Des exemples de telles fonctions sont la sigmoïde, l’unité linéaire rectifiée (ReLU) [161], ou encore leaky-ReLU [236].

Lors de l’apprentissage, le réseau de neurones se met à jour à l’aide du calcul l’erreur entre l’entrée x et le sortie x_{rec} dont les valeurs des unités doivent être identiques. Généralement, cette erreur est calculée à l’aide de l’erreur des moindres carrés. Puis, les poids de ce réseau sont mis à jour afin de faire converger les valeurs des unités de la couche de sortie avec celles en entrée. Cela peut être effectué par une descente de gradient, à l’aide de l’algorithme classique du gradient stochastique et d’optimiseurs qui ajustent les paramètres d’apprentissage (tels que le taux d’apprentissage, le taux de décroissance exponentiel, etc.). Par exemple, l’optimiseur RMSProp [217] divise le gradient par une moyenne mobile de sa valeur récente, ou encore ADAM (adaptive moment estimation) [132] qui a comme avantages d’être efficace, d’avoir peu de besoins de mémoire, d’être utilisable pour des données de grande dimension et d’être adapté aux gradients bruités.

Nous avons cependant vu dans la Section 1.8.1 que les auto-encodeurs et leurs variantes (par exemple VAE et AAE) ne permettent pas facilement de représenter des données séquentielles. En effet, une méthode serait de mettre la séquence de données en entrée, mais cette astuce a plusieurs inconvénients, tels que l’agrandissement de la dimension d’entrée, et la nécessité de fixer la taille N de la séquence de données. C’est pourquoi nous présentons ci-dessous un type d’auto-encodeur permettant de conserver l’information temporelle.

3.4.2 Compression de dimension à l’aide de la méthode *VTSFE*

VTSFE n’étant pas une méthode développée dans cette thèse, nous nous contentons ici d’en résumer le fonctionnement général. Pour plus d’informations, le lecteur pourra se reporter à l’article dédié à cette méthode [47].

La méthode VTSFE (extracteurs de caractéristiques de séries temporelles variationnelles) est une amélioration de la méthode VAE-DMP proposée dans [48]. Ces deux méthodes ont pour but de compresser des positions provenant d’une trajectoire, tout en conservant l’information de la dynamique de celle-ci. Contrairement aux simples AEs, ces deux méthodes ne se contentent plus d’auto-encoder des données, mais d’apprendre un espace latent dynamiquement cohérent, en conservant une proximité des postures compressées dans le temps et donc dans l’espace (c’est-à-dire, que deux positions proches dans l’espace réel restent proches dans l’espace latent). Pour ce faire, la phase d’apprentissage de ces méthodes comporte des similitudes avec celle de la méthode DMP, présentée dans la Section 1.7.1 : les paramètres d’une “ fonction objectif ”

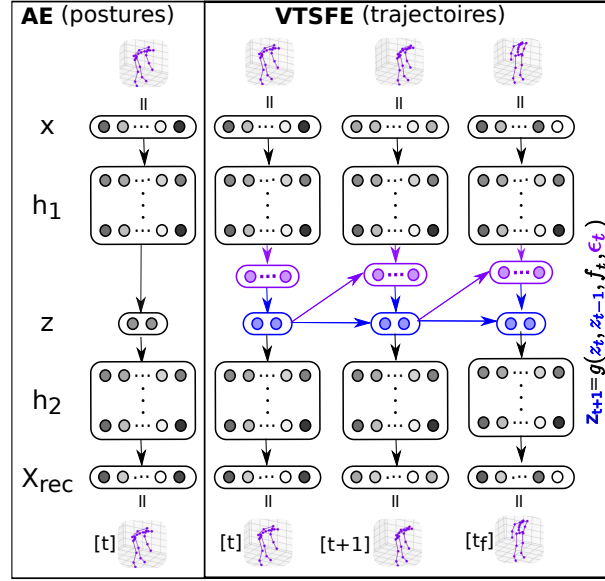


FIGURE 3.6 – Relation entre les méthodes AEs et VTSFE permettant d’encoder des trajectoires dans un espace latent de petite dimension.

(*loss function*, notée f_t) sont optimisés afin de conserver la dynamique des trajectoires réelles dans l’espace latent. Pour cela, cette méthode est composée d’une chaîne de VAEs, dont les pondérations sont communes. La Figure 3.6 schématise cette chaîne. Après apprentissage, ces deux méthodes utilisent un unique VAE indépendant du temps, dont les pondérations correspondent à celles apprises par la chaîne de VAEs. Grâce à ces pondérations, la dynamique des données est conservée dans l’espace latent de ce VAE. Ainsi, deux postures proches dynamiquement dans l’espace réel restent proches dans l’espace latent.

Ces deux méthodes diffèrent dans la construction de leur espace latent, comme le représente les graphes de la Figure 3.7. Nous détaillons ci-dessous les différences et améliorations principales apportées par la méthode VTSFE.

Premièrement, la méthode VTSFE modélise le “bruit de la prédiction”⁴¹ ϵ_t de manière plus précise, par le biais d’une distribution. Cette distribution est calculée à partir de l’erreur empirique, c’est-à-dire qu’elle correspond à la différence entre la prédiction faite par le modèle et les postures réelles. L’encodeur peut alors inférer le bruit ϵ_t à l’aide de l’équation :

$$q_\phi(\epsilon_t | f_t, x_{t+1}, z_t, z_{t-1}) = \mathcal{N}(\mu_{\epsilon,t}, \sigma_{\epsilon,t}^2 I).$$

Avec x_{t+1} la posture suivante, z_t la posture compressée actuelle, et z_{t+1} la suivante. Il s’agit d’une distribution gaussienne.

Deuxièmement, cette méthode ne nécessite plus la connaissance de la position finale de la trajectoire x_T . Cet avantage est nécessaire pour l’étude de la Section 6, puisque la position finale des trajectoires doit pouvoir varier.

Troisièmement, la prédiction des trajectoires est améliorée par la conception de la fonction objectif. Celle-ci, définie en fonction d’une fonction f définie ci-dessous, permet de rendre l’optimisation finale plus proche de l’optimisation théorique réelle indiquée par l’inférence variationnelle. La fonction f est une fonction représentant la dynamique des trajectoires apprises. On peut

41. Ce bruit correspond à l’erreur d’estimation lors de la prédiction.

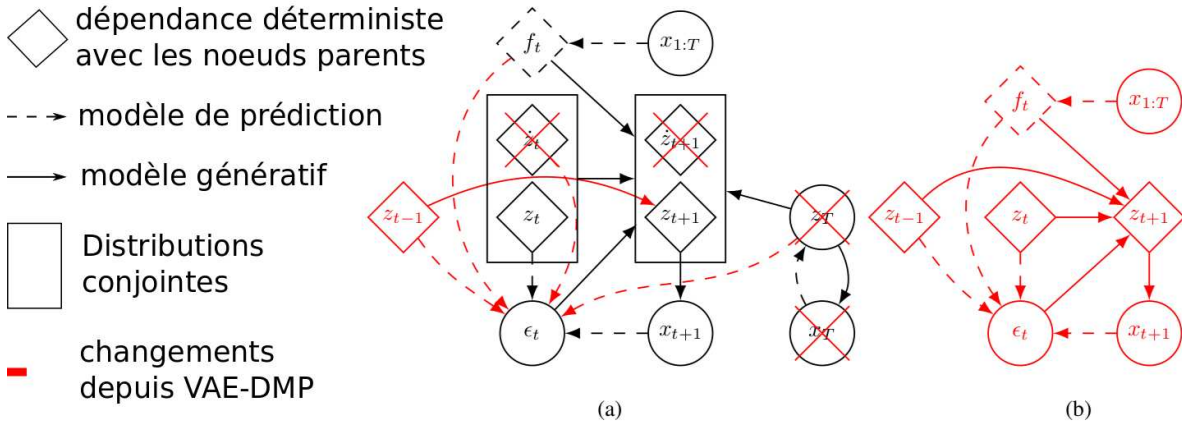


FIGURE 3.7 – *Schéma de comparaison entre les méthodes VTSFE et VAE-DMP.* – La méthode VTSFE consiste en la superposition de deux modèles bayésiens. (a) Les parties en rouge représentent les changements faits depuis la méthode VAE-DMP [48] afin d’obtenir VTSFE. (b) Représentation du modèle VTSFE.

la comparer à une force qui serait appliquée à chaque instant t dans l’espace latent et qui influencerait les trajectoires encodées. Cette fonction est calculée par démonstrations, pour chaque type de mouvement. Après apprentissage, une fois que l’espace latent est fixé, celle-ci n’est plus utilisée. Ainsi, cette fonction force les trajectoires encodées à suivre la même dynamique que les démonstrations.

Finalement, nous pouvons voir que certaines variables ont été supprimées, afin de simplifier le modèle de transition. En effet, ce modèle de transition n’utilise plus le modèle classique DMP, mais un simple modèle d’accélération avec en plus un terme de contrainte $f_t + \epsilon_t$.

Ainsi, on passe de :

$$\begin{pmatrix} z_{t+1} \\ \dot{z}_{t+1} \end{pmatrix} = \begin{pmatrix} 1 - \tau dt^2 \alpha \beta & \tau dt^2 \alpha + dt \\ -\alpha \beta \tau dt & 1 - \alpha \tau dt \end{pmatrix} \begin{pmatrix} z_t \\ \dot{z}_t \end{pmatrix} + b = g(z_t, \dot{z}_t, z_T, f_t, \epsilon_t, \alpha, \beta) \quad (\text{VAE - DMP}).$$

Avec τ une variable permettant de changer la vitesse des trajectoires (temporal scaling term) et $b = (dt, 1)^\top (\alpha \beta z_T + f_t + \epsilon_t) \tau dt$, à simplement :

$$z_{t+1} = (f_t + \epsilon_t) dt^2 + 2z_t - z_{t-1} = g(z_t, z_{t-1}, f_t, \epsilon_t) \quad (\text{VTSFE}).$$

Ce modèle de transition n’a donc plus besoin ni de z_T , ni de \dot{z}_t , ni des paramètres α et β des DMPs, qui doivent normalement être optimisés par d’autres algorithmes.

Après avoir compressé les postures de la trajectoire dans un espace réduit, ces trajectoires compressées sont ensuite modélisées à l’aide de la méthode ProMP. Nous présentons dans la prochaine Section la méthode AE-ProMP, principalement utilisée dans cette thèse.

3.4.3 Prédiction de la poursuite de trajectoires compressées

Les méthodes AE-ProMPs et VTSFE-ProMPs permettent de compresser des distributions de trajectoires dans un espace latent.

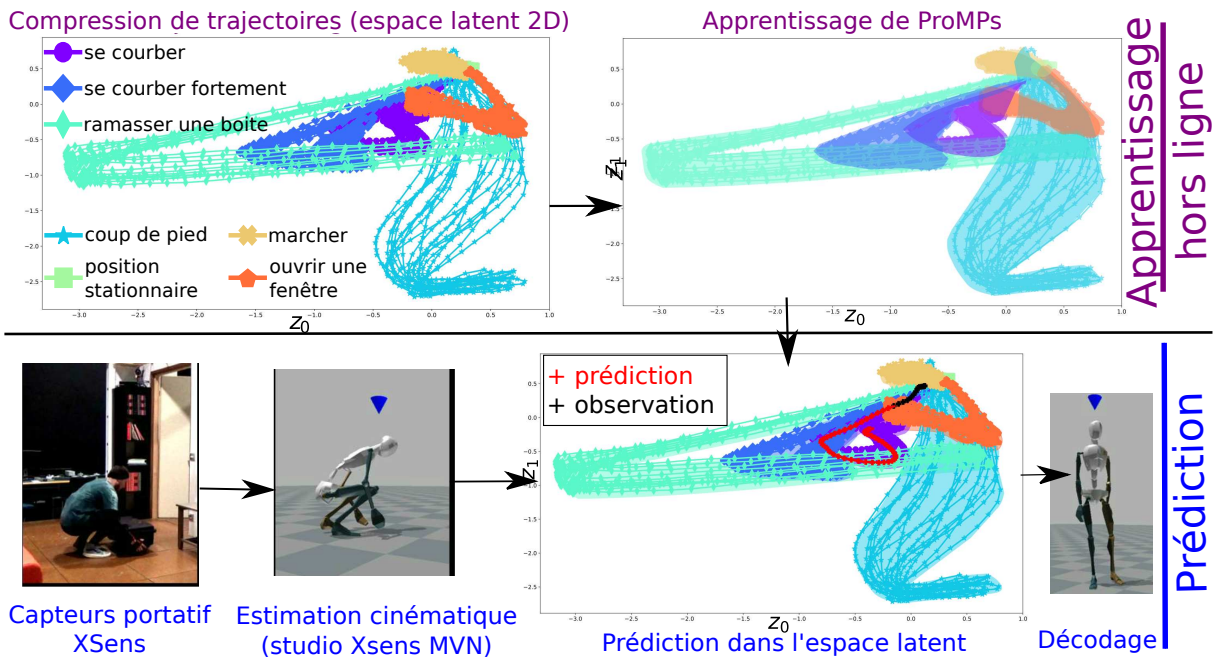


FIGURE 3.8 – *Concept de la méthode de prédiction de trajectoires à grande dimension* – En haut à gauche : les postures sont compressées dans un espace latent à petite dimension (ici à 2 dimensions : z_0 et z_1). – En haut à droite : La méthode ProMP modélise alors ces trajectoires compressées en primitives d’actions. – En bas : À partir de ces distributions (information a priori) et d’observations initiales d’une trajectoire (en noir), la méthode permet de prédire la fin du mouvement (en rouge).

Pour cela, la méthode AE ou VTSFE est utilisée afin qu’un réseau de neurones apprenne à compresser/décompresser des postures. Après apprentissage, des trajectoires de démonstration (c’est-à-dire, une succession de postures) sont alors compressées dans cet espace latent, puis la méthode ProMP est utilisée afin de représenter les primitives de mouvements compressés correspondantes. Ce concept est illustré dans les Figures 3.8 et 3.9.

Ces deux méthodes fonctionnent de la manière suivante :

1. L’auto-encodeur (AE ou VTSFE) est d’abord entraîné, afin de compresser les données originales dans l’espace latent.

La méthode AE encode des postures instantanées, de dimension N , dans un espace latent de dimension R , avec $R \ll N$ (c.f., Section 3.4.1 pour plus de détails).

La méthode VTSFE apprend en plus la dynamique des trajectoires, par l’encodage des trajectoires complètes lors de cette phase d’apprentissage.

Pour cela, les postures représentant chaque instant $t = \{1, \dots, t_f\}$ de la trajectoire sont encodées dans une cascade de $|t_f|$ VAEs avec $|t_f|$, le nombre d’instant utilisé pour représenter la trajectoire. La taille de l’espace latent est alors $R \ll N$, avec N la dimension des postures (c.f., Section 3.4.2 pour plus de détails).

Dans ces deux cas, les trajectoires décodées correspondent alors à :

$$\mathbf{x}_{rec}(t) = dec(enc(\mathbf{x}(t))) = \mathbf{x}(t) + \epsilon_v(t), \forall t \in [1 : t_f].$$

Avec $\epsilon_v(t)$, l'erreur de reconstruction que l'on minimise lors de l'apprentissage.

Ce premier apprentissage est représenté par la première image de la Figure 3.8.

2. Puis, pour N_A actions, les N_A ProMPs correspondantes sont apprises. Cet apprentissage est effectué à l'aide de l'ensemble des trajectoires de démonstration compressées dans l'espace latent :

$$\xi(t) = [z_1(t), \dots, z_R(t)]^\top = \Phi_t \omega + \epsilon_\xi.$$

Ce second apprentissage est représenté par la seconde image de la Figure 3.8.

3. Ces ProMPs sont alors utilisées afin de prédire la poursuite des trajectoires initiées, à partir de l'observation partielle de celles-ci.

La phase de prédiction se déroule alors de la manière suivante :

- (a) Soit $\mathbf{x}^o(t) = [x_1^o(t), \dots, x_N^o(t)]^\top, \forall t \in [1 : n_o]$, les observations de la trajectoire initiée.
- (b) Ces observations sont tout d'abord compressées dans l'espace latent à l'aide de la méthode AE ou VTSFE par :

$$\xi_t^o = enc(\mathbf{x}^o(t)) = [z_1^o(t), \dots, z_R^o(t)]^\top, \forall t \in [1 : n_o].$$

- (c) À partir de la trajectoire partielle ainsi compressée $\Xi^o = \{\xi_1^o, \dots, \xi_{n_o}^o\}$, la méthode ProMP est utilisée afin de prédire sa continuation :

$$\hat{\Xi} = [\hat{\xi}_{n_o+1} \dots \hat{\xi}_{t_f}]^\top.$$

Cette phase de prédiction est représentée par la cinquième image de la Figure 3.8

- (d) Finalement, la méthode AE ou VTSFE permet de décoder la trajectoire compressée prédite, afin d'obtenir la prédiction du mouvement du corps entier futur :

$$dec(\hat{\Xi}) = [\hat{\mathbf{x}}_{n_o+1} \dots \hat{\mathbf{x}}_{t_f}]^\top.$$

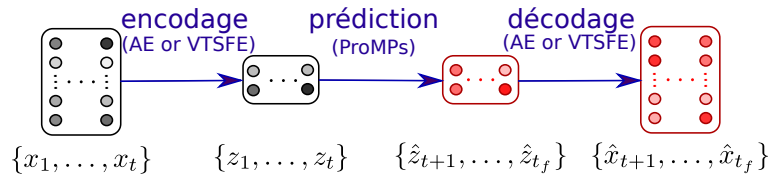


FIGURE 3.9 – Les actions d'encodage et de décodage de AE et VTSFE combinées avec les prédictions effectuées à partir des ProMPs apprises dans l'espace latent.

Ainsi, les méthodes AE-ProMP et VTSFE-ProMP sont très similaires dans leur fonctionnement. La différence principale a lieu lors de l'apprentissage, où l'espace latent est construit de manière à respecter la dynamique des trajectoires (c'est-à-dire, que l'espace latent est dynamiquement plus cohérent). Le but est alors de tester si avec un espace latent plus cohérent dynamiquement, la précision du modèle des trajectoires à grande dimension est améliorée. C'est ce qui sera testé lors de l'étude de la Section 6.

Compréhension de l'interaction physique humain-robot (Étude 1)



FIGURE 4.1 – *Communication physique naturelle entre deux personnes* – À l'aide d'un guidage physique, la personne qui se tient debout aide et demande à la femme assise de se lever. Cette dernière interprète le mouvement effectué (direction, vitesse, etc.) et adapte son mouvement afin qu'il corresponde aux attentes de sa partenaire.

Lors de collaboration entre êtres humains, ceux-ci doivent comprendre l'intention de leur partenaire et agir en conséquence. Pour cela, ils communiquent de différentes manières (c'est-à-

dire, en dialoguant, par des gestes, ou par contact). Dans cette étude, nous nous intéressons à la communication non verbale utilisant l'information haptique provenant du contact entre les individus. La Figure 4.1 et la Vidéo [HHI] montrent un exemple de communication physique naturelle entre deux personnes. Nous exploitons ici ce même mécanisme afin de permettre au robot d'interagir physiquement avec les humains par le biais de prédiction de l'intention. L'état de l'art sur ce type d'interaction est présenté dans la Section 1.3.

Dans cette étude, le robot apprend plusieurs primitives de mouvements de manière probabiliste, en utilisant la méthode ProMPs, décrite dans la Section 3.3. Chaque primitive représente un type de mouvement différent, aboutissant sur une position finale particulière, liée à une tâche correspondante. En utilisant ces ProMPs, le robot est capable de prédire la poursuite d'une trajectoire initiée manuellement par l'utilisateur. À l'aide de cette prédiction, le robot doit alors finir le mouvement initié de manière autonome.

L'expérience se focalise d'abord sur le robot iCub simulé dans *Gazebo* (c.f., Section 2.2) puis sur le robot réel.

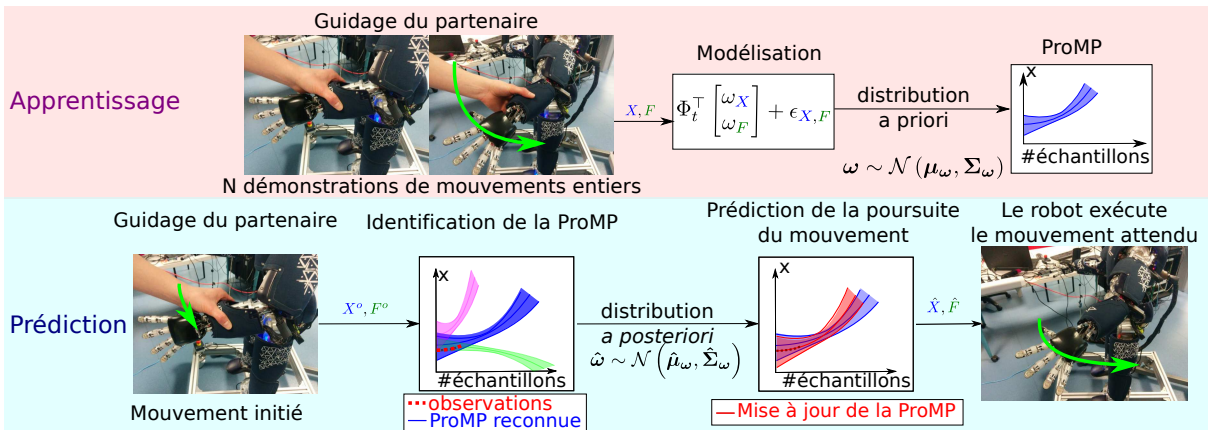


FIGURE 4.2 – **Concept d'utilisation des ProMPs** – Le robot prédit la trajectoire à effectuer. En haut : phase d'apprentissage, où les ProMPs sont apprises à partir de plusieurs démonstrations guidées par un humain. En bas : phase de prédiction, où le robot reconnaît, à partir d'un mouvement initié par son partenaire, la ProMP courante et où il prédit l'intention de son partenaire (c'est-à-dire, l'évolution future de la trajectoire initiée).

Notre approche est représentée dans la Figure 4.2. Dans la partie supérieure, on représente l'étape d'apprentissage d'une primitive : l'utilisateur commence par guider physiquement le robot afin qu'il effectue le mouvement dans sa totalité. Cela est répété N fois afin d'acquérir N démonstrations. Pendant que l'utilisateur effectue ces mouvements de démonstration, le robot collecte des mesures cinématiques (*par ex.*, positions cartésiennes) et dynamiques (*par ex.*, couples). Le robot utilise alors ces N trajectoires pour apprendre les ProMPs. Pour cela, il calcule des modèles de ces mouvements (c.f., Équation 3.4) puis de ces modèles, il apprend une ProMP, qui correspond à une *distribution a priori* du type de mouvement que l'utilisateur est susceptible de lui demander d'effectuer (c.f., Équation 3.5).

Dans le cas où le robot apprend différentes tâches, ce processus d'apprentissage est répliqué afin d'avoir une ProMP pour chaque tâche.

En utilisant ces ProMPs, le robot est capable d'inférer qu'elles sont les tâches correspondant

aux mouvements initiés par l'utilisateur et de prédire la trajectoire future qu'il doit effectuer, comme présenté dans la partie inférieure de la figure.

Nous formulons ci-dessous le problème de reconnaissance de l'intention, comme le problème d'estimer la continuation ou future d'une trajectoire à partir d'observations initiales d'un geste initié par un humain. Nous l'appliquons au cas du robot *iCub*, qui doit alors apprendre des primitives de mouvement définies par les positions cartésiennes en $3D$ de la main gauche du robot et des forces/couples $6D$ qu'il reçoit sur cette main, lors de ces mouvements.

4.1 Cadre théorique adapté à l'apprentissage de trois primitives multidimensionnelles

Dans cette section, la méthode présentée dans la Section 3.3 est adaptée afin que les ProMPs modélisent plusieurs types de données provenant des trajectoires. Pour cela, ces données sont enregistrées dans un vecteur paramètre de dimension $D = 9$:

$$\forall t, \xi_t = \begin{bmatrix} X_t F_t \end{bmatrix} \in \mathbb{R}^9.$$

Avec $X_t \in \mathbb{R}^3$ correspond à la position cartésienne de la main gauche du robot *iCub* et $F_t \in \mathbb{R}^6$ les forces/couples qu'il reçoit sur sa main, provenant notamment des forces de contact exercées par l'utilisateur.

La modélisation d'une telle ProMP correspond à :

$$\xi_t = \begin{bmatrix} X_t F_t \end{bmatrix} = \Phi_{\alpha t} \omega + \epsilon_t.$$

Avec $\omega \in \mathbb{R}^{D \cdot M}$, le vecteur paramètre indépendant du temps ; $\epsilon_t = \begin{bmatrix} \epsilon_{X_t} \epsilon_{F_t} \end{bmatrix} \in \mathbb{R}^D$, le bruit blanc gaussien ; et $\Phi_{\alpha t} \in \mathbb{R}^{D \times D \cdot M}$, la matrice contenant la valeur des *RBFs* à l'instant αt .

Puisqu'il s'agit d'un cas multidimensionnel, cette matrice est diagonale par bloc $\Phi_{\alpha t}$, définie par :

$$\Phi_{\alpha t} = \begin{bmatrix} \phi_1(\alpha t) & & 0 \\ & \ddots & \\ 0 & & \phi_D(\alpha t) \end{bmatrix} \in \mathbb{R}^{D \times D \cdot M}.$$

Il s'agit d'une matrice diagonale composée de D ensembles de M fonctions gaussiennes. Chaque ensemble représente l'une des dimensions du vecteur ξ_t et leur M fonctions sont éparpillées dans le temps, afin d'être à valeur sur $[0 : \bar{s}]$. Pour plus d'information, voir l'Annexe B.

4.1.1 Apprentissage de Primitives de Mouvement

Durant la phase d'apprentissage de chaque primitive de mouvement $k \in [1 : K]$ ($K = 3$ dans nos expériences), le robot observe différentes trajectoires $S_k = \{\Xi_1, \dots, \Xi_n\}_k$ (*c.f.*, Section A.1.2). Pour chacune des trajectoires $\Xi_{i[1:t_{f_i}]} = [\xi_i(1), \dots, \xi_i(t_{f_i})]^\top$, le robot calcule le vecteur paramètre optimal ω_{ki} , afin que la modélisation approxime au mieux la trajectoire (*c.f.*, Section 3.3.5).

L'Équation 3.3 de cette Section est adaptée dans le logiciel développé pour cette étude (*c.f.*, [soft.Pred.Mult]), afin d'effectuer un calcul matriciel à la place de t_{f_i} calculs itératifs. Ainsi, le calcul devient :

$$\omega_{ki} = (\Phi_{\alpha[1:t_{f_i}]}^\top \Phi_{\alpha[1:t_{f_i}]})^{-1} \Phi_{\alpha[1:t_{f_i}]}^\top * \Xi_{i[1:t_{f_i}]} \quad (4.1)$$

Avec $\Phi_{\alpha[1:t_{f_i}]} = [\Phi_{\alpha 1}, \Phi_{\alpha 2} \dots, \Phi_{\alpha t_{f_i}}]^\top$.

4.1.2 Prédiction de l'évolution de la trajectoire à partir d'observations initiales

Après cette étape d'apprentissage, le robot peut prédire la poursuite de mouvements initiés. Pour cela, les calculs présentés dans les Sections 3.3.3, 3.3.4 et 3.3.5 sont adaptés à l'étude actuelle de la manière suivante.

Soit $\Xi^o = [X^o F^o] = [\Xi_1 \dots \Xi_{n_o}]^\top$ les premières observations de la trajectoire initiée.

Tout d'abord, le robot reconnaît quel type de mouvement est initié (c'est-à-dire, quelle est la ProMP correspondante) uniquement à partir des mesures de la position de sa main, puisqu'une même trajectoire peut être effectuée et reconnaissable quel que soit son profil de forces. Ainsi, pour cette étape de reconnaissance, seule une sous-partie de ces observations est considérée : $X^o = [X_1 \dots X_{n_o}]^\top$.

À partir de cette observation partielle de la trajectoire, le robot estime à la fois le paramètre de modulation du temps (*c.f.*, Section 3.3.4, donnant l'information de la vitesse et de la durée totale de la trajectoire \hat{t}_f) et la ProMP courante $\hat{k} \in [1 : 3]$ (*c.f.*, Section 3.3.5).

Puis, l'ensemble des observations initiales Ξ^o est utilisé afin de mettre à jour la ProMP reconnue (*c.f.*, Section 3.3.3). Pour ce faire, un calcul matriciel est effectué, basé sur l'Équation 3.14 :

$$\begin{cases} \hat{\mu}_{\omega_k} &= \mu_{\omega_k} + K(\Xi^o - \Phi_{\alpha[1:n_o]}\mu_{\omega_k}) \\ \hat{\Sigma}_{\omega_k} &= \Sigma_{\omega_k} - K(\Phi_{\alpha[1:n_o]}\Sigma_{\omega_k}) \\ K &= \Sigma_{\omega_k} \Phi_{\alpha[1:n_o]}^\top (\Sigma_{\xi^o} + \Phi_{\alpha[1:n_o]}\Sigma_{\omega_k}\Phi_{\alpha[1:n_o]}^\top)^{-1} \end{cases}$$

À partir de cette distribution a posteriori des paramètres $\omega_k \sim p(\hat{\mu}_{\omega_k}, \hat{\Sigma}_{\omega_k})$ et de la ProMP a posteriori associée $\mathcal{N}(\Xi | \Phi_{\alpha[1:\hat{t}_f]}\hat{\mu}_{\omega_k}, \Phi_{\alpha[1:\hat{t}_f]}\hat{\Sigma}_{\omega_k}\Phi_{\alpha[1:\hat{t}_f]}^\top + \Sigma_\Xi)$, la trajectoire prédite $\hat{\Xi} = \{\hat{\xi}_1, \dots, \hat{\xi}_{\hat{t}_f}\}$ est récupérée par :

$$\forall t \in [1 : \hat{t}_f], \hat{\xi}_t = \begin{bmatrix} \hat{X}_t \\ \hat{F}_t \end{bmatrix} = \Phi_{\alpha t} \hat{\mu}_{\omega_k}$$

Notons que lors de nos expériences, les forces/couples (*wrench*) simulées dans Gazebo à partir du dispositif haptique étaient peu précis. C'est pourquoi, dans l'exemple où le robot est simulé, ces informations ne sont pas utilisées, contrairement à l'exemple sur robot réel, où l'intérêt d'utiliser ces mesures est mis en avant (*c.f.*, Section 4.3).

4.2 Application à l'aide de la simulation du robot

L'expérience se focalise d'abord sur le robot simulé dans *Gazebo*. Les trois positions à atteindre sont placées à différentes hauteurs, en face du robot. La Figure 4.3 représente les positions but par trois balles colorées.

Mise en place de l'expérience

Pour effectuer cette expérience, pour chaque type de mouvement, on constitue un ensemble de 40 trajectoires de démonstration, en faisant intervenir 4 personnes différentes (soit 10 par personne). Pour chaque test, chaque ensemble de trajectoires est divisé en deux, le robot utilise la première partie afin d'apprendre la ProMP correspondant au type de mouvements, et la seconde afin de tester la robustesse des prédictions (nommées « *trajectoires tests* »). Cette seconde partie correspond à une trajectoire sélectionnée de manière aléatoire, suivant la méthode du « *leave-one-out cross-validation* ». Ainsi, pour chaque primitive de mouvement et pour chaque test, 39 trajectoires de démonstration sont utilisées lors de la phase d'apprentissage et une dernière permet d'estimer les performances de la prédiction.

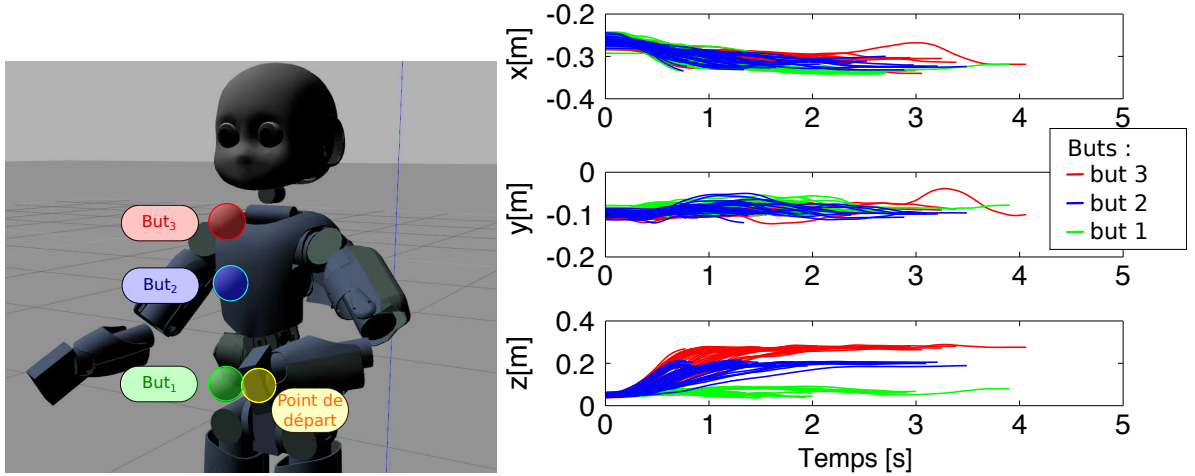


FIGURE 4.3 – *Étude en simulation – Scénario.* La figure de gauche présente les trois buts colorés que le robot doit atteindre depuis sa position de départ; le robot utilise les trajectoires permettant d'atteindre ces positions afin d'apprendre trois primitives, représentant trois actions. Les courbes présentées dans la figure de droite correspondent aux informations cartésiennes de ces trajectoires.

Le nombre d'échantillons de référence choisi pour ce test est $\bar{s} = 100$. De plus, $M = 5$ *RBFs* sont utilisées pour représenter chaque donnée d'entrée de la ProMP, où chaque *RBF* ψ_i a pour paramètres $c(i) = \frac{i}{M}$, $h(i) = \frac{1}{M^2}$. Le bruit des mesures est ajusté selon le but désiré. Dans notre cas, il s'agit de forcer le robot à passer par les observations initiales de la trajectoire de manière précise, donc l'estimation de ce bruit doit être petit : $\epsilon_t = 0.00001$.

La Figure 4.4 représente les trois ProMPs permettant d'atteindre ces trois positions but. On voit bien qu'elles représentent des types de trajectoires différentes, car elles sont réparties à différentes hauteurs. Afin de clarifier cette figure, seul l'axe z de la position cartésienne est présenté.

Apprentissage

Lors de chaque trajectoire de démonstration, le robot enregistre les positions cartésiennes de sa main $X(t) \in \mathbb{R}^3$ et la durée totale de la trajectoire t_f . Puis, il calcule le paramètre de modulation du temps qui correspond à cette durée $\alpha = \frac{\bar{s}}{t_f}$ et les ProMPs qui correspondent à ces trajectoires.

Prédiction

Après avoir appris les trois primitives de mouvement, le robot doit prédire la fin des trajectoires initiées par l'utilisateur. Afin de tester cette capacité de prédiction, on fournit au robot un début de la trajectoire de démonstration (celle qui n'a pas été utilisée lors de l'apprentissage). Ce début de trajectoire correspond à un nombre n_o d'observations, choisi afin qu'il représente un pourcentage de la trajectoire totale désirée : $n_o = \left\lfloor \frac{\text{nbPourcentageTest}}{100} * t_f \right\rfloor$ où t_f correspond à la durée totale du mouvement testé.

Pour inférer la durée totale de la trajectoire initiée, quatre méthodes sont proposées afin que le robot infère le paramètre de modulation du temps $\hat{\alpha}$. Ces méthodes ont été présentées dans la Section 3.3.4. Dans l'expérience actuelle, chacune de ces méthodes est testée afin de comparer leur performance.

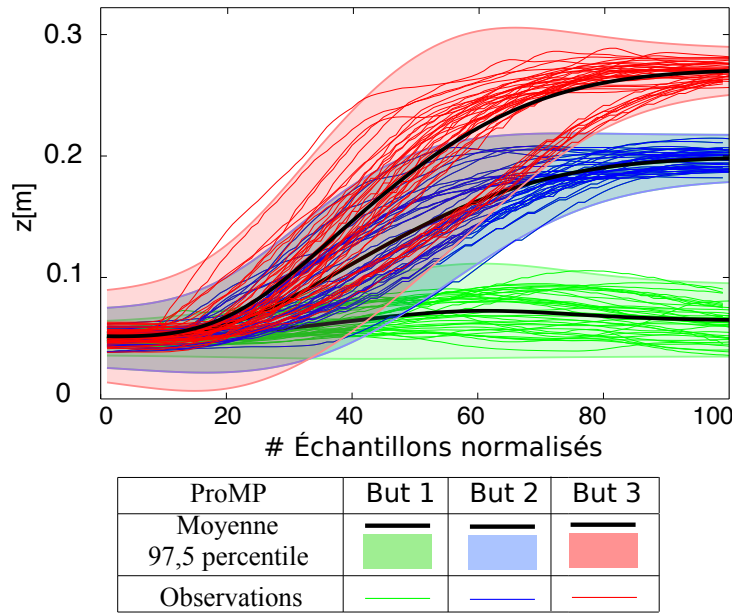


FIGURE 4.4 – *Étude en simulation – Exemple 1D des 3 ProMPs apprises.* Représentation de l'axe z de la position cartésienne de la main gauche du robot, pour les trois ProMPs permettant d'atteindre des buts positionnés à différentes hauteurs. Chaque ProMP est créée à partir de 39 trajectoires de démonstration et où chaque dimension des données de cette trajectoire sont représentées par $M = 5$ fonctions de base radiales (ici des Gaussiennes), paramétrées par $c = \frac{1}{M}$, $h = \frac{1}{M^2}$ et $\bar{s} = 100$.

La Figure 4.5 représente l'erreur moyenne de l'estimation de ce paramètre $\hat{\alpha}$ lors de 10 tests de prédiction, en fonction à la fois du pourcentage d'observations de la trajectoire (de 30% à 90% de la trajectoire totale) et de la méthode utilisée.

