Contributions to shared control architectures for advanced telemanipulation

Firas Abi-Farraj

THESE DE DOCTORAT DE

L'UNIVERSITE DE RENNES 1
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *« Automatique, Productique et Robotique »*

Par
« Firas ABI-FARRAJ »

« Contributions to Shared Control Architectures for Advanced Telemanipulation »

Thèse présentée et soutenue à « Rennes », le « 18/12/2018 »
Unité de recherche : Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA-UMR6074)
Thèse N° :

Rapporteurs avant soutenance :

Antonio Bicchi Full Professor in Robotics at the University of Pisa, Italy
Jee-Hwan Ryu Full Professor in Mechanical Engineering, KOREATECH, Republic of Korea

Composition du Jury :

Examinateurs :	Antonio Bicchi	Full Professor in Robotics at the University of Pisa, Italy
	François Chaumette	DR1 INRIA, Rennes, France
	Katherine Kuchenbecker	Director of the Haptic Intelligence Department, Max Planck Institute for Intelligent Systems, Germany
	Maud Marchal	Maitre de Conferénces, Univ. Rennes, INSA, IRISA
	Christian Ott	Senior Reseach Scientist, Head of Department "Analysis and Control of Advanced Robotic Systems" at Institute of Robotics and Mechatronics, DLR, Germany
	Jee-Hwan Ryu	Full Professor in Mechanical Engineering, KOREATECH, Republic of Korea
Dir. de thèse :	Paolo Robuffo Giordano	DR2 CNRS, Rennes France

Acknowledgements

Three years are coming to their end, and I am about to get my doctorate degree! Coming to Rennes, I didn't know what to expect, but it was a special and fruitful experience both at the scientific and personal level. I would like to dedicate these (few) lines to acknowledge the contribution of everyone who, in a way or another helped in reaching this point.

I would like to first thank Dr. Paolo Robuffo Giordano. Besides his thorough technical experience and continuous support, he is a great person to work with. Our (joyful) meetings made it always a pleasure to chat and discuss about any technical, scientific or general topic. I honestly couldn't have had a better adviser and I hope that this relationship goes beyond the PhD.

I would like to also thank Dr. François Chaumette for the excellent working environment and team spirit which he has developed as a group leader over the past years and for providing the guidance and support wherever needed. Special thanks as well to M. Fabien Spindler whose invaluable contribution and support to myself and to the group added significantly to this thesis. It was a pleasure working with you.

During my PhD I have also spent 6 months at the German Aerospace Center (DLR) in the Munich region. I would like to primarily thank Dr. Maximo Roa for accepting this visit and for his warm welcome and continuous support throughout my stay. My gratitude is also extended to Dr. Christian Ott for the same reasons. In this respect, I cannot refrain from thanking my DLR colleagues whose support was crucial to make the stay successful among whom I especially mention Bernd Henze, Alexander Werner and Oliver Porgers.

Special thanks go to my previous and current Rainbow/Lagadic colleagues whom I will refrain from mentioning by name as the list is long. Thank you for your continuous personal and technical support throughout the journey.

Finally, I would like to dedicate these last lines of acknowledgment to my family and friends in France, Lebanon and around the world for many reasons which I do not need to detail!

Résumé en français

Sujet de la thèse

Cette thèse s'inspire de la riche histoire de l'IHR pour analyser la relation entre l'opérateur humain et les systèmes autonomes. Les contrôleurs autonomes sont en évolution constante et rapide, en corrélation avec les progrès des capacités matérielles (par exemple, caméras de profondeur, capteurs plus efficaces, puissance de calcul) et le développement d'architectures logicielles (par exemple, méthodes basées sur la vision, contrôle de force, apprentissage automatique, IA). Cette évolution entraîne un déplacement continu d'autorité et de dépendance entre l'opérateur humain et l'autonomie des systèmes IHR, permettant une interaction plus significative entre les deux et un spectre plus large d'applications. Dans ce contexte, cette thèse vise à proposer une variété d'architectures de contrôle partagé pour les différentes phases des applications de télémanipulation à distance, caractérisées par différents niveaux d'autonomie pour le composant robotique, et différents moyens de fournir un retour d'information à l'opérateur sur l'exécution de la tâche et sur la faisabilité de ses commandes. En plus de l'intérêt scientifique général pour avancer l'état de l'art dans le domaine du contrôle partagé, cette thèse est également motivée par les besoins pratiques du project européen H2020 "Robotic Manipulation for Nuclear Sort and Segregation" (RoMaNS), qui a servi d'étude de cas (et de motivation sociétale) pour les différentes architectures de contrôle partagé proposées dans la thèse. Nous procédons maintenant à résumer brièvement le projet RoMaNS pour également fournir un contexte supplémentaire aux contributions de la thèse.

Le projet RoMaNS aborde des systèmes de contrôle autonomes, téléopératoires et partagés pour la manipulation à distance des déchets nucléaires hérités. Au Royaume-Uni, le nettoyage des déchets nucléaires du dernier demi-siècle représente l'un des projets d'assainissement de l'environnement les plus importants d'Europe, avec 1,4 millions de mètres cubes de déchets de niveau intermédiaire à traiter. Cer-

tains de ces déchets ont été stockés temporairement dans des conteneurs, dont la plupart contiennent des niveaux de contamination mixtes et parfois des contenus inconnus. Il peut être nécessaire d'altérer ou de couper ces conteneurs pour vérifier leur contenus avant de les trier et de les séparer. D'un autre côté, de grandes quantités de machinerie et d'infrastructures de centrales déclassées hautement contaminées devront être démolies, coupées et redimensionnées et traitées au besoin. Une grande partie de ce travail ne peut être effectuée que par des méthodes de manipulation à distance, car les niveaux élevés de matières radioactives sont dangereux pour l'homme.

Actuellement, il est proposé que les opérateurs trient et séparent en utilisant la téléopération manuelle qui, dans ce contexte, utilise le « teach pendant » standard ou un simple joystick pour déplacer manuellement le robot. Cette méthode soulève des problèmes de sécurité, de fiabilité et de débit, car l'opérateur n'est pas à proximité du robot et doit se fier aux images de la caméra et / ou aux images déformées par les fenêtres en verre au plomb (voir Fig. 1.4). Le projet a l'intention de :

- Développer un nouveau matériel pour doter les bras et les pinces des robots de fonctionnalités avancées, mais adaptés au déploiement dans des environnements à forte radiation.
- Développer des méthodes de perception robustes exploitant plusieurs modalités de perception (par exemple, caméras standard, IR).
- Développer des méthodes d'autonomie avancées pour des actions de saisie et de manipulation automatiques hautement adaptatives.
- Contrôler en temps réel le mouvement des bras du manipulateur à partir de données visuelles.
- Combiner les méthodes d'autonomie et de téléopération en utilisant l'état de l'art sur la planification d'initiatives mixtes, de l'autonomie variable et des approches de contrôle partagé.
- Fournir une interface visuelle et haptique efficace à l'opérateur humain.

Le consortium des projets est composé de cinq partenaires : l'Université de Birmingham (Royaume-Uni ; chef), le Commissariat à l'Énergie atomique et aux Énergies alternatives (France), la Technische Universität Darmstadt (Allemagne), le National Nuclear Laboratory (Royaume-Uni), le CNRS (France). Notre contribution au CNRS (Irisa et Inria, Rennes) porte sur les trois derniers points et notamment sur les buts suivants :

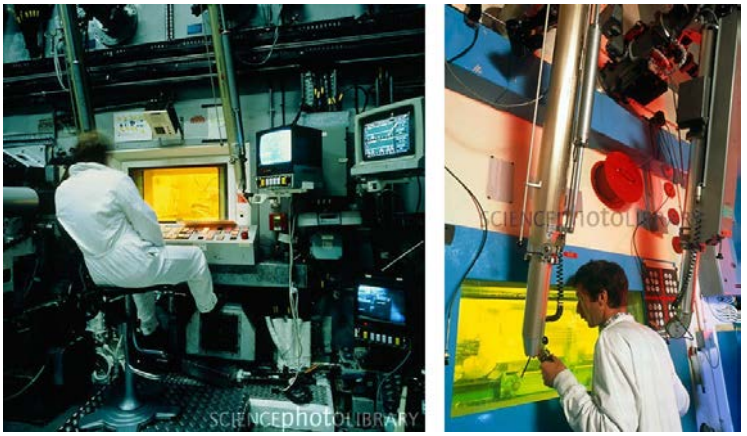


FIGURE 1 : Espace de travail actuel de l'opérateur humain pour l'application du tri et de la séparation. L'opérateur doit regarder à travers une petite fenêtre pour actionner les bras mécaniques.

- Développer des méthodologies de suivi visuel efficaces pour contrôler les manipulateurs en utilisant des approches d'asservissement visuel.
- Développer des interfaces haptiques efficaces et des dispositifs haptiques pouvant contribuer à une exécution plus efficace des tâches.
- Développer des approches de contrôle partagé avec des niveaux d'autonomie variables qui aident l'opérateur humain et diminuent sa charge cognitive.

Dans cette vision, cette thèse utilise le logiciel de suivi visuel et de contrôle développé par l'équipe pour proposer des solutions visuelles innovantes pour la télémanipulation assistée avancée. Elle se concentre sur trois aspects principaux : (*i*) augmenter l'efficacité et la rapidité de l'exécution des tâches, (*ii*) assurer la sécurité du système et (*iii*) améliorer l'expérience de l'opérateur et réduire sa charge cognitive. À cette fin, la thèse développe des architectures semi-autonomes qui peuvent aider l'opérateur humain à contrôler des parties du système qui peuvent être difficiles à contrôler. Elle s'appuie sur la littérature existante pour explorer différentes formes de guidage haptique informatif utilisant des dispositifs kinesthésiques et cutanés. Enfin, la thèse présente également des interfaces visuelles basées sur la réalité virtuelle qui permettent à l'utilisateur de mieux connaître la scène grâce à des marqueurs visuels, des couleurs et des objets augmentés.

Structure de la thèse

L'essentiel de cette thèse est divisé en trois parties principales. La première partie propose une revue des principales techniques utilisées dans les architectures de

téléopération et de contrôle partagé. Les deuxième et troisième parties présentent par contre les contributions originales de ce travail dans le contexte des architectures de contrôle partagées pour les tâches de télé-manipulation à distance. Deux systèmes sont considérés : (i) les manipulateurs fixes en série, à un ou deux bras et (ii) les robots humanoïdes. Les manipulateurs sont équipés d'une pince pour effectuer les opérations de manipulation nécessaires et d'une caméra pour observer la scène. Dans certains scénarios, la pince est installée sur un manipulateur et la caméra sur l'autre, tandis que dans d'autres cas, la pince et la caméra sont toutes deux installées sur le même manipulateur. D'autre part, le robot humanoïde utilisé est un robot à contrôle d'effort de 31 degrés de liberté équipé de deux bras humanoïdes. Les manipulateurs à base fixe sont très efficaces pour la télémanipulation d'objets dans des environnements contrôlés. Le projet RoMaNS H2020 est en réalité un exemple marquant d'un tel scénario où les déchets nucléaires provenant de conteneurs existants sont vidés sur une table devant le robot pour être triés et séparés. D'autre part, la mobilité est essentielle pour élargir l'espace de travail du robot, ou lorsqu'une intervention imprévue est nécessaire. La catastrophe de Fukushima au Japon est un témoignage regrettable d'un cas où la technologie robotique existante était incapable d'intervenir pour arrêter les noyaux du réacteur nucléaire et éviter les implications ultérieures.

Aperçu de la Partie I

Dans la première partie, une revue de l'état de l'art IHR est proposée. Nous classons la littérature en différentes catégories en fonction du niveau d'interaction entre l'opérateur humain et l'autonomie et donnons une brève description de chaque catégorie. La revue se concentre sur deux catégories principales : (i) le contrôle direct (par exemple, la téléopération) et (ii) le contrôle assisté (par exemple, les architectures de contrôle partagées et l'autonomie variable). Nous mettons en évidence les différentes approches ainsi que leurs avantages, inconvénients et limitations techniques. Cela nous amène naturellement à discuter de l'impact des capacités de perception modernes, de la puissance de calcul accrue et des architectures logicielles innovantes sur le terrain.

Aperçu de la Partie II

La deuxième partie présente les contributions principales de la thèse en lien avec la manipulation à un ou deux bras pendant les phases de manipulation avant, durant et après la saisie. Les travaux présentés dans cette partie ont été publiés et présentés dans [22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]. En particulier :

Au chap. 3 nous présentons deux modalités de contrôle partagé basées sur la vision pour permettre à un opérateur humain de commander un système à deux bras dans le but d'approcher et de saisir un objet cible. La première modalité donne à l'utilisateur un contrôle instantané sur un sous-ensemble du système DoF tout en permettant une autonomie pour commander le reste. La seconde modalité étend la première vers une architecture de planification partagée dans laquelle l'opérateur et l'autonomie collaborent pour modifier et optimiser les futures trajectoires des manipulateurs (au lieu de fournir des commandes instantanées). Dans les deux cas, un retour d'effort informe l'utilisateur de toute contrainte du système susceptible d'empêcher une exécution correcte des commandes de l'utilisateur, et le guide vers des positions plus sûres. Nous présentons enfin une série d'expériences réalisées pour tester et valider l'approche proposée ainsi qu'une étude utilisateur pour évaluer l'architecture de contrôle partagé proposée par rapport à la téléopération classique.

Au chap. 4 nous étendons les idées introduites dans le chap. 3 en mettant l'accent sur les différents aspects du contrôle partagé et en décrivant les généralisations potentielles des approches proposées vers les espaces de travail partagés, les environnements encombrés et les mécanismes de saisie avancés. Nous proposons également une architecture de contrôle partagé basée sur l'apprentissage pour l'ajustement dynamique en ligne de l'équilibre opérateur / autonomie en fonction de la confiance en l'autonomie lors de l'exécution de la tâche. Des expériences pertinentes et des études approfondies auprès des utilisateurs sont également présentées pour tester la validité des architectures proposées et l'impact du contrôle partagé sur les performances générales de l'opérateur humain.

Au chap. 5 nous abordons la phase de manipulation post-saisie, une phase qui est généralement ignorée dans les architectures de contrôle partagées. Nous proposons un nouveau système dans lequel l'autonomie aide un être humain à téléopérer un ensemble bras / pince esclave distant à l'aide d'un appareil maître haptique. Le système est conçu pour exploiter l'expertise de l'opérateur humain en matière de sélection des prises stables (toujours un sujet de recherche ouvert en robotique autonome). Pendant ce temps, un agent autonome transmet des signaux de force à l'homme, afin d'encourager la sélection de positions de saisie avec un maximum de manipulabilité. Nous montrons que le fait de suivre les indications entraîne une réduction significative de l'effort de contrôle du manipulateur, comparé à d'autres saisies réalisables, et démontrons l'efficacité de l'approche par des expériences avec des robots réels et simulés.

Aperçu de la Partie III

La troisième partie de ce travail aborde les problèmes qui se posent lors de la téléopération des bras doubles d'un robot humanoïde et des problèmes de stabilité et d'équilibre associés. Ce travail a été effectué de janvier à juin 2018 lors de la visite de recherche de l'auteur au Centre aérospatial allemand (DLR) à Oberpfaffenhofen et a été publié (ou est en cours d'examen) dans [33, 34].

Au chap. 6 nous présentons une interface haptique «pertinente pour la tâche» pour la téléopération humanoïde, qui comble la distance entre la tâche à accomplir et l'équilibre du robot. L'opérateur contrôle les mains de l'humanoïde et est informé par des signaux haptiques de l'impact de ses actions potentielles sur la stabilité du robot. De plus, un contrôleur autonome à espace nul agit dans l'espace nul de l'opérateur pour lui fournir un espace de travail plus large et faciliter l'exécution de la tâche. L'architecture est conçue pour compléter un contrôleur de conformité existant pour un robot humanoïde à contrôle d'effort. Des expériences sur le robot humanoïde TORO ont été rapportées pour démontrer la faisabilité et l'efficacité de l'approche.

Au chap. 7 nous traitons le problème de l'équilibrage des humanoïdes tout en effectuant des tâches d'interaction de force élevée, élément essentiel pour permettre à un opérateur humain d'interagir librement avec l'environnement lors de la téléopération d'un tel robot. Nous présentons une architecture qui étend un cadre d'équilibrage du corps entier basé sur la passivité pour garantir l'équilibre d'un robot humanoïde, tout en effectuant différentes tâches d'interaction où les forces (élevées) agissant sur le robot sont difficiles à prévoir. Au lieu de contrôler le centre de masse, le contrôleur proposé utilise directement les informations du cône de la force gravito-inertielle (Gravito-Inertial Wrench Cone) pour garantir la faisabilité des forces d'équilibrage. La performance de l'approche est validée par un certain nombre d'essais expérimentaux réussis.

Conclusions et annexes

Le chapitre 7 conclut la description des contributions principales de ce travail. En plus du contenu présenté jusqu'ici, la thèse contient également un chapitre de conclusion supplémentaire et une annexe.

Au chap. 8 nous fournissons un examen final des principaux résultats de la thèse en soulignant également quelques questions qui restent à résoudre. Nous proposons également un certain nombre d'extensions possibles du travail présenté dans

cette thèse qui mériteraient d'être étudiées. Certaines sont effectivement le sujet de l'activité de recherche de l'auteur.

Dans l'annexe A nous présentons des détails techniques supplémentaires pour la dérivation de certains des résultats contenus dans la thèse. Ce contenu n'est pas essentiel pour comprendre le reste de ce travail, mais il est néanmoins inclus ici par souci de complétude.

Contents

Chapter 1 Introduction **1**
1.1 Thesis Overview 4
1.2 Thesis Structure 6
1.2.1 Outline of Part I 7
1.2.2 Outline of Part II 7
1.2.3 Outline of Part III 8
1.2.4 Conclusions and Appendices 9
1.3 Thesis Publications 9

Part I Preliminaries and State of the Art **13**

Chapter 2 State of the Art **15**
2.1 Manual Control 18
2.1.1 Teleoperation 18
2.1.2 Telepresence 19
2.1.3 Haptic Interfaces 19
2.1.4 Teleoperation Limitations 21
2.1.4.1 Stability Issues 22
2.2 Supervisory Control 23
2.3 Assisted Control 24
2.3.1 Adaptability 25
2.3.2 Domains of Application 26
2.3.3 Partition of the Control Space 27
2.3.3.1 Shared Space 27
2.3.3.2 Partitioned Space 29
2.3.4 Human Authority vs Haptic Autonomy 29
2.3.4.1 Optimal Level of Autonomous Assistance 30

2.3.5 Control Level . . . 31
2.4 Collaborative Robotics . . . 32
2.5 Peer-to-Peer Collaboration . . . 33
2.6 Conclusions . . . 33

Part II Contributions: Shared Control of Serial Manipulators 37

Chapter 3 Pre-Grasping Phase 39
3.1 Problem Setting . . . 42
3.2 Instantaneous Shared-Control . . . 43
3.2.1 Gripper control . . . 43
3.2.2 Camera Control . . . 45
3.2.2.1 Point-Based IBVS . . . 46
3.2.2.2 Moment-Based IBVS . . . 48
3.2.3 Master Side . . . 52
3.2.3.1 Velocity-velocity control . . . 53
3.2.3.2 Position-Velocity Control . . . 54
3.2.4 Kinematic Constraints . . . 55
3.2.5 Passivity Analysis . . . 57
3.3 Trajectory-Based Shared Control . . . 60
3.3.1 Overview . . . 60
3.3.2 Trajectory Representation . . . 61
3.3.2.1 Definition of a Pose Trajectory . . . 61
3.3.2.2 Velocity Twist of a Pose Trajectory . . . 62
3.3.2.3 Final Considerations . . . 64
3.3.3 System Constraints . . . 65
3.3.3.1 Trajectory Length . . . 67
3.3.3.2 Final Cost Function . . . 69
3.3.4 The Overall Architecture . . . 69
3.3.4.1 Forward Interface . . . 70
3.3.4.2 Backward Interface . . . 73
3.3.4.3 Final Remarks . . . 74
3.4 Experimental Results . . . 76
3.4.1 Instantaneous Shared Control . . . 76
3.4.1.1 First Experiment . . . 77
3.4.1.2 Second Experiment . . . 78
3.4.2 Human-Subject Evaluation . . . 80
3.4.2.1 Experimental setup . . . 82
3.4.2.2 Experimental conditions and task . . . 82

3.4.2.3 Participants . . . 84
3.4.2.4 Results . . . 84
3.4.2.5 Discussion . . . 86
3.4.3 Shared Trajectory Planning . . . 88
3.4.3.1 Experimental Setup . . . 88
3.4.3.2 First Experiment . . . 90
3.4.3.3 Second Experiment . . . 90
3.4.3.4 Third Experiment . . . 93
3.5 Summary . . . 95

Chapter 4 Extensions **97**
4.1 Assisted Teleoperation in a Shared Workspace . . . 98
4.1.1 System Architecture . . . 99
4.1.1.1 System Modeling . . . 100
4.1.1.2 Constraints on the Slave Side . . . 100
4.1.1.3 Slave Control . . . 102
4.1.1.4 Haptic guidance . . . 104
4.1.2 Passivity Analysis . . . 105
4.1.3 Experiments in Simulation . . . 106
4.1.3.1 Experimental setup, task, and participants . . . 106
4.1.3.2 Experimental conditions . . . 108
4.1.3.3 Results . . . 109
4.1.4 Experiment in a Real Environment . . . 111
4.1.5 Discussion . . . 111
4.2 Grasping Assistance in a Cluttered Environment . . . 113
4.2.1 Overview on the Architecture . . . 113
4.2.2 Point Cloud and Grasp Pose Generation . . . 115
4.2.3 Haptic shared-control architecture . . . 116
4.2.3.1 Master/slave coupling . . . 116
4.2.3.2 Haptic guidance to avoid kinematic constraints . . . 117
4.2.3.3 Haptic guidance toward suitable grasping poses . . . 118
4.2.4 Experimental evaluation . . . 123
4.2.4.1 Experimental setup and task . . . 124
4.2.4.2 Representative repetition of the sorting task . . . 124
4.2.4.3 Experiment #1: pick and place of one object . . . 125
4.2.4.4 Experiment #2: pick and place in a cluttered scenario 128
4.2.5 Discussion . . . 130
4.3 Learning-Based Haptic Design in Shared Control Systems . . . 133
4.3.1 Related Work . . . 134

4.3.2 Modeling . . . 135
4.3.2.1 Demonstrated Trajectory Distributions . . . 136
4.3.3 Shared Control Architecture . . . 137
4.3.4 Trajectory Refinement through Interactive Task Executions . 138
4.3.5 Experimental Results . . . 140
4.4 Summary . . . 144

Chapter 5 Post-Grasping Phase 147
5.1 Related Works . . . 150
5.2 Problem Formulation . . . 151
5.3 Task-Oriented Velocity Manipulability . . . 151
5.3.1 TOV Definition . . . 152
5.3.2 TOV Gradient . . . 154
5.4 Haptic Feedback . . . 156
5.5 Experimental Results . . . 157
5.5.1 First Experiment . . . 157
5.5.2 Second Experiment . . . 158
5.6 Summary . . . 160

Part III Shared Control and Balancing of Humanoid Robots 163

Chapter 6 Humanoid Teleoperation 165
6.1 State of the Art . . . 166
6.2 Background . . . 168
6.2.1 Dynamic Model . . . 168
6.2.2 Underlying Compliance Controller . . . 169
6.3 Shared-Control Architecture . . . 171
6.3.1 Whole-Body Motion . . . 171
6.3.2 Master Side . . . 172
6.3.3 Balancing and Physical Constraints . . . 173
6.3.4 CoM Control Law . . . 175
6.3.5 Haptic Feedback . . . 176
6.4 Experiments and Results . . . 177
6.4.1 Experiment I . . . 178
6.4.2 Experiment II . . . 179
6.4.3 Experiment III . . . 179
6.5 Summary . . . 183

Chapter 7 Humanoid Balancing while Performing High-Force Interaction Tasks 185

7.1 Background and State of the Art . . . 186
7.2 Controller Design . . . 187
7.2.1 Force Polyhedron and Support Polygon . . . 188
7.2.2 Proposed Controller . . . 189
7.3 Contact Switching . . . 191
7.4 Experiments and Results . . . 193
7.4.1 Experimental Setup . . . 193
7.4.2 Experimental Results: Balancing . . . 194
7.4.2.1 Experiment 1. Carrying a Heavy Box . . . 194
7.4.2.2 Experiment 2. Table Pushing . . . 194
7.4.2.3 Experiment 3. Interacting with the Right Foot . . . 197
7.4.3 Experimental Results: Automatic Contact Switching . . . 197
7.5 Summary . . . 199

Part IV Conclusions and Future Work 203

Chapter 8 Conclusions 205
8.1 Part II: Shared Control for Serial Manipulators . . . 205
8.2 Part III: Humanoid Teleoperation and Balancing . . . 209
8.3 Final Considerations . . . 210

Appendix A Additional technical details associated to chapter 3 213
A.1 Proving the Validity and Orthogonality of the Basis Proposed in (3.7) 213
A.2 Derivation of the Orientation Jacobians in Sec. 3.3.2.2 . . . 214
A.2.1 The Orientation Jacobian w.r.t. Control Points Velocities . . 214
A.2.2 The Orientation Jacobian w.r.t. the Traveling Speed Along the Trajectory . . . 217

Bibliography 219

Chapter 1

Introduction

Contents

1.1 Thesis Overview 4
1.2 Thesis Structure 6
1.2.1 Outline of Part I 7
1.2.2 Outline of Part II 7
1.2.3 Outline of Part III 8
1.2.4 Conclusions and Appendices 9
1.3 Thesis Publications 9

The rise of modern-day robotics can be traced back to the late 1940s and the emergence of the nuclear industry [1]. The pressing need for handling dangerous radioactive material motivated Raymond Goertz to build the first known master-slave system at the Argonne National Laboratory in 1949 [2]. The master-slave mechanical (MSM) teleoperation system presented by Goertz could grasp and move objects in all six degrees-of-freedom (DoF) and employed a pure mechanical coupling between the master and the slave. During operation, instantaneous haptic feedback, resulting from the direct mechanical linkage, was received by the human operator behind her/his thick layer of lead glass. Few years later, the Commissariat à l'Énergie atomique et aux Énergies alternatives (CEA) lab, led by Jean Vertut, emerged as a major player in the field presenting significant contributions to MSMs as well [4, 5, 6]. In parallel, similar teleoperation systems were being employed in applications for particle accelerators [7, 8, 9]. In fact, MSM teleoperation systems proved very efficient over the years and dominated the nuclear industry for a long-time. Surprisingly enough, these systems are still playing a central role in the nuclear industry in our very day.

In the 1950s, few years after the introduction of the first MSMs, hydraulic actuation and electromechanical servomechanisms brought life to the manipulators we

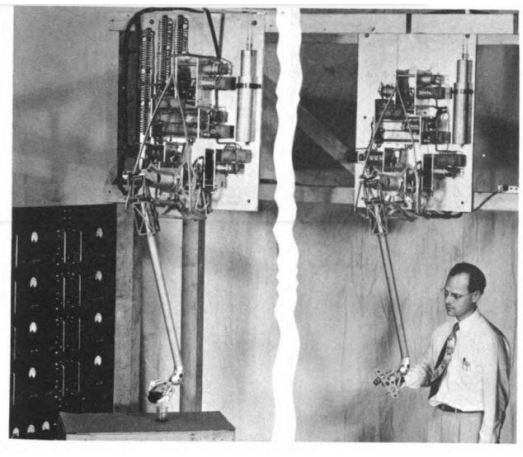
are more familiar with today. However, in the absence of 'computer-control', these manipulators were manually controlled by human operators without an overlaying logic. They were insensitive to their environment and the control action flew in one direction from the operator to the manipulator with no 'intelligent' feedback loops. In fact, this sheds light on the old art of 'Automata' blurring the lines of when robotics was actually born. Artisans have perfected complex mechanical systems and impressed the public with dazzling performances for thousands of years [10]. Was Jaquet-Droz's 18th-century 'automaton' writing "Je ne pense pas, ne serais-je donc pas point?"[1] a robot?

On the other hand, remotely operated devices have been around since quite some time as well. A boat "incorporating a borrowed mind", as Nicola Tesla described his invention, was presented to the public in 1898. The "borrowed mind", however, was actually Tesla himself as he commanded the boat remotely using radio signals. Several other types of remote-controlled vehicles followed and Tesla hypothesized, "...you see there the first of a race of robots, mechanical men which will do the laborious work of the human race." An 'electric dog' was later presented by the Naval Research Laboratory in 1923 followed by different remotely piloted vehicles or mechanical creatures along the same lines [13]. Again, non of the above had any 'intelligence' or serious computational capabilities.

Alongside these significant advances in machinery, digital computers have started to appear in the late 1930s and were directly used for controlling the early MSM

[1] "I do not think, do I therefore not exist?"

(a) The first master-slave mechanical system.



(b) An early electric master-slave system.

Figure 1.1: Early master-slave systems built at the Argonne National Laboratory. Similar systems are still in use in the nuclear industry. Photo courtesy [3].

Figure 1.2: "The writer": A surviving Jaquet-Droz's automaton from 1774 currently present at the Musée d'Art et d'Histoire of Neuchâtel, in Switzerland. Photo courtesy [11, 12].

manipulators. However, the 'interface' between the computers and the robot was the human operator himself who did the needed computations on the computer and used the results to manually command to the robot. Heinrich Ernst is arguably the first to have completely removed the human intermediary and given a computer full command over a manipulator [14]. The manipulator used for the experiments was one of the early electromechanical manipulators designed by Raymond Goertz and equipped with tactile sensing. The robot used the tactile sensors to autonomously search for a box on a table. It then searched for cubes on the table, grasped them, and placed them in the box. No previous knowledge of the position of the box or the cubes was needed and the robot could also adapt to changes in the box position during the experiment.

There is no need to stress the impact which computer-controlled systems had on all the aspects of the modern human society and the thousands of robotic applications which resulted from them. However, while robotic autonomy was satisfactory for some applications, it was not sufficient for others and the human intervention

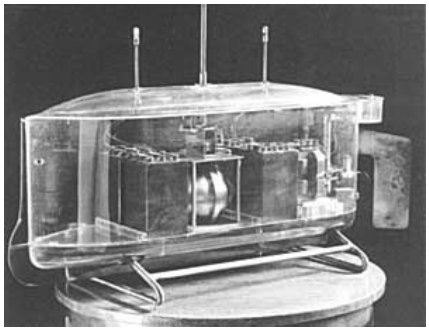


Figure 1.3: Tesla's Boat. Photo courtesy [13].

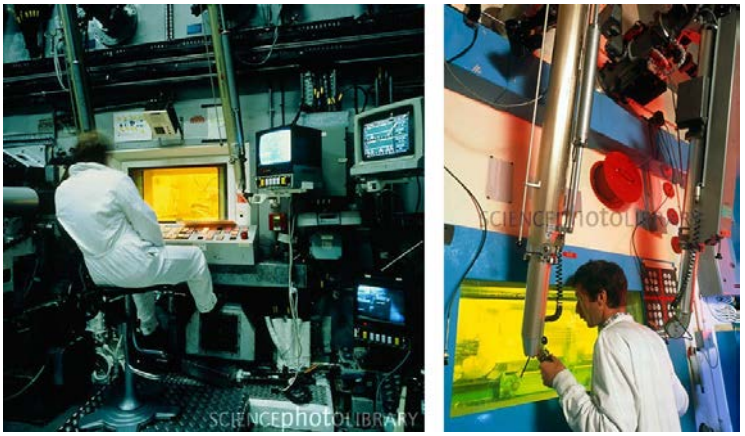


Figure 1.4: Current workspace of the human operator for the sort and segregate application. The operator has to look through a small window to operate the mechanical arms.

was needed. This soon gave rise to new approaches fusing the human intelligence with the precision and efficiency of autonomous systems in the form of human-robot shared-control architectures [1, 13, 15]. Early forms of Human-Robot Interaction (HRI) experiments, namely supervisory control, were reported by Ferral and Sheridan in [15] and followed by varied contributions ranging from advanced control theoretic methods to teleoperation-oriented software languages, visual enhancements and hybrid representations [16, 17, 18, 19]. The field evolved slowly over the years and was not established as an independent multi-disciplinary field until the 1990s [13].

1.1 Thesis Overview

This thesis takes inspiration from the rich history of HRI to analyse the relationship between the human operator and autonomous systems. Autonomous controllers are rapidly evolving as a result of advances in hardware capabilities (e.g., depth cameras, more efficient sensors, computational power) and software architectures (e.g., vision-based approaches, force control, machine learning, AI). This evolution drives a continuous shift of authority and dependency between the human operator and autonomy in HRI systems allowing for a more meaningful interaction between the two and a wider spectrum of applications. In this context, this thesis aims at proposing a variety of shared-control architectures for the different phases of remote telemanipulation characterized by different levels of autonomy for the robotic component and different ways of providing a feedback to the operator about the task execution and feasibility of her/his commands. Besides the general scientific interest in advancing the state-of-the-art in the shared control field, the work in this thesis is

also motivated by the the needs of the European H2020 "Robotic Manipulation for Nuclear Sort and Segregation" (RoMaNS) project[2], which has served as a concrete case study (and societal motivation) for the several proposed shared-control architectures. We then now proceed to briefly summarize the RoMaNS project for also providing some additional context to the thesis contributions.

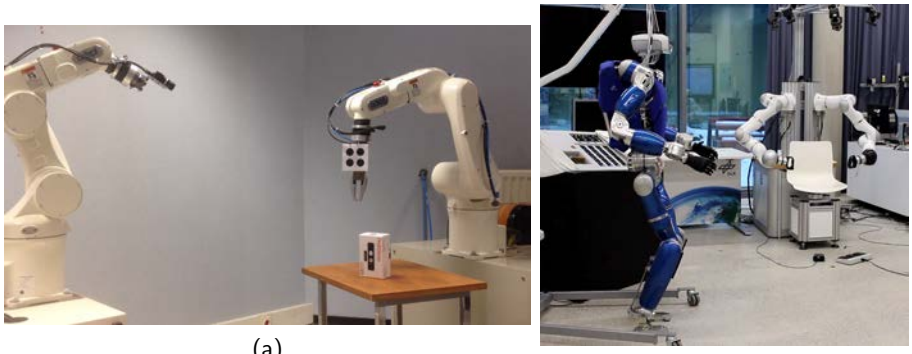
The consortium of the RoMaNS projects is consisted of five partners: University of Birmingham (UK; lead), Commissariat à l'Énergie atomique et aux Énergies alternatives (France), Technische Universität Darmstadt (Germany), National Nuclear Laboratory (UK), CNRS (France). The project tackles autonomous, teleoperative and shared control systems for remote manipulation of legacy nuclear waste. Cleaning up the past half century of nuclear waste, in the UK alone, represents one of the largest environmental remediation projects in Europe with 1.4 million cubic metres [20] of Intermediate Level Waste (ILW) to be processed. Some of this waste have been temporarily stored in containers, many of which have contents of mixed contamination levels, and sometimes unknown contents. These containers may need to be disrupted or cut open, to investigate their contents, before sorted and segregated. On the other hand, vast quantities of highly contaminated decommissioned plant machinery and infrastructure will have to be demolished, cut and resized, and treated as needed. Much of this work can only be done by remote manipulation methods, because the high levels of radioactive material are hazardous to humans.

Currently, it is proposed that the operators will sort and segregate using manual tele-operation, which in this context is simply using the standard teach pendant or a basic joystick to manually move the robot. There are associated safety, reliability and throughput concerns with this method because the operator is not in close proximity to the robot and must rely upon camera views and/or distorted views through lead glass windows (see Fig. 1.4).

Our contribution at CNRS (Irisa and Inria, Rennes) is notably on:

- Developing efficient visual tracking methodologies for controlling the manipulators using visual servoing approaches.
- Developing efficient haptic interfaces and haptic devices which can contribute to a more efficient task execution.
- Developing shared-control approaches with variable levels of autonomy that help the human operator and decrease her/his cognitive load.

[2] https://www.h2020romans.eu/



(a) (b)

Figure 1.5: Experimental Setup: (a) Two serial manipulators equipped with a gripper and a camera; (b) TORO, DLR's torque-controlled humanoid robot.

In particular, this thesis uses visual tracking and control software for proposing innovative visual-based solutions for advanced assisted telemanipulation. It focuses on three main aspects: (*i*) increasing the efficiency and speed of task execution, (*ii*) ensuring the safety and security of the system, and (*iii*) ameliorating the operator's experience and decreasing her/his cognitive load. To this end, the thesis develops semi-autonomous architectures that can assist the human operator in controlling parts of the system which can be cumbersome to manually/directly control. It builds on existing literature to explore different forms of informative haptic guidance using kinesthetic and cutaneous devices.

1.2 Thesis Structure

The core of this thesis is divided into three main parts. The first part proposes a review of the state-of-the-art and main techniques used in teleoperation and shared control architectures. The second and third parts present instead the original contributions of this work in the context of shared control architectures for remote telemanipulation tasks. Two systems are considered: (*i*) fixed single or dual-arm serial manipulators and (*ii*) humanoid robots. In the serial manipulators case, the robots are equipped with a gripper to perform the needed manipulation actions and a camera to observe the scene (see Fig. 1.2). In some scenarios, the gripper is installed on one of the manipulators and the camera on the other while in other scenarios the gripper and the camera are both installed on the same manipulator. On the other hand, the humanoid robot used is TORO from DLR, a redundant 31-DoF torque-controlled robot equipped with two humanoid hands (see Fig. 1.2).

Fixed-based manipulators are relevant in our context since they are highly efficient for telemanipulating objects in controlled environments. For instance, the

RoMaNS H2020 project offers a prominent example of such a scenario where nuclear waste from legacy containers is emptied on a table in front of the robot to be sorted and segregated. On the other hand, mobility can be essential for several reasons like the need for a wider workspace of the robot or where an unforeseen human intervention is needed. The Fukushima disaster in Japan is an unfortunate testimony of such a case where existing robotic technology was unable to intervene as needed for shutting down the cores of the nuclear reactor and avoiding the subsequent implications [21]. In such cases, the use of a highly articulated "mobile manipulator" such as a humanoid robot is more appropriate, thereby motivating the shared control methods proposed in the thesis for interfacing a human operator with a humanoid robot.

1.2.1 Outline of Part I

In the first part, a review of the state of the art of HRI is proposed. We classify the literature into different categories depending on the level of interaction between the human operator and autonomy and give a brief description of each category. The review focuses on two main categories: (*i*) Direct Control (e.g., teleoperation) and (*ii*) Assisted Control (e.g., shared control architectures and variable autonomy). We highlight the different approaches along with their advantages, disadvantages, and technical limitations. This naturally leads us to discussing the impact of modern sensing capabilities, increased computational power, and innovative software architectures on the field.

1.2.2 Outline of Part II

The second part presents the main contributions of the thesis in the context of single and dual serial manipulator arms during the pre-grasp, grasp and post-grasp phases of manipulation. The work presented in this part has been published and presented in [22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]. In particular:

In Chap. 3 we present two visual-based shared-control modalities for allowing a human operator to command a dual-arm system with the objective of approaching and grasping a target object. One of the manipulators is equipped with a gripper while the other is equipped with a camera. The first modality gives the user instantaneous control over a subset of the system DoF while allowing autonomy to command the rest (using visual information from the camera). The second modality extends the first one towards a shared-planning architecture in which the operator and autonomy collaborate to modify and optimize the future trajectories of the manipulators (instead of providing instantaneous commands). A simulator

is used to to visualize future trajectories and increase the user's awareness of the scene through visual markers, colors and augmented objects. In both modalities, an informative haptic feedback informs the user about any system constraint that may prevent a correct execution of the user's commands. It also guides the user towards safer configurations. We finally present a set of experiments performed to test and validate the proposed approach along with a user study for benchmarking the proposed shared-control architecture against classical teleoperation.

In Chap. 4 we extend the ideas introduced in Chap. 3 towards shared workspaces and cluttered environments. We combine the haptic guidance with shared-control algorithms for autonomous orientation control and collision avoidance meant to further simplify the execution of grasping tasks. Moreover, while the model of the target object was assumed to be known in Chap. 3, we hereby employ a depth camera to retrieve a point cloud of the scene. The point cloud is then used along with an autonomous grasping algorithm to assist the user towards choosing potentially feasible grasp candidates. On the other hand, we also propose a learning-based shared-control architecture for the online dynamic adjustment of the operator/autonomy balance in function of the confidence of autonomy during the task execution. Relevant experiments and extensive user studies are also presented to test the validity of the proposed architectures and the impact of shared-control on the general performance of the human operator.

In Chap. 5 we tackle the post-grasp phase of manipulation, a phase which is usually ignored in shared-control architectures. We propose a novel system in which autonomy assists a human operator in teleoperating a remote slave arm/gripper using a haptic master device. The system is designed to exploit the human operator's expertise in selecting stable grasps (still an open research topic in autonomous robotics). Meanwhile, an autonomous agent transmits force cues to the human, to encourage maximally manipulable grasp pose selections. We show that following the cues results in significantly reduced control effort of the manipulator, compared to other feasible grasps and demonstrate the effectiveness of the approach by experiments with both real and simulated robots.

1.2.3 Outline of Part III

The third part of this work addresses the issues which arise when teleoperating the arms of a humanoid robot like stability and balance. This work described in this part was conducted between January to June of 2018 during the author's research visit at the German Aerospace Center (DLR) in Oberpfaffenhofen and has been published (or is under review) in [33, 34].

In Chap. 6 we present a 'task-relevant' haptic interface for humanoid teleoperation, which bridges the gap between the task at hand and the balance of the robot. The operator is given command over the humanoid's hands and is informed through haptic cues about the potential impact of her/his actions on the robot stability. Moreover, a null-space autonomous controller acts in the operator's null-space to provide her/him with a wider workspace and help in the successful execution of the task. The architecture is designed to top an existing compliance controller for a torque-controlled humanoid robot. Experiments on the humanoid robot TORO are reported to demonstrate the feasibility and effectiveness of the approach.

In Chap. 7 we tackle the issue of humanoid balancing while performing high-force interaction tasks, an essential building block for allowing a human operator to interact freely with the environment when teleoperating such a robot. We present an architecture which extends a passivity-based whole-body balancing framework to guarantee the equilibrium of a humanoid robot while performing different interaction tasks where the (high) task forces acting on the robot are difficult to foresee. Instead of controlling the center of mass, the proposed controller directly uses information from the Gravito-Inertial Wrench Cone to guarantee the feasibility of the balancing forces. The performance of the approach is validated in a number of successful experimental tests.

1.2.4 Conclusions and Appendices

In addition to the content outlined so far, the thesis also contains an additional conclusive chapter and an appendix.

In Chap. 8 we provide a final overall review of the main results of the thesis while also highlighting some open issues that still remain to be solved. We also propose a certain number of possible extensions to the work presented in this thesis that could be worth investigating. Some of them, are, indeed the subjects of the author's current research activity.

In Appendix A we include some additional technical details for the derivation of some of the results contained in the thesis. This content is not essential to understand the rest of this work, but it is nevertheless included here for completeness.

1.3 Thesis Publications

- F. Abi-Farraj, N. Pedemonte, and P. Robuffo Giordano, "A visual-based shared control architecture for remote telemanipulation," in *Proc. IEEE/RSJ Int.*

Conf. on Intelligent Robots and Systems, Deajeon, Republic of Korea, 2016.

- N. Pedemonte, F. Abi-Farraj, and P. R. Giordano, "Visual-based shared control for remote telemanipulation with integral haptic feedback," in *Proc. IEEE Int. Conf. Robotics and Automation*, Singapore, 2017.

- F. Abi-Farraj, T. Osa, N. Pedemonte, J. Peters, G. Neumann, and P. Robuffo Giordano, "A learning-based shared control architecture for inter- active task execution," in *Proc. IEEE Int. Conf. Robotics and Automation*, Singapore, 2017.

- A. Ghalamzan, F. Abi-Farraj, P. Robuffo Giordano, and R. Stolkin, "Human-in-the-loop optimisation: mixed initiative grasping for optimally facilitating post-grasp manipulative actions," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Vancouver, Canada, 2017.

- M. Selvaggio, F. Abi-Farraj, C. Pacchierotti, P. R. Giordano, and B. Siciliano, "Haptic-based shared-control methods for a dual-arm system," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4249-4256, Oct 2018.

- F. Abi-Farraj, B. Henze, A. Werner, M. Panzirsch, C. Ott, and M. A. Roa, "Humanoid Teleoperation using Task-Relevant Haptic Feedback," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Madrid, Spain, Oct. 2018.

- F. Abi-Farraj, C. Pacchierotti, and P. Robuffo Giordano, "User evaluation of a haptic-enabled shared-control approach for robotic telemanipulation," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Madrid, Spain, 2018.

- F. Abi-Farraj, B. Henze, , C. Ott, P. Robuffo Giordano, and M. A. Roa, "Torque-based balancing for a humanoid robot performing high-force interaction tasks," *IEEE Robotics and Automation Letters*, Under consideration for publication.

- F. Abi-Farraj, C. Pacchierotti, O. Arenz, G. Neumann, and P. R. Giordano, "A haptic shared-control architecture for guided robotic grasping in unknown environments," *IEEE Transactions on Haptics*, Under consideration for publication.

- F. Abi-Farraj, R. Spica, and P. Robuffo Giordano, "Trajectory-Based Shared Control for TeleManipulation," in *IEEE Transactions on Control Systems Technology*, Under consideration for publication.