La méthode la plus basique est utilisée en référence (courbes noires) : elle consiste à effectuer la moyenne des paramètres α observés lors de l'apprentissage. Par rapport à cette méthode, cette figure montre que les autres sont plus précises. De plus, lors de l'observation des premiers instants de la trajectoire (c'est-à-dire, 30%-50% d'observations de la trajectoire totale), la méthode qui utilise une modélisation du paramètre α est la plus performante. Ainsi, comme le montre le graphe de gauche, c'est cette même méthode qui permet d'inférer au mieux la trajectoire désirée. De plus, on peut remarquer que la méthode du maximum de vraisemblance est aussi performante, or elle a aussi l'avantage d'être moins spécifique aux données (en effet, la méthode de modélisation se base sur un a priori sur les trajectoires apprises). C'est pourquoi on utilise la méthode de *modélisation* pour nos expériences, lorsqu'il s'agit de reconnaître la trajectoire à partir de la position cartésienne du robot. Lorsqu'il s'agit d'autres informations, telles que les forces/couples exercées sur le bras du robot, on compare cette méthode avec celle du maximum de vraisemblance.

4.3 Application sur le robot *iCub* réel

Dans cette section, nous nous intéressons à deux expériences effectuées sur le robot *iCub* réel.

La première expérience s'inspire de celle précédente (*c.f.*, Section 4.2), où les "tâches" que le robot doit effectuer sont représentées par des trajectoires de démonstration dirigées vers des buts distincts. Un collaborateur humain guide ces démonstrations. Cette expérience permet

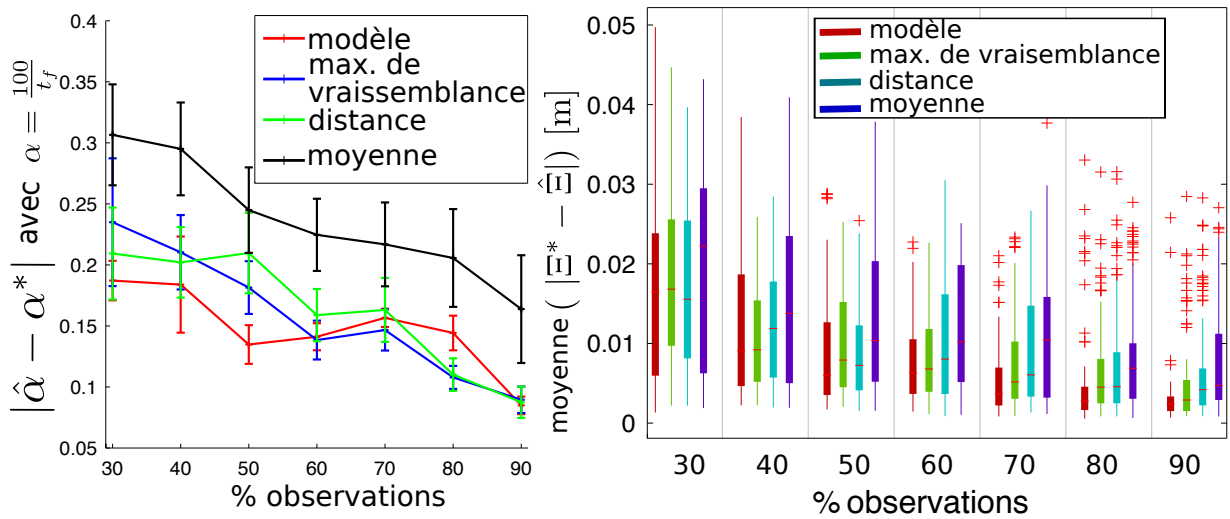


FIGURE 4.5 – *Étude en simulation – Résultats.* Le graphique de gauche représente l'erreur d'estimation du paramètre α . Celui de droite, l'erreur de la prédiction de la trajectoire, suivant le nombre d'observations connues par le robot et la méthode utilisée. Dans chaque cas, 10 tests sont effectués.

d'explorer comment les informations sur les forces et les couples peuvent permettre d'améliorer la reconnaissance des trajectoires.

Dans un premier temps, afin de pouvoir évaluer la qualité des prédictions, les trajectoires tests présentées partiellement au robot sont en réalité connues dans leur totalité, afin d'avoir une "vérité terrain".

La seconde expérience consiste en un scénario collaboratif plus réaliste, inspiré par le tri d'objets de manière collaborative.

Dans de telles applications, le robot permet de soulever des objets qui peuvent être lourds, dangereux, ou que l'humain ne peut pas manipuler (*par ex.*, des objets chimiques ou de la nourriture). Le partenaire humain inspecte alors l'objet et décide s'il est accepté ou rejeté. Selon cette décision, l'objet est mis dans une corbeille positionnée en face du robot, ou dans une poubelle placée à côté du robot.

Notons que les trajectoires de démonstration se font en manipulant le bras de l'*iCub*, ce qui n'est pas évident dus à la limitation de son espace opérationnel et au fait qu'il ne peut pas porter d'objets lourds. C'est pourquoi l'expérience a été simplifiée. Elle correspond alors au déplacement de petits objets et met en jeux deux corbeilles. Le partenaire humain n'a alors qu'à initier un mouvement en guidant le bras du robot, puis de le lâcher pour que le robot finisse le mouvement à accomplir par lui-même.

Pour garantir la sécurité du robot, les trajectoires prédites sont alors présentées sur une interface graphique représentant le monde du robot, puis sont validées sur-le-champ par l'opérateur humain afin que le robot effectue le mouvement.

Pour des scénarios de collaboration plus compliqués, les tâches pourront être des tâches élémentaires telles que pointer, attraper, atteindre, ou encore manipuler des objets. Dans cette étude, nous n'avons pas effectué ce type de tâche, puisque ces tâches sont lancées automatiquement, selon le type de primitive de mouvement reconnu.

4.3.1 Apprentissage de trois ProMPs avec informations sur les forces/couples

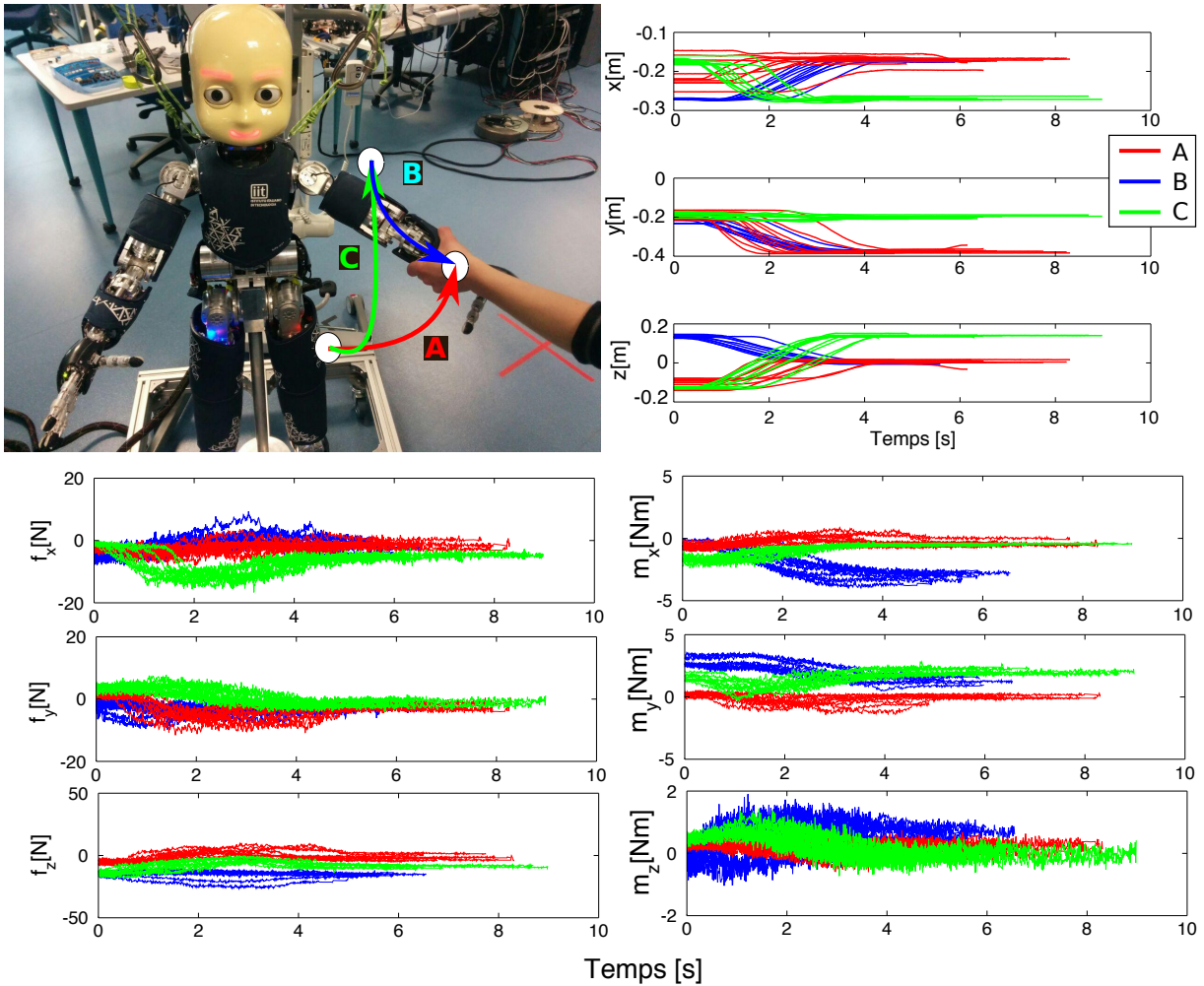


FIGURE 4.6 – 1^{ère} étude : robot réel – Scénario. En haut à gauche : présentation des trois positions buts que doit atteindre le robot, avec les trajectoires correspondantes qui définissent les trois tâches A-B-C. En haut à droite : positions cartésiennes des trajectoires de démonstrations pour ces trois tâches. En bas de gauche à droite : les forces et moments des trajectoires de démonstration.

Dans ce test, les trajectoires sont définies à la fois par les positions cartésiennes de la main du robot et par par les forces et les couples reçus par le robot au niveau de l'interaction physique humain-robot (c'est-à-dire, au niveau du bras). Cet ajout de données est utile dans le cas de primitives de mouvements collaboratifs puisque cela permet au robot de prendre en compte l'interaction physique entre lui-même et l'utilisateur.

Notons que lors de la prédiction, les valeurs des forces/couples \hat{F}_t que prédit le robot correspondent à celles qu'il devrait percevoir si le partenaire le guidait manuellement pendant tout le mouvement. En effet, ces données sont calculées à partir des trajectoires de démonstrations, où l'utilisateur guidait le robot jusqu'au bout du mouvement, pour lui permettre d'apprendre les différentes primitives de mouvement. La prédiction des trajectoires des forces/couples a différents avantages, selon l'application visée. Par exemple, le robot peut utiliser ces informations afin

de détecter des forces anormales qui lui sont appliquées pendant son mouvement. On pourrait alors lui permettre d'adapter son mouvement à de nouveaux environnements (*par ex.*, évitement d'obstacles) ou de comprendre qu'il doit apprendre de nouvelles trajectoires (le partenaire reprend le contrôle pour lui apprendre un nouveau mouvement). Les prochains paragraphes concernent la réalisation de l'expérience, où le robot réel prédit l'intention de l'utilisateur qui amorce un mouvement avec son bras.

Démonstrations

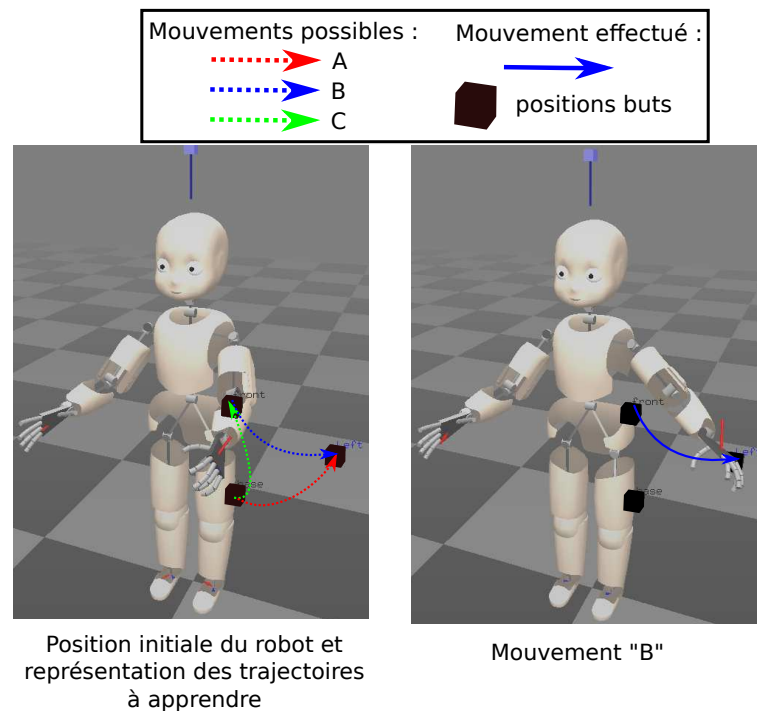


FIGURE 4.7 – Représentation du *iCub* dans *iCubGui* et des trajectoires de l'expérience.

Le robot apprend trois trajectoires, représentant chacune des tâches différentes. La Figure 4.6 représente ces trajectoires. La première, représentée en rouge, démarre devant le robot et va vers la gauche (tâche A). La seconde, représentée en vert, démarre de la même position et finit en hauteur (tâche C). Finalement la dernière, représentée en bleu, démarre en hauteur et qui finit sur la gauche, à la même position finale que la trajectoire précédente (tâche B).

Pour effectuer des trajectoires de démonstrations de ces mouvements, le collaborateur humain utilise une représentation de ces trois buts dans l'interface "*iCub_GUI*". Il s'agit d'un module de l'*iCub* qui présente l'état en temps réel du robot et qui représente avec des flèches les forces qu'il reçoit. La Figure 4.7 représente cette interface, où les carrés représentent les trois positions buts que l'utilisateur doit atteindre en manipulant le bras du robot.

Notons qu'un des challenges lors de la reconnaissance de tels mouvements, est que les utilisateurs novices du robot *iCub* ont des difficultés à manipuler le bras du robot et à lui faire exécuter des trajectoires complexes en 3D [119]. De plus, même expérimenté, l'utilisateur peut effectuer un même mouvement de différentes façons, puisque les bras du robot sont composés de beaucoup d'articulations. La méthode ProMP permet de répondre à ce problème, puisqu'elle prend en compte la variabilité des démonstrations.

Dans cette expérience, le robot récupère 10 trajectoires de démonstrations par primitive de mouvement, toutes fournies par un même utilisateur. Les données enregistrées correspondent à la position cartésienne de la main gauche du robot ainsi que les forces/couples que celle-ci reçoit lors de ces mouvements. Lorsque le robot observe une trajectoire, il mesure les données de manière régulière (toutes les 0.01 secondes).

Apprentissage des ProMPs

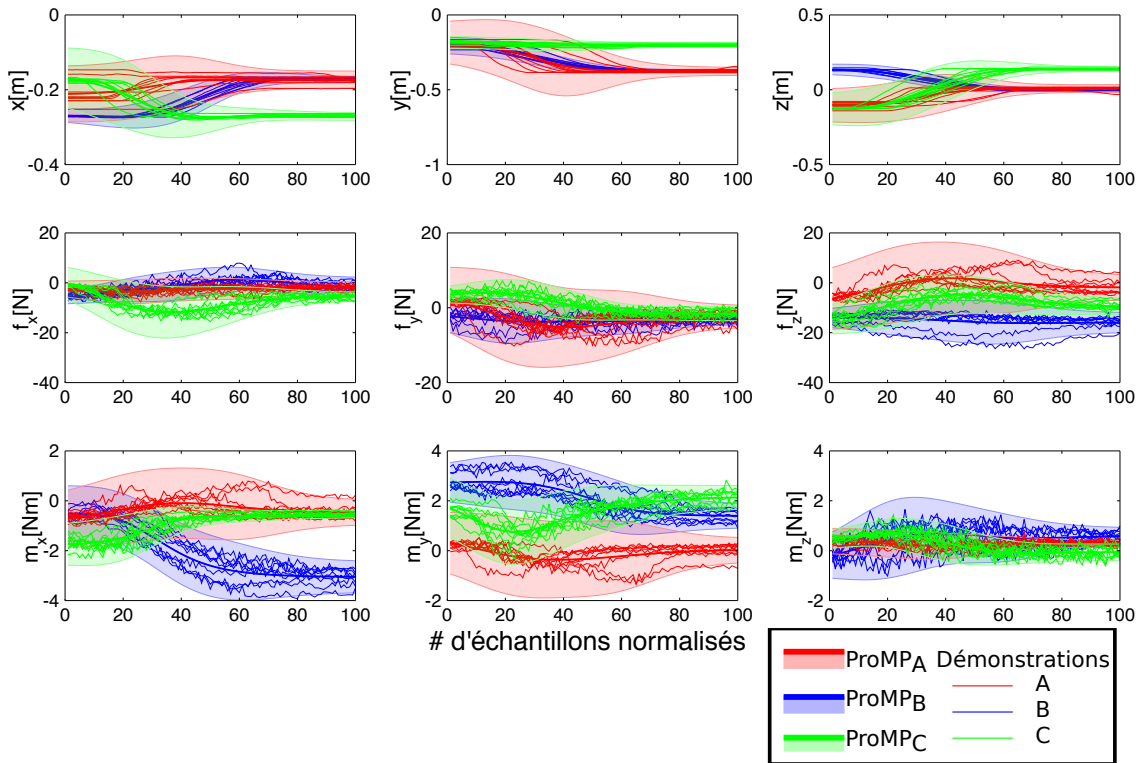


FIGURE 4.8 – 1^{ère} étude robot réel – Représentation des ProMPs apprises. par le robot à partir des démonstrations représentées dans la Figure 4.6.

Après avoir récupéré les trajectoires de démonstration, le robot apprend les ProMPs correspondantes. Celles-ci sont représentées dans la Figure 4.8. Dans cette figure, les distributions sont visiblement superposées :

- Le long de toutes les trajectoires, en ce qui concerne les informations des forces/couples.
- Lors des premiers instants des trajectoires (< 40%), en ce qui concerne l'information des positions cartésiennes.

Ainsi, on voit que l'information des forces semble peu utile pour identifier les trajectoires. Cependant, cette information permet au robot de connaître l'amplitude des forces/couples qu'il est supposé mesurer et nous avons vu précédemment différents cas où cette connaissance peut être utile.

Prédiction de la poursuite des trajectoires initiées

Après cette étape d'apprentissage, l'utilisateur choisi la ProMP à tester et le pourcentage d'observations de la trajectoire totale à partir de laquelle le robot doit prédire la poursuite du

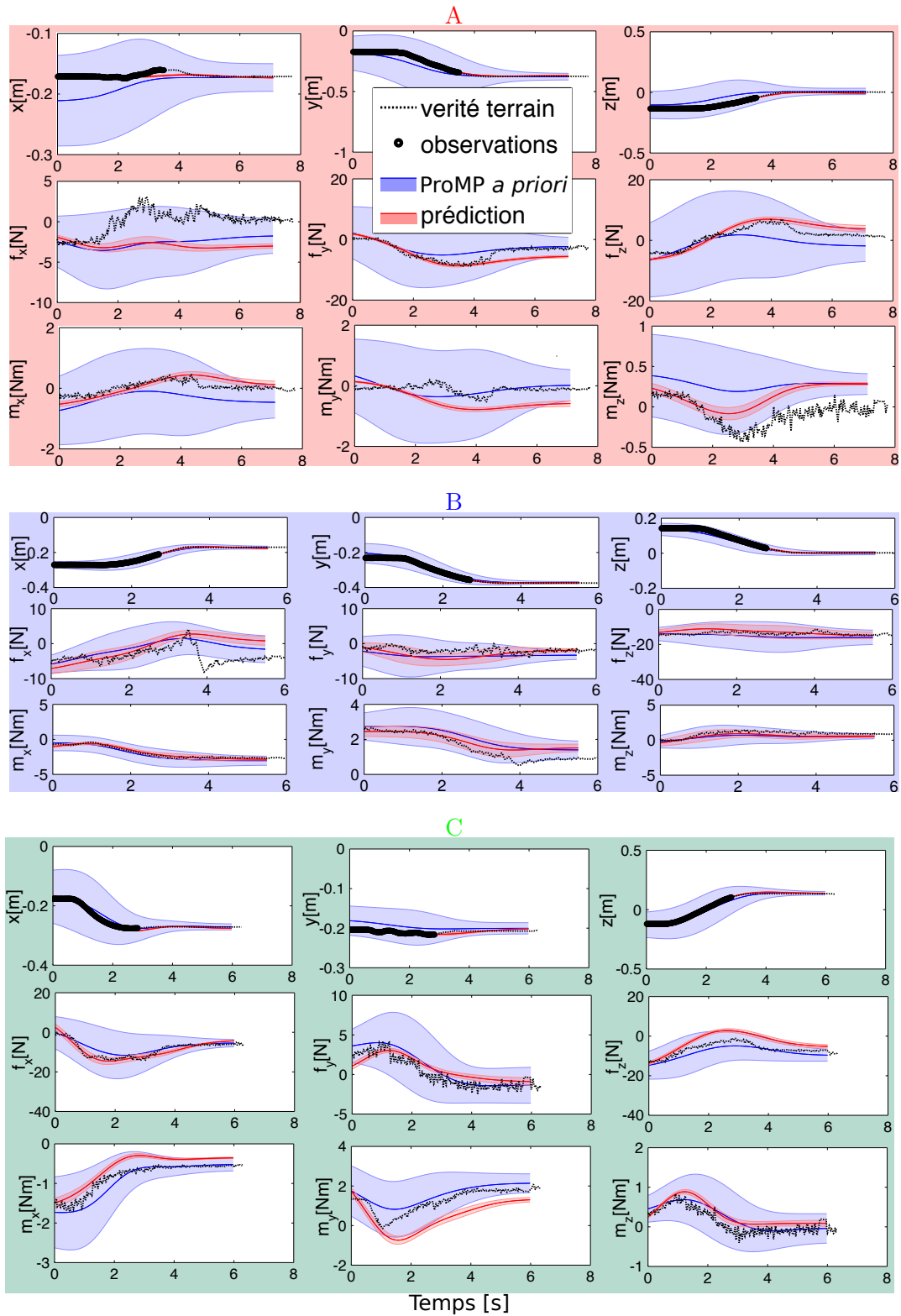


FIGURE 4.9 – 1^{ère} étude robot réel – Exemple de prédictions. Prédiction de trajectoires futures après l'observation partielle (40%) de celles-ci (une par type de mouvement).

Cette prédiction est effectuée à l'aide des ProMPs définies par leurs positions cartésiennes.

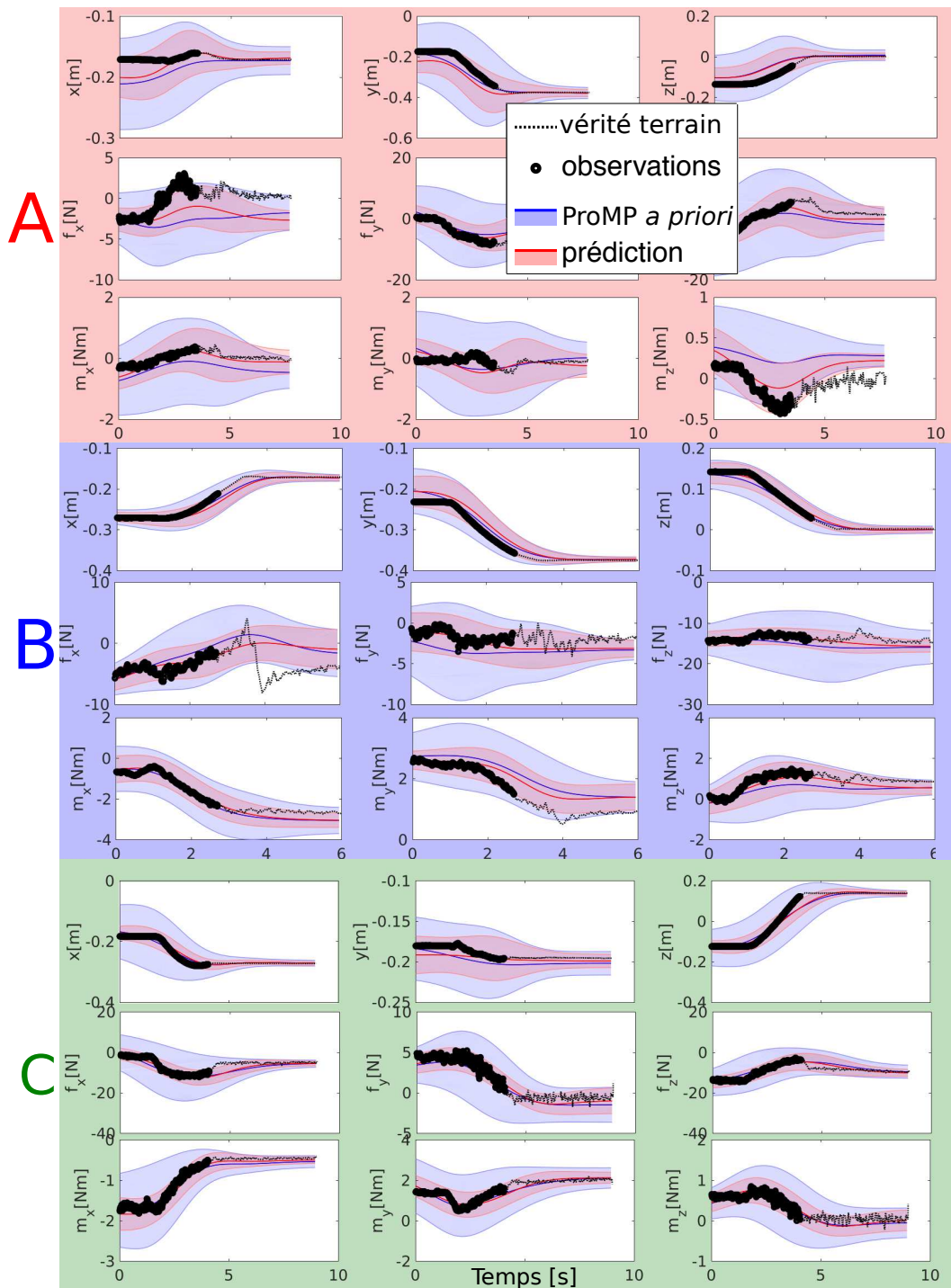


FIGURE 4.10 – 1^{ère} étude robot réel – Exemple de prédictions prenant en compte les forces. Prédiction de trajectoires futures après l'observation partielle (40%) de celles-ci (une par type de mouvement).

Cette prédiction est effectuée à l'aide des ProMPs définies par leurs positions cartésiennes et les forces/couples externes.

mouvement. Le logiciel développé calcule alors le nombre d’observations n_o qui correspond à ce pourcentage. Notons que n_o n’est pas identique pour toutes les trajectoires testées, puisque ce nombre dépend de leur durée totale.

Le robot infère alors à la fois le paramètre de modulation du temps α à partir de la méthode de modélisation du paramètre α (*c.f.*, Section 3.3.4) et la ProMP correspondante.

Puis, le robot prédit la poursuite de la trajectoire, dont sa position finale.

La Figure 4.9 représente la prédiction de trois trajectoires testées (chacune correspondant à des tâches différentes), sans prendre en compte l’information des forces/couples.

Rappelons que ces tests sont effectués de manière hors ligne, car lors de l’évaluation d’un test, on compare la trajectoire prédite avec sa vérité terrain (c’est-à-dire, la trajectoire réelle, désirée par l’utilisateur). Or, cette vérité terrain ne peut être connue que si celle-ci a déjà été effectuée en amont.

Dans cette figure, ces vérités terrains sont représentées par des petits points noirs et les portions partielles de ces trajectoires utilisées lors des prédictions (c’est-à-dire, les “observations initiales”) sont représentées par des gros points noirs. Les courbes rouges ainsi que leurs parties ombrées représentent la ProMP *a posteriori* (après prédiction) calculée par le robot et les courbes bleues ainsi que leurs parties ombrées représentent la ProMP *a priori* (avant la prédiction).

Dans ces tests, on peut voir que les trajectoires cartésiennes des mouvements sont correctement prédites, même s’il y a une erreur d’une seconde en ce qui concerne l’estimation de la durée totale de la trajectoire correspondant à la tâche A. Par contre, la prédiction des trajectoires des forces/couples n’est pas précise. Cela est possiblement dû au fait que les variations des forces/couples ne sont pas corrélées aux variations des positions cartésiennes de la main du robot or, dans ces tests, le robot prédit les trajectoires à l’aide de l’information des positions cartésiennes seule, sans l’information provenant des forces/couples.

La Figure 4.10 quant à elle, représente ce même type de tests, mais en prenant cette fois-ci en compte l’information des forces/couples. Il est notable que dans ce cas, l’estimation de la durée des trajectoires est plus précise. Une hypothèse découlant de ce résultat est que, sachant que l’accélération de la trajectoire peut être calculée à partir des forces, l’information des forces contient donc implicitement l’information de la vitesse de la trajectoire et donc sa durée totale. Par contre, la position cartésienne prédite est un peu moins précise, puisque la distribution de la ProMP est mise à jour en faisant un compromis entre les mesures des positions cartésiennes et celles des forces/couples.

Afin de confirmer ces résultats, nous avons effectué plusieurs tests permettant d’analyser la qualité des prédictions des trajectoires et de l’estimation des paramètres α , en fonction de la quantité d’observations mesurées par le robot (de 30 à 90% de la trajectoire totale). Pour chacun de ces pourcentages, 20 tests ont été effectués, avec ou sans données sur les forces et les moments. Chaque graphique de la Figure 4.11 représente les erreurs lors de ces 20 tests. En haut, le critère d’erreur correspond à la distance moyenne entre la trajectoire prédite et celle réelle. Ce graphique montre que la prédiction de la trajectoire cartésienne est plus précise lorsque le robot ne prend pas en compte l’information des forces et des couples.

En bas, le critère d’erreur correspond à la distance moyenne entre l’estimation du paramètre α et sa vraie valeur. On remarque ici aussi que les données des forces et des couples permettent d’améliorer l’inférence du paramètre α .

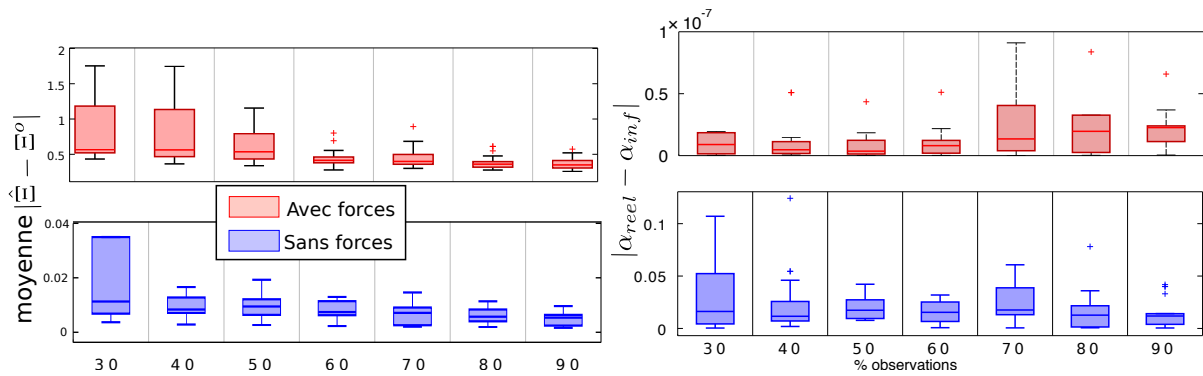


FIGURE 4.11 – 1^{ère} étude : robot réel – Résultat avec ou sans l'information des forces. À gauche : erreur de prédiction des trajectoires ; À droite : erreur d'estimation du paramètre de modulation du temps des trajectoires prédites. En rose avec et en bleu sans prendre en compte l'information des forces et des couples. Ces boîtes à moustaches sont créées à partir de tests effectués sur tous les types de trajectoires, avec le robot *iCub* réel (c.f., Figure 4.8).

Ainsi, ces tests confirment les résultats préliminaires provenant des Figures 4.9 et 4.10.

LeTableau 4.1 retranscrit ces résultats et fournit les valeurs de médianes, moyennes et variances de ces erreurs de prédiction. En ce qui concerne l'inférence du paramètre de modulation du temps, l'erreur correspond à l'erreur quadratique moyenne (*NRMSE*) de $|\alpha_{\text{prediction}} - \alpha_{\text{real}}|$ et pour la prédiction de la trajectoire, le *NRMSE* de $|\Xi_{\text{prediction}} - \Xi_{\text{real}}|$.

De plus, ce tableau reporte les temps moyens de calcul nécessaires à l'inférence du paramètre de modulation du temps ainsi qu'au calcul de la distribution a posteriori. Les calculs sont effectués sur Matlab, sur un ordinateur à un coeur (c'est-à-dire, sans parallélisation).

Ces résultats montrent que les temps nécessaires aux calculs n'utilisant pas les informations des forces/couples sont suffisamment faibles pour des applications en temps réel souple, alors qu'ils deviennent trop longs lorsqu'elles sont utilisées.

Basés sur ces résultats, les prochaines expériences respecteront les choix suivants :

- Dans l'expérience suivante présentée dans la Section 4.3.2, nous utilisons l'information des couples et des forces uniquement lors de l'estimation du paramètre de modulation du temps α .

Cela permet d'avoir à la fois la meilleure estimation de la durée de la trajectoire, mais aussi d'avoir la meilleure prédiction de trajectoire.

- D'après les résultats sur les temps de calcul, la troisième étude de cette thèse a pour but de compresser la dimension des ProMPs, afin que le robot puisse interpréter et prédire les mouvements corps complet de personnes, tout en agissant en temps réel.

Avec les forces et les couples

% d'observations de la trajectoire totale		30(\simeq 180)	50(\simeq 300)	70(\simeq 419)	90(\simeq 539)
Err. d'inférence du paramètre α	médiane	2.26e-08	1.35e-08	3.61e-09	8.95e-09
	moyenne	4.26e-02	7.81e-09	2.19e-08	2.09e-08
	variance	1.08e-02	1.09e-16	5.44e-16	2.24e-16
Err. de prédiction de la trajectoire (NRMSE) [m]	médiane	2.73e-01	5.51e-01	4.52e-01	5.38e-01
	moyenne	8.11e-01	5.86e-01	5.29e-01	3.42e-01
	variance	7.45e-01	1.36e-01	7.74e-02	2.93e-02
Temps de calcul [s]	moyenne	0.25	0.74	1.92	3.59
	variance	0.01	0.27	2.77	4.81

Sans forces et les couples

% d'observations de la trajectoire totale		30(\simeq 180)	50(\simeq 300)	70(\simeq 419)	90(\simeq 539)
Err. d'inférence du paramètre α	médiane	1.19e-02	1.76e-02	1.74e-02	1.62e-02
	moyenne	2.81e-02	1.92e-02	2.45e-02	1.43e-02
	variance	9.51e-04	1.00e-04	3.37e-04	1.82e-04
Err. de prédiction de la trajectoire (NRMSE) [m]	médiane	4.53e-02	4.73e-02	7.20e-02	6.20e-02
	moyenne	1.56e-01	7.36e-02	5.75e-02	4.29e-02
	variance	5.04e-02	1.80e-03	1.80e-03	6.0e-04
Temps de calcul [s]	moyenne	6.89e-02	8.49e-02	1.43e-01	2.58e-01
	variance	2.83e-03	1.19e-03	7.31e-03	2.45e-03

TABLEAU 4.1 – *Tableau bilan de la première étude – Moyennes et variances : du calcul des NRMSE des erreurs de prédiction de la trajectoire et d'inférence du paramètre α ; et de leur temps de calcul. Ces calculs sont effectués sous Matlab, avec un ordinateur à un coeur (c'est-à-dire, sans parallélisation).*

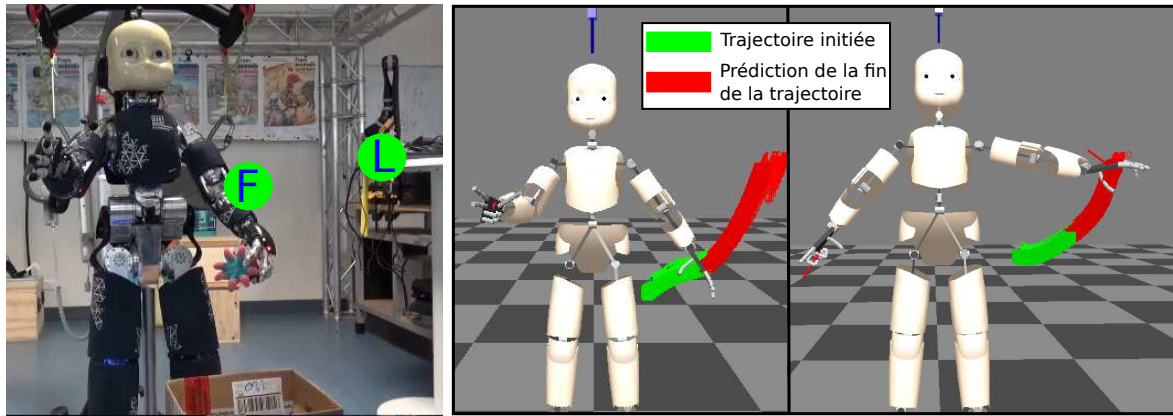


FIGURE 4.12 – 2eme expérience robot réel - Scénario. Triage d'objets dans deux corbeilles, à partir du guidage partiel de l'utilisateur.

À gauche : photo du robot réel avec les représentations des positions finales à atteindre "F" et "L".

À droite : visualisation de la prédiction sur *iCubGui* : au moment où l'utilisateur lâche le bras, puis à la fin du mouvement.

4.3.2 Tri collaboratif d'objets

Cette seconde expérience avec l'*iCub* réel a été introduite en début de Section 4.3.