- F. Abi-Farraj, R. Spica, and P. Robuffo Giordano, "Trajectory-based shared control with integral haptic feedback," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems Workshop: Human in-the-loop robotic manipulation: on the influence of the human role*, Vancouver, Canada, Oct. 2017.

- F. Abi-Farraj, R. Spica, and P. Robuffo Giordano, "Trajectory-based shared control with integral haptic feedback," in *10th Int. Workshop on Human Friendly Robotics*, Naples, Italy, Nov. 2017.

- F. Abi-Farraj, C. Pacchierotti, and P. Robuffo Giordano, "Human-subject evaluation of shared-control approaches for robotic telemanipulation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems Workshop: Human in-the-loop robotic manipulation: on the influence of the human role*, Vancouver, Canada, Oct. 2017.

- F. Abi-Farraj, C. Pacchierotti, and P. Robuffo Giordano, "Human-subject evaluation of shared-control approaches for robotic telemanipulation," in *10th Int. Workshop on Human Friendly Robotics*, Naples, Italy, Nov. 2017.

Part I

Preliminaries and State of the Art

Chapter 2

State of the Art

Contents

2.1 Manual Control **18**
2.1.1 Teleoperation 18
2.1.2 Telepresence 19
2.1.3 Haptic Interfaces 19
2.1.4 Teleoperation Limitations 21
2.2 Supervisory Control **23**
2.3 Assisted Control **24**
2.3.1 Adaptability 25
2.3.2 Domains of Application 26
2.3.3 Partition of the Control Space 27
2.3.4 Human Authority vs Haptic Autonomy 29
2.3.5 Control Level 31
2.4 Collaborative Robotics **32**
2.5 Peer-to-Peer Collaboration **33**
2.6 Conclusions **33**

Automation is the process by which a particular system is controlled using sensory information without any human assistance. From the household thermostat controlling a boiler to large industrial control systems with thousands of sensor measurements and output control signals, autonomy is an essential component of every aspect of modern life with a very wide range of applications. In robotics, autonomy has revolutionized dummy mechanical systems and equipped them with the long sought 'logic' in forms of basic or highly complicated feedback-control loops. However, while advancements in robot autonomy

have flourished in structured environments, serious limitations have hindered a similar progress in unstructured settings where the presence of a human operator is often still necessary.

Human-Robot Interaction (HRI), as defined by Goodrich and Schultz [13] is "a field of study dedicated to understanding, designing, and evaluating robotic systems for use by or with humans." In fact, even the most autonomous systems are operated by humans and used to fulfil a certain human need which makes a minimal interaction between the human and the machine indispensable. However, HRI is more concerned with the cases in which a continuous interaction between the human and the machine is needed.

Different robotic applications require different forms of interaction which can range from social interaction to physical interaction, remote control, assistive-control or supervisory control. Social HRI can be distinguished from other forms of HRI in that it focuses on the human experience itself rather than the efficient fulfilment of the task at hand. It is more concerned with the cognitive, social and emotional aspects of the interaction. However, as the focus of the thesis is on the technical rather than social aspect of the interaction, social HRI is omitted from the presented review.

With autonomy being a mean in HRI and not a goal in itself, the level of autonomy varies widely from one application to another. Different descriptions of the levels of autonomy have been proposed in literature. The most widely cited description is by Tom Sheridan [35]. Sheridan's scale ranges from the robotic system being fully commanded by an operator to being completely autonomous. The different levels are defined as follows:

1. Computer offers no assistance; human does it all.
2. Computer offers a complete set of action alternatives.
3. Computer narrows the selection down to a few choices.
4. Computer suggests a single action.
5. Computer executes that action if human approves.
6. Computer allows the human limited time to veto before automatic execution.
7. Computer executes automatically then necessarily informs the human.
8. Computer informs human after automatic execution only if human asks.
9. Computer informs human after automatic execution only if it decides too.

10. Computer decides everything and acts autonomously, ignoring the human.

Several variations of this scale have been proposed by different authors [36] while others argued that it is most useful when applied to each subtask within a problem domain but not on the entire problem domain itself [37]. However, and while the scale gives a significant insight on how autonomous a robotic system is, the level of interaction between the machine and the human operator is of more significance in HRI. On one extreme, the operator is in full control of the system with no input from autonomy and very limited human-robot interaction. On the other extreme, the system is fully autonomous and interacting with the human operator on a peer-to-peer basis. Inspired from the scale presented in [13], we classify the different HRI approaches as follows:

- **Manual Control** – The human is in full control of the robot DoF with no autonomy present in the robotic system (e.g., basic 6-DoF master-slave manipulators).

- **Supervisory Control** – The human is not in control of particular DoF of the system. She/He picks and chooses from a set of pre-defined subtasks which the system executes autonomously (e.g., Mars Rovers are controlled in a supervisory control mode because of the large time delay between the Earth and Mars preventing any instantaneous communication).

- **Assisted Control** – The human operator can be in partial or full control of the system but she/he is assisted by autonomy throughout the task execution (e.g., an operator commanding a fleet of UAVs where the particular position of each is decided by autonomy while the general behaviour of the fleet is commanded by the operator).

- **Collaborative Robotics** – The robotic system has the capabilities for a nearly-autonomous task execution but follows the lead of the human operator in a high-level master-slave hierarchy. Unlike supervisory control, the operator is here fully immersed in the task execution hand-in-hand with the autonomous system (e.g., a robot assisting a human operator in carrying a heavy load).

- **Peer-to-peer collaboration** – The system is fully autonomous and minimally operated by a human (Turning it on and off, maintenance, and so on). The interaction between the operator and the fully autonomous machine may be on a peer-to-peer basis.

(a) (b)

Figure 2.1: (a) Position/Velocity controlled manipulators placed in a cage and teleoperated by a human operator. Photo courtesy [38]. (b) A human operator interacting physically with a lightweight robot. Photo courtesy [39].

We will now provide a summary of the state of the art of each of the described categories with more elaboration on Shared Control approaches which are the focus of the thesis.

2.1 Manual Control

Manual Control refers to robotic systems which are fully controlled by a human operator. While the command-feedback loop between the operator and the machine is continuous in this scenario, the dynamic interaction between the two is limited because of the lack of autonomy of the machine which operates in a totally passive manner. The most prominent example of direct control is Teleoperation or the operation of vehicles, manipulators and other machines at a distance. However, the notion of 'distance' here is flexible. Historically, robots working in close proximity to humans have been separated from them with cages or other forms of barriers for safety concerns. It can be thus argued that all manually-controlled robots were 'teleoperated' using some form of input device or teach pendant. This changed in recent years especially with the introduction of lightweight manipulators which brought robots out of their cages (Fig. 2.1) and allowed for direct physical interaction between the robot and the operator. Such manipulators can be manually controlled as well and may fit under this label.

2.1.1 Teleoperation

As discussed in chapter 1, the need to manipulate nuclear waste without exposing a human operator to radioactivity was the major drive behind modern-day robotics. Other early motives included remotely piloting bombers during World War II and

the desire to create remotely controlled vehicles and mechanical creatures for entertainment purposes. The development of proper communication channels for remote control and the human desire to explore dangerous and far-reached territories allowed teleoperation systems to spread rapidly to new fields such as space [40, 41] and deep ocean exploration [42, 43].

Teleoperation is also central in miniature manipulation for micro robotics and medical applications. Minimally invasive surgeries revolutionized the medical field allowing for new efficient procedures which drastically decreased the recovery time of the patients and decreased the complexity of the operations for the surgeons. One of the best known surgical systems is the da Vinci Robot. It scales the surgeon's actions over a very small communication-delay enhancing the performance for a variety of laparoscopic surgeries [44].

2.1.2 Telepresence

Telepresence is a main aspect of teleoperation were the operator receives audiovisual and, usually, haptic feedback from the remote site. While a certain level of telepresence is needed for any long-distance manipulation to be feasible, telepresence may be an end in itself as well. Different applications include teleconferencing, search and rescue missions, intelligence, surveillance-related missions and many others. While a basic camera-monitor combination creates some sense of telepresence, more sophisticated systems are sought to provide the user with a better experience. The teleoperated robot can, for example, follow the head movements of the user and provide her/him with visual feedback over simple monitors, a head-mounted display or a more complex telepresence system. Similarly, audio feedback can be provided to the user through a regular speaker or a stereo sound system depending on the operator's need for the audio feedback and its value for the task at hand. Experiments on teleoperating a mine drill machine for example proved that an accurate transmission of the sound is extremely valuable in such applications [44]. On the other hand, haptic feedback is also believed to play an important role in transmitting a sense of presence to the human operator. The field has taken a lot of attention since the emergence of early teleoperation systems and several haptic interfaces have been proposed.

2.1.3 Haptic Interfaces

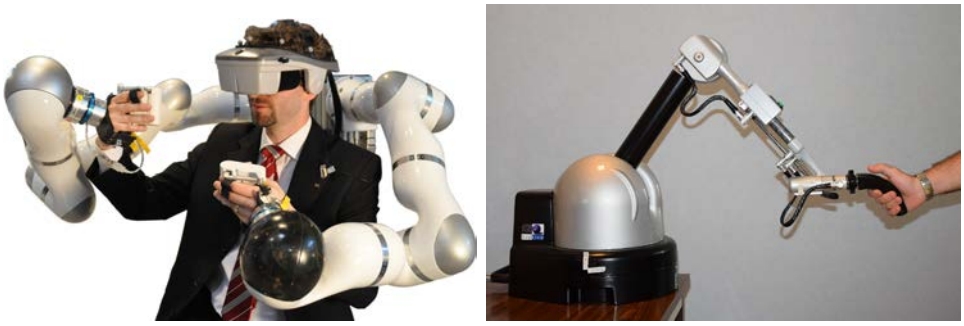
Haptic feedback is provided to the human operator through kinesthetic and cutaneous stimuli. Kinesthetic stimuli provide humans with information about the applied forces and torques as well as the position and velocity of neighbouring objects (through sensing collisions). They are detected by means of receptors in muscles

and joints like muscle spindles, which transduce muscle stretch, and Golgi tendon organs, which sense change in muscle tension [45, 46]. On the other hand, cutaneous stimuli are felt by mechanoreceptors in the skin and provide the user information about the local properties of objects such as shape, edges, and texture. The human brain analyses measures of the location, intensity, direction, and timing of contact forces on the fingertips for exploration and manipulation purposes [47, 48]. A wide variety of haptic devices have been proposed over the years for providing the user with the desired haptic experience of the remote environment. They range from the more conventional kinesthetic and cutaneous devices [47] to the less popular touchable [49] or even ultrasound [50] variations.

Kinesthetic haptic devices represent the bulk of haptic research in terms of device design and rendering algorithms. A kinesthetic haptic device provides a force at its end-effector by applying forces and torques over its joints. The provided force can be a full 6 DoF force tensor or a simple 1 DoF force depending on the design of the device. Fig. 2.2 depicts two examples of a dual-arm and a single-arm kinesthetic haptic devices. In a teleoperation scenario, kinesthetic feedback is used to block or influence the motion of the human operator by mapping the haptic interactions of the slave manipulator with the environment (ex. collisions and friction). This information is valuable for the exploration and manipulation of the remote environment. Moreover, while kinesthetic haptic feedback focuses on the forces the operator is receiving in her/his muscles and joints, it also stimulates the skin through the held tool, effectively providing cutaneous feedback in addition to the actively controlled kinesthetic feedback. In addition to the kinesthetic and implicit cutaneous stimuli, vibratory stimuli can be rendered through kinesthetic devices as well to reflect the vibrations we feel during collisions for example or to reflect the urgency of the message or the identity of the sender [51].

Pure cutaneous feedback is based on applying a shear or vibratory force on the user's skin to induce skin deformations. The idea is to mimic the skin deformations that naturally occur during haptic interaction especially on the finger tips which are more sensitive to shear forces than to normal forces [54]. Cutaneous haptic devices vary but are usually wearable and focused on the finger tips of the hand. Other cutaneous devices were proposed for other parts of the body like the wrist and the forearm [55], the back [56] or the legs [57]. However, in a classical teleoperation approach, finger tips cutaneous devices (see Fig. 2.3) are the most popular [58, 47, 51].

In addition to teleoperation, haptic interfaces can be used in Virtual Reality (VR) and Augmented Reality (AR) applications to enhance the virtual experience. Indeed, the immersive experience is missing if the user doesn't receive haptic stimuli when manipulating objects in the virtual environment. Moreover, the rich possi-



(a) (b)

Figure 2.2: Examples of kinesthetic haptic devices. (a) HUG, a dual-arm kinesthetic haptic device built by DLR. Photo courtesy [52]. (b) Virtuous 6D, a 6-DoF haptic device by haption [53].

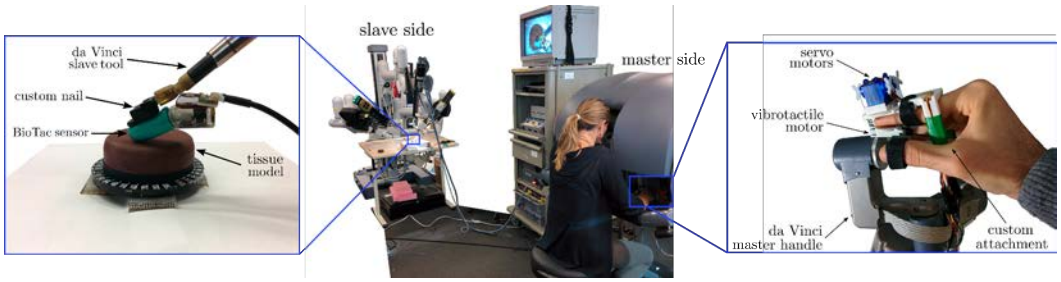


Figure 2.3: Cutaneous Feedback of Fingertip Deformation and Vibration tested for Palpation in Robotic Surgery with the da Vinci robot. Photo courtesy [59].

bilities of visual rendering can be used to enhance the haptic experience in such scenarios [51]. Visuo-haptic illusions seek to use the dominance of the visual experience and haptic illusions to enhance the haptic experience itself. It has been used to augment the cutaneous haptic feedback for increased stiffness rendering [60], to alter the perception of rotational alignment [61] or to influence the haptic perception of bumps and other features on a surface [62].

2.1.4 Teleoperation Limitations

Teleoperation is a field which dates back to the foundation of robotics itself and a massive amount of literature has been written on the subject tackling the different issues which face teleoperation systems and proposing solutions and novel ideas for more robust and efficient teleoperation and, especially, haptic rendering. For general reading on the topic, we refer the reader to the several surveys available in the literature and notably [63] and [64].

There are several design parameters that can be taken into account when designing a teleoperation system and a trade-off is often necessary. Typical objectives

are robustness, feeling of presence, task performance, and stability.

2.1.4.1 Stability Issues

The main control issue in bilateral teleoperation is the stability of the closed-loop system despite unmodelled dynamics and faulty communication channels. In designing the bilateral controller, a classic engineering tradeoff between transparency and stability is unavoidable, since transparency must often be reduced in order to guarantee stable operation in the wide range of environment impedances (for example, in terms of stiffness of "free space" and "hard contact"). Enforcing stability has actually motivated the development of many different control schemes over the past decades. A thorough survey of control solutions proposed in literature to counteract these issues (dating back to 2006) is available in [63] and summed up nicely in [44].

Among the many approaches to teleoperation stability, passivity-based approaches have become very popular over the last decades. Early approaches used passivity ideas for the stable control of force-feedback-enabled teleoperation systems with time delay [65, 66] and for deriving fixed parameter virtual couplings [67]. Passivity is a sufficient condition for assuring a stable teleoperation system and it has some attractive features:

- it employs intuitive energy concepts: a system is passive if and only if the energy flowing out of the system is less than the energy flowing in at all time.
- individual passive blocks can be easily composed in different ways for obtaining a guaranteed global passive system.
- it applies to linear and non-linear systems.
- it can be assumed that the human operator is a passive system at frequencies of interest, which, under the assumption of a passive environment, allows to conclude stability of the complete closed-loop teleoperation system.

Later in [68], the authors proposed a control scheme based on the wave variables where they introduced an energy input/output balance monitoring mechanism which limits the energy that the system can generate. The concept of 'energy tanks' have since been popular for assuring the stability of the teleoperated system [69, 70, 71, 72]. The time domain definition of passivity have also been exploited for ensuring stability [73, 74, 75, 76]. In [73], an adaptive virtual damping is introduced to satisfy the passivity constraints. The framework is extended in [74] to a 2-port network, and the issues in implementing the passivity observer (PO) and passivity controller (PC) to teleoperation systems are studied. A more

accurate PO/PC approach is proposed in [75] after introducing a new sampled-time definition of passivity.

Lately, a general and flexible framework, the passive set-position modulation (PSPM) approach, has been presented in [77]. The framework allows for connecting a (continuous-time) multi-DOF nonlinear robotic system to a sequence of (discrete-time) set-position signal via a simple spring coupling with damping injection. The algorithm modulates the original set-position signal in such a way that the modulated signal is as close to the original signal as possible (i.e., maximum information recovery for better performance), yet only to the extent permissible by the available energy in the system (i.e., passivity constraints).

A major limitation in teleoperation is also the hardware itself. The availability of convenient high-quality sensors on the slave side and the presence of efficient master devices capable of mapping the sensory information to the master is crucial. For example, tactile sensors are still very sensitive, expensive and limited in capabilities. Moreover, limitations on the master side are even more significant especially for haptic rendering. While kinesthetic haptic devices are relatively efficient, rendering tactile sensing on the human skin is tricky. Some devices are available for local rendering of tactile sensing (on finger-tips for example) but a fully immersive haptic experience is still at far-reach.

2.2 Supervisory Control

Supervisory control or human supervisory control is a form of high-level teleoperation where the master and the slave are loosely connected. The machine at the slave side exhibits a certain level of autonomy and is capable of carrying a limited series of actions on its own. It sends sensory information back to the human operator who updates/modifies the instructions as necessary. The level of autonomy of the slave side in such scenarios is quite high and the interaction with the human operator is limited. Supervisory control is common in performing routine tasks like handling parts on manufacturing assembly lines and accessing and delivering packages, components, mail, and medicines in warehouses, offices, and hospitals [1]. It is also common in remote environments like space where direct control is not possible due to long time delays. Examples include early successes of the Soviet Lunokhods [43] and later experimentation with ROTEX on-board spacelab D2 on shuttle COLUMBIA [78] and NASA's Mars rovers [79, 80].

A significant volume of the literature on supervisory control dates back to the 1980s and early 1990s. At the time, supervised robots were commanded by updating the computer commands controlling them online using a specific programming language. To this end, several authors proposed teleoperation-oriented software lan-

Figure 2.4: The tablet interface used in the SUPVIS Justin experiment. A rich graphical interface with augmented reality overlays allow for an easy and user-friendly experience of supervisory teleoperation. Photo courtesy [83].

guages to facilitate the task for the operator [16, 81]. Other enhancements tackled visual interfaces using predictive display [82], advanced control methods such as Lyapunov-based analysis [17] and hybrid representation [18].

Lately, the availability of richer and advanced graphic tools allowed for more user-friendly supervisory control interfaces. A recent example is DLR and ESA's METERON project where a wheeled humanoid robot, Rollin' Justin, is controlled by astronauts in the International Space Station to perform some operations like wiping a solar panel or plugging and unplugging cables. A tablet interface (Fig. 2.4) was used for the experiments allowing for a richer experience with easy and straightforward operation modes [83].

2.3 Assisted Control

Assisted control encompasses a wide spectrum of control techniques including virtual fixtures, haptic shared control, semi-autonomous systems, shared control and human-to-multi-robot systems. An interesting metaphor explaining assisted control is proposed in [84]. The metaphor is the following. "Imagine to be riding a bike and not knowing where to go. In order to look at the map, you need to stop. Would it be possible to do both actions (riding and reading) and the same time? Yes, if riding a Horse." A horse can avoid obstacles, keep balance and maintain a sense of direction even if the rider is not looking. Different forms of autonomous assistance exist. In a typical scenario, the user is operating a robotic system and can receive assistance in the form of:

- Visual feedback (e.g. proposing alternative solutions or highlighting a con-

straint).

- Haptic feedback (e.g. repulsive haptic cues when approaching a constraint).
- Auditory feedback (e.g. beeps of different magnitude and nature informing about particular events).
- Assistance in the control (e.g. The control of the system DoF is divided between the operator and an autonomous controller).

Different nomenclatures can be found for similar assisted control approaches and this may lead to confusion. However, important distinctions between the different architectures can be made based on their adaptability, their application domains, the partition of the control space and the sought control-level.

2.3.1 Adaptability

We characterize the adaptability of the system based on its ability to react to changes in the environment and to adapt its behaviour by interpreting the intention of the operator. Along these lines, we define three main categories:

- Static Fixtures: A previous knowledge of the environment and the constraints of the system is assumed and the system is designed according to that knowledge. The fixtures are insensitive to changes in the environment or in the operator's behaviour. See [85, 86, 87].
- Dynamic Assistance: These systems rely on rich sensory information (from vision-like sensors for example) to react actively to unexpected and dynamic changes in the environment. However, they are also insensitive to the operator's intentions. See [88, 89, 90, 91].
- 'Intelligent' Assistance: Unlike the previous two categories, these systems exhibit a form of 'intelligence' to interpret the intention of the operator depending on her/his actions and gestures and adapt the form/level of the provided assistance accordingly. See [92, 93, 94].

While identifying these categories is important for understanding the different components of an assistive system, most of the architectures proposed in literature are a mix of two or more. As an example to demonstrate the differences in the proposed characterization, we consider a human operator teleoperating a serial manipulator. She/He may be guided to avoid the kinematic limitations of the manipulator and to stay away from potential self-collisions. In this case, both of the described constraints are static fixtures which are 'self-contained' in the model of

the system itself. If the user is, however, to be also guided away from unknown dynamic obstacle in the robot's workspace, a vision-like sensor and additional dynamic assistance needs to be incorporated in the system. On the other hand, the attitude of the human operator may also carry important information regarding her/his intentions. For example, a firm grasp of the master device may imply that the operator desires more control of the system while a loose grasp may be an indication of hesitation or that autonomy must take control. Enriching the system with an 'intelligent' assistance capable of interpreting such gestures can be very valuable for the assisted control architecture.

2.3.2 Domains of Application

Assisted control systems have been applied to a wide variety of application domains which range from teleoperating serial manipulators [95, 96, 97, 98, 99, 85, 100, 101, 92, 102] to controlling wheelchairs [103, 104] or walkers [105], surgical tools [106, 107, 108], vehicle guidance [109, 84, 110, 111], quadrotor UAVs [112, 113, 114], and mobile robots [115, 116].

Teleoperation : Teleoperating manipulators in 6-DoF space is not an easy task and has been a central topic in assistive control systems [95, 96, 97, 98, 99, 85, 100, 101, 92, 102], and, indeed, it is also one of the main points addressed in this thesis. To begin with, it is complicated to make good judgement of depth when looking at the remote environment from behind standard monoscopic screens [117]. Moreover, the simultaneous coordination of rotational and translational motions in a 6-DoF environment is cognitively demanding. In this respect, it has been observed that humans tend to heavily rely on translations when given command of a 6-DoF robotic system [118, 119]. Rotations are usually overlooked and avoided except if utterly necessary. The same studies also observe an 'incremental' behavior when the operator needs to actuate a rotational motion: users actuate one rotation direction at a time instead of controlling all three rotations together as they usually do when commanding translations. This behavior is in line with interesting psychological studies arguing that human subjects are incapable of mentally rotating objects in 3D space [120].

Several assisted control systems for manipulators have been proposed to facilitate the task for the human operator. Applications range from grasping [95, 121, 122, 123] to peg-in-hole [124], bolt-and-spanner [96, 97], circular saw [98], reciprocating saw [99] or powered socket tools [99].

2.3.3 Partition of the Control Space

The partition of the control space between autonomy and the human operator is a sensitive and crucial topic in assisted control systems. We identify two main categories:

- **Shared Space** where the operator and autonomy command the same DoF of the system. The commands of the human operator may be modified or overridden by autonomy and vice versa.
- **Partitioned Space** where the human operator is given control over a particular set of the system DoF while autonomy commands the rest. No interference takes place between the two.

2.3.3.1 Shared Space

The most popular shared space approaches are potential fields and virtual fixtures. Potential fields are defined either as repulsive force fields (potential hills - in order to avoid objects), or attractive force fields (potential wells - in order to direct the user to determined target regions). Controllers based on potential field architectures were proposed by [85, 124, 86, 87, 125] and covered a variety of applications.

In [85], for example, the author proposed a system for online obstacle and kinematic limitations avoidance. The operator receives repulsive haptic cues generated from potential fields encoding the constraints of the system to stay away from dangerous configurations. In [124], potential fields are used to generate autonomous velocity commands which are fed to the system alongside the human operator's commands within a shared-control framework. They are also used in [86] to assist a surgeon for avoiding obstacles and limiting the robot's workspace. In this work, the surgeon is also guided towards pre-defined geometric fixtures like a curve or a surface.

This takes us to virtual fixtures which were initially proposed in [100]. Virtual fixtures are defined from sensory information and have been used for teleoperation or/and training systems. In [100], the authors described the benefits of virtual fixture in teleoperation by comparing them to the common physical fixture of a ruler: "Like a ruler guiding a pencil in the real world, virtual fixtures are intended to reduce mental processing required to perform remote tasks, reduce the work load of certain sensory modalities, and most of all allow precision and performance to exceed natural human abilities." Several works proved the improvement in performances due to the introduction of virtual fixture techniques [100, 88, 89, 90, 91]. A virtual fixture can provide haptic guidance which is much more complex than a simple straight line. Anisotropic compliances, for example, can be defined as in [126].

Movements inside volumes can be considered as well as in [90] where a tube, a cone and their combinations were considered, both in the micro and in the macro scale.

As a matter of fact, virtual fixtures are highly task dependent. For instance, in [127], the authors compared several forbidden-region virtual fixtures (FRVF), i.e., computer-generated constraints, on four control architectures using three different metrics (tracking, safety and submittance) for analysing performances. Different FRVRs performed best for each of the three metrics and no general optimal solution could be defined. The choice of the best fixture was shown to be directly related to the desired application of the telemanipulator. Another important limitation of the virtual fixture technique is the correct definition of the level of assistance. This problem is discussed in details in Sec. 2.3.4.

The use of virtual fixtures implies the definition of on-line/off-line defined constant/variable repulsive or attractive areas. In classic shared control, however, there is a division of tasks between two agents, i.e., the human user and the autonomous controller (see Sec. 2.3.3.2). A slightly different approach than virtual fixtures, named haptic shared control, was proposed [128]. In haptic shared control, "both the human and the [guidance] system exert forces on a control interface, of which its output (its position) remains the direct input to the controlled system" [128]. Haptic shared control was applied to car-following applications [129, 128], and to teleoperation [97, 96, 130, 131]. A similar approach was also presented in [116] for teleoperating mobile robots. An admittance controller was proposed in order to receive as input the sum of the user force and of the autonomous system force, which is computed using an impedance model.

In haptic shared control, the human operator should have continuous interaction and communication with autonomy and should retain final authority [128]. The effects of inaccuracies of the haptic guidance, due for instance to a non-perfect knowledge of the environment or inaccurate sensory information were studied in [130, 132] for a peg-in-hole task in a virtual environment. The system was found relatively robust against small inaccuracies whereas large inaccuracies substantially degraded task performance. Conflicts between the human operator and the autonomous controller were also studied and a trial-by-trial adaptation was found to be the most promising approach to mitigate conflicts. On the other hand, this approach is best suitable for repetitive motion tasks [131].

As correctly resumed by [64], the main challenges for applying virtual fixtures, and similar techniques in general, are the choice of the right fixture, the optimal trade-off between completely human-commanded and purely computer-controlled operation (by regulating the stiffness of the haptic feedback), and the recognition of task primitives.

2.3.3.2 Partitioned Space

In more complex robotic systems, shared control architectures were proposed as a potential solution. The space is partitioned such that the human controls high-level decision making while the robotic system controls low-level operations, such as obstacle avoidance and force management. The main challenge of designing a shared control architecture is the task allocation between the human operator and the autonomous controller. The panoply of shared control architectures comprehends several solutions to this main question. Consider for instance the problem of navigation, which can be equally applied to telemanipulation and mobile robots. Varying the task allocation, simple shared control solutions like autonomous obstacle avoidance systems via some sensory information were proposed [103, 104, 102, 116] as well as more elaborated motion planners that can for instance suggest alternative paths to the user [112, 114] or help performing smooth trajectories [133]. Solutions for predicting the user's motion intentions were also presented [92, 93].

Another important issue is the definition of the informative feedback that is provided to the user. Commonly, kinesthetic haptic feedback [134, 135] is employed to complement the simple visual feedback that is normally available to the human operator. The typical approach is to define the haptic feedback as the mismatch between the desired and the actual motion [136], but an integral haptic feedback that takes into account the global execution of the path can also be implemented [113, 114]. Audio alarms were proposed but they may also be a cause of confusion for the operator [137]. On the other hand, a virtual-reality simulator can be used to provide the user with a virtual reconstruct of the remote scene. The replacement of camera images with a 3D virtual scene may lead to superior performances in case of dynamic and time-critical conditions, whereas the combination of the two visual feedbacks may cause confusion and lead to poorer results [115]. A similar approach, called the hidden robot concept, was proposed in [138]. The idea is that the operator first performs the desired task in a virtual environment. Then, the slave robot reproduces the motion in the remote environment. The master and slave sides are consequently two decoupled closed-loop systems. This, of course, needs a very accurate reconstruction of the target environment.

2.3.4 Human Authority vs Haptic Autonomy

As mentioned before, the definition of the system autonomy with respect to the user authority is one of the most challenging problem when designing an assistive control system [139]. First of all, the virtual assistance can be defined as a fixed or a variable entity. It can be reformulated as the problem of the authority allocation between the user and the controller. In several shared control architectures presented in the

past, the autonomous controller was initially given full control of some or most of the aspects of motion [140, 141, 142] and tasks could not be reassigned. In the case of virtual fixtures for example, this corresponds to the initial definition of the optimal value for the virtual constraints [100, 91]. A fixed level of autonomy, however, can substantially degrade the user's detection of automation failures [143, 144].

Variable assistances have been discussed since the nineteen seventies [145] in supervisory control. The variation can be defined over pre-determined discrete levels, which is the most common approach in the literature, [146, 147, 148, 84, 149] or in a continuous fashion [105, 150, 110, 151, 101, 92]. In variable assistance methods, the autonomous controller is typically in charge of switching between the different levels of assistance, depending on the user performances [105], potentially dangerous situations [110, 129] or the robot's confidence in the prediction of the user's motions and the difficulty of the task [101, 92]. The user can also be permitted to vetoing the switching [84, 152, 148].

Some blending techniques were presented with the aim of combining user's and autonomous controller's commands [153, 154, 101, 92]. In [154], the blending function is tested on a 1-DoF navigation problem and the user's command is evaluated with respect to the optimal solution of the problem. In [101] and [92], the concept of policy blending is introduced in order to formalize assistance as an arbitration of two policies: the user's input and the robot's prediction of the user's intent. Hence, the autonomous controller must accomplish two tasks for providing the proper assistance: predicting what the user wanted and deciding how to use this prediction to assist. The arbitration of the 'aggressiveness' of the autonomous system's intervention is to be moderated by the robot's confidence in the prediction.

2.3.4.1 Optimal Level of Autonomous Assistance

As a matter of fact, there exist several different approaches to the task allocation problem which is quite complex and task-dependent. The challenge is to determine the optimal solution, if one exists. Literature does not offer a formalized and globally accepted solution. The problem is twofold: 1) to implement the most performing solution, e.g., in terms of collision avoidance or completion time, 2) to develop a shared control architecture that is positively evaluated by the human operator.

In the case of a shared control space, the problem translates as that of defining the best virtual fixture or guiding haptic cues. In [91], the authors tried to define the optimal assistance level by studying two main scenarios: one where the user and the virtual assistance agreed and the other where they disagreed. In the former case, harder virtual fixtures guaranteed better performances. In the latter, the harder the virtual fixtures were, the poorer the performances. In [90] a visual tracking system

was used to define virtual fixtures. It was observed that assisted execution always outperforms unassisted execution for hard, medium, and soft virtual fixturing scenarios. Hard virtual fixtures assured the best performance. However, they seriously limit the user's ability to control the direction of motion and problems arise when the visual tracking is erroneous. In [155], the authors proposed the use of adaptive virtual fixtures able to cope with unexpected obstacles or incorrect modeling. The concept was to make virtual fixtures adaptable by dividing the main task into several subtasks, namely, dividing the trajectory into a series of straight lines. The subtasks, and so the virtual fixtures, were switched autonomously. However, the authors assume it would be more efficient if the operator was allowed to define the number of states and when to switch. Hidden Markov Model (HMM)-based algorithms were also proposed to recognize the user's motion intentions and hence apply the appropriate virtual fixture [147, 156].

In the case of a partitioned space, several researchers supported the assumption of a human-centered control architecture [157, 158, 139, 111, 128]. This approach prevents the human operator from losing task execution skills or situational awareness [159, 157, 144, 160, 95]. Besides, a limited assistance was shown to result in higher retention in the context of motion learning [159]. Other authors argued that the autonomous controller should have the final authority over the human operator for particular or dangerous situations [139, 84, 129, 161]. Highly autonomous systems were also proposed to allow, for instance, teleoperating a 6-DoF robotic arm with a computer mouse [141], grasping object with the minimum effort [95] or manipulating objects with a robotic arm mounted on a wheelchair [142].

2.3.5 Control Level

In the vast majority of assisted control literature, the human operator has instantaneous (and continuous) control over some DoF of the system. However, it is not always interesting for the operator to have this form of control. In a grasping scenario for example, the user can be interested in the final grasping pose and less concerned about the path to arrive there. This also applies in navigation problems where the user may be interested in controlling some DoF of the planned trajectory without having to be in continuous command of the robot.

Controlling the high level goals of the system rather than its low-level details draws us back to supervisory control. In supervisory control however, the operator is behind a software interface and selecting from a pre-defined set of actions to be performed by the system. She/He can have more control through specialized programming languages which allow for the tuning of lower-level details. But the immersive experience of assistive control is lost. To counteract that, few works have proposed

immersive shared-planning architectures to replace continuous shared-control techniques.

In [162], the authors propose a shared-control architecture for controlling the trajectory of a mobile robot. The user sketches a trajectory on the visual feedback from the remote environment by acting on a haptic input device. The sketch is generated by interpreting the user's motion and a path is planned accordingly. The system makes use of the operator's cognitive capability to decide on the general behaviour and generate autonomously the corresponding collision-free path.

A different approach is adopted in [114, 113]. Here, the user, assisted by autonomy, acts on a haptic device to modify some DoF of a pre-planned trajectory. The target environment contains obstacles (modeled as potential hills) and points of interest (modeled as potential wells). As the user modifies some parameters of the trajectory, autonomy optimizes the remaining DoF against the system constraints. The user is informed of the discrepancy between her/his commands and the executed control inputs through haptic feedback. She/He also receives haptic cues guiding them to points of interest. The described system is meant for controlling a mobile robot in 2D space.

2.4 Collaborative Robotics

Collaborative robots are robots which exhibit a significant level of autonomous behaviour. They support the human operator in performing a desired task by following her/his lead in a high-level master-slave hierarchy. In contrast to assistive systems, the human operator has no command over the robot. Contrary to that, the robot 'observes' the behaviour of the operator to interpret her/his intentions and react accordingly.

A typical example of collaborative robots are Caster-like robot partners which can be used, for example, in human-robot joint bulky load transport. The robot can reactively compensate for the dynamics of the object while following the lead of the human operator who defines the direction of motion [163, 164]. However, such a follower strategy implements merely a trolley for heavier loads rather than an actual cooperation partner and, while simple tasks can successfully be fulfilled, more complex tasks including environmental constraints typically require an active contribution to the task by the robot [165]. Active contribution from the robot can also decrease the cognitive load and physical effort required from the human partner [166, 167, 168]. On the other hand, an experience-driven robotic assistant was proposed in [169]. The robot acquires human knowledge through observation and continuous repetitions to improve its haptic cooperation with the human partner.

For further details on cobots, their potential applications and safety recommendations, we refer the interested reader to the concise survey presented in [170].

2.5 Peer-to-Peer Collaboration

Peer-to-Peer collaboration is having a fully autonomous robot working hand-in-hand with the human operator as her/his equal. The robot needs to have the same cognitive capabilities of the human operator regarding the task at hand and be able to transmit its 'thoughts' to its human-partner. While collaborative robots are already an active research topic, the human operator is always regarded as the master and her/his judgement as the gold standard. We are yet to have an autonomous system that can challenge that.

2.6 Conclusions

Table 2.6 attempts to provide an overview of some of the main authors/groups who have worked on HRI. The table highlights the main topics which each of the groups have tackled. Categorizing the literature can be a complex task as lines are blurry between categories. Nevertheless, this table is meant to highlight the main focus of each group rather than to encompass the enormous volume of literature on the topic.

With respect to the existing literature discussed above and briefly summarized in Table 2.6, this thesis proposes a number of novel shared-control frameworks for telemanipulation. We propose semi-autonomous visual-based techniques for controlling single and dual arm serial manipulators for approaching and grasping target objects. We also present a task-relevant haptic interface for teleoperating the arms of a humanoid robot. The user is informed about the impact of her/his commands on the balance of the humanoid through informative force feedback that guides her/him towards more stable positions. Therefore, our work mainly fits under the "manipulators" and "humanoids" categories in the table.

It is also worth paying attention to the right-most side of Table 2.6 which highlights the 'control-level' in shared-control systems (see also Sec. 2.3.5). One can note that most of the groups who have worked on shared control have focused on instantaneous control rather than higher-level shared planning techniques or controlling the future trajectories of the robot. The user usually commands a set of DoF of the system and receives feedback informing about its current state. In contrast, this thesis also considers the possibility of shared-trajecotry planning and proposes new modalities in this regard. In particular, we generalise the approach described in [114, 113] from 2D space to 3D space applying it to two 6-DoF serial

manipulators rather than planar mobile robots. Moreover, we incorporate vision in the loop and inform the user about the impact of her/his actions on the quality of visual tracking (besides other constraints) through haptic feedback.

Another important aspect to be considered when controlling a robotic system is the final goal of the control. For example, works tackling grasping usually focus on assisting the user towards a successful grasp while neglecting the post-grasp phase of manipulation. The success of the post-grasp task can, in fact, be highly impacted by the choice of the grasp itself. In this respect, we propose a shared-control framework which takes into account the feasibility of the 'post-task' when assisting the operator in controlling the system. This is a topic which, up to our knowledge, has not been considered in literature before.

						Shared Control						
						System Type				Control Level		
	Manual Control	Passivity/ Stability	Supervisory Control	Collaborative Robotics	Haptic Devices	Manipulators	Mobile Robots	Multi-Robot	Humanoids	Instantaneous	Trajectory Based	Post-Task Feasibility
Prof. Abbink's group	x				x	x	x			x		
Prof. Rodriguez's group						x				x		
Prof. Bicchi's group	x		x		x	x			x	x		
DLR	x	x	x		x	x	x			x		
Prof. Gillespie's group	x				x		x			x		
Prof. Hannaford's group	x	x	x	x	x	x				x		
Prof. Hirche's group	x	x		x								
Prof. Khatib's group	x					x			x	x		
Dr. Kheddar's group	x	x		x		x		x	x	x		
Dr. Kuchenbecker's group	x				x	x			x	x		
Prof. Okamura's group	x		x	x	x	x				x		
Prof. Patel's group	x					x				x		
Prof. Peer's group	x			x	x	x				x		
Prof. Ryu's group	x	x				x	x	x		x	x	
Prof. Secchi's group	x	x		x			x	x		x		
Prof. Stramigioli's group	x	x					x			x		
This Thesis						x			x	x	x	x

Table 2.1: The different domains of HRI and the focus of some authors/groups on each.

Part II

Contributions: Shared Control of Serial Manipulators

Chapter 3

Pre-Grasping Phase

Contents

3.1 Problem Setting **42**
3.2 Instantaneous Shared-Control **43**
3.2.1 Gripper control 43
3.2.2 Camera Control 45
3.2.3 Master Side 52
3.2.4 Kinematic Constraints 55
3.2.5 Passivity Analysis 57
3.3 Trajectory-Based Shared Control **60**
3.3.1 Overview 60
3.3.2 Trajectory Representation 61
3.3.3 System Constraints 65
3.3.4 The Overall Architecture 69
3.4 Experimental Results **76**
3.4.1 Instantaneous Shared Control 76
3.4.2 Human-Subject Evaluation 80
3.4.3 Shared Trajectory Planning 88
3.5 Summary **95**

As discussed in the previous chapter, assisted control systems have proven their potential as a convenient middle ground between manual operation and full autonomy. The human presence is still highly desirable in many applications for reasons ranging from the frailty of autonomous systems in unknown environments to the sensitivity of the application, e.g., assisted surgery or nuclear robotics. The shared-control architecture presented in this chapter is motivated by the European H2020 RoMaNS project described in Sec. 1.1. In the RoMaNS

Figure 3.1: Top figures: an illustrative sketch of the robotic testbed envisioned for the RoMaNS sort and segregation tasks (left) with a human operator in partial control of the overall motion (right). Bottom figures: examples of the typical waste material to be handled by the robotic cell.

scenario (see Fig, 3.1), a human operator has access to a system consisting of two robotic arms, one equipped with a gripper and the other one with a camera, with the goal of approaching and grasping nuclear waste for sort and segregation purposes. Manipulation can be divided into three main phases. The pre-grasp phase where the user commands the manipulator to approach and grasp an object, the grasping phase itself, and the post-grasp phase were the desired manipulation action is performed (e.g., sorting into the respective containers). This chapter along with chapter 4 tackles the pre-grasp phase whereas chapter 5 comments on issues related to the post-grasp task. The grasping problem itself is out of the scope of this thesis. However, an off-the-shelf grasping algorithm is used in chapter 4 for showcasing how one could take into account actual grasp requirements.

As a step towards an efficient framework for sort and segregation purposes, we propose in Sec. 3.2 a shared control architecture in which a (visual-based) autonomous algorithm is in charge of regulating a subset of the gripper DoF for facilitating the approaching phase towards an object of interest. At the same time, the human operator is given the possibility to steer the gripper along the remaining null-space directions w.r.t. the main task by acting on a force feedback device. Due

to the complexity of its motion, the camera is instead autonomously controlled so as to keep a good vantage point w.r.t. the scene and, thus, allow for a successful reconstruction of both the gripper and object poses. Finally, suitable force cues are fed to the operator in order to assist her/him during the task. In contrast with classical implementations, where the force feedback is, in general, proportional to some position/task error, the novel design of the proposed force cues can provide more structured information about the feasibility of the user's commands w.r.t. possible constraints of the robotic system such as joint limits, singularities and visibility constraints.

The approach discussed in Sec. 3.2, albeit effective in many scenarios, suffers however from a 'locality' issue since the operator can only provide instantaneous velocity commands (in a suitable task space), and receive instantaneous force feedback cues. It may be interesting for the user instead to act on a whole *future trajectory* in task space, and receive a corresponding integral force feedback along the whole planned trajectory (because of any constraint of the considered system). To this end, in Sec. 3.3, we extend the approach of Sec. 3.2 toward a shared trajectory-planning architecture where the user is assisted by an autonomous component to plan and optimize a future trajectory of both manipulators. This actually lies in the scope of some recent works in the context of shared control of mobile robots (and quadrotors) where shared-planning of 3D planar trajectories was explored along with novel criteria for haptic cues besides the sole (and typical) position/task error see, e.g., [171, 172]. The architecture described in Sec. 3.3, presents the generalization of the mentioned strategies to the 6-dimensional case (the full gripper pose), and it applies the architecture to a 12-DoF dual arm manipulator system while accounting for its different visual and kinematic limitations.

The chapter is organized as follows. In Sect. 3.1 the general problem is introduced, while the 'instantaneous' shared control architecture is described in detail in Sect. 3.2. Sect. 3.3 elaborates toward the shared trajectory planning approach and Sect. 3.4 reports the results of several experiments. Finally Sect. 3.5 concludes the chapter and discusses some future directions.

Some of the results contained in this chapter have been presented in different international venues [22, 26, 24, 29, 30, 31, 32]. Related descriptive media is available under:

- Instantaneous shared-control (Sec. 3.2): `https://youtu.be/_dBvk9K6E0Q`.
- User subject Evaluation (Sec. 3.2): `https://youtu.be/eh2_425ryYE`.
- Shared Trajectory-Planning (Sect. 3.3): `https://youtu.be/p9X8ZKJ77m4`.

3.1 Problem Setting

The scenario considered in this section consists of two 6-dof serial manipulators, one equipped with a monocular (calibrated) camera and the other one with a gripper, aiming at grasping an object of interest (See Fig. 3.2). We consider four frames of reference: $\mathcal{F}_o : \{\mathcal{O}_o; \boldsymbol{x}_o, \boldsymbol{y}_o, \boldsymbol{z}_o\}$ attached to the object to be grasped, $\mathcal{F}_g : \{\mathcal{O}_g; \boldsymbol{x}_g, \boldsymbol{y}_g, \boldsymbol{z}_g\}$ attached to the gripper and $\mathcal{F}_c : \{\mathcal{O}_c; \boldsymbol{x}_c, \boldsymbol{y}_c, \boldsymbol{z}_c\}$ attached to the camera; in addition to the world frame $\mathcal{W}$. We assume that $\boldsymbol{z}_g$ is aligned with the gripper approaching direction, and that (as usual) $\boldsymbol{z}_c$ is aligned with the camera optical axis.

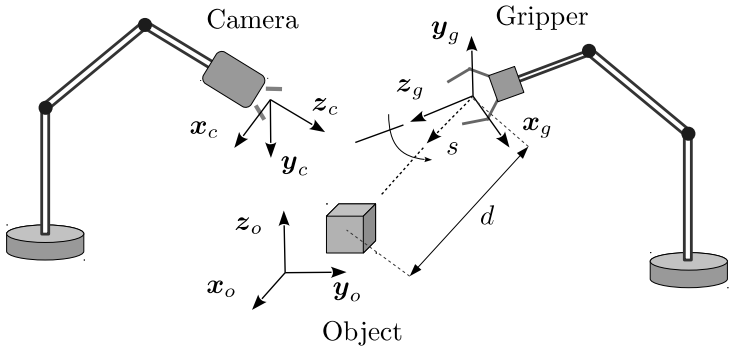


Figure 3.2: An illustrative representation of the two 6-dof serial manipulator arms equipped with a camera and a gripper, respectively, together with other quantities of interest

We let $({}^c\boldsymbol{p}_g, {}^c\boldsymbol{R}_g) \in \mathbb{R}^3 \times SO(3)$ represent the 3D pose of $\mathcal{F}_g$ w.r.t. $\mathcal{F}_c$ expressed in $\mathcal{F}_c$ and, similarly, $({}^c\boldsymbol{p}_o, {}^c\boldsymbol{R}_o) \in \mathbb{R}^3 \times SO(3)$ represent the 3D pose of $\mathcal{F}_o$ w.r.t. $\mathcal{F}_c$ expressed in $\mathcal{F}_c$. We assume that an accurate-enough 3D model of both the object to be grasped and of the gripper is available beforehand. This allows to leverage any model-based tracker, such as those present in the ViSP library [173], for retrieving *online* a reliable estimation of the camera/object and camera/gripper relative poses in the camera frame. We can then consider the relative gripper/object poses in the gripper frame $\mathcal{F}_g$, that is, the quantities ${}^g\boldsymbol{R}_o = {}^g\boldsymbol{R}_c\,{}^c\boldsymbol{R}_o$ and ${}^g\boldsymbol{p}_o = {}^g\boldsymbol{R}_c({}^c\boldsymbol{p}_o - {}^c\boldsymbol{p}_g)$, as known from the corresponding reconstructed poses in $\mathcal{F}_c$. Finally, we let

$$\boldsymbol{v}_c = [\dot{\boldsymbol{p}}_c^T, \boldsymbol{\omega}_c^T]^T \in \mathbb{R}^6 \tag{3.1}$$

and

$$\boldsymbol{v}_g = [\dot{\boldsymbol{p}}_g^T, \boldsymbol{\omega}_g^T]^T \in \mathbb{R}^6 \tag{3.2}$$

represent the camera and gripper linear/angular velocities expressed in their respective frames, which will play the role of control inputs in the following developments.

3.2 Instantaneous Shared-Control

The goal of the proposed shared control architecture is to (*i*) let an autonomous algorithm control a part of the gripper/object relative pose for facilitating the approaching phase towards the object to be grasped, (*ii*) let a human operator control the remaining free DoF of the gripper/object relative pose via a force-feedback device, (*iii*) provide the human operator with online force cues informing about the feasibility of her/his motion commands w.r.t. possible constraints of the robotic system, and, finally, (*iv*) let an autonomous algorithm control the camera motion so as to keep a suitable vantage point w.r.t. the observed scene (i.e., both the gripper and the object).

We now proceed to detail the components of the shared control architecture.

3.2.1 Gripper control

As discussed, the gripper controller should constrain a suitable subset of the gripper DoF while delegating the remaining free DoF to a human operator. This goal can be achieved by considering the constrained DoF as a 'primary task' to be autonomously regulated towards some desired value, and by allowing the human operator to actuate the resulting null-space motions w.r.t. the main task. To this end, let $\boldsymbol{s} \in \mathbb{R}^m$, $m < 6$, be a m-dimensional subset of the gripper/object relative pose representing the primary (autonomous) task, and let $\dot{\boldsymbol{s}} = \boldsymbol{L}_s \boldsymbol{v}_g$, with $\boldsymbol{L}_s \in \mathbb{R}^{m \times 6}$ being the corresponding interaction matrix with $\text{rank}(\boldsymbol{L}_s) = r \leq m$ and, thus, $\dim(\ker(\boldsymbol{L}_s)) = 6 - r = n$. Let also $\boldsymbol{N} = [\dots \boldsymbol{n}_i \dots] \in \mathbb{R}^{6 \times n}$ be a basis of the n-dimensional null-space of $\boldsymbol{L}_s$, and $\boldsymbol{\lambda} = [\dots \lambda_i \dots]^T \in \mathbb{R}^n$ a vector collecting the n pseudo-velocity commands λ_i that will be exploited for actuating the individual null-space motions $\boldsymbol{n}_i$.

The following control law[1]

$$\boldsymbol{v}_g = k_g \boldsymbol{L}_s^{\dagger}(\boldsymbol{s}_d - \boldsymbol{s}) + \boldsymbol{N}\boldsymbol{\lambda}, \quad k_g > 0, \tag{3.3}$$

with $\boldsymbol{L}_s^{\dagger}$ representing the Moore-Penrose pseudo-inverse of matrix $\boldsymbol{L}_s$, achieves the two stated objectives:

[1] In the Visual Servoing nomenclature, controller (3.3) would be classified as a eye-to-hand PBVS scheme since it aims at regulating the 3D object/gripper pose reconstructed from an off-board camera [174].

1. autonomous regulation of the m 'constrained DoF' $\boldsymbol{s}$ (the primary task) towards a desired value $\boldsymbol{s}_d$;

2. concurrent actuation of the remaining n null-space motions spanned by vectors $\boldsymbol{n}_i$ via the corresponding pseudo-velocity commands λ_i (which will be specified online by a human operator).

Clearly, any basis $\boldsymbol{N}$ for the null-space of $\boldsymbol{L}_s$ is a valid choice in (3.3). However, for the sake of providing a usable interface to the human operator, it is important to select a basis for which the motion directions $\boldsymbol{n}_i$ have a clear/intuitive physical meaning in order to ease the operator's intervention during the task.

As an illustrative (but significant) case study, we hereby choose to define the primary task variables $\boldsymbol{s}$ as the 3D direction towards the object to be grasped, i.e., the unit-norm vector

$$\boldsymbol{s} = \frac{{}^g\boldsymbol{p}_o}{\|{}^g\boldsymbol{p}_o\|} \in \mathbb{S}^2, \tag{3.4}$$

see Fig. 3.2. This is meant to ensure an autonomous/precise control of the gripper alignment w.r.t. the target object during the pre-grasping approaching phase. The interaction matrix associated to this choice of $\boldsymbol{s}$ is then (see, e.g., [175, 176])

$$\boldsymbol{L}_s = \begin{bmatrix} -\frac{1}{d}\boldsymbol{P}_s & [\boldsymbol{s}]_\times \end{bmatrix} \in \mathbb{R}^{3\times 6} \tag{3.5}$$

where $\boldsymbol{P}_s = \boldsymbol{I} - \boldsymbol{s}\boldsymbol{s}^T$ is the orthogonal projector onto the tangent space of the unit sphere $\mathbb{S}^2$ at $\boldsymbol{s}$, $d = \|{}^g\boldsymbol{p}_o\|$, and $[\cdot]_\times$ represents the usual skew-symmetric matrix operator. Note that, in this case, $m = 3$ but $r = \text{rank}(\boldsymbol{L}_s) = 2$ because of the unit norm constraint in (3.4). Therefore, there will exist $n = 4$ independent directions in the null-space of $\boldsymbol{L}_s$. Also, since $\boldsymbol{L}_s^T\boldsymbol{s} = \boldsymbol{0}$ and $\text{span}(\boldsymbol{L}^\dagger) = \text{span}(\boldsymbol{L}^T)$, the control law (3.3) can be simplified into

$$\boldsymbol{v}_g = k_g\boldsymbol{L}_s^\dagger\boldsymbol{s}_d + \boldsymbol{N}\boldsymbol{\lambda}. \tag{3.6}$$

Among the many possibilities, we found a convenient choice for the null-space basis $\boldsymbol{N}$ to be

$$\boldsymbol{n}_1 = \begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{0} \end{bmatrix}, \boldsymbol{n}_2 = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{s} \end{bmatrix}, \boldsymbol{n}_3 = \begin{bmatrix} -[\boldsymbol{s}]_\times\boldsymbol{e}_y \\ -\boldsymbol{P}_s\boldsymbol{e}_y/d \end{bmatrix}, \boldsymbol{n}_4 = \begin{bmatrix} [\boldsymbol{s}]_\times\boldsymbol{e}_x \\ \boldsymbol{P}_s\boldsymbol{e}_x/d \end{bmatrix}, \tag{3.7}$$

with $\boldsymbol{e}_x = [1\ 0\ 0]^T$ and $\boldsymbol{e}_y = [0\ 1\ 0]^T$.

The advantage of the proposed basis is that it yields an 'interface' to the human operator with a clear (and decoupled) physical interpretation (see appendix A.1 for the proof of validity and orthogonality of the basis). In particular, when plugging (3.7) in (3.6) one obtains the following:

- the null-space direction $\boldsymbol{n}_1$ realizes a motion along $\boldsymbol{s}$ which results in $\dot{d} = \lambda_1$. Therefore, the operator gains direct control over the rate of change of the distance $d(t)$ along the pointing direction towards the target. Furthermore, $d(t)$ is not affected by any other null-space direction;

- the null-space direction $\boldsymbol{n}_2$ realizes a rotation about $\boldsymbol{s}$ with angular speed λ_2. Therefore, the operator gains direct control over the gripper angular velocity about the pointing direction towards the target. Furthermore, the rotation about $\boldsymbol{s}$ is not affected by any other null-space direction;

- the null-space directions $\boldsymbol{n}_3$ and $\boldsymbol{n}_4$ realize two coordinated motions (linear/angular velocity) that displace the gripper over a sphere centered at the target object (with, thus, $d(t) = const$). In particular, $\boldsymbol{n}_3$ makes the gripper move in the positive $\boldsymbol{x}_g$ direction and $\boldsymbol{n}_4$ in the positive $\boldsymbol{y}_g$ direction *without* affecting, in both cases, the distance along $\boldsymbol{s}$ (only actuated by $\boldsymbol{n}_1$) and the angle about $\boldsymbol{s}$ (only actuated by $\boldsymbol{n}_2$).

Figure 3.3 provides a visual illustration of the four null-space directions (3.7).

We conclude by noting that the reported experimental case studies always considered a (typical) situation in which $\boldsymbol{s}_d = \boldsymbol{e}_z = [0\ 0\ 1]^T$ so as to force the target direction $\boldsymbol{s}(t)$ to be aligned with the gripper approaching direction $\boldsymbol{z}_g$. In this case, the null-space motions (3.7) evaluated at $\boldsymbol{s} = \boldsymbol{s}_d$ reduce to (i) a translation along $\boldsymbol{z}_g$ with speed λ_1, (ii) a rotation about $\boldsymbol{z}_g$ with speed λ_2, a translation along $\boldsymbol{x}_g$ with speed λ_3 (combined with a rotation about $\boldsymbol{y}_g$ with speed λ_3/d), and a translation along $\boldsymbol{y}_g$ with speed λ_4 (combined with a rotation about $\boldsymbol{x}_g$ with speed λ_4/d). Substituting in eq. (3.6), the final control law governing the motion of the gripper becomes

$$\boldsymbol{v}_g = k_g \boldsymbol{L}_s^{\dagger} \boldsymbol{e}_z + \boldsymbol{N}\boldsymbol{\lambda}. \tag{3.8}$$

3.2.2 Camera Control

As discussed in Sect. 3.2, the camera should control its motion in order to keep a suitable vantage point w.r.t. both the gripper and the target object for allowing an accurate 3D reconstruction of their poses. Towards this end, we first discuss in Sect. 3.2.2.1 a simple strategy of maintaining the projection of the object center of mass (CoM) ${}^c\boldsymbol{p}_o$ and of the gripper CoM ${}^c\boldsymbol{p}_g$ at some desired locations on the camera image plane for ensuring that the gripper and the object stay within visibility during operation. The advantage of this strategy lies in its simplicity of implementation. It has, however, some drawbacks (discussed at the end of Sect. 3.2.2.1) which have

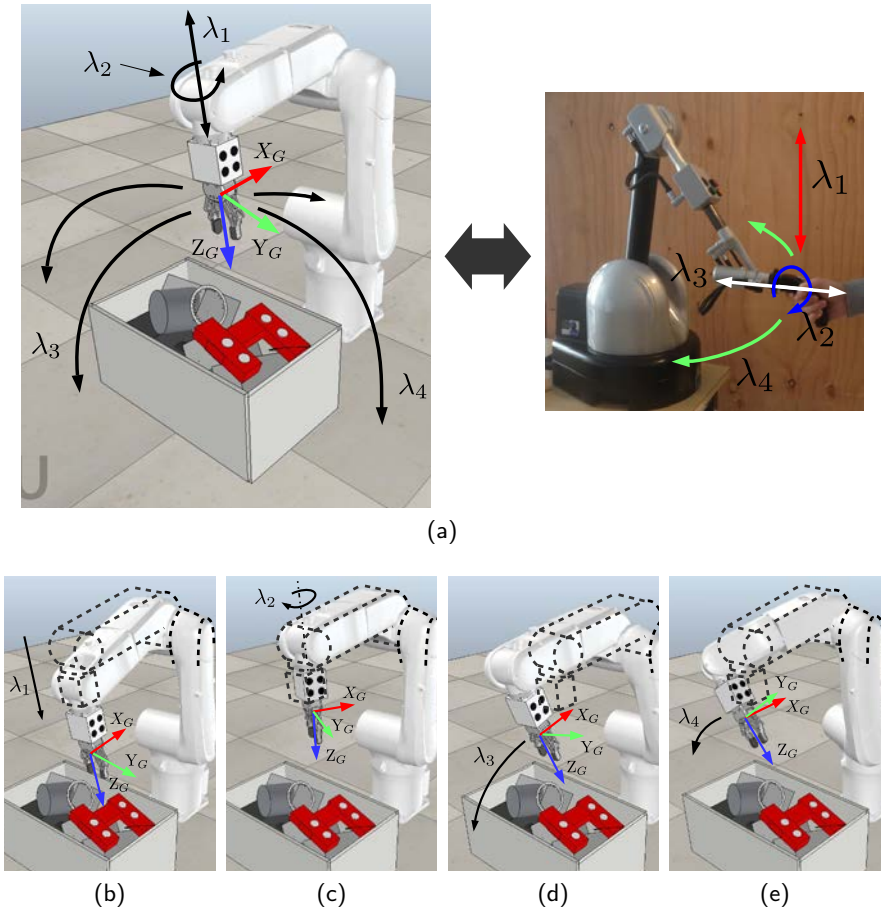


Figure 3.3: A visualization of the four null-space directions defined in (3.7). (a): the four directions and the associated pseudo-velocity commands λ_i projected on the slave and the master. (b–e): an illustration of the motion resulting from actuating each individual direction $\boldsymbol{n}_i$.

led us to subsequently propose a more advanced approach based on image moments which is described in Sect. 3.2.2.2.

3.2.2.1 Point-Based IBVS

Figure 3.4 illustrates the main quantities of interest with, in particular, $\bar{\boldsymbol{p}}_o$ representing the point feature projection of ${}^c\boldsymbol{p}_o$, and $\bar{\boldsymbol{p}}_g$ the point feature projection of ${}^c\boldsymbol{p}_g$. Since our aim is to have control over the image plane location of ${}^c\boldsymbol{p}_o$ and ${}^c\boldsymbol{p}_g$, we resort to an IBVS approach [174] with the values of $\bar{\boldsymbol{p}}_o$ and $\bar{\boldsymbol{p}}_g$ obtained from the projection of the (reconstructed) 3D poses ${}^c\boldsymbol{p}_o$ and ${}^c\boldsymbol{p}_g$. As the target object is assumed static in the scene, the dynamics of $\bar{\boldsymbol{p}}_o$ is just

$$\dot{\bar{\boldsymbol{p}}}_o = \boldsymbol{L}_p(\bar{\boldsymbol{p}}_o,\, z_o)\boldsymbol{v}_c$$

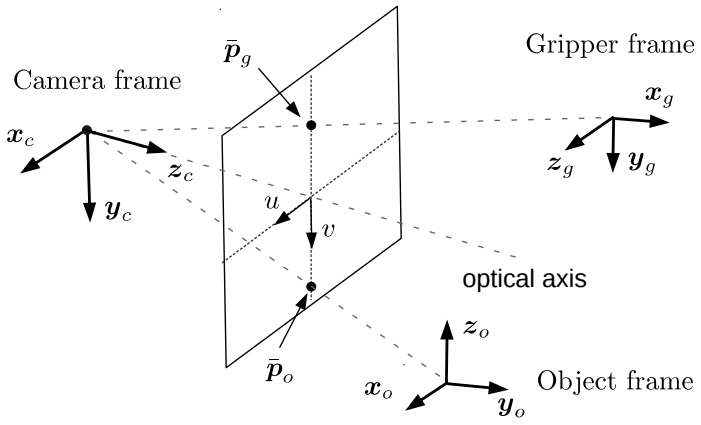


Figure 3.4: Feature points $\boldsymbol{p}_g$ and $\boldsymbol{p}_o$ exploited for controlling the camera motion

where z_o is the depth associated to $\bar{\boldsymbol{p}}_o$ and $\boldsymbol{L}_p \in \mathbb{R}^{2\times 6}$ is the interaction matrix [174] for a point feature $\bar{\boldsymbol{p}} = [\bar{x} \;\; \bar{y}]$ defined as

$$\boldsymbol{L}_p = \begin{bmatrix} -1/z & 0 & \bar{x}/z & \bar{x}\bar{y} & -(1+\bar{x}^2) & \bar{y} \\ 0 & -1/z & \bar{y}/z & 1+\bar{y}^2 & -\bar{x}\bar{y} & -\bar{x} \end{bmatrix}. \tag{3.9}$$

The dynamics of $\bar{\boldsymbol{p}}_g$ is instead also affected by the gripper's own motion generated by controller (3.8). In particular, it is straightforward to obtain the relationship

$$\dot{\bar{\boldsymbol{p}}}_g = \boldsymbol{L}_p(\bar{\boldsymbol{p}}_g, z_g) \begin{bmatrix} \dot{\boldsymbol{p}}_c - {}^c\dot{\boldsymbol{p}}_g \\ \boldsymbol{\omega}_c \end{bmatrix}$$

Letting now $\bar{\boldsymbol{p}}_t = [\bar{\boldsymbol{p}}_o^T \;\; \bar{\boldsymbol{p}}_g^T]^T$, the following IBVS controller

$$\boldsymbol{v}_c = \begin{bmatrix} \boldsymbol{L}_p(\bar{\boldsymbol{p}}_o, z_o) \\ \boldsymbol{L}_p(\bar{\boldsymbol{p}}_g, z_g) \end{bmatrix}^{\dagger} \left(k_c(\bar{\boldsymbol{p}}_d - \bar{\boldsymbol{p}}) + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{L}_p(\bar{\boldsymbol{p}}_g, z_g) \end{bmatrix} \begin{bmatrix} {}^c\dot{\boldsymbol{p}}_g \\ \boldsymbol{0} \end{bmatrix} \right), \tag{3.10}$$

$k_c > 0$, achieves the regulation of $\bar{\boldsymbol{p}}$ towards a desired value $\bar{\boldsymbol{p}}_d$ while compensating for the (known) gripper motion generated by (3.8).

While the described camera controller serves the basic purpose of keeping the projections of the center of the object and that of the gripper in a pre-defined desired position, it has some limitations:

- There is no restriction on the distance between the camera and the scene. As long as $\bar{\boldsymbol{p}}_o$ and $\bar{\boldsymbol{p}}_g$ are at their desired locations on the image plane, the camera can go farther shrinking the size of the object and the gripper in the image. This deteriorates the quality of tracking with risks of potential failure. The camera can also go closer such that parts of the gripper and the object leave the FoV.

- Nothing prevents the gripper and the object from occluding one another in the image which may lead to tracking failures as well.

- Fixing the center of the projections in pre-defined positions may be over-restrictive for the system. In fact, we are interested in keeping the projections inside the FoV and not necessarily at a particular position.

To address these issues, we then propose an image-moment-based approach tackling these limitations.

3.2.2.2 Moment-Based IBVS

To sum up the points detailed above, the purpose of the controller of the camera is to maintain visibility of both the object and the gripper in the FoV, i.e., to keep them within the image plane boundaries and prevent any mutual occlusion. This is indeed crucial for successfully retrieving online the relative gripper/object pose from visual input as well as for providing the human operator with a proper visual feedback of the remote environment. These constraints (keeping the gripper/object in the FoV and avoiding mutual occlusions) can be encoded in a suitable cost function h_V whose minimization can be exploited for the camera control purposes. We now proceed to define such a "visual constraint" cost function h_V which will be used later in Sec. 3.3.3.

Let the gripper and the object be approximated as spheres in 3D space: the sphere representing the gripper is centered at ${}^c\boldsymbol{p}_g$ with a radius R_g , and similarly the sphere representing the object is centered at ${}^c\boldsymbol{p}_o$ with a radius R_o. As before, ${}^c\boldsymbol{p}_g$ and ${}^c\boldsymbol{p}_o$ can be obtained online from the used tracker. Given the known pose and radius of each sphere in the camera frame, one can easily compute its corresponding projection on the image plane which, in the general case, is an ellipse with center $(\bar{x}, \bar{y})$ and major/minor axes r_1 and r_2 (the detailed expression of these quantities as a function of the sphere centers and radii can be found in, e.g., [177]).

In order to ensure the visibility constraints, the distances among these two ellipses, and their distances with the image plane borders need to remain above some minimum threshold. These distance constraints can be simplified (with practically

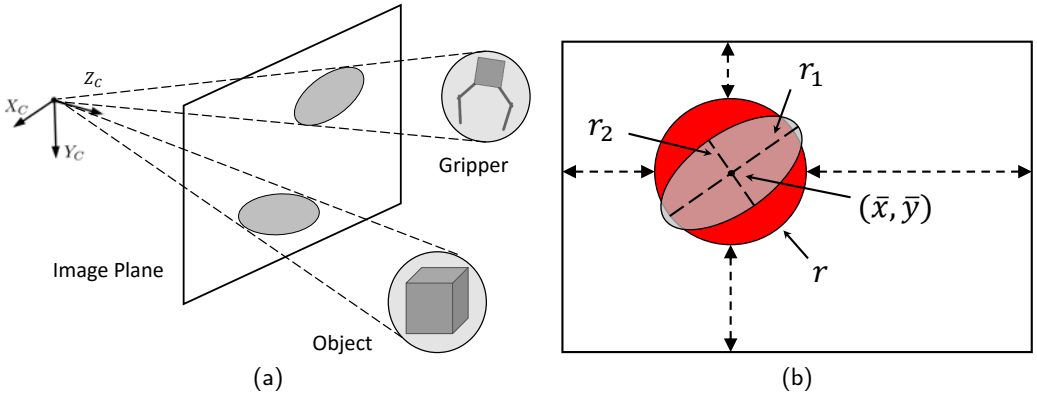


Figure 3.5: (a) The projection of the object and the gripper on the image plane. (b) The estimation of the ellipse as a circle using the p-norm of the major and minor axis.

negligible effects) by approximating each ellipse on the image plane as a circle with the same ellipse center $(\bar{x}, \bar{y})$ and 'bounding' radius

$$r = \sqrt[p]{r_1^p + r_2^p}. \tag{3.11}$$

Indeed, for a sufficiently large value of p (e.g. $p = 5$ as in our implementation), this expression provides a smooth approximation of the max function thus ensuring that the circles contain the original ellipses [178].

Let then $(\bar{x}_g, \bar{y}_g)$ and r_g be the center/radius of the circle approximating the projection of the gripper sphere on the image plane, and $(\bar{x}_o, \bar{y}_o)$ and r_o the center/radius for the object sphere, and consider the quantities

$$\boldsymbol{d}_g = \begin{bmatrix} \bar{x}_{max} - \bar{x}_g - r_g \\ \bar{x}_{min} + \bar{x}_g - r_g \\ \bar{y}_{max} - \bar{y}_g - r_g \\ \bar{y}_{min} + \bar{y}_g - r_g \end{bmatrix} \tag{3.12}$$

and

$$\boldsymbol{d}_o = \begin{bmatrix} \bar{x}_{max} - \bar{x}_o - r_o \\ \bar{x}_{min} + \bar{x}_o - r_o \\ \bar{y}_{max} - \bar{y}_o - r_o \\ \bar{y}_{min} + \bar{y}_o - r_o \end{bmatrix} \tag{3.13}$$

which collect the distances of both circles from the image plane boundaries (here represented by the intervals $[\bar{x}_{min}, \bar{x}_{max}]$ and $[\bar{y}_{min}, \bar{y}_{max}]$), as well as

$$d_{o,g}(s) = \sqrt{(\bar{x}_g - \bar{x}_o)^2 + (\bar{y}_g - \bar{y}_o)^2} - r_g - r_o \tag{3.14}$$

which represents the distance among the two circles. Figure 3.6 provides a visual illustration of $\boldsymbol{d}_g$, $\boldsymbol{d}_o$ and $d_{o,g}$.

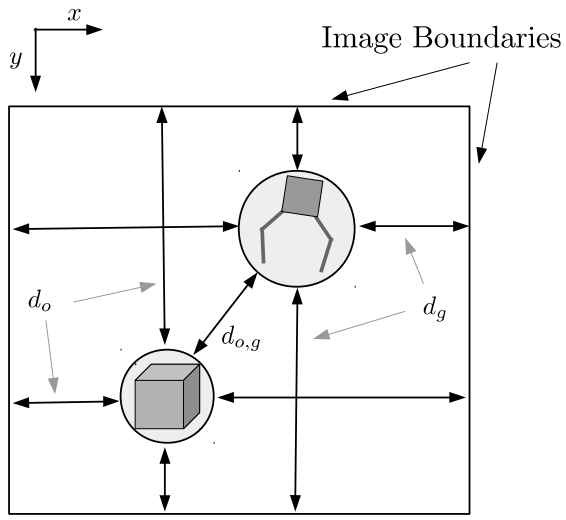


Figure 3.6: The definition of the visual constraints.

Since the aim is to keep both the gripper and the object inside the image plane and to avoid overlaps, we define a cumulative 'visibility' cost function as

$$h_V = k_v \sum_{i=1}^{4} \left(\frac{1}{d_{g,i}} + \frac{1}{d_{o,i}} \right) + k_v \frac{1}{d_{o,g}}, \quad k_v > 0, \tag{3.15}$$

which clearly grows unbounded as any of the considered distances approaches zero.

The camera can then be controlled so as to minimize h_V ensuring that all the constraints are respected. To this end, we need an expression for the rate of change of h_V in terms of the velocity of the camera $\boldsymbol{v}_c$. We start by noting that, from [177, 174], one has

$$\begin{cases} \dot{x}_g &= \boldsymbol{L}_{x_g}(\boldsymbol{v}_c - {}^c\boldsymbol{v}_g) \\ \dot{y}_g &= \boldsymbol{L}_{y_g}(\boldsymbol{v}_c - {}^c\boldsymbol{v}_g) \\ \dot{r}_{1,g} &= \boldsymbol{L}_{r_{1,g}}(\boldsymbol{v}_c - {}^c\boldsymbol{v}_g) \\ \dot{r}_{2,g} &= \boldsymbol{L}_{r_{2,g}}(\boldsymbol{v}_c - {}^c\boldsymbol{v}_g) \end{cases} \tag{3.16}$$

where $\boldsymbol{L}_{x_g} \in \mathbb{R}^{1\times 6}$, $\boldsymbol{L}_{y_g} \in \mathbb{R}^{1\times 6}$, $\boldsymbol{L}_{r_{1,g}} \in \mathbb{R}^{1\times 6}$, and $\boldsymbol{L}_{r_{2,g}} \in \mathbb{R}^{1\times 6}$ are the *interaction matrices* of the considered visual features (check [177] for details). Using (3.11)–(3.16) one also obtains

$$\dot{r}_g = \frac{\partial r_g}{\partial r_{1,g}} \dot{r}_{1,g} + \frac{\partial r_g}{\partial r_{2,g}} \dot{r}_{2,g} = \boldsymbol{L}_{r_g}(\boldsymbol{v}_c - {}^c\boldsymbol{v}_g).$$

Similarly, for the object ellipse we have

$$\begin{cases} \dot{x}_o &= \boldsymbol{L}_{x_o}\boldsymbol{v}_c \\ \dot{y}_o &= \boldsymbol{L}_{y_o}\boldsymbol{v}_c \\ \dot{r}_{1,o} &= \boldsymbol{L}_{r_{1,o}}\boldsymbol{v}_c \\ \dot{r}_{2,o} &= \boldsymbol{L}_{r_{2,o}}\boldsymbol{v}_c \end{cases} \tag{3.17}$$

where no compensation term is present since the object is assumed to be static[2] in the scene, and

$$\dot{r}_o = \frac{\partial r_o}{\partial r_{1,o}}\dot{r}_{1,o} + \frac{\partial r_o}{\partial r_{2,o}}\dot{r}_{2,o} = \boldsymbol{L}_{r_o}\boldsymbol{v}_c.$$

By then exploiting all the previous definitions together, we can obtain

$$\begin{aligned} \dot{h}_V = &\left(\frac{\partial h_V}{\partial x_o}\boldsymbol{L}_{x_o} + \frac{\partial h_V}{\partial y_o}\boldsymbol{L}_{y_o} + \frac{\partial h_V}{\partial r_o}\boldsymbol{L}_{r_o} + \frac{\partial h_V}{\partial x_g}\boldsymbol{L}_{x_g} + \frac{\partial h_V}{\partial y_g}\boldsymbol{L}_{y_g} + \frac{\partial h_V}{\partial r_g}\boldsymbol{L}_{r_g}\right)\boldsymbol{v}_c - \\ &- \left(\frac{\partial h_V}{\partial x_g}\boldsymbol{L}_{x_g} + \frac{\partial h_V}{\partial y_g}\boldsymbol{L}_{y_g} + \frac{\partial h_V}{\partial r_g}\boldsymbol{L}_{r_g}\right){}^c\boldsymbol{v}_g \end{aligned} \tag{3.18}$$

The control law governing the motion of the camera and ensuring the minimization of the visibility cost function h_V can then be written as

$$\boldsymbol{v}_c = -k_{h_v}\left(\frac{\partial h_V}{\partial x_o}\boldsymbol{L}_{x_o} + \frac{\partial h_V}{\partial y_o}\boldsymbol{L}_{y_o} + \frac{\partial h_V}{\partial r_o}\boldsymbol{L}_{r_o} + \frac{\partial h_V}{\partial x_g}\boldsymbol{L}_{x_g} + \frac{\partial h_V}{\partial y_g}\boldsymbol{L}_{y_g} + \frac{\partial h_V}{\partial r_g}\boldsymbol{L}_{r_g}\right)^T \tag{3.19}$$

where k_{h_v} is a positive control gain to be regulated.

As explained, this camera control strategy can yield a better performance than the simpler strategy of Sect. 3.2.2.1. First, instead of constraining the gripper and the object at some (arbitrary) fixed locations on the image plane, it just requires that both remain inside the FoV of the camera and do not overlap. Furthermore, the actual size of the camera/gripper on the image plane is taken into account whereas the controller in Sect. 3.2.2.1 approximated the camera/griper with a single point feature which is clearly sub-optimal. Note, however, that this strategy is more sensitive to noise as the whole gripper and object need to be tracked instead of simple markers. Moreover, constraining the gripper and the object to pre-defined positions on the image plane (while constraining for the motion of the camera) may decrease the confusion for the user and simplify the visual interface.

[2] In practice, if the object moves sufficiently slow, the proposed strategy can reactively compensate for this unmodeled motion as shown in the experiments.

3.2.3 Master Side

The final component of the shared control architecture is the design of some suitable force feedback cues which, as discussed in Sect. 3.2, are meant to inform the human operator about the feasibility of her/his commands $\boldsymbol{\lambda}$ w.r.t. possible constraints/requirements of the gripper/arm robotic system.

Following the classical *bilateral force-feedback* framework [179, 180], we then assume the presence of a *master device* upon which the operator can act for sending the commands $\boldsymbol{\lambda}$ to the *slave side* (the gripper/manipulator arm) and receiving force feedback cues (see Fig. 3.3). The master device is modeled as a generic (gravity pre-compensated) mechanical system

$$\boldsymbol{M}_m(\boldsymbol{x}_m)\dot{\boldsymbol{v}}_m + \boldsymbol{C}_m(\boldsymbol{x}_m, \boldsymbol{v}_m)\boldsymbol{v}_m = \boldsymbol{\tau} + \boldsymbol{\tau}_h \tag{3.20}$$

where $\boldsymbol{x}_m \in \mathbb{R}^m$ is the device configuration vector, $\boldsymbol{v}_m \in \mathbb{R}^m$ encodes its velocity, $\boldsymbol{M}_m(\boldsymbol{x}_m) \in \mathbb{R}^{m\times m}$ is the positive-definite and symmetric inertia matrix, $\boldsymbol{C}_m(\boldsymbol{x}_M, \boldsymbol{v}_m) \in \mathbb{R}^{m\times m}$ accounts for Coriolis/centrifugal terms, and $\boldsymbol{\tau}, \boldsymbol{\tau}_h \in \mathbb{R}^m$ are the control and human forces, respectively.

Two control modes can be used for the forward mapping from the master device to the slave side

- velocity-velocity (VV): velocities of the master device are coupled to the the velocities of the slave robot, modulo a constant roto-translation and scaling factor.
- position-velocity (PV): the configuration of the master device is used to command the velocities of the slave robot.

The advantage of the velocity-velocity control modality is the lower cognitive load required with respect to the position-velocity modality. In the velocity-velocity case the user's commands are directly reflected to the slave (e.g., the robot stops when the operator stops, and it moves when the operator moves). On the other hand, in position-velocity mode, this relationship is not as evident for the operator, who needs to think more carefully about the impact of her/his actions on the slave manipulator. However, limitations in the workspace of the master device are more evident in velocity-velocity mode than in position-velocity mode, with the former requiring clutching. The amount of clutching required is directly dependent on the scaling factor between the master/slave workspaces. If the scaling is high, the user will not be required to clutch a lot. However, this may results in fast and abrupt movements of the slave robot. On the other hand, if the scaling is low, the user will be able to more precisely and smoothly control the motion of the slave robot.

However, more clutching will be necessary. The details of both modes, along with the corresponding force cues, are illustrated below.

3.2.3.1 Velocity-velocity control

In the velocity-velocity control mode, the master/slave coupling is governed by

$$\boldsymbol{\lambda} = k_{\lambda,vel}\boldsymbol{S}_{\lambda,vel}\boldsymbol{v}_m, \tag{3.21}$$

where $\boldsymbol{S}_{\lambda,vel} \in \mathbb{R}^{n\times m}$ is a selection matrix that extracts the n components of $\boldsymbol{x}_m$ of interest and $k_{\lambda,vel}$ is a positive scaling factor. A button on the master handle works as a clutch. Clutching allows the user to pause the remote operation, move the haptic device to a more comfortable or suitable position, and then resume the control of the robot. This approach is commonly used to address issues of limited workspace on the master side.

Force cues are then defined simply as

$$\boldsymbol{\tau} = -\boldsymbol{B}_m\boldsymbol{v}_m + \boldsymbol{f}, \tag{3.22}$$

Here, $\boldsymbol{B}_m \in \mathbb{R}^{m\times m}$ is a positive definite damping matrix for stabilizing the haptic device. Vector $\boldsymbol{f} = [\dots f_i \dots]^T \in \mathbb{R}^m$ represents instead the force cues provided to the human operator: as explained, the design of these cues is aimed at informing the operator about the feasibility of her/his motion command $\boldsymbol{\lambda}$ w.r.t. possible constraints/limitations of the gripper/arm system such as, for instance, proximity to joint limits, to singularities, or to collisions with the surrounding environment. We now proceed to detail the general structure of the proposed cueing algorithm.

Let $\boldsymbol{q}_g \in \mathbb{R}^6$ be the joint configuration vector of the manipulator arm carrying the gripper, and $\boldsymbol{J}_g(\boldsymbol{q}_g) \in \mathbb{R}^{6\times 6}$ be the geometric Jacobian mapping joint velocities $\dot{\boldsymbol{q}}_g$ onto the gripper linear/angular velocities $\boldsymbol{v}_g = [\dot{\boldsymbol{p}}_g^T, \boldsymbol{\omega}_g^T]^T$. Let also $H(\boldsymbol{q}_g) \geq 0$ be a scalar cost function quantifying the proximity to any constraint of interest (the closer to a constraint, the larger the value of $H(\boldsymbol{q}_g)$). Intuitively, our idea is to design each f_i so as to inform about how much $H(\boldsymbol{q}_g)$ would decrease by moving along the i-th null-space direction $\boldsymbol{n}_i$. Such a force feedback can then assist the operator in understanding (i) which directions $\boldsymbol{n}_i$ are close to be unfeasible because of the robot constraints (thanks to the magnitude of f_i) and (ii) how to actuate the direction $\boldsymbol{n}_i$ for moving away from the robot constraints (thanks to the sign of f_i). To this end, by analysing the time variation of $H(\boldsymbol{q}_g)$ at steady-state (i.e., when the primary task has been regulated), we get

$$\begin{aligned}\dot{H}(\boldsymbol{q}_g) &= \frac{\partial H(\boldsymbol{q}_g)}{\partial \boldsymbol{q}_g}\dot{\boldsymbol{q}}_g \\ &= \frac{\partial H(\boldsymbol{q}_g)}{\partial \boldsymbol{q}_g}\boldsymbol{J}_g^{-1}\boldsymbol{v}_g \\ &= \frac{\partial H(\boldsymbol{q}_g)}{\partial \boldsymbol{q}_g}\boldsymbol{J}_g^{-1}\boldsymbol{N}\boldsymbol{\lambda} \\ &= k_{\lambda,vel}\frac{\partial H(\boldsymbol{q}_g)}{\partial \boldsymbol{q}_g}\boldsymbol{J}_g^{-1}\boldsymbol{N}\boldsymbol{S}_{\lambda,vel}\boldsymbol{v}_m\end{aligned} \tag{3.23}$$

The force cues informing the user about the directions $\boldsymbol{v}_m^*$ which minimize $H(\boldsymbol{q}_g)$ can simply be designed as the negative of the gradient relating $H(\boldsymbol{q}_g)$ to $\boldsymbol{v}_m$ such that

$$\boldsymbol{f} = \boldsymbol{v}_m^* = -\left(k_{\lambda,vel}\frac{\partial H(\boldsymbol{q}_g)}{\partial \boldsymbol{q}_g}\boldsymbol{J}_g^{-1}(\boldsymbol{q}_g)\boldsymbol{N}\boldsymbol{S}_{\lambda,vel}\right)^T, \tag{3.24}$$

Remark. *While other designs of the force cues $\boldsymbol{f}$ are possible, the described design is the only one which guarantees the minimization of $H(\boldsymbol{q}_g)$ also in case not all directions $\boldsymbol{v}_{m,i}^*$ of $\boldsymbol{v}_m^*$ are actuated. To show this, assume that the user actuates a particular set $\boldsymbol{S}\boldsymbol{v}_m^*$ of the desired directions $\boldsymbol{v}_m^*$ instead of following the whole vector ($\boldsymbol{S}$ is a diagonal selection matrix). By plugging $\boldsymbol{S}\boldsymbol{v}_m^*$ into (3.23), we get*

$$\dot{H}(\boldsymbol{q}_g) = -\left(k_{\lambda,vel}\frac{\partial H(\boldsymbol{q}_g)}{\partial \boldsymbol{q}_g}\boldsymbol{J}_g^{-1}(\boldsymbol{q}_g)\boldsymbol{N}\boldsymbol{S}_{\lambda,vel}\right)\boldsymbol{S}\left(k_{\lambda,vel}\frac{\partial H(\boldsymbol{q}_g)}{\partial \boldsymbol{q}_g}\boldsymbol{J}_g^{-1}(\boldsymbol{q}_g)\boldsymbol{N}\boldsymbol{S}_{\lambda,vel}\right)^T. \tag{3.25}$$

Since $\boldsymbol{S}$ is diagonal and positive semi-definite, the right-hand side of (3.25) is always non-positive, thus ensuring that $\dot{H}(\boldsymbol{q}_g) \leq 0$, regardless of the particular form of $\boldsymbol{S}$ (i.e., of which directions of $\boldsymbol{v}_m^*$ are actuated). Therefore, the user may choose to follow the cues given by $\boldsymbol{v}_m^*$ along some directions and to resist them along other directions. In all cases, $H(\boldsymbol{q}_g)$ is guaranteed to not increase (and, in general, to decrease).

3.2.3.2 Position-Velocity Control

In the position-velocity control mode, the human control actions are implemented by setting

$$\boldsymbol{\lambda} = k_{\lambda,pos}\boldsymbol{S}_{\lambda,pos}\boldsymbol{x}_m, \tag{3.26}$$

with $\boldsymbol{S}_{\lambda,pos} \in \mathbb{R}^{n\times m}$ being a selection matrix and $k_{\lambda,pos}$, a positive scaling factor. Note that the difference w.r.t. (3.21) is in the coupling of $\boldsymbol{\lambda}$ with the master configuration $\boldsymbol{x}_m$ instead of the master velocity $\boldsymbol{v}_m$. This coupling then allows the operator

to directly control the speed along the n null-space directions $\boldsymbol{n}_i$ by adjusting the position of the master device. The force feedback is instead designed as

$$\boldsymbol{\tau} = -\boldsymbol{B}_m \dot{\boldsymbol{x}}_m - \boldsymbol{K}_m \boldsymbol{x}_m + \boldsymbol{f}. \tag{3.27}$$

where $\boldsymbol{K}_m \in \mathbb{R}^{m \times m}$ is a positive definite diagonal matrix meant to implement a 'soft spring' centered at the device rest position. By means of this spring, the user will be provided with a perception of the distance from a zero-commanded velocity, see also [171, 172, 22, 181, 182]. The force cues vector $\boldsymbol{f}$ is designed in a similar approach to that of sect. 3.2.3.1. However, the variation of $H(\boldsymbol{q}_g)$ is here mapped to the position of the master device $\boldsymbol{x}_m$ instead of its velocity. At steady-state, (3.23) becomes

$$\dot{H}(\boldsymbol{q}_g) = k_{\lambda,pos} \frac{\partial H(\boldsymbol{q}_g)}{\partial \boldsymbol{q}_g} \boldsymbol{J}_g^{-1} \boldsymbol{N} \boldsymbol{S}_{\lambda,pos} \boldsymbol{x}_m. \tag{3.28}$$

The desired configuration, $\boldsymbol{x}_m^*$, of the master device (mapped to velocity commands $\boldsymbol{\lambda}^*$) which minimizes $H(\boldsymbol{q}_g)$ can be defined as

$$\boldsymbol{x}_m^* = -\left(k_{\lambda,pos} \frac{\partial H(\boldsymbol{q}_g)}{\partial \boldsymbol{q}_g} \boldsymbol{J}_g^{-1}(\boldsymbol{q}_g) \boldsymbol{N} \boldsymbol{S}_{\lambda,pos} \right)^T, \tag{3.29}$$

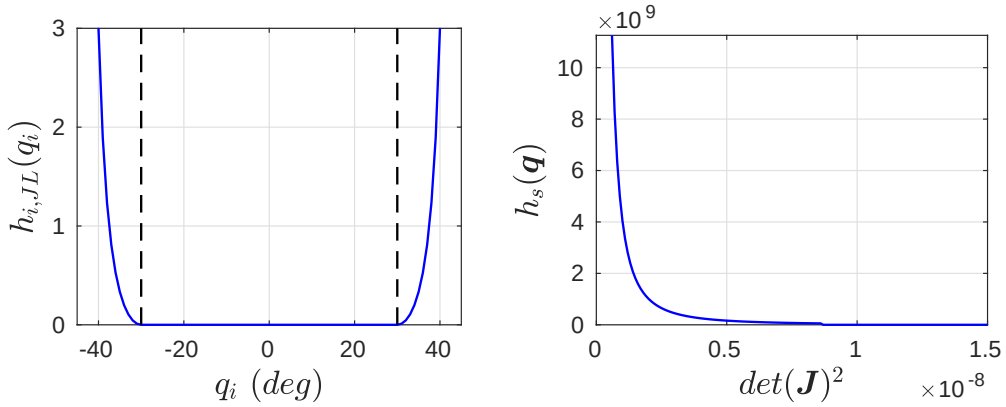
We design the force cues $\boldsymbol{f}$ such that

$$\boldsymbol{f} = \boldsymbol{K}_f \left(\boldsymbol{x}_m^* - \boldsymbol{x}_m \right), \tag{3.30}$$

where $\boldsymbol{K}_f \in \mathbb{R}^{m \times m}$ is a positive definite diagonal matrix for implementing a spring-like action driving the user toward the desired configuration $\boldsymbol{x}_m^*$.

3.2.4 Kinematic Constraints

Joint limits and singular configurations of both, the manipulator with the camera and that with the gripper, are considered as the two kinematic constraints which may negatively impact the execution of the task. We now proceed to describe the cost functions associated with each constrain for a generic 6-DoF manipulator of joint configuration vector $\boldsymbol{q} = [... \; q_i \; ...] \in \mathbb{R}^6$.