Sachant que dans ce scénario, le robot doit pouvoir changer l'orientation de sa main afin d'atteindre les différentes corbeilles cibles, la ProMP prend cette fois-ci en compte, en plus de la position cartésienne de sa main $X_t = [x_t, y_t, z_t] \in R^3$, son orientation exprimée en quaternion : $A_t \in R^4$. Le modèle donc définit par : $\xi_t = [X_t A_t] = \Phi_{\alpha t} \omega + \epsilon_t$.

Pour cette expérience, l'utilisateur a fourni au robot une douzaine de trajectoires de démonstrations afin qu'il effectue son apprentissage de ProMPs.

Lors de la phase de prédiction, l'utilisateur attrape la main du robot et commence un mouvement en direction de l'une des corbeilles, puis il la lâche au moment où il désire que le robot finisse le mouvement de manière autonome.

Pour cela, les capteurs tactiles sont utilisés deux fois : tout d'abord pour identifier quand le mouvement commence puis quand il finit. Ainsi, le robot considère que le mouvement partiel, guidé par l'utilisateur commence lorsque l'utilisateur attrape son bras et qu'il finit lorsqu'il le relâche.

Puis, tout comme les expériences précédentes, le robot utilise cette trajectoire partielle et les ProMPs apprises lors de la phase d'apprentissage, afin de reconnaître la tâche initiée (c'est-à-dire, la ProMP suivie) et de prédire de la poursuite du mouvement désiré par l'utilisateur, qu'il effectue alors de manière autonome.

Cette expérience est présentée dans la vidéo [PredReel]. Une pause artificielle est introduite après que le robot ait prédit le mouvement, afin que l'utilisateur "valide" cette prédiction, à l'aide de sa représentation dans l'interface *iCubGui*, comme le montre l'image de droite de la Figure 4.12. Il s'agit d'une pause artificielle, que nous avons ajoutée afin d'éviter de causer des dommages au robot, mais elle n'est pas nécessaire. Si elle n'était pas utilisée, le robot continuerait le mouvement tout de suite après que l'utilisateur ait lâché son bras. La Figure 4.13 représente

une de ces prédictions, effectuée par le robot réel après que l'utilisateur ait relâché la main du robot. Dans cette figure, il n'y a plus de trajectoires représentant la "vérité terrain", puisque cette information est inconnue. Cependant, il s'agit de la bonne trajectoire attendue par l'utilisateur.

4.4 Améliorations envisagées et étude complémentaire

La méthode proposée permet donc au robot de prédire l'intention d'utilisateurs à partir de guidage physique, intention qui est modélisée par des trajectoires. Le robot peut notamment inférer la vitesse avec laquelle il doit effectuer les mouvements et, selon la description des trajectoires choisie, prédire les forces qu'il est susceptible de recevoir lors de ces mouvements. Finalement, le robot peut effectuer cette prédiction en temps réel mou.

Cependant, plusieurs améliorations peuvent être apportées à cette méthode, présentées ci-dessous.

- Amélioration de l'estimation du paramètre de modulation du temps -

Les expériences proposées ont montré que l'estimation du paramètre de modulation du temps α , qui permet de déterminer la durée de la trajectoire, améliore grandement la qualité de prédiction de celle-ci. Pour estimer ce paramètre, quatre méthodes sont étudiées (*c.f.*, Section 3.3.4) et les expériences avec le robot réel ont montré que la méthode "modélisation" permet d'obtenir les meilleurs résultats. Cependant, cette modélisation est spécifique au type de mouvement effectué⁴² lors de cette expérience et ne peut pas être généralisé pour tous les cas possibles. C'est pourquoi il serait bénéfique d'améliorer l'estimation de ce paramètre, en utilisant par exemple les techniques présentées dans la Section 1.7.3.

- Amélioration de la prédiction -

Cette première étude montre que l'estimation de la durée de la trajectoire est améliorée avec la prise en compte des mesures des forces/couples exercées sur le robot. Pour améliorer la prédiction des trajectoires, d'autres types de mesures pourraient aussi être utilisées. Il s'agit du sujet de la prochaine étude.

De plus, on pourrait utiliser les informations provenant du contexte des trajectoires. Par exemple, le robot pourrait détecter les différents objets présents dans son espace d'action, afin d'accroître la probabilité que ses mouvements se dirigent vers ce but. De même, il pourrait recentrer les trajectoires qu'il prédit vers l'objet ou le lieu le plus probablement visé.

En outre, il serait intéressant d'essayer d'identifier automatiquement les caractéristiques des profils de vitesses et d'accélération. En effet, ces caractéristiques jouent un rôle important dans la compréhension des intentions des mouvements humains. Par exemple, la vitesse des bras et la configuration des bras sont des indices permettant de détecter le but final d'une séquence de tâches exécutée par une personne. Extraire ces indices de manière automatique et utiliser l'information du paramètre de modulation du temps permettrait probablement d'améliorer l'inférence de la tâche à effectuer.

Finalement, notons que si la trajectoire guidée par l'utilisateur du robot ne correspond pas à l'une de celles apprises, alors le robot identifie la ProMP la plus proche de la trajectoire initiée. Il essaie alors d'adapter la ProMP reconnue afin d'achever le mouvement amorcé. De ce fait, le robot pourrait effectuer des mouvements non souhaités par l'utilisateur, mouvements qui peuvent être dangereux pour lui et pour son environnement. Une solution serait de définir une "enveloppe

42. Il s'agit d'une modélisation pour des trajectoires cartésiennes sans variation brusque.

probabiliste" autour de chaque ProMP. Ainsi, on considèrera qu'une trajectoire correspond à une certaine ProMP si elle est incluse dans son enveloppe, c'est-à-dire si sa probabilité d'appartenir à la distribution est supérieure à un certain seuil. Ce seuil pourra être fixé de différentes manières. Par exemple, il pourra être défini lors de la phase d'apprentissage de la ProMP, en faisant en sorte que chaque trajectoire de démonstration soit dans l'enveloppe probabiliste de sa propre ProMP, tout en cherchant à minimiser cette enveloppe. Un autre exemple serait de la fixer au 97.5e percentile, correspondant à un indice de confiance de 95% que la trajectoire appartienne bien à la ProMP.

- Prédiction en continu -

Dans le dernier paragraphe, nous avons proposé une méthode permettant au robot d'éviter de se tromper de trajectoire à suivre. Pour améliorer encore plus les capacités du robot, il serait intéressant de lui permettre de réagir lorsqu'il se trompe de mouvement. Cela pourrait par exemple arriver, lorsque deux tâches sont exécutées à partir de mouvements trop similaires (c'est-à-dire, lorsque l'objet est déplacé à partir d'une même position initiale et se dirige vers différents buts assez proches). Cela peut aussi arriver si l'utilisateur lâche le bras du robot avant que celui-ci ait mesuré suffisamment de données. Il serait alors pertinent de permettre au robot d'estimer si sa reconnaissance est suffisamment fiable avant d'effectuer l'action. Afin que le robot puisse reconnaître et prédire en continu des mouvements, cela nécessite simplement de modifier les calculs afin qu'ils soient effectués de manière itérative. Par contre, pour que le robot puisse changer en cours de route sa trajectoire lorsqu'il détecte une erreur d'identification, il faudra pouvoir assurer que son mouvement soit sécurisé pour lui-même et pour son environnement. De plus, que devrait faire le robot dans le cas où celui-ci ne serait plus suffisamment sûr de la trajectoire à effectuer ? Devrait-il s'arrêter ? La réponse à cette question dépendrait alors du type d'application.

- Modification du modèle du bruit -

Afin de permettre au robot d'améliorer sa capacité de prédiction, sachant que les mesures des forces et des moments sont plus bruitées que celles des positions cartésiennes, il serait intéressant de modifier le modèle du bruit des données $\epsilon_{\Xi} \sim \mathcal{N}(0, \Sigma_{\Xi})$, de telle sorte que les observations initiales des forces/couples soient considérées comme plus bruitées que celles des positions. Cela permettrait au robot de s'écarter légèrement de ses observations, en début de mouvement. Ainsi, dans de futures versions, il s'agira d'ajouter la possibilité de faire varier les modèles de bruit, selon les trajectoires observées. Par exemple, dans notre cas, nous aurions $\Sigma_{\Xi}^o = \begin{bmatrix} \Sigma_X^o & 0 \\ 0 & \Sigma_F^o \end{bmatrix}$. Dans cet exemple, le robot considèrerait alors qu'il y a plus de variabilité possible en ce qui concerne l'information des forces que celles de positions.

- Amélioration du temps de calcul -

Le logiciel implémenté dans le cadre de cette thèse a été effectué sous Matlab, sans optimisation particulière des temps de calcul. Ainsi, celui-ci peut facilement être amélioré, en commençant par l'implémenter en C++. Dans la dernière étude de cette thèse, nous proposerons en plus une méthode permettant au robot d'effectuer ses prédictions en temps réel mou, même lorsque les trajectoires sont décrites par de nombreuses données.

- Apprendre des tâches avec des objets -

Dans beaucoup de scénarios (*par ex.*, port d'objets ou travail d'assemblage coopératif), l'interaction physique entre l'humain et le robot peut avoir lieu par l'intermédiaire d'objets (c'est-à-dire, une interaction indirecte). Dans le cas où les manipulations sont spécifiques aux objets, la méthode que nous proposons s'applique toujours, mais pas uniquement au robot. Cela

nécessite alors d’adapter les ProMPs afin qu’elles modélisent les trajectoires des objets. Nous pourrions par exemple représenter l’objet par des points d’intérêts et apprendre une modélisation des mouvements de ces points d’intérêts. Puisque les ProMPs permettent la co activation (produit) et la mixture (fonction d’activation de l’ensemble des ProMPs) de primitives, ces propriétés pourraient être exploitées afin d’apprendre une distribution conjointe des trajectoires du robot et des trajectoires des différents objets.

De plus, l’ajout de l’information des forces/couples dans la modélisation des mouvements du robot pourrait être utilisé afin que le robot puisse effectuer différentes trajectoires selon le type d’objet (chaque objet serait alors identifiable par son poids).

- Comparaison avec CMPs -

Par rapport à la méthode de l’étude [63] (*c.f.*, Section 1.7.1) qui permet au robot de retrouver les paramètres des primitives de mouvements en fonction des contraintes posées, la technique ProMP permet quant à elle au robot d’apprendre directement à varier ses trajectoires, selon différentes contraintes (*c.f.*, Section 3.3). Pour cela, lors des expériences avec le robot réel, les ProMPs enregistrent, en plus de la position cartésienne de la main du robot, les différentes forces que celui-ci subit. Cela a notamment permis au robot d’améliorer l’inférence de la durée des trajectoires (et en conséquence, leur vitesse) en fonctions des contraintes qu’il subit au début du mouvement, au travers des forces ressenties. Il serait intéressant d’étendre la comparaison entre ces deux méthodes afin d’évaluer leurs avantages et inconvénients respectifs.

4.5 Conclusions

Dans cette première étude, nous proposons une méthode permettant de prédire l’intention de l’utilisateur en interaction physique avec le robot *iCub*, dans des tâches de coopération. La prédiction de l’intention est représentée comme la prédiction du but et de la trajectoire “future” désirée par le partenaire. Cette intention est identifiée à partir d’observations initiales de la trajectoire et est modélisée à l’aide de la méthode Primitive de Mouvement Probabiliste. Nous avons choisi cette méthode, car elle permet de prendre en compte la variabilité des trajectoires permettant de répondre à une même tâche, en représentant ces trajectoires sous la forme d’une distribution, calculée à l’aide de plusieurs démonstrations de cette même tâche.

À partir des informations contenues dans de telles ProMPs, le robot est alors capable de calculer la finition des trajectoires initiées, en conditionnant la ProMP afin qu’elle corresponde aux données initiales de la trajectoire observée. Cette méthode a d’autres avantages, tels que la possibilité d’estimer la durée du mouvement attendu, de reconnaître la tâche courante parmi celles apprises et de pouvoir prédire ces trajectoires, à l’aide d’informations multimodales.

Le Chapitre 3 décrit la structure théorique et les Sections 2.5.1–4.3–A présentent le logiciel open source qui fournit l’implémentation de la méthode proposée. Ce logiciel est accessible sur *GitHub* où des tutoriels et vidéos sont fournis.

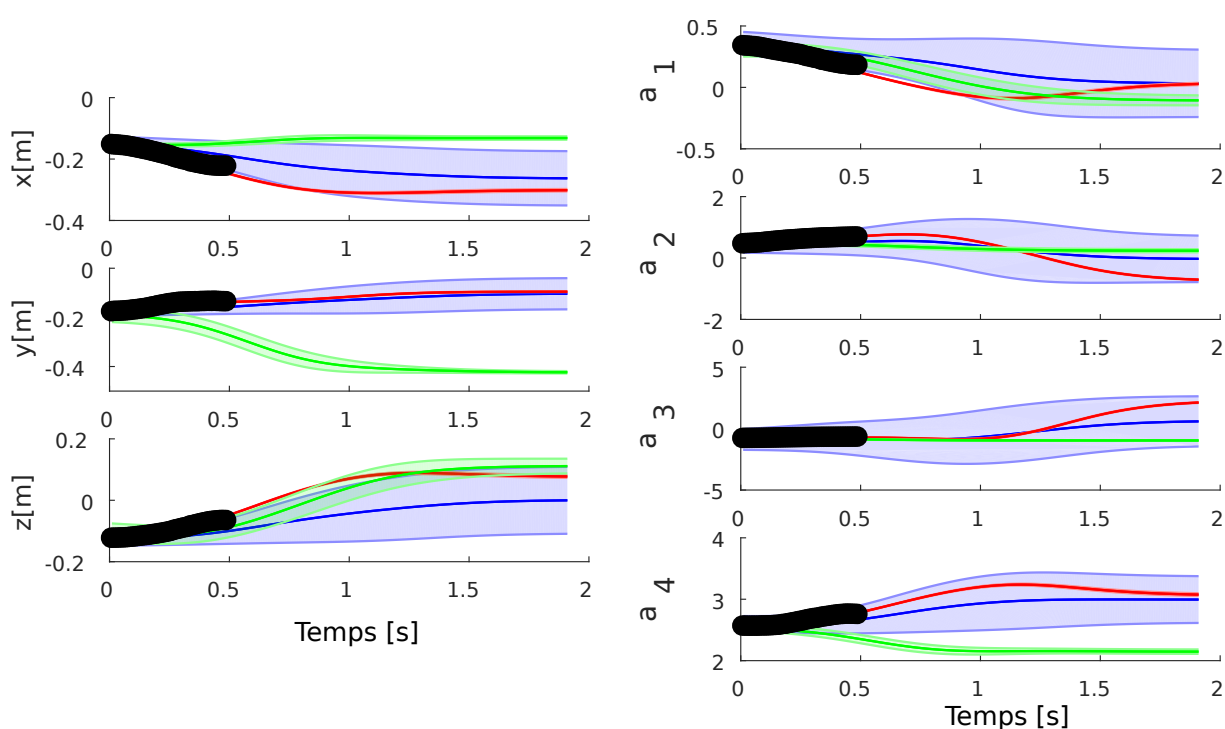
Les expériences présentées ont été effectuées à la fois avec le robot réel et le robot simulé. Dans celles-ci, le robot apprend un ensemble de primitives de mouvement qui correspondent à des tâches différentes, à partir de plusieurs trajectoires de démonstration guidées par un utilisateur. Les ProMPs ainsi apprises correspondent à une modélisation a priori des trajectoires à effectuer et sont utilisées par la suite pour permettre de prédire l’intention d’un utilisateur.

Quand celui-ci commence une des tâches collaboratives, le robot utilise les observations correspondant à ce mouvement initial afin d’inférer quelle tâche l’utilisateur est en train d’effectuer, puis de prédire la finition de la trajectoire correspondante, souhaitée par l’utilisateur. Dès que

l'utilisateur lâche le bras du robot, la trajectoire prédite est utilisée par le robot, afin qu'il effectue par lui-même la tâche attendue.

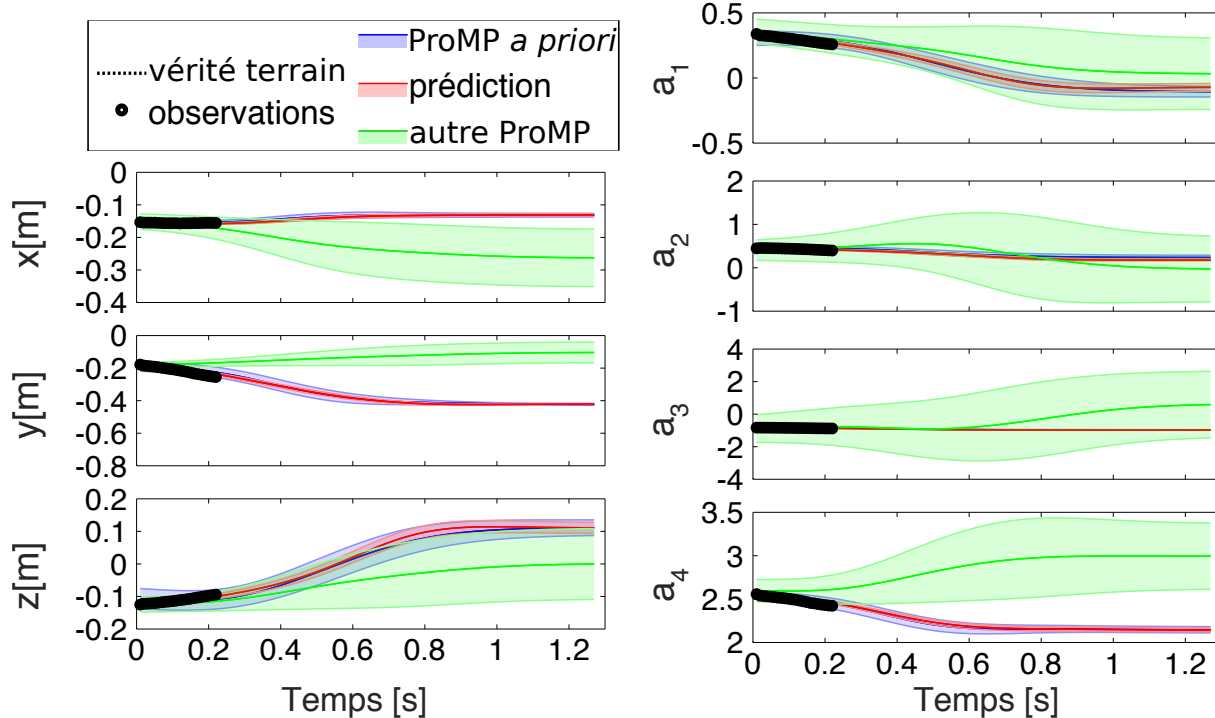
Dans la Section 4.4, nous proposons quelques améliorations du logiciel, afin de répondre à des enjeux non-pris en compte dans la version actuelle, ce qui permettrait ainsi à ce logiciel d'être utilisé dans un répertoire encore plus large de scénarios de collaboration entre l'humain et le robot. Dans la troisième étude de cette thèse, la priorité sera alors d'accélérer le temps de calcul nécessaire à la prédiction, ce qui permettra alors au robot de prédire de manière continue les trajectoires à exécuter de manière encore plus rapide.

Avant cela, nous souhaitons étudier si cette méthode peut permettre au robot de prédire la trajectoire qu'il doit effectuer avec différents types de senseurs, afin d'offrir plus de liberté à l'utilisateur sur la manière de guider le robot.



(a) Action dirigée vers l'avant (F).

Le robot reconnaît le mouvement après 0.5 sec.



(b) Action dirigée vers la gauche (L).

Le robot reconnaît le mouvement après 0.25 sec.

FIGURE 4.13 – 2eme expérience robot réel – Prédiction. Prédiction des trajectoires dont les positions but sont présentées par la Figure 4.12). La distribution a priori de la ProMP reconnue est représentée en bleu (l'autre en vert) et la distribution a posteriori de cette ProMP est représentée en rouge.

Multimodalité dans l'interaction humain-robot (Étude 2)



FIGURE 5.1 – *Communication visuelle naturelle entre humains* – À l'aide d'un mouvement de tête, la femme de la cafétéria demande à sa cliente de récupérer la salade. Celle-ci peut adapter son mouvement de tête afin de donner des indices supplémentaires sur l'action et le mouvement à effectuer : par exemple, avec un mouvement rapide, la personne qui doit agir comprend qu'elle a intérêt à se dépêcher...

5.1 Résumé

Cette étude propose d'utiliser la méthode des ProMPs afin que le robot puisse prédire de manière multimodale (c'est-à-dire, provenant de différents capteurs) l'intention de son partenaire,

intention qui est définie comme un mouvement et le but associé. Cette étude vise toujours l'interaction dyadique entre un robot et un humain, dans des scénarios de collaboration. Cette fois-ci, le robot modélise l'action collaborative à l'aide de primitives de mouvements probabilistes multimodales, en décrivant l'action effectuée par son bras gauche à la fois avec des informations visuelles (direction du regard de son partenaire) et des informations cinématiques (mouvement de son bras, guidé par son partenaire). Cette expérience met en avant que si l'utilisateur guide le robot en effectuant un mouvement de tête, le robot reconnaît la primitive d'action associée à ce mouvement, même dans des cas d'actions ambigus (par exemple, lorsque la trajectoire est initiée de la même manière). De plus, cette information a priori (c'est-à-dire, l'identification par le regard du type de mouvement et de l'action associée) permet aussi d'améliorer la prédiction du futur de la trajectoire initiée par l'utilisateur, lorsque celui-ci guide le robot manuellement par contact physique. Cette expérience a été à nouveau effectuée avec le robot *iCub* réel.

5.2 Introduction

Dans la Section 1, on a montré que les êtres humains ont des compétences très développées en ce qui concerne la prédiction et l'adaptation de leurs actions lorsqu'ils collaborent. Pour cela, ils utilisent des indices multimodaux (auditifs, visuels, etc.) leur permettant de prédire l'intention de leur partenaire de manière robuste [225], comme présenté dans la Section 1.6.1.

Pour collaborer efficacement avec les êtres humains, sachant que ceux-ci exhibent des compétences d'anticipation, les robots doivent eux aussi être capable de prédire l'intention de leurs utilisateurs. Basé sur ce qui a été présenté dans la Section 1.2, nous considérons que la prédiction de l'intention de l'utilisateur à partir de ses mouvements n'est possible que lorsque ces mouvements sont à la fois *lisibles* et *prédictibles*. En effet, ces conditions sont nécessaires afin que le robot puisse rapidement inférer le but du mouvement et prédire la poursuite du mouvement.

Ici, nous défendons l'idée qu'en utilisant des informations multimodales [66, 230], la qualité de prédiction du robot peut être améliorée.

Dans l'étude précédente (*c.f.*, Section 4, article [D.P]), nous posons le problème de prédire les futurs mouvements du bras du robot, initiés physiquement par l'utilisateur en interaction avec ce bras. Pour cela, la méthode ProMPs [170] était utilisée afin d'apprendre des primitives de mouvements à partir d'un set de démonstration ; puis de calculer la trajectoire attendue par l'utilisateur, à partir de l'observation partielle d'un mouvement initié par l'utilisateur.

Dans cette étude, nous ajoutons la modalité visuelle au robot, afin qu'il prédise l'intention de son partenaire à partir de mesures visuelles et cinématiques. Cette hypothèse s'appuie sur la façon de communiquer des êtres humains, où la compréhension de l'intention des partenaires passe souvent par le regard (*c.f.*, Section 1.5). La Vidéo [HHI] et la Figure 5.1 illustrent ce type de communication non verbale passant par le regard.

Cette fois-ci, le robot apprend donc la combinaison de mesures sur la cinématique de son bras, lorsqu'un utilisateur le déplace, ainsi que le mouvement du regard de ce dernier. Ces mesures permettent ainsi au robot d'apprendre des ProMPs multimodales, à partir de trajectoires de démonstration.

À partir de la prédiction par guidage physique, le robot est capable de reproduire le mouvement, ainsi que d'achever un mouvement initié par le partenaire, et ce, même avec peu d'observations. À partir du guidage visuel, le robot est capable de prédire et d'effectuer des tâches, lorsqu'elles ne nécessitent pas que l'utilisateur adapte la trajectoire en guidant le début du mouvement. De plus,

(a) *Guidage visuel.*(b) *Guidage physique.*

FIGURE 5.2 – *2eme étude - Scénario.* Le robot humanoïde iCub a) reconnaît la primitive de mouvement espérée, à l'aide de la direction du regard de son partenaire ; b) prédit le mouvement à effectuer, à l'aide du mouvement initial de son bras guidé par son partenaire.

le guidage visuel permet au robot de désambiguïser facilement des primitives dont les mouvements commencent de manière identique.

La Section 5.3 formule la problématique de cette étude. La Section 5.4 exprime les spécificités de cette étude, afin d'adapter le cadre théorique présenté dans la Section 3.3 à l'apprentissage d'informations multimodales provenant de capteurs visuels et cinématiques.

La Section 5.5 présente le déroulement de cette étude et ses résultats. Ces résultats montrent que la prédiction de la trajectoire future est améliorée par l'apprentissage multimodal.

Finalement, la Section 5.6 discute de l'approche proposée, de ses limitations et des perspectives envisagées.

5.3 Formulation du Problème

Le scénario d'utilisation consiste en des interactions dyadiques entre l'humain et le robot, interactions qui peuvent être à la fois visuelles et/ou haptiques. Afin de présenter et d'évaluer cette méthode, des expériences sont réalisées, qui consistent à des déplacements d'objets (*c.f.*, Figure 5.2), en suivant des trajectoires différentes.

Pour cela, l'utilisateur du robot choisit de le guider soit physiquement en déplaçant son bras (interaction physique), soit en présentant la trajectoire à effectuer à l'aide de mouvement de tête (c'est-à-dire, interaction visuelle). Le robot doit alors utiliser cette information afin d'achever le mouvement par lui-même.

Lorsque l'utilisateur guide visuellement le robot, celui-ci suit l'orientation de la tête de son partenaire afin d'en déduire l'intention sous-jacente. Pour cela, le robot identifie, parmi celles apprises, la primitive de mouvement correspondant le plus à la trajectoire du regard du partenaire. À l'aide de cette identification, le robot prédit alors la tâche qu'il doit exécuter (en effet, à chaque primitive correspond une tâche à effectuer). Le robot prédit alors la tâche courante et le futur attendu du mouvement initié. À l'aide de cette prédiction, il accomplit alors la tâche attendue en plaçant l'objet à la position désirée, en suivant la trajectoire voulue par son partenaire.

Lors de guidage physique, l'utilisateur commence par déplacer le bras du robot physiquement afin d'amorcer l'action ; après quelques observations, le robot prédit le futur de la trajectoire à effectuer.

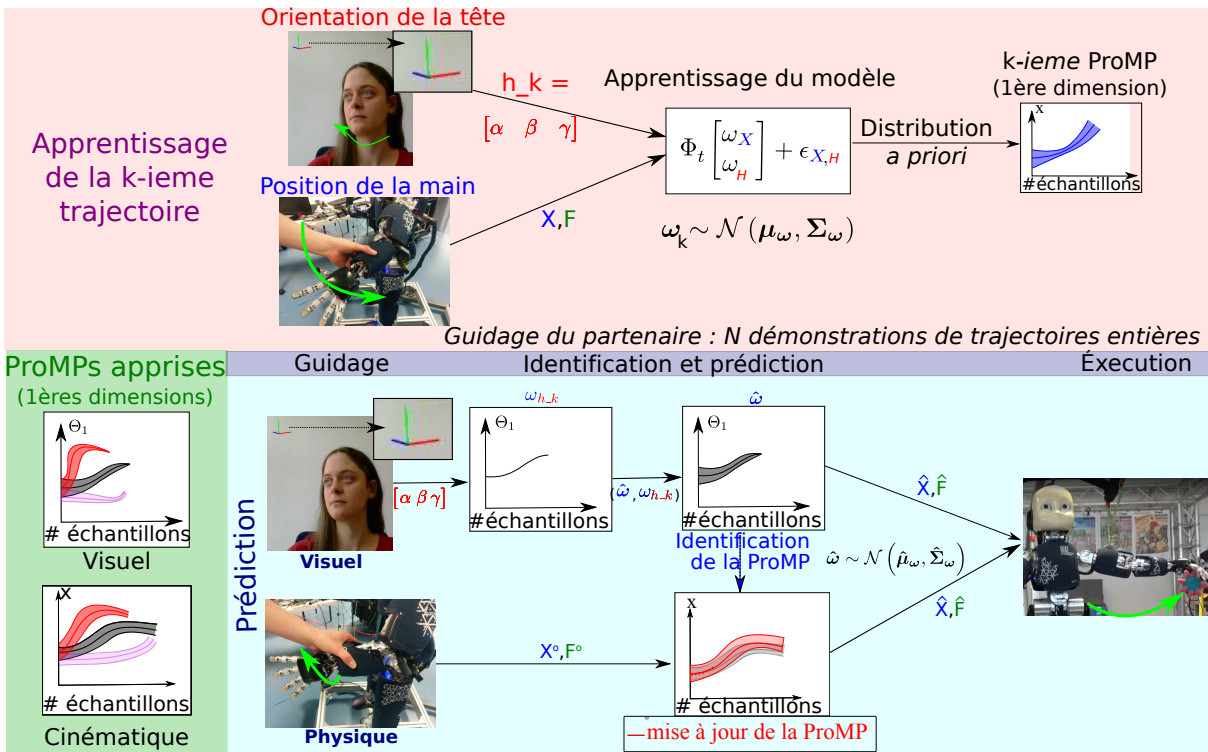


FIGURE 5.3 – *Utilisation conceptuelle des ProMPs afin de prédire la trajectoire que l'utilisateur souhaite que le robot exécute. Lors de la phase d'apprentissage (en haut), les ProMPs sont apprises à partir de trajectoires de démonstration guidées par l'utilisateur. Lors de la phase de prédiction (en bas), le robot reconnaît la ProMP correspondant à la trajectoire initiée par le partenaire, à l'aide de mesures visuelles et/ou haptiques.*

Si l'utilisateur utilise les deux modalités, la primitive de mouvement peut être reconnue à l'aide du guidage visuel (c'est-à-dire, à partir de la primitive de mouvement *a priori* reconnue qui permet la reconnaissance de la tâche à effectuer) ; puis le guidage physique permet d'améliorer la trajectoire future prédite (c'est-à-dire, qu'il met à jour la distribution reconnue à l'aide de l'information visuelle, pour que celle-ci corresponde davantage à la trajectoire initiée).

Afin de réaliser ce scénario, différentes hypothèses ont été posées.

Tout d'abord, sachant qu'il est difficile de suivre le regard d'une personne à l'aide de la direction de ses yeux, dû à ses saccades oculaires, le robot détermine l'intention de son partenaire à l'aide de l'orientation de la tête de l'utilisateur.

De plus, la position de l'utilisateur est supposée fixe par rapport à la position du robot, que ce soit lors de la phase d'apprentissage ou de reconnaissance. En effet, le robot a besoin d'estimer l'orientation de la tête de son partenaire afin de reconnaître la primitive. Or, cette orientation est relative à la position du robot et de l'utilisateur.

Une représentation conceptuelle de ce problème est présentée par la Figure 5.3.

Afin d'apprendre la primitive de mouvement (graphique du haut), deux partenaires présentent un ensemble de mouvements de démonstrations : l'un fait faire le mouvement à apprendre au robot en guidant son bras gauche ; tandis que l'autre effectue un mouvement correspondant avec sa tête.

À partir de ces démonstrations, le robot enregistre la position cartésienne de son bras gauche

ainsi que la direction du regard de son partenaire (c'est-à-dire, l'orientation de sa tête). Ces trajectoires correspondent alors à la base d'apprentissage des primitives de mouvement.

Le bas de la figure représente la phase de prédiction. Durant cette phase, le guidage peut être effectué par un seul des partenaires, qui a alors le choix de guider le robot avec un mouvement de tête et/ou en guidant le bras du robot. Lorsque la prédiction est effectuée, le robot finit alors le mouvement par lui-même, puis il lâche l'objet à la position désirée.

Afin de mettre en avant l'amélioration de cette étude par rapport à celle d'avant, les trajectoires apprises ont une position initiale et finale identiques, ce qui complexifie la tâche de prédiction.

5.4 Cadre théorique adapté à l'application : apprentissage de primitives multimodales

Les ProMPs apprises correspondent à différentes primitives d'actions. Le robot les utilise comme une connaissance a priori, afin de prédire qu'elle est l'action initiée par un utilisateur et comment achever le mouvement correspondant de manière autonome.

Ici, les observations initiales de la trajectoire correspondent à un sous-ensemble des variables apprises :

$$\Xi^o = [\Xi_1 \dots \Xi_{n_o}]^\top = \{X^o \| A^o \| [X^o A^o]\} \quad (5.1)$$

Où X^o est la mesure haptique et A^o , la mesure visuelle.

La première étape du processus de reconnaissance consiste à reconnaître la ProMP qui correspond au mouvement courant $\hat{k} \in [1 : K]$ (avec $K = 2$ dans nos expériences), ainsi que le paramètre de modulation du temps $\hat{\alpha}$ et ce, à partir des observations partielles Ξ^o .

Pour cela, le robot calcule le couple, "paramètre de modulation du temps, ProMP" $(\hat{\alpha}_{\hat{k}}, \hat{k})$ qui correspond le plus probablement à la trajectoire initiée. Pour cela, nous avons présenté dans la Section 3.3.4 et testé dans l'étude précédente quatre méthodes permettant ces calculs. Dans cette étude, nous nous concentrons sur les deux méthodes les plus performantes, soit :

- La méthode du « **maximum de vraisemblance** » (c.f., Équation 3.10).
- la méthode de « **modélisation** » (c.f., Équation 3.12), où ici, la "variation des observations" concerne la modalité utilisée, soit :

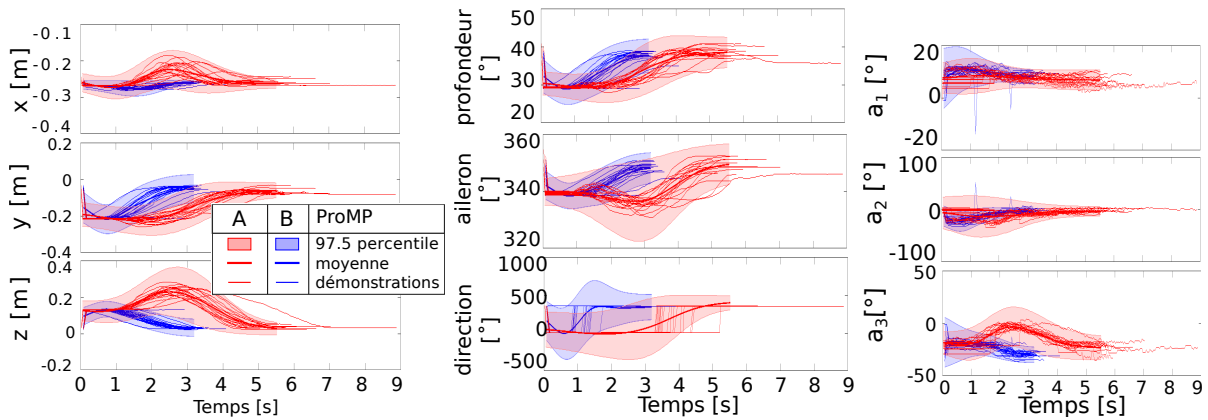
" $\delta_{n_o} = A(n_o) - A(1)$ " lorsque le robot utilise la modalité visuelle ; " $\delta_{n_o} = X(n_o) - X(1)$ " lorsqu'il utilise la modalité haptique ou encore " $\delta_{n_o} = \Xi(n_o) - \Xi(1)$ " lorsqu'il utilise les deux modalités.

Une fois ce couple $(\hat{\alpha}_{\hat{k}}, \hat{k})$ identifié, la distribution a posteriori de la ProMP \hat{k} est calculée à l'aide de l'Équation 3.14

Le robot peut alors finir le mouvement en récupérant la trajectoire future la plus probable de cette distribution, c'est-à-dire, sa moyenne : $\forall t \in [1 : \hat{t}_f], \hat{\xi}(t) = \Phi_t \hat{\mu}_{\omega_{\hat{k}}}$, avec $\hat{t}_f = \frac{\bar{s}}{\hat{\alpha}_{\hat{k}}}$ la durée estimée de la trajectoire. Le robot suit alors le mouvement cartésien correspondant à : $\hat{X} = [\hat{X}_{n_o+1} \dots \hat{X}_{\hat{t}_f}]^\top$.

5.5 Application au robot réel

Dans cette expérience, le robot se tient en face d'une table. Le scénario met en jeu différents objets et différentes positions à atteindre. Un exemple d'un tel scénario est illustré dans la



(a) Position de la main du robot. (b) Orientation de la main du robot. (c) Orientation de tête.

FIGURE 5.4 – **ProMPs multimodales** – Trajectoires de démonstration et primitives associées. En rouge (ProMP A) la trajectoire “en cloche” et en bleu (ProMP B) la trajectoire “directe”.

Figure 5.2. Afin de récupérer une approximation de la direction du regard d’un utilisateur, le robot mesure les angles représentant l’orientation de sa tête. Ces angles sont extraits à partir des images récupérées par les yeux du robot et d’un traitement de ces images effectué par le logiciel *Intraface* [234] (*c.f.*, Section 2.3 pour plus de détails sur ce logiciel.) Afin de récupérer l’information cartésienne du robot, le robot utilise un module préexistant nommé “*iKinCartesianSolver*”, qui permet de récupérer les positions et orientations cartésiennes du robot.

5.5.1 Apprentissage des ProMPs multimodales

Dans cette expérience, le robot apprend deux primitives de mouvement multimodales. Ces primitives lui permettent de lâcher un objet dans une même zone, où la position précise varie selon la volonté de l’utilisateur. Alors que les positions initiales et finales de ces deux primitives de mouvement sont identiques, ces primitives de mouvement suivent deux trajectoires différentes qui sont identiques ou corrélées aux mouvements de l’utilisateur suivant si celui-ci guide le robot physiquement ou visuellement. Ce scénario soulève les limites des capacités de prédiction des trajectoires à partir du guidage physique partiel de l’utilisateur : puisque le début des trajectoires est identique, nous supposons que le robot aura du mal à comprendre quelle trajectoire l’utilisateur souhaite qu’il exécute.