(a) Joint limits cost ($q_{th} = 10\ deg$) (b) Singularity cost ($\delta = 10^{-8}$)

Figure 3.7: The potentials associated with the joint limits and singularities of the manipulator.

Joint limits The cost describing the proximity to the limits of a joint $i \in [1, \ldots, 6]$ is defined as

$$h_{i,J}(q_i) = k_J \begin{cases} tan\left(\frac{\pi}{2}\left(1 - \frac{q_{\text{i,max}} - q_i}{q_{\text{th}}}\right)\right)^2 \\ \qquad\qquad \text{if} \quad q_{\text{i,max}} - q_{\text{th}} < q_i < q_{\text{i,max}}, \\ tan\left(\frac{\pi}{2}\left(1 - \frac{q_i - q_{\text{i,min}}}{q_{\text{th}}}\right)\right)^2 \\ \qquad\qquad \text{if} \quad q_{\text{i,min}} < q_i < q_{\text{i,min}} + q_{\text{th}}, \\ 0 \qquad\qquad\qquad\qquad \text{otherwise}, \end{cases} \tag{3.31}$$

where q_i is the i-th joint value, ($q_{\text{i,min}}$, $q_{\text{i,max}}$) are the i-th min/max joint limits, k_J is a positive constant, and q_{th} is the range (from the limit) in which the cost function is active. As shown in Fig. 3.7a, the joint cost $h_{i,J}(\boldsymbol{q})$ is zero in the middle of the joint range and it grows to infinity at the joint limits.

The total cost function describing the proximity to the joint limits of the manipulator is then defined as

$$h_J = \sum_i h_{i,J}(\boldsymbol{q})\,. \tag{3.32}$$

Singularities The singularity of a serial manipulator results in a loss of rank in its jacobian matrix. The determinant of this jacobian, which goes to zero when the jacobian loses rank, can thus be used as a measure of the robot's proximity to a singular configuration. Inspired by the preliminary implementations of [183], the

cost describing the proximity to singularities can be defined as

$$h_S(\boldsymbol{q}) = k_S \begin{cases} tan\left(\frac{\pi}{2}\left(1 - \frac{\det(\boldsymbol{J}(\boldsymbol{q}))^2}{\delta}\right)\right)^2 \\ \qquad\qquad \text{if} \;\; 0 < \det(\boldsymbol{J}(\boldsymbol{q}))^2 < \delta, \\ 0 \qquad\qquad \text{otherwise.} \end{cases} \tag{3.33}$$

where k_S is a positive gain, and δ is a positive threshold. As shown in Fig. 3.7b, the singularity cost $h_S(\boldsymbol{q})$ grows to infinity as $\det(\boldsymbol{J}(\boldsymbol{q}))^2 \to 0$, and it goes gradually to zero as $\det(\boldsymbol{J}(\boldsymbol{q}))^2 \to \delta$.

The total cost function H used in (3.24) and (3.30) to define the haptic cues provided to the user can then be calculated from the described individual cost functions depending on what constraints are to be considered. For example, in the case described in Sec. 3.2.2.1, where the motion of the manipulator with the camera is directly coupled to the motion of that with the gripper, the operator can be informed through haptic cues about the proximity of any of the two manipulators to kinematic constraints and H can be defined as

$$H(\boldsymbol{q}_g, \boldsymbol{q}_c) = h_{J,g}(\boldsymbol{q}_g) + h_{S,g}(\boldsymbol{q}_g) + h_{J,c}(\boldsymbol{q}_c) + h_{S,c}(\boldsymbol{q}_c). \tag{3.34}$$

Its partial derivative w.r.t. $\boldsymbol{q}_g$ is

$$\frac{\partial H(\boldsymbol{q}_g, \boldsymbol{q}_c)}{\partial \boldsymbol{q}_g} = \frac{\partial h_{J,g}}{\partial \boldsymbol{q}_g} + \frac{\partial h_{S,g}}{\partial \boldsymbol{q}_g} + \frac{\partial h_{J,c}}{\partial \boldsymbol{q}_g} + \frac{\partial h_{S,c}}{\partial \boldsymbol{q}_g}. \tag{3.35}$$

Note that $\partial h_{J,c}/\partial \boldsymbol{q}_g$ and $\partial h_{S,c}/\partial \boldsymbol{q}_g$ are not zero since, following (3.10), the motion of the camera is coupled to that of the gripper. Therefore, the user can be informed about how much the gripper motion would push the camera towards/away any of the camera constraints (which is essential to ensure that the camera can follow the motion and keep the object and the gripper in the FoV). The resulting force cues informing the user about the proximity of the system to kinematic constraints can then be defined following (3.24) and (3.30).

3.2.5 Passivity Analysis

Enforcing passivity is is a very popular way of guaranteeing a stable closed-loop behaviour when dealing with teleoperation systems. Passivity has been extensively studied and applied to haptic feedback teleoperation systems, especially when time delays are present [184]. In this section, our aim is to analyze the passivity of the considered teleoperation system subject to the two control modalities introduced in Secs. 3.2.3.1 and 3.2.3.2.

By considering the two velocity-controlled robots equipped with the gripper and the camera as simple integrators, the total energy of the system can be written as

$$\mathcal{V}(\boldsymbol{l}_m, \boldsymbol{q}_g, \boldsymbol{q}_c) = \frac{1}{2}\boldsymbol{l}_m^{\mathrm{T}}\boldsymbol{M}_m^{-1}\boldsymbol{l}_m + H(\boldsymbol{q}_g, \boldsymbol{q}_c) \tag{3.36}$$

where $\boldsymbol{l}_m = \boldsymbol{M}_m\boldsymbol{v}_m$ is the haptic device momentum, $\boldsymbol{q}_g$, $\boldsymbol{q}_c \in \mathbb{R}^6$ are the generalized coordinates of the two slave manipulators, respectively, and $H(\boldsymbol{q}_g, \boldsymbol{q}_c)$, introduced in (3.34), is the potential energy associated with the system constraints. We can thus analyze passivity by showing that the closed-loop system equations can be put in port-Hamiltonian form [185]. Formally, a port-Hamiltonian system (PHS) can be represented by the following set of equations

$$\begin{cases} \dot{\boldsymbol{x}} = [J(\boldsymbol{x}) - R(\boldsymbol{x})]\dfrac{\partial \mathcal{V}}{\partial \boldsymbol{x}} + g(\boldsymbol{x})\,\boldsymbol{u} \\ \boldsymbol{y} = g^{\mathrm{T}}(\boldsymbol{x})\dfrac{\partial \mathcal{V}}{\partial x} \end{cases} \tag{3.37}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ represents the system state, $\mathcal{V}(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ is the Hamiltonian function, namely the sum of system energies, $J(\boldsymbol{x}) = -J(\boldsymbol{x})^{\mathrm{T}}$ represents the internal interconnection, $R(\boldsymbol{x}) \geq 0$ the internal dissipation, $g(\boldsymbol{x})$ the input matrix, $\boldsymbol{u}$ the system input, and $\boldsymbol{y}$ the system output. It is easy to show that for PHS the following inequality holds

$$\boldsymbol{u}^{\mathrm{T}}\boldsymbol{y} = \dot{\mathcal{V}}(\boldsymbol{x}) + \frac{\partial^{\mathrm{T}}\mathcal{V}}{\partial \boldsymbol{x}} R(\boldsymbol{x}) \frac{\partial \mathcal{V}}{\partial \boldsymbol{x}} \geq \dot{\mathcal{V}}(\boldsymbol{x}). \tag{3.38}$$

Equation (3.38) establishes the inherent passivity condition of a PHS with respect to the input-output pair $(\boldsymbol{u}, \boldsymbol{y})$ with storage function $\mathcal{V}(\boldsymbol{x})$.

Two control modalities have been proposed for controlling the gripper:

- Velocity-Velocity mode (Sec. 3.2.3.1).
- Position-Velocity mode (Sec. 3.2.3.2).

On the other hand, we also described two control modes for the camera:

- Point-based control mode (Sec. 3.2.2.1).
- Moment-based control mode (Sec. 3.2.2.2).

In the following, we show that the velocity-velocity control modality for the gripper leads to a closed-loop PHS formulation regardless the control mode of the camera. For the analysis, we consider the primary tasks in (3.3) and (3.10) to have reached a steady state ($\boldsymbol{s} \to \boldsymbol{s}_d$ and $\bar{\boldsymbol{p}} \to \bar{\boldsymbol{p}}_d$).

Point-Based Control Mode: In the point-based camera control mode, the motion of the camera is directly proportional to that of the gripper as per (3.10). The only input to the system is then $\dot{\boldsymbol{q}}_g$ and the corresponding closed-loop system can be written as follows

$$\begin{bmatrix} \dot{\boldsymbol{l}}_m \\ \dot{\boldsymbol{q}}_g \end{bmatrix} = \left(\begin{bmatrix} 0 & -k_{\lambda,vel} \left(\boldsymbol{J}_g^{\dagger} \boldsymbol{N} \right)^T \\ k_{\lambda,vel} \boldsymbol{J}_g^{\dagger} \boldsymbol{N} & 0 \end{bmatrix} + - \begin{bmatrix} \boldsymbol{B}_m & 0 \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} \nabla_{\boldsymbol{l}_m} \mathcal{V} \\ \nabla_{\boldsymbol{q}_g} \mathcal{V} \end{bmatrix} + \begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix} \boldsymbol{f}_h. \tag{3.39}$$

Moment-Based Control Mode: In the moment-based control mode, the camera manipulator is commanded by (3.19) and $\dot{\boldsymbol{q}}_g$ and $\dot{\boldsymbol{q}}_c$ are independent inputs to the system. In this case, the closed-loop system can be written as follows

$$\begin{bmatrix} \dot{\boldsymbol{l}}_m \\ \dot{\boldsymbol{q}}_g \\ \dot{\boldsymbol{q}}_c \end{bmatrix} = \left(\begin{bmatrix} 0 & -k_{\lambda,vel} \left(\boldsymbol{J}_g^{\dagger} \boldsymbol{N} \right)^T & 0 \\ k_{\lambda,vel} \boldsymbol{J}_g^{\dagger} \boldsymbol{N} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + - \begin{bmatrix} \boldsymbol{B}_m & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & k_{h_v} \end{bmatrix} \right) \begin{bmatrix} \nabla_{\boldsymbol{l}_m} \mathcal{V} \\ \nabla_{\boldsymbol{q}_g} \mathcal{V} \\ \nabla_{\boldsymbol{q}_c} \mathcal{V} \end{bmatrix} + \begin{bmatrix} \mathbf{1} \\ 0 \\ 0 \end{bmatrix} \boldsymbol{f}_h. \tag{3.40}$$

Therefore, since in both cases the closed-loop systems can be put in a PHS form (the resulting interconnection and dissipation matrices being always skew-symmetric and positive semi-definite, respectively), one can conclude the passivity of the both w.r.t. the pair $(\boldsymbol{v}_m, \boldsymbol{f}_h)$ with energy function $\mathcal{V}(\boldsymbol{l}_m, \boldsymbol{q}_A, \boldsymbol{q}_B)$ as sought.

As for the position-velocity control mode of the gripper (Sec. 3.2.3.2), the closed-loop system can not be put into a PHS form due to the different master-slave mapping. However, in order to guarantee stability in this case, we can exploit the passive set-position modulation (PSPM) approach from [186], a very general and flexible framework for guaranteeing stability (passivity) of the master side and of the closed-loop teleoperation system. To recall, the force feedback provided to the user in the position-velocity gripper control modality follows (3.30) and has the form

$$\boldsymbol{\tau} = -\boldsymbol{B}_m \dot{\boldsymbol{x}}_m - \boldsymbol{K}_m \boldsymbol{x}_m + \boldsymbol{K}_f \left(\boldsymbol{x}_m^* - \boldsymbol{x}_m \right). \tag{3.41}$$

Let $\bar{\boldsymbol{x}}_m^*$ be the PSPM version of $\boldsymbol{x}_m^*$ that is sampled and sent from the manipulators to the haptic interface through the (possibly non-ideal) communication

channel. Exploiting the PSPM action, the final passive implementation of $\boldsymbol{\tau}$ then becomes

$$\boldsymbol{\tau} = -\boldsymbol{B}_m \dot{\boldsymbol{x}}_m - \boldsymbol{K}_m \boldsymbol{x}_m + \boldsymbol{K}_f \left(\bar{\boldsymbol{x}}_m^* - \boldsymbol{x}_m \right). \tag{3.42}$$

When the force feedback has the general form (3.30), the PSPM action suitably modulates $\bar{\boldsymbol{x}}_m^*$ so as to always ensure the input/output stability (passivity) of the master device. This is then sufficient for guaranteeing stability (passivity) of the overall bilateral teleoperation, see [186] for more details.

3.3 Trajectory-Based Shared Control

3.3.1 Overview

In the previous sections, an instantaneous shared-control architecture for commanding a dual arm system was introduced. The user controls a subset of the DoF of the manipulator equipped with the gripper and is informed through haptic feedback about the proximity of both manipulators (the one with the camera and the one with the gripper) to dangerous configurations and guided away from them. However, for grasping applications, the user is mainly interested in reaching the final grasping pose regardless of the trajectory followed to reach that pose. To this end, we hereby extend the proposed approach to a shared-planning architecture where the user and autonomy jointly define a trajectory to be followed by the manipulator. This allows the user to be in control of some DoF of interest of a whole future trajectory (from the current to the final pose of the gripper), while an autonomous component optimizes the remaining trajectory DoF against the system constraints. Therefore, the human operator can gain a better awareness of the expected motion of the manipulators over a future time window, and better react/correct whenever necessary.

We now then summarize the main features of the proposed trajectory-based approach. The main goals are: (*i*) let a human operator modify *online* the approaching trajectory towards the object to be grasped via the master device, (*ii*) let an autonomous algorithm verify that the operator's commands respect all the possible constraints of the robotic system and, in case it does not, modify them accordingly, (*iii*) provide the human operator with online force cues informing about any discrepancy between the commanded trajectory and the actual one (attenuated by the autonomous algorithm in order to avoid the robot constraints), thus informing about the future consequences of the operator's actions and, finally, (*iv*) let an autonomous algorithm control the camera motion so as to keep a suitable vantage

point w.r.t. the observed scene (i.e., both the gripper and the object) and act on any redundant degree of freedom in order to optimize a given cost function.

We now proceed to detail the components of the proposed trajectory-based shared control architecture. For ease of exposition, we first focus on the technical details concerning the chosen trajectory representation and system constraints, and then illustrate the general architecture in Sec. 3.3.4.

3.3.2 Trajectory Representation

In our implementation, we represent the trajectory of the system using parametric curves for the 6-DOF pose trajectories of both the gripper and the camera. In particular, we exploit the classical B-splines [187, 172] for representing the position component of each trajectory. As for the orientation, we take advantage of a B-spline parametrization of quaternion trajectories proposed in [188]. The use of B-splines is motivated by their powerful geometric and numerical properties which make them a very popular choice for planning and optimization purposes. Nevertheless the strategy presented here could readily be applied to other trajectory parametrizations as long as they are amenable for an online evaluation of all necessary partial derivatives.

3.3.2.1 Definition of a Pose Trajectory

Given a sequence of scalar *knots* $(s_1, \dots, s_k), s_i \in \mathbb{R}$ and a set of l *control points* $\boldsymbol{t} = (\boldsymbol{t}_1, \dots, \boldsymbol{t}_l) \in \mathbb{R}^{3l}$ with $\boldsymbol{t}_i \in \mathbb{R}^3$ and $k, l \in \mathbb{N}$, a *position trajectory* is defined as:

$$\boldsymbol{p}(\boldsymbol{t},\, s) = \sum_{i=1}^{l} \boldsymbol{t}_i B_i(s), \tag{3.43}$$

where s is the independent variable representing the position along the curve, and B_i are the p-order B-spline basis functions [187, 189]. For simplicity, we assume that $s \in [0, 1]$ with $s = 0$ being the starting point and $s = 1$ the ending point of the trajectory.

Following [188], for the orientation we consider the same knot sequence $(s_1, \dots, s_k)$ and a set of l (quaternion) control points $\boldsymbol{\rho} = (\boldsymbol{\rho}_1, \dots, \boldsymbol{\rho}_l) \in \mathbb{S}^{3l}$, with $\boldsymbol{\rho}_i \in \mathbb{S}^3$, and we define the *orientation trajectory* as:

$$\begin{aligned} \boldsymbol{\eta}(\boldsymbol{\rho},\, s) &= \bigotimes_{i=1}^{l} \exp[\log(\boldsymbol{\rho}_{i-1}^{*} \otimes \boldsymbol{\rho}_i)\widetilde{B}_i(s)] \\ &= \bigotimes_{i=1}^{l} ({}^{i-1}\boldsymbol{\rho}_i)^{\widetilde{B}_i(s)} = \bigotimes_{i=1}^{l} \widetilde{\boldsymbol{\rho}}_i(s), \end{aligned} \tag{3.44}$$

where $\otimes$ is the quaternion product, $\boldsymbol{\rho}_i^*$ is the conjugate quaternion of $\boldsymbol{\rho}_i$, i.e. such that $\boldsymbol{\rho}_i^* \otimes \boldsymbol{\rho}_i = \boldsymbol{\rho}_i \otimes \boldsymbol{\rho}_i^* = (1,0,0,0)$, $\widetilde{B}_i(s) = \sum_{j=i}^{l} B_i(s)$ are the B-splines *cumulative basis functions* [188], and we defined ${}^{i-1}\boldsymbol{\rho}_i = \boldsymbol{\rho}_{i-1}^* \otimes \boldsymbol{\rho}_i$, and $\widetilde{\boldsymbol{\rho}}_i = ({}^{i-1}\boldsymbol{\rho}_i)^{\widetilde{B}_i(s)}$. The exponential and logarithmic maps are defined as follows (see, e.g. [190]):

$$\log(\boldsymbol{\rho}) : \boldsymbol{\rho} = (\rho_0, \boldsymbol{\rho}_v) \in \mathbb{S}^3 \mapsto \frac{\boldsymbol{\rho}_v}{\operatorname{sinc}\left(\operatorname{atan2}(\|\boldsymbol{\rho}_v\|, \rho_0)\right)} \in \mathbb{R}^3 \tag{3.45}$$

$$\exp(\boldsymbol{v}) : \boldsymbol{v} \in \mathbb{R}^3 \mapsto (\cos(\|\boldsymbol{v}\|), \operatorname{sinc}(\|\boldsymbol{v}\|)\boldsymbol{v}) \in \mathbb{S}^3 \tag{3.46}$$

with $\operatorname{sinc}(\alpha) = \sin(\alpha)/\alpha$. B-spline quaternion curves inherit some important B-spline properties, such as differentiability and local controllability [188], which motivate their use in this work for representing an orientation trajectory.

A pair $\boldsymbol{\gamma}(\boldsymbol{t}, \boldsymbol{\rho}, s) = (\boldsymbol{p}(\boldsymbol{t}, s), \boldsymbol{\eta}(\boldsymbol{\rho}, s)) \in \mathbb{R}^3 \times \mathbb{S}^3$ is then defined as a *pose (Cartesian) trajectory* (expressed in $\mathcal{F}_o$) to be tracked by the manipulators. For notational ease, we will often drop the dependency on the control points $\boldsymbol{t}$ and $\boldsymbol{\rho}$, and just highlight the dependency on the trajectory parameter s, i.e., $\boldsymbol{\gamma}(s) = (\boldsymbol{p}(s), \boldsymbol{\eta}(s))$. We also note that each pose trajectory $\boldsymbol{\gamma}(s)$ implicitly defines a corresponding trajectory in joint space $\boldsymbol{q}_{\boldsymbol{\gamma}}(s)$ that realizes it[3].

3.3.2.2 Velocity Twist of a Pose Trajectory

In this subsection we provide a detailed expression of the body-frame velocity twist $\boldsymbol{v}(\boldsymbol{t}, \boldsymbol{\rho}, s) = (\dot{\boldsymbol{p}}(\boldsymbol{t}, \boldsymbol{\rho}, s), \boldsymbol{\omega}(\boldsymbol{t}, \boldsymbol{\rho}, s)) \in \mathbb{R}^6$, with $\dot{\boldsymbol{p}}$ and $\boldsymbol{\omega}$ being the body-frame linear/angular velocities associated to $\boldsymbol{\gamma}(\boldsymbol{t}, \boldsymbol{\rho}, s)$ and generated by the control point velocities $\dot{\boldsymbol{t}}$, $\dot{\boldsymbol{\rho}}$, and traveling speed $\dot{s}$. This relationship will allow both the autonomous component and the human operator to affect the shape of $\boldsymbol{\gamma}(\boldsymbol{t}, \boldsymbol{\rho}, s)$ by acting on $\boldsymbol{t}$, $\boldsymbol{\rho}$, and the position along $\boldsymbol{\gamma}(\boldsymbol{t}, \boldsymbol{\rho}, s)$ by acting on s.

We first note that, from (3.43) and (3.44), one has

$$\dot{\boldsymbol{p}} = \sum_{i=1}^{l} \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{t}_i} \dot{\boldsymbol{t}}_i + \frac{\partial \boldsymbol{p}}{\partial s} \dot{s} \tag{3.47}$$

$$\dot{\boldsymbol{\eta}} = \sum_{i=1}^{l} \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\rho}_i} \dot{\boldsymbol{\rho}}_i + \frac{\partial \boldsymbol{\eta}}{\partial s} \dot{s}. \tag{3.48}$$

The control point velocities are designed as

$$\dot{\boldsymbol{t}}_i = \boldsymbol{R}_i \boldsymbol{\mu}_i \tag{3.49}$$

$$\dot{\boldsymbol{\rho}}_i = \frac{1}{2} \boldsymbol{\rho}_i \otimes \begin{bmatrix} 0 \\ \boldsymbol{\xi}_i \end{bmatrix} \tag{3.50}$$

[3]Since each manipulator is non-redundant (6-dofs), there exists a *finite* number of joint trajectories realizing a given pose trajectory $\boldsymbol{\gamma}(s)$. We define $\boldsymbol{q}_{\boldsymbol{\gamma}}(s)$ as the (unique) joint trajectory in this finite set such that $\boldsymbol{q}_{\boldsymbol{\gamma}}(0) = \boldsymbol{q}(t_0)$ (i.e., matching the initial joint configuration of the manipulator under consideration).

where $\boldsymbol{R}_i$ is the rotation matrix corresponding to $\boldsymbol{\rho}_i$, and $(\boldsymbol{\mu}_i, \boldsymbol{\xi}_i)$ are the linear and angular velocities of the control point i expressed in its own reference frames. The traveling speed $\dot{s}$ is, instead, assumed to be directly controllable, i.e.,

$$\dot{s} = \sigma. \tag{3.51}$$

For simplicity of notation, we also define $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_l) \in \mathbb{R}^{3l}$, $\boldsymbol{\xi} = (\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_l) \in \mathbb{R}^{3l}$, and $\boldsymbol{u} = (\boldsymbol{\mu}, \boldsymbol{\xi}) \in \mathbb{R}^{6l}$ as the collective control point linear/angular velocities. The quantity $\boldsymbol{u}$ then represents the (velocity) control input available for modifying the shape of the pose trajectory $\boldsymbol{\gamma}(\boldsymbol{t}, \boldsymbol{\rho}, s)$, and σ the (velocity) control input for traveling along the trajectory.

Exploiting (3.43)–(3.49), the first term in (3.47) can be rewritten as

$$\sum_{i=1}^{l} \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{t}_i} \dot{\boldsymbol{t}}_i = \sum_{i=1}^{l} B_i(s) \dot{\boldsymbol{t}}_i = \sum_{i=1}^{l} B_i(s)\, \boldsymbol{R}_i \boldsymbol{\mu}_i = \boldsymbol{R}(\boldsymbol{\eta}(s)) \boldsymbol{J}_{\boldsymbol{p},\boldsymbol{\mu}} \boldsymbol{\mu} \tag{3.52}$$

where $\boldsymbol{R}(\boldsymbol{\eta}(s))$ stands for the rotation matrix associated to the quaternion $\boldsymbol{\eta}(s)$, and the Jacobian $\boldsymbol{J}_{\boldsymbol{p},\boldsymbol{\mu}}$ defined as

$$\boldsymbol{J}_{\boldsymbol{p},\boldsymbol{\mu}} = \boldsymbol{R}(\boldsymbol{\eta}(s))^T \begin{bmatrix} B_1 \boldsymbol{R}_1 & \cdots & B_l \boldsymbol{R}_l \end{bmatrix} \in \mathbb{R}^{3 \times 3l}$$

maps the control point velocities $\boldsymbol{\mu}$ to a 'body-frame' linear velocity. Analogously,

$$\frac{\partial \boldsymbol{p}}{\partial s} \dot{s} = \sum_{i=1}^{l} \frac{\mathrm{d}B_i}{\mathrm{d}s}(s) \boldsymbol{t}_i \dot{s} = \boldsymbol{R}(\boldsymbol{\eta}(s)) \boldsymbol{J}_{\boldsymbol{p},s} \sigma \tag{3.53}$$

with the Jacobian $\boldsymbol{J}_{\boldsymbol{p},s} \in \mathbb{R}^3$ defined as

$$\boldsymbol{J}_{\boldsymbol{p},s} = \boldsymbol{R}(\boldsymbol{\eta}(s))^T \sum_{i=1}^{l} \frac{\mathrm{d}B_i}{\mathrm{d}s}(s) \boldsymbol{t}_i.$$

Plugging (3.52–3.53) into (3.47) then yields

$$\dot{\boldsymbol{p}} = \boldsymbol{R}(\boldsymbol{\eta}(s))(\boldsymbol{J}_{\boldsymbol{p},\boldsymbol{\mu}} \boldsymbol{\mu} + \boldsymbol{J}_{\boldsymbol{p},s} \sigma) \tag{3.54}$$

which shows that the body-frame linear velocity $\dot{\boldsymbol{p}}$ associated to the trajectory $\boldsymbol{\gamma}(\boldsymbol{t}, \boldsymbol{\rho}, s)$ can be expressed in terms of the available control inputs as

$$\dot{\boldsymbol{p}} = \boldsymbol{J}_{\boldsymbol{p},\boldsymbol{\mu}} \boldsymbol{\mu} + \boldsymbol{J}_{\boldsymbol{p},s} \sigma. \tag{3.55}$$

One can proceed in a similar way for what concerns the body-frame angular velocity $\boldsymbol{\omega}$. Indeed, the orientation spline dynamics takes the expression

$$\dot{\boldsymbol{\eta}}(s) = \frac{1}{2} \boldsymbol{\eta} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} = \frac{1}{2} \boldsymbol{\eta} \otimes \begin{bmatrix} 0 \\ \boldsymbol{J}_{\boldsymbol{\eta},\boldsymbol{\xi}} \boldsymbol{\xi} + \boldsymbol{J}_{\boldsymbol{\eta},s} \sigma \end{bmatrix} \tag{3.56}$$

where, however, the derivation of the Jacobians $\boldsymbol{J}_{\eta,\xi} \in \mathbb{R}^{3\times 3l}$ and $\boldsymbol{J}_{\eta,s} \in \mathbb{R}^3$ is more involved and their full expressions are reported in A.2. Therefore, the body-frame angular velocity $\boldsymbol{\omega}$ associated to the trajectory $\boldsymbol{\gamma}(\boldsymbol{t}, \boldsymbol{\rho}, s)$ can be expressed in terms of the available control inputs as

$$\boldsymbol{\omega} = \boldsymbol{J}_{\eta,\xi}\boldsymbol{\xi} + \boldsymbol{J}_{\eta,s}\sigma. \tag{3.57}$$

Using (3.55–3.57), we can then summarize the expression for the body-frame twist $\boldsymbol{v}$ as

$$\boldsymbol{v} = \begin{bmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_{p,\mu} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{J}_{\eta,\xi} \end{bmatrix} \boldsymbol{u} + \begin{bmatrix} \boldsymbol{J}_{p,s} \\ \boldsymbol{J}_{\eta,s} \end{bmatrix} \sigma = \boldsymbol{J}_{\boldsymbol{u}}\boldsymbol{u} + \boldsymbol{J}_s\sigma. \tag{3.58}$$

This differential expression will be fundamental in the following developments since it relates the available control inputs $\boldsymbol{u}$ and σ to the body-frame twist $\boldsymbol{v}$. We finally note that the twist $\boldsymbol{v}$ also determines the joint velocities $\dot{\boldsymbol{q}}_\gamma$ that realize it via the (inverse of the) usual geometric Jacobian of the manipulator under consideration as

$$\dot{\boldsymbol{q}}_\gamma = \boldsymbol{J}_G^{-1}(\boldsymbol{q}_\gamma)\boldsymbol{v}, \qquad \boldsymbol{J}_G \in \mathbb{R}^{6\times 6}. \tag{3.59}$$

3.3.2.3 Final Considerations

We conclude by highlighting that two instantiations of the pose trajectory $\boldsymbol{\gamma}(\boldsymbol{t}, \boldsymbol{\rho}, s)$ are considered in this work: a pair $\boldsymbol{\gamma}_g(\boldsymbol{t}_g, \boldsymbol{\rho}_g, s) = (\boldsymbol{p}_g(\boldsymbol{t}_g, s), \boldsymbol{\eta}_g(\boldsymbol{\rho}_g, s))$ to be followed by the gripper manipulator, and a pair $\boldsymbol{\gamma}_c(\boldsymbol{t}_c, \boldsymbol{\rho}_c, s) = (\boldsymbol{p}_c(\boldsymbol{t}_c, s), \boldsymbol{\eta}_c(\boldsymbol{\rho}_c, s))$ to be followed by the camera manipulator, with associated the two joint trajectories $\boldsymbol{q}_{\gamma,g}(s)$ and $\boldsymbol{q}_{\gamma,c}(s)$. Likewise, two instances of all the various quantities introduced in the previous sections (e.g., body-frame twist and associated Jacobians) must also be considered. To this end, throughout the following developments a subscript g or c will be appended to any relevant quantity whenever necessary.

Finally, for notational ease, we collect all the available control points (for both the camera and gripper pose trajectories) in a single parameter vector

$$\boldsymbol{\theta} = (\boldsymbol{t}_g, \boldsymbol{\rho}_g, \boldsymbol{t}_c, \boldsymbol{\rho}_c) \in \mathbb{R}^{6l} \times \left(\mathbb{S}^3\right)^{2l} \tag{3.60}$$

and, similarly, all the available control point velocities into a single control vector

$$\boldsymbol{u}_{\boldsymbol{\theta}} = (\boldsymbol{u}_g, \boldsymbol{u}_c) \in \mathbb{R}^{12l}. \tag{3.61}$$

For the reader's convenience, we conclude by summarizing in Table 3.1 the role and properties of the main quantities introduced.

Quantity	Role
$\boldsymbol{t} = (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_l) \in \mathbb{R}^{3l}$	the l control points of the position trajectory
$\boldsymbol{\rho} = (\boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_l) \in \mathbb{S}^{3l}$	the l control points of the orientation trajectory
$s \in [0,\ 1]$	the curve parameter of a position/orientation trajectory
$\boldsymbol{p}(\boldsymbol{t},\ s) \in \mathbb{R}^3$	the position trajectory
$\boldsymbol{\eta}(\boldsymbol{\rho},\ s) \in \mathbb{S}^3$	the orientation trajectory
$\boldsymbol{\gamma}(\boldsymbol{t}, \boldsymbol{\rho},\ s) = (\boldsymbol{p}(\boldsymbol{t},\ s), \boldsymbol{\eta}(\boldsymbol{\rho},\ s)) \in \mathbb{R}^3 \times \mathbb{S}^3$	the pose trajectory
$\boldsymbol{q}_{\boldsymbol{\gamma}}(\boldsymbol{t}, \boldsymbol{\rho},\ s) \in \mathbb{R}^6$	the joint trajectory that realizes the pose trajectory $\boldsymbol{\gamma}(\boldsymbol{t}, \boldsymbol{\rho},\ s)$
$\sigma \in \mathbb{R}$	traveling speed along the pose trajectory
$\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_l) \in \mathbb{R}^{3l}$	body-frame linear velocities of the control points $\boldsymbol{t}$
$\boldsymbol{\xi} = (\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_l) \in \mathbb{R}^{3l}$	body-frame angular velocities of the control points $\boldsymbol{\rho}$
$\boldsymbol{u} = (\boldsymbol{\mu},\ \boldsymbol{\xi}) \in \mathbb{R}^{6l}$	collective linear/angular control point velocities
$\dot{\boldsymbol{p}} \in \mathbb{R}^3$	body-frame linear velocity of a pose trajectory induced by $\boldsymbol{\mu}$ and σ
$\boldsymbol{\omega} \in \mathbb{R}^3$	body-frame angular velocity of a pose trajectory induced by $\boldsymbol{\xi}$ and σ
$\boldsymbol{v} = (\dot{\boldsymbol{p}},\ \boldsymbol{\omega}) \in \mathbb{R}^6$	body-frame velocity twist of a pose trajectory induced by $\boldsymbol{\mu}$, $\boldsymbol{\xi}$ and σ
$\boldsymbol{\theta} = (\boldsymbol{t}_g,\ \boldsymbol{\rho}_g,\ \boldsymbol{t}_c,\ \boldsymbol{\rho}_c) \in \mathbb{R}^{6l} \times \left(\mathbb{S}^3\right)^{2l}$	stack of all the position/orientation control points for the two gripper and camera pose trajectories
$\boldsymbol{u}_{\boldsymbol{\theta}} = (\boldsymbol{u}_g,\ \boldsymbol{u}_c) \in \mathbb{R}^{12l}$	stack of all the position/orientation control points velocities for the two gripper and camera pose trajectories

Table 3.1: The table summarizes the nomenclature of the main variables used in this section.

3.3.3 System Constraints

As mentioned in 3.3.1, the autonomous component of the shared control architecture aims at keeping the two trajectories for the camera and gripper manipulators away from the system constraints for optimizing the overall performance. The following constraints have been considered in this work: joint limit and singularity avoidance for both manipulators, a visibility constraint for ensuring that the gripper and the object always remain in the camera FoV and do not overlap (for allowing an online

reconstruction of the gripper/object pose in the camera frame) and, a *regularity* constraint aimed at minimizing the trajectory lengths and ensuing that the control points are evenly spread along the trajectories.

The proximity to each of the constraints is encoded in a cost function as described in Sec. 3.2.4 and Sec. 3.2.2.2. To this end, consider a joint trajectory $\boldsymbol{q}_{\boldsymbol{\gamma},g}(s)$ associated with the gripper pose trajectory $\boldsymbol{\gamma}_g(s)$ (the camera case being analogous). Let the cost function $h_{J,g}(\boldsymbol{q}_{\boldsymbol{\gamma},g}(s))$ associated with the joint limits of the gripper manipulator for each $s \in [0,\, 1]$ be defined as in (3.32). Similarly, $h_{S,g}(\boldsymbol{q}_{\boldsymbol{\gamma},g}(s))$ (see (3.33)) is associated with the singularities of the gripper manipulator. As for the manipulator carrying the camera, $h_{J,c}(\boldsymbol{q}_{\boldsymbol{\gamma},c}(s))$ and $h_{S,c}(\boldsymbol{q}_{\boldsymbol{\gamma},c}(s))$ encode the proximity to its joint limits and singularities respectively. Finally, $h_V(\boldsymbol{q}_{\boldsymbol{\gamma},g}(s), \boldsymbol{q}_{\boldsymbol{\gamma},c}(s))$ (see (3.15)) is associated with the visibility constraints.

Joint Limits: The joint limit cost of the gripper manipulator over the whole trajectory can be defined as

$$H_{J,g}(\boldsymbol{t}_g,\, \boldsymbol{\rho}_g) = \int_0^1 h_{J,g}(\boldsymbol{q}_{\boldsymbol{\gamma},g}(\boldsymbol{t}_g,\, \boldsymbol{\rho}_g,\, s))\mathrm{d}s \tag{3.62}$$

where we highlighted the dependency of the total cost $H_{J,g}$ on the control point positions/orientations $\boldsymbol{t}_g$ and $\boldsymbol{\rho}_g$.

In view of the following developments, we also explicit the relationship between the rate of change of $H_{J,g}$ and the control inputs $\boldsymbol{u}_g$ (gripper control point velocities). Exploiting (3.58)–(3.59), and noting that $\boldsymbol{u}_g$ is independent of s, one has

$$\begin{aligned}\dot{H}_{J,g} &= \int_0^1 \frac{\partial h_{J,g}}{\partial \boldsymbol{q}_{\boldsymbol{\gamma},g}} \boldsymbol{J}_{G,g}^{-1} \boldsymbol{J}_{\boldsymbol{u},g} \boldsymbol{u}_g \mathrm{d}s = \left(\int_0^1 \frac{\partial h_{J,g}}{\partial \boldsymbol{q}_{\boldsymbol{\gamma},g}} \boldsymbol{J}_{G,g}^{-1} \boldsymbol{J}_{\boldsymbol{u},g} \mathrm{d}s \right) \boldsymbol{u}_g \\ &= \nabla_{\boldsymbol{u}_g}^T H_{J,g} \boldsymbol{u}_g,\end{aligned} \tag{3.63}$$

where, by an abuse of notation, we denote by $\nabla_{\boldsymbol{u}_g}^T H_{J,g}$ the mapping between $\boldsymbol{u}_g$ and $\dot{H}_{J,g}$.

By considering an analogous cost $H_{J,c}(\boldsymbol{t}_c,\, \boldsymbol{\rho}_c)$ for the camera trajectory, the final total cost accounting for joint limits for both manipulator arms is then

$$H_J(\boldsymbol{\theta}) = H_{J,g}(\boldsymbol{t}_g,\, \boldsymbol{\rho}_g) + H_{J,c}(\boldsymbol{t}_c,\, \boldsymbol{\rho}_c) \tag{3.64}$$

with rate of change

$$\dot{H}_J = \begin{bmatrix} \nabla_{\boldsymbol{u}_g}^T H_{J,g} & \nabla_{\boldsymbol{u}_c}^T H_{J,c} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_g \\ \boldsymbol{u}_c \end{bmatrix} = \nabla_{\boldsymbol{u}_{\boldsymbol{\theta}}}^T H_J \boldsymbol{u}_{\boldsymbol{\theta}}. \tag{3.65}$$

Singularities: Similarly, the singularity cost function over the whole trajectory is then

$$H_{S,g}(\boldsymbol{t}_g,\ \boldsymbol{\rho}_g) = \int_0^1 h_{S,g}(\boldsymbol{q}_{\gamma,g}(\boldsymbol{t}_g,\ \boldsymbol{\rho}_g,\ s))\mathrm{d}s, \tag{3.66}$$

and its rate of change takes the expression

$$\begin{aligned}\dot{H}_{S,g} &= \int_0^1 \frac{\partial h_{S,g}}{\partial \boldsymbol{q}_{\gamma,g}} \boldsymbol{J}_{G,g}^{-1} \boldsymbol{J}_{\boldsymbol{u},g} \boldsymbol{u}_g \mathrm{d}s = \left(\int_0^1 \frac{\partial h_{S,g}}{\partial \boldsymbol{q}_{\gamma,g}} \boldsymbol{J}_{G,g}^{-1} \boldsymbol{J}_{\boldsymbol{u},g} \mathrm{d}s \right) \boldsymbol{u}_g \\ &= \nabla_{\boldsymbol{u}_g}^T H_{S,g} \boldsymbol{u}_g.\end{aligned} \tag{3.67}$$

By considering the analogous singularity cost $H_{S,c}(\boldsymbol{t}_c,\ \boldsymbol{\rho}_c)$ for the camera manipulator, the total singularity cost is

$$H_S(\boldsymbol{\theta}) = H_{S,g}(\boldsymbol{t}_g,\ \boldsymbol{\rho}_g) + H_{S,c}(\boldsymbol{t}_c,\ \boldsymbol{\rho}_c) \tag{3.68}$$

with rate of change

$$\dot{H}_S = \begin{bmatrix} \nabla_{\boldsymbol{u}_g}^T H_{S,g} & \nabla_{\boldsymbol{u}_c}^T H_{S,c} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_g \\ \boldsymbol{u}_c \end{bmatrix} = \nabla_{\boldsymbol{u}_{\boldsymbol{\theta}}}^T H_S \boldsymbol{u}_{\boldsymbol{\theta}}. \tag{3.69}$$

Visibility Constraints: The same goes for the cost function associated with the visibility constraints where the total cost over the trajectory is

$$H_V(\boldsymbol{\theta}) = \int_0^1 h_V(\boldsymbol{\theta},\ s)\mathrm{d}s \tag{3.70}$$

and by exploiting the definitions in Sec. 3.2.2.2 with (3.58), and considering that $\boldsymbol{u}_c$ and $\boldsymbol{u}_g$ are as usual independent of s, we can finally obtain

$$\begin{aligned}\dot{H}_V = &\left(\int_0^1 \left(\frac{\partial h_V}{\partial x_o} \boldsymbol{L}_{x_o} + \frac{\partial h_V}{\partial y_o} \boldsymbol{L}_{y_o} + \frac{\partial h_V}{\partial r_o} \boldsymbol{L}_{r_o} + \right.\right. \\ &\left.\left. + \frac{\partial h_V}{\partial x_g} \boldsymbol{L}_{x_g} + \frac{\partial h_V}{\partial y_g} \boldsymbol{L}_{y_g} + \frac{\partial h_V}{\partial r_g} \boldsymbol{L}_{r_g} \right) \boldsymbol{J}_{\boldsymbol{u},c} \mathrm{d}s \right) \boldsymbol{u}_c \\ &- \left(\int_0^1 \left(\frac{\partial h_V}{\partial x_g} \boldsymbol{L}_{x_g} + \frac{\partial h_V}{\partial y_g} \boldsymbol{L}_{y_g} + \frac{\partial h_V}{\partial r_g} \boldsymbol{L}_{r_g} \right) {}^c\boldsymbol{M}_g \boldsymbol{J}_{\boldsymbol{u},g} \mathrm{d}s \right) \boldsymbol{u}_g \\ &= \begin{bmatrix} \nabla_{\boldsymbol{u}_c}^T H_V & \nabla_{\boldsymbol{u}_g}^T H_V \end{bmatrix} \boldsymbol{u}_{\boldsymbol{\theta}} = \nabla_{\boldsymbol{u}_{\boldsymbol{\theta}}}^T H_V \boldsymbol{u}_{\boldsymbol{\theta}}.\end{aligned} \tag{3.71}$$

3.3.3.1 Trajectory Length

If the number of control points l is sufficiently large, multiple different trajectories may be, in general, compatible with all the previous constraints and still satisfy the user commands. One can then exploit this additional redundancy for optimizing any additional performance index: in this work, we considered the minimization of the total linear and angular length of both manipulator trajectories as an additional objective. In order to reduce the computational burden, and exploiting the

properties of the B-spline parameterization, we approximate the trajectory length with the sum of the inter-(linear and angular) distances between consecutive control points. In particular, focusing on the translational part of the gripper trajectory (the camera case being equivalent), the following scalar cost is considered

$$H_{t,g}(\boldsymbol{t}_g) = \frac{1}{2}\sum_{i=1}^{l-1}(\boldsymbol{t}_{i+1,g} - \boldsymbol{t}_{i,g})^2 \tag{3.72}$$

with rate of change given by

$$\dot{H}_{t,g} = \sum_{i=1}^{l-1}(\boldsymbol{t}_{i+1,g} - \boldsymbol{t}_{i,g})^T(\boldsymbol{R}_{i+1,g}\boldsymbol{\mu}_{i+1,g} - \boldsymbol{R}_{i,g}\boldsymbol{\mu}_{i,g}) = \boldsymbol{J}_{t,g}\boldsymbol{u}_g.$$

where

$$\boldsymbol{J}_{t,g} = \\ \begin{bmatrix}(\boldsymbol{t}_{1,g} - \boldsymbol{t}_{2,g})^T\,\boldsymbol{R}_{1,g} & \cdots & (2\boldsymbol{t}_{i,g} - \boldsymbol{t}_{i-1,g} - \boldsymbol{t}_{i+1,g})^T\,\boldsymbol{R}_{i,g} & \cdots & (\boldsymbol{t}_{l,g} - \boldsymbol{t}_{l-1,g})^T\,\boldsymbol{R}_{l,g} & \boldsymbol{0}_{3l}^T\end{bmatrix}.$$

As for the angular part, we can exploit the fact that the inner product $\boldsymbol{\rho}_i^T\,\boldsymbol{\rho}_j$ between two quaternions $\boldsymbol{\rho}_i$ and $\boldsymbol{\rho}_j$ is equal to the scalar part of the quaternion $\boldsymbol{\rho}_i^* \otimes \boldsymbol{\rho}_j$ representing the relative rotation between $\boldsymbol{\rho}_i$ and $\boldsymbol{\rho}_j$. Consequently, $\boldsymbol{\rho}_i^T\,\boldsymbol{\rho}_j = \cos\frac{\alpha}{2}$ where α is the angle between $\boldsymbol{\rho}_i$ and $\boldsymbol{\rho}_j$. The quantity $(\boldsymbol{\rho}_i^T\,\boldsymbol{\rho}_j)^2$ is therefore equal to 1 if and only if $\boldsymbol{\rho}_i = \pm\,\boldsymbol{\rho}_j$ (i.e. if the two quaternions represent the same rotation), and it vanishes if the quaternions are 180° apart. Because of these considerations, we consider the following cost function

$$H_{\rho,g}(\boldsymbol{\rho}_g) = \frac{1}{2}\sum_{i=1}^{l-1} 1 - (\boldsymbol{\rho}_{i+1,g}^T\,\boldsymbol{\rho}_{i,g})^2 \tag{3.73}$$

for measuring the angular length of a trajectory. Differentiating w.r.t. time yields

$$\begin{aligned}\dot{H}_{\rho,g} &= -\sum_{i=1}^{l-1}\boldsymbol{\rho}_{i+1,g}^T\dot{\boldsymbol{\rho}}_{i,g} + \boldsymbol{\rho}_{i,g}^T\dot{\boldsymbol{\rho}}_{i+1,g} \\ &= -\frac{1}{2}\sum_{i=1}^{l-1}\boldsymbol{\rho}_{i+1,g}^T\left(\boldsymbol{\rho}_{i,g}\otimes\begin{bmatrix}0\\ \boldsymbol{\xi}_{i,g}\end{bmatrix}\right) + \boldsymbol{\rho}_{i,g}^T\left(\boldsymbol{\rho}_{i+1,g}\otimes\begin{bmatrix}0\\ \boldsymbol{\xi}_{i+1,g}\end{bmatrix}\right) \\ &= -\frac{1}{2}\sum_{i=1}^{l-1}\boldsymbol{\rho}_{i+1,g}^T\boldsymbol{W}(\boldsymbol{\rho}_{i,g})\boldsymbol{\xi}_{i,g} + \boldsymbol{\rho}_{i,g}^T\boldsymbol{W}(\boldsymbol{\rho}_{i+1,g})\boldsymbol{\xi}_{i+1,g} \\ &= \boldsymbol{J}_{\rho,g}\boldsymbol{u}_g\end{aligned}$$

where

$$\boldsymbol{W}(\boldsymbol{\rho}) : \boldsymbol{\rho} = (\rho_0, \boldsymbol{\rho}_v) \in \mathbb{S}^3 \mapsto \begin{bmatrix}-\,\boldsymbol{\rho}_v^T \\ \rho_0\boldsymbol{I}_3 + [\boldsymbol{\rho}_v]_\times\end{bmatrix} \in \mathbb{R}^{4\times 3}$$

$$\boldsymbol{J}_{\rho,g} = \\ = -\tfrac{1}{2}\begin{bmatrix}\boldsymbol{0}_{3l}^T & \boldsymbol{\rho}_{2,g}^T\boldsymbol{W}(\boldsymbol{\rho}_{1,g}) & \cdots & (\boldsymbol{\rho}_{i-1,g} + \boldsymbol{\rho}_{i+1,g})^T\boldsymbol{W}(\boldsymbol{\rho}_{i,g}) & \cdots & \boldsymbol{\rho}_{l-1,g}^T\boldsymbol{W}(\boldsymbol{\rho}_{l,g})\end{bmatrix}.$$

By applying the same cost functions to the camera trajectory, i.e., $H_{t,c}(\boldsymbol{t}_c)$ and $H_{\rho,c}(\boldsymbol{\rho}_c)$, we can then define the total linear/angular gripper/camera trajectory cost as

$$H_{t,\rho}(\boldsymbol{\theta}) = H_{t,g}(\boldsymbol{t}_g) + H_{\rho,g}(\boldsymbol{\rho}_g) + H_{t,c}(\boldsymbol{t}_c) + H_{\rho,c}(\boldsymbol{\rho}_c) \tag{3.74}$$

with rate of change

$$\dot{H}_{t,\rho} = [\boldsymbol{J}_{t,g} \quad \boldsymbol{J}_{\rho,g} \quad \boldsymbol{J}_{t,c} \quad \boldsymbol{J}_{\rho,c}]\boldsymbol{u}_{\boldsymbol{\theta}} = \nabla^T_{\boldsymbol{u}_{\boldsymbol{\theta}}} H_{t,\rho} \boldsymbol{u}_{\boldsymbol{\theta}}. \tag{3.75}$$

3.3.3.2 Final Cost Function

The overall cost function, accounting for all the various constraints/requirements introduced so far, is simply defined as the sum of all the previous costs, i.e.,[4]

$$\begin{aligned} H(\boldsymbol{\theta}) =& H_{J,g}(\boldsymbol{t}_g,\ \boldsymbol{\rho}_g) + H_{J,c}(\boldsymbol{t}_c,\ \boldsymbol{\rho}_c) + H_{S,g}(\boldsymbol{t}_g,\ \boldsymbol{\rho}_g) + \\ &+ H_{S,c}(\boldsymbol{t}_c,\ \boldsymbol{\rho}_c) + H_V(\boldsymbol{\theta}) + H_{t,\rho}(\boldsymbol{\theta}), \end{aligned} \tag{3.76}$$

with rate of change given by

$$\begin{aligned} \dot{H} = &\left(\nabla^T_{\boldsymbol{u}_{\boldsymbol{\theta}}} H_{J,g} + \nabla^T_{\boldsymbol{u}_{\boldsymbol{\theta}}} H_{J,c} + \nabla^T_{\boldsymbol{u}_{\boldsymbol{\theta}}} H_{S,g} + \nabla^T_{\boldsymbol{u}_{\boldsymbol{\theta}}} H_{S,c}\right. \\ &\left. + \nabla^T_{\boldsymbol{u}_{\boldsymbol{\theta}}} H_V + \nabla^T_{\boldsymbol{u}_{\boldsymbol{\theta}}} H_{t,\rho}\right) \boldsymbol{u}_{\boldsymbol{\theta}} = \nabla^T_{\boldsymbol{u}_{\boldsymbol{\theta}}} H \boldsymbol{u}_{\boldsymbol{\theta}} \end{aligned} \tag{3.77}$$

3.3.4 The Overall Architecture

We can now proceed to illustrate how this architecture is implemented by leveraging the various components introduced in the previous Sections. We assume that the system is initialized with $s(t_0) = 0$ (both trajectories start at their initial pose) and with the parameters (control points) $\boldsymbol{\theta}(t_0)$ initialized such that (i) the pose trajectories $\boldsymbol{\gamma}_g(\boldsymbol{t}_g(t_0),\ \boldsymbol{\rho}_g(t_0),\ s(t_0))$ and $\boldsymbol{\gamma}_c(\boldsymbol{t}_c(t_0),\ \boldsymbol{\rho}_c(t_0),\ s(t_0))$ match the real initial poses of the gripper/camera manipulators and (ii) the two trajectories are sufficiently far from any constraint $\forall s \in [0,\ 1]$ (i.e., $H(\boldsymbol{\theta}(t_0))$ is sufficiently small).