Ces primitives sont constituées des mesures des positions et d’orientations cartésiennes de la main gauche du robot (récupérées lorsque l’un des utilisateurs guide le bras du robot) et les mesures de l’orientation de la tête du second utilisateur (récupérées lorsque celui-ci suit visuellement le mouvement du premier utilisateur). Ces mesures correspondent donc à $\xi(t) = [X(t), A(t)]^\top$, avec $X(t) \in \mathbb{R}^6$ la position et l’orientation cartésienne du bras robotique et $A(t)$ les angles de lacet, de tangage et de roulis représentant l’orientation de la tête du partenaire.

Le robot apprend ainsi ces deux ProMPs à partir de 20 trajectoires de démonstration par primitive. La Figure 5.4 représente ces trajectoires de démonstration et les ProMPs apprises correspondantes.

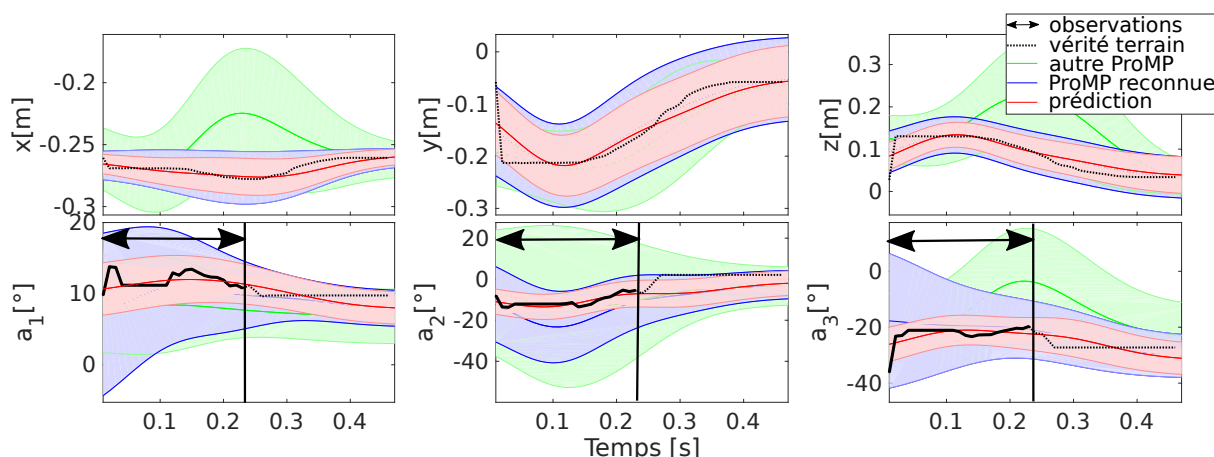


FIGURE 5.5 – **Information visuelle** – Exemple de prédiction d’une trajectoire à partir de 50% d’observations de l’orientation de la tête du partenaire. Les points représentent la trajectoire que doit effectuer le robot (vérité terrain). Les courbes noires représentent les observations mesurées par le robot. La distribution bleue représente la ProMP reconnue et la distribution verte l’autre ProMP. La distribution rouge représente la distribution a posteriori de la ProMP bleue, calculée à partir de données mesurées.

5.5.2 Activation des primitives à l’aide du regard

L’information du regard est utilisée afin d’identifier l’action courante. Cette procédure comporte deux avantages. Tout d’abord, elle ne requiert aucune interaction physique avec le robot, ce qui facilite l’interaction entre le robot et les personnes, qui ont parfois une attitude négative envers le robot. De plus, cela permet d’améliorer la prédiction de la trajectoire souhaitée par l’utilisateur, surtout lorsque les primitives sont ambiguës, comme lorsqu’elles sont initiées de la même manière. C’est le cas du scénario choisi dans cette expérience, et un schéma intuitif est présenté dans la Figure 5.6 pour illustrer cette ambiguïté.

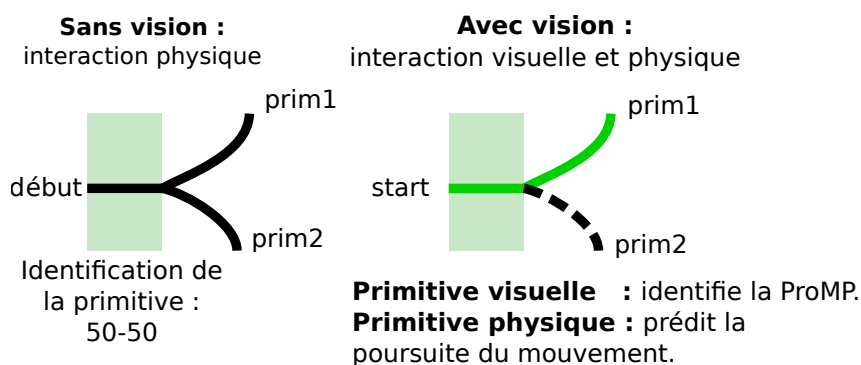
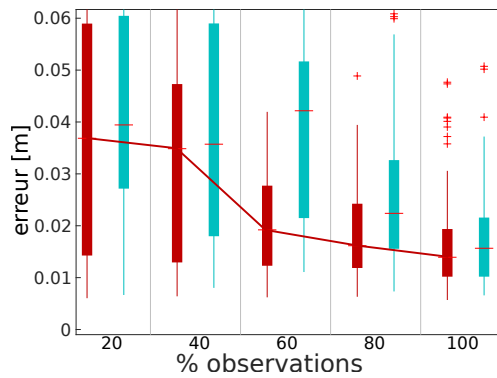
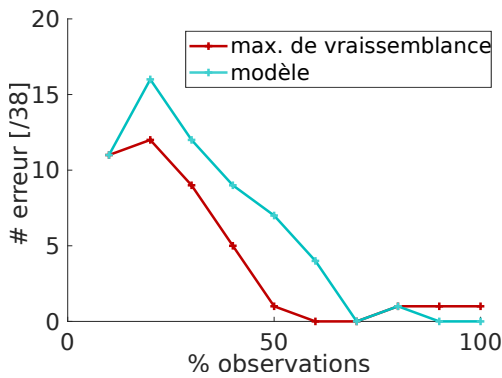


FIGURE 5.6 – Le regard de l’utilisateur permet de désambigüiser deux primitives qui se superposent.

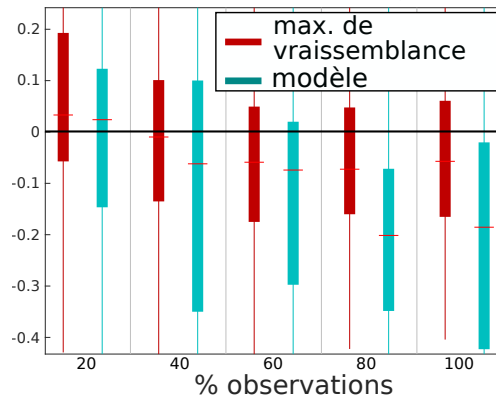
La Figure 5.5 représente un exemple de la prédiction de la poursuite d’une trajectoire après que le robot ait observé 50% du mouvement facial de son partenaire. Pour clarifier cette figure, la prédiction de l’orientation de la main du robot n’a pas été représentée. La trajectoire prédite correspond à la moyenne de la distribution a posteriori représentée en rouge.

Remarquons que cette distribution a posteriori est incluse dans la distribution a priori et



(a) Moyenne du nombre d'erreurs de reconnaissance du type de trajectoire, sur 38 tests.

(b) Erreur moyenne de l'inférence de la trajectoire cartésienne, calculée par la distance $|X_{des} - \hat{X}|$.



(c) RNMSE de la différence entre les distributions a priori et a posteriori.

FIGURE 5.7 – Analyses du guidage visuel.

qu'elle passe par les observations du robot avec un peu de *flexibilité*. Cette flexibilité correspond au bruit blanc gaussien fixé a priori dans le modèle afin de représenter le bruit des observations du robot. Ce résultat montre que, bien que les observations faciales du partenaire soient imprécises⁴³, la capacité de prédiction du robot est suffisante pour permettre à celui-ci d'effectuer le mouvement correctement.

La Figure 5.7a représente l'erreur de reconnaissance des ProMPs en fonction du pourcentage d'observations de la trajectoire testée. Cette figure montre bien que plus la partie de la trajectoire observée par le robot est grande, plus l'erreur de prédiction est petite, et ce, pour les deux méthodes de calcul du paramètre de modulation du temps. On remarque cependant que lors des derniers pourcentages testés (c'est-à-dire, 80%, 90%, 100%), des erreurs réapparaissent. Cela est probablement dû au fait que le logiciel utilisé pour la reconnaissance faciale ne fonctionne pas bien lorsque la tête de l'utilisateur est trop tournée : ainsi, les orientations perçues lorsque l'utilisateur tourne trop la tête étant incorrectes, des petites erreurs de reconnaissance peuvent avoir lieu.

De plus, cette figure montre que la méthode de modélisation est moins précise que la méthode du maximum de vraisemblance lorsque le robot observe moins de 70% de la trajectoire totale, tandis qu'avec plus d'observations, la méthode de *modélisation* est plus précise. Puisque les

⁴³. En effet, la méthode utilisée ne permet pas d'estimer avec précision l'orientation de la tête du partenaire (c.f., Section 2.3).

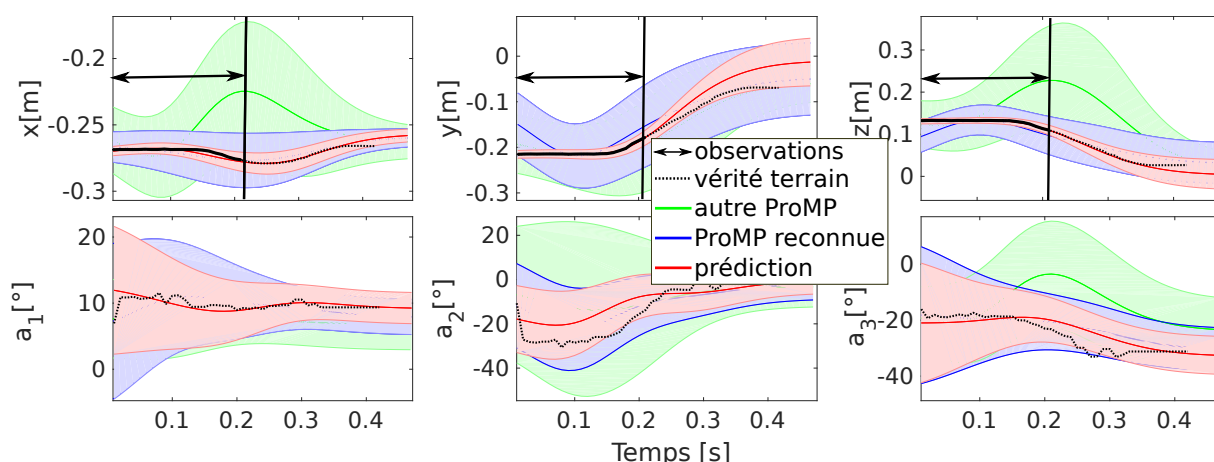


FIGURE 5.8 – Exemple de prédiction d’une trajectoire à partir de guidage physique.

mouvements de tête sont rapides, le robot peut utiliser le mouvement de tête dans sa totalité tout en continuant à réagir rapidement. Ainsi, la méthode de *modélisation* peut être utilisée afin de permettre au robot de reconnaître la trajectoire à suivre à partir du guidage visuel d’un utilisateur. Finalement, avec 70% d’observations de la trajectoire (ce qui correspond à environ 3 secondes) il n’y a plus d’erreur de reconnaissance de la trajectoire à effectuer, le robot peut inférer la primitive de mouvement qu’il doit suivre à partir de ce nombre d’observations, afin d’effectuer une trajectoire globalement correcte.

La Figure 5.7b représente l’erreur moyenne des trajectoires cartésiennes prédites. Elle montre que l’écart entre la trajectoire désirée et la prédiction de celle-ci va de 4cm (lorsque le robot observe 10% de la trajectoire) à 2cm (lorsqu’il observe 80% de la trajectoire). Ainsi, plus le robot observe la trajectoire faciale de l’utilisateur, plus il est capable d’achever de manière précise la trajectoire désirée par l’utilisateur.

Cependant, on peut se demander si la distribution a posteriori est plus précise que la distribution a priori. C’est le cas si les observations de la trajectoire partielle permettent au robot d’ajuster la trajectoire qu’il a apprise. Cela voudrait alors dire que la trajectoire faciale de l’utilisateur est corrélée à la trajectoire des mains du robot et que les mesures de ce mouvement facial sont suffisamment précises. La Figure 5.7c représente la différence entre l’*erreur moyenne quadratique normalisée (NRMSE)* de la distribution a posteriori et de la distribution a priori. À partir du moment où le robot observe 40% de la trajectoire totale, cette différence est inférieure à zéro, ce qui signifie que la mise à jour de la distribution (c’est-à-dire, la distribution a posteriori) permet au robot d’améliorer la prédiction de la fin de la trajectoire. Ainsi, guider visuellement avec des mouvements de tête est suffisant pour que le robot reconnaisse la primitive de mouvement à suivre, mais aussi à ce qu’il adapte son mouvement à la trajectoire visuelle observée de manière précise. Nous supposons maintenant qu’avec un guidage physique du robot, celui-ci pourrait prédire de manière encore plus précise la poursuite de la trajectoire. Afin de vérifier cette hypothèse, la Section suivante présente les résultats de cette étude lorsque l’utilisateur utilise le guidage physique.

5.5.3 Prédiction de la trajectoire désirée à l’aide de mesures cinématiques

La même expérience de prédiction est présentée ici, mais cette fois-ci à l’aide du guidage physique. La Figure 5.8 présente un exemple de ce type de prédiction. Par rapport à l’expérience

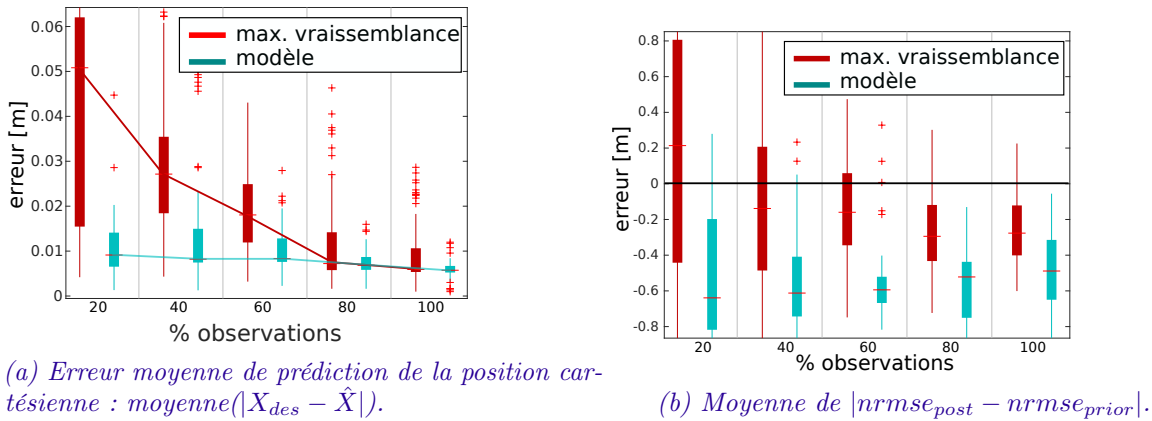


FIGURE 5.9 – Analyses du guidage physique.

précédente, la trajectoire prédite (c'est-à-dire, la moyenne de la distribution a posteriori représentée en rouge) est plus proche de la trajectoire représentant la vérité terrain.

La Figure 5.9a permet de vérifier cette idée. Elle représente la distance moyenne entre la trajectoire prédite et la trajectoire désirée (c'est-à-dire, $moyenne(|\hat{X} - X_{des}|)$). De plus, ces résultats montrent que les prédictions de trajectoires effectuées à l'aide du guidage physique sont plus précises que celles effectuées à l'aide du guidage visuel. Ce résultat est valable, quelle que soit la méthode permettant d'estimer le paramètre de modélisation du temps. En effet, la distance moyenne entre la trajectoire prédite et la trajectoire réelle est inférieure à 1cm pour la *modélisation* et va de 3cm (lorsque le robot observe 40% de la trajectoire) à 1cm (lorsque le robot observe 80% de la trajectoire) pour le maximum de vraisemblance. Cela montre aussi que la méthode de modélisation du paramètre de modulation du temps est plus précise que la méthode du maximum de vraisemblance.

De plus, la Figure 5.9b montre que la distribution a posteriori de la ProMP améliore la précision de la trajectoire prédite, en particulier pour la méthode de *modélisation*, ce qui correspond au fait que la distance d'erreur de cette méthode est plus petite dans la figure précédente.

Nous supposons maintenant qu'en utilisant les deux modalités à la fois, les performances de prédiction augmentent. La prochaine section propose donc une expérience multimodale, basée sur le même ensemble de trajectoires.

5.5.4 Prédiction de la trajectoire désirée à l'aide de mesures multimodales

La Figure 5.10 représente la prédiction de la trajectoire cartésienne effectuée par le robot lorsque celui-ci observe 50% des mouvements provenant à la fois du guidage physique et visuel. Ces observations sont représentées par les gros points noirs sur ces graphiques.

Dans cet exemple, la trajectoire prédite (c'est-à-dire, la moyenne de la distribution a posteriori représentée en rouge) est proche de la trajectoire désirée par le partenaire (représentée par les petits points noirs). En comparant visuellement ce graphe exemple avec celui de la Figure 5.8, il n'est pas clair que le guidage multimodal permet d'améliorer la robustesse de la prédiction.

C'est pourquoi afin de pouvoir comparer les prédictions provenant des mesures multimodales, visuelles ou physiques, les Figures 5.11 et 5.12 ont été créées. Elles représentent les statistiques pour tous ces types de prédiction, pour chacune des méthodes d'inférence du temps.

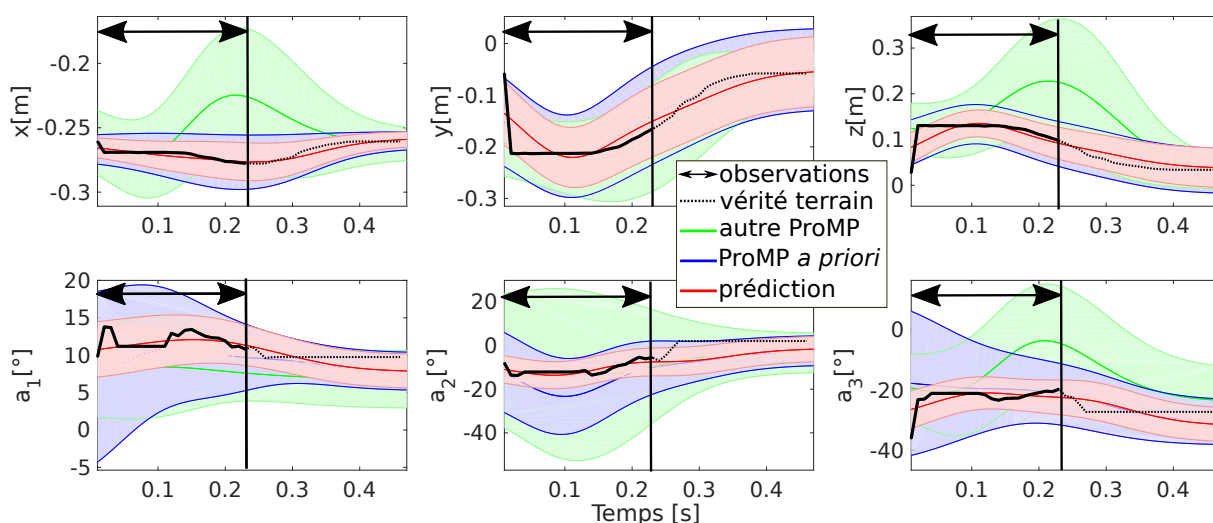
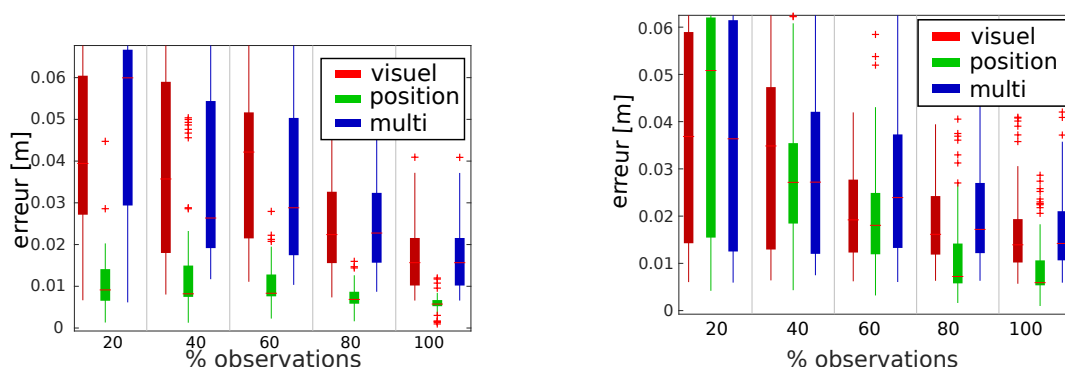


FIGURE 5.10 – Exemple de prédiction d’une trajectoire, à partir de 50% d’observations du mouvement facial d’un utilisateur et de la trajectoire guidée physiquement.



(a) Avec la méthode de modélisation.

(b) Avec la méthode du maximum de vraisemblance.

FIGURE 5.11 – Erreur de prédiction de la position cartésienne. Moyenne sur l’ensemble des tests de $|X_{des} - \hat{X}|$ selon la modalité utilisée.

La première Figure 5.11 représente l’erreur de distance entre la position cartésienne de la trajectoire réelle et de celle estimée. Elle montre que, que ce soit avec la méthode de modélisation (5.11a) ou du maximum de vraisemblance (5.11b), la prédiction effectuée par guidage physique est plus précise que par guidage visuel ou multimodal. De plus, comme vu précédemment, les prédictions sont plus performantes avec la méthode de modélisation, avec une erreur en moyenne inférieure à 1cm et ce, dès l’observation de 20% de la trajectoire. Notons que ces résultats sur le guidage physique correspondent bien à l’étude précédente.

La seconde Figure 5.12 représente le nombre d’erreurs de reconnaissance de la ProMP courante, en fonction du type de guidage. Un résultat intéressant est qu’en utilisant la méthode de modélisation (*c.f.*, Figure 5.12a), le robot est totalement capable de reconnaître la ProMP qui correspond au mouvement initié, lorsqu’il observe 70% de ce mouvement, quel que soit le type de guidage. De surcroît, lorsqu’il s’agit de guidage multimodal, le robot ne fait plus du tout d’erreur de reconnaissance à partir de ces observations.

De plus, avec la méthode du maximum de vraisemblance, il n’a fait qu’une erreur de recon-

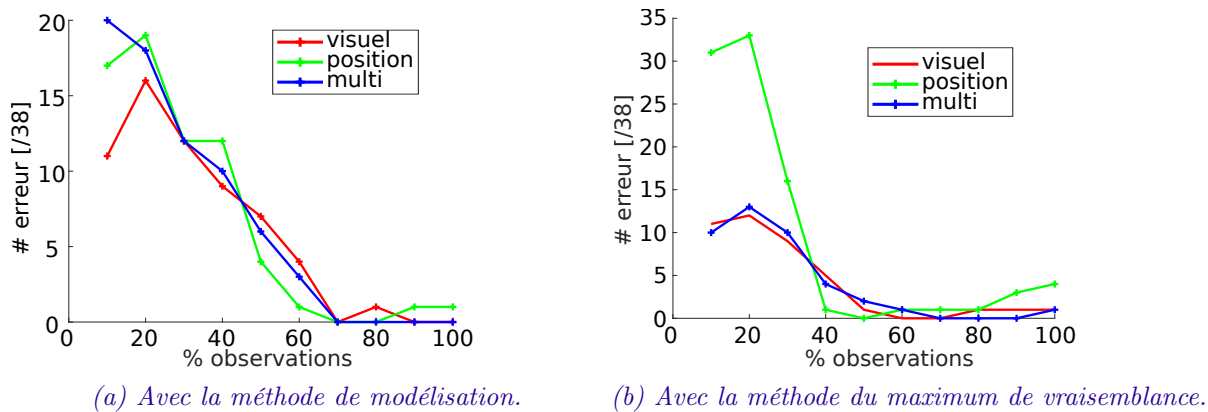


FIGURE 5.12 – Erreur de prédiction selon la modalité utilisée.

naissance sur les 38 tests (soit 2% d'erreur), lorsque le guidage multimodal est utilisé et que le robot a perçu au moins 70% de la trajectoire.

Ainsi, l'utilisation de mesures multimodales améliore clairement la reconnaissance de la ProMP mise en jeu, même si ces mesures ne permettent pas d'améliorer la précision de la trajectoire prédite (c'est-à-dire, que la distribution a posteriori n'est pas plus précise).

5.6 Conclusions

Cette étude présente une méthode multimodale qui permet au robot de prédire la trajectoire que son partenaire souhaite qu'il exécute, lors de leur interaction. Cette prédiction se base sur des indices visuels et/ou haptiques. Cette méthode a été testée dans un scénario d'interaction entre l'utilisateur et le robot *iCub*, où ce dernier devait attraper un objet en suivant différentes trajectoires. L'utilisateur guide physiquement le bras du robot et/ou effectue des mouvements de tête afin de guider le robot.

Il s'agit donc directement d'une amélioration de l'étude précédente (*c.f.*, Section 4 et Article [D.P]). Lorsque l'utilisateur décide de guider le robot par interaction physique, celui-ci utilise l'information haptique afin de reconnaître l'action courante (c'est-à-dire, la ProMP qui correspond au mouvement initié). Puis, il prédit la poursuite du mouvement et ainsi la position but à atteindre, ainsi que la durée de ce mouvement. À la fin de l'étude précédente, nous avons discuté de la limitation de ce type de guidage, qui est que le robot est incapable de déterminer quelle est la primitive de mouvement à suivre lorsque les premières observations des trajectoires sont ambiguës, par exemple lorsque des ProMPs sont identiques en début de mouvement. C'est pourquoi, dans cette deuxième étude, on permet à l'utilisateur de guider visuellement le robot pour résoudre ce problème (en effet, le mouvement entier peut être présenté rapidement par l'utilisateur.)

En ce qui concerne le guidage visuel, la même prédiction est faite en utilisant la direction du regard du partenaire, approximé ici par l'orientation de sa tête. L'association entre les informations du regard et les primitives de mouvement du bras du robot est effectuée à l'aide d'une phase d'apprentissage multimodal.

La modalité visuelle a deux avantages principaux. Premièrement, elle n'exige pas que le partenaire touche physiquement le robot lors de la phase de guidage. Deuxièmement, elle permet

une reconnaissance plus rapide de la primitive d'action que lorsque l'utilisateur guide physiquement le robot. Cependant, les résultats montrent qu'en utilisant le guidage visuel plutôt que le guidage physique, la précision des prédictions diminue légèrement. Cela peut être dû au manque de précision du logiciel utilisé pour estimer l'orientation du regard (*c.f.*, Section 2.3) mais aussi parce que lorsque l'utilisateur effectue un mouvement de tête, c'est forcément moins précis que lorsqu'il guide le bras du robot. De plus, il est aussi difficile pour un humain de guider par le regard avec sa tête plutôt qu'avec les yeux. C'est pourquoi on exagère les mouvements de tête, ce qui peut être un peu artificiel.

Avec ces résultats, on pourrait se demander s'il ne serait pas plus pertinent que le robot apprenne uniquement à reconnaître la position finale de l'orientation de la tête de l'utilisateur, plutôt qu'une trajectoire entière, puisque cette trajectoire n'est utilisée que pour la reconnaissance du type de trajectoire et non pas pour l'adapter. Cependant, notons que dans le scénario testé, la position finale à atteindre est identique, ce qui fait qu'en se limitant à l'information de la position finale à atteindre, le robot ne pourrait pas reconnaître la trajectoire à suivre. De plus, deux positions proches auraient finalement la même orientation de la tête, car l'humain affine sa focalisation du regard avec les yeux.

Afin d'améliorer la précision des prédictions visuelles, nous envisageons différentes possibilités. La *Kinect* pourrait être utilisée afin d'avoir des données plus pertinentes. De plus, un autre logiciel de reconnaissance faciale pourrait être utilisé à la place d'*Intraface* : par exemple, lorsqu'une solution plus invasive n'est pas dérangeante, un *eye-trackor*⁴⁴ permettrait d'affiner la reconnaissance de la zone de focalisation du regard. Il est également possible d'ajouter une autre modalité "non humaine" afin de permettre au robot de dépasser les compétences de prédiction humaine, en guidant par exemple le robot à partir d'une montre. Celle-ci contiendrait des capteurs qui permettent de détecter la position et l'orientation du bras de l'utilisateur. Le robot utiliserait alors ces mesures afin d'apprendre et de reconnaître des ProMPs.

En ce qui concerne la prédiction à partir d'observations multimodales, les résultats montrent que l'ajout de la reconnaissance visuelle à la reconnaissance physique n'améliore pas la précision de la trajectoire prédite (c'est-à-dire, que la précision de la distribution a posteriori n'est pas améliorée). Par contre, cela améliore l'inférence du type de ProMP.

Ainsi, pour avoir les meilleures compétences de prédiction, le robot peut utiliser l'information multimodale afin de reconnaître le mouvement ou l'action à exécuter, puis l'information physique afin de prédire la trajectoire future en fonction des premières observations.

Une limitation de cette étude est que lors du guidage multimodal, soit le robot doit être guidé par deux partenaires humains (l'un devant le robot pour le guider avec sa tête et l'autre pour le guider physiquement), soit un même utilisateur doit exécuter les deux types de guidages l'un après l'autre. Pour répondre à ce problème, nous souhaitons utiliser dans la prochaine étude la combinaison *Xsens*, qui récupère l'information des positions cartésiennes de membres de l'utilisateur au cours du temps. Le robot pourra ainsi être en mesure de prédire les mouvements qu'il doit effectuer en fonction des mouvements corps complet de l'utilisateur, sans que celui-ci n'ait besoin de guider le robot. Cette méthode permettra notamment de diminuer au maximum la charge mentale et physique des personnes.

La prochaine étude concerne donc une méthode qui permet au robot d'interpréter et d'effectuer des prédictions de trajectoires à partir de mouvements naturels corps complet de son partenaire.

44. Logiciel qui détecte l'orientation des pupilles

Compréhension de mouvements corps-complet humains (Étude 3)

Prédiction de l'intention



Réaction



FIGURE 6.1 – Scénario représentant comment les capacités de prédiction humaines sont utilisées lors de collaboration entre humains. La personne de gauche anticipe les actions de sa partenaire et exécute une action d'assistance appropriée. Dans cette étude, le robot apprend à prédire l'intention de son partenaire. Le but à plus long terme sera qu'il utilise cette capacité afin choisir l'action d'assistance à exécuter.

6.1 Résumé

Pour interagir entre eux, les êtres humains ont développé des compétences de prédiction de mouvements corps complet de leurs partenaires, après avoir observé les premiers instants de ces mouvements. Par exemple, dans l’illustration de la Figure 6.1, la personne sur la gauche utilise cette capacité afin d’éviter à l’autre personne de faire un mauvais mouvement.

En informatique, qu’il s’agisse, comme dans notre cas, de créer des robots collaboratifs capables d’anticiper des actions humaines, ou de créer des technologies d’assistance capables d’alerter des personnes lorsque des mouvements d’humains ne sont pas ergonomiques ou potentiellement dangereux, il est nécessaire de créer un algorithme capable de prédire la poursuite de mouvements corps complet initiés. La difficulté d’un tel algorithme et que, lorsqu’il s’agit de mouvement corps complet, les informations à traiter sont de grandes dimensions et provoquent généralement des coûts calculatoires importants. Or, pour être efficace, un tel logiciel doit fonctionner au moins en temps réel mou.

Pour répondre à cette problématique, nous proposons dans cette étude un algorithme de prédiction de la poursuite de trajectoires lorsque celles-ci sont représentées par de nombreuses données (c’est-à-dire, que l’ensemble de données à apprendre est de dimension importante), en cartographiant ces trajectoires dans un espace latent réduit, à l’aide de variantes d’auto-encoders (AEs). En ce qui concerne la prédiction, elle est toujours effectuée à l’aide de ProMPs, mais cette fois-ci à partir des trajectoires compressées dans l’espace latent, ce qui réduit considérablement le temps de calcul de la prédiction et permet ainsi d’utiliser la méthode dans des applications en temps réel.

Pour ce faire, nous proposons deux méthodes qui combinent la capacité de prédiction des ProMPs avec celle de réduction de dimensionnalité des AEs et VTSFE (*c.f.*, Sections 1.8 et 3.4). Nous les appelons respectivement AE-ProMPs et VTSFE-ProMPs. Ces deux méthodes suivent deux idées différentes : dans AE-ProMPs, la compression concerne les postures, tandis que dans VTSFE-ProMPs, la compression concerne la trajectoire posturale entière (*c.f.*, Section 3.4 pour plus de détails sur ces méthodes). Dans les deux cas, la méthode ProMP permet de prédire les trajectoires futures dans l’espace latent.

Afin d’évaluer ces méthodes, les expériences effectuées dans cette étude consistent en la prédiction de la poursuite de mouvements corps complet correspondant à sept actions différentes effectuées par un être humain et enregistrées par une combinaison de suivi de mouvement XSENS (*c.f.*, Section 2.4).

Nous montrerons que pour prédire les trajectoires futures à partir d’observations initiales, la méthode AE-ProMPs est plus rapide et plus performante que VTSFE-ProMPs ou la méthode ProMP seule.

6.2 Introduction

Afin de collaborer efficacement, les êtres humains sont capables de prédire le mouvement futur de leurs partenaires [111]. Cette capacité n’implique pas seulement la “prédiction de l’intention”, souvent formalisée comme la prédiction de l’objectif d’une action, mais aussi la “prédiction du mouvement futur prévu”, que nous avons récemment formalisée comme la prédiction de la trajectoire future calculée à partir d’observations initiales de celle-ci (*c.f.*, Sections 4, 5 et [D.P]).

Cette capacité de prédiction du mouvement futur est également cruciale pour les robots collaboratifs (afin qu’ils puissent anticiper les actions humaines) ainsi que pour les technologies

d’assistance (afin par exemple d’alerter si un mouvement particulier est non ergonomique ou potentiellement dangereux pour la santé humaine [115]). Pour agir en conséquence, cette prédiction doit être rapide et être faite dès les premières observations des mouvements, malgré la variabilité et la forte dimensionnalité de ceux-ci.

Lors des deux premières études, nous avons donc utilisé la méthode ProMP afin d’apprendre la distribution des trajectoires des actions robotiques et afin de prédire les mouvements futurs prévus lors d’interactions humain-robot. Dans ces études, un robot utilisait une partie initiale d’une trajectoire, appelée “trajectoire partielle”, afin de prédire sa continuation jusqu’au but [D.P]. Les trajectoires étaient présentées au robot que ce soit à l’aide d’interaction physique, de repères visuels, ou encore les deux à la fois [D.M]. Ces expériences se limitaient aux mouvements du bras du robot, tout en combinant les informations cinématiques et des signaux dynamiques.

Dans l’étude actuelle, nous nous intéressons à la prédiction de la poursuite de mouvements corps entier d’êtres humains, à partir d’observations initiales de trajectoires partielles. Un enjeu de cette étude est que cette prédiction soit suffisamment rapide afin que le robot puisse planifier une action d’assistance appropriée si nécessaire. Puisque nous voulons prédire les trajectoires futures de tous les membres de l’humain exécutant l’action, notre prédiction est effectuée dans un espace dimensionnel élevé. Or, les ProMPs ne sont pas optimales d’un point de vue temps de calcul, comme le vérifie la première expérience de cette étude (*c.f.*, Section 6.4.1), et ne convient donc pas pour notre application.

Afin de résoudre ce problème, nous proposons maintenant de réduire la dimensionnalité de l’espace de données. Les trajectoires de haute dimension sont alors mappées dans un espace latent à petite dimension (*latent space*, LS). Ensuite, les ProMPs sont apprises directement depuis ce LS, à partir duquel nous prédisons également le futur de ces trajectoires. La compression se fait à l’aide d’un auto-encoder (AE), qui *encode* d’abord les trajectoires originales dans le LS, puis *décodes* les trajectoires prédites et compressées du LS, vers l’espace original de haute dimension.

Nous appelons cette méthode AE-ProMP et est présentée dans la Section 3.4.3. La Figure 6.2 représente le concept de cette étude.

Nous comparons cette méthode avec une autre méthode que nous avons créée, nommée VTSFE-ProMP (basée sur la méthode VTSFE, présentée dans la Section 3.4.2), qui se distingue de AE-ProMP par le fait qu’elle encode non plus des postures, mais des trajectoires posturales.

Si cette dernière méthode est intéressante pour sa capacité à produire un espace latent cohérent et à préserver la continuité des trajectoires, elle est coûteuse en durée d’apprentissage et nécessite des calculs compliqués.

De plus, la complexité de cette méthode est due à la modélisation dynamique du LS, or cette modélisation dynamique n’est pas utile dans notre cas, puisque, contrairement à dans [48], nous ne considérons pas les trajectoires compressées dans le LS individuellement, mais nous apprenons des distributions de probabilité à partir de l’ensemble des trajectoires compressées. Or, les imprécisions des trajectoires compressées, qui correspondent à des “à-coups” dans l’espace latent, n’ont pas d’impact lorsque les distributions sont apprises, puisque ces distributions sont calculées à partir de l’ensemble des trajectoires de démonstrations.

6.3 Formulation du problème

Le but de ces expériences est donc de comparer les trois méthodes nommées ProMP, AE-ProMP et VTSFE-ProMP lorsqu’elles sont utilisées pour prédire des mouvements humains futurs,

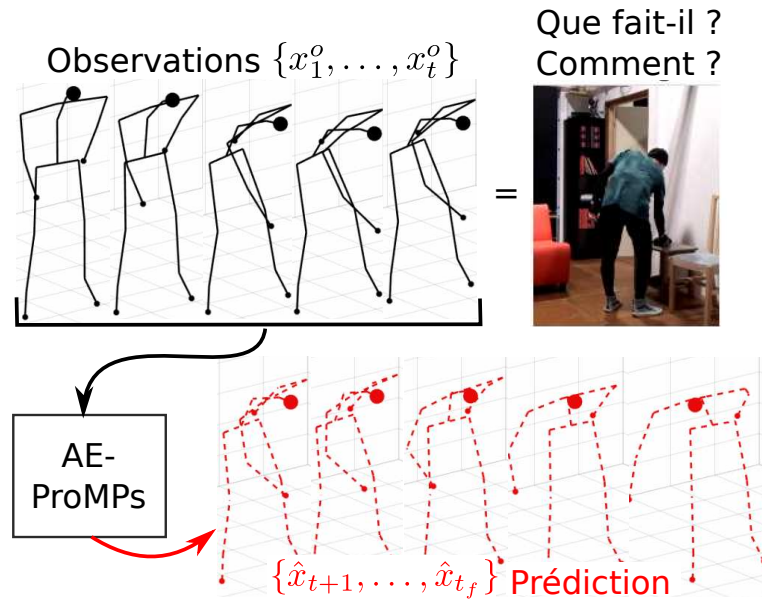


FIGURE 6.2 – *Concept de l'étude* – Le but est de prédire au temps t la poursuite du mouvement corps complet initié par l'humain ($\hat{x}_{[t+1:t_f]}$) à partir des t observations partielles de ce mouvement ($x_{[1:t]}^o$). La méthode AE-ProMP est utilisée pour faire cette prédiction.

à partir d'observations initiales.

Pour ce faire, nous utilisons une base de données d'actions proposée dans [146], composée de 10 trajectoires de démonstrations pour sept actions différentes, actions qui sont représentées dans la Figure 6.3 : se pencher en avant, se pencher fortement en avant, soulever une boîte, donner un coup de pied, ouvrir une fenêtre, marcher et se tenir debout.

Ces 70 trajectoires de démonstrations ont été enregistrées à l'aide de la combinaison XSens MVN [235] (*c.f.*, Section 2.4). Cette combinaison permet le suivi des mouvements de l'humain qui la porte, à l'aide d'un modèle squelettique composé de 23 degrés de liberté. Ce modèle squelettique représente ainsi la cinématique de la posture de l'humain.

À partir de ces enregistrements, nous récupérons les positions cartésiennes en 3-dimensions de segments provenant du modèle cinématique de l'humain calculé dans XSens MVN Studio. Ainsi, la posture de l'opérateur humain est représentée par $3 \times 23 = 69$ variables, correspondant aux positions cartésiennes des membres de l'humain.

Chaque trajectoire de démonstration est alors ré-échantillonnée pour être représentée par soixante-dix pas de temps ($t_f = 70$). Ce ré-échantillonnage est nécessaire, car la méthode VTSFE a besoin d'une trajectoire à durée fixe, comme expliqué dans [47], et pour pouvoir comparer les méthodes entre-elles, nous appliquons ce ré-échantillonnage quelque soit la méthode utilisée.

Pour chacune des trois méthodes présentées ci-dessous, les ProMPs sont modélisées par 10 fonctions de base radiales. De plus, la matrice de covariance complète est calculée, afin d'enregistrer la redondance d'information entre les différentes positions articulaires de chaque posture.

6.3.1 ProMPs-seuls

Dans cette expérience, les ProMPs des 7 actions différentes sont donc calculées sans réduction de dimension. L'objectif est de montrer la précision de la prédiction lorsque la méthode ProMP

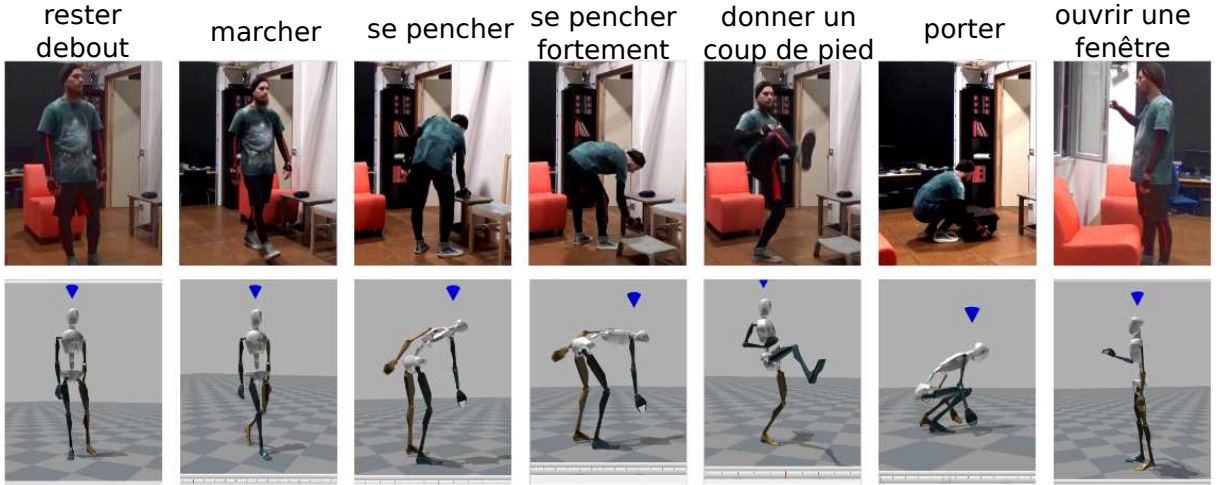


FIGURE 6.3 – *3e étude – Scénario.* Les sept actions provenant des démonstrations fournies par la base de données [146], avec l’estimation cinématique de la posture de l’humain effectuée à l’aide du studio Xsens MVN.

est utilisée, mais aussi sa limitation principale, qui correspond à l’augmentation du temps de calcul lorsque la dimension des données d’entrées augmente. Dans cette expérience,

le vecteur $\xi(t) \in \mathbb{R}^{69}$ de l’Équation 3.1 contient toutes les positions des segments qui représentent le mouvement corps complet de l’humain : $\xi(t) = [a_{1,1}(t), \dots, a_{23,3}(t)]^\top$, avec $a_{i,j}$ la position du i^{eme} membre, de coordonnée $j \in \{x, y, z\}$.

6.3.2 AE-ProMPs

Nous utilisons ici un simple auto-encodeur afin de compresser, dans un espace latent à petite dimension, les 69 données représentant les postures humaines.

Dans cette expérience, les ProMPs sont apprises à partir des postures encodées (c’est-à-dire, à partir de l’espace latent de l’auto-encodeur), avec par exemple $\xi(t) = [z_1(t), z_2(t)]^\top$ quand $R = 2$.

Pour effectuer cette compression, nous utilisons l’auto-encodeur suivant :

- Une couche d’entrée composée de N unités $x = \{x_1, \dots, x_{69}\}$ correspondant aux données d’origine des postures (c’est-à-dire, 69 unités). Nous appelons les unités de cette couche $x = \{x_1, \dots, x_{69}\}$.
- Une couche compressée (espace latent) dont le nombre R d’unités peut être choisi (*par ex.*, 10 unités) et qui correspondent à $z = \{z_1, \dots, z_R\}$, avec R la dimension de l’espace latent.
- Une couche de sortie permettant de récupérer la posture décodée, et qui a donc la même dimension N que la couche d’entrée (c’est-à-dire, 69 unités). Nous appelons ces unités $x_{rec} = \{x_{1,rec}, \dots, x_{69,rec}\}$.
- Enfin, deux couches cachées, l’une entre la couche d’entrée et la couche compressée et l’autre entre la couche compressée et la couche de sortie (*par ex.*, 500 unités). Nous appelons ces couches h_j avec $j \in [1, \dots, R]$ et sa i^{eme} unité : h_{jx_i} .

Les poids de ce réseau de neurones sont initialisés à l’aide de l’initialisation Xavier [93], ce qui signifie que les poids sont initialisés à l’aide d’une distribution uniforme.

En ce qui concerne la fonction d’activation de toutes ces unités, nous choisissons “leaky ReLU” [236], qui est similaire à “ReLU” (unité linéaire rectifiée), mais où la fonction n’est pas

nulle lorsque les valeurs sont négatives : $f(x) = \mathbf{1}_{x < 0} \alpha x + \mathbf{1}_{x \geq 0} x$, avec $\alpha = 0.5$ dans notre cas. Cette fonction a été choisie après avoir comparé ses performances avec les fonctions d'activation sigmoïde et ReLu.

Enfin, les réseaux neuronaux sont appris en utilisant comme fonction objectif l'erreur quadratique moyenne entre x et x_{rec} et de l'optimiseur ADAM [132].

Concernant la première phase d'apprentissage (*c.f.*, Section 3.4.3), l'auto-encodeur a été entraîné à l'aide de $\frac{2}{3}$ des postures de chacune des 70 trajectoires de démonstrations, correspondant à un ensemble de 30916 postures. Ensuite, l'auto-encodeur a été testé à l'aide du dernier tiers des postures.

Concernant la seconde phase d'apprentissage, les ProMPs ont été apprises à l'aide de 69 trajectoires de démonstrations sur les 70, après que celles-ci soient compressées par l'auto-encodeur. Nous les appelons trajectoires de démonstration compressées. La dernière trajectoire est utilisée afin de tester la capacité de prédiction du robot.

Ces étapes d'apprentissage sont effectuées 35 fois, ce qui correspond à 5 tests pour chacune des 7 actions, en utilisant le *leave-one-out cross-validation*.

6.3.3 VTSFE-ProMPs

Nous utilisons dans cette expérience la méthode VTSFE afin d'encoder directement la trajectoire posturale (correspondant à 69 dimensions \times 70 instants = 4830), dans un espace latent dynamiquement cohérent, à partir duquel les ProMPs sont apprises afin d'effectuer des prédictions.

Dans cette expérience, différentes dimensions R de l'espace latent sont testées, par exemple $\xi(t) = [z_1(t), z_2(t)]^\top$ lorsque $R = 2$. L'objectif est ici de vérifier si l'encodage de la trajectoire entière au lieu de postures instantanées (AEs) améliore non seulement l'espace latent (comme montré dans [48] et [47]) mais aussi la capacité de prédiction.

6.4 Résultats et Discussion

Nous nous intéressons à l'évaluation de la performance des trois méthodes du point de vue du temps de calcul, de la complexité, de l'inférence et des capacités de prédiction.

Tous les résultats (ainsi que leurs représentations à l'aide de boîtes à moustaches) se font, comme cela a été précisé en amont, à l'aide de *leave-one-out cross-validation* lors d'essais hors-ligne et correspondent à des trajectoires représentées par 70 échantillons (trajectoires ré-échantillonnées).

Pour tester la performance des méthodes proposées, nous utilisons trois mesures de distance différentes :

1. Err_{AE} ou *erreur d'auto-encodage*, évalue les performances de l'auto-encodeur.

Cette erreur correspond à la distance moyenne entre la trajectoire reconstruite $x_{rec} = dec(enc(x))$ et la trajectoire réelle x :

$$Err_{AE} = \frac{1}{t_f N} \sum_{i=1}^{t_f} \sum_{j=1}^N |x_{j,rec}(i) - x_j(i)|$$

2. Err_P ou *erreur de prédiction*, évalue les performances de prédiction.

Cette erreur correspond à la distance moyenne entre la trajectoire reconstruite x_{rec} et la reconstruction de la trajectoire prédite dans l'espace latent $\hat{x} = dec(\hat{\xi})$:

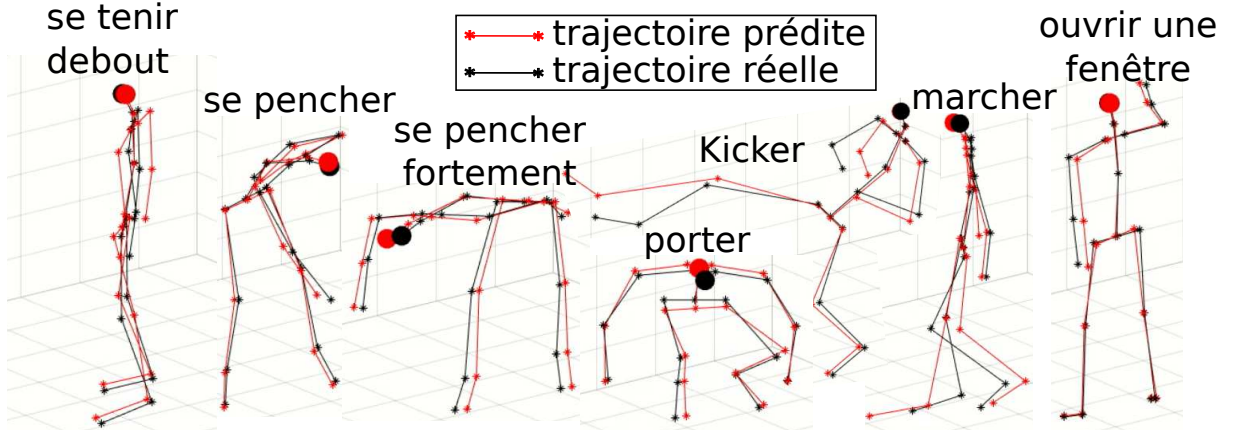
$$Err_P = \frac{1}{t_f N} \sum_{i=1}^{t_f} \sum_{j=1}^N |x_{j,rec}(i) - \hat{x}_j(i)|$$

3. Err_{AE+P} ou *erreur globale*, évalue la méthode dans sa totalité, donc la performance à la fois de l'auto-encodeur et de la prédiction.

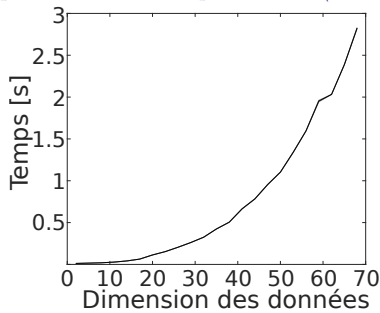
Cette erreur correspond à la distance moyenne entre la trajectoire réelle x et celle prédite \hat{x} :

$$Err_{AE+P} = \frac{1}{t_f N} \sum_{i=1}^{t_f} \sum_{j=1}^N |\hat{x}_j(i) - x_j(i)|$$

6.4.1 Prédiction du futur d'un mouvement, à l'aide de ProMPs seulement

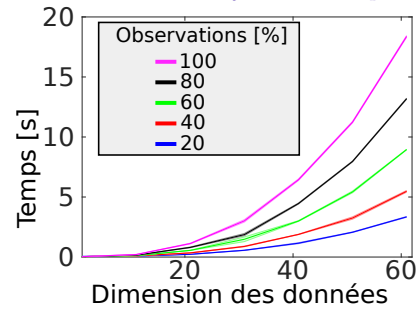


(a) Quelques postures provenant de la vidéo [MultiDim] représentant les trajectoires réelles (en noir) et les prédictions correspondantes (en rouge) après avoir observé 10% de la trajectoire complète (6 échantillons).



(b) Temps de calcul des distributions

à partir de 10 trajectoires de démonstration.



(c) Temps nécessaire à la prédiction.

FIGURE 6.4 – *ProMPs-seulement* – Les trajectoires sont représentées par 70 échantillons.

La Figure 6.4 représente la prédiction effectuée à l'aide des ProMPs dans l'espace de données d'origine ($N = 69$) après avoir observé 10% de la trajectoire totale (c'est-à-dire, après 6 échantillons). Les images proviennent de la vidéo [MultiDim] et ont été prises à des moments représentatifs des actions.

Bien que cette méthode soit performante en ce qui concerne la représentation de trajectoires [171, 143], son temps de calcul nécessaire à la prédiction augmente de manière quadratique avec le nombre de données représentées par les ProMPs [50].

Dans cette étude justement, la prédiction se fait sur un vecteur de 69 dimensions.

La Figure 6.4b représente le temps moyen nécessaire au calcul des distributions de mouvement, lorsque toutes les démonstrations sont utilisées, et la Figure 6.4c représente le temps de calcul moyen nécessaire pour prédire la suite du mouvement.

Ces temps de calcul sont trop longs pour l'application ciblée, où la trajectoire humaine future doit être prédite en quelques millisecondes.

Cette limitation motive notre approche qui consiste à réduire la dimension des données, dans le but de développer des capacités prédictives plus rapides.

La vidéo présentée dans [humanoids] résume les résultats de cette expérience.

6.4.2 Prédiction du mouvement future, à l'aide de *AE-ProMPs*

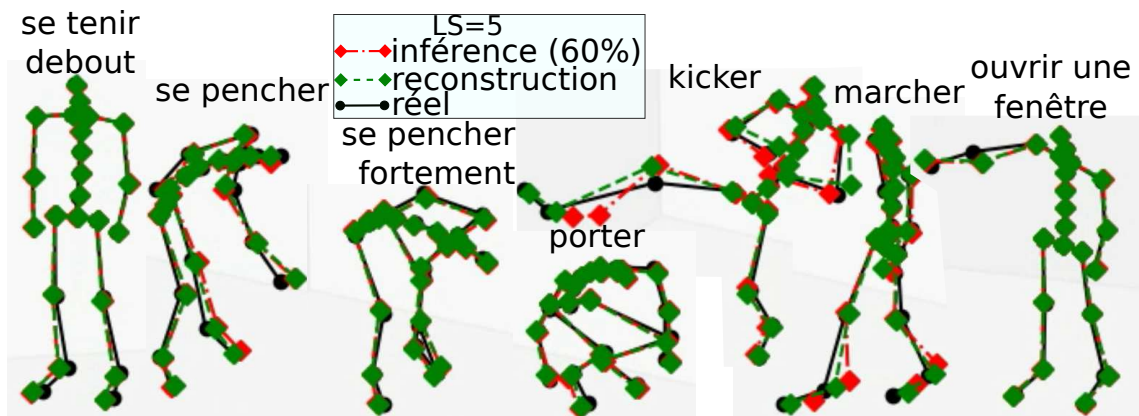


FIGURE 6.5 – *AE-ProMPs* – Quelques postures provenant de la vidéo [AE-ProMPs] représentant les trajectoires réelles (en noir), la reconstruction de celles-ci après compression dans un espace latent 5D (en vert) et les prédictions correspondantes (en rouge) après avoir observé 60% de la trajectoire complète (42 échantillons).

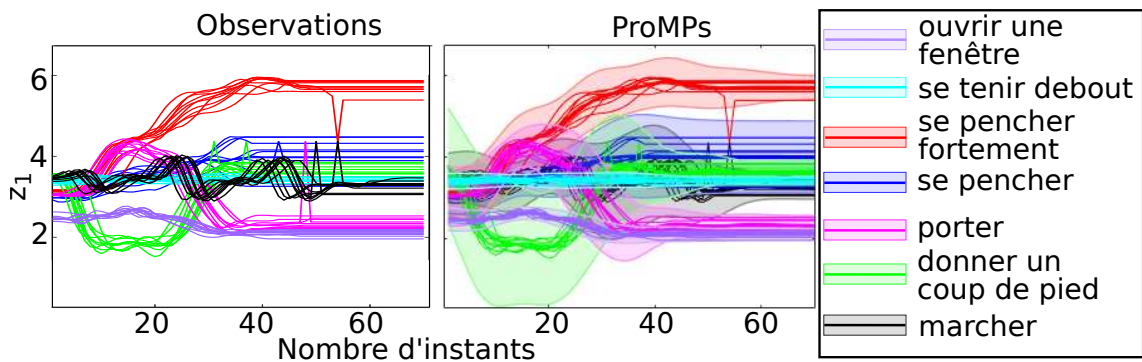


FIGURE 6.6 – *AE-ProMPs* – Exemple d'un espace latent 5D : représentation des trajectoires de démonstration après compression pour la dimension z_1 et leur ProMP correspondante.

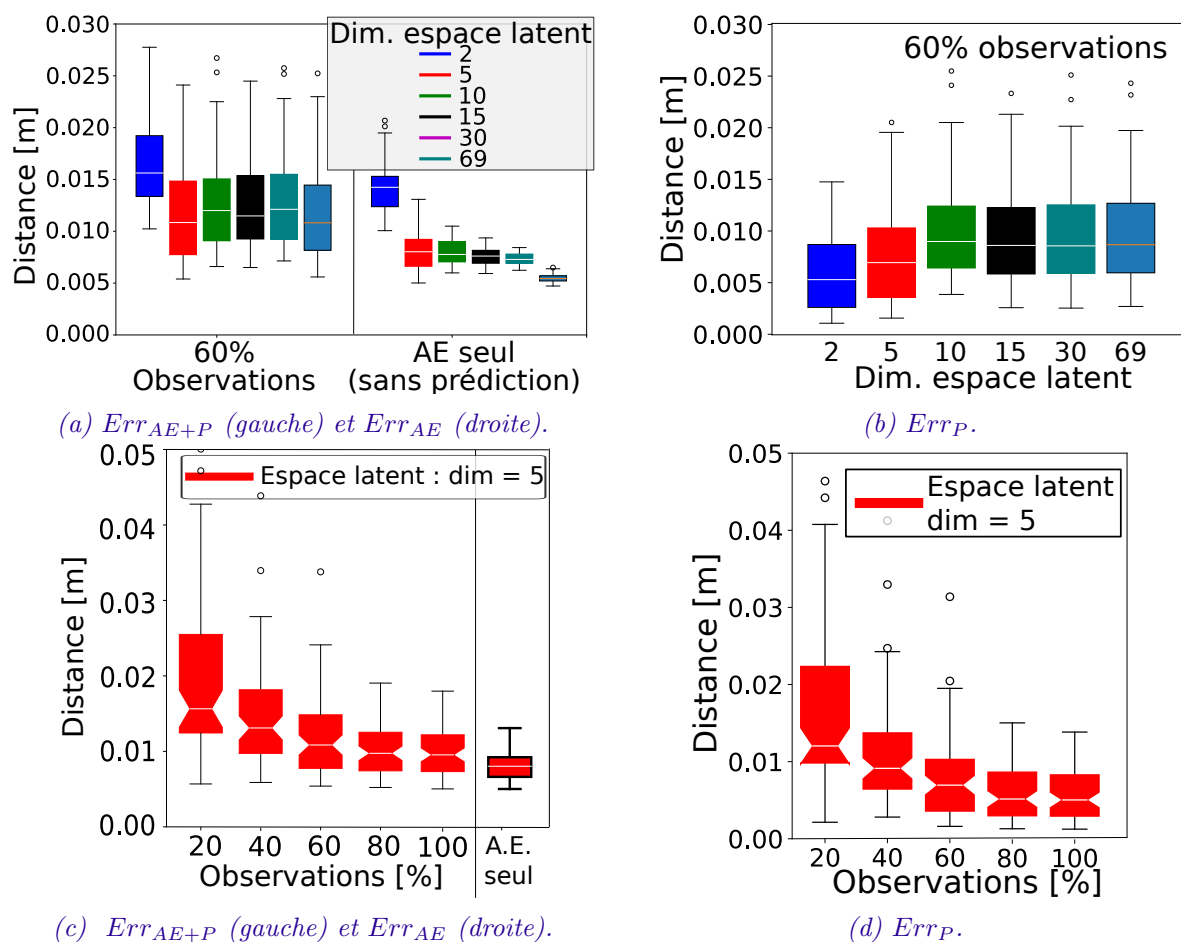


FIGURE 6.7 – **AE-ProMPs** - Précision en fonction de la dimension de l'espace latent (haut) et du pourcentage d'observations (bas).

La Figure 6.5 représente la prédiction des trajectoires humaines à partir de leur compression dans un espace latent de dimension 5. De nouveau, les images ont été prises à des moments représentatifs des différents mouvements, à partir de la vidéo [AE-ProMPs].

La Figure 6.6 représente des extraits des trajectoires de l'espace latent, pour la dimension z_1 . On peut voir que les quelques irrégularités dans ces trajectoires compressées (figure “observation”) n’affectent pas les ProMPs (figure “ProMPs”) et donc, ne causent pas de problème lors de la phase de prédiction.

La Figure 6.7 représente, à l’aide de boîtes à moustaches, la précision des AE-ProMPs en fonction de la dimension de l'espace latent ainsi qu’en fonction du pourcentage d’observations partielles à partir duquel la prédiction est effectuée.

Les boîtes à moustaches des graphiques des Figures 6.7a et 6.7b comparent les différentes distances d’erreurs, avec de gauche à droite : l’erreur de la méthode complète (comportant l’erreur du à l’auto-encodeur et de la prédiction) ; l’erreur d’auto-encodage seulement ; puis l’erreur de prédiction seulement. Dans tous les cas, ces résultats sont obtenus lorsque les prédictions sont faites à partir de l’observation de 60% des trajectoires.

Les résultats de la Figure 6.7a montrent que la distance d’erreur diminue lorsque la taille de l’espace latent augmente. De plus, ces résultats montrent que l’espace latent de dimension 5

semble un bon compromis afin d’avoir de bonnes capacités d’inférence, tout en évitant des temps calculatoires trop longs.

Les résultats présentés dans la Figure 6.7b montrent au contraire que, dans le cas où la distance d’erreur correspond seulement à l’inférence à l’intérieur de l’espace latent, c’est-à-dire, entre la trajectoire compressée réelle et la trajectoire compressée prédite, cette distance augmente avec l’augmentation de la dimension de l’espace latent. En effet, plus l’espace latent est de petite dimension, moins la trajectoire compressée contient d’informations provenant de la trajectoire réelle et donc, l’erreur est plus petite, car ces trajectoires compressées contiennent moins de variabilité. Et quand cet espace latent est petit et que donc la trajectoire compressée contient moins d’information concernant la trajectoire réelle, alors la décompression de la trajectoire est moins bonne, comme le montrait la Figure 6.7a. On peut remarquer que cette distance diminue avec l’augmentation de la dimension de l’espace latent, ce qui était attendu. À partir de ces graphiques, nous choisissons la dimension $R = 5$ pour l’espace latent, puisqu’il s’agit d’un bon compromis entre bonne capacité de prédiction et temps de calcul.

Les boîtes à moustaches des graphiques des Figures 6.7c et 6.7d comparent les différentes distances d’erreurs en fonction du nombre d’observations (en pourcentage), lorsque l’on choisit pour dimension d’espace latent $R = 5$. Ces graphes permettent de visualiser le fait que l’erreur induite par l’auto-encodage Err_{AE} est relativement faible et que l’erreur globale est surtout affectée par l’erreur de prédiction Err_P qui diminue, comme prévu, avec l’augmentation du nombre de postures observées. Cela était d’ailleurs déjà visible dans le graphe (a) de cette figure.

En conclusion, en choisissant un espace latent de dimension $R = 5$, et en effectuant la prédiction après avoir observé 60% de celle-ci (42 échantillons), la méthode AE-ProMP permet de prédire la trajectoire future “corps complet” avec une distance d’erreur d’environ $1cm$, ce qui est une très bonne performance pour l’application ciblée.

La Figure 6.8 présente l’ensemble des données d’analyse de cette expérience.

6.4.3 Prédiction du mouvement futur à l’aide de *VTSFE-ProMPs*

La Figure 6.9a représente des exemples de postures humaines des trajectoires prédites à l’aide de la méthode VTSFE-ProMP, encodées dans un espace latent de dimension $R = 5$.

Les boîtes à moustaches présentées dans les autres graphes de la Figure 6.9 donnent des informations concernant la précision des VTSFE-ProMPs, en fonction du nombre d’observations (Figure 6.9c) et de la dimension de l’espace latent choisie (Figure 6.9b).

D’après la Figure 6.9b, on peut voir que la reconstruction de l’auto-encodeur n’est pas suffisamment précise. En effet, quelle que soit la dimension de l’espace latent choisie, les distances d’erreurs d’auto-encodage Err_{AE} ainsi que les distances d’erreurs globales Err_{AE+P} sont constantes et presque identiques (environ égales à $4cm$), ce qui suggère que l’erreur de prédiction Err_P est significativement plus petite que l’erreur d’encodage Err_{AE} , puisqu’elle n’influence pas la performance globale de la méthode.

Pour vérifier cette hypothèse, la Figure 6.9c compare l’erreur de prédiction Err_P pour deux dimensions de l’espace latent différentes (2 et 5), en fonction du nombre d’observations utilisé pour la prédiction. On remarque alors que Err_P diminue avec l’augmentation du nombre d’observations disponibles, et ce pour ces deux dimensions, validant l’hypothèse d’étude (*c.f.*, Section 3.1). En effet, lorsque la prédiction s’effectue après que le robot ait observé 20% de la trajectoire, la distance d’erreur est d’environ $5mm$, et dès 60% d’observation, moins de $2.5mm$. Cependant,

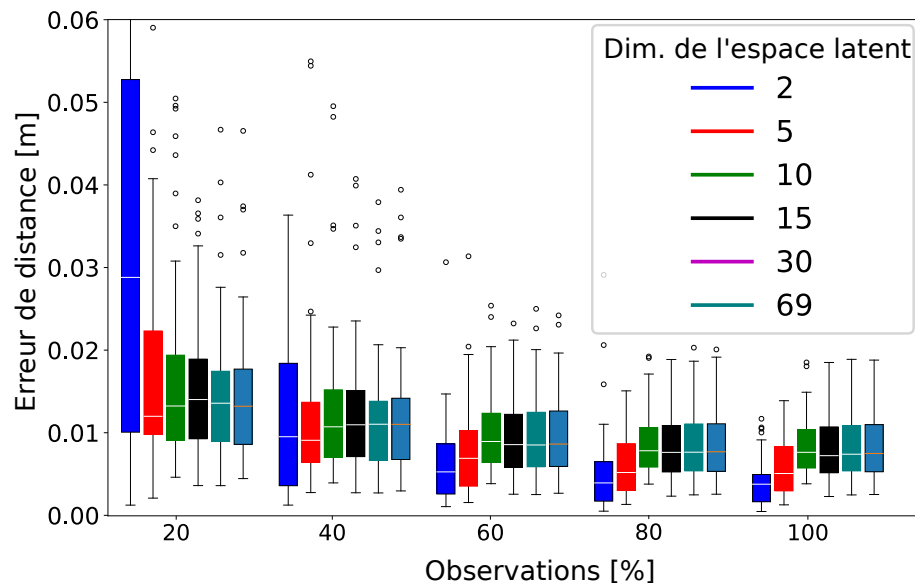
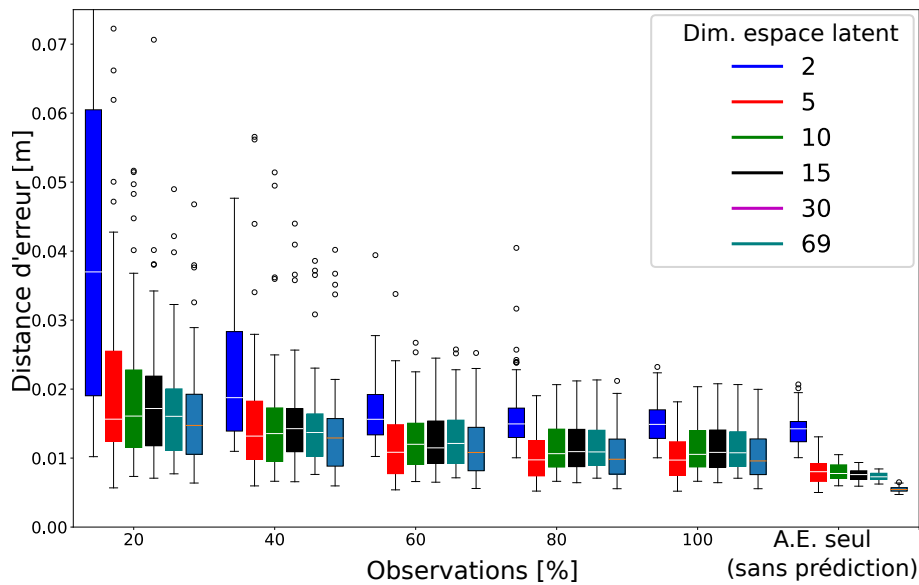
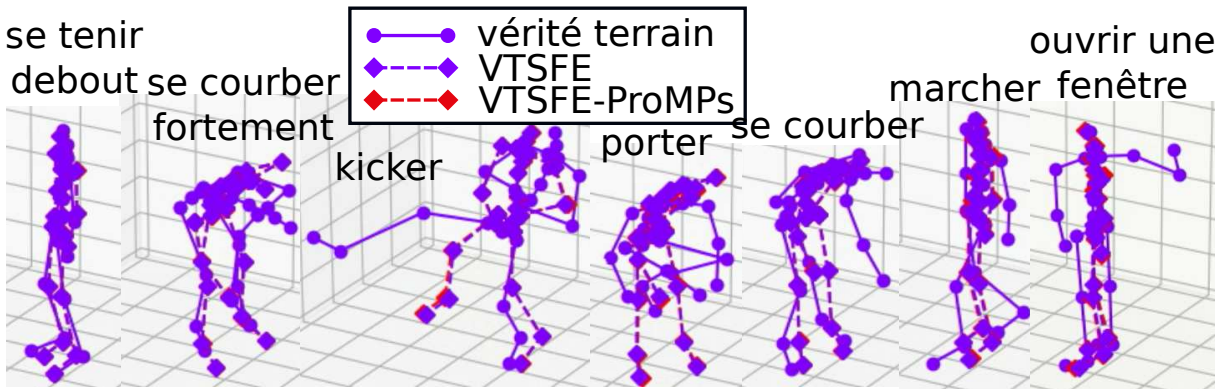
(a) Err_P .(b) Err_{AE+P} et Err_{AE} .

FIGURE 6.8 – **AE-ProMP** – Ensemble des statistiques. Distances d’erreurs pour les 70 tests effectués à l’aide de la méthode “leave-one-out-cross-validation” en fonction du nombre d’observations (en pourcentage) et de la dimension de l’espace latent choisi.

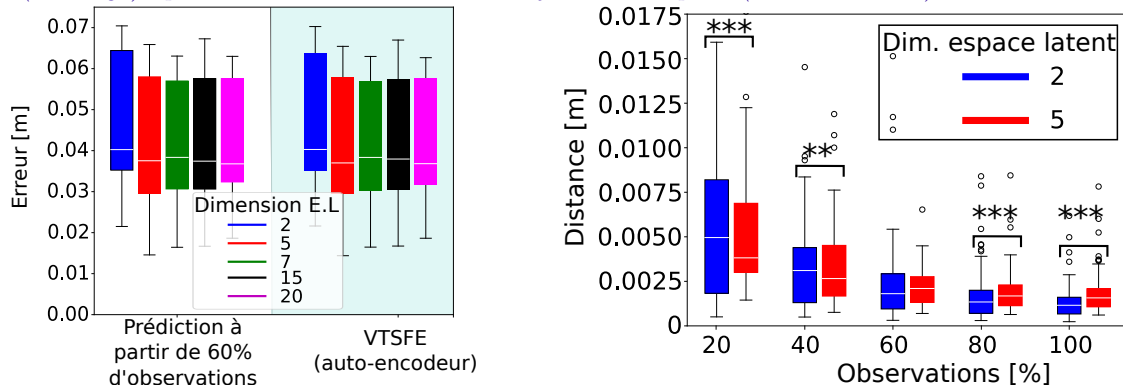
cette erreur est, en ordre de grandeur, de 1 à 2 fois plus petite que l’erreur d’encodage, ce qui vérifie que sa contribution n’a pas d’impact sur la performance globale de la méthode.

Ces résultats révèlent que cette méthode a un problème de construction de l’espace latent. Nous proposons ci-dessous deux hypothèses pouvant expliquer ce problème.

Premièrement, le fait de forcer les trajectoires de l’espace latent à être dynamiquement cohérent provoque un lissage trop important des postures encodées. L’espace latent serait alors



(a) Quelques postures provenant de la vidéo [VTSFE-ProMPs] représentant les trajectoires réelles (lignes violettes), la reconstruction de celles-ci après compression (pointillés violets) et les prédictions correspondantes (en rouge) après avoir observé 20% de la trajectoire complète (14 échantillons).



(b) Err_{AE+P} (gauche) et Err_{AE} (droite).

(c) Err_P .

FIGURE 6.9 – VTSFE-ProMPs - Expérience avec un espace latent de dimension $R = 5$. Les trajectoires sont représentées par 70 échantillons.

incapable d’encoder les fortes variations posturales des trajectoires, ce qui est cohérent avec nos observations (*c.f.*, vidéo [VTSFE-ProMPs]) où l’on peut voir que la méthode VTSFE n’encode pas correctement les mouvements lorsque la position de certains de leurs membres varie fortement, comme c’est le cas pour les genoux et les pieds lors des actions “coups de pied” et “marche”. Les trajectoires encodées restent ainsi proches d’une posture moyenne. Ce problème a été également observé dans [47], problème qui est d’ailleurs hérité de la méthode VAE-DMP. En comparaison, la méthode AE-ProMP n’a pas ce problème, puisque l’AE apprend à coder des postures instantanées, sans forcer une continuité posturale entre des échantillons successifs. Ainsi, la méthode AE-ProMP permet d’encoder les postures les plus extrêmes et de reconstruire correctement les trajectoires, même lorsque la position de certains de leurs membres varie fortement.

Deuxièmement, si l’on regarde comment est entraîné le réseau de neurones de l’auto-encodeur de la méthode VTSFE, on voit qu’il s’effectue à partir de 70 trajectoires de démonstration. Or, ces démonstrations sont utilisées pour entraîner $70 * 69 = 4830$ unités d’entrée, ce qui semble insuffisant.

La Figure 6.10 présente l’ensemble des erreurs en fonction du pourcentage de données observées.

6.4.4 Précision vs temps de calcul

Le Tableau 6.1 fournit une comparaison entre les trois méthodes testées en ce qui concerne la précision et le temps de calcul nécessaire à la prédiction du mouvement futur du corps complet.