A *forward interface* (Sect. 3.3.4.1) allows the operator to either travel along or modify some dofs of the gripper pose trajectory while an autonomous algorithm continuously optimizes both the gripper/camera trajectories in the null-space of the operator's commands in order to stay as far as possible from the considered constraints. This interface couples the master device configuration $\boldsymbol{x}_m$ with the traveling speed σ (see (3.51)) and with the control point velocities $\boldsymbol{u}_{\boldsymbol{\theta}}$ (see (3.61)). In particular, the interface will split $\boldsymbol{u}_{\boldsymbol{\theta}}$ into two orthgonal terms

$$\boldsymbol{u}_{\boldsymbol{\theta}} = \boldsymbol{u}_{\boldsymbol{\theta},H} + \boldsymbol{u}_{\boldsymbol{\theta},A}, \quad \boldsymbol{u}^T_{\boldsymbol{\theta},H} \boldsymbol{u}_{\boldsymbol{\theta},A} = 0, \tag{3.78}$$

[4] With an abuse of notation, we use H (which was also used in (3.34)) to define the total cost function encoding all the considered system constraints even if these constraints may change from one system to another.

with $\boldsymbol{u}_{\boldsymbol{\theta},H}$ representing the operator's modification to the trajectory shape and $\boldsymbol{u}_{\boldsymbol{\theta},A}$ the autonomous optimization of the trajectories (the orthogonality of these two terms will guarantee that these two actions remain independent as sought).

A *backward interface* (Sect. 3.3.4.2) informs instead the operator about the proximity to the constraints and how the operator could act in order to avoid them by producing a suitable haptic feedback $\boldsymbol{\tau}$ (see (3.20)).

3.3.4.1 Forward Interface

Traveling along the planned trajectory: in our implementation, the operator is given direct control over the traveling speed σ (see (3.51)) by coupling it with one dofs of the master device configuration $\boldsymbol{x}_m$ (see (3.20)). This is obtained by setting

$$\sigma = k_\sigma \boldsymbol{S}_\sigma \boldsymbol{x}_m \tag{3.79}$$

where $k_\sigma > 0$ is a suitable scaling factor and $\boldsymbol{S}_\sigma \in \mathbb{R}^{1\times m}$ a selection matrix that extracts the component of $\boldsymbol{x}_m$ used for commanding the traveling speed. The law (3.79) can also be easily complemented by any safety mechanism that sets $\sigma = 0$ whenever s is at the boundaries of the interval $[0,\ 1]$ and the user tries commanding a traveling speed that would make s leaving $[0,\ 1]$.

Modifying the planned trajectory: As explained before, the main idea of the algorithm is to provide an interface that allows the operator to affect a suitable subset of the gripper trajectory DoF while an autonomous algorithm concurrently optimizes the trajectory shape in the null-space of the operator's commands. While many different choices are possible depending on the context and tasks, we hereby rely on the architecture described in Sec. 3.2 where the focus is on the approaching (pre-grasping) phase towards the target object.

The *final* pose of the gripper trajectory is then constrained so as to always have the gripper pointing towards the object and let the user commanding the remaining four null-space motions of the gripper final pose. Since we use an *open uniform* knot vector (i.e. with p equal knots at each end of the knot span), the final pose of the gripper simply coincides with the last control point of the corresponding gripper trajectory, that is

$$\begin{bmatrix} \boldsymbol{t}_{l,g} \\ \boldsymbol{\rho}_{l,g} \end{bmatrix} = \boldsymbol{\gamma}_g(\boldsymbol{t}_g,\ \boldsymbol{\rho}_g,\ 1).$$

Following the machinery developed in Sec. 3.2, and particularly (3.8), the control law governing the motion of the last control point $(\boldsymbol{t}_{l,g}, \boldsymbol{\rho}_{l,g})$ is defined as

$$\boldsymbol{u}_{l,g} = \begin{bmatrix} \boldsymbol{\mu}_{l,g} \\ \boldsymbol{\xi}_{l,g} \end{bmatrix} = k_g \boldsymbol{L}_s^\dagger \boldsymbol{e}_z + \boldsymbol{N}\boldsymbol{\lambda}. \tag{3.80}$$

The control law realizes the sought result: the approaching direction of the final gripper pose towards the object is regulated at $\boldsymbol{e}_z$ (first term), and the gripper can still be maneuvered along four null-space directions spanning $\boldsymbol{N} = \begin{bmatrix}\boldsymbol{n}_1 & \dots & \boldsymbol{n}_4\end{bmatrix}^T$ by actuating the velocity-commands $\boldsymbol{\lambda} = \begin{bmatrix}\lambda_1 & \dots & \lambda_4\end{bmatrix}^T$ (second term).

In the context of this work, we can assume w.l.o.g. that the control points $\boldsymbol{\theta}(t_0)$ are initialized so as to already have the gripper pointing towards the object at the final pose, so that, during the task execution, the first term in (3.80) starts and remains practically negligible. Therefore, for our purposes (3.80) reduces to

$$\boldsymbol{u}_{l,g} = \boldsymbol{N}\boldsymbol{\lambda}. \tag{3.81}$$

The operator is then given control over the four null-space motion directions by coupling the commands $\boldsymbol{\lambda}$ with four DoF of the configuration of the master device $\boldsymbol{x}_m$ in a position-velocity coupling (see Sect. 3.2.3.2) such that

$$\boldsymbol{\lambda} = k_\lambda \boldsymbol{S}_\lambda \boldsymbol{x}_m \tag{3.82}$$

where k_λ a suitable scaling factor and $\boldsymbol{S}_\lambda \in \mathbb{R}^{4\times m}$ a selection matrix that extracts the four components of $\boldsymbol{x}_m$ of interest.

Plugging (3.82) into (3.81), the forward interface between the operator's commands and the final gripper pose is then

$$\boldsymbol{u}_{l,g} = k_\lambda \boldsymbol{N}\boldsymbol{S}_\lambda \boldsymbol{x}_m. \tag{3.83}$$

We note that, obviously, the coupling (3.83) only affects the 6 linear/angular velocities $\boldsymbol{u}_{l,g}$ associated to the control points $(\boldsymbol{t}_{l,g}, \boldsymbol{\rho}_{l,g})$, while the remaining $12l-6$ control point velocities in $\boldsymbol{u}_{\boldsymbol{\theta},H}$ are not affected. For completeness, and in view of the next developments, it is then useful to restate (3.83) in terms of the whole control vector $\boldsymbol{u}_{\boldsymbol{\theta},H}$ to be plugged in (3.78) as

$$\boldsymbol{u}_{\boldsymbol{\theta},H} = k_\lambda \boldsymbol{S}_H \boldsymbol{N}\boldsymbol{S}_\lambda \boldsymbol{x}_m \tag{3.84}$$

where

$$\boldsymbol{S}_H = \begin{bmatrix}\underbrace{\boldsymbol{-0-}}_{6(l-1)} & \boldsymbol{I} & \underbrace{\boldsymbol{-0-}}_{6l}\end{bmatrix}^T \in \mathbb{R}^{12l\times 6}$$

maps the user commands to the correct selection of control point velocities in $\boldsymbol{u}_{\boldsymbol{\theta},H}$.

Concurrent trajectory optimization: As explained, while the user affects the shape and position along the gripper/camera trajectories, an *autonomous* component continuously optimizes the resulting trajectories by minimizing the cumulative

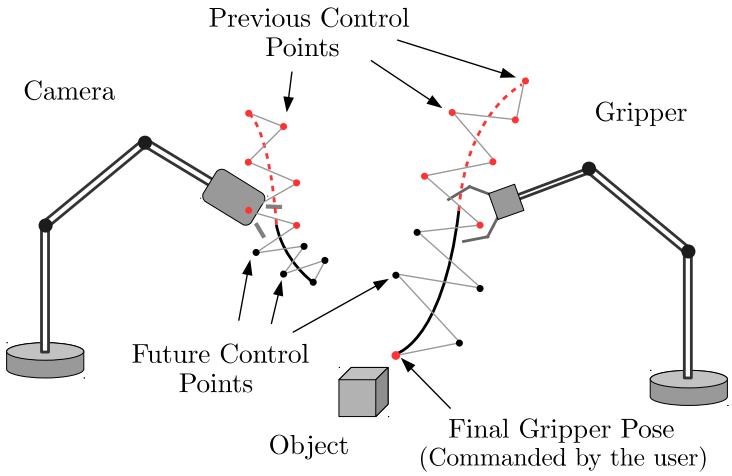


Figure 3.8: Autonomous Trajectory Modification: The controller acts on the control points which do not impact the trajectory already covered by the manipulators. Moreover, the controller does not act on the last control point which is directly comannded by the user.

cost function $H(\boldsymbol{\theta})$ in (3.76) for staying as far as possible from the system constraints. From (3.77), this is obtained by setting

$$\boldsymbol{u}_{\boldsymbol{\theta},A} = -k_A \boldsymbol{S}_A \nabla_{\boldsymbol{u}_{\boldsymbol{\theta}}} H(\boldsymbol{\theta}) \tag{3.85}$$

where $k_A > 0$ is an optimization gain and $\boldsymbol{S}_A \in \mathbb{R}^{12l \times 12l}$ a suitable diagonal selection matrix. The purpose of matrix $\boldsymbol{S}_A$ is to prevent any coupling between the minimization of $H(\boldsymbol{\theta})$ and (i) the control points associated to the final gripper pose (which must only be affected by the human operator via (3.84)), and (ii) the control points associated to the portion of the camera/gripper trajectories that has already been traveled along by the user, see also Fig. 3.8. The rationale behind this last requirement is to allow for the possibility of moving *backwards* (by setting a negative traveling speed $\sigma < 0$) along the exact same camera/gripper trajectories that the user has traveled along when moving forward from the initial pose[5].

Let then $\bar{s} \in [0,\ 1]$ represent the *current* position that the user has reached along the gripper/camera trajectories. By exploiting the properties of B-splines [189], it is possible to determine the index $1 \leq \iota_g \leq l$ such that the 'past' gripper trajectory $\boldsymbol{\gamma}_g(s)$ in the interval $s \in [0,\ \bar{s}]$ is only affected by control points $(\boldsymbol{t}_{i,g},\ \boldsymbol{\rho}_{i,g})$, $i \in \{1 \dots \iota_g\}$. Control points $(\boldsymbol{t}_{i,g},\ \boldsymbol{\rho}_{i,g})$, $i \in \{\iota_g + 1 \dots l\}$, on the other, only affect

[5]Clearly one could also relax this requirement and optimize the 'past' camera/gripper trajectories as well. However, we empirically found that the users feel more comfortable in moving backwards along the same path that they have traveled along in the forward direction.

the 'future' gripper trajectory $\boldsymbol{\gamma}_g(s)$ in the interval $s \in [\bar{s}, 1]$. By analogous considerations, one also obtains a corresponding index $1 \leq \iota_c \leq l$ for the camera trajectory $\boldsymbol{\gamma}_c(s)$. The selection matrix $\boldsymbol{S}_A$ must then filter any action affecting the control points in the sets $\{1 \dots \iota_g\}$ and $\{1 \dots \iota_c\}$, as well as the l-th gripper control point $(\boldsymbol{t}_{l,g},\ \boldsymbol{\rho}_{l,g})$ which is directly controlled by the human operator via (3.84)). This can be easily obtained by taking $\boldsymbol{S}_A$ as a diagonal selection matrix with all zeros on the main diagonal apart from the entries $\{6\iota_g + 1 \dots 6(l-1)\}$ and $\{6(l+\iota_c)+1 \dots 12l\}$ which are set to one.

We finally note that, by construction, $\boldsymbol{S}_A^T \boldsymbol{S}_H = \boldsymbol{0}$ thus implying that $\boldsymbol{u}_{\boldsymbol{\theta},A}^T \boldsymbol{u}_{\boldsymbol{\theta},H} = 0$ as desired: the optimization action (3.85) does not affect the human commands (3.84) and viceversa.

3.3.4.2 Backward Interface

As described in the previous sections, the role of the backward interface is to inform the operator about the proximity to any system constraint and on how to act in order to avoid them via a suitable design of the force feedback $\boldsymbol{\tau}$ on the master device (3.20). We note that the chosen forward interface (3.79) –(3.84) couples the master *configuration* with the control point *velocities*, thus effectively realizing a position-velocity coupling between the master and slave sides. This is actually the same position-velocity coupling used in Sec. 3.2.3.2 for the instantaneous case. While the user was directly commanding the gripper velocity in (3.26), she/he is here acting on the control points velocities (3.84). The force feedback is thus designed as in (3.27) such that

$$\boldsymbol{\tau} = -\boldsymbol{B}_m \dot{\boldsymbol{x}}_m - \boldsymbol{K}_m \boldsymbol{x}_m + \boldsymbol{f} \tag{3.86}$$

where $\boldsymbol{B}_m \in \mathbb{R}^{m \times m}$ is a positive definite damping matrix for stabilizing the haptic device and $\boldsymbol{K}_m \in \mathbb{R}^{m \times m}$ is a positive definite diagonal matrix meant to implement a 'soft spring' centered at the device rest position[6].

Vector $\boldsymbol{f} \in \mathbb{R}^m$ represents instead the force cues informing the operator about the system constraints and needs to be designed depending on the particular application. In our case, one can note that the relationship (3.84) maps the configuration space of the master device onto the space of control point velocities. One can then invert (3.84) for defining a 'virtual' configuration $\boldsymbol{x}_m^*$

$$\boldsymbol{x}_m^* = -(\boldsymbol{S}_H \boldsymbol{N} \boldsymbol{S}_\lambda)^\dagger \nabla_{\boldsymbol{u}_{\boldsymbol{\theta}}} H(\boldsymbol{\theta}) = -\boldsymbol{S}_\lambda^T \boldsymbol{N}^T \boldsymbol{S}_H^T \nabla_{\boldsymbol{u}_{\boldsymbol{\theta}}} H(\boldsymbol{\theta})$$

where, thanks to structure of (3.84) ($\boldsymbol{S}_H$ and $\boldsymbol{N}$ having orthonormal columns and $\boldsymbol{S}_\lambda$ having orthonormal rows) the pseudoinvrse simply reduces to a transpose. The

[6] Therefore, by means of this spring the user is provided with a perception of the distance from a zero-commanded velocity.

'virtual' configuration $\boldsymbol{x}_m^*$ represents where the user should place the master device in order to make the control points of the gripper/camera trajectories moving along the negative gradient of $H(\boldsymbol{\theta})$ (and thus having the slave side moving away from the constraints). Therefore, we design the force cues $\boldsymbol{f}$ as

$$\boldsymbol{f} = \boldsymbol{K}_f(\boldsymbol{x}_m^* - \boldsymbol{x}_m), \tag{3.87}$$

with $\boldsymbol{K}_f \in \mathbb{R}^{m \times m}$ being a positive definite diagonal matrix, for implementing a spring-like action that will continuously cue the operator towards the virtual configuration $\boldsymbol{x}_m^*$.

3.3.4.3 Final Remarks

Summarizing, by acting on the master configuration $\boldsymbol{x}_m$, the operator can control the traveling speed along the gripper/camera trajectories via (3.79), and part of the gripper final pose via (3.84) (in particular, the four null-space motion directions (3.7)). At the same time, an autonomous algorithm continuously optimizes both trajectories via (3.85) for staying away from the constraints without, however, affecting neither the final gripper pose (which is under the operator's control) nor the parts of gripper/camera trajectories which have already been traveled along. Finally, the operator receives the force cues (3.87) which inform her/him about where to place the master device for optimally moving away from the constraints.

While all these components effectively implement the sought forward/backward interface, we identified two shortcomings during our preliminary tests: (*i*) in their original design, the force cues (3.87) are, in general, always active even when the slave side is far enough from the constraints. This can result annoying for the user who must continuously resist (and interpret) the received cues also when not strictly necessary. In addition, (*ii*) because of the unavoidable saturations on any master device, nothing can in practice prevent the user to overcome the received (and practically saturated) force cues for then steering the slave side into any constraint (for example, by accidentally creating an occlusion along the trajectory that would prevent the reconstruction of the gripper/object poses).

A possible workaround to these two shortcomings is to define some *activation function* for (*i*) enabling the force cues only when close enough to a constraint and (*ii*) attenuating (and, in the limit, nullifying) the user's commands when too close to a constraint. To this end, we define the activation function

$$\alpha(H(\boldsymbol{\theta})) : \boldsymbol{\theta} \mapsto [0, 1] \tag{3.88}$$

which smoothly increases from 0 to 1 as the constraint cost function $H(\boldsymbol{\theta})$ grows unbounded. In particular, we chose this function

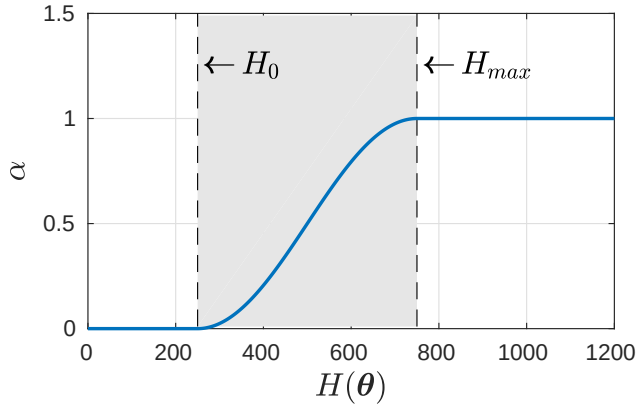


Figure 3.9: Activation function α used for regulating the haptic feedback and attenuating the user's commands as H gets close to growing unbounded.

$$\alpha = \begin{cases} 1 & if\ H > H_{max} \\ \frac{1}{2} + \frac{1}{2} sin\left(-\frac{\pi}{2} + \pi \frac{H - H_{th}}{H_{max} - H_{th}}\right) & if\ H_{th} < H < H_{max} \\ 0 & otherwise \end{cases} \tag{3.89}$$

H_{th} and H_{max} being the thresholds in which α is activated, whose graph is reported in Fig. 3.9.

Exploiting $\alpha(\boldsymbol{\theta})$, we can then modify (3.87) as

$$\boldsymbol{f} = \alpha(\boldsymbol{\theta}) \boldsymbol{K} (\boldsymbol{x}_m^* - \boldsymbol{x}_m), \tag{3.90}$$

for obtaining the sought result of (smoothly) activating the force cues only when close enough to a constraint. If the user continues to push towards a constraint while $\alpha > 0$, his commands are attenuated and nullified if H reaches the limit H_{max} ($\alpha = 1$). This is achieved by scaling the input $\boldsymbol{x}_m$ such that

$$x_{i,m} = \begin{cases} (1 - \alpha) x_{i,m} & if\ x_{i,m} x_{i,m}^* < 0 \\ x_{i,m} & otherwise \end{cases} \tag{3.91}$$

where $\boldsymbol{x}_m = [... \ x_{i,m} \ ...]$.

And as a final step we can adopt the same PSPM algorithm discussed in Sec. 3.2.5 for coping with the typical stability issues of any bilateral force feedback loop because of communication delays, packet losses, master/slave kinematic/dynamic dissimilarities, and other shortcomings.

3.4 Experimental Results

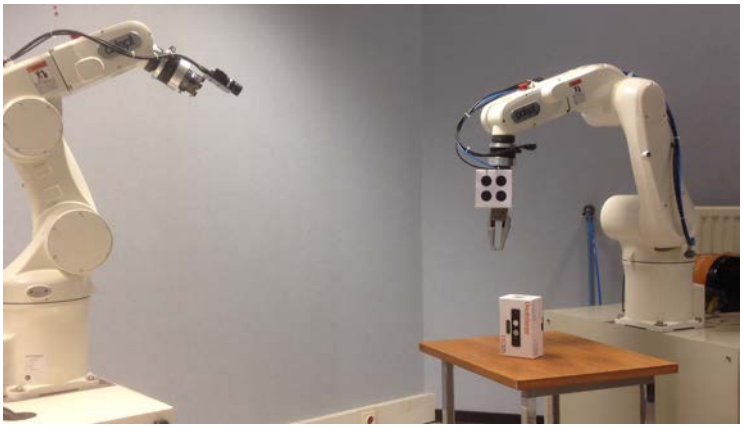
We now report the results of several experiments conducted to illustrate and validate the two proposed shared control architectures. The experiments are divided into three parts. The first part reports on experiments performed to validate the "instantaneous" architecture described in Sec. 3.2. The second discusses a user subject test which compares that same architecture to classical teleoperation for different control modes, and the last reports on experiments performed to illustrate and validate the trajectory-based shared control approach presented in Sec. 3.3.

3.4.1 Instantaneous Shared Control

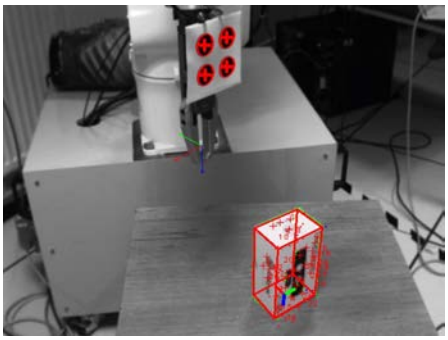
Figure 3.10 depicts the experimental setup used in these experiment. The master side consists of the Haption VIRTUOSE 6D haptic device[7], a high performance force feedback device with three translational DoF and three rotational DoF. The maximum force/torque is about 30 [N]/3 [Nm], the workspace has a spherical-like shape with an approximated radius of 0.9 [m], and the device exchanges data over ethernet with a control PC at 1 kHz. Four DoF of the Haption device were left unconstrained for actuating the $n = 4$ null-space directions $\boldsymbol{n}_i$ in (3.7), while the remaining two DoF were constrained via software to a constant value, see Fig. 3.10(b). A position-velocity forward map (see Sec. 3.2.3.2) which couples the position of the master along the four free DoF to the pseudo-velocity commands $\boldsymbol{\lambda}$ via (3.26) is used. The slave side consists of two 6-DoF Viper S850 robotic arms carrying a parallel gripper and a camera. Experiments were performed with real and simulated manipulators. Simulation experiments were done in the popular V-REP environment[8]. The poses of the gripper and of the target object in the camera frame were reconstructed by feeding the model-based ViSP tracker [173] with the segmented location of some fiducial markers acquired at 30 Hz, see Fig. 3.10(a).

Finally, in all the experiments reported in this section we set the desired $\boldsymbol{s}_d = [0\ 0\ 1]^T$ in (3.8) so as to force the object pointing direction $\boldsymbol{s}(t)$ to be always aligned with the gripper approaching direction $\boldsymbol{z}_g$. Furthermore, the manipulator equipped with the camera is controlled using the point-based IBVS scheme as in (3.10) and the cost function used to generate the force cues is as described in (3.34). We now report two sets of experiments conducted for validating the described shared control architecture. The interested reader is also encouraged to watch the video at https://youtu.be/_dBvk9K6E0Q for a better appreciation of the combined gripper/camera motion under the operator's commands.

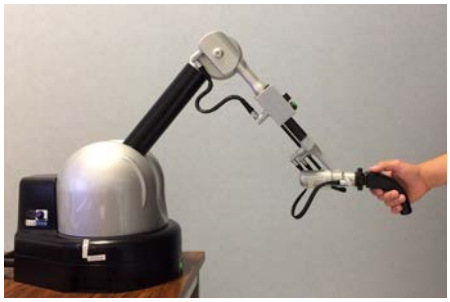
[7] www.haption.com.
[8] www.coppeliarobotics.com.



(a)



(b)



(c)

Figure 3.10: Experimental Setup: (a) the two 6-DoF manipulators carrying the camera and the gripper; (b) camera image with the segmentation of the object and the fiducial markers on the gripper exploited by the model-based tracker for reconstructing the gripper and object pose in the camera frame; (c) the VIRTUOSE 6D haptic device.

3.4.1.1 First Experiment

The first experiment is meant to illustrate the main features of our approach, i.e., the possibility of actuating the n null-space directions $\boldsymbol{n}_i$ while receiving a force feedback informing about the proximity in violating the robot joint limits. The experiment is split into three main phases: during the first phase (0 [s] $\leq t \leq 32$ [s]), the operator keeps the haptic device at its neutral position and gives no commands along the directions $\boldsymbol{n}_i$ ($\boldsymbol{\lambda} = \mathbf{0}$). During the second phase (32 [s] $\leq t \leq 105$ [s]), the operator starts actuating the null-space directions $\boldsymbol{n}_i$ one at the time with the aim of isolating the effects of each individual command. Then, during the last phase ($t \geq 105$ [s]), the operator provides a generic motion command that actuates all the null-space directions at once.

Figures 3.11(a–e) report the experimental results. In particular, Figs. 3.11(a–b) show the behavior of the target object direction $\boldsymbol{s}(t)$ and of the two point features

$\bar{\boldsymbol{p}}(t)$ during motion. One can then verify how the gripper/camera controllers (3.8)–(3.10) are able to regulate the values of $\boldsymbol{s}(t)$ and of $\bar{\boldsymbol{p}}(t)$ towards their desired (constant) values during the whole experiment despite the various null-space motions commanded by the operator (as expected). Presence of noise in the plots is mainly due to the 3D pose estimation by the ViSP model-based tracker which, as in any vision-based reconstruction, propagates the image noise in segmenting the fiducial markers on the gripper/object.

Figures 3.11(c–d) report the behavior of the pseudo-velocity commands $\boldsymbol{\lambda}(t)$ and of the force cues $\boldsymbol{f}(t)$, while Fig. 3.11(e) shows the behavior of the cost function $H(\boldsymbol{q}_g, \boldsymbol{q}_c)(t)$ over time. During the second phase of the experiment (20 [s] $\leq t \leq$ 105 [s]) the user commanded the n individual null-space motions until (intentionally) approaching the joint limits and, as a consequence, she/he received a corresponding force feedback cue. One can verify how the force cues $\boldsymbol{f}_i$ were activated only when approaching a joint limit with a corresponding increase of $H(\boldsymbol{q}_g, \boldsymbol{q}_c)$. It is also interesting to note that, although the user commanded an individual null-space direction $\boldsymbol{n}_i$ at the time, force cues along multiple axes were generated when approaching a joint limit. This is expected since, obviously, multiple directions in the joint space can potentially lead to a decrease of $H(\boldsymbol{q}_g, \boldsymbol{q}_c)$. Nevertheless, the direction of the main force cue (in terms of magnitude) was always well correlated with the direction actuated by the human operator, who thus felt the largest "resistant force" opposing her/his commands.

Presence of additional force cues along different axes than the one individually actuated can, anyway, have a beneficial role. Indeed, as discussed in Sect. 3.2.3, this kind of feedback can help the operator understanding which directions to (potentially) actuate in order to make the cost function $H(\boldsymbol{q}_g, \boldsymbol{q}_c)$ decrease. The operator can then keep on commanding a null-space motion along a direction of interest (e.g., for approaching the target object), while being automatically guided along the other null-space directions so as to stay away as much as possible from any joint limits. As explained, these considerations can be straightforwardly generalized to any additional criterium of interest (such as collision avoidance or torque limits) by a proper shaping of the cost function $H(\boldsymbol{q}_g, \boldsymbol{q}_c)$.

3.4.1.2 Second Experiment

The second experiment is meant to complement the previous one by showing the effectiveness of the force cues $\boldsymbol{f}$ in guiding the operator away from the considered robot constraints (joint limits). In this experiment, the operator first intentionally steers the robot towards a joint limit by actuating one of the four motion directions $\boldsymbol{n}_i$ and resisting to the received force feedback. As the force cues become significant,

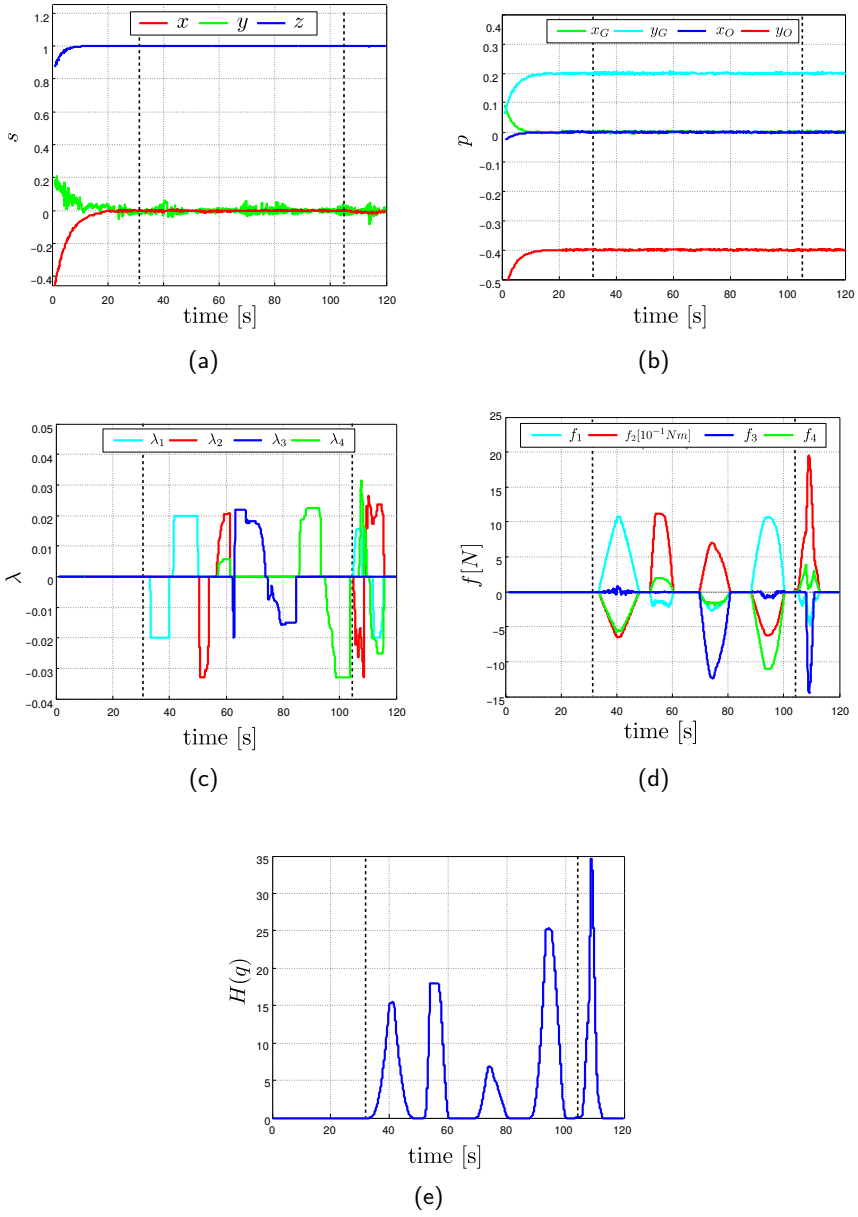


Figure 3.11: Results of the first experiment. (a) behavior of the primary task $\boldsymbol{s}(t)$; (b) behavior of the gripper and object positions $\bar{\boldsymbol{p}} = [\bar{\boldsymbol{p}}_g^T \ \bar{\boldsymbol{p}}_o^T]^T$ on the image plane; (c) Behavior of the four operator's commands $\boldsymbol{\lambda}$ for actuating the null-space directions $\boldsymbol{n}_i$. (d) Behavior of the four force cues f_i displayed to the human operator during the robot motion; (e) behavior of the scalar cost function $H(\boldsymbol{q}_g, \boldsymbol{q}_c)$ quantifying the proximity to joints limits.

the operator then stops applying her/his command and passively follows the master device guided by the forces f_i. This sequence is repeated twice over two different motion directions, and the obtained results are illustrated in Fig. 3.12(a–e).

Analogously to the previous case, Figs. 3.12(a–b) illustrate how the gripper/-camera controllers (3.8)–(3.10) effectively regulated the quantities $\boldsymbol{s}(t)$ and $\bar{\boldsymbol{p}}(t)$ at their desired values for the whole experiment duration (as, again, expected). Figures 3.12(c–e) show the operator's commands $\boldsymbol{\lambda}(t)$, the received force cues $\boldsymbol{f}(t)$ and the behavior of $H(\boldsymbol{q}_g(t), \boldsymbol{q}_c(t))$. During the first phase of the experiment (22 [s] $\leq t \leq$ 37 [s]), the operator intentionally steered the manipulator towards a joint limit by actuating λ_3. As shown in Fig. 3.12(e), this caused an increase of the cost function $H(\boldsymbol{q}_g(t), \boldsymbol{q}_c(t))$ and an associated activation of the force cues f_i along some axes of the master device. As the operator stopped commanding her/his motion ($t = 37$ [s]) and passively followed the received cues, the cost function $H(\boldsymbol{q}_g(t), \boldsymbol{q}_c(t))$ could quickly decrease thus moving away from the joint limit constraints. We also note the good correspondence between the activated force feedback signals (mainly f_1 and f_3) and the actuated motion directions (mainly λ_1 and λ_3) during this latter phase. A similar pattern can also be found during the second part of the experiment (70 [s] $\leq t \leq$ 75 [s]) in which the operator commanded a different motion direction (λ_4) for steering the robot towards the joint limits and then passively followed the received force cues. Again, the received cues were helpful in quickly guiding the operator towards a configuration far from any join limit.

In conclusion, the chosen force cues proved to be both informative and efficient in assisting the user in keeping the gripper/manipulator away from undesired configuration by either moving back along the operator's commanded direction, or by manoeuvring over the other available motion directions as a function of the magnitude (and sign) of the received haptic information.

3.4.2 Human-Subject Evaluation

In order to further demonstrate the feasibility and effectiveness of our method, we conducted a telemanipulation experiment in a real environment. We compared the proposed instantaneous shared-control approach (described in Sec. 3.2) with a more classic teleoperation architecture, in which the human operator is in charge of controlling all the DoF of the slave manipulator. Moreover, we also compared the two different approaches to control the motion of the robotic manipulator through the haptic interface:

- position-velocity: positions of the master handle are used to command the velocities of the slave robot (this is the modality described in Sec. 3.2.3.2);

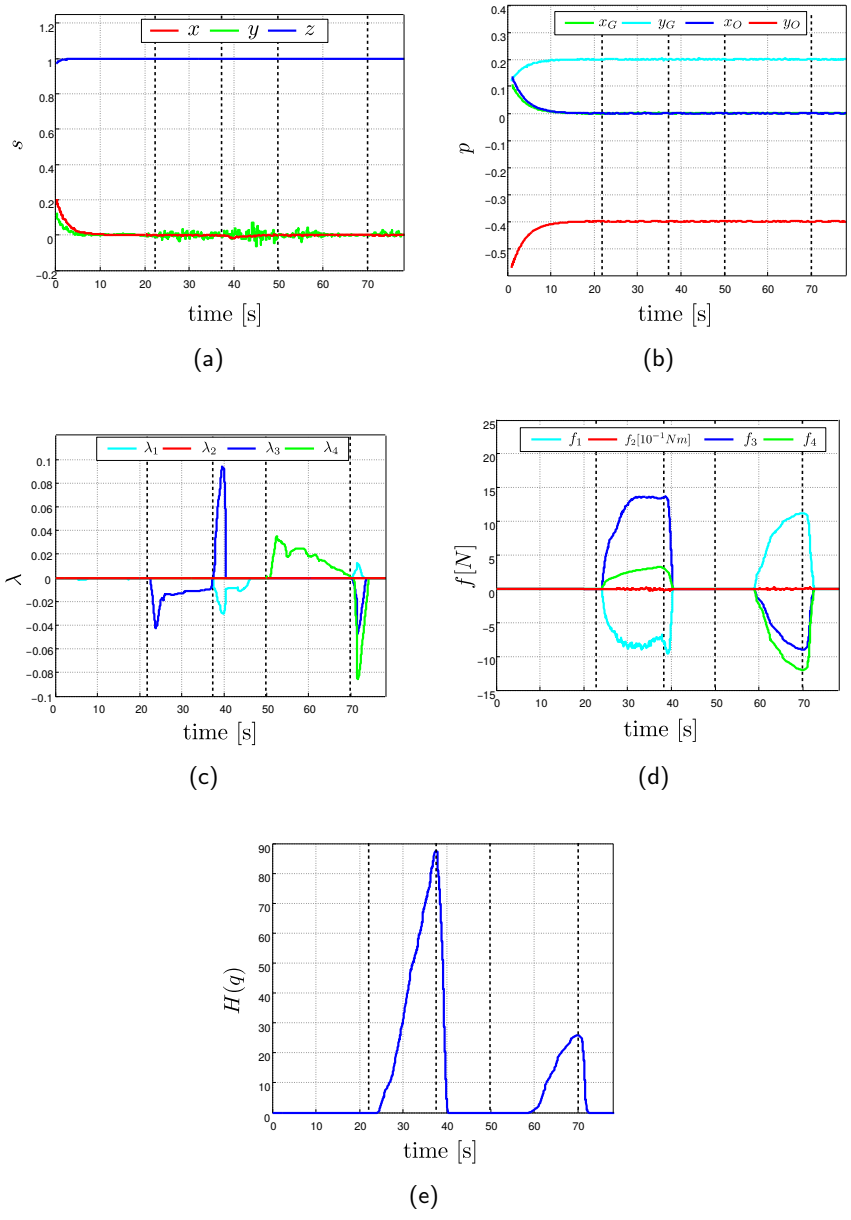
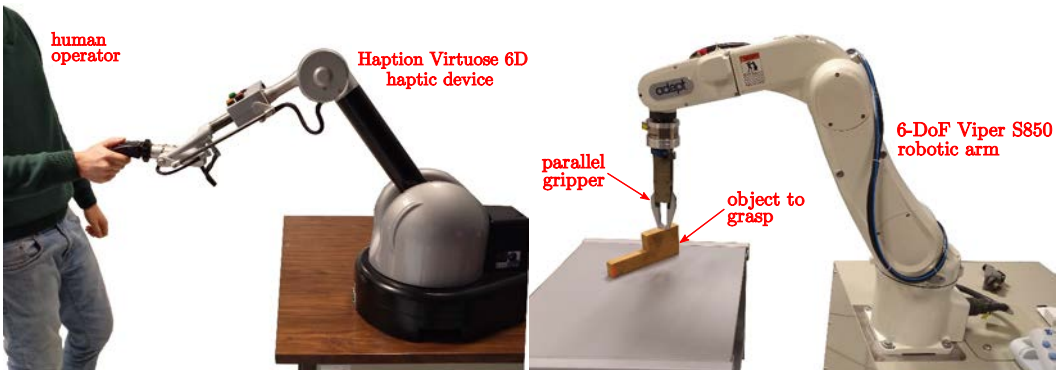


Figure 3.12: Results of the second experiment. (a) behavior of the primary task $\boldsymbol{s}(t)$; (b) behavior of the gripper and object positions $\boldsymbol{p} = [\boldsymbol{p}_g^T \ \boldsymbol{p}_o^T]^T$ on the image plane; (c) Behavior of the four operator's commands $\boldsymbol{\lambda}$ for actuating the null-space directions $\boldsymbol{n}_i$. (d) Behavior of the four force cues f_i displayed to the human operator during the robot motion; (e) behavior of the scalar cost function $H(\boldsymbol{q})$ quantifying the proximity to joints limits.

- velocity-velocity: velocities of the master handle are coupled to the the velocities of the slave robot, modulo a constant roto-translation scaling factor (this is the modality described in Sec. 3.2.3.1).