Prédiction à partir de 20% d'observations		Précision de la prédiction [m]	Temps de calcul [s]
ProMPs (69 dimensions)	moyenne	0.0145	2.5378
	var	1.0038e-04	0.0357
VTSFE-ProMPs (L.S.= 5)	moyenne	0.04219	0.0565
	var	0.002	0.0024
AE-ProMPs (L.S.= 5)	moyenne	0.02793	0.0516
	var	0.003	0.0028

TABLEAU 6.1 – *Moyenne et variance de la distance d'erreur entre les trajectoires réelles et les trajectoires prédites, ainsi que les temps de calcul des prédictions pour les méthodes ProMP, VTSFE-ProMP et AE-ProMP.*

Bien que cette prédiction ne soit faite à partir de peu d'observations (20% des trajectoires), on peut voir que le temps de calcul des ProMPs est beaucoup plus long que les deux autres méthodes. De plus, la méthode VTSFE-ProMP est la moins précise, pour les raisons que nous avons déjà expliquées. Ainsi, pour l'application ciblée, la meilleure méthode est AE-ProMP, qui surpasse ProMP en ce qui concerne le temps de calcul et VTSFE-ProMP pour sa précision de prédiction et de reconstruction.

6.5 Conclusion

Dans cette étude, nous proposons une nouvelle méthode de prédiction de mouvements complexes, de haute dimension, appelée AE-ProMP. Cette méthode combine réduction de la dimension des trajectoires (à l'aide d'un AEs) et capacités de prédiction (à l'aide de la méthode ProMP).

L'AE comprime les postures dans un espace latent de petite dimension, puis décode ces postures compressées, afin de récupérer les postures dans leur dimension originale.

La méthode ProMP, quant à elle, permet d'effectuer la prédiction de la poursuite d'un mouvement initié, compte tenu des observations posturales initiales et des distributions de trajectoires apprises. Cette méthode est utilisée sur les trajectoires compressées dans l'espace latent de l'AE. Nos résultats montrent que la méthode AE-ProMP permet de prédire avec précision les mouvements corps complet, et ce, avec un faible temps calculatoire.

Nous avons comparé cette méthode avec les prédictions effectuées à l'aide de la méthode ProMP seule (c'est-à-dire, sans compression des données) ainsi qu'avec une méthode combinant les méthodes VTSFE et ProMP, où l'encodage conserve la dynamique des trajectoires. Les résultats montrent que la méthode ProMP seule met trop de temps à effectuer la prédiction, pour des applications que l'on souhaite être temps réel. Quant à la méthode VTSFE-ProMPs, bien que son espace latent soit dynamiquement plus cohérent, le décodage de la trajectoire est imprécis.

Il s'agit ici d'une première étude, permettant d'explorer l'idée de combiner des méthodes de compression et de prédictions de trajectoire. Dans cette étude, la méthode AE-ProMP fournit de bons résultats, c'est pourquoi nous aimerions améliorer cette méthode et pousser les expérimentations davantage. Nous présentons ci-dessous quelques exemples de telles améliorations et expérimentations.

Tout d’abord, on peut supposer que plus la dimension des données d’entrée est grande, moins la prédiction est précise. Pour aller plus loin dans cette étude, il serait donc intéressant de regarder comment la précision de la prédiction est impactée par une augmentation de la taille des données d’origine, pour une même taille d’espace latent.

De plus, puisque la compression de mouvements permet au robot de capturer les caractéristiques principales des mouvements, cela devrait lui permettre de généraliser ses connaissances. Ainsi, le robot devrait reconnaître des mouvements qu’il n’a jamais vus, tant que ceux-ci respectent les caractéristiques des mouvements appris. C’est pourquoi, pour étendre cette étude, nous chercherons à tester si le robot sait généraliser ses connaissances à des trajectoires ayant des caractéristiques communes à celles apprises. Par exemple, nous pourrions tester si le robot est capable de généraliser l’action “ouvrir une fenêtre” testée dans cette étude, à l’action plus générale d’ouvrir quelque chose, par exemple une porte.

Plusieurs pistes restent à explorer afin d’améliorer cette nouvelle méthode AE-ProMPs. Tout d’abord, elle peut être améliorée en permettant la prédiction de trajectoires à durée variable, comme cela a été fait pour les ProMPs seules, par exemple dans [D.P]. De plus, il serait pertinent de créer une technique permettant de régler automatiquement la dimension de l’espace latent de l’auto-encodeur. Finalement, d’autres variantes d’AEs pourraient être combinées avec les ProMPs. Dans ce cas, des tests seront nécessaires afin d’évaluer leurs avantages par rapport à la méthode AE-ProMP. Notons cependant qu’il s’agit ici de l’AE le plus simple et puisque celui-ci semble déjà suffisamment précis pour reproduire les trajectoires, les autres variantes d’AEs risquent de ne pas être pertinentes.

De même, le logiciel que nous avons développé à l’aide de cette nouvelle méthode peut être amélioré de différentes manières.

Premièrement, le fait que les ProMPs soient codées sous Matlab, que l’auto-encodeur soit codé en Python et que ces deux composantes communiquent entre-elles par réseau à l’aide de YARP implique que le temps pour prédire la poursuite du mouvement est ralenti. Il serait alors intéressant de coder ces deux composantes en C++, afin d’accélérer les temps de calcul et d’éviter d’avoir à faire communiquer ces deux parties par réseaux en l’intégrant dans un logiciel unique.

Deuxièmement, dans cette étude, l’interaction entre l’humain et le robot est artificielle et nécessite que l’utilisateur porte une combinaison spécifique. Afin de rendre l’interaction avec iCub plus naturelle, nous chercherons dans de futures études à permettre au robot de détecter la posture du corps-complet des personnes avec qui il interagit. Pour cela, il utilisera ses propres yeux et des algorithmes qui détectent la zone de focalisation du regard des individus.

Troisièmement, dans cette application, nous nous sommes limités à la prédiction des mouvements corps complet de l’utilisateur, sans nous intéresser à comment le robot doit réagir en fonction ces prédictions. Ainsi, pour augmenter les fonctionnalités du robot, il faudrait lui apprendre à exécuter des mouvements et des actions en rapport avec les mouvements de l’utilisateur, comme cela est proposé pour des mouvements de bras robotique dans [77]. Cela pourrait par exemple permettre au robot de tendre son bras afin de soutenir une personne lorsque le robot infère que celle-ci commence à se lever d’une chaise et qu’elle lève la main vers le robot pour lui demander de l’aide. De plus, pour améliorer cette aide, le robot pourrait prédire à partir de son début de mouvement, comment la personne a envie de se lever et, en fonction d’où elle déplace son bras, il pourrait se positionner afin d’améliorer le soutien de la personne.

Les codes sources utilisés dans cette étude sont disponibles dans [soft.WB.Pred].

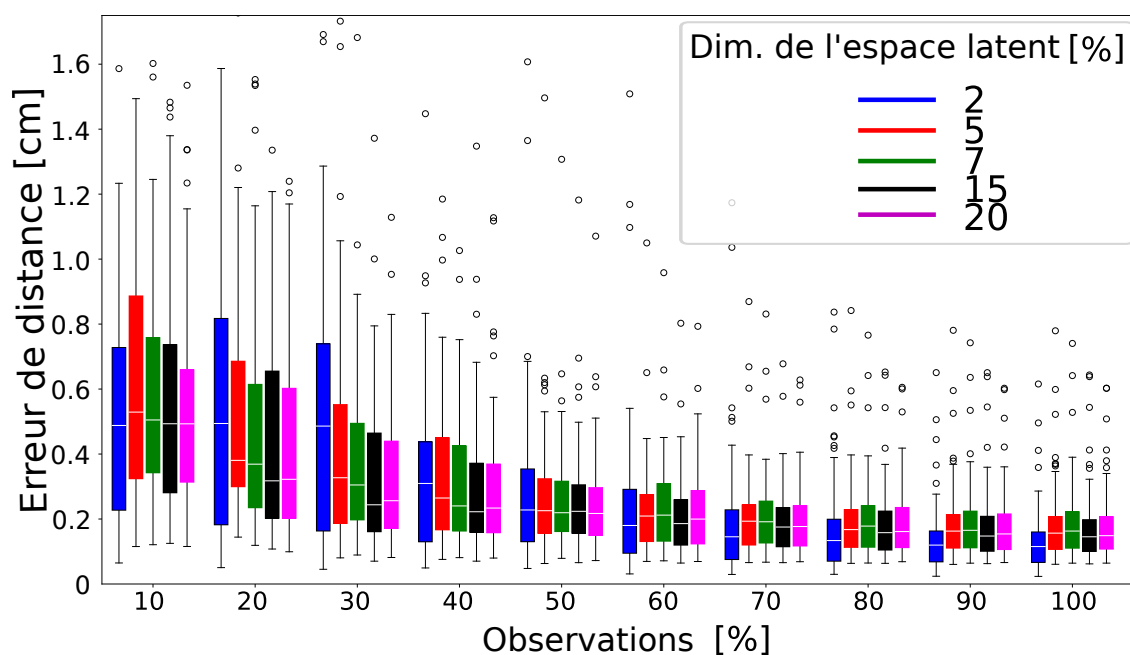
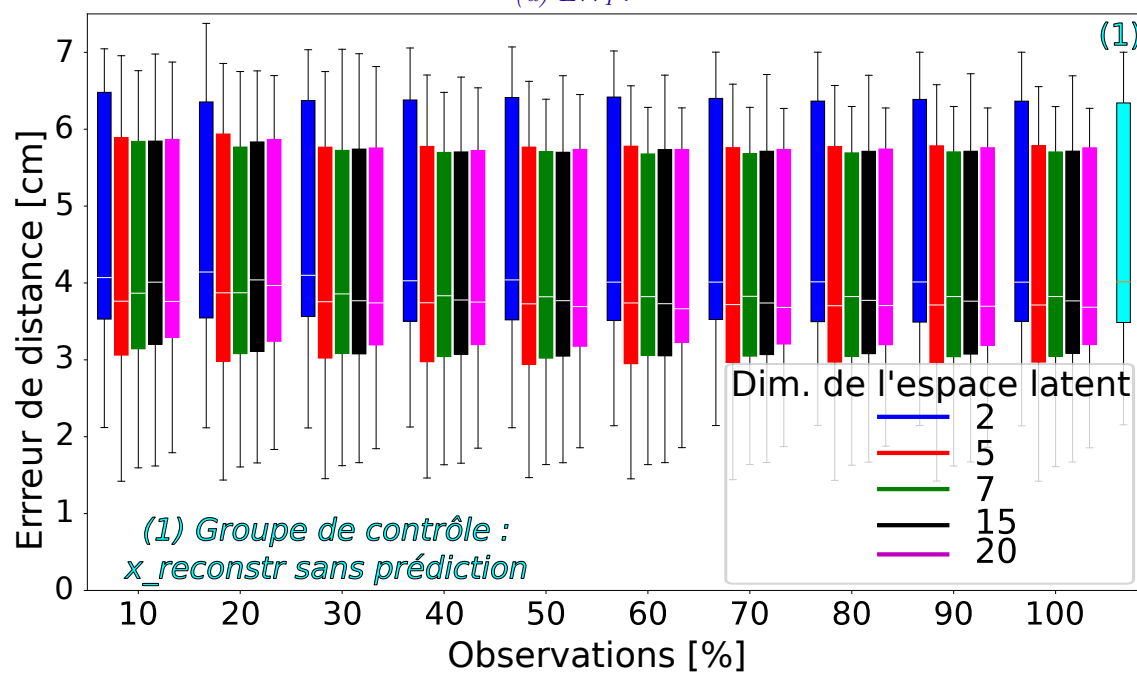
(a) Err_p .(b) Err_{AE} (en cyan) et Err_{AE+P} (autres couleurs).

FIGURE 6.10 – *VTSFE-ProMP* – Ensemble des statistiques - Distances d'erreurs lors des 70 tests effectués à l'aide de la méthode leave-one-out cross-validation, en fonction du nombre d'observations de la trajectoire et de la dimension de l'espace latent de l'auto-encodeur.

Conclusions

7.1 Discussion

Dans ce chapitre, nous rappelons les grandes lignes des discussions et des réflexions émises au cours de ce mémoire.

Ce travail se situe dans le domaine de l'intelligence artificielle appliqué à la cobotique, domaine dans lequel les robots collaborent avec les humains pour les assister dans des tâches souvent répétitives et pénibles. Cette thèse se focalise sur le décodage des interactions non verbales humain-robot dans le but de prédire l'intention de l'utilisateur à partir des gestes qu'il produit.

Cette thèse concerne principalement la modélisation de mouvements humains, afin de permettre à des robots d'interpréter et de générer des mouvements, ou même de continuer un mouvement amorcé par l'utilisateur, à l'aide de capacités prédictives. De plus, il s'agit de généraliser cette modélisation à n'importe quel type de mouvements, que ce soit des mouvements gestuels, de tête, posturaux, ou encore des mouvements composés de différentes informations complémentaires, telles que les forces exercées sur le robot. Nous avons aussi abordé la notion de compression de données, afin de garantir une prédiction de trajectoires à grande dimension en temps réel. En outre, cette compression de données permet de généraliser les mouvements. En effet, lors de l'apprentissage de l'espace latent, celui-ci est construit afin de capturer l'information qui se retrouve dans les différentes postures des trajectoires de démonstration, ce qui revient à enregistrer leurs caractéristiques principales, en s'extrayant des autres informations.

La problématique principale de cette thèse est présentée dans la Section 4. Elle consiste à permettre à un robot humanoïde de comprendre et d'exécuter les mouvements attendus par la personne avec laquelle il interagit, par la prédiction de son intention, sans qu'elle n'ait besoin de le programmer. De plus, il s'agit de diminuer au maximum la charge mentale et physique de cette personne, en lui permettant de communiquer son intention au travers de différents indices provenant : d'interactions physiques, visuelles, ou même d'interactions artificielles, via des capteurs portés par l'utilisateur. Afin de se concentrer sur cette problématique, nous limitons la notion de prédiction de l'intention à la compréhension et à la prédiction des mouvements et actions attendus par l'utilisateur.

Première étude : prédiction de mouvements lors d'interactions physiques humain-robot

Cette étude se focalise sur l'interaction physique humain-robot, où un utilisateur guide manuellement le bras robotique, afin de lui transmettre son intention. Dans une première phase, l'utilisateur effectue des mouvements de démonstration en guidant le bras du robot pour effectuer le

geste complet. Cela permet au robot d'apprendre une distribution de ces mouvements. L'utilisateur répète ce processus pour chaque geste qu'il souhaite apprendre au robot. À la suite de cet apprentissage, le robot peut générer différents mouvements correspondant à ce geste. De plus, l'utilisateur peut amorcer un mouvement avec le bras du robot, afin que celui-ci le termine et exécute l'action correspondante.

Pour cela, nous avons développé un logiciel qui utilise la méthode de Primitives de Mouvements Probabilistes (ProMPs [171]) dans le but d'apprendre des distributions de trajectoires. Nous avons alors adapté cette méthode afin de modéliser l'ensemble des informations qui peuvent être utiles au robot pour exécuter correctement les mouvements voulus. Ces informations concernent la position cartésienne de sa main, ainsi que les forces exercées sur le bras du robot.

De plus, nous avons développé différentes techniques (*c.f.*, Section 3.3.4) qui permettent au robot d'inférer correctement la vitesse des mouvements initiés et d'en déduire la durée du mouvement complet. Les résultats ont alors montré une nette amélioration de la qualité des prédictions par rapport à une approche sans estimation de la vitesse.

Par ailleurs, l'inclusion de l'information des forces dans la modélisation des trajectoires offre différents avantages :

- L'inférence de la vitesse du mouvement est améliorée. Pour expliquer ce résultat, notons que l'accélération et la vitesse d'un corps sont proportionnelles aux forces exercées sur celui-ci.
- Cette modélisation peut permettre à l'utilisateur d'exercer des forces sur le bras robotique lorsque celui-ci exécute son mouvement, afin de lui faire comprendre qu'il souhaite "reprendre la main". Le robot adapte alors sa trajectoire en interprétant ces forces, ou apprend une nouvelle primitive.
- Le robot peut adapter sa trajectoire en fonction du poids des objets qu'il déplace ou encore interpréter les forces supplémentaires qu'il subit, comme des obstacles à éviter.

Nous n'avons cependant pas testé ces scénarios afin de privilégier des problématiques plus poussées techniquement et théoriquement : la prédiction multimodale (*c.f.*, Chapitre 5) et la prédiction de trajectoires à grande dimension (*c.f.*, Chapitre 6).

Le dernier apport de cette étude est la création d'un logiciel composé de deux codes. Le premier code, composé de programmes Matlab, est une boîte à outils qui permet de déployer simplement des ProMPs. Les types de données en entrée sont paramétrables. L'utilisateur peut alors modéliser, reconnaître et générer des trajectoires ainsi que prédire la fin d'une trajectoire partielle et sa durée.

Le second code, composé de programmes C, permet d'intégrer cette boîte à outils dans les modules du robot iCub afin qu'il puisse apprendre à générer, à reconnaître ainsi qu'à prédire des mouvements au niveau de son bras.

Ces deux parties communiquent entre elles grâce à YARP (*c.f.*, Section 2.2). Un tutoriel expliquant comment utiliser ce logiciel est disponible dans l'annexe A. Le papier scientifique [D.P] concerne cette première étude, où un effort particulier a été fourni afin que le lecteur puisse comprendre la méthode ProMP.

Cette première étude comporte cependant différentes limitations, présentées dans la Section 4.4. Nous présentons ici celles qui ont inspiré les études suivantes.

Premièrement, afin d'améliorer la qualité de l'interaction humain-robot, l'utilisateur devrait pouvoir guider le robot sans interaction physique avec celui-ci, pour différentes raisons. Tout d'abord, nous avons montré dans l'Étude [M.E] que certains utilisateurs n'aiment pas toucher les robots. De plus, un guidage partiel ne permet pas au robot de désambiguïser des trajectoires

qui débutent de la même manière. Enfin, guider physiquement le robot peut s'avérer difficile, voire impossible, dans certaines situations soit parce que le robot est trop éloigné, soit qu'il est trop lourd à guider, ou que sa structure est trop complexe pour être manipulé (bas du corps d'un humanoïde).

Une seconde limitation concerne le scénario retenu dans cette première étude, qui se limite au mouvement de la main robotique. Pour pouvoir étendre ce scénario à des mouvements corps complet, cela nécessite de modéliser des mouvements plus complexes, composés de beaucoup plus de données. Ainsi la dimension des trajectoires que devra apprendre le robot augmentera fortement. Nous supposons alors que la méthode utilisée serait trop longue en calcul, ce que nous avons vérifié en pratique au début de la troisième étude.

Deuxième étude : prédiction de mouvements lors d'interactions multimodales entre l'humain et le robot

Dans cette étude, le robot modélise les trajectoires de manière multimodale, par le biais de capteurs proprioceptifs (capteurs de forces et moments) et visuels (caméras). Le scénario proposé permet au robot d'apprendre par démonstration des trajectoires à partir d'un guidage visuel (correspondant à des mouvements faciaux) et d'un guidage manuel (comme lors de l'étude précédente) simultanés. Lors de cet apprentissage, une première personne effectue les mouvements avec le bras robotique, pendant que la seconde, positionnée dans le champ de vision du robot, suit le mouvement avec sa tête, en exagérant le mouvement. Le robot modélise alors ces trajectoires à partir des mesures de la position de sa main et de l'orientation de la tête de la personne située dans son champ de vision. Les utilisateurs peuvent alors choisir de guider le robot par un simple geste de tête et/ou en amorçant un mouvement en guidant manuellement le bras robotique.

Cette étude a donc permis au robot d'interpréter et de prédire les mouvements qu'il doit exécuter, à l'aide d'indices visuels et proprioceptifs. Les résultats ont alors montré les différents avantages de cette modélisation multimodale de trajectoires, dont les principaux avantages sont :

- Le guidage visuel permet de décrire sans contact et de façon grossière le mouvement attendu, mais surtout de désambiguïser des mouvements similaires. Les résultats montrent effectivement que la reconnaissance du type de trajectoire est améliorée avec cette modalité.
- Le guidage physique permet au robot d'affiner sa prédiction, en adaptant précisément la primitive de mouvement reconnue au début de trajectoire initié le partenaire. Les résultats montrent effectivement qu'avec un guidage physique, le mouvement prédit est plus proche du mouvement attendu.
- Le guidage multimodal combine ces avantages. En effet, nous obtenons les meilleurs résultats lorsque le robot utilise les mesures visuelles afin de reconnaître le mouvement puis les mesures physiques, afin d'affiner la prédiction de la fin du mouvement.
- La modélisation multimodale permet aux utilisateurs de choisir leur façon d'interagir avec le robot.

À la fin de cette étude, nous avons proposé différentes améliorations dans la Section 5.6. L'une d'entre elles consiste en l'amélioration de la reconnaissance des mouvements de l'utilisateur en utilisant un logiciel qui, au lieu de se limiter à l'orientation faciale de l'utilisateur, mesure la posture de son corps complet. Pour cela, nous proposons d'utiliser la combinaison Xsens, puisque celle-ci fournit les positions cartésiennes en 3D des membres de l'utilisateur (*c.f.*, Section 2.4). Cela évite notamment l'utilisation des capteurs visuels du robot qui restent assez imprécis. De plus, différentes difficultés restent à surmonter avant de pouvoir permettre au robot d'interagir naturellement avec les humains. Parmi ces points, les questionnements qui ont le plus orienté

notre recherche sont : *Comment évoluent les temps de calcul de notre application, lorsque la dimension des données des trajectoires augmente ? Est-ce que notre logiciel permet une prédiction en temps réel mou de trajectoires à grande dimension ?*

Nous traitons ces points dans la dernière étude de ce mémoire. Celle-ci se focalise sur la prédiction en temps réel mou des trajectoires à grande dimension.

Troisième étude : prédiction de mouvements corps complet

Les deux premières études ont donc permis au robot d’interagir avec l’utilisateur et d’effectuer les gestes que celui-ci souhaite qu’il exécute, à partir de différentes modalités. Cependant, il ne s’agit que de simples gestes de bras. Nous souhaitons maintenant aller plus loin, en permettant au robot d’interagir avec les personnes à l’aide de son corps complet. Pour cela, celui-ci doit tout d’abord comprendre et interpréter les mouvements corps complet des personnes avec qui il interagit.

Nous nous intéressons alors à la prédiction de la poursuite de mouvements corps complet d’individus. Cette étude comporte une difficulté majeure : la prédiction de trajectoires corps complet à grande dimension ralentit les temps de calculs, comme cela est présenté dans la Section 6.4.1. Or, pour que le robot puisse interagir avec les personnes, cela nécessite que ses prédictions se fassent en temps réel mou.

Nous proposons alors une nouvelle technique de réduction de la dimension de données, pour l’apprentissage de primitives probabilistes du mouvement. Les ProMPs permettent de prédire la poursuite de trajectoires, à partir d’un ensemble de postures compressées dans un espace latent à faible dimension. Cet espace latent est soit obtenu par un simple auto-encodage (AE) de la posture humaine, soit par la méthode VTSFE, qui renforce la continuité des postures compressées dans l’espace latent. Dans tous les cas, la construction de l’espace latent permet de conserver les caractéristiques principales des postures. Ainsi, nous avons créé deux méthodes, nommées AE-ProMP et VTSFE-ProMP.

L’étude se porte alors sur la prédiction de mouvements humains en se basant sur un système de capture de mouvement, qui mesure la position 3D de 36 segments du modèle cinématique de l’utilisateur. Afin de comparer nos différentes méthodes, appelées ProMPs-seul, AE-ProMPs et VTSFE-ProMPs, nous avons étudié 7 mouvements humains. Par rapport à l’approche des ProMPs qui fonctionne sur l’espace d’entrée originale à haute dimension, notre approche est plus efficace d’un point de vue computationnel et permet d’effectuer des prédictions en temps réel mou.

Les résultats de cette étude ont montré que :

- La méthode ProMPs-seul offre de bonnes capacités de prédiction de mouvements corps complet, mais ses calculs sont trop longs pour permettre au robot de prédire le futur de mouvements de grande dimension en respectant la contrainte temps réel mou.
- La méthode AE-ProMPs permet au robot de prédire correctement la poursuite de trajectoires de grande dimension, et ce, en respectant la contrainte temps réel mou.
- La méthode VTSFE-ProMPs a les mêmes avantages que la méthode AE-ProMPs et permet d’encoder les trajectoires dans un espace latent dynamiquement plus cohérent. Cependant, la décompression des postures est encore imprécise et doit être améliorée pour être utilisable. Pour cela, nous avons émis différentes propositions dans la Section 6.5.

Finalement, la Section 6.5 présente les différentes extensions que nous pouvons apporter à cette étude, telles qu’optimiser et rendre automatique le choix de la dimension de l’espace latent ou encore étudier d’autres méthodes de compressions des données.

7.2 Perspectives

Dans la Section 2, nous imaginions comment agirait un robot idéal. Ce robot devrait pouvoir collaborer avec des personnes et se comporter comme un humain. Pour cela, il devrait comprendre l’intention des personnes avec qui il interagit, d’agir en conséquence et même de pouvoir prendre des initiatives par lui-même, dans l’intérêt de ces personnes.

Dans cette thèse, nous nous sommes focalisés sur la partie prédiction de l’intention, en permettant au robot de comprendre l’intention d’un utilisateur, lorsque celui-ci effectue un geste de tête ou initie un mouvement avec le bras du robot. Le robot est alors capable d’effectuer le geste voulu, limité à la partie supérieure de son corps. La dernière étude a en plus permis à notre robot de prédire la poursuite de mouvement corps complet de l’utilisateur, à l’aide d’une interaction artificielle (le robot connaît la posture d’un utilisateur, par le biais de capteurs intégrés dans une combinaison portée par celui-ci) et ce, en temps réel mou.

Pour continuer les travaux de cette thèse, la prochaine étape est alors de permettre au robot d’utiliser cette prédiction corps complet, afin d’effectuer les mouvements attendus. Cela nécessitera qu’il apprenne à effectuer des mouvements en corrélation avec ceux de l’utilisateur, comme Ewerton *et al.* l’ont étudié [77] pour des gestes de bras. Cela nécessitera qu’il apprenne à exécuter des mouvements avec son corps complet. Pour ce faire, l’utilisateur ne pouvant pas guider manuellement tous les membres du robot, nous utiliserons un guidage par téléopération, comme cela est proposé par Penco *et al.* dans l’Étude [179].

Le robot pourrait aussi utiliser les informations contextuelles des trajectoires pour améliorer ses capacités prédictives. Par exemple, il pourrait détecter les différents objets présents dans son espace d’action. Il augmenterait la probabilité que ses mouvements se dirigent vers ce but, ou il pourrait recentrer les trajectoires qu’il prédit vers l’objet ou le lieu le plus probablement visé.

Notons également que si la trajectoire guidée par l’utilisateur du robot ne correspond pas à l’une de celles apprises, alors le robot identifie la ProMP la plus proche de la trajectoire initiée. Il essaie alors d’adapter la ProMP reconnue afin d’achever le mouvement amorcé. De ce fait, le robot pourrait effectuer des mouvements non souhaités par l’utilisateur, mouvements qui peuvent être dangereux pour lui et pour son environnement. Une solution serait de définir une “enveloppe probabiliste” autour de chaque ProMP. Ainsi, on considérera qu’une trajectoire correspond à une certaine ProMP si elle est incluse dans son enveloppe, c’est-à-dire si sa probabilité d’appartenir à la distribution est supérieure à un certain seuil. Ce seuil pourra être fixé de différentes manières. Par exemple, il pourra être défini lors de la phase d’apprentissage de la ProMP, en faisant en sorte que chaque trajectoire de démonstration soit dans l’enveloppe probabiliste de sa propre ProMP, tout en cherchant à minimiser cette enveloppe. Un autre exemple serait de la fixer au 97.5^e percentile, correspondant à un indice de confiance de 95% que la trajectoire appartienne bien à la ProMP.

Jusqu’ici, l’interaction humain-robot n’a eu lieu que dans un sens, puisque seul l’humain communique avec le robot. C’est pourquoi il faudra par la suite permettre au robot de communiquer avec les humains, afin d’obtenir une interaction naturelle. Par exemple, inspiré de l’étude de Yamazaki *et al.* [238], le robot pourra apprendre à déplacer sa tête en direction du but, en amont de ses gestes.

Ensuite, les capacités du robot devront être améliorées afin que le robot *planifie* lui-même des tâches à effectuer, *en anticipant les intentions futures de l'utilisateur*. Pour cela, le robot pourrait apprendre une séquence de mouvements/d'actions nécessaires à la réalisation d'une tâche à long terme. Puis, il inférerait quelles sont les actions qu'il peut effectuer afin d'aider l'utilisateur, sans entrer en conflit avec ses mouvements futurs, mouvements que le robot aura prédits.

Cette thèse ouvre ainsi des perspectives de résolution du problème de la prédiction de l'intention et de l'exécution des mouvements appropriés.

A

Tutoriel d'utilisation du logiciel

A.1 Présentation du logiciel de prédiction de mouvements

Ce logiciel permet de modéliser des trajectoires et d'interpréter des mouvements partiels dans le but d'en prédire leur futur. La Section 2.5.1 en avait présenté l'organisation générale. La Section courante présente comment est configuré le robot simulé, puis comment les données sont récupérées.

A.1.1 Configurations du robot simulé

Afin de tester nos applications en simulation, nous avons créé un prototype sur *Gazebo*, où le robot doit atteindre différentes positions but, avec l'aide de son partenaire humain. Afin que celui-ci puisse interagir physiquement avec le robot simulé, le dispositif haptique Geomagic touch 2.3 est utilisé.

De manière plus détaillée, cette simulation est effectuée à l'aide :

- du robot iCub simulé dans *Gazebo* 2.2, comprenant l'information de la dynamique du robot fournie par l'application *iDyn* [117] avec ses applications (*par ex.*, *wholeBodyDynamicsTree*) et l'information sur les positions cartésiennes du robot fournie par l'application *iKinCartesianController* ;
- du dispositif Geomagic Touch (nommé par la suite Geomagic), qui peut être installé suivant les instructions du *README* de notre logiciel [soft.Pred.Mult]. Ces instructions permettent à la fois d'installer le kit de développement et les pilotes de ce dispositif, ainsi que d'informer comment installer ces pilotes sur *YARP* ;
- un module C++ (*c.f.*, le dossier "CppProgram" de notre logiciel) qui renvoie les commandes provenant du Geomagic à l'iCub simulé et qui permet d'enregistrer des trajectoires effectuées par le robot simulé dans un fichier. Un tutoriel permettant de comprendre comment utiliser ce module est fourni avec le logiciel.

Les connexions entre les différents modules sont représentées dans la Figure 2.6. À noter que pour la simulation, le module *Matlab* présenté sur cette figure n'est pas utilisé. La pointe du stylo du Geomagic est virtuellement attachée à la main du robot :

$$x_{geo} \rightarrow x_{icub_hand}$$

Quand l'opérateur bouge le Geomagic, la position de la pointe du stylo du Geomagic x_{geo} est redimensionnée dans l'environnement de l'iCub simulé en tant que x_{icub_hand} et le contrôleur cartésien du robot est utilisé afin de déplacer la main du robot par rapport soit à une position de

départ fixé dans la simulation (l'utilisateur doit alors positionner le Geomagic afin d'atteindre cette position), soit à la position prise au départ par l'utilisateur lorsqu'il appuie sur l'un des boutons du Geomagic :

$$x_{icub_hand} = hapticDriverMapping(x_0 + x_{geo}).$$

Où *hapticDriverMapping* correspond à la transformation appliquée par le dispositif haptique, qui associe la fenêtre de référence du Geomagic à la fenêtre de référence de l'iCub. Par défaut, dans cette application, aucune force n'est renvoyée à l'opérateur (c'est-à-dire, celui qui utilise le dispositif haptique), puisqu'il s'agit de simuler le mode de contrôle à couple nul de l'iCub réel (il s'agit d'un mode de contrôle où le robot n'oppose aucune force de résistance, permettant ainsi au partenaire humain de le guider). Une orientation par défaut est définie dans cette application.

A.1.2 Acquisition de données



FIGURE A.1 – Présentation des boutons du Geomagic Touch.

Le bouton sombre du Geomagic (*c.f.*, Figure A.1) est utilisé afin de commencer et d'arrêter l'enregistrement de chaque trajectoire. L'opérateur doit alors appuyer et maintenir ce bouton durant la totalité du mouvement, puis le relâcher lorsque la trajectoire est finie. Celle-ci est alors sauvegardée dans un fichier nommé "*recordX.txt*", où *X* s'incrémente pour chaque nouvelle trajectoire. Le fichier est structuré de la manière suivante :

```

1 #time xgeo ygeo zgeo fx fy fz mx my mz x_icub_hand y_icub_hand z_icub_hand
2 5.96046e-06 -0.0510954 -0.0127809 -0.0522504 0.284382 -0.0659538 -0.0239582 ...
   -0.0162418 -0.0290078 -0.0607215 -0.248905 -0.0872191 0.0477496$

```

Afin de rejouer la *i*-ième trajectoire enregistrée, l'utilisateur peut cliquer sur le bouton gris du Geomagic puis inscrire dans le terminal le numéro *i*.

La vidéo [Tutoriel] représente cette manipulation, où l'opérateur guide le bras de l'iCub simulé à l'aide du dispositif haptique. Le graphique de la Figure 4.3 représente quelques trajectoires enregistrées à l'aide du Geomagic, lors de la levée du bras gauche du robot *iCub*.

Les trajectoires de démonstrations et leurs forces correspondantes peuvent être enregistrées directement, en accédant à l'interface cartésienne du robot et au module *wholeBodyDynamics-Tree*.⁴⁵

45. Dans notre exemple, les informations des forces et des couples ne sont pas utilisées, car elles sont trop bruitées. Cependant, nous fournissons le code permettant la récupération et l'utilisation de ces informations, lorsque le Geomagic.

Dans notre projet Github, nous fournissons l'ensemble des données des trajectoires que nous avons enregistrées, pour les utilisateurs qui souhaitent tester le code avec ces trajectoires. Deux ensembles de données sont disponibles dans le sous-dossier “MatlabProgram/Data” de notre logiciel [soft.Pred.Mult]⁴⁶.

A.2 Exemple fourni par le logiciel : apprentissage d'une primitive à une dimension

Cette section présente comment utiliser le logiciel afin d'apprendre une ProMP à une dimension. Cet exemple nécessite uniquement l'utilisation du dossier *MatlabProgram*, composé de :

- Un sous dossier nommé “Data”, où un ensemble de trajectoires permet d'apprendre des ProMPs. Ces trajectoires sont enregistrées dans des fichiers textes, contenant les informations suivantes :
 - **paramètres d'entrée** : # e₁ # e₂ [...]
 - **paramètre d'entrée avec information temporelle** : # temps # e₁ # e₂ [...]
 - programme d'enregistrement de trajectoire *recordTrajectories.cpp* : *c.f.*, Section A.1.2 pour plus d'information.
- Un sous dossier nommé “used_functions”. Celui-ci contient toutes les fonctions utilisées pour récupérer les trajectoires, de calculer les ProMPs, de prédire les trajectoires, ainsi que de représenter les résultats dans des figures. L'utilisation de ce logiciel ne nécessite pas la compréhension de l'ensemble de ces fonctions.

Au début de ces fonctions, des lignes expliquent leur fonctionnement et précisent à quoi correspondent les paramètres d'entrée et de sortie.
- Des scripts Matlab dénommés “demo_*.m”. Il s'agit d'exemples basiques afin de comprendre comment utiliser ce logiciel.

Le script `demo_plot1DOF.m` permet de calculer une ProMP et de continuer un mouvement amorcé. La ProMP est calculée à partir d'un ensemble de trajectoires récupéré d'un fichier “.mat”, appelé *traj1_1DOF.mat*. Dans ce script, les variables sont d'abord définies afin de correspondre à l'ensemble des trajectoires :

46. Un script *Matlab* permet d'apprendre les ProMPs à partir d'ensemble de trajectoires enregistrées comme celles présentées ci-dessus. Il s'agit du script `demo_plotProMPs.m`.

Assigination	Commentaire
DataPath = 'Datatraj1_1DOF.mat';	Peut correspondre à un fichier “.mat” ou “.txt”. Dans la démonstration courante, vous pouvez aussi écrire DataPath = 'Data/traj1' si vous souhaitez utiliser les fichiers textes de cet ensemble de données.
typeRecover= '.mat'	ou .txt, selon le choix du type de fichier de données.
inputName = 'z[m]';	Étiquette des données d'entrée des trajectoires. Ici, z représente la coordonnée cartésienne d'axe z .
s_ref=100;	Nombre d'échantillons utilisé en tant que référence afin de ré-échantillonner l'ensemble des trajectoires pour qu'elles aient la même longueur temporelle.
nbInput = 1;	Dimension du vecteur paramètre contenant les mesures des trajectoires.
M = 5;	Nombre des fonctions de base radiale par donnée d'entrée.
expNoise = 0.00001;	Estimation de l'erreur des mesures due aux capteurs.
percentData = 20;	Pourcentage des données observées avant la prédiction.

Dans ces variables, il y a :

- DataPath est le chemin d'accès menant aux données enregistrées. Si les données sont enregistrées dans des fichiers textes, cette variable contient le nom du dossier où les fichiers textes sont enregistrés. Ces fichiers textes sont appelés “record X .txt”, avec $X \in [0 : n - 1]$ si ils sont n trajectoires. Un dossier est utilisé pour apprendre une ProMP. Si les données sont déjà contenues dans un fichier “.mat”, le chemin d'accès total doit être inscrit, avec l'extension. Les données dans le fichier “.mat” coïncident avec les données de sortie de la fonction Matlab loadTrajectory.
- nbInput= D correspond à la dimension du vecteur d'entrée ξ_t .
- expNoise = Σ_ξ^o correspond au bruit estimé des mesures de la trajectoire initiée. Plus cette variable est petite, plus la modification de la ProMP va être importante, afin de correspondre plus précisément aux observations.

Le script va maintenant être détaillé. Afin de récupérer les données enregistrées dans le fichier “.txt”, la fonction suivante est appelée :

```
t{1} = loadTrajectory(PATH, nameT, varargin)
```

Ces paramètres d'entrée précisent le chemin d'accès des données enregistrées, ainsi que l'étiquette représentant le type de trajectoire. D'autres informations peuvent être ajoutées, à l'aide de la variable varargin (*c.f.*, informations commentées en tête de cette fonction). Les paramètres de sortie de cette fonction correspondent à un objet contenant toutes les informations des trajectoires de démonstrations. Ces paramètres sont le nombre de trajectoires : nbTraj; le nombre d'itérations représentant les trajectoires : realTime; l'ensemble des vecteurs (et matrices) représentant les : y (et yMat); etc.. Ainsi, t{1}.y{i} contient la i^e trajectoire.