3.4.2.1 Experimental setup

Figure 3.13 shows the experimental setup. The master system is composed of the same Haption Virtuose 6D haptic device used in the previous experiments. The slave system is composed of a 6-DoF Viper S850 robotic arm carrying a pneumatic parallel gripper. A wooden object with dimensions $21 \times 9 \times 3$ cm and weight 280 g is placed on a table in front of the robotic manipulator. The user had a direct view of the slave system and the object to be grasped. The setup is simplified by removing the manipulator with the camera and the object is assumed to be static and its pose known beforehand. The pose of the gripper is retrieved online from the kinematics of the robot.



(a) Master side. (b) Slave side.

Figure 3.13: Experimental setup. The master system is composed of the Haption Virtuose 6D haptic device, while the the slave system is composed of a 6-DoF Viper S850 robotic arm carrying a pneumatic parallel gripper. A wooden object with dimensions $21 \times 9 \times 3$ cm and weight 300 g is placed on a table in front of the robotic manipulator.

3.4.2.2 Experimental conditions and task

Participants were required to control the motion of the robotic manipulator and gripper to grasp the wooden piece and lift it from the ground. The task started when the manipulator moved for the very first time and it was considered successfully completed when the object was lifted from the ground.

We considered two different ways of commanding the motion of the robot through the haptic interface (position-velocity vs. velocity-velocity) and two different levels

of human involvement in the control (shared control vs. teleoperation), ending up with four different experimental conditions:

S+PV: shared-control with position-velocity motion control, where the subject controls only 4 DoF of the manipulator, and positions of the haptic device are mapped into velocities of the manipulator (see Sec. 3.2.3.2);

S+VV: shared-control with velocity-velocity motion control, where the subject controls only 4 DoF of the manipulator, and velocities of the haptic device are mapped into velocities of the manipulator (see Sec. 3.2.3.1)

T+PV: teleoperation with position-velocity motion control, where the subject controls all the 6 DoF of the manipulator, and positions of the haptic device are mapped into velocities of the manipulator;

T+VV: teleoperation with velocity-velocity motion control, where the subject controls all the 6 DoF of the manipulator, and velocities of the haptic device are mapped into velocities of the manipulator.

The shared-control architecture, employed in conditions S+PV and S+VV, is the one detailed in Sec. 3.2.1.

In T+PV and T+VV conditions, the subject is in *full* control of the manipulator's 6 DoF. In this respect, the master/slave coupling in T+PV is

$$\boldsymbol{v}_g = \boldsymbol{K}_{PV}\, \boldsymbol{x}_m, \tag{3.92}$$

where the configuration vector of the master device, $\boldsymbol{x}_m$, is now in $\mathbb{R}^6$ and $\boldsymbol{K}_{PV} \in \mathbb{R}^{6\times 6}$ is a matrix mapping $\boldsymbol{x}_m$ to velocity commands on the slave side. Similarly to Sec. 3.2.3, the force cues fed to the user are

$$\boldsymbol{\tau} = -\boldsymbol{B}_m \dot{\boldsymbol{x}}_m - \boldsymbol{K}_m \boldsymbol{x}_m + \boldsymbol{f}, \tag{3.93}$$

where $\boldsymbol{B}_m \in \mathbb{R}^{6\times 6}$ and $\boldsymbol{K}_m \in \mathbb{R}^{6\times 6}$ are the damping and stiffness matrices of a spring pushing the master handle back to the "zero velocity" position, and, from (3.30), $\boldsymbol{f}$ is defined as

$$\boldsymbol{f} = -k_f \left(\boldsymbol{x}_m^* - \boldsymbol{x}_m\right), \tag{3.94}$$

since now no primary task is present and, therefore, the null-space basis $\boldsymbol{N}_B$ is just the identity matrix.

On the other hand, in the T+VV condition the coupling is

$$\boldsymbol{v}_g = \boldsymbol{K}_{VV}\, \boldsymbol{v}_m, \tag{3.95}$$

where $\boldsymbol{K}_{VV} \in \mathbb{R}^{6\times6}$ is a diagonal scaling matrix. Haptic feedback is then designed similarly to Sec. 3.2.3.1, with

$$\boldsymbol{\tau} = -\boldsymbol{B}_m \boldsymbol{v}_m + \boldsymbol{f}, \tag{3.96}$$

where $\boldsymbol{f}$ follows (3.24).

3.4.2.3 Participants

Ten right-handed subjects (average age 27.2) participated in the study. Three of them had previous experience with haptic interfaces. None of the participants reported any deficiencies in their visual or haptic perception abilities. The experimenter explained the procedures and spent about two minutes adjusting the setup to be comfortable before the subject began the experiment. Each subject then spent about three minutes practicing the control of the telemanipulation system before starting the experiment. Each subject carried out 8 randomized repetitions of the grasping task, 2 for each experimental condition. A video showing trials in all experimental conditions is available as supplemental material.

3.4.2.4 Results

To evaluate the effectiveness of our system in grasping the considered object, the usefulness of the proposed shared-control approach, and the effectiveness of haptic stimuli to render robotic setup constraints, we recorded (*i*) the completion time, (*ii*) the linear trajectory followed by the robotic end-effector, (*iii*) the angular motion of the robotic end-effector, and (*iv*) the perceived effectiveness of the different conditions. To compare the different metrics, we ran both two-way and one-way repeated-measures ANOVA tests on the data. In the two-way analysis, motion control (position-velocity vs. velocity-velocity) and human involvement in the control (shared control vs. teleoperation) were treated as within-subject factors. All data passed the Shapiro-Wilk normality test. Sphericity was assumed for variables with only two levels of repeated measures. The two-way analysis enables us to understand the role of each variable considered within-subject factor, while the one-way analysis provides us with an overview on the performances of the four conditions.

Figure 3.14a shows the average task completion time. The two-way ANOVA test revealed a statistically significant change in the task completion time for the human involvement in the control variable (shared control vs. teleoperation, $F(1, 9) = 25.852$, $p = 0.001$). The interaction effect between these two factors was not statistically significant. The one-way ANOVA test revealed a statistically significant change in the task completion time across the conditions ($F(3, 27) = 9.312$, $p < 0.001$). Post hoc analysis with Bonferroni adjustments revealed a statistically

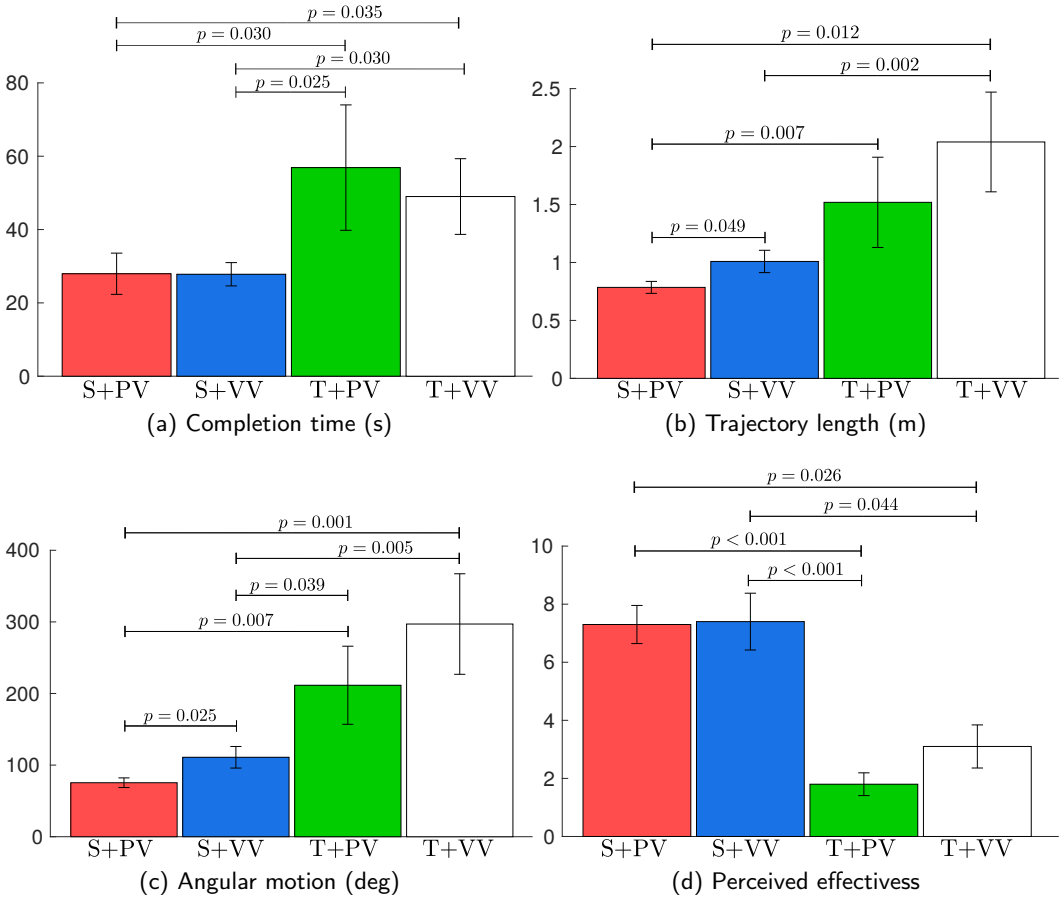


Figure 3.14: Experimental evaluation. Mean and 95% confidence interval of (a) completion time, (b) trajectory length, (c) angular motion, and (d) perceived effectiveness of the four feedback conditions are plotted.

significant difference between S+VV vs. T+VV ($p = 0.030$), S+VV vs. T+PV ($p = 0.035$), S+PV vs. T+VV ($p = 0.031$), and S+PV vs. T+PV ($p = 0.025$). The Bonferroni correction is used to reduce the chances of obtaining false-positive results when multiple pair-wise tests are performed on a single set of data.

Figure 3.14b shows the average linear motion covered by the robotic gripper during the task. The two-way ANOVA test revealed a statistically significant change in the trajectory length for both the human involvement in the control ($F(1, 9) = 30.968$, $p < 0.001$) and the motion control type (velocity vs. position, $F(1, 9) = 9.035$, $p = 0.015$) variables. The interaction effect between these two factors was not statistically significant. The one-way ANOVA test revealed a statistically significant change in the trajectory length across the conditions ($F(1.929, 17.360) = 14.072$, $p < 0.001$). Post hoc analysis with Bonferroni adjustments revealed a statistically significant difference between S+VV vs. S+PV ($p = 0.049$), S+VV vs. T+VV ($p = 0.043$), S+VV vs. T+PV ($p = 0.002$), and S+PV vs. T+PV ($p = 0.012$).

Figure 3.14c shows the average angular motion covered by the robotic gripper during the task. The two-way ANOVA test revealed a statistically significant change in the angular motion for both the human involvement in the control ($F(1, 9) = 39.350$, $p < 0.001$) and the motion control type (position-velocity vs. velocity-velocity, $F(1, 9) = 8.202$, $p = 0.015$) variables. The interaction effect between these two factors was not statistically significant. The one-way ANOVA test revealed a statistically significant change in the trajectory length across the conditions ($F(3, 27) = 12.994$, $p < 0.001$). Post hoc analysis with Bonferroni adjustments revealed a statistically significant difference between S+VV vs. S+PV ($p = 0.025$), S+VV vs. T+VV ($p = 0.007$), S+VV vs. T+PV ($p = 0.001$), S+PV vs. T+VV ($p = 0.039$), and S+PV vs. T+PV ($p = 0.005$).

Immediately after the experiment, subjects were also asked to report the effectiveness of each feedback condition in completing the given task using bipolar Likert-type nine-point scales. Fig. 3.14d shows the perceived effectiveness for the four experimental conditions. A Friedman test showed a statistically significant difference between the means of the four feedback conditions ($\chi^2(3) = 26.753$, $p < 0.001$, $a = 0.05$). The Friedman test is the non-parametric equivalent of the more popular repeated-measures ANOVA. The latter is not appropriate here since the dependent variable was measured at the ordinal level. Post hoc analysis with Bonferroni adjustments revealed a statistically significant difference between S+VV vs. T+VV ($p < 0.001$), S+VV vs. T+PV ($p = 0.026$), S+PV vs. T+VV ($p < 0.001$), and S+PV vs. T+PV ($p = 0.044$).

Finally, all ten subjects found conditions using the shared-control approach to be the most effective at completing the grasping task. Seven subjects out of ten chose the shared-control condition employing velocity-velocity control to be the most effective.

3.4.2.5 Discussion

Results showed that, generally, and in all the considered metrics, the shared-control approach significantly outperformed the more classic teleoperation architecture. Moreover, all the subjects preferred the shared-control architecture with respect to teleoperation. This proves our hypothesis that shared-control can be a viable and very effective approach to improve currently-available teleoperation systems in remote manipulation tasks. However, it is important to notice that our subjects were not expert in using the experimental setup. In this respect, it may happen that the recorded significant difference in performance between shared control vs. teleoperation might become less significant in the presence of experienced users.

In addition to this first result, the 1-way statistical analysis gave us insights

about the differences between the composite conditions. Results show a significant difference between S+PV vs. S+VV in the trajectory length and angular motion metrics, with the former condition outperforming the latter. This result came as a surprise, since it is in contrast with the results of the user experience evaluation. In fact, both the perceived effectiveness and the choice of the preferred condition clearly show that users preferred conditions employing velocity-velocity control with respect to position-velocity control. In this respect, all subjects complained that position-velocity conditions required more attention and, in general, a higher cognitive load. Three subjects, who indicated S+VV as their preferred condition, asserted that they would have probably preferred condition S+PV if the task would have required more time to complete, since S+PV does not require clutching (see Sec. 3.2.3).

Clutching is indeed another interesting point to discuss. Even in the velocity-velocity control approach, given a grounded haptic interface and a grounded slave manipulator, it is always possible to define an appropriate scaling factor between master and slave velocities such that the operator does not require clutching. However, as the difference between the master and slave workspaces increases, this mapping requires higher and higher gains, resulting in a telemanipulation system which is very hard to control, since the operator's accuracy/resolution in positioning the slave arm is degraded. The RoMaNS project presents us with the perfect example: the custom rig at the National Nuclear Laboratory is composed of (i) the same grounded haptic interface we are using in this paper, and (ii) a 500-kg-payload Kuka KR 500 manipulator. Although it is theoretically possible to map the workspace of the Virtuose 6D to the (much) larger workspace of the KUKA robot, this would result in very high motion gains (i.e., a small movement of the master interface would cause a big movement of the slave robot). For this reason, we decided to implement the velocity-velocity modality using the clutch. In this respect, we are also interested in understanding how to best tune the master-slave motion scaling factor, with the final objective of finding a good trade-off between high precision of movement and low need of clutching.

Finally, all subjects appreciated the presence of haptic feedback to provide information about the manipulator's joint limits. This approach enabled them to always complete the task successfully, pushing them away from dangerous robot configurations in a very intuitive and non-obtrusive way. Subjects described the feeling due to the haptic feedback "as if the system was trying to nudge them towards a safer configuration of the robot."

3.4.3 Shared Trajectory Planning

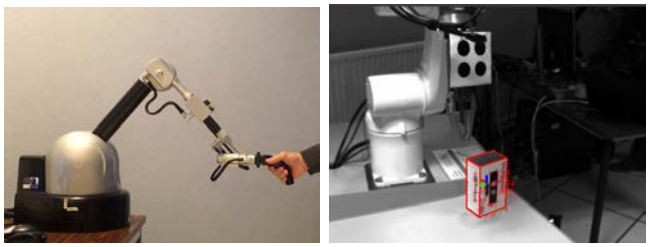
The last set of experiments in this chapter aim at testing and validating the trajectory-based shared-planning strategy described in Sec. 3.3. We refer the reader to the video available on https://youtu.be/p9X8ZKJ77m4 for a better understanding of the performed experiments. We now proceed to detail the experimental setup used and the three performed experiments.

3.4.3.1 Experimental Setup

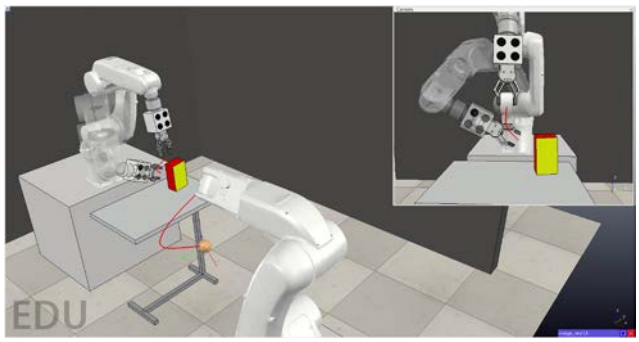
Fig. 3.15 shows the setup used to test the described architecture. The master side is always the Haption Virtuous 6D where, as in the previous experiments, the user was given command over 4 degrees of freedom while the remaining 2 were blocked via software. The degrees of freedom commanded by the operator are shown in Fig. 3.3.

The slave side is consisted of two 6-DoF Adept Viper serial manipulators. An S850 equipped with a pneumatic gripper and an S650 equipped with a Pointgrey Flea 2 monocular camera. The 30 Hz feed from the camera is used to calculate the pose of the object in real time. To this end, an efficient mixed tracker from ViSP library [173] was used. The tracker uses the model of the object, which is assumed to be known, in addition to KLM-features to efficiently calculate the pose of the object in real time. The kinematic data from both manipulators, in addition to the data retrieved from the tracker, are used to create a virtual reconstruction of the scene using the virtual robot experimentation platform V-REP (see Fig. 3.15c).

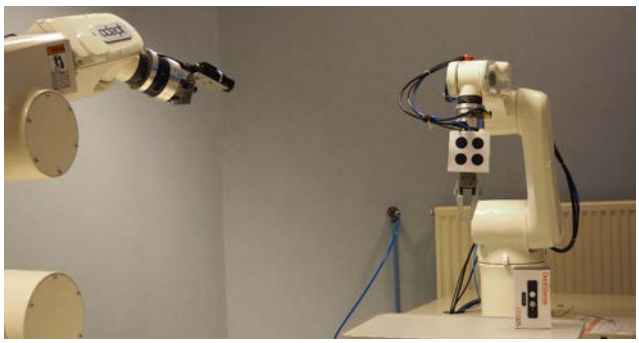
The 'phantom' robot in Fig. 3.15c represents the final pose of the gripper's trajectory. As the user acts on the master device adapting the final pose, the 'phantom' robot follows accordingly while maintaining its orientation towards the target object (see Sec. 3.3.4.1). The trajectories of both, the gripper and the camera, are then optimized following the user commands under the action of the autonomous component $\boldsymbol{u}_{\boldsymbol{\theta},A}$ (eq. (3.85)). If the user happens to drive the trajectories toward a constraint of the system, the color of the plotted trajectory changes from green (far from constraints) when $\alpha = 0$ to red (close to constraints) as $\alpha \mapsto 1$. To recall, $\alpha(\boldsymbol{\theta})$ is a saturation function that goes from zero to 1 as $H(\boldsymbol{\theta})$ increases in the proximity of constraints. This provides the user with a supportive visual feedback which goes hand in hand with the haptic cues she/he receives on the master device. When the user converges to a desired grasping pose and moves the robot along the planned trajectory, the motion is simultaneously executed in the real and simulated environments.



(a) (b)



(c)



(d)

Figure 3.15: The figure shows the experimental setup used to validate the approach. (a) shows the master side whereas (b) shows the tracker used to retrieve the pose of the object to be grasped in real time. (c) shows the virtual environment reconstructed using the tracker's and kinematic data and used as an interface with the operator while (d) is the two slave manipulators arms along with the object to be grasped.

3.4.3.2 First Experiment

In order to test the impact of the autonomous component on the trajectories, the algorithm was first launched without any input from the human operator. Fig. 3.16 shows the variation of each of the cost functions with time. The total cost function $H(\boldsymbol{\theta})$, plotted in Fig. 3.16a, decreases continuously, as expected, and converges towards a minimum. The behavior of the individual cost functions is plotted in Fig. 3.16b, Fig. 3.16c and Fig. 3.16d representing the visual constraints, the kinematic constraints of the gripper manipulator and those of the camera manipulator respectively. While all individual cost functions are decreasing in the beginning of the experiment, the cost corresponding to the visual constraints (H_V) and that corresponding to the gripper manipulator ($H_{J,g} + H_{S,g}$) increase slightly towards its end. On the other hand, the cost corresponding to the kinematic constraints of the camera manipulator ($H_{J,c} + H_{S,c}$) keeps decreasing. This is an expected behavior since the cost functions are summed together in $H(\boldsymbol{\theta})$ without enforcing the individual monotonic decrease of each. However, since the cost functions grow unbounded as the user approaches a constraint, it is guaranteed that all constraints are respected.

3.4.3.3 Second Experiment

The second experiment is meant to show the behavior of the system as the user actuates the different motion directions $\boldsymbol{n}_i$ (eq. 3.7) and to test the efficiency of the haptic interface in guiding the user out of undesired configurations. To this end, we let the operator actuate each motion direction, one at a time, until intentionally hitting a constraint. The operator is consequently provided with force cues informing her/him of the proximity to that constraint and the directions which would drive her/him toward a safer configuration. The user then follows the provided cues retracting away from the constraints toward a more convenient position. The is repeated for all the 4 DoF which the user is commanding.

Fig. 3.17 reports the results obtained from this experiment. In particular, Fig. 3.17a shows the user commands fed through the master device, Fig. 3.17b reports the force cues received over each of the commanded motion directions. Fig. 3.17c plots the total cost function $H(\boldsymbol{\theta})$ and Fig. 3.17d depicts the saturation function $\alpha(\boldsymbol{\theta})$.

It is noted from the graphs that every 'significant' increase in $H(\boldsymbol{\theta})$ (implying the proximity of the system to a constraint) is clearly reflected in the profile of the saturation function, $\alpha(H(\boldsymbol{\theta}))$, and in the generated force cues. However, there is no clear correlation between the force cues received by the operator and the direction she/he is commanding. For example, for ($0\ sec\ <\ t\ <\ 5\ sec$), the operator was

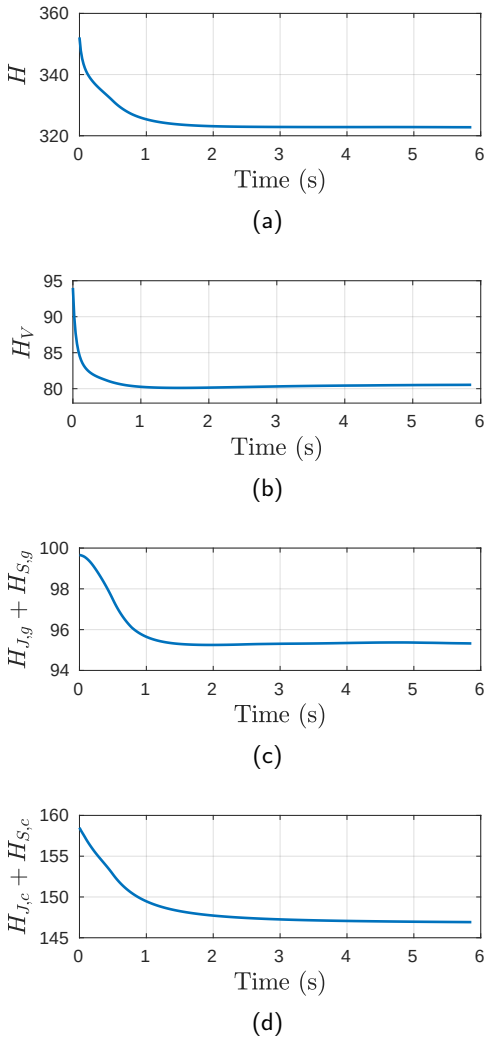


Figure 3.16: The decrease of the cost functions as the system is initially activated and the initial trajectory is optimized.

commanding the system along $\boldsymbol{n}_1$ but received force cues along $\boldsymbol{f}_1$, $\boldsymbol{f}_2$ and $\boldsymbol{f}_3$. In fact, the direction of maximal decrease (gradient decent) of the cost function is not necessarily the opposite of the direction which the user is commanding. For ex., if the user approaches a particular constraint by moving the system along a direction, say $\boldsymbol{n}_1$, the gradient of the total cost function $H(\boldsymbol{\theta})$ may have components on $\boldsymbol{n}_2$ and $\boldsymbol{n}_3$ in addition to $\boldsymbol{n}_1$ (and the user gets force cues on all three directions). Moving along the provided cues on any of these directions would guarantee the decrease of $H(\boldsymbol{\theta})$ and drive the system away from the constraint. This point is particularly important in the design of the haptic cues. Instead of providing the user with resistive cues which just block her/his commands, the system guides her/him towards the direction which best drives her/him out of the current (undesired) configuration. Back to the experiment, and as the user received forces over $\boldsymbol{f}_1$,

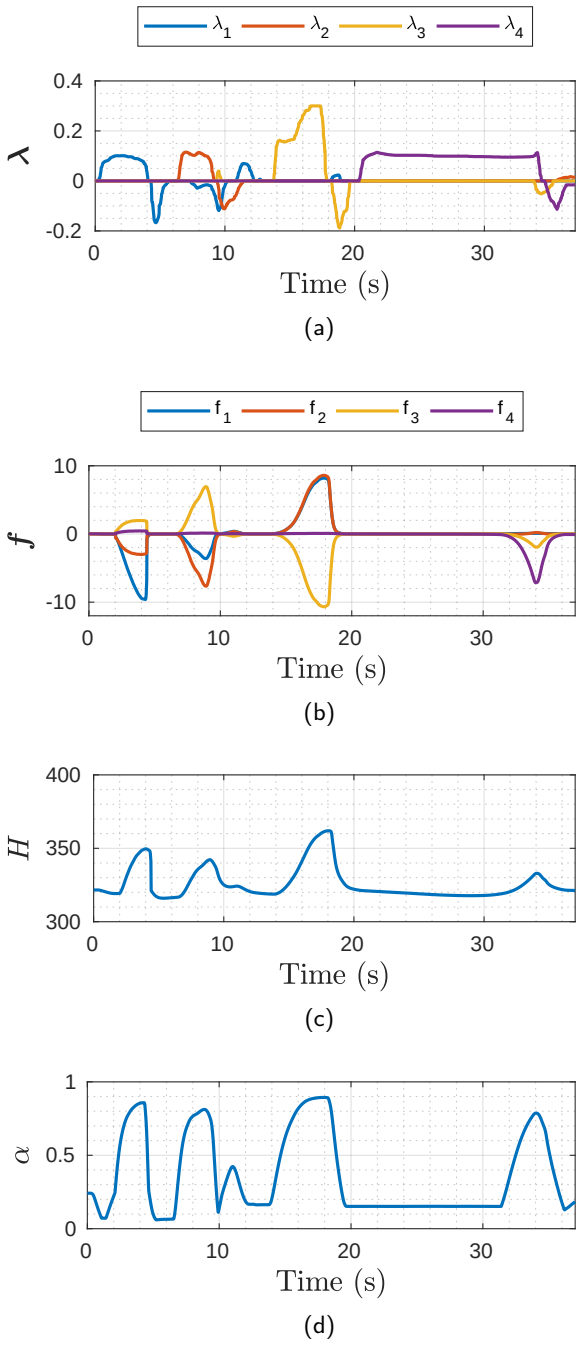


Figure 3.17: The first experiment showing the activation of each of the motion directions controlled by the user. The user activates each until a force feedback is received due to proximity to a constraint.

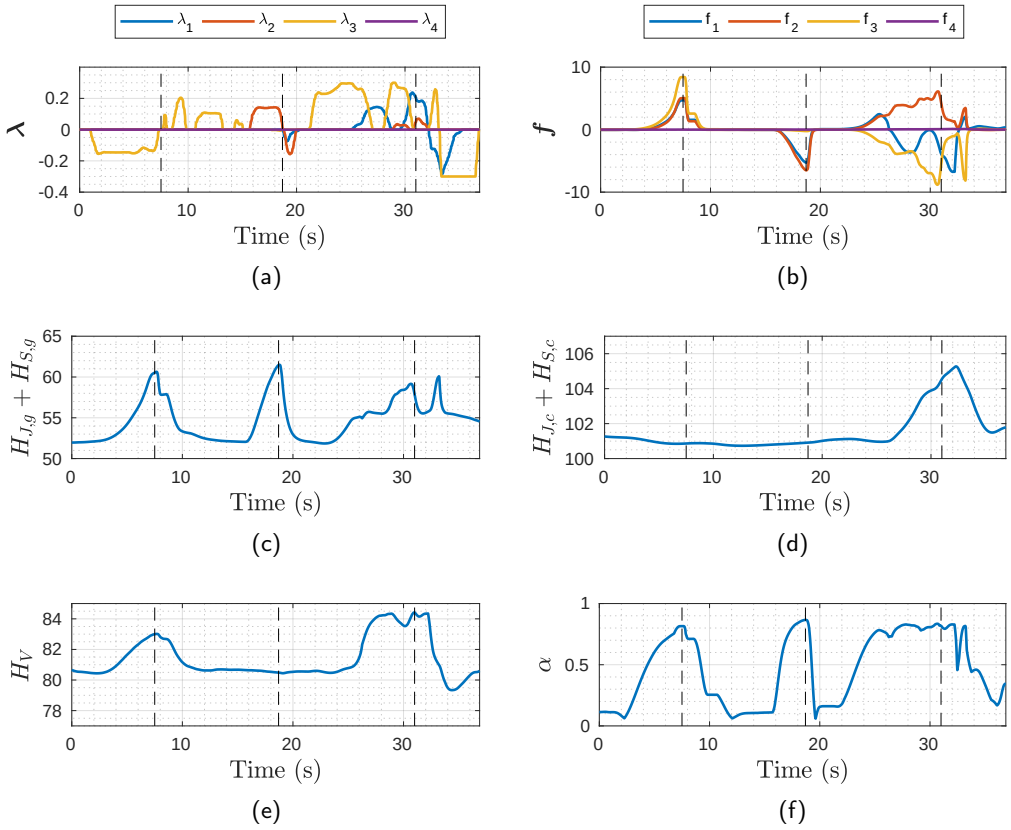


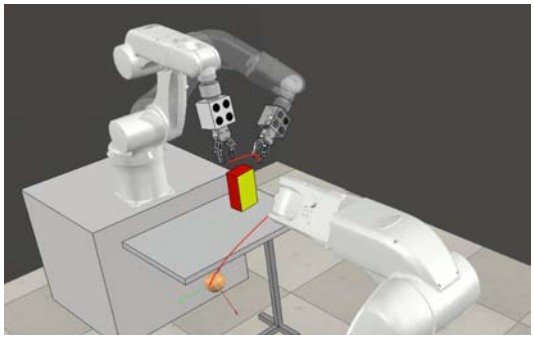
Figure 3.18: The user activates the different decreases of freedom until a constraint is reached. The constraints reached at the dotted lines are highlighted in the below images.

$\boldsymbol{f}_2$ and $\boldsymbol{f}_3$, it is her/his decision to move along this or that direction to avoid the constraints depending on the task at hand. This is also reflected in the magnitude of the cues which is limited to $10N$ (Fig. 3.17b) and can be overridden by the operator.

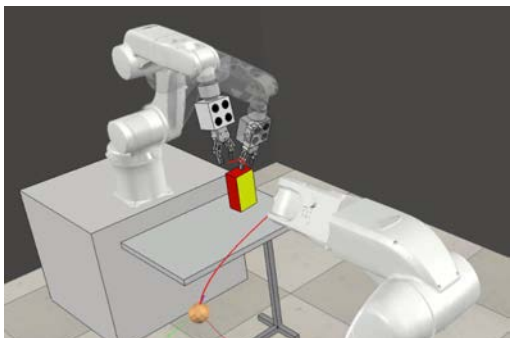
3.4.3.4 Third Experiment

Fig. 3.18 and Fig. 3.19 report the results of the third experiment which aims at highlighting the different individual constraints of the system. Unlike the second experiment, the concentration here is not on the different motion directions but on highlighting the constraints themselves and visualizing them. The user drives the system towards the different constraints using any convenient input combination.

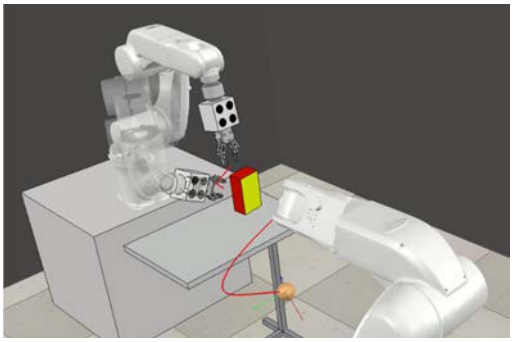
Fig. 3.18a shows the operator's velocity commands, Fig. 3.18b shows the received force cues, Figs. 3.18c–3.18e plot the integral cost function associated with the kinematic constraints of the gripper manipulator $H_{J,g} + H_{S,g}$, the kinematic constraints of the camera manipulator,$H_{J,c} + H_{S,c}$, and the visual constraints, H_V,



(a)



(b)



(c)

Figure 3.19: The constraint approached during experiment two (Fig. 3.18) at t=8, t=19, and t=31 sec.

respectively, and Fig. 3.18f encodes the evolution of the saturation function α.

The three dotted lines at $t = 8s$, $t = 19s$ and $t = 31s$ denote significant instants of the experiment. At $t = 8s$, the visual constraints and the kinematic constraints of the gripper's manipulator were simultaneously approached. This proximity is reflected in $H_{J,g} + H_{S,g}$ and H_V. It is also visible in Fig. 3.19a showing the state of the manipulators where the arm equipped with the gripper is at its limits. Moreover, the gripper is quite close to the object. Recalling that one of the enforced visibility constraints is preventing the gripper and the object from obscuring one another by keeping $d_{o,g} > 0$, this explains why visibility constraints are also active here.

At $t = 19s$, it is only the gripper's kinematic constraints which are approached while the system is far enough from the visual constraints and the kinematic constraints of the camera. As one can observe in Fig. 3.19b, the 5^{th} joint of the manipulator equipped with the gripper is close to its limit. On the other hand, the trajectory of the camera bended downwards accounting for the motion of the gripper and preventing it from obscuring the object to keep the system away from visibility constraints. In contrast to the previous two instances, all the constraints of the system were simultaneously approached at $t = 31s$. Fig. 3.19c shows the

trajectory of the camera being stretched to provide a better visibility of the object. The gripper is also close to obscuring the object that the visibility constraints are activated as $d_{o,g} \to 0$.

3.5 Summary

In this chapter we presented a shared control framework for allowing a human operator being in partial control of the pre-grasp approaching phase towards an object of interest via two serial manipulator arms, one carrying a gripper and another one a camera looking at the scene (gripper and object). A visual-based control law was implemented for the autonomous part of the architecture (control of a subset of the gripper/object DoF), and the operator was given the possibility of directly commanding the remaining null-space directions w.r.t. the main visual servoing task. Moreover, an informative high-level haptic feedback was proposed with the aim of informing the operator about the feasibility of her/his motion commands against possible constraints of the considered robotic system (in particular, joint limits and singularities of both manipulator arms and visibility constraints). Finally, the camera motion was optimized so as to keep a good vantage point w.r.t. the scene (object and gripper).

In order to demonstrate the feasibility and effectiveness of our method, we conducted a remote telemanipulation experiment where ten human subjects were asked to control the motion of the 6-DoF telemanipulator to grasp a target object. We tested the performance of the proposed shared-control system against a more classic teleoperation approach, in which the human operator was able to freely control all the degrees of freedom of the robotic manipulator. Moreover, we considered two ways of controlling the motion of the robot through the haptic interface - velocity-velocity and position-velocity, - ending up with four experimental conditions. As a measure of performance, we considered the average completion time, trajectory length, angular motion, and perceived effectiveness. We carried out a thorough statistical analysis to be able to separately compare the different ways of commanding the motion of the robot through the haptic interface (velocity-velocity vs. position-velocity) and the different levels of human involvement in the control (shared-control vs. teleoperation). Results showed that, in all the considered metrics, the shared-control approach significantly outperformed the more classic teleoperation architecture. Moreover, all the subjects preferred the shared-control architecture with respect to teleoperation.

The approach was then extended to a higher level planning-based shared control architecture. In this case, the operator is given the possibility of modifying the linear and angular trajectories of two manipulator arms while assisted by an autonomous

component to optimize these trajectories against system constraints. She/He acts on a virtual simulation of the real environment which visualizes the trajectories and gives an insight into the expected future behaviour of the manipulators. The user is also provided with haptic and visual feedback to keep her/him aware of the feasibility of her/his actions and inform her/him about the proximity of the system to the different constraints (at any point of the future trajectories). The effectiveness of the proposed architecture with integral haptic cues was then demonstrated in a set of experiments on real hardware.

Chapter 4

Extensions

Contents

4.1 Assisted Teleoperation in a Shared Workspace **98**
4.1.1 System Architecture 99
4.1.2 Passivity Analysis 105
4.1.3 Experiments in Simulation 106
4.1.4 Experiment in a Real Environment 111
4.1.5 Discussion 111
4.2 Grasping Assistance in a Cluttered Environment **113**
4.2.1 Overview on the Architecture 113
4.2.2 Point Cloud and Grasp Pose Generation 115
4.2.3 Haptic shared-control architecture 116
4.2.4 Experimental evaluation 123
4.2.5 Discussion 130
4.3 Learning-Based Haptic Design in Shared Control Systems **133**
4.3.1 Related Work 134
4.3.2 Modeling 135
4.3.3 Shared Control Architecture 137
4.3.4 Trajectory Refinement through Interactive Task Executions 138
4.3.5 Experimental Results 140
4.4 Summary **144**

The previous chapter presented a shared-control approach for controlling a serial manipulator in the pre-grasping phase. Two modes of actuation were considered: 1) instantaneous, where the user directly commands some DoF of the robot, and 2) trajectory-based, where the user acts on a master device to modify a future trajectory. Moreover, instead of providing a high-fidelity haptic

feedback, reflecting the actual physical contacts between the slave manipulator and the environment, the human operator was provided with haptic cues informing about the proximity of the system to different constraints with the idea that this design could better help the operator in successfully performing the approaching task (as also confirmed by the reported users' studies). In particular, kinematic and visibility constraints were considered. This chapter expands these previous ideas in three directions:

- Considering the case of teleoperating a serial manipulator in a shared workspace where an autonomous manipulator with independent, and possibly conflicting, goals is present and accounting for the additional constraints arising from such a scenario (Sec. 4.1).
- Generalizing the shared-control approach for grasping multiple objects in a cluttered and unknown environment (Sec. 4.2).
- Using data gathered from demonstrations by expert operators to tune the operator/autonomy balance depending on the confidence of the system in its guess (Sec. 4.3).

Some of the results contained in this chapter are based on work done in collaboration with the University of Naples Federico II and published in [27] and on work done in collaboration with the University of Darmstad and published in [23, 28]. Related descriptive media is available at:

- Assisted Teleoperation in a Shared Workspace: https://youtu.be/uWzVEGW2i5c.
- Grasping Assistance in a Cluttered Environment: https://youtu.be/Bb4M3UjwAGY.
- Learning-Based Haptic Design in Shared Control Systems: https://youtu.be/hvzxmwqAH5s.

4.1 Assisted Teleoperation in a Shared Workspace

When the slave side is a complex robotic system (e.g., dual-arm, highly redundant) or environmental obstacles are present, (self-)collision avoidance methods need to be considered. In the past, several collision avoidance algorithms have been developed to implement reactive control strategies or to plan collision-free paths for redundant robots [191, 192, 193, 194]. Self-collisions and joint limits have been also used as a criteria to find optimal inverse kinematic solutions for robotic manipulators [195, 196] and for haptic-guided teleoperation as well [197].

In this section, we consider the case of a dual-arm robotic system: one robot is (partially) teleoperated by a human, while the other autonomously executes a task in the same workspace. The two manipulators have independent and sometimes conflicting tasks which raises the risk of dangerous collisions. Towards a suitable haptic shared-control interface to assist the operator, we adapt and improve the approaches presented in the previous chapter and in [197], combining their features, extending their capabilities, and providing a novel, extensive human subject evaluation and results analysis. Specifically,

- we improve the shared-control approach in Sec. 3.2 by (i) dividing the approach-to-grasp phase into multiple parts, according to the gripper distance from the target object, and (ii) devising different, optimized strategies for each of these parts;
- we improve the self-collision-aware approach presented in [197] by (i) also considering collisions with the surrounding environment and (ii) replacing the convex meshes collision model of the robot with discretized sphere volumes;
- we combine the proposed, new above mentioned techniques to create a novel shared-control framework able to manage a dual-arm system, where one robot is (partially) controlled by the operator while the other one autonomously performs a task in the same environment;
- we present a port-Hamiltonian model of the overall system subject to various control modalities, proving the passivity of the system w.r.t. the operator actions;
- we validate the proposed architecture with an extensive human subject evaluation in simulated and real environments, enrolling a total of 20 participants.

4.1.1 System Architecture

The considered telemanipulation system consists of two serial manipulators sharing a workspace in an industrial setting. One of the manipulators is performing an independent fully-autonomous task (in this case, a visual inspection with an onboard camera), while the second manipulator is (partially) teleoperated by a human operator whose aim is to grasp a target object (see Fig. 4.1). The operator faces a number of constraints while controlling such a system: kinematic/workspace constraints for the manipulator she/he is directly controlling, as well as possible interferences with the task performed by the fully-autonomous arm. As explained, the purpose of the architecture is to facilitate the operator's task and decrease her/his mental workload.

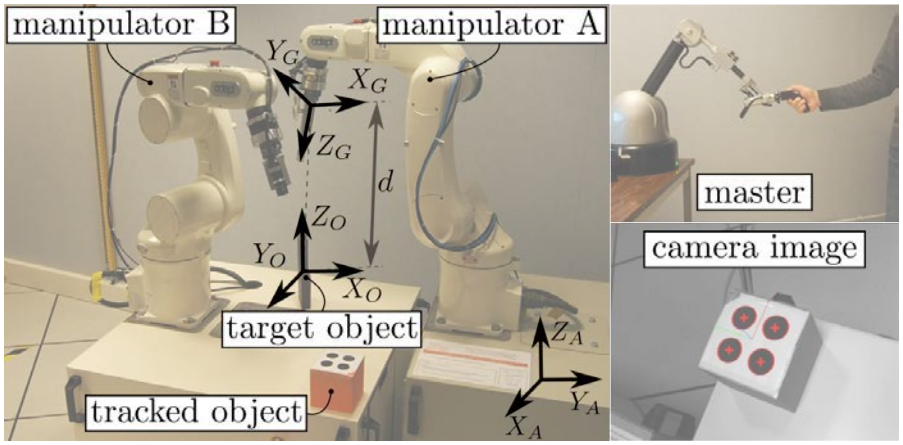


Figure 4.1: System architecture and main reference frames attached to the gripper and the object to be grasped.

4.1.1.1 System Modeling

With reference to Fig. 4.1, and similarly to the previous chapter (Sec. 3.1), the slave system consists of two 6-DoF velocity-controlled manipulator arms, A and B, with joint configurations $\boldsymbol{q}_A \in \mathbb{R}^6$ and $\boldsymbol{q}_B \in \mathbb{R}^6$, respectively. Manipulator A is (partially) controlled by a human operator, while manipulator B autonomously performs a visual task with a potential overlap with the workspace of manipulator A. We also let

$$\boldsymbol{v}_A = \begin{bmatrix} \dot{\boldsymbol{p}}_A \\ \boldsymbol{\omega}_A \end{bmatrix} = \boldsymbol{J}_A \dot{\boldsymbol{q}}_A, \qquad \boldsymbol{v}_B = \begin{bmatrix} \dot{\boldsymbol{p}}_B \\ \boldsymbol{\omega}_B \end{bmatrix} = \boldsymbol{J}_B \dot{\boldsymbol{q}}_B \tag{4.1}$$

be the Cartesian linear/angular velocities of the gripper (for manipulator A) and of the camera (for manipulator B) in their respective frames, and ($\boldsymbol{J}_A$, $\boldsymbol{J}_B$) the associated geometric Jacobians. In the following, we will equivalently consider joint velocities ($\dot{\boldsymbol{q}}_A$, $\dot{\boldsymbol{q}}_B$) or Cartesian velocities ($\boldsymbol{v}_A$, $\boldsymbol{v}_B$) as available control inputs, depending on the context. The master device is modeled as in section 3.2.3

4.1.1.2 Constraints on the Slave Side

Similarly to the approach adopted in the previous chapter, constraints at the slave side are encoded in a suitable cost function $H(\boldsymbol{q}_A, \boldsymbol{q}_B)$, whose gradient w.r.t. the joint configuration vectors is exploited to generate force cues $\boldsymbol{\tau}$ provided to the operator and to implement the reactive behavior of manipulator B (see Sec. 4.1.1.3). We consider that both manipulator arms are subject to kinematic constraints, notably joint limits and singularities, which we encode in separate cost functions, $h_J(\boldsymbol{q})$ and $h_S(\boldsymbol{q})$, as in Sec. 3.2.4. Moreover, we consider an additional constraint corresponding to potential collisions between the two slave manipulators.