La fonction Matlab drawRecoverData(t{1}, inputName, 'namFig', nFig, varargin) permet d'afficher sur une figure (numérotée nFig) les mesures des trajectoires récupérées. Un exemple est montré dans la Figure 3.1, sur la gauche. Les différentes durées de trajectoires y sont visibles : en moyenne, cette durée est de 1.17 ± 0.42 secondes.

De cet ensemble de trajectoires de démonstration t{1}, un sous ensemble permet d'effectuer l'apprentissage des ProMPs et un autre afin d'effectuer des tests sur ces ProMPs. Pour cela, la fonction suivante est appelée :

```
[train, test] = partitionTrajectory(t{1}, partitionType, percentData, s_ref)
```

si `partitionType=1`, seulement une trajectoire est utilisée pour les tests, et si `partitionType>1`, cette variable correspond alors au pourcentage de trajectoires qui sera inclus dans le sous-ensemble utilisé pour l'apprentissage.

La ProMP est alors calculée à partir de ce sous-ensemble de trajectoires d'apprentissage, à l'aide de la fonction : `promp = computeDistribution(train, M, s_ref, c, h)`

Le paramètre de sortie `promp` est un objet qui contient l'ensemble des informations de la ProMP. Les trois premiers paramètres d'entrée ont déjà été présentés lors de l'explication des fonctions précédentes : `train` est l'ensemble des trajectoires d'apprentissage ; `M` est le nombre de *RBFs* ; `s_ref` est le nombre d'échantillons utilisés pour ré-échantillonner toutes les trajectoires. Les deux derniers paramètres `c` et `h` donnent la forme des *RBFs* utilisés dans la modélisation des ProMPs : $c \in \mathbb{R}^M$ est le centre des Gaussiennes et $h \in \mathbb{R}$ leur variance.

Afin de visualiser cette ProMP, comme montre dans la Figure 3.1, la fonction suivante est appelée : `drawDistribution(promp, inputName, s_ref)` Afin de permettre le débogage ainsi que de comprendre comment régler les paramètres des ProMPs, il est intéressant de représenter dans un graphique les fonctions de base radiale en fonction du temps. En effet, choisir le bon nombre de fonctions de base radiale est important puisque, lorsqu'elles ne sont pas assez, les trajectoires ne peuvent pas être bien approximées et lorsqu'elles sont trop nombreuses, cela peut provoquer des problèmes de surapprentissage. Pour représenter ces fonctions dans un graphique, la fonction suivante peut être appelée :

```
drawBasisFunction(promp.PHI, M)
```

où `promp.PHI` est un ensemble de *RBFs* évaluées dans la plage temporelle normalisée : $t \in [1 : \bar{s}]$.

La Figure 3.2 représente en haut les fonctions de base radiale avant la normalisation, et en bas la ProMP dont la modélisation se base sur ces fonctions.

Après que la ProMP soit apprise, le robot est capable de reproduire le mouvement en utilisant la moyenne de la distribution. De plus, il peut aussi reconnaître le mouvement qui a été initié dans cette distribution, et prédire comment le finaliser. Dans ce but, à partir de n_o observations initiales d'un mouvement, le robot met à jour la distribution a priori afin qu'elle passe par les mesures observées. Pour cela, en utilisant le conditionnement, il trouve la distribution a posteriori, qui peut être utilisée par le robot afin d'exécuter la continuation du mouvement par lui même.

La première étape pour prédire l'évolution de la trajectoire est d'inférer la durée de cette trajectoire, qui est encodée par le paramètre de modulation du temps $\hat{\alpha}$. Le calcul de cette inférence, détaillé dans la Section 3.3.4, peut être effectué à l'aide de la fonction :

```
[expAlpha, type, x]=inferenceAlpha(promp, test{1}, M, s_ref, c, h, test{1}.nbData, ...
expNoise, typeReco)
```

où `typeReco` est le critère utilisé afin d'estimer le paramètre de modulation du temps ('MO', 'DI' ou 'ML' pour "modélisation", "distance" ou "maximum de vraisemblance"); `expAlpha` = $\hat{\alpha}$ l'estimation du paramètre de modulation ; `type` (noté k dans cette thèse) l'indice de la ProMP à partir de laquelle `expAlpha` a été calculée.

Afin de prédire l'évolution de la trajectoire, on utilise l'Équation 3.6 de la Section 3.3.3. Dans Matlab, ce calcul est effectué au sein de la fonction :

```
infTraj = inference(promp, test{1}, M, s_ref, c, h, test{1}.nbData, expNoise, ...
expAlpha)
```

où `test{1}.nbData` est calculé durant l'étape `partitionTrajectory`. Cette variable représente le nombre n_o d'observations, calculée à partir du pourcentage `percentData` de données de la trajectoire `test` observées par le robot. `infTraj=Ê` correspond à la trajectoire prédite. Finalement, la trajectoire prédite est représentée à l'aide de la fonction

```
drawInference(prompt, inputName, infTraj, test1, s_ref).
```

La qualité de la trajectoire prédite en fonction du nombre d'observations est représentée dans la Figure 3.3.

Cette figure met en avant que, lorsqu'une partie importante de la trajectoire est observée, la prédiction de la continuation du mouvement est plus précise.

Après cette étape de prédiction, nous proposons de mesurer la qualité de ces prédictions. Soit $\Xi^* = [\xi^o(1), \dots, \xi^o(n_o), \xi^*(n_o + 1), \dots, \xi^*(t_f^*)]$ la trajectoire réelle, espérée par l'utilisateur (ou vérité terrain). Afin de mesurer la qualité de la prédiction, plusieurs mesures peuvent être utilisées :

- La vraisemblance d'avoir une trajectoire Ξ^* sachant la distribution a posteriori $p(\hat{\omega})$.
- La distance entre la trajectoire désirée Ξ^* et la trajectoire prédite $\hat{\Xi}$.

Cependant, suivant la mesure `typeReco` utilisée pour l'estimation du paramètre de modulation du temps α (estimation faite à partir des données initiales observées), un écart visible peut avoir lieu entre la trajectoire prédite et la trajectoire réelle, et ce, même lorsque beaucoup de données sont observées. Cet écart est provoqué par l'erreur d'estimation du paramètre de modulation du temps. C'est pourquoi, dans la Section 3.3.4, différentes méthodes seront présentées afin de permettre de prédire la durée de la trajectoire à continuer. Ces méthodes sélectionnent le paramètre $\hat{\alpha}$ le plus probable selon différents critères : distance ; maximum de vraisemblance ; modélisation de la variable α ⁴⁷ ; et la moyenne des paramètres α observés lors de la phase d'apprentissage.

La Figure 3.5 représente les différentes trajectoires prédites après l'observation de $n_o = 40\%$ de la trajectoire désirée, en fonction de la méthode utilisée pour estimer le paramètre de modulation du temps.

A.3 Enregistrement de données avec l'iCub simulé

Le bouton sombre du *Geomagic* est utilisé afin de commencer et d'arrêter l'enregistrement de chaque trajectoire. L'opérateur doit alors cliquer et maintenir ce bouton durant tout son mouvement, puis relâcher celui-ci à la fin de la trajectoire. La trajectoire est sauvegardée dans un fichier nommé *recordX.txt* pour la X^e trajectoire. La structure de ce fichier est présentée dans l'encadré A.1.2.

Afin de rejouer une trajectoire parmi les N enregistrées, l'opérateur peut cliquer sur le bouton gris du *Geomagic* puis entrer le numéro de la trajectoire (c'est-à-dire, que s'il veut rejouer son 4e enregistrement, il écrit le nombre 4) qu'il souhaite rejouer dans le terminal.

Une vidéo représente cette manipulation [Tutoriel], où l'opérateur guide le bras de l'iCub simulé à l'aide du dispositif haptique. Le graphe de la Figure 4.3 représente quelques trajectoires enregistrées à l'aide du *Geomagic*, lors de la levée du bras gauche du robot *iCub*.

Les trajectoires de démonstrations et leurs forces correspondantes peuvent être enregistrées directement, en accédant à l'interface cartésienne du robot et au module *wholeBodyDynamics-Tree*.⁴⁸

Dans notre projet *Github* [soft.Pred.Mult], nous fournissons l'ensemble des données des trajectoires que nous avons enregistré, pour les utilisateurs qui souhaitent tester le code avec ces trajectoires. Deux ensembles de données sont disponibles dans le sous dossier de notre logiciel

47. Cette modélisation se base sur l'hypothèse que le paramètre de modulation du temps peut être estimé à partir de la connaissance de la variation "brute" de la position lors des n_o premières données observées.

48. Dans notre exemple, les informations des forces et des couples ne sont pas utilisées, car elles sont trop bruitées. Cependant, nous fournissons le code permettant de montrer comment les récupérer et comment les utiliser à partir du *Geomagic*.

“*MatlabProgram/Data*” : le premier, appelé “heights”, est composé de trajectoires utilisées afin d’atteindre trois positions buts, dont la hauteur varie ; le second, appelé “FLT”, est composé de trajectoires effectuées par le vrai robot. Ces trajectoires permettent d’atteindre trois types de buts : un but positionné devant le robot, un autre sur sa gauche et le dernier en hauteur.

Un script Matlab permet d’apprendre les ProMPs à partir d’ensemble de trajectoires enregistrées. Il s’agit du script `demo_plotProMPs.m` qui contient les étapes présentées ci-dessous.

Afin de récupérer l’ensemble de données “heights”, le script suivant est appelé :

```
1 t{1} = loadTrajectory('Data/heights/bottom', 'bottom', 'refNb', s_bar, ...
    'nbInput',nbInput, 'Specific', 'FromGeom');
2 t{2} = loadTrajectory('Data/heights/top', 'top', 'refNb', s_bar, ...
    'nbInput',nbInput, 'Specific', 'FromGeom');
3 t{3} = loadTrajectory('Data/heights/middle', 'forward', 'refNb', s_bar, ...
    'nbInput',nbInput, 'Specific', 'FromGeom');
```

La Figure 4.3 représente ces trois groupes de trajectoires de démonstration contenus dans l’ensemble “height”. Dans cet ensemble, 40 trajectoires ont été enregistrées pour représenter chaque primitive de mouvement.

A.4 Tutoriel application sur l’iCub simulé

A.4.1 Apprentissage de trois ProMPs

Tout d’abord, les ProMPs qui correspondent aux trois mouvements observés doivent être apprises. Pour cela, l’ensemble des données est d’abord séparé en deux, la première partie afin d’apprendre les ProMPs et la seconde afin de vérifier que la prédiction se fait correctement (trajectoires tests). Pour ne conserver qu’une trajectoire test sélectionnée de manière aléatoire, la fonction suivante est appelée :

```
1 [train{i},test{i}] = partitionTrajectory(t{i},1,percentData,s_bar);
```

Le second paramètre d’entrée spécifie qu’une seule trajectoire doit être sélectionnée de manière aléatoire afin de tester la ProMP apprise. Maintenant, les trois ProMPs sont apprises avec :

```
1 promp{1} = computeDistribution(train{1}, M, s_bar,c,h);
2 promp{2} = computeDistribution(train{2}, M, s_bar,c,h);
3 promp{3} = computeDistribution(train{3}, M, s_bar,c,h)
```

avec comme paramètres :

- `s_bar=100` : le nombre d’échantillons de référence, noté \bar{s} dans cette thèse.
- `nbInput(1)= 3; nbInput(2)= 6` : la dimension du vecteur paramètre qui contient les états de la trajectoire. Il est composé de la position cartésienne en $3D$ et des forces et couples en $6D$.
- `M(1)= 5; M(2)= 5` : le nombre de *RBFs* utilisé pour représenter chacune des `nbInput` dimensions.
- `c = 1/M; h = 1/(M*M)` : les paramètres des *RBF* (*c.f.*, Équation 3.2).
- `expNoise = 0.00001` : le bruit supposé des données.
- `percentData = 40` : cette variable spécifie la valeur du pourcentage de données de la trajectoire totale que le robot doit mesurer avant de prédire la continuation de la trajectoire.

Ces paramètres peuvent être modifiés par l'utilisateur au début du script Matlab.

La Figure 4.4 présente les trois ProMPs permettant d'atteindre trois positions but réparties à différentes hauteurs. Afin de clarifier ce graphique, seul l'axe z de la position cartésienne est présenté.

```

1 drawRecoverData(t{1}, inputName, 'Specolor','b','namFig',1);
2 drawRecoverData(t{1}, inputName, 'Interval', [4 7 5 8 6 9], ...
   'Specolor','b','namFig',2);
3 drawRecoverData(t{2}, inputName, 'Specolor','r','namFig',1);
4 drawRecoverData(t{2}, inputName, 'Interval', [4 7 5 8 6 9], ...
   'Specolor','r','namFig',2);
5 drawRecoverData(t{3}, inputName, 'Specolor','g','namFig',1);
6 drawRecoverData(t{3}, inputName, 'Interval', [4 7 5 8 6 9], ...
   'Specolor','g','namFig',2);

```

A.4.2 Prédiction du mouvement désiré

Maintenant que les trois ProMPs sont apprises, la prédiction de la fin de la trajectoire peut être effectuée, à partir de n_o observations initiales. Ce nombre est calculé à partir de la variable présentée précédemment `percentData` : $n_o = \lfloor \frac{\text{percentData}}{100} * t_{fi} \rfloor$, où i est l'indice de la trajectoire test.

Afin de préparer la prédiction, le paramètre de modulation du temps de chaque trajectoire est calculé à l'aide d'une modélisation :

```

1 w = computeAlpha(test.nbData,t, nbInput);
2 promp{1}.w_alpha= w{1};
3 promp{2}.w_alpha= w{2};
4 promp{3}.w_alpha= w{3};

```

Cette modélisation s'appuie sur la variation "brute" (c'est-à-dire, la différence entre la valeur au temps $t = n_o$ et celle au temps $t = 0$) de la position cartésienne durant les n_o premières observations. Les calculs effectués par cette modélisation sont expliqués dans la Section 3.3.4.

Maintenant, l'estimation de ce paramètre de modulation du temps est effectuée à l'aide de la fonction `inferenceAlpha` :

```

1 [alphaTraj,type, x] = ...
   inferenceAlpha(promp,test{1},M,s_bar,c,h,test{1}.nbData, expNoise, 'MO');

```

où `alphaTraj` contient l'estimation du paramètre de modulation du temps $\hat{\alpha}$ et la variable `type` correspond à l'indice qui caractérise la ProMP reconnue. Le dernier paramètre de sortie `x` est utilisé dans des buts de débogage.

En utilisant cette estimation du paramètre de modulation du temps, la continuation de la trajectoire initiée est alors prédite avec :

```

1 infTraj = inference(promp, test{1}, M, s_bar, c, h, test{1}.nbData, expNoise, ...
   alphaTraj);

```

Comme présenté dans l'exemple précédent, la qualité de la prédiction du futur de la trajectoire dépend de la précision de l'estimation du paramètre de modulation du temps. Cette estimation est effectuée de différentes manières :

```

1 %En utilisant une modelisation :
2 [alphaTraj,type, x] = ...
   inferenceAlpha(promp,test{1},M,s_bar,c,h,test{1}.nbData, expNoise, 'MO');
3 %En utilisant un critere de distance :
4 [alphaTraj,type, x] = ...
   inferenceAlpha(promp,test{1},M,s_bar,c,h,test{1}.nbData, expNoise, 'DI');
5 %En utilisant le maximum de vraisemblance :
6 [alphaTraj,type, x] = ...
   inferenceAlpha(promp,test{1},M,s_bar,c,h,test{1}.nbData, expNoise, 'ML');
7 %En utilisant la moyenne des parametres de modulation du temps calculés lors ...
   de l'apprentissage :
8 alphaTraj = (promp{1}.mu_alpha + promp{2}.mu_alpha + promp{3}.mu_alpha) /3.0;

```

A.5 Application sur l'iCub réel

Dans cette expérience, le robot récupère 10 trajectoires de démonstrations par primitive de mouvement, toutes fournies par un même utilisateur. Les données enregistrées correspondent à la position cartésienne de la main gauche du robot ainsi que les forces et moments qu'elle reçoit lors de ces mouvements.

A.5.1 Trois actions simples, avec informations sur les forces et les couples

Les données sont récupérées à l'aide de la fonction du sous-dossier “used_functions”

`retrieveRealDataWithoutOrientation.m`. Les paramètres de sorties de cette fonction correspondent à trois objets (un par ProMP) qui contiennent toutes les informations requises afin d'apprendre les ProMPs.

Dans cette fonction, les informations des forces et des couples sont filtrées à l'aide d'une fonction Matlab appelée `envelope.m`⁴⁹ : pour chaque trajectoire `traj` et sa sous-matrice $M = F([1 : t])$:

```

1 [envHigh, envLow] = envelope(traj.M);
2 traj.M = (envHigh+envLow)/2;

```

Ces trois objets sont sauvegardés dans '`Data/realIcub.mat`'. Un script Matlab appelé

`demo_plotProMPsIcub.m` permet de récupérer ces données, à l'aide de la fonction `load('Data/realIcub.mat')`. Ce script est organisé de la même manière que ceux présentés dans les Sections A.2 et 4. En lançant ce script, les données récupérées sont d'abord représentées.

Puis, les ProMPs sont apprises et affichées, comme présenté dans la Figure 4.8. Dans cette figure, les distributions sont visiblement superposées :

- le long de toutes les trajectoires en ce qui concerne les informations des couples ;
- lors des 40 premiers pourcentages d'observations des trajectoires en ce qui concerne l'information des positions cartésiennes.

⁴⁹. Les informations de cette fonction peuvent être récupérées ici : <https://fr.mathworks.com/help/signal/ref/envelope.html?requestedDomain=www.mathworks.com>

B

Informations concernant la matrice de fonctions de base radiale

En ce qui concerne le calcul des ProMPs, la Sous-Section 3.3.2 présente le modèle pour un cas simple (c'est-à-dire, pour un instant t seulement, et une entrée monodimensionnelle) par l'équation 3.1, et précise que les fonctions de base radiale (RBFs) sont définies par l'équation 3.2.

Ici, cette annexe cherche à éclaircir comment est formé cet ensemble de fonctions, de ce simple cas (c'est-à-dire, pour un instant t , représentant des données monodimensionnelles) aux cas plus complexes (c'est-à-dire, représentation d'une trajectoire complète $t = [1 : t_f]$, et pour des données d'entrées à plus grosse dimension).

Tout d'abord, pour chaque échantillon monodimensionnel $\xi(t) \in \mathbb{R}$, Φ_t correspond à un vecteur comprenant les M RBFs :

$$\Phi_t = [\psi_1(t), \psi_2(t), \dots, \psi_M(t)] = \frac{[e^{-\frac{(\frac{t}{s}-c_1)^2}{2h}}, \dots, e^{-\frac{(\frac{t}{s}-c_M)^2}{2h}}]}{\sum_{j=1}^M \psi_j(t)} \in \mathbb{R}^{1 \times M}$$

Rappelons que ces fonctions sont pondérées par $\omega \in \mathbb{R}^{1 \times M}$. Avec l'ajout du redimensionnement temporel (*c.f.*, 3.3.4) cela devient :

$$\Phi_{\alpha t} = [\psi_1(\alpha t), \psi_2(\alpha t), \dots, \psi_M(\alpha t)] \in \mathbb{R}^{1 \times M}$$

Lorsque l'on représente la trajectoire complète (c'est-à-dire, on prend tous les échantillons de la trajectoire : $t = [1 : t_f]$, avec en entrée $\Xi = \{\xi(1), \dots, \xi(t_f)\}$), cela devient :

$$\Phi_{\alpha[1:t_f]} = \begin{pmatrix} \psi_1(\alpha) & \psi_2(\alpha) & \dots & \psi_M(\alpha) \\ \dots & \dots & \dots & \dots \\ \psi_1(\alpha t_f) & \psi_2(\alpha t_f) & \dots & \psi_M(\alpha t_f) \end{pmatrix} \in \mathbb{R}^{|t_f| \times M}$$

Notons que la dimension des pondérations ne change pas, puisque ces paramètres sont indépendants du temps.

Et lorsque l'on représente une trajectoire définie par des données multidimensionnelles $\xi(t) \in \mathbb{R}^D$, cela devient une matrice par bloc :

$$\Phi_{\alpha[1:t_f]} = \begin{pmatrix} \psi_1(\alpha) & \dots & \psi_M(\alpha) & & \\ & \ddots & & & \\ \psi_1(\alpha t_f) & \dots & \psi_M(\alpha t_f) & & \\ & & & \ddots & \\ & & & & \psi_1(\alpha) & \dots & \psi_M(\alpha) \\ & & & & & \ddots & \\ & & & & \psi_1(\alpha t_f) & \dots & \psi_M(\alpha t_f) \end{pmatrix} \in \mathbb{R}^{|t_f| \cdot D \times M \cdot D}$$

De même, le nombre de pondérations s'adapte au nombre de fonctions : $\omega \in \mathbb{R}^{D \cdot M}$.

Notons que dans le cas de nos expériences, nous avons tout le temps utiliser la même famille de fonctions : les fonctions de base radiale. Celles-ci permettent de représenter des trajectoires qui varient de manière continue, sans variation brutale. Or, lorsqu'il s'agit de trajectoires variant brutalement, ou encore de trajectoires sinusoïdales, cette ensemble de fonctions doit changer. Par exemple, imaginons que la dernière dimension de la trajectoire présentée ci-dessus change. Imaginons que les données de cette dimension suivent une trajectoire sinusoïdale, alors nous pourrions adapter l'ensemble des fonctions qui permet de modéliser la trajectoire par :

$$\Phi_{\alpha[1:t_f]} = \begin{pmatrix} \psi_1(\alpha) & \dots & \psi_M(\alpha) & & \\ & \ddots & & & \\ \psi_1(\alpha t_f) & \dots & \psi_M(\alpha t_f) & & \\ & & & \ddots & \\ & & & & \Psi_1(\alpha) & \dots & \Psi_M(\alpha) \\ & & & & & \ddots & \\ & & & & \Psi_1(\alpha t_f) & \dots & \Psi_M(\alpha t_f) \end{pmatrix} \in \mathbb{R}^{|t_f| \cdot D \times M \cdot D}$$

et $\Psi(t) = e^{\frac{1}{h} \cos(2\pi(t/\bar{s} - c_1))}$ (fonctions de base Von Mises, comme proposé dans [170]).

C

Informations concernant les formules de prédiction

Dans ces annexes, les formules de prédictions sont développées. Pour cela, les lois gaussiennes marginales et conditionnelles sont utilisées⁵⁰. Étant donnée la distribution gaussienne marginale de x et la distribution gaussienne de y sachant x :

$$p(x) = \mathcal{N}(x|\mu, \Delta^{-1}) \quad p(y|x) = \mathcal{N}(Ax + b, L^{-1}) \quad (\text{C.1})$$

la distribution marginale de y et la distribution conditionnelle de x sachant y sont données par :

$$p(y) = \mathcal{N}(y|A\mu + b, L^{-1} + A\Delta^{-1}A^\top) \quad (\text{C.2})$$

$$p(x|y) = \mathcal{N}(x|\Sigma A^\top L(y - b) + \Delta\mu, \Sigma) \quad (\text{C.3})$$

où

$$\Sigma = (\Delta + A^\top L A)^{-1}$$

À partir de l'ensemble des mouvements observés, les paramètres de la distribution gaussienne marginale sont alors calculés par :

$$p(\omega) \sim \mathcal{N}(\mu_\omega, \Sigma_\omega) \quad (\text{C.4})$$

À partir de la formule $\Xi_t = \Phi_{[1:t_f]} \omega + \epsilon_\Xi$, la distribution gaussienne conditionnelle de Ξ sachant ω est donnée par :

$$p(\Xi|\omega) = \mathcal{N}(\Xi|\Phi_{[1:t_f]} \omega, \Sigma_\Xi) \quad (\text{C.5})$$

Puis, à l'aide de l'Équation C.2, la distribution a priori de la ProMP est obtenue :

$$p(\Xi) = \mathcal{N}(\Xi|\Phi_{[1:t_f]} \mu_\omega, \Sigma_\Xi + \Phi_{[1:t_f]} \Sigma_\omega \Phi_{[1:t_f]}^\top) \quad (\text{C.6})$$

Soit $\Xi^o = [\xi^o(1), \dots, \xi^o(n_o)]$, les n_o premières observations de la trajectoire à prédire. Soit $\hat{\Xi} = [\hat{\xi}^o(1), \dots, \hat{\xi}^o(n_o), \hat{\xi}(n_o + 1), \dots, \hat{\xi}(t_{i_f})]$ la trajectoire escomptée. La distribution a posteriori de la ProMP peut alors être calculée, en utilisant l'équation gaussienne conditionnelle C.3 :

$$p(\omega|\Xi^o) = \mathcal{N}(\omega|\mu_\omega + K(\Xi^o - \Phi_{[1:n_o]} \mu_\omega), \Sigma_\omega - K \Phi_{[1:n_o]} \Sigma_\omega) \quad (\text{C.7})$$

50. Provenant du livre [18]

$$\text{with } K = \Sigma_{\omega} \Phi_{[1:n_o]}^{\top} (\Sigma_{\Xi} + \Phi_{[1:n_o]} \Sigma_{\omega} \Phi_{[1:n_o]}^{\top})^{-1} \quad (\text{C.8})$$

Ainsi, la distribution a posteriori de la ProMP est obtenue : $p(\boldsymbol{\omega} | \Xi^o) = \mathcal{N}(\boldsymbol{\omega} | \hat{\boldsymbol{\mu}}_{\omega}, \hat{\Sigma}_{\omega})$ avec :

$$\begin{cases} \hat{\boldsymbol{\mu}}_{\omega} &= \boldsymbol{\mu}_{\omega} + K(\Xi^o - \Phi_{[1:n_o]} \boldsymbol{\mu}_{\omega}) \\ \hat{\Sigma}_{\omega} &= \Sigma_{\omega} - K(\Phi_{[1:n_o]} \Sigma_{\omega}) \\ K &= \Sigma_{\omega} \Phi_{[1:n_o]}^{\top} (\Sigma_{\xi}^o + \Phi_{[1:n_o]} \Sigma_{\omega} \Phi_{[1:n_o]}^{\top})^{-1} \end{cases} \quad (\text{C.9})$$

Bibliographie

- [1] Rachid Alami, Alin Albu-Schäffer, Antonio Bicchi, Rainer Bischoff, Raja Chatila, Alessandro De Luca, Agostino De Santis, Georges Giralt, Jérémie Guiochet, Gerd Hirzinger, et al. Safe and dependable physical human-robot interaction in anthropic domains : State of the art and challenges. In *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, pages 1–16, 2006.
- [2] Rachid Alami, Raja Chatila, Aurélie Clodic, Sara Fleury, Matthieu Herrb, Vincent Montreuil, and Emrah Akin Sisbot. Towards human-aware cognitive robots. In *The fifth international cognitive robotics workshop (the AAI-06 workshop on cognitive robotics)*, 2006.
- [3] Rachid Alami, Aurélie Clodic, Vincent Montreuil, Emrah Akin Sisbot, and Raja Chatila. Toward human-aware robot task planning. In *AAAI spring symposium : to boldly go where no human-robot team has gone before*, pages 39–46, 2006.
- [4] A Albu-Schaffer, A Bicchi, G Boccadamo, R Chatila, A De Luca, A De Santis, G Giralt, G Hirzinger, V Lippiello, R Mattone, et al. Physical human-robot interaction in anthropic domains : Safety and dependability. In *Proceedings of the 4th IARP/IEEE-RAS/EURON Workshop on Technical Challenges for Dependable Robots in Human Environments, T17-06, Nagoya*, volume 3, 2005.
- [5] RobotCub Consortium Alessandro Scalzo. icubgui documentation, 2009.
- [6] Heni Ben Amor, Gerhard Neumann, Sanket Kamthe, Oliver Kroemer, and Jan Peters. Interaction primitives for human-robot cooperation tasks. In *Robotics and Automation (ICRA)*, pages 2831–2837. IEEE, 2014.
- [7] Salvatore M Anzalone, Sofiane Boucenna, Serena Ivaldi, and Mohamed Chetouani. Evaluating the engagement with social robots. *International Journal of Social Robotics*, 7(4) :465–478, 2015.
- [8] Salvatore M Anzalone, Serena Ivaldi, Olivier Sigaud, and Mohamed Chetouani. Multimodal people engagement with icub. In *Biologically inspired cognitive architectures 2012*, pages 59–64. Springer, 2013.
- [9] Thomas Bader, Matthias Vogelgesang, and Edmund Klaus. Multimodal integration of natural gaze behavior for intention recognition during object manipulation. In *Proceedings of the 2009 international conference on Multimodal interfaces*, pages 199–206. ACM, 2009.
- [10] Shumeet Baluja and Dean Pomerleau. Non-intrusive gaze tracking using artificial neural networks. In *Advances in Neural Information Processing Systems*, pages 753–760, 1994.
- [11] Jimmy Baraglia, Maya Cakmak, Yukie Nagai, Rajesh Rao, and Minoru Asada. Initiative in robot assistance during collaborative task execution. In *Human-Robot Interaction (HRI), 11th ACM/IEEE International Conference on*, pages 67–74, 2016.

- [12] Christoph Bartneck, Dana Kulić, Elizabeth Croft, and Susana Zoghbi. Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. *International journal of social robotics*, pages 71–81, 2009.
- [13] Luca Bascetta, Gianni Ferretti, Paolo Rocco, Håkan Ardö, Herman Bruyninckx, Eric Demeester, and Enrico Di Lello. Towards safe human-robot interaction in robotic cells : an approach based on visual tracking and intention estimation. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 2971–2978, 2011.
- [14] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning : A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8) :1798–1828, 2013.
- [15] Viridiana L Benitez and Linda B Smith. Predictable locations aid early object name learning. *Cognition*, 125(3) :339–352, 2012.
- [16] B. Berret, S. Ivaldi, F. Nori, and G. Sandini. Stochastic optimal control with variable impedance manipulators in presence of uncertainties and delayed feedback. In *Proc. of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS*, pages 4354–4359, San Francisco, CA, USA., 2011.
- [17] Aude Billard and Maja J Matarić. Learning human arm movements by imitation : : Evaluation of a biologically inspired connectionist architecture. *Robotics and Autonomous Systems*, 37(2) :145–160, 2001.
- [18] Christopher M. Bishop. *Pattern recognition and machine learning, 5th Edition*. Information science and statistics. Springer, 2007.
- [19] MJ Bishop and Elizabeth A Thompson. Maximum likelihood alignment of dna sequences. *Journal of molecular biology*, 190(2) :159–165, 1986.
- [20] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference : A review for statisticians. *CoRR*, 2016.
- [21] Matthew Botvinick and Jonathan Cohen. Rubber hands ‘feel’ touch that eyes see. *Nature*, 391(6669) :756, 1998.
- [22] Sofiane Boucenna, Philippe Gaussier, Pierre Andry, and Laurence Hafemeister. A robot learns the facial expressions recognition and face/non-face discrimination through an imitation game. *International Journal of Social Robotics*, 6(4) :633–652, 2014.
- [23] Matthew Brand, Nuria Oliver, and Alex Pentland. Coupled hidden markov models for complex action recognition. In *Computer vision and pattern recognition. proceedings., IEEE computer society conference on*, pages 994–999, 1997.
- [24] Myles Brand. *Intending and Acting : Toward a Naturalized Action Theory*. Bradford Bks. Mit Press, 1984.
- [25] M. Bratman. *Intention, plans, and practical reason*. Harvard University Press, Cambridge, MA, 1987.
- [26] Inge Bretherton. Intentional communication and the development of an understanding of mind. *Children’s theories of mind : Mental states and social understanding*, pages 49–75, 1991.
- [27] Rechele Brooks and Andrew N Meltzoff. The importance of eyes : how infants interpret adult looking behavior. *Developmental psychology*, 38(6) :958, 2002.
- [28] Martin D Buhmann. Radial basis functions. *Acta numerica*, 9 :1–38, 2000.

-
- [29] Hung H Bui, Dinh Q Phung, and Svetha Venkatesh. Hierarchical hidden markov models with general state hierarchy. In *Proceedings of the National Conference on Artificial Intelligence*, pages 324–329. Menlo Park, CA ; Cambridge, MA ; London ; AAAI Press ; MIT Press ; 1999, 2004.
- [30] Denis Burnham and Barbara Dodd. Auditory–visual speech integration by prelinguistic infants : Perception of an emergent consonant in the mcgurk effect. *Developmental psychobiology*, 45(4) :204–220, 2004.
- [31] Baptiste Busch, Jonathan Grizou, Manuel Lopes, and Freek Stulp. Learning legible motion from human–robot interactions. *International Journal of Social Robotics*, pages 1–15, 2017.
- [32] George Butterworth and Nicholas Jarrett. What minds have in common is space : Spatial mechanisms serving joint visual attention in infancy. *British journal of developmental psychology*, 9(1) :55–72, 1991.
- [33] Martin V Butz, Olivier Sigaud, and Pierre Gerard. Internal models and anticipations in adaptive learning systems. *Lecture Notes in Computer Science*, pages 86–109, 2003.
- [34] Hilary Buxton. Learning and understanding dynamic scene activity : a review. *Image and vision computing*, 21(1) :125–136, 2003.
- [35] S. Calinon. A tutorial on task-parameterized movement learning and retrieval. *Intelligent Service Robotics*, 9(1) :1–29, 2016.
- [36] S. Calinon and al. Statistical dynamical systems for skills acquisition in humanoids, <http://www.calinon.ch/showPubli.php?publi=3031>, 2012.
- [37] S. Calinon, D. Bruno, and D. G. Caldwell. A task-parameterized probabilistic model with minimal intervention control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3339–3344, May 2014.
- [38] S. Calinon, Z. Li, T. Alizadeh, N. G. Tsagarakis, and D. G. Caldwell. Statistical dynamical systems for skills acquisition in humanoids. In *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*, pages 323–329, Osaka, Japan, 2012.
- [39] Sylvain Calinon. pbdlib-matlab, <https://gitlab.idiap.ch/rli/pbdlib-matlab/>, 2015.
- [40] Sylvain Calinon, Florent D’halluin, Eric L Sauser, Darwin G Caldwell, and Aude G Billard. Learning and reproduction of gestures by imitation. *IEEE Robotics & Automation Magazine*, 17(2) :44–54, 2010.
- [41] Angelo Cangelosi, Matthew Schlesinger, and Linda B Smith. *Developmental robotics : From babies to robots*. MIT Press, 2015.
- [42] Tom Carlson and Yiannis Demiris. Human-wheelchair collaboration through prediction of intention and adaptive assistance. In *Robotics and Automation. ICRA. IEEE International Conference on*, pages 3926–3931, 2008.
- [43] Ginevra Castellano, André Pereira, Iolanda Leite, Ana Paiva, and Peter W McOwan. Detecting user engagement with a robot companion using task and social interaction-based features. In *Proceedings of the 2009 international conference on Multimodal interfaces*, pages 119–126. ACM, 2009.
- [44] Harish chaandar Ravichandar, Avnish Kumar, and Ashwin Dani. Bayesian human intention inference through multiple model filtering with gaze-based priors. In *Information Fusion (FUSION), 2016 19th International Conference on*, pages 2296–2302. IEEE.
- [45] Charlie Chaplin. *Les temps modernes*, 1936.

- [46] Konstantinos Chatzilygeroudis and Jean-Baptiste Mouret. Using parameterized black-box priors to scale up model-based policy search for robotics. *arXiv preprint arXiv :1709.06917*, 2017.
- [47] M. Chaverroche, A. Malaisé, F. Colas, F. Charpillet, and S. Ivaldi. A Variational Time Series Feature Extractor for Action Prediction. *ArXiv e-prints*, July 2018.
- [48] Nutan Chen, Maximilian Karl, and Patrick van der Smagt. Dynamic movement primitives in latent space of time-dependent variational autoencoders. In *16th IEEE-RAS, Humanoids, Cancun, Mexico*, pages 629–636, 2016.
- [49] J Edward Colgate, J Edward, Michael A Peshkin, and Witaya Wannasuphprasit. *Cobots : Robots for collaboration with human operators*. 1996.
- [50] Adrià Colomé, Gerhard Neumann, Jan Peters, and Carme Torras. Dimensionality reduction for probabilistic movement primitives. In *Humanoid Robots (Humanoids)*, pages 794–800. IEEE, 2014.
- [51] David Cristinacce and Tim Cootes. Feature detection and tracking with constrained local models. In *Proc. British Machine Vision Conference*, volume 3, pages 929–938, 2006.
- [52] David Cristinacce and Timothy F Cootes. A comparison of shape constrained facial feature detectors. In *Automatic Face and Gesture Recognition, 2004. Proceedings. 6-th IEEE International Conference on*, pages 375–380, 2004.
- [53] David Cristinacce and Timothy F Cootes. Facial feature detection and tracking with automatic template selection. In *Automatic Face and Gesture Recognition. FGR. 7th International Conference on*, pages 429–434. IEEE, 2006.
- [54] Gergely Csibra and György Gergely. 'obsessed with goals' : Functions and mechanisms of teleological interpretation of actions in humans. *Acta psychologica*, 124(1) :60–78, 2007.
- [55] Andrea d'Avella, Philippe Saltiel, and Emilio Bizzi. Combinations of muscle synergies in the construction of a natural motor behavior. *Nature neuroscience*, 6(3) :300, 2003.
- [56] Donald Davidson. *Essays on actions and events : Philosophical essays*, volume 1. Oxford University Press on Demand, 2001.
- [57] Fred D Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, pages 319–340, 1989.
- [58] Agostino De Santis, Bruno Siciliano, Alessandro De Luca, and Antonio Bicchi. An atlas of physical human–robot interaction. *Mechanism and Machine Theory*, 43(3) :253–270, 2008.
- [59] Gedeon O Deak, Anna M Krasno, Jochen Triesch, Joshua Lewis, and Leigh Sepeta. Watch the hands : infants can learn to follow gaze by seeing adults manipulate objects. *Developmental science*, 17(2) :270–281, 2014.
- [60] Sarah Degallier, Ludovic Righetti, Sebastien Gay, and Auke Ijspeert. Toward simple control for complex, autonomous robotic applications : combining discrete and rhythmic motor primitives. *Autonomous Robots*, 31(2-3) :155–181, 2011.
- [61] Yiannis Demiris. Prediction of intent in robotics and multi-agent systems. *Cognitive processing*, 8(3) :151–158, 2007.
- [62] Yiannis Demiris and Bassam Khadhouri. Hierarchical attentive multiple models for execution and recognition of actions. *Robotics and autonomous systems*, 54(5) :361–369, 2006.
- [63] Miha Deniša, Andrej Gams, Aleš Ude, and Tadej Petrič. Learning compliant movement primitives through demonstration and statistical generalization. *IEEE/ASME Transactions on Mechatronics*, 21(5) :2581–2594, 2016.

-
- [64] Travis DeWolf. Dynamic movement primitives part 1 : The basics, 2013.
- [65] Travis DeWolf. pydmps, <https://github.com/studywolf/pydmps>, 2013.
- [66] Rüdiger Dillmann, Regine Becher, and Peter Steinhaus. Armar ii-a learning and cooperative multimodal humanoid robot system. *International Journal of Humanoid Robotics*, 1(01) :143–155, 2004.
- [67] Carl Doersch. Tutorial on variational autoencoders. *CoRR*, 2016.
- [68] Anca Dragan and Siddhartha Srinivasa. Generating legible motion. In *Proceedings of Robotics : Science and Systems*, Berlin, Germany, June 2013.
- [69] Anca Dragan and Siddhartha Srinivasa. Integrating human observer inferences into robot motion planning. *Autonomous Robots*, 37(4) :351–368, 2014.
- [70] Alain Droniou, Serena Ivaldi, and Olivier Sigaud. Deep unsupervised network for multimodal perception, representation and classification. *RAS*, 71 :83–98, 2015.
- [71] Gareth J Edwards, Timothy F Cootes, and Christopher J Taylor. Face recognition using active appearance models. In *European conference on computer vision*, pages 581–595. Springer, 1998.
- [72] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2) :179–211, 1990.
- [73] Elise Lucet et Al. Cash investigation - travail : ton univers impitoyable ; 16e minute., 2017.
- [74] Marco Ewerton. Learning motor skills from partially observed movements executed at different speeds, <https://github.com/studywolf/pydmps>, 2016.
- [75] Marco Ewerton. Learning motor skills from partially observed movements executed at different speeds, <https://github.com/studywolf/pydmps>, 2016.
- [76] Marco Ewerton, Guilherme Maeda, Jan Peters, and Gerhard Neumann. Learning motor skills from partially observed movements executed at different speeds. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 456–463, 2015.
- [77] Marco Ewerton, Gerhard Neumann, Rudolf Lioutikov, Heni Ben Amor, Jan Peters, and Guilherme Maeda. Learning multiple collaborative tasks with a mixture of interaction primitives. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 1535–1542, 2015.
- [78] Maddalena Fabbri-Destro and Giacomo Rizzolatti. Mirror neurons and mirror systems in monkeys and humans. *Physiology*, 23(3) :171–179, 2008.
- [79] Yunis Fanger, Jonas Umlauf, and Sandra Hirche. Gaussian processes for dynamic movement primitives with application in knowledge-based cooperation. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 3913–3919, 2016.
- [80] Gonzalo Ferrer and Alberto Sanfeliu. Bayesian human motion intentionality prediction in urban environments. *Pattern Recognition Letters*, 44 :134–140, 2014.
- [81] Ian C Fiebelkorn, John J Foxe, and Sophie Molholm. Dual mechanisms for the cross-sensory spread of attention : how much do learned associations matter ? *Cerebral Cortex*, 20(1) :109–120, 2009.
- [82] Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical hidden markov model : Analysis and applications. *Machine learning*, 32(1) :41–62, 1998.
- [83] Paul M Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology : General*, 121(3) :262, 1992.

- [84] Leonardo Fogassi, Pier Francesco Ferrari, Benno Gesierich, Stefano Rozzi, Fabian Chersi, and Giacomo Rizzolatti. Parietal lobe : from action organization to intention understanding. *Science*, 308(5722) :662–667, 2005.
- [85] G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3) :268–278, 1973.
- [86] Alexandra Fort, Claude Delpuech, Jacques Pernier, and Marie-Helene Giard. Early auditory–visual interactions in human cortex during nonredundant target identification. *Cognitive Brain Research*, 14(1) :20–30, 2002.
- [87] Abram L Friesen and Rajesh PN Rao. Gaze following as goal inference : A bayesian model. In *Proceedings of the Cognitive Science Society*, volume 33, 2011.
- [88] Geomagic. Geomagic touch informations.
- [89] Milad Geravand, Fabrizio Flacco, and Alessandro De Luca. Human-robot physical interaction and collaboration using an industrial robot with a closed control architecture. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 4000–4007, 2013.
- [90] György Gergely and Gergely Csibra. Teleological reasoning in infancy : The naive theory of rational action. *Trends in cognitive sciences*, 7(7) :287–292, 2003.
- [91] György Gergely, Zoltán Nádasdy, Gergely Csibra, and Szilvia Bíró. Taking the intentional stance at 12 months of age. *Cognition*, 56(2) :165–193, 1995.
- [92] Simon F Giszter, Ferdinando A Mussa-Ivaldi, and Emilio Bizzi. Convergent force fields organized in the frog’s spinal cord. *Journal of neuroscience*, 13(2) :467–491, 1993.
- [93] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.
- [94] Alvin I Goldman. *Simulating minds : The philosophy, psychology, and neuroscience of mindreading*. Oxford University Press, 2006.
- [95] Alvin I Goldman. *Theory of human action*. Princeton University Press, 2015.
- [96] Alvin I Goldman et al. Theory of mind. *The Oxford handbook of philosophy of cognitive science*, pages 402–424, 2012.
- [97] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [98] Robert M Gordon. Folk psychology as simulation. *Mind & Language*, 1(2) :158–171, 1986.
- [99] Elena Gribovskaya, Abderrahmane Kheddar, and Aude Billard. Motion learning and adaptive impedance for robot control during physical interaction with humans. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 4326–4332, 2011.
- [100] M. Hersch, F. Guenter, S. Calinon, and Billard. Dynamical system modulation for robot adaptive learning via kinesthetic demonstrations, <http://lisa.epfl.ch/sourcecode/counter.php?ID=11&index=1>, 2008.
- [101] Guy Hoffman. Anticipation in human-robot interaction. In *AAAI Spring Symposium : It’s All in the Timing*, 2010.
- [102] Matthew W. Hoffman, David B. Grimes, Aaron P. Shon, and Rajesh P.N. Rao. A probabilistic model of gaze imitation and shared attention. *Neural Networks*, 19(3) :299 – 310, 2006. The Brain Mechanisms of Imitation Learning.

-
- [103] Heiko Hoffmann, Peter Pastor, Dae-Hyung Park, and Stefan Schaal. Biologically-inspired dynamical systems for movement generation : automatic real-time goal adaptation and obstacle avoidance. In *Robotics and Automation. ICRA. IEEE International Conference on*, pages 2587–2592, 2009.
- [104] Neville Hogan. An organizing principle for a class of voluntary movements. *Journal of Neuroscience*, 4(11) :2745–2754, 1984.
- [105] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6) :417, 1933.
- [106] Chien-Ming Huang and Bilge Mutlu. Learning-based modeling of multimodal behaviors for humanlike robots. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 57–64. ACM, 2014.
- [107] Chien-Ming Huang and Bilge Mutlu. Anticipatory robot control for efficient human-robot collaboration. In *Human-Robot Interaction (HRI), 11th ACM/IEEE International Conference on*, pages 83–90, 2016.
- [108] Sandy H Huang, David Held, Pieter Abbeel, and Anca D Dragan. Enabling robots to communicate their objectives. *arXiv preprint arXiv :1702.03465*, 2017.
- [109] Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamical movement primitives : learning attractor models for motor behaviors. *Neural computation*, 25(2) :328–373, 2013.
- [110] Ryo Ishii, Yuta Shinohara, T Nakano, and Toyooki Nishida. Combining multiple types of eye-gaze information to predict user’s conversational engagement. In *2nd workshop on eye gaze on intelligent human machine interaction*, 2011.
- [111] S. Ivaldi. Intelligent human-robot collaboration with prediction and anticipation. *ERCIM news*, 114 :9–11, July 2018.
- [112] S. Ivaldi, M. Fumagalli, F. Nori, M. Baglietto, G. Metta, and G. Sandini. Approximate optimal control for reaching and trajectory planning in a humanoid robot. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS*, pages 1290–1296, Taipei, Taiwan, 2010.
- [113] S. Ivaldi, S. M. Nguyen, N. Lyubova, A. Droniou, V. Padois, D. Filliat, P. Y. Oudeyer, and O. Sigaud. Object learning through active exploration. *IEEE Transactions on Autonomous Mental Development*, 6(1) :56–72, March 2014.
- [114] Serena Ivaldi, Salvatore Anzalone, Woody Rouseau, Olivier Sigaud, and Mohamed Che-touani. Robot initiative in a team learning task increases the rhythm of interaction but not the perceived engagement. *Frontiers in Neurorobotics*, 8 :5, 2014.
- [115] Serena Ivaldi, Lars Fritzsche, Jan Babič, Freek Stulp, Michael Damsgaard, Bernhard Graimann, Giovanni Bellusci, and Francesco Nori. Anticipatory models of human movements and dynamics : the roadmap of the andy project. In *DHM*, 2017.
- [116] Serena Ivaldi, Matteo Fumagalli, Marco Randazzo, Francesco Nori, Giorgio Metta, and Giulio Sandini. Computing robot internal/external wrenches by means of inertial, tactile and f/t sensors : theory and implementation on the icub. In *Humanoid Robots (Humanoids), 11th IEEE-RAS International Conference on*, pages 521–528, 2011.
- [117] Serena Ivaldi, Matteo Fumagalli, Marco Randazzo, Francesco Nori, Giorgio Metta, and Giulio Sandini. icub dynamics, 2011.

- [118] Serena Ivaldi, Sebastien Lefort, Jan Peters, Mohamed Chetouani, Joelle Provasi, and Elisabetta Zibetti. Towards engagement models that consider individual factors in hri : on the relation of extroversion and negative attitude towards robots to gaze and speech during a human-robot assembly task. *arXiv preprint arXiv :1508.04603*, 2015.
- [119] Serena Ivaldi, Sebastien Lefort, Jan Peters, Mohamed Chetouani, Joelle Provasi, and Elisabetta Zibetti. Towards engagement models that consider individual factors in hri : on the relation of extroversion and negative attitude towards robots to gaze and speech during a human-robot assembly task. *International Journal of Social Robotics*, 9 :63–86, 2017.
- [120] Herbert Jaeger. The “echo state” approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany : German National Research Center for Information Technology GMD Technical Report*, 148(34) :13, 2001.
- [121] L. Jamone, E. Ugur, A. Cangelosi, L. Fadiga, A. Bernardino, J. Piater, and J. Santos-Victor. Affordances in psychology, neuroscience and robotics : a survey. *IEEE Transactions on Cognitive and Developmental Systems*, PP(99) :1–1, 2017.
- [122] Nathanaël Jarrassé, Jamie Paik, Viviane Pasqui, and Guillaume Morel. How can human motion prediction increase transparency? In *Robotics and Automation. ICRA. IEEE International Conference on*, pages 2134–2139, 2008.
- [123] Jonas T Kaplan and Marco Iacoboni. Getting a grip on other minds : Mirror neurons, intention understanding, and cognitive empathy. *Social neuroscience*, 1(3-4) :175–183, 2006.
- [124] Maximilian Karl, Maximilian Sölch, Justin Bayer, and Patrick van der Smagt. Deep variational bayes filters : Unsupervised learning of state space models from raw data. *CoRR*, 2016.
- [125] Eamonn Keogh. Exact indexing of dynamic time warping. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 406–417. VLDB Endowment, 2002.
- [126] Mohammad Khansari. Dynamical systems approach to learn robot motions, <https://bitbucket.org/khansari/seds>, 2011.
- [127] S Mohammad Khansari-Zadeh and Aude Billard. Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics*, 27(5) :943–957, 2011.
- [128] Seyed Mohammad Khansari-Zadeh and Aude Billard. A dynamical system approach to realtime obstacle avoidance. *Autonomous Robots*, 32(4) :433–454, 2012.
- [129] Oussama Khatib, Emel Demircan, Vincent De Sapio, Luis Sentis, Thor Besier, and Scott Delp. Robotics-based synthesis of human motion. *Journal of Physiology-Paris*, 103(3-5) :211–219, 2009.
- [130] Joseph Kim, Christopher J Banks, and Julie A Shah. Collaborative planning with encoding of users’ high-level strategies. In *AAAI Conference on Artificial Intelligence (AAAI-17)*, 2017.
- [131] Seungsu Kim, ChangHwan Kim, and Jong Hyeon Park. Human-like arm motion generation for humanoid robots using motion capture database. In *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, pages 3486–3491, 2006.
- [132] Diederik P Kingma and Jimmy Ba. Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*, 2014.
- [133] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, 2013.

-
- [134] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2149–2154, Sendai, Japan, Sep 2004.
- [135] Kazuhiro Kosuge and Norihide Kazamura. Control of a robot handling an object in cooperation with a human. In *Robot and Human Communication. RO-MAN. Proceedings. 6th IEEE International Workshop on*, pages 142–147, 1997.
- [136] Hideki Kozima and Eric Vatikiotis-Bateson. Communicative criteria for processing time/space-varying information. In *Robot and Human Interactive Communication. Proceedings. 10th IEEE International Workshop on*, pages 377–382, 2001.
- [137] Hideki Kozima and Hiroyuki Yano. A robot that learns to communicate with human caregivers. In *Proceedings of the First International Workshop on Epigenetic Robotics*, pages 47–52, 2001.
- [138] Gary D Langolf, Don B Chaffin, and James A Foulke. An investigation of fitts’ law using a wide range of movement amplitudes. *Journal of Motor Behavior*, 8(2) :113–128, 1976.
- [139] Ryan Lober. Stochastic machine learning toolbox, <https://github.com/rlober/smlt>, 2014.
- [140] Ryan Lober, Vincent Padois, and Olivier Sigaud. Multiple task optimization using dynamical movement primitives for whole-body reactive control. In *Humanoid Robots (Humanoids)*, pages 193–198. IEEE, 2014.
- [141] Chunling Ma, Helmut Prendinger, and Mitsuru Ishizuka. Eye movement as an indicator of users’ involvement with embodied interfaces at the low level. In *Proc. AISB Symposium on Conversational Informatics for Supporting Social Intelligence and Interaction-Situational and Environmental Information Enforcing Involvement in Conversation, University of Hertfordshire, Hatfield, UK*, pages 136–143, 2005.
- [142] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states : A new framework for neural computation based on perturbations. *Neural computation*, 14(11) :2531–2560, 2002.
- [143] Guilherme Maeda, Marco Ewerton, Rudolf Lioutikov, Heni Ben Amor, Jan Peters, and Gerhard Neumann. Learning interaction for collaborative tasks with probabilistic movement primitives. In *Humanoid Robots (Humanoids)*, pages 527–534. IEEE, 2014.
- [144] Guilherme J Maeda, Gerhard Neumann, Marco Ewerton, Rudolf Lioutikov, Oliver Kroemer, and Jan Peters. Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks. *Autonomous Robots*, pages 1–20, 2016.
- [145] Guilherme J Maeda, Gerhard Neumann, Marco Ewerton, Rudolf Lioutikov, Oliver Kroemer, and Jan Peters. Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks. *Autonomous Robots*, 41(3) :593–612, 2017.
- [146] Adrien Malaisé, Pauline Maurice, Francis Colas, François Charpillet, and Serena Ivaldi. Activity recognition with multiple wearable sensors for industrial applications. In *Advances in Computer-Human Interactions*, 2018.
- [147] Olivier Mangin and Pierre-Yves Oudeyer. Learning semantic components from subsymbolic multimodal perception. In *Development and Learning and Epigenetic Robotics (ICDL)*, pages 1–7. IEEE, 2013.
- [148] Harry McGurk and John MacDonald. Hearing lips and seeing voices. *Nature*, 264(5588) :746, 1976.

- [149] David McNeill. *Hand and mind : What gestures reveal about thought*. University of Chicago press, 1992.
- [150] Franziska Meier and Stefan Schaal. A probabilistic representation for dynamic movement primitives. *arXiv preprint arXiv :1612.05932*, 2016.
- [151] Alfred R Mele. *Springs of action : Understanding intentional behavior*. Oxford University Press on Deine 4639 mand, 1992.
- [152] Andrew N Meltzoff. Understanding the intentions of others : Re-enactment of intended acts by 18-month-old children. *Developmental psychology*, 31(5) :838, 1995.
- [153] Andrew N Meltzoff. 'Like me' : a foundation for social cognition. *Developmental science*, 10(1) :126–134, 2007.
- [154] Andrew N Meltzoff and Rechele Brooks. Eyes wide shut : The importance of eyes in infant gaze following and understanding other minds. *Gaze following : Its development and significance*, ed. R. Flom, K. Lee & D. Muir. Erlbaum.[EVH], 2007.
- [155] Giorgio Metta, Paul Fitzpatrick, and Lorenzo Natale. Yarp : yet another robot platform. *International Journal of Advanced Robotic Systems*, 3(1) :8, 2006.
- [156] Giorgio Metta, Lorenzo Natale, Francesco Nori, Giulio Sandini, David Vernon, Luciano Fadiga, Claes Von Hofsten, Kerstin Rosander, Manuel Lopes, José Santos-Victor, et al. The icub humanoid robot : An open-systems platform for research in cognitive development. *Neural Networks*, 23(8-9) :1125–1134, 2010.
- [157] Hersch Micha, , and Billard Aude. Dynamical system modulation for robot learning via kinesthetic demonstrations. *IEEE Transactions on Robotics*, 24(6), 2008.
- [158] Ikuhisa Mitsugami, Norimichi Ukita, and Masatsugu Kidode. Robot navigation by eye pointing. *Lecture notes in computer science*, 3711 :256, 2005.
- [159] Alexander Mörtl, Martin Lawitzky, Ayse Kucukyilmaz, Metin Sezgin, Cagatay Basdogan, and Sandra Hirche. The role of roles : Physical cooperation between humans and robots. *The International Journal of Robotics Research*, 31(13) :1656–1674, 2012.
- [160] Katharina Mülling, Jens Kober, Oliver Kroemer, and Jan Peters. Learning to select and generalize striking movements in robot table tennis. *The International Journal of Robotics Research*, 32(3) :263–279, 2013.
- [161] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.
- [162] Lorenzo Natale, Francesco Nori, Giorgio Metta, Matteo Fumagalli, Serena Ivaldi, Ugo Pattacini, Marco Randazzo, Alexander Schmitz, and Giulio Sandini. The icub platform : a tool for studying intrinsically motivated learning. In *Intrinsically motivated learning in natural and artificial systems*, pages 433–458. Springer, 2013.
- [163] Tatsuya Nomura, Takayuki Kanda, and Tomohiro Suzuki. Experimental investigation into influence of negative attitudes toward robots on human–robot interaction. *Ai & Society*, 20(2) :138–150, 2006.
- [164] Tatsuya Nomura, Takayuki Kanda, Tomohiro Suzuki, and Kennsuke Kato. Psychology in human-robot communication : An attempt through investigation of negative attitudes and anxiety toward robots. In *Robot and Human Interactive Communication. ROMAN. 13th IEEE International Workshop on*, pages 35–40, 2004.

-
- [165] Francesco Nori, Silvio Traversaro, Jorhabib Eljaik, Francesco Romano, Andrea Del Prete, and Daniele Pucci. icub whole-body control through force regulation on rigid non-coplanar contacts. *Frontiers in Robotics and AI*, 2 :6, 2015.
- [166] Ori Novanda, Maha Salem, Joe Saunders, Michael L Walters, and Kerstin Dautenhahn. What communication modalities do users prefer in real time hri? *arXiv preprint arXiv :1606.03992*, 2016.
- [167] Elisabeth Pacherie. La dynamique des intentions. *Dialogue : Canadian Philosophical Review/Revue canadienne de philosophie*, 42(3) :447–480, 2003.
- [168] Rainer W Paine and Jun Tani. Motor primitive and sequence self-organization in a hierarchical recurrent neural network. *Neural Networks*, 17(8-9) :1291–1309, 2004.
- [169] Oskar Palinko, Alessandra Sciutti, Laura Patané, Francesco Rea, Francesco Nori, and Giulio Sandini. Communicative lifting actions in human-humanoid interaction. In *Humanoid Robots (Humanoids)*, pages 1116–1121. IEEE, 2014.
- [170] Alexandros Paraschos, Christian Daniel, Jan R Peters, and Gerhard Neumann. Probabilistic movement primitives. In *Advances in neural information processing systems*, pages 2616–2624, 2013.
- [171] Alexandros Paraschos, Gerhard Neumann, and Jan Peters. A probabilistic approach to robot trajectory generation. In *Humanoid Robots (Humanoids)*, pages 477–483. IEEE, 2013.
- [172] Alexandros Paraschos, Elmar Rueckert, Jan Peters, and Gerhard Neumann. Model-free probabilistic movement primitives for physical interaction. In *Intelligent Robots and Systems (IROS)*, pages 2860–2866. IEEE, 2015.
- [173] Ugo Pattacini. forward-inverse kinematics of serial-links chains of revolute joints and icub limbs, 2011.
- [174] Ugo Pattacini. Modular cartesian controllers for humanoid robots : Design and implementation on the icub. *Dizertacná práca, Istituto Italiano di Tecnologia*, 2011.
- [175] Ugo Pattacini. Modular cartesian controllers for humanoid robots : Design and implementation on the icub. *Dizertacná práca, Istituto Italiano di Tecnologia*, 2011.
- [176] Ugo Pattacini. Generic yarp driver for haptic devices, 2015.
- [177] Alexis Pautrat. *Experimental quantitative studies that the solidification of the superalloys 718 in investment casting*. Theses, Ecole Nationale Supérieure des Mines de Paris, July 2013.
- [178] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11) :559–572, 1901.
- [179] Lugi Penco, Brice Clement, Valerio Mougno, Wolfman Enrico Mingo, Gabriele Nava, Daniele Pucci, Nikos Tsagarakis, Jean-Baptiste Mouret, and Serena Ivaldi. Robust real-time whole-body motion retargeting from human to humanoid. *humanoids*, 2018.
- [180] Giovanni Pezzulo, Martin V Butz, and Cristiano Castelfranchi. The anticipatory approach : definitions and taxonomies. In *The challenge of anticipation*, pages 23–43. Springer, 2008.
- [181] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2) :257–286, 1989.
- [182] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer, 2004.
- [183] European Union’s Horizon 2020 Research and Innovation Programme. Advancing anticipatory behaviors in dyadic human-robot collaboration, 2017.

- [184] Daniel C Richardson and Natasha Z Kirkham. Multimodal events and moving locations : Eye movements of adults and 6-month-olds reveal dynamic spatial indexing. *Journal of Experimental Psychology : General*, 133(1) :46, 2004.
- [185] Giacomo Rizzolatti and Laila Craighero. The mirror-neuron system. *Annu. Rev. Neurosci.*, 27 :169–192, 2004.
- [186] Giacomo Rizzolatti, Luciano Fadiga, Léonardo Fogassi, and Vittorio Gallese. Resonance behaviors and mirror neurons. *Archives italiennes de biologie*, 137(2) :85–100, 1999.
- [187] Giacomo Rizzolatti and Corrado Sinigaglia. *Mirrors in the brain : How our minds share actions and emotions*. Oxford University Press, USA, 2008.
- [188] Daniel Roetenberg, Henk Luinge, and Per Slycke. Xsens mvn : full 6dof human motion tracking using miniature inertial sensors. *Xsens Motion Technologies BV, Tech. Rep*, 1, 2009.
- [189] Robert Rosen. *Anticipatory Systems : Philosophical, Mathematical*. Pergamon Press, 1985.
- [190] Leonel Roza Castañeda, Sylvain Calinon, Darwin Caldwell, Pablo Jimenez Schlegl, and Carme Torras. Learning collaborative impedance-based robot behaviors. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pages 1422–1428, 2013.
- [191] Erol Şahin, Maya Çakmak, Mehmet R Doğar, Emre Uğur, and Göktürk Üçoluk. To afford or not to afford : A new formalization of affordances toward affordance-based robot control. *Adaptive Behavior*, 15(4) :447–472, 2007.
- [192] Stefan Schaal. Dynamic movement primitives : a framework for motor control in humans and humanoid robotics. In *Adaptive motion of animals and machines*, pages 261–280. Springer, 2006.
- [193] Stefan Schaal and Dagmar Sternad. Programmable pattern generators. In *3rd International Conference on Computational Intelligence in Neuroscience*, pages 48–51, 1998.
- [194] Benjamin Schrauwen, David Verstraeten, and Jan Van Campenhout. An overview of reservoir computing : theory, applications and implementations. In *Proceedings of the 15th European Symposium on Artificial Neural Networks. p. 471-482 2007*, pages 471–482, 2007.
- [195] Alessandra Sciutti, Ambra Bisio, Francesco Nori, Giorgio Metta, Luciano Fadiga, and Giulio Sandini. Robots can be perceived as goal-oriented agents. *Interaction Studies*, 14(3) :329–350, 2013.
- [196] Alessandra Sciutti, Laura Patane, Francesco Nori, and Giulio Sandini. Understanding object weight from human and humanoid lifting actions. *IEEE Transactions on Autonomous Mental Development*, 6(2) :80–92, 2014.
- [197] John R Searle. *Intentionality : An essay in the philosophy of mind*. Cambridge university press, 1983.
- [198] Natalie Sebanz, Harold Bekkering, and Günther Knoblich. Joint action : bodies and minds moving together. *Trends in cognitive sciences*, 10(2) :70–76, 2006.
- [199] RobotCub Consortium Serena Ivaldi, Matteo Fumagalli. idyn documentation, 2010.
- [200] Julie Shah, James Wiken, Brian Williams, and Cynthia Breazeal. Improved human-robot team performance using chaski, a human-inspired plan execution system. In *Proceedings of the 6th international conference on Human-robot interaction*, pages 29–36. ACM, 2011.
- [201] Aaron P Shon, Keith Grochow, and Rajesh PN Rao. Robotic imitation from human motion capture using gaussian processes. In *Humanoid Robots, 5th IEEE-RAS International Conference on*, pages 129–134, 2005.

-
- [202] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer, 2016.
- [203] Diego F Silva and Gustavo EAPA Batista. Speeding up all-pairwise dynamic time warping matrix calculation. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 837–845. SIAM, 2016.
- [204] Emrah Akin Sisbot, Luis F Marin-Urias, Rachid Alami, and Thierry Simeon. A human aware mobile robot motion planner. *IEEE Transactions on Robotics*, 23(5) :874–883, 2007.
- [205] JF Soechting. Effect of target size on spatial and temporal characteristics of a pointing movement in man. *Experimental Brain Research*, 54(1) :121–132, 1984.
- [206] Harold Soh and Yiannis Demiris. Learning assistance by demonstration : Smart mobility with shared control and paired haptic controllers. *Journal of Human-Robot Interaction*, 4(3) :76–100, 2015.
- [207] Le Song, Byron Boots, Sajid M Siddiqi, Geoffrey J Gordon, and Alex Smola. Hilbert space embeddings of hidden markov models. *Proceedings of the 27th International Conference on Machine Learning*, 2010.
- [208] Waltraud Stadler, Derek VM Ott, Anne Springer, Ricarda I Schubotz, Simone Schütz-Bosbach, and Wolfgang Prinz. Repetitive tms suggests a role of the human dorsal premotor cortex in action prediction. *Frontiers in human neuroscience*, 6, 2012.
- [209] F. Stulp, G. Raiola, A. Hoarau, S. Ivaldi, and O. Sigaud. Learning compact parameterized skills with a single regression. In *Proc. IEEE-RAS International Conference on Humanoid Robots - HUMANOIDS*, pages 1–7, 2013.
- [210] Freek Stulp. `DmpBbo` – a c++ library for black-box optimization of dynamical movement primitives, <https://github.com/stulp/dmpbbo>, 2014.
- [211] William H Sumby and Irwin Pollack. Visual contribution to speech intelligibility in noise. *The journal of the acoustical society of america*, 26(2) :212–215, 1954.
- [212] Aravind Suresh. Repository for eye gaze detection and tracking, 2015.
- [213] Durk Talsma, Daniel Senkowski, Salvador Soto-Faraco, and Marty G Woldorff. The multifaceted interplay between attention and multisensory integration. *Trends in cognitive sciences*, 14(9) :400–410, 2010.
- [214] the RobotCub Consortium. an open source cognitive humanoid robotic platform, 2004.
- [215] the RobotCub Consortium. icub balancing via external force control : first year results of the codyco project, 2014.
- [216] Serge Thill and Tom Ziemke. The role of intention in human-robot interaction. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '17, pages 427–428, New York, NY, USA. ACM.
- [217] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop : Divide the gradient by a running average of its recent magnitude. *COURSERA : Neural networks for machine learning*, 4(2) :26–31, 2012.
- [218] Fabian Timm and Erhardt Barth. Accurate eye centre localisation by means of gradients. *Visapp*, 11 :125–130, 2011.
- [219] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 61(3) :611–622, 1999.
- [220] Michael Tomasello and Michael Jeffrey Farrar. Joint attention and early language. *Child development*, pages 1454–1463, 1986.

- [221] V Javier Traver, Angel Pasqual del Pobil, and Miguel Pérez-Francisco. Making service robots human-safe. In *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. IEEE/RSJ International Conference on*, volume 1, pages 696–701, 2000.
- [222] Toru Tsumugiwa, Ryuichi Yokogawa, and Kei Hara. Variable impedance control based on estimation of human arm stiffness for human-robot cooperative calligraphic task. In *Robotics and Automation. Proceedings. ICRA. IEEE International Conference on*, volume 1, pages 644–650, 2002.
- [223] Toru Tsumugiwa, Ryuichi Yokogawa, and Kei Hara. Variable impedance control with virtual stiffness for human-robot cooperative peg-in-hole task. In *Intelligent Robots and Systems. IEEE/RSJ International Conference on*, volume 2, pages 1075–1081, 2002.
- [224] Cosimo Urgesi, Marta Maieron, Alessio Avenanti, Emmanuele Tidoni, Franco Fabbro, and Salvatore Maria Aglioti. Simulating the future of actions in the human corticospinal system. *Cerebral Cortex*, 20(11) :2511–2521, 2010.
- [225] Arlene S Walker-Andrews. Infants’ perception of expressive behaviors : differentiation of multimodal information. *Psychological bulletin*, 121(3) :437, 1997.
- [226] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models. In *NIPS*, volume 18, page 3, 2005.
- [227] Zheng Wang, Angelika Peer, and Martin Buss. An HMM approach to realistic haptic human-robot interaction. In *EuroHaptics conference, and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics. Third Joint*, pages 374–379. IEEE, 2009.
- [228] Zhikun Wang, Marc Peter Deisenroth, Heni Ben Amor, David Vogt, Bernhard Schölkopf, and Jan Peters. Probabilistic modeling of human movements for intention inference. In *Robotics : Science and Systems*. Citeseer, 2012.
- [229] Zhikun Wang, Katharina Mülling, Marc Peter Deisenroth, Heni Ben Amor, David Vogt, Bernhard Schölkopf, and Jan Peters. Probabilistic movement modeling for intention inference in human-robot interaction. *The International Journal of Robotics Research*, 32(7) :841–858, 2013.
- [230] Martin Weser, Daniel Westhoff, Markus Huser, and Jianwei Zhang. Multimodal people tracking and trajectory prediction based on learned generalized motion patterns. In *Multisensor Fusion and Integration for Intelligent Systems*, pages 541–546. IEEE, 2006.
- [231] Wiktionnaire. compliance — wiktionnaire,, 2017.
- [232] Gregor Wolbring and Sophya Yumakulov. Social robots : views of staff of a disability service organization. *International journal of social robotics*, 6(3) :457–468, 2014.
- [233] Rachel Wu and Natasha Z Kirkham. No two cues are alike : Depth of learning during infancy is dependent on what orients attention. *Journal of Experimental Child Psychology*, 107(2) :118–136, 2010.
- [234] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
- [235] © Xsens. Xsens – the leading innovator in 3d motion tracking technology., 2017.
- [236] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv :1505.00853*, 2015.

-
- [237] Yuichi Yamashita and Jun Tani. Emergence of functional hierarchy in a multiple timescale neural network model : a humanoid robot experiment. *PLoS computational biology*, 4(11) :e1000220, 2008.
- [238] Akiko Yamazaki, Keiichi Yamazaki, Yoshinori Kuno, Matthew Burdelski, Michie Kawashima, and Hideaki Kuzuoka. Precision timing in human-robot interaction : coordination of head movement and utterance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 131–140. ACM, 2008.
- [239] Chen Yu and Linda B Smith. Joint attention without gaze following : Human infants and their parents coordinate visual attention to objects through eye-hand coordination. *PloS one*, 8(11) :e79659, 2013.
- [240] Andrea Maria Zanchettin, Luca Bascetta, and Paolo Rocco. Achieving humanlike motion : Resolving redundancy for anthropomorphic industrial manipulators. *IEEE Robotics & Automation Magazine*, 20(4) :131–138, 2013.
- [241] A. Zube, J. Hofmann, and C. Frese. Model predictive contact control for human-robot interaction. In *Proceedings of ISR 2016 : 47st International Symposium on Robotics*, pages 1–7, June 2016.

Résumé

Cette thèse se situe à l'intersection de l'apprentissage automatique et de la robotique humanoïde, dans la thématique de l'interaction homme-robot, et dans le domaine de la cobotique (robotique collaborative). Elle se focalise sur les interactions non verbales humain-robot, en particulier sur l'interaction gestuelle. La prédiction de l'intention, la compréhension et la reproduction de gestes sont donc des questions centrales de cette thèse.

Dans un premier temps, il s'agit de faire apprendre au robot des gestes par démonstration : un utilisateur prend le robot par le bras et lui fait réaliser les gestes à apprendre et ce, plusieurs fois. Le robot doit ensuite être capable de reproduire ces différents mouvements tout en les généralisant pour s'adapter au contexte. Pour cela, à l'aide de ses capteurs proprioceptifs, il interprète les signaux perçus pour comprendre le mouvement que lui fait réaliser l'utilisateur, afin d'en générer des similaires par la suite. Dans un second temps, le robot apprend à reconnaître l'intention de l'humain avec lequel il interagit et cela, à partir des gestes que ce dernier initie : il s'agit ensuite pour le robot de produire les gestes adaptés à la situation et correspondant aux attentes de l'utilisateur. Cela nécessite que le robot comprenne la gestuelle de l'utilisateur. Pour cela, différentes modalités perceptives ont été explorées. À l'aide de capteurs proprioceptifs, le robot ressent les gestes de l'utilisateur au travers de son propre corps : il s'agit alors d'interaction physique humain-robot. À l'aide de capteurs visuels, le robot interprète le mouvement de la tête de l'utilisateur. Enfin, à l'aide de capteurs externes, le robot reconnaît et prédit le mouvement corps entier de l'utilisateur. Dans ce dernier cas, l'utilisateur porte lui-même des capteurs (vêtement X-Sens) qui transmettent sa posture au robot. De plus, le couplage de ces modalités a été étudié. D'un point de vue méthodologique, nous venons de voir que les questions d'apprentissage et de reconnaissance de séries temporelles (les gestes) ont été centrales dans cette thèse. Pour cela, deux approches ont été développées. La première est fondée sur la modélisation statistique de primitives de mouvements (correspondant aux gestes) : les ProMPs. La seconde, ajoute à la première du Deep Learning, par l'utilisation d'auto-encodeurs, afin de modéliser des gestes corps entier contenant beaucoup d'informations, tout en permettant une prédiction en temps réel mou. Lors de cette thèse, différents enjeux ont notamment été pris en compte pour la création et le développement de nos méthodes. Ces enjeux concernent : la prédiction des durées des trajectoires, la réduction de la charge cognitive et motrice imposée à l'utilisateur, le besoin de rapidité (temps réel mou) et de précision dans les prédictions.

Mots-clés: Prédiction, primitives de mouvements, apprentissage automatique, Interaction Homme-Robot, cobotique.

Abstract

This thesis lies at the intersection between machine learning and humanoid robotics, under the theme of human-robot interaction and within the cobotics (collaborative robotics) field. It focuses on prediction for non-verbal human-robot interactions, with an emphasis on gestural interaction. The prediction of the intention, understanding, and reproduction of gestures are therefore central topics of this thesis. First, the robots learn gestures by demonstration : a user grabs its arm and makes it perform the gestures to be learned several times. The robot must

then be able to reproduce these different movements while generalizing them to adapt them to the situation. To do so, using its proprioceptive sensors, it interprets the perceived signals to understand the movement made by the user in order to generate similar ones later on. Second, the robot learns to recognize the intention of the human partner based on the gestures that the human initiates : the robot then has to perform the gestures adapted to the situation and corresponding to the user's expectations. This requires the robot to understand the user's gestures. To this end, different perceptual modalities have been explored. Using proprioceptive sensors, the robot feels the user's gestures through its own body : it is then a question of physical human-robot interaction. Using visual sensors, the robot interprets the movement of the user's head. Finally, using external sensors, the robot recognizes and predicts the user's whole body movement. In that case, the user wears sensors (in our case, a wearable motion tracking suit by XSens) that transmit his posture to the robot. In addition, the coupling of these modalities was studied. From a methodological point of view, the learning and the recognition of time series (gestures) have been central to this thesis. In that aspect, two approaches have been developed. The first is based on the statistical modeling of movement primitives (corresponding to gestures) : ProMPs. The second adds Deep Learning to the first one, by using auto-encoders in order to model whole-body gestures containing a lot of information while allowing a prediction in soft real time. Various issues were taken into account during this thesis regarding the creation and development of our methods. These issues revolve around : the prediction of trajectory durations, the reduction of the cognitive and motor load imposed on the user, the need for speed (soft real-time) and accuracy in predictions.

Keywords: prediction, movement primitives, Human-Robot-interaction, machine learning, robotics.