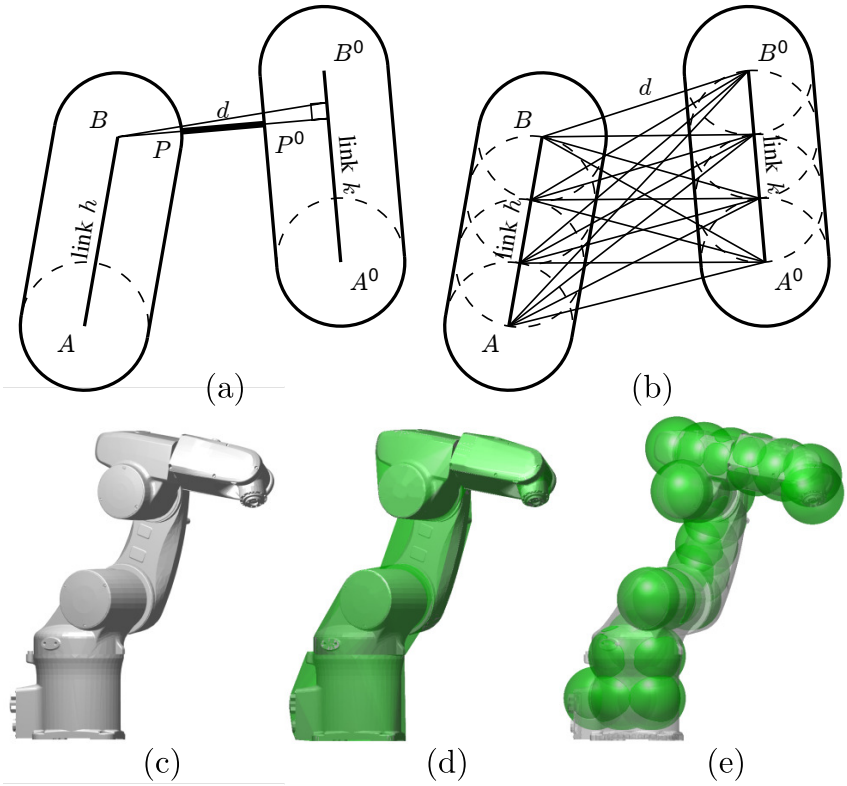


Figure 4.2: Minimum distance between two line sphere-swept bounding volumes (a) and distances among discrete sphere-swept bounding volumes (b). Solid (c), convex (d), and discrete-sphere (e) collision models of the robot.

Any cost function encoding the collision avoidance constraint needs to exploit some measure of the minimum distance between the two arms. In order to facilitate the computation of the minimum distance between two bodies, we take inspiration from the line sphere-swept model [198] (see Fig. 4.2a). However, as well known, the minimum distance between two (also convex) bounding volumes can have a discontinuous gradient when the pair of closest points on the two shapes suddenly jumps due to small perturbations of the shape locations. We therefore opted for discretizing the bounding volumes with a finite number of spheres, and exploiting all the possible inter-sphere distances to build the cost function associated to the collision avoidance (see Fig. 4.2b). An illustrative example of the resulting collision model is given in Fig. 4.2e, where one can appreciate how the adopted discretized-sphere model represents a fairly good approximation of the manipulator original and convex mesh models (Fig. 4.2c-d).

Let then $d_{ij}(\boldsymbol{q}_A, \boldsymbol{q}_B)$ represent the minimum distance between the i-th sphere on manipulator A and the j-th sphere on manipulator B. The collision avoidance

cost function is defined as

$$h_C(\boldsymbol{q}_A, \boldsymbol{q}_B) = \rho_C \sum_{i=1}^{S_A} \sum_{j=1}^{S_B} e^{-\alpha_C d_{ij}} d_{ij}^{-\beta_C}, \tag{4.2}$$

where S_A and S_B are the number of spheres used for discretizing the bodies of manipulators A and B, and $(\rho_C, \alpha_C, \beta_C)$ are scalar positive constants. One can easily verify that $h_C(\boldsymbol{q}_A, \boldsymbol{q}_B) \to \infty$ as any of the inter-sphere distances vanishes while $h_C(\boldsymbol{q}_A, \boldsymbol{q}_B) \to 0$ as all the inter-sphere distances become large enough.

Finally, the total cost function accounting for all the above mentioned constraints is

$$H(\boldsymbol{q}_A, \boldsymbol{q}_B) = h_J(\boldsymbol{q}_A) + h_J(\boldsymbol{q}_B) + h_S(\boldsymbol{q}_A) + h_S(\boldsymbol{q}_B) + h_C(\boldsymbol{q}_A, \boldsymbol{q}_B). \tag{4.3}$$

As explained, the gradient of $H(\boldsymbol{q}_A, \boldsymbol{q}_B)$ w.r.t. the joint vector $\boldsymbol{q}_A$ can be used for cueing the operator about the feasibility of her/his commands against the constraints of the slave side, while the gradient of $H(\boldsymbol{q}_A, \boldsymbol{q}_B)$ w.r.t. the joint vector $\boldsymbol{q}_B$ can be used for implementing a reactive behavior in manipulator B for avoiding possible collisions with manipulator A (see Sec. 4.1.1.3). Note that these two actions (force cues and reactive behavior) are potentially coupled because of the mixed term $h_C(\boldsymbol{q}_A, \boldsymbol{q}_B)$.

4.1.1.3 Slave Control

We start by detailing the control architecture of manipulator B, which performs an autonomous task to drive a certain state of interest $\boldsymbol{s} \in \mathbb{R}^n$ $(n < 6)$ to a desired value $\boldsymbol{s}_d$. Let $\boldsymbol{L}_{\boldsymbol{s}} \in \mathbb{R}^{n \times 6}$ be the interaction matrix associating the variation of $\boldsymbol{s}$ to the velocity of the manipulator, $\boldsymbol{v}_B$, such that $\dot{\boldsymbol{s}} = \boldsymbol{L}_{\boldsymbol{s}} \boldsymbol{v}_B$. Manipulator B is then controlled by employing the usual projected gradient control [199] (hereafter we use H to indicate $H(\boldsymbol{q}_A, \boldsymbol{q}_B)$)

$$\boldsymbol{v}_B = k_{B1} \underbrace{\boldsymbol{L}_{\boldsymbol{s}}^{\dagger}(\boldsymbol{s}_d - \boldsymbol{s})}_{\text{primary task}} - k_{B2} \underbrace{(\boldsymbol{I} - \boldsymbol{L}_{\bar{\boldsymbol{p}}}^{\dagger} \boldsymbol{L}_{\bar{\boldsymbol{p}}}) \left(\nabla_{\boldsymbol{q}_B} H \boldsymbol{J}_B^{\dagger} \right)^T}_{\substack{\text{secondary task} \\ \text{(minimize } H\text{)}}}, \tag{4.4}$$

with $k_{B1} > 0$ and $k_{B2} \geq 0$. The primary task consists in the regulation of $\boldsymbol{s}$ toward $\boldsymbol{s}_d$. Its null-space is then exploited for the secondary task of minimizing the constraint cost function H in (4.3). This null-space action will keep manipulator B away from its singularities and joint limits, as well as maintain a safe distance from manipulator A. This action can be deactivated by setting $k_{B2} = 0$, asking manipulator B to carry out only the primary visual servoing task.

As an illustrative example, we opt for a visual servoing task to keep an object of interest in visibility during the operation. Let $\bar{\boldsymbol{p}} \in \mathbb{R}^2$ represent the image plane location of a representative point on the target object (see Fig. 4.1), $\bar{\boldsymbol{p}}_d$ a desired value

for $\bar{\boldsymbol{p}}$, and $\boldsymbol{L}_{\bar{\boldsymbol{p}}} \in \mathbb{R}^{2\times 6}$ the associated point feature interaction matrix (see (3.9)), such that $\dot{\bar{\boldsymbol{p}}} = \boldsymbol{L}_{\bar{\boldsymbol{p}}}\boldsymbol{v}_B$. Equation (4.4) then becomes

$$\boldsymbol{v}_B = k_{B1} \underbrace{\boldsymbol{L}_{\bar{\boldsymbol{p}}}^{\dagger}(\bar{\boldsymbol{p}}_d - \bar{\boldsymbol{p}})}_{\substack{\text{primary task} \\ \text{(visual servoing)}}} - k_{B2} \underbrace{(\boldsymbol{I} - \boldsymbol{L}_{\bar{\boldsymbol{p}}}^{\dagger}\boldsymbol{L}_{\bar{\boldsymbol{p}}}) \left(\nabla_{\boldsymbol{q}_B} H \boldsymbol{J}_B^{\dagger}\right)^T}_{\substack{\text{secondary task} \\ \text{(minimize } H\text{)}}}. \tag{4.5}$$

For manipulator A, we instead considered two possible control modalities: *Full Teleoperation* and *Shared Control*. To simplify the notation, in each modality, a different velocity configuration vector of the master device $\boldsymbol{v}_m \in \mathbb{R}^m$ is selected where m is the number of DoF commanded by the operator. The other directions of the 6-DoF the master device are blocked.

Full Teleoperation : In this modality, the user is given full control over the 6-DoF pose of the gripper on manipulator A. This is achieved by simply setting

$$\boldsymbol{v}_A = k_A \boldsymbol{v}_m, \quad k_A > 0 \tag{4.6}$$

thereby implementing a classic velocity-velocity coupling between master and slave gripper linear/angular velocities. $\boldsymbol{v}_m \in \mathbb{R}^6$ in this modality.

Shared Control : The adopted shared-control modality is similar to that described in 3.2.1. However, it is split into two phases. In fact, if the gripper is far enough from the object, $d \geq d_{th}$, there is no need to limit the slave manoeuvrability by constraining the gripper to be oriented toward the object. Therefore, when $d \geq d_{th}$, the shared-control algorithm used here switches to a modality in which the operator has control over the *translational* motion of the gripper, while the gripper orientation is autonomously controlled so as to minimize the cost function H and, thus, stay away as much as possible from the system constraints. This is obtained by replacing (3.3) with

$$\boldsymbol{v}_A = k_{A1}\boldsymbol{S}\boldsymbol{v}_m - k_{A2}\boldsymbol{Z}\left(\nabla_{\boldsymbol{q}_A} H \boldsymbol{J}_A^{\dagger}\right)^T, \tag{4.7}$$

where

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{I}_3 \\ \boldsymbol{0} \end{bmatrix} \in \mathbb{R}^{6\times 3} \text{ and } \boldsymbol{Z} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_3 \end{bmatrix} \in \mathbb{R}^{6\times 6}. \tag{4.8}$$

The configuration vector of the master device $\boldsymbol{v}_m \in \mathbb{R}^3$ is its linear velocity.

The control law in (4.7) ensures the minimization of H when $\boldsymbol{v}_m = \boldsymbol{0}$ (i.e., no translational motion) and $\boldsymbol{v}_B = \boldsymbol{0}$ (manipulator B does not move) by acting on the

angular velocity of manipulator A. This can be proven by relating the velocity of manipulator A to the variation of H where

$$\dot{H} = \nabla_{\boldsymbol{q}_A} H \dot{\boldsymbol{q}}_A + \vartheta = \nabla_{\boldsymbol{q}_A} H \boldsymbol{J}_A^{\dagger} \boldsymbol{v}_A + \vartheta, \tag{4.9}$$

where ϑ is the variation resulting from manipulator B and is zero if B is static. If we plug $\boldsymbol{v}_A$ into (4.9) we will then get

$$\dot{H} = -\left(\nabla_{\boldsymbol{q}_A} H \boldsymbol{J}_A^{\dagger}\right) k_{A2} \boldsymbol{Z} \left(\nabla_{\boldsymbol{q}_A} H \boldsymbol{J}_A^{\dagger}\right)^T < 0. \tag{4.10}$$

On the other hand, if the gripper is close enough to the object $d < d_{th}$, the shared-control architecture described in Sec. 3.2 is adopted and the control law governing the motion of the gripper is as in (3.8)

$$\boldsymbol{v}_A = k_{A1} \boldsymbol{L}_s^{\dagger} \boldsymbol{e}_z + k_{A2} \boldsymbol{N} \boldsymbol{v}_m. \tag{4.11}$$

where $\boldsymbol{v}_m \in \mathbb{R}^4$ and the forward map between the master and slave is implemented in velocity-velocity mode similarly to Sec. 3.2.3.1.

4.1.1.4 Haptic guidance

As mentioned before, the control forces $\boldsymbol{f}$ on the master side (3.27) are exploited to inform the operator about the feasibility of her/his commands against the system constraints, encoded in the cost function H. This is obtained as follows, depending on the control modality considered for manipulator A:

Full Teleoperation : In this case, manipulator A is controlled by (4.6), and the force cues are generated as

$$\boldsymbol{f} = -k_f \left(\nabla_{\boldsymbol{q}_A} H \boldsymbol{J}_A^{\dagger}\right)^T, \qquad k_M > 0, \tag{4.12}$$

the rationale being that (4.12) provides the force that, when applied to all the master device DoF, would make the master move so as to minimize H at the slave side. Therefore, cues (4.12) provide the human operator with information about where to move in order to stay away from the constraints.

Shared Control : When manipulator A is close enough to the object ($d < d_{th}$), its behavior is controlled by (3.8). In this case, the force cues are generated as described in (3.24)

$$\boldsymbol{f}_m = -k_f \left(\nabla_{\boldsymbol{q}_A} H \boldsymbol{J}_A^{\dagger} \boldsymbol{N}\right)^T, \qquad k_M > 0. \tag{4.13}$$

Similarly to the previous case, cues (4.13) represent the forces that should be applied to the free directions of the master device for letting the manipulator A minimize H along the null-space directions spanned by $\boldsymbol{N}$ (where the operator can act).

When manipulator A is instead far from the object ($d \geq d_{th}$), its behavior is regulated by (4.7), and the force cues are

$$\boldsymbol{f}_m = -k_f \left(\nabla_{\boldsymbol{q}_A} H \boldsymbol{J}_A^{\dagger} \boldsymbol{S} \right), \quad k_M > 0, \tag{4.14}$$

with again the idea of providing a force feedback that, when applied to the free master DoF (i.e., the translational ones in this case), would make manipulator A move so as to minimize H.

4.1.2 Passivity Analysis

As discussed in the previous chapter (see Sec. 3.2.5), passivity is an important aspect when dealing with teleoperation systems for guaranteeing a stable closed-loop behavior. In this section, our aim is to analyze the passivity of the considered teleoperation system subject to the control modalities introduced in Secs. 4.1.1.3 and 4.1.1.4. As in Sec. 3.2.5, we consider the two velocity-controlled robots as simple integrators and the total energy of the system can again be written as

$$\mathcal{V}(\boldsymbol{l}_m, \boldsymbol{q}_A, \boldsymbol{q}_B) = \frac{1}{2} \boldsymbol{l}_m^{\mathrm{T}} \boldsymbol{M}_m^{-1} \boldsymbol{l}_m + H(\boldsymbol{q}_A, \boldsymbol{q}_B) \tag{4.15}$$

where $\boldsymbol{l}_m = \boldsymbol{M}_m \boldsymbol{v}_m$ is the haptic device momentum, $\boldsymbol{q}_A$, $\boldsymbol{q}_B \in \mathbb{R}^6$ are the generalized coordinates of the two slave manipulators, respectively, and $H(\boldsymbol{q}_A, \boldsymbol{q}_B)$, introduced in (4.3), is the potential energy associated with the system constraints. It is then sufficient to put the closed-loop system equations in port-Hamiltonian form to prove the stability of the system.

In the following, we then show that the three control modalities described in Sec. 4.1.1 lead to a closed-loop PHS formulation. For the analysis, we consider the primary task in (4.5) and (4.11) to have reached a steady state ($\boldsymbol{s} \to \boldsymbol{s}_d$ and $\boldsymbol{l} \to \boldsymbol{l}_d$).

Full teleoperation: setting $k_f = k_A = k$, the closed-loop system can be written as follows

$$\begin{bmatrix} \dot{\boldsymbol{l}}_m \\ \dot{\boldsymbol{q}}_A \\ \dot{\boldsymbol{q}}_B \end{bmatrix} = \left(\begin{bmatrix} 0 & -k{\boldsymbol{J}_A^{\dagger}}^T & 0 \\ k\boldsymbol{J}_A^{\dagger} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \right. \\ \left. - \begin{bmatrix} \boldsymbol{B}_m & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \boldsymbol{P} \end{bmatrix} \right) \begin{bmatrix} \nabla_{\boldsymbol{l}_m} \mathcal{V} \\ \nabla_{\boldsymbol{q}_A} \mathcal{V} \\ \nabla_{\boldsymbol{q}_B} \mathcal{V} \end{bmatrix} + \begin{bmatrix} \mathbf{1} \\ 0 \\ 0 \end{bmatrix} \boldsymbol{f}_h. \tag{4.16}$$

Shared control ($d < d_{th}$): setting $k_f = k_{A2} = k$, the closed-loop system can be written as follows

$$\begin{bmatrix} \dot{\boldsymbol{l}}_m \\ \dot{\boldsymbol{q}}_A \\ \dot{\boldsymbol{q}}_B \end{bmatrix} = \left(\begin{bmatrix} 0 & -k\left(\boldsymbol{J}_A^{\dagger}\boldsymbol{N}\right)^T & 0 \\ k\boldsymbol{J}_A^{\dagger}\boldsymbol{N} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \right. \\ \left. - \begin{bmatrix} \boldsymbol{B}_m & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \boldsymbol{P} \end{bmatrix} \right) \begin{bmatrix} \nabla_{\boldsymbol{l}_m}\mathcal{V} \\ \nabla_{\boldsymbol{q}_A}\mathcal{V} \\ \nabla_{\boldsymbol{q}_B}\mathcal{V} \end{bmatrix} + \begin{bmatrix} \mathbf{1} \\ 0 \\ 0 \end{bmatrix} \boldsymbol{f}_h. \tag{4.17}$$

Shared control ($d \geq d_{th}$): setting $k_f = k_{A1} = k$, the closed-loop system can be written as follows

$$\begin{bmatrix} \dot{\boldsymbol{l}}_m \\ \dot{\boldsymbol{q}}_A \\ \dot{\boldsymbol{q}}_B \end{bmatrix} = \left(\begin{bmatrix} 0 & -k\left(\boldsymbol{J}_A^{\dagger}\boldsymbol{S}\right)^T & 0 \\ k\boldsymbol{J}_A^{\dagger}\boldsymbol{S} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \right. \\ \left. - \begin{bmatrix} \boldsymbol{B}_m & 0 & 0 \\ 0 & k_{A2}\boldsymbol{J}_A^{-1}\boldsymbol{Z}\boldsymbol{J}_A^{-T} & 0 \\ 0 & 0 & \boldsymbol{P} \end{bmatrix} \right) \begin{bmatrix} \nabla_{\boldsymbol{l}_m}\mathcal{V} \\ \nabla_{\boldsymbol{q}_A}\mathcal{V} \\ \nabla_{\boldsymbol{q}_B}\mathcal{V} \end{bmatrix} + \begin{bmatrix} \mathbf{1} \\ 0 \\ 0 \end{bmatrix} \boldsymbol{f}_h, \tag{4.18}$$

where $\boldsymbol{P} = \left(\boldsymbol{I} - (\boldsymbol{L}_s\boldsymbol{J}_B)^{\dagger}(\boldsymbol{L}_s\boldsymbol{J}_B)\right)$ is a null-space projector such that $\boldsymbol{P} = \boldsymbol{P}^T \geq 0$. Therefore, since in all cases the closed-loop systems can be put in a PHS form (the resulting interconnection and dissipation matrices being always skew-symmetric and positive semi-definite, respectively), one can conclude the passivity of the three modalities w.r.t. the pair $(\boldsymbol{v}_m, \boldsymbol{f}_h)$ with energy function $\mathcal{V}(\boldsymbol{l}_m, \boldsymbol{q}_A, \boldsymbol{q}_B)$ as sought[1].

4.1.3 Experiments in Simulation

4.1.3.1 Experimental setup, task, and participants

The experimental setup is shown in Fig. 4.3. The slave side is simulated using V-REP, and it is composed of our two velocity-controlled manipulators: an Adept Viper 850, controlled by the human operator (manipulator A), and an Adept Viper 650, controlled by an autonomous algorithm (manipulator B). The master side is composed of a Haption Virtuose 6D Desktop interface, which is used to control

[1] We note that the presented analysis does not account for the controller switching in the shared-control modality when close/far from the target object. If this switch needs to be taken into account because of non-negligible effects on the total energy, one could employ the energy tank machinery for passifying potential instabilities due to the switching mechanism [200]. However, we empirically found the switching to have a negligible effect on the system stability during our experiments.

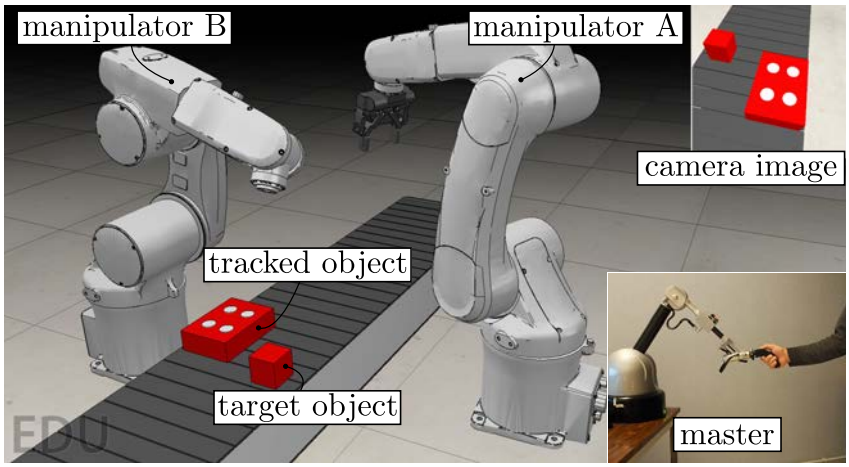


Figure 4.3: Experimental setup used for user study: slave side composed of simulated robots and vision; master side comprising the haptic interface.

the Viper 850 robot and provides kinesthetic haptic feedback. Manipulator A is endowed with a ROBOTIQ 2-finger gripper, while manipulator B is equipped with a vision sensor. The remote environment is composed of two objects, a cube and a rectangular parallelepiped, placed on a conveyor belt. As detailed in Sec. 4.1.1, the autonomous manipulator B robot is in charge of tracking the parallelepiped using standard visual servoing techniques [201]. On the other hand, participants are required to control the motion of manipulator A to grasp the cube and lift it from the ground, avoiding collisions with the fully-autonomous robot. The task starts when the manipulator moves for the very first time, and it is considered successfully completed when the object is lifted from the ground. All the control policies described are implemented in MATLAB/SIMULINK and interfaced with V-REP using ROS through the matlab_ros_bridge (`https://github.com/lagadic/matlab_ros_bridge`). The control loop runs at 100 Hz. A video of the experiment is available at `https://youtu.be/uWzVEGW2i5c`.

Fifteen right-handed subjects (average age 25.9, 11 males, 4 females) participated in the study. Four of them had previous experience with haptic interfaces. Each subject spent about ten minutes practicing the control of the telemanipulation system before starting the experiment. Participants were briefed about all the tasks and afterwards signed an informed consent, including the declaration of having no conflict of interest. All of them were able to give the consent autonomously. The study was done in accordance with the 1964 Helsinki declaration and its later amendments.

4.1.3.2 Experimental conditions

We considered two different levels of human involvement in the control of manipulator A (teleoperation vs. shared control, T vs. S), two haptic feedback modalities (haptic feedback about the distance from the second robot and workspace/joints constraints vs. no haptic feedback, H vs. $\not{\text{H}}$), and two control policies for manipulator B (reactive vs. non-reactive, R vs. $\not{\text{R}}$).

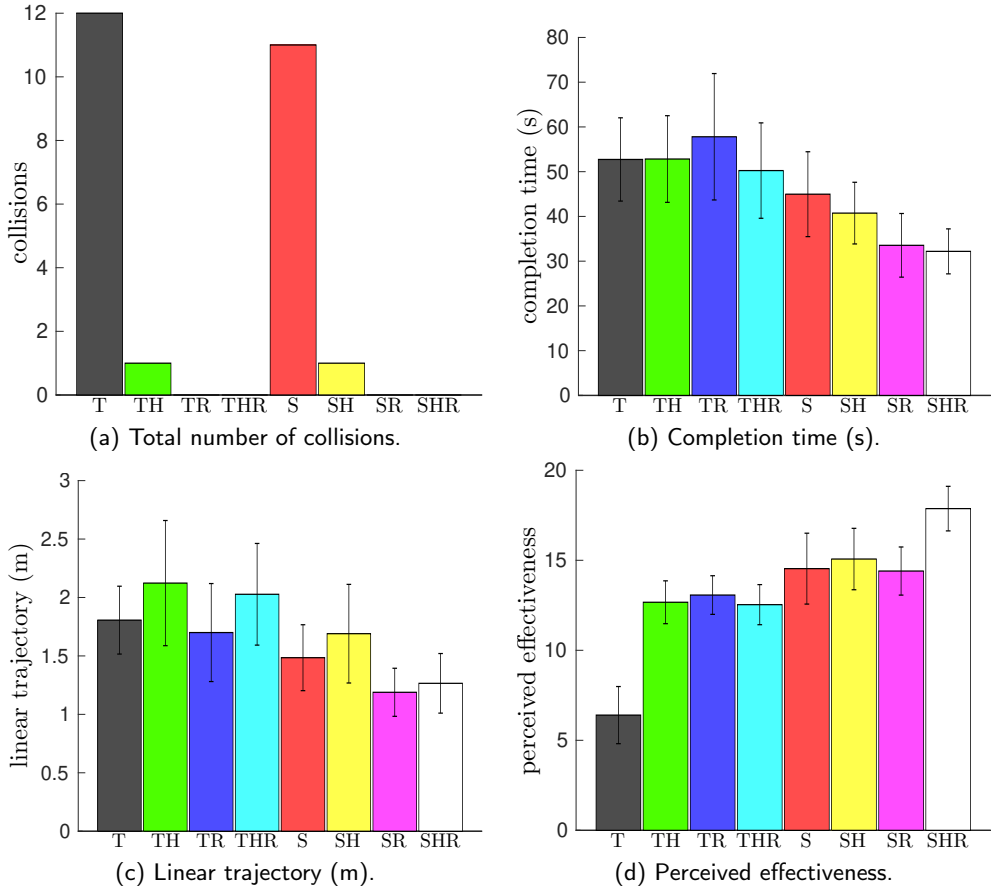
Human involvement (T vs. S) : In condition T, subjects are able to control all the 6 DoF of manipulator A through the grounded master interface. Conversely, in condition S, the orientation of the gripper is controlled by an autonomous algorithm. When the gripper is close to the object to grasp, it is automatically oriented towards it; otherwise, the gripper is oriented to stay as much as possible away from the system constraints (see Sec. 4.1.1.3).

Haptic feedback (H vs. $\not{\text{H}}$) : When haptic feedback is activated (H), subjects receive haptic stimuli about the feasibility of their commands against system constraints, such as joint limits, singular configurations, and collisions with manipulator B (see Sec. 4.1.1.4). Conversely, in condition $\not{\text{H}}$, subjects do not receive any haptic feedback.

Control of the fully-autonomous manipulator (R vs. $\not{\text{R}}$) : Manipulator B always performs an autonomous visual task to keep an object of interest visible. In condition R, it also uses the null-space of the above-mentioned primary visual task to minimize the constraint cost function $H(\boldsymbol{q}_A, \boldsymbol{q}_B)$ (see Sec. 4.1.1.3). Notably, this secondary action will enable manipulator B to keep a safe distance from manipulator A, re-actively moving away when the latter comes too close. Conversely, in $\not{\text{R}}$, manipulator B only focuses on the primary visual task, indifferent to what the other manipulator does.

Considering all the possible combinations, we ended up with eight different experimental conditions: THR, TH$\not{\text{R}}$, T$\not{\text{H}}$R, T$\not{\text{H}}\not{\text{R}}$, SHR, SH$\not{\text{R}}$, S$\not{\text{H}}R, S\not{\text{H}}\not{\text{R}}$. For brevity, from now on we will omit the $\not{\text{H}}$ and $\not{\text{R}}$ variables (e.g., S$\not{\text{H}}\not{\text{R}}$ is called S).

The cost functions introduced in Sec. 4.1.1 include several parameters, which let us control the curvature, rate of increase, and proximity to the limits of these functions. The choice of these parameters is challenging, system-dependent, and very delicate, as it must ensure a smooth force feedback. To choose the right parameters for our system and target application, we asked 2 expert operators to repeatedly carry out the task, changing the parameters at runtime (i.e., ρ_C, α_C, $\beta_C, k_{A1}, k_{A2}, k_{B1}, k_{B2}, k_f$) to make the teleoperation as intuitive, safe, and comfort-



(a) Total number of collisions. (b) Completion time (s). (c) Linear trajectory (m). (d) Perceived effectiveness.

Figure 4.4: Experimental evaluation. Mean and 95% confidence interval of (a) total number of collisions, (b) completion time, (c) linear motion, and (d) perceived effectiveness of the eight feedback conditions are plotted.

able as possible. Finally, we asked them to find a consensus on the parameters' values and we used those in our implementation.

4.1.3.3 Results

Average task success rate across conditions was 92.4 ± 6.3%. A Friedman test showed no statistically significant difference between the means of the eight feedback conditions. Figure 4.4a shows the total number of collisions occurred during the experiment between the two manipulators.

To compare other metrics, we ran three-way repeated-measures ANOVA tests on the data. Human involvement in the control (shared control vs. teleoperation, S vs. T), presence of haptic feedback (haptic feedback vs. no haptic feedback, H vs. $\not$H), and behavior of the fully-autonomous robot (reactive vs. non-reactive, R vs. $\not$R) were treated as within-subject factors. All data passed the Shapiro-Wilk normality test. Sphericity was assumed for all variables, since they all have two levels of

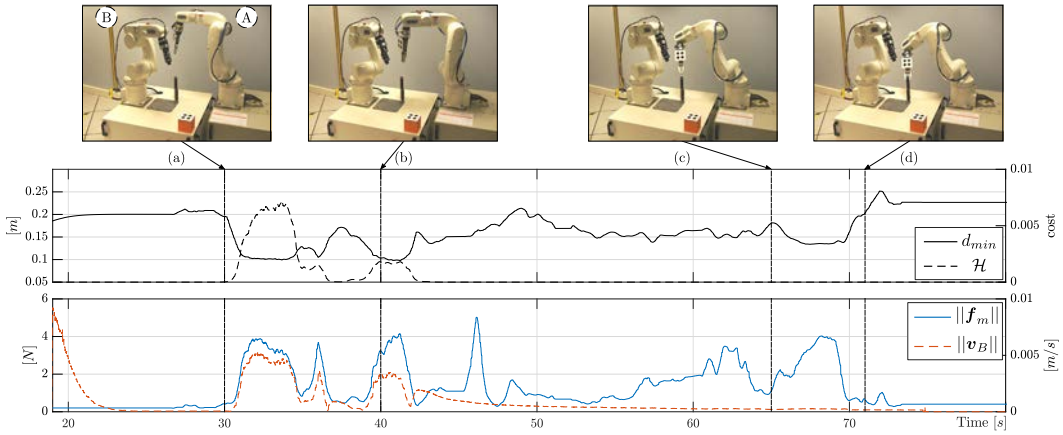


Figure 4.5: Data time history recorded during the real experiments. Upper graph: minimum distance d_{min} and collision cost value H; bottom graph: haptic guidance $\boldsymbol{f}_m$ and escaping velocity $\dot{\boldsymbol{p}}_B$ norms.

repeated measures. Interaction effects between the factors were not statistically significant. Figure 4.4b shows the completion time. The three-way ANOVA test revealed a statistically significant change in the metric for the human involvement in the control ($F(1, 14) = 52.165$, $p < 0.001$, shared control was better) and the control behavior for manipulator B ($F(1, 14) = 6.400$, $p = 0.024$, reactive mode was better) variables. Figure 4.4c shows the linear motion covered by the robotic gripper during the task. The three-way ANOVA test revealed a statistically significant change in the metric for the human involvement in the control ($F(1, 14) = 13.599$, $p = 0.002$, shared control was better) and the control behavior for the second robot ($F(1, 14) = 6.567$, $p = 0.023$, reactive mode was better) variables. Immediately after the experiment, subjects were also asked to report the effectiveness of each feedback condition in completing the given task using bipolar Likert-type twenty-two-point scales. Figure 4.4d shows the perceived effectiveness for the eight experimental conditions. The three-way ANOVA test revealed a statistically significant change in the metric for the human involvement in the control ($F(1, 14) = 34.700$, $p < 0.001$), the presence of haptic feedback ($F(1, 14) = 33.217$, $p < 0.001$, shared control was better), and the control behavior for the second robot ($F(1, 14) = 25.305$, $p < 0.001$, reactive mode was better) variables. Finally, all fifteen subjects found conditions using the shared-control approach to be the most effective at completing the grasping task. Ten subjects chose SHR as the most effective, three SH, and two SR.

4.1.4 Experiment in a Real Environment

We also carried out an experiment in the real environment. The setup and task are similar to the simulated scenario of Sec. 4.1.3. The setup is shown in Fig. 4.1. Five right-handed subjects participated in the study. Three of them had previous experience with haptic interfaces. In this real-world case, we implemented control modalities TR, THR, SR, and SHR, which led to no collisions in Sec. 4.1.3. Of course, in this real-world case, we could not consider conditions which may lead to a collision between the two manipulators. All subjects successfully completed the task in all conditions and no collisions occurred. All subjects chose SHR to be the most intuitive and effective condition. A video of the experiment in the SHR condition (shared control, haptic feedback, and reactive control of manipulator B) is available at https://youtu.be/uWzVEGW2i5c. A sequence of frames taken from the video is shown in Fig. 4.5, where it is possible to recognize all the relevant phases of the grasping task. In snapshots (a)-(b), it is possible to see the manipulator B reactively avoiding collisions while keeping track of the desired object. In snapshots (c)-(d), the shared-control algorithm is active and automatically orients the gripper of manipulator A toward the object to grasp. From the bottom graphs, we can see that the first half of the experiment is characterized by persistent haptic forces and by a significant reactive velocity of manipulator B. In the second half, the risk of collisions is lower and the operator can safely approach the object, aided by the action of the shared-control algorithm.

4.1.5 Discussion

In the described experiments, fifteen subjects used different control modes to control a manipulator A, equipped with a gripper, for approaching and grasping a target object placed on a conveyor belt. We tested eight experimental conditions, considering two levels of human involvement in the control (shared control vs. classic teleoperation, S vs. T), two feedback modalities (haptic feedback about imminent collisions and workspace/joints constraints vs. no haptic feedback, H vs. $\not$H), and two control policies for manipulator B (reactively moving away from the other manipulator vs. non-reactive/still, R vs. $\not$R). Results proved the effectiveness and viability of our haptic-enabled shared-control approaches. Using shared control (S) on manipulator A and the reactive mode (R) on manipulator B significantly improved the performance in most metrics (completion time, linear motion, perceived effectiveness). Conditions employing shared control were also the most preferred, confirming the all-round viability of such approach. Moreover, as expected, in conditions R, manipulator B was always able to prevent collisions with the other robot by moving away when the latter was approaching. Nonetheless, even in conditions

$\not{R}$, repulsive haptic feedback (H) provided when the robots were too close showed good results (only two collisions happened in conditions TH, SH). This result is very promising, as haptic feedback acts only at the master side, leaving the action of manipulator B unaffected. In this respect, it is interesting to notice that, although the applied force $\boldsymbol{f}_m$ should go to infinite as the distance between the two robots goes to zero (see (4.2)), we still experienced two collisions in conditions H. This is due to the limited actuation capabilities of our haptic interface, which is obviously not able to provide arbitrarily high forces. Finally, the experiment in the real scenario confirmed the results obtained with the simulated setup. Haptic guidance effectively steered the user toward the safe zones of the workspace, the reactive behavior enabled a safe interaction between manipulators, and the shared control made the task fast, easy, and intuitive to complete.

4.2 Grasping Assistance in a Cluttered Environment

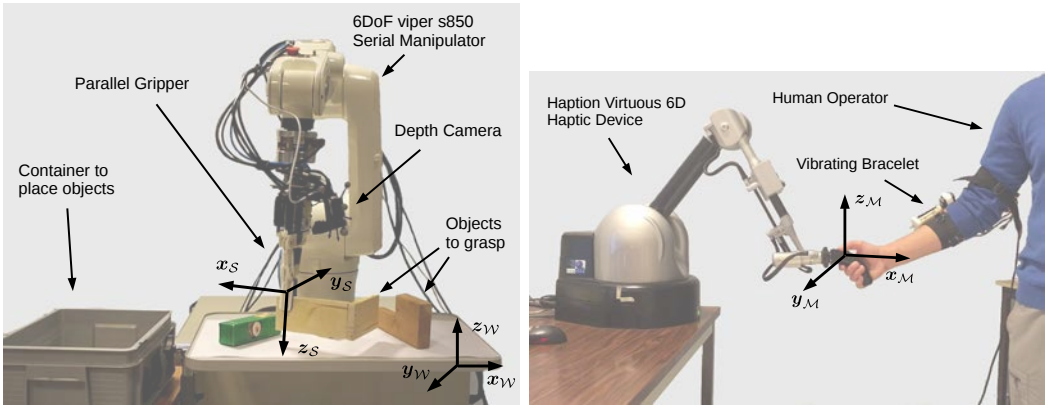
Although the approach proposed in chapter 3 proved to be quite efficient and robust, it has two significant limitations. First, it can only consider one object at a time: the user has to choose the object to grasp at the beginning of the task and cannot intuitively switch to another one in the scene. Second, the algorithm keeps the gripper oriented toward the object's center of mass, which may not be the best way of grasping the considered object. The grasping will fail if the object needs to be grasped otherwise.

This section presents a more general shared-control approach enabling the human operator to intuitively handle multiple objects with different shapes. A point cloud scan of the target environment is used to find potential grasp candidates for all the objects in the scene. These grasping poses are then used by the algorithm to provide haptic guidance to the human operator. Dynamic active constraints gently steer the operator toward feasible grasping poses, enabling her/him to intuitively navigate the environment and safely switch between different grasping candidates placed on the same or on a different object. Moreover, the algorithm also ensures that the operator complies with certain constraints of the system by introducing additional active constraints. In order to enable the operator to differentiate between these two haptic cues (guiding toward a grasping pose vs. proximity to unsafe configuration), we use kinesthetic and vibrotactile feedback. Active constraints providing grasping guidance are enforced by conveying kinesthetic feedback through a 6-DoF grounded haptic interface; active constraints enforcing the safety limits are conveyed via kinesthetic feedback provided through the same 6-DoF haptic interface *and* vibrotactile feedback provided by a custom haptic bracelet. A picture of our robotic setup and representative cluttered scene is shown in Fig. 4.6.

In the following, the teleoperation system is detailed in sec. 4.2.1 after which the algorithm used for generating the grasp candidates from the point cloud is described in sec. 4.2.2; Sec. 4.2.3 describes the shared control algorithm as well as how the active constraints are generated and enforced; Sec. 4.2.4 details the two experimental evaluations, which are then discussed in Sec. 4.2.5.

4.2.1 Overview on the Architecture

Fig. 4.7 illustrates the proposed framework. The human operator commands the system by applying a force $\boldsymbol{\tau}_h$ to a grounded haptic interface. She/He is then guided toward the potential grasp candidates ($\boldsymbol{\tau}_g$) while being kept away from possibly unsafe kinematic configurations of the system ($\boldsymbol{\tau}_c$). This information is provided to the operator via a combination of kinesthetic and vibrotactile stimuli, provided through the grounded haptic interface and a vibrotactile bracelet, respectively. The



(a) Slave side (b) Master side

Figure 4.6: The experimental setup showing the slave robotic arm on the top and the master haptic arm on the bottom.

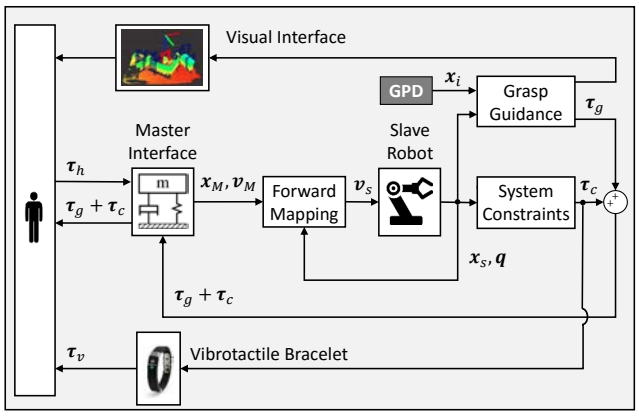


Figure 4.7: A schematic illustration of the proposed architecture.

user is also provided with a visual representation of the scene showing the point cloud, the current gripper pose, and the grasping candidates (as in Fig. 4.8). Finally, the pose of the master device $\boldsymbol{x}_m$ and its velocity $\boldsymbol{v}_m$ are mapped into velocity commands $\boldsymbol{v}_s$ driving the slave telemanipulator. Details on how these quantities are calculated are reported in the following sections.

To demonstrate the feasibility and effectiveness of the proposed approach, we employed the robotic system shown in Fig. 4.6. The master side consists of a Haption Virtuose 6D haptic interface, a high performance grounded device with three active translational DoF and three active rotational DoF. The slave side is composed of an Adept Viper s850 6-DoF serial manipulator equipped with a Sake EzGripper and an Intel Realsense SR-300 RGB-D sensor. The remote environment is composed of different objects placed on a small table in front of the manipulator. Of course, the proposed shared control approach is quite general and can be used

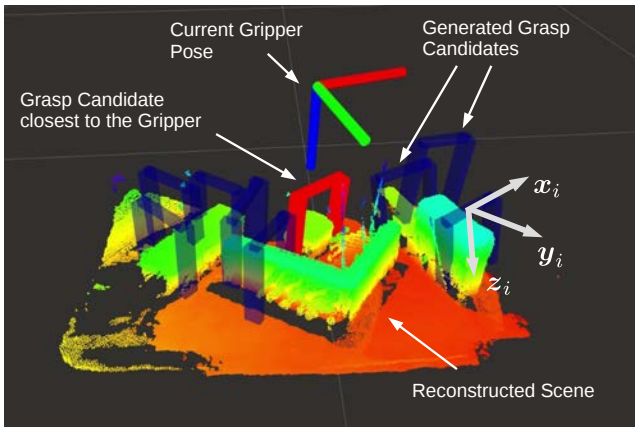


Figure 4.8: A screenshot of the visual feedback. A point cloud that was generated by an automated scanning routine serves as 3D reconstruction of the scene. The grasp candidates produced by the grasp planner are shown in blue, except for the one that is currently used for computing the haptic feedback, which is drawn in red. The current pose of the endeffector is indicated by a coordinate frame.

with any similar master-slave robotic system.

4.2.2 Point Cloud and Grasp Pose Generation

At the beginning of the task, we generate potential grasp candidates using a point cloud model of the scene. The same point cloud is also used during the teleoperation to provide visual feedback to the human operator.

To retrieve a comprehensive view of the environment, we attached a RGB-D camera to the end-effector of our robotic manipulator, as shown in Fig. 4.6, and we performed an automated scanning of the scene. The scanning routine iteratively builds a point cloud by driving the robot to 18 different pre-programmed positions around the scene. At each position, a new point cloud is recorded and merged with the previous ones using the Iterative Closest Point (ICP) algorithm [202].

The final point cloud is then used to find 6-DoF grasp candidates. To do so, we employed the grasp pose detection (GPD) algorithm [203, 204], which provides us with a list of grasp poses scored according to their predicted performance. However, we cannot directly use the grasp candidates produced by GPD algorithm, as they are often cluttered and include poses that are difficult or impossible to reach with our robotic system. Therefore, we performed an additional filtering on the candidates generated by GPD. Firstly, we discarded all grasps that were not within the workspace of our robot. Then, we selected the most promising candidates, based on their GPD score and distribution over the objects. For the scene shown in Fig. 4.8, the number of remaining grasp poses was $N_g = 9$. Throughout the paper, we will re-

fer to the grasp poses by $\boldsymbol{x}_i = (\boldsymbol{p}_i,\ \boldsymbol{R}_i)$, $1 \leq i \leq N_g$, where $\boldsymbol{p}_i \in \mathbb{R}^3$ and $\boldsymbol{R}_i \in \mathcal{SO}(3)$ represent the corresponding gripper position and orientation in a fixed world frame $\mathcal{W}$, respectively. The set of grasp candidates after filtering is shown in Fig. 4.8.

4.2.3 Haptic shared-control architecture

The human operator is given control over the full six DoF of the manipulator through a position-force bilateral teleoperation coupling. She/He commands the motion of the slave robot by applying forces $\boldsymbol{\tau}_h$ on the master handle, and, at the same time, she/he receives haptic feedback about potential grasp candidates $\boldsymbol{\tau}_g$ and the presence of possibly unsafe configurations of the system $\boldsymbol{\tau}_c$ (see Fig. 4.7).

When the operator is far from any object, she/he only receives haptic feedback regarding the presence of possibly unsafe kinematic configurations, e.g., joint limits and singularities. As she/he gets within a pre-defined distance $d_{\max}$ from any grasp candidate, she/he also starts to receive haptic cues guiding her/him toward the closest suitable pose. To enable the operator to differentiate between these two haptic cues (guiding toward a grasping pose vs. proximity to unsafe configuration), we use kinesthetic and vibrotactile feedback. If the haptic feedback is due to the proximity to a kinematic constraint, the operator is provided with kinethetic *and* vibrotactile feedback. On the other hand, if the cues are guiding the operator toward a grasping pose, we only provide kinesthetic feedback.

4.2.3.1 Master/slave coupling

We consider the frames of reference $\mathcal{F}_m$, attached to the end-effector of the master interface, $\mathcal{F}_s$, attached to the end-effector of the slave manipulator, and $\mathcal{W}$, a fixed world frame (see Fig. 4.6). Let the pose of frame $\mathcal{F}_m$ w.r.t. $\mathcal{W}$, expressed in $\mathcal{W}$, be denoted by $\boldsymbol{x}_m = (\boldsymbol{p}_m,\ \boldsymbol{R}_m) \in \mathbb{R}^3 \times \mathcal{SO}(3)$. Similarly, let $\boldsymbol{x}_s = (\boldsymbol{p}_s,\ \boldsymbol{R}_s) \in \mathbb{R}^3 \times \mathcal{SO}(3)$ represent the poses of $\mathcal{F}_s$ w.r.t. $\mathcal{W}$, always expressed in $\mathcal{W}$. Finally, the translational and rotational velocities of $\mathcal{F}_m$ and $\mathcal{F}_s$ are defined in $\mathcal{W}$ as $\boldsymbol{v}_m = (\dot{\boldsymbol{p}}_m^T, \boldsymbol{\omega}_m^T)^T \in \mathbb{R}^6$ and $\boldsymbol{v}_s = (\dot{\boldsymbol{p}}_s^T, \boldsymbol{\omega}_s^T)^T \in \mathbb{R}^6$, respectively.

The master interface is modelled, similarly to the previous chapter, as a gravity pre-compensated mechanical system. However, the 6 DoF of the device are all exploited and its dynamic model can be written as

$$\boldsymbol{M}_m(\boldsymbol{x}_m)\dot{\boldsymbol{v}}_m + \boldsymbol{C}_m(\boldsymbol{x}_m,\ \boldsymbol{v}_m)\boldsymbol{v}_m = \boldsymbol{\tau} + \boldsymbol{\tau}_h, \tag{4.19}$$

where $\boldsymbol{M}_m(\boldsymbol{x}_m) \in \mathbb{R}^{6\times6}$ is the positive-definite and symmetric inertia matrix, $\boldsymbol{C}_m(\boldsymbol{x}_m,\ \boldsymbol{v}_m) \in \mathbb{R}^{6\times6}$ accounts for the Coriolis/centrifugal terms, and $\boldsymbol{\tau}$, $\boldsymbol{\tau}_h \in \mathbb{R}^6$ are the feedback and human forces applied at the master handle, respectively. In

this case, haptic feedback $\boldsymbol{\tau}$ accounts for three components,

$$\boldsymbol{\tau} = \boldsymbol{\tau}_c + \boldsymbol{\tau}_g - B_m \boldsymbol{v}_m, \tag{4.20}$$

where $\boldsymbol{\tau}_c$ are the haptic cues informing the operator about the kinematic constraints of the system (see Sec. 4.2.3.2), $\boldsymbol{\tau}_g$ the haptic cues guiding the operator toward the proposed grasp candidates (see Sec. 4.2.3.3), and $\boldsymbol{B}_m \in \mathbb{R}^{6\times6}$ a damping factor which improves the stability of the system.

As for the slave, we assume that its end-effector can be controlled in velocity (as most industrial manipulators, including our Viper robot). The slave is coupled to the master with a classical position-position cartesian coupling modulo a roto-translational scale, such that

$$\boldsymbol{v}_s = \lambda \begin{bmatrix} \Delta\boldsymbol{p}_m - \Delta\boldsymbol{p}_s \\ \boldsymbol{R}_s \, {}^s\theta\boldsymbol{u}_{s,d} \end{bmatrix} + \boldsymbol{v}_m, \tag{4.21}$$

where λ is a control gain, $\Delta\boldsymbol{p}_m = \boldsymbol{p}_m - \boldsymbol{p}_{m,0}$ and $\Delta\boldsymbol{p}_s = \boldsymbol{p}_s - \boldsymbol{p}_{s,0}$, with $\boldsymbol{p}_{m,0}$ and $\boldsymbol{p}_{s,0}$ representing the respective positions of the master and the slave at the beginning of the experiment. Similarly for the orientation, ${}^s\theta\boldsymbol{u}_{s,d}$ is the angle-axis representation of the relative rotation between the desired and the current orientation of the slave ${}^s\boldsymbol{R}_{s,d} = {}^{s_0}\boldsymbol{R}_s^T \, {}^{m_0}\boldsymbol{R}_m$, where ${}^{s_0}\boldsymbol{R}_s$ is the current orientation of the slave w.r.t. its orientation at the beginning of the experiment and ${}^{m_0}\boldsymbol{R}_m$ is the current orientation of the master w.r.t. its orientation at the beginning of the experiment. Therefore, the first term of (4.21) ensures a good responsiveness of the system, while the second term prevents drifts.

4.2.3.2 Haptic guidance to avoid kinematic constraints

We use kinesthetic and vibrotactile haptic feedback to keep the human operator away from possibly dangerous configurations of the system, namely joint limits and singularities. The constraints are described, as in Sec. 3.2.4, via cost functions $h_J(\boldsymbol{q})$ and $h_S(\boldsymbol{q})$ depicting the proximity of the manipulator to joint limits and singularities respectively. The gradient of $\mathcal{H}(\boldsymbol{q}) = h_J(\boldsymbol{q}) + h_S(\boldsymbol{q})$ w.r.t. the joint configuration vector $\partial\mathcal{H}(\boldsymbol{q})/\partial\boldsymbol{q}$ can then be used to generate the haptic feedback $\boldsymbol{\tau}_c$ provided to the operator such that

$$\boldsymbol{\tau}_c = -k_f \left(\frac{\partial\mathcal{H}(\boldsymbol{q})}{\partial\boldsymbol{q}} \boldsymbol{J}^\dagger \right)^T. \tag{4.22}$$

Haptic feedback $\boldsymbol{\tau}_c$ is provided to the human operator through the grounded master interface. However, to enable the operator to differentiate the action of $\boldsymbol{\tau}_c$ from that of $\boldsymbol{\tau}_g$ (check Sec. 4.2.3.3), we also provide additional vibrotactile haptic

feedback. In this work, we provide the additional vibrotactile cues through a haptic armband [55] (see Fig. 4.6), which is composed of four Precision Microdrives 307-100 Pico Vibe 9mm vibration motors, an Arduino Mini Pro 3.3 V, a 3.7 V LiPo battery, and a RN-42 Bluetooth 2.1 antenna. The electronics and battery are embedded into a 3D-printed case, and the same is done for each motor. The devices have the form of an elastic wristband with a VELCRO strap. When worn, the motors are positioned evenly around the arm, at 90 degrees from each other. All the motors are driven in the same way, and the magnitude of the commanded vibrations grows with the norm of $\boldsymbol{\tau}_c$,

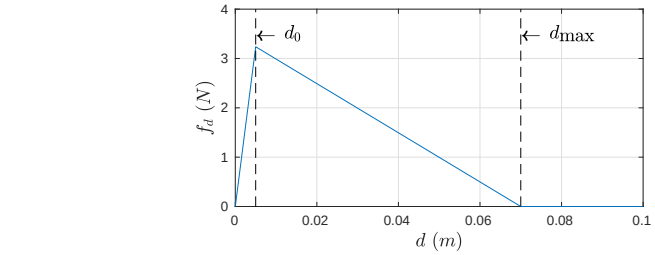
$$\tau_v = \nu_v ||\boldsymbol{\tau}_c||, \tag{4.23}$$

where ν_v is a positive gain chosen to fit the vibration range of the bracelet.

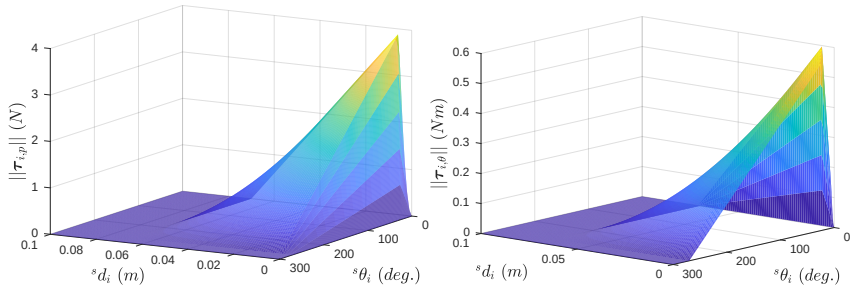
4.2.3.3 Haptic guidance toward suitable grasping poses

In addition to providing information about the proximity to dangerous configurations of the robot, we also provide haptic guidance toward suitable grasping poses. Whenever the robotic gripper comes close to an object, the data generated by the grasping algorithm is used to assist the operator in approaching a suitable grasp candidate. Of course, there are several ways to implement such a behavior. One of the most popular approaches consists in implementing a virtual spring between the current pose of the robot and the closest target one [205, 172]. In this case, the operator receives a force which increases as the gripper moves away from the target pose and decreases as it approaches it. However, it is not trivial to manage the switching between two neighboring target poses. As the user moves from one target pose to another, the considered virtual spring suddenly changes, resulting in abrupt and possibly dangerous changes in the direction and magnitude of the force feedback.

To avoid these issues, here we adopt a different approach, designed to always guarantee a continuous and smooth behavior as the user switches between grasp candidates. To this end, we consider a force profile that *increases* as the user gets closer to the target pose. This approach, akin to a "magnetic" behavior, ensures that the direction of the guiding force always points toward the closest suitable pose. Moreover, it also enables to consider all the target grasping poses at once. Each of them will contribute to the force feedback according to their distance from the current pose of the gripper: near poses will exert a stronger influence in the force feedback, while far poses will exert a feebler influence. As we detail below and as it is shown in Fig. 4.9, this combined approach ensures the continuity and smoothness of the received haptic feedback.



(a) $||\boldsymbol{\tau}_{i,p}||$ vs. ${}^s d_i$



(b) $||\boldsymbol{\tau}_{i,p}||$ vs. ${}^s d_i$, ${}^s \theta_i$ (c) $||\boldsymbol{\tau}_{i,\theta}||$ vs. ${}^s d_i$, ${}^s \theta_i$

Figure 4.9: Linear and angular force profiles for one grasping pose i. This approach considers a force profile that increases as the user gets closer to the target pose, akin to a "magnetic" behavior. Thresholds $d_{\max}$ and $\theta_{\max}$ indicates the distance after which the haptic guidance is activated; thresholds $d_0 > 0$ and $\theta_0 > 0$ prevents any abrupt change in the direction of the force when the user is very close to the target pose.

Guidance for one grasping pose The proposed haptic guidance consists of a 6-DoF force feedback, divided into linear and angular parts. Each grasping pose contributes to the overall resulting force according to its linear and angular distance from the current pose of the gripper. For the linear force contributions, let

$$\Delta_{i,p} = \frac{\boldsymbol{p}_i - \boldsymbol{p}_s}{||\boldsymbol{p}_i - \boldsymbol{p}_s||}$$

denote the normalized translation from the current gripper position $\boldsymbol{p}_s$ to a given grasping position $\boldsymbol{p}_i$ in the world frame. The contribution of this grasping position to the linear part of the guiding force should be directed along $\Delta_{i,p}$, with a positive magnitude that monotonically approaches zero as we increase the distance from the grasping position. We choose a linearly decreasing scaling factor

$$k_{\mathrm{p}}({}^s d_i) = 1 - \frac{1}{d_{\max}} {}^s d_i$$

that equals zero when the euclidean distance ${}^s d_i = ||\boldsymbol{p}_i - \boldsymbol{p}_s||$ between the gripper position and the grasp position equals a threshold $d_{\max}$. When the euclidean distance ${}^s d_i$ is larger than $d_{\max}$, the respective grasp position is ignored by setting its force contribution to zero (see Fig. 4.9a). On the other hand, in the close vicinity of

a target grasping position, we linearly decrease the force contribution back to zero to avoid any abrupt changes in the direction of the force. This is achieved by using the scaling factor

$$k_{\mathrm{p},0}({}^{s}d_i) = \left(\frac{1}{d_0} - \frac{1}{d_{\max}}\right) {}^{s}d_i$$

instead of $k_{\mathrm{p}}({}^{s}d_i)$, where d_0 is a small positive distance threshold at which the force contribution has its maximum value. The linear force contribution $\boldsymbol{\tau}_{i,p}$ of grasping position i is thus a piecewise continuous linear function given by

$$\boldsymbol{\tau}_{i,p} = \rho_p \begin{cases} k_{\mathrm{p},0}({}^{s}d_i)\ \Delta_{i,p} & \text{if } {}^{s}d_i < d_0 \\ k_{\mathrm{p}}({}^{s}d_i)\ \Delta_{i,p} & \text{if } d_0 < {}^{s}d_i < d_{\max}\ , \\ \mathbf{0} & \text{if } {}^{s}d_i > d_{\max} \end{cases} \tag{4.24}$$

where the constant factor ρ_p controls the maximum force. Fig. 4.9a shows the profile of $||\boldsymbol{\tau}_{i,p}||$ vs. ${}^{s}d_i$, with threshold values $d_0 = 0.005$ m and $d_{\max} = 0.07$ m.

For the angular torque contributions, let ${}^{s}\theta_i$ and $\Delta_{i,r}$ be the angular and axial parts of the angle-axis representation $({}^{s}\theta_i, \Delta_{i,r})$ of ${}^{s}\boldsymbol{R}_i = \boldsymbol{R}_s^T \boldsymbol{R}_i$, respectively. Analogously to the linear case, we define scaling factors to regulate the torque contribution as a function of the angular distance between each grasping pose and the gripper, ${}^{s}\theta_i$. The scaling factors are given by

$$k_r({}^{s}\theta_i) = 1 - \frac{1}{\theta_{\max}} {}^{s}\theta_i$$

and

$$k_{r,0}({}^{s}\theta_i) = \left(\frac{1}{\theta_0} - \frac{1}{\theta_{\max}}\right) {}^{s}\theta_i,$$

where $\theta_{\max}$ is the farthest angular distance after which the haptic guidance is activated and θ_0 is the threshold angular distance where the maximum torque is attained. The angular torque contribution of grasping position i is thus defined as

$$\boldsymbol{\tau}_{i,\theta} = \rho_\theta \begin{cases} k_{r,0}({}^{s}\theta_i)\ \Delta_{i,r} & \text{if } {}^{s}\theta_i < \theta_0 \\ k_r({}^{s}\theta_i)\ \Delta_{i,r} & \text{if } \theta_0 < {}^{s}\theta_i < \theta_{\max}\ , \\ \mathbf{0} & \text{if } {}^{s}\theta_i > \theta_{\max} \end{cases} \tag{4.25}$$

where ρ_θ controls the maximum torque. As above, having $\theta_0 > 0$ prevents any abrupt change in the direction of the force when the gripper is on the target pose.

Of course, we cannot consider the linear and angular components as two separate and independent contributions to the haptic feedback. It is important to account for the current gripper orientation $({}^{s}\theta_i, \Delta_{i,r})$ in the generation of the linear force cues $\boldsymbol{\tau}_{i,p}$, and vice-versa. The roto-translational distance $({}^{s}\theta_i, {}^{s}d_i)$ between the gripper

and a grasp candidate is to be considered altogether. To this end, accounting for (4.25) in (4.24), we get

$$\boldsymbol{\tau}_{i,p} = \rho_p \begin{cases} k_{\mathrm{p},0}({}^s d_i)\ \Delta_{i,p} & \text{if } {}^s d_i < d_0 \ \&\ {}^s\theta_i < \theta_0 \\ k_{\mathrm{p},0}({}^s d_i) k_{\mathrm{r}}({}^s\theta_i)\ \Delta_{i,p} & \text{if } {}^s d_i < d_0 \\ & \quad \&\ \theta_0 < {}^s\theta_i < \theta_{\max} \\ k_{\mathrm{p}}({}^s d_i)\ \Delta_{i,p} & \text{if } d_0 < {}^s d_i < d_{\max} \\ & \quad \&\ {}^s\theta_i < \theta_0 \\ k_{\mathrm{p}}({}^s d_i) k_{\mathrm{r}}({}^s\theta_i)\ \Delta_{i,p} & \text{if } d_0 < {}^s d_i < d_{\max} \\ & \quad \&\ \theta_0 < {}^s\theta_i < \theta_{\max} \\ \mathbf{0} & \text{if } {}^s d_i > d_{\max} \ ||\ {}^s\theta_i > \theta_{\max} \end{cases} \tag{4.26}$$

A similar approach can be used to generate the guiding torques $\boldsymbol{\tau}_{i,\theta}$ accounting for (4.24) in (4.25)

$$\boldsymbol{\tau}_{i,\theta} = \rho_\theta \begin{cases} k_{\mathrm{r},0}({}^s\theta_i)\ \Delta_{i,r} & \text{if } {}^s\theta_i < \theta_0 \ \&\ {}^s d_i < d_0 \\ k_{\mathrm{r},0}({}^s\theta_i) k_{\mathrm{p}}({}^s d_i)\ \Delta_{i,r} & \text{if } {}^s\theta_i < \theta_0 \\ & \quad \&\ d_0 < {}^s d_i < d_{\max} \\ k_{\mathrm{r}}({}^s\theta_i)\ \Delta_{i,r} & \text{if } \theta_0 < {}^s\theta_i < \theta_{\max} \\ & \quad \&\ {}^s d_i < d_0 \\ k_{\mathrm{r}}({}^s\theta_i) k_{\mathrm{p}}({}^s d_i)\ \Delta_{i,r} & \text{if } \theta_0 < {}^s\theta_i < \theta_{\max} \\ & \quad \&\ d_0 < {}^s d_i < d_{\max} \\ \mathbf{0} & \text{if } \theta_i > {}^s\theta_{\max} \ ||\ {}^s d_i > d_{\max} \end{cases} \tag{4.27}$$

Fig. 4.9b and Fig. 4.9c show the behavior of $\boldsymbol{\tau}_{i,p}$ and $\boldsymbol{\tau}_{i,\theta}$ as a function of ${}^s d_i$ and ${}^s\theta_i$, with threshold values $d_0 = 0.005$ m, $d_{\max} = 0.07$ m, $\theta_0 = 5$ deg and $\theta_{\max} = 270$ deg. If the gripper is far away from any target pose, i.e., ${}^s d_i > d_{\max}$ or ${}^s\theta_i > \theta_{\max}$, the operator does not receive any force feedback. Then, as the gripper is driven closer to a grasp candidate, both translation and orientation feedbacks increase. Finally, to avoid any abrupt change in the direction of the force, the feedback goes back to zero when the gripper is at the target pose. The choice of threshold values d_0, $d_{\max}$, θ_0 and $\theta_{\max}$ is challenging, system-dependent, and rather delicate, as it must ensure a smooth and safe force feedback. For example, choosing $d_0 = \theta_0 = 0$ would result in keeping the magnetic effect active until the gripper is exactly on the target pose (see Fig. 4.9a). This behavior may lead to abrupt changes in the direction of the force when ${}^s d_i$ is close to zero (and, therefore, $\boldsymbol{p}_i - \boldsymbol{p}_s$ may change direction very fast). To choose the right parameters for our system and target application, we asked 2 expert operators to repeatedly carry out a pick-and-place task, changing the abovementioned parameters at runtime to make the teleoperation as intuitive,

safe, and comfortable as possible. Finally, we asked them to find a consensus on the parameters' values and we used those in our implementation (see Sec. 4.2.4 for details).

Guidance for multiple grasping poses Equations (4.26) and (4.27) describe the linear and angular components of our guiding feedback for a generic grasping pose $\boldsymbol{x}_i$, respectively. However, as mentioned before, an interesting feature of our approach is that we can consider all the grasping candidates at once. A straightforward way to calculate the total haptic guidance feedback is

$$\boldsymbol{\tau}_g = [\boldsymbol{\tau}_{g,\boldsymbol{p}}^T, \boldsymbol{\tau}_{g,\theta}^T]^T = \sum_i \boldsymbol{\tau}_i, \tag{4.28}$$

where $\boldsymbol{\tau}_i = [\boldsymbol{\tau}_{i,\boldsymbol{p}}^T, \boldsymbol{\tau}_{i,\theta}^T]^T$ represents the force and torque cues associated with grasp candidate $\boldsymbol{x}_i$, as defined in (4.26), (4.27).

As an example, Fig. 4.10 shows the behavior of the linear part of the guidance, as defined in (4.28), when the gripper moves between two grasp candidates. For simplicity, we assume that the gripper $\boldsymbol{x}_s$ and the grasp candidates $\boldsymbol{x}_1, \boldsymbol{x}_2$ are all placed along the x axis, with $\boldsymbol{p}_1 = [0, 0, 0]^T$ m, $\boldsymbol{p}_2 = [0.03, 0, 0]^T$ m, and $\boldsymbol{p}_s$ moving between $[-0.1, 0, 0]^T$ m and $[0.1, 0, 0]^T$ m (see Fig. 4.10a). As the gripper moves from $[-0.1, 0, 0]^T$ m to $[0.1, 0, 0]^T$ m, the linear haptic cues $\boldsymbol{\tau}_{g,\boldsymbol{p}}$ guide the user first toward $\boldsymbol{x}_1$ and then toward $\boldsymbol{x}_2$, as expected. However, since (4.28) sums up all the poses contributions, the haptic guidance around $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ will not go to zero *exactly* at the grasping poses (see red dots in Fig. 4.10b). In fact, in both cases, the haptic guidance is slightly shifted toward the other grasp candidate, as the attraction force $\boldsymbol{\tau}_{2,p}$ toward grasp candidate $\boldsymbol{x}_2$ is active also in the vicinity of $\boldsymbol{x}_1$, and viceversa. This behavior happens only when grasping poses are closer than d_0 (see Fig. 4.10b). To avoid this undesired behavior, whenever the gripper gets *very* close to a target pose, i.e., $|\boldsymbol{x}_d - \boldsymbol{x}_s| < d_0 + \mu\theta_0$, we can progressively fade out the contributions of the other grasp candidates,

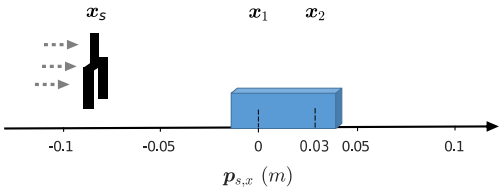
$$\boldsymbol{\tau}_g = \begin{cases} \boldsymbol{\tau}_d + \displaystyle\sum_{i \neq d} \frac{|\boldsymbol{x}_d - \boldsymbol{x}_s|}{d_0 + \mu\theta_0} \boldsymbol{\tau}_i & \text{if } |\boldsymbol{x}_d - \boldsymbol{x}_s| < d_0 + \mu\theta_0 \\ \displaystyle\sum_i \boldsymbol{\tau}_i & \text{otherwise} \end{cases}, \tag{4.29}$$

where $\boldsymbol{\tau}_d$ is the haptic guidance due to the closest grasping pose $\boldsymbol{x}_d$,

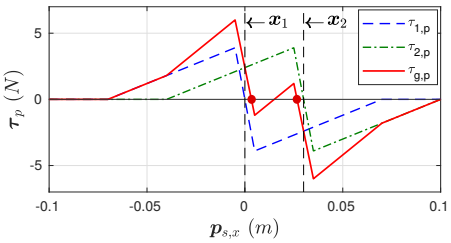
$$|\boldsymbol{x}_d - \boldsymbol{x}_s| = \min_i(|\boldsymbol{x}_i - \boldsymbol{x}_s|), \tag{4.30}$$

where $|\boldsymbol{x}_i - \boldsymbol{x}_s|$ denotes the roto-translational distance between $\boldsymbol{x}_i$ and the gripper pose $\boldsymbol{x}_s$,

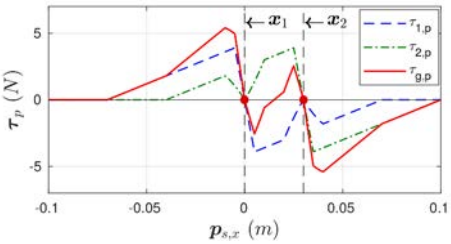
$$|\boldsymbol{x}_i - \boldsymbol{x}_s| = {}^s d_i + \mu {}^s\theta_i, \tag{4.31}$$



(a) The gripper x_s moves first toward x_1 and then toward x_2.



(b) Linear forces when summing up all the contributions (see eq. (4.28)).



(c) Linear forces when fading out farther contributions (see eq. (4.29)).

Figure 4.10: The figures show the linear haptic cues guiding the user toward grasping poses x_1 and x_2 as a function of the position of the gripper in a mono-directional case. They highlight the difference between a simple summation of the forces of attraction to all grasping poses (Fig. (b)) and the adopted solution, which centers the attraction force on the closest grasp candidate (Fig. (c)).

where $\mu > 0$ is used to properly scale the angular component of the distance with respect to the linear one.

Fig. 4.10c shows the refined behavior of the linear part of the guidance, as defined in (4.29). Now, as the gripper moves to a grasp candidate, haptic cues are exactly centered on that pose.

4.2.4 Experimental evaluation

To evaluate the effectiveness and viability of our shared-control approach, we carried out two pick and place experiments, enrolling fifteen human subjects. In the first experiment, we consider only one object to grasp, comparing the proposed shared-control approach vs. standard teleoperation and a state-of-the-art shared-control technique. In the second experiment, we extend the evaluation to picking up multiple objects. These tasks have been chosen following a discussion within the RoMaNS project, which considered them as good representatives of sort and segregation of nuclear waste.

4.2.4.1 Experimental setup and task

The experimental setup is shown in Fig. 4.6, and it is described at the beginning of Sec. 4.2.1. The master side consists of a Haption Virtuose 6-DoF haptic grounded interface. The slave side consists of an Adept Viper s850 6-DoF serial manipulator equipped with a Sake EzGripper gripper and an Intel Realsense SR-300 RGB-D sensor. Depending on the experiment, the remote environment is composed of either one or four objects placed on a table in front of the manipulator: one cardboard container (filled with foam) (14×4×4 cm, 80 g, left object in Fig. 4.6), one wooden cube (4×4×4 cm, 20 g, not visible in Fig. 4.6), and two L-shaped wood pieces (one composed of two 14×1.2×6.5 cm rectangles, 125 g, center object in Fig. 4.6; and one made of 10×2.6×4 cm and 11×2.6×9 cm rectangles, 280 g, right object in Fig. 4.6). A bin for dropping the grasped items is placed on the right hand side of the manipulator. To enable the operator to see the environment, the master interface is placed two meters in front of the slave robot. Since the workspace of the master interface is smaller than that of the slave robot, we used a button on the master interface as a clutch. Whenever the button is pressed, the motions of the master and slave systems are decoupled. This approach allows the user to start the motion at an arbitrary position, then pause, move to a more comfortable or suitable position, and then resume control of the robot [206]. Of course, clutching can be avoided by acting on the scaling of the robot motion with respect to the master interface. However, in the following experiments, no motion scaling is used.

We asked the human subjects to use the master interface to control the motion of the slave manipulator. The task consisted in grasping the object(s) placed in front of the robot and placing them into the bin.

4.2.4.2 Representative repetition of the sorting task

Before starting with our human subject experiments, we carried out a representative repetition of the sorting task, employing all the four objects. Fig. 4.11 shows the results of this preliminary run. Fig. 4.11a shows the roto-translational distance $|\boldsymbol{x}_m - \boldsymbol{x}_s|$ between the current gripper pose $\boldsymbol{x}_s$ and the closest grasp candidate $\boldsymbol{x}_m$ vs. time. Fig. 4.11b and Fig. 4.11c show the linear and angular guiding cues provided to the operator vs. time, respectively. As expected, the haptic feedback increases as the gripper approaches a potential grasp candidate (i.e., when $d_0 < {}^s d_i < d_{\max}$). Moreover, to avoid any abrupt change in the direction of the force, whenever the gripper is *very* close to the target pose (i.e., ${}^s d_i < d_0$), the force slowly diminishes (see Sec. 4.2.3.3 and Fig. 4.9). This behavior leads to a convex U shape for both the linear and angular force graphs as $|\boldsymbol{x}_m - \boldsymbol{x}_s|$ approaches zero. An exception to this 'U-shaped' behavior is at $t = 41$ s, where the linear cues $\boldsymbol{\tau_p}$ look different.

In fact, in this case, the user was not entirely satisfied by the grasp pose proposed by the architecture, and therefore he adjusted it to one which he redeemed more convenient. This behavior is desirable, as we want to guide the human user, but also leave her/him free to make the final decision. It is also interesting to notice that the operator received linear haptic feedback at $t = [73, 77]$ s and $t = [90, 95]$ s (denoted in red rectangles), although the gripper was not in the proximity of any grasp candidate (see Figs. 4.11a and 4.11b). This behavior can be explained by the sudden increase of the cost function $\mathcal{H}$, shown in Fig. 4.11d, meaning that the robot approached one of the system constraints (e.g., joint limits). The user is able to distinguish the nature of the haptic feedback (guiding toward a grasping pose vs. proximity to unsafe system configurations) thanks to the additional vibrotactile stimuli we provide when the cues are due to the proximity to system constraints.

4.2.4.3 Experiment #1: pick and place of one object

We consider the robotic system described in Sec. 4.2.4.1 and shown in Fig. 4.6. For this experiment, the remote environment is composed of only one object, i.e., the wooden piece shown on the right hand side of Fig. 4.6. Participants were required to control the motion of the robotic manipulator and gripper to grasp the object, lift it from the ground, and place it in the bin. The task started when the manipulator moved for the very first time and it was considered successfully completed when the object was released in the bin.

We consider three different ways of commanding the motion of the robot through the haptic interface:

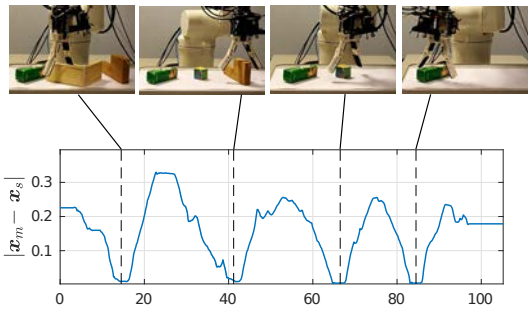
Condition T: classic teleoperation, where the subject controls all the 6 DoF of the manipulator and receives no haptic guidance about suitable grasping poses. The master/slave coupling is thus the same as that described in Sec. 4.2.3.1 while the term $\boldsymbol{\tau}_g$ is removed from eq. (4.20), which becomes

$$\boldsymbol{\tau} = \boldsymbol{\tau}_c - \boldsymbol{B}\boldsymbol{v}_m, \tag{4.32}$$

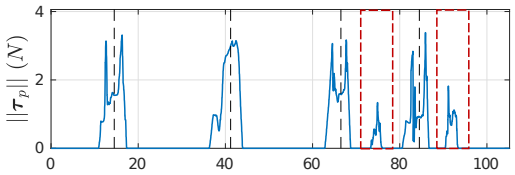
where $\boldsymbol{\tau}_c$ accounts for joint limits and singularities, and $\boldsymbol{B}$ is a damping matrix (see Sec. 4.2.3).

Condition SF: The shared-control approach described in Sec. 3.2 where autonomy ensures the orientation of the gripper towards the object to be grasped while the operator commands the remaining 4 DoF.

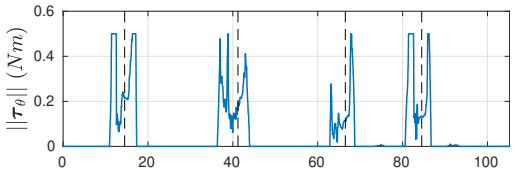
Condition SH: our proposed haptic shared control approach, where the subject controls all the 6 DoF of the manipulator and receives haptic guidance



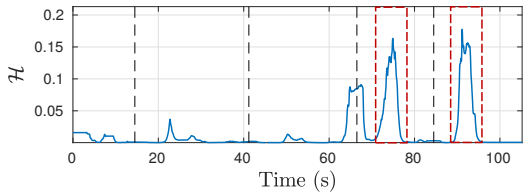
(a) Roto-translational distance between the gripper pose x_m and the closest grasp candidate pose x_s vs. time.



(b) Linear force feedback vs. time.

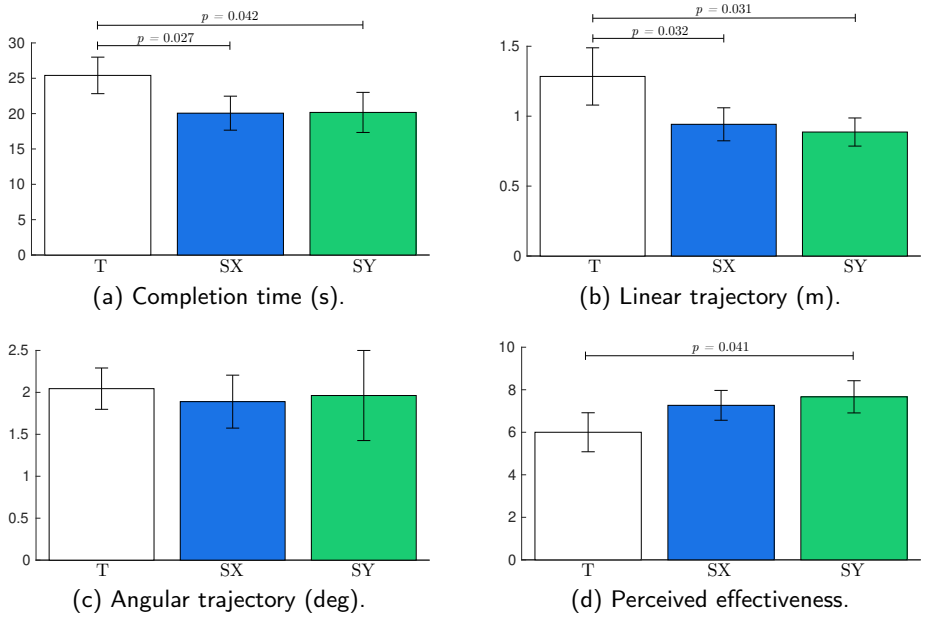


(c) Angular torque feedback vs. time.



(d) Cost function $\mathcal{H}$ vs. time.

Figure 4.11: Sample experiment for picking and placing several objects in a cluttered scene. (a) Roto-translational distance between the gripper and the closest grasp candidate, (b) linear force received by the operator, (c) angular torques received by the operator, and (d) the evolution of the cost function describing the constraints.



(a) Completion time (s). (b) Linear trajectory (m).

(c) Angular trajectory (deg). (d) Perceived effectiveness.

Figure 4.12: Experiment #1. Mean and 95% confidence interval of (a) completion time, (b) linear trajectory length, (c) angular motion, and (d) perceived effectiveness of the three feedback conditions are plotted.

about suitable grasping poses (kinesthetic feedback) and proximity to possibly unsafe configurations of the system (kinesthetic and vibrotactile feedback);

Each subject carried out six randomized repetitions of the grasping task, two for each experimental condition. A video showing trials in all experimental conditions is available as supplemental material and at https://youtu.be/Bb4M3UjwAGY.

Participants Fifteen right-handed subjects (average age 26.4) participated in the study. Five of them had previous experience with haptic interfaces. None of the participants reported any deficiencies in their visual or haptic perception abilities. The experimenter explained the procedures and spent about two minutes adjusting the setup to be comfortable before the subject began the experiment. Each subject then spent about three minutes practicing the control of the telemanipulation system before starting the experiment.

Results To evaluate the effectiveness of our system in grasping the considered object, the usefulness of the proposed shared-control approach, and the effectiveness of haptic stimuli in such a task, we recorded (*i*) the completion time, (*ii*) the linear trajectory followed by the robotic end-effector, and (*iii*) the angular motion of the robotic end-effector. Moreover, immediately after the experiment, subjects were

also asked to report the effectiveness of each feedback condition in completing the given task using bipolar Likert-type eleven-point scales. To compare the different metrics, we ran one-way repeated-measures ANOVA tests on the data. The control modality (standard teleoperation vs. constrained shared control vs. our haptic-enabled sharedcontrol, T vs. SF vs. SH) was the within-subject factors. All data passed the Shapiro-Wilk normality test. Figure 4.12a shows the average task completion time. Data passed the Mauchly's Test of Sphericity. The one-way ANOVA test revealed a statistically significant change in the task completion time across the conditions ($F(2, 28) = 7.183$, $p = 0.003$, $a = 0.05$). Post hoc analysis with Bonferroni adjustments revealed a statistically significant difference between T vs. SF ($p = 0.027$) and T vs. SH ($p = 0.042$). The Bonferroni correction is used to reduce the chances of obtaining false-positive results when multiple pair-wise tests are performed on a single set of data. Figure 4.12b shows the linear motion covered by the robotic gripper during the task. Mauchly's Test of Sphericity indicated that the assumption of sphericity had been violated ($\chi^2(2) = 17.415$, $p < 0.001$, $a = 0.05$). The one-way repeated-measure ANOVA with a Greenhouse-Geisser correction revealed a statistically significant change in the linear motion across the conditions ($F(1.151, 16.110) = 8.319$, $p = 0.001$). Post hoc analysis with Bonferroni adjustments revealed a statistically significant difference between T vs. SF ($p = 0.032$) and T vs. SH ($p = 0.031$). Figure 4.12c shows the average angular motion covered by the robotic gripper during the task. Mauchly's Test of Sphericity indicated that the assumption of sphericity had been violated ($\chi^2(2) = 6.306, p = 0.043$). The one-way repeated-measure ANOVA with a Greenhouse-Geisser correction found no statistically significant change in the angular motion across the conditions. Fig. 4.12d shows the perceived effectiveness for the three experimental conditions. A Friedman test showed a statistically significant difference between the means of the four feedback conditions ($\chi^2(2) = 6.536$, $p = 0.038$, $a = 0.05$). The Friedman test is the non-parametric equivalent of the more popular repeated-measures ANOVA. The latter is not appropriate here since the dependent variable was measured at the ordinal level. Post hoc analysis with Bonferroni adjustments revealed a statistically significant difference between T vs. SH ($p = 0.041$). Finally, thirteen subjects out of fifteen found the two shared-control conditions to be the most effective at completing the grasping task: nine preferred condition SH while four preferred SF.

4.2.4.4 Experiment #2: pick and place in a cluttered scenario

Experimental setup and feedback conditions We consider here the same robotic system as in the first experiment of Sec. 4.2.4.3. For this experiment, the remote environment is composed of the four objects described in Sec. 4.2.4.1 and shown in Fig. 4.6. As before, participants were asked to control the motion of the

robotic manipulator and gripper to grasp the four objects, one by one, lift them from the ground, and place them in the bin. The task started when the manipulator moved for the very first time and it was considered successfully completed when the last object was released in the bin.

Since the shared-control approach in Sec. 3.2, referred to as SF in the previous experiment, can only consider one object at a time, it is not suitable for this second experiment. For this reason, here we consider only two ways of commanding the motion of the robot through the haptic interface (see Sec. 4.2.4.3):

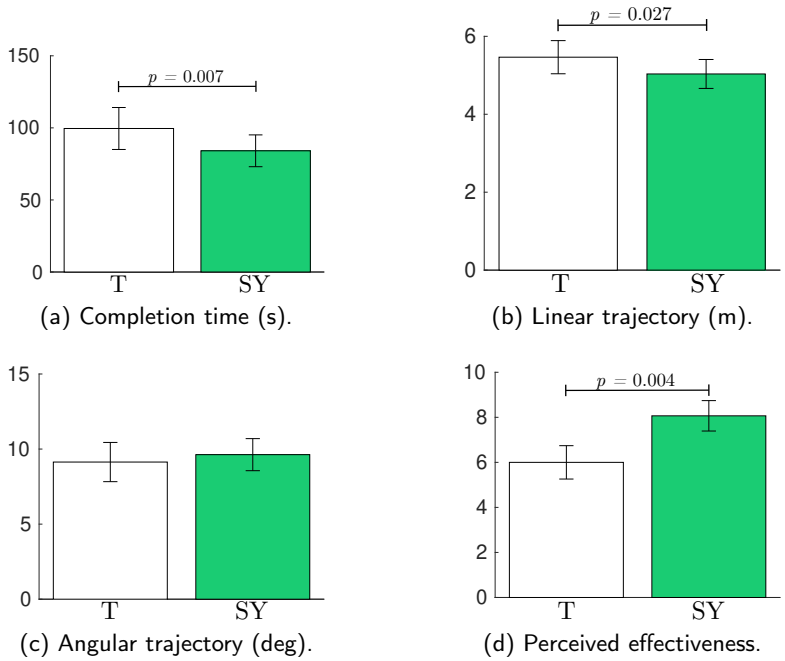
T: standard teleoperation, where the subject controls all the 6 DoF of the manipulator and receives no haptic guidance about suitable grasp poses.

SH: our proposed haptic shared control approach, where the subject controls all the 6 DoF of the manipulator and receives haptic guidance about suitable grasping poses and proximity to possibly unsafe configurations of the system.

Each subject carried out four randomized repetitions of the grasping task, two for each experimental condition. A video showing trials in all experimental conditions is available as supplemental material and at https://youtu.be/Bb4M3UjwAGY?t=1m49s.

Participants The same fifteen subjects who participated in the first experiment also participated in this one (see Sec. 4.2.4.3).

Results As in Sec. 4.2.4.3, we recorded (*i*) the completion time, (*ii*) the linear trajectory followed by the robotic end-effector, (*iii*) the angular motion of the robotic end-effector, and (*iv*) the perceived effectiveness of the different conditions. Since here we only consider two conditions (T vs. SH), we ran paired-samples t-test to compare the metrics. All data passed the Shapiro-Wilk normality test. Figure 4.13a shows the average task completion time. The paired-samples t-test revealed a statistically significant change in the task completion time across the conditions (T vs. SH, $t(14) = 3.176$, $p = 0.007$, $a = 0.05$). Figure 4.13b shows the linear motion covered by the robotic gripper during the task. The paired-samples t-test revealed a statistically significant change in the linear motion across the conditions (T vs. SH, $t(14) = 2.464$, $p = 0.027$, $a = 0.05$). Figure 4.13c shows the angular motion covered by the robotic gripper during the task. The paired-samples t-test did not reveal a statistically significant change in the linear motion across the conditions. As before, immediately after the experiment, subjects were also asked to report the effectiveness of each feedback condition in completing the given task using bipolar Likert-type eleven-point scales. Fig. 4.13d shows the perceived effectiveness for the



(a) Completion time (s). (b) Linear trajectory (m). (c) Angular trajectory (deg). (d) Perceived effectiveness.

Figure 4.13: Experiment #2. Mean and 95% confidence interval of (a) completion time, (b) linear trajectory length, (c) angular motion, and (d) perceived effectiveness of the three feedback conditions are plotted.

three experimental conditions. A related-samples Wilcoxon signed rank test showed a statistically significant difference between the means of the feedback conditions ($Z = 2.887$, $p = 0.004$, $a = 0.05$). Finally, thirteen subjects out of fifteen found condition SH to be the most effective at completing the grasping task.

4.2.5 Discussion

To demonstrate the feasibility and effectiveness of the shared-control approach presented in this section, we conducted two remote telemanipulation experiments enrolling fifteen human subjects. The first experiment is described in Sec. 4.2.4.3. The task consisted in picking up one object and placing it inside a bin. We tested the performance of the proposed shared-control system (SH) against a more classic teleoperation approach (T), in which the user is able to freely control all the DoF of the manipulator, and a standard constrained shared-control approach (SF), where the control of the robot's DoF are strictly divided between the human and an autonomous algorithm. As a measure of performance, we considered the average completion time, trajectory length, angular motion, and perceived effectiveness. Results showed that, in all the considered metrics but one (angular trajectory), our proposed shared-control approach significantly outperformed the more classic teleoperation architecture. Moreover, all subjects preferred one of the two shared-

control architectures with respect to teleoperation. This proves our hypothesis that shared-control can be a viable and effective approach to improve currently-available teleoperation systems in remote manipulation tasks. However, in this first experiment, we did not find any significant difference between the two shared-control approaches (SH vs. SF). This result means that, if we are dealing with only one object, the described approach may not improve the task performance with respect to the approach proposed in chapter 3. However, in SF, multiple objects in a cluttered environment can be considered at once, something which is not possible in SH, where a particular object must be chosen beforehand. This gives the operator the flexibility to judge, on the fly, the sequence of objects to be picked.

For this reason, we carried out a second experiment, considering the same experimental setup and task as before but with four different objects to move (see Sec. 4.2.4.4). We tested the performance of our shared-control system (SH) only against classic teleoperation (T). As a measure of performance, we considered again the average completion time, trajectory length, angular motion, and perceived effectiveness. In all the considered metrics but angular motion, SH outperformed T. Moreover, all subjects but two found our shared-control approach more effective than classic teleoperation. This second experiment proves that our technique is indeed a viable, effective, and flexible approach to improve the performance of teleoperation systems in cluttered environments.

It is important to also notice that none of our subjects was experienced in using the experimental setup. In fact, the recorded significant difference between shared control vs. teleoperation might change in the presence of experienced users. This is something worth studying in the coming future, since all the operators in the target scenario (at the Sellafield nuclear site) are skilled and experienced.

Clutching is another interesting point to discuss. Given a grounded haptic interface and a grounded slave manipulator, it is always possible to define an appropriate scaling factor between master and slave velocities such that the operator does not require clutching. However, as the difference between the master and slave workspaces increases, this mapping requires higher and higher gains, resulting in a telemanipulation system which is very hard to control, since the operator's accuracy/resolution in positioning the slave arm is degraded. Our collaborative project presents us with the perfect example: the custom rig at the National Nuclear Laboratory (UK) is composed of (i) the same grounded haptic interface we are using in this paper, and (ii) a 500-kg-payload Kuka KR 500 manipulator. Although it is theoretically possible to map the workspace of the Virtuose 6D to the (much) larger workspace of the KUKA robot, this would result in very high motion gains (i.e., a small movement of the master interface would cause a big movement of the slave robot). For this reason, we decided to use the clutch in the described implementation. In this respect,

it is interesting to analyse how to best tune the master-slave motion scaling factor, with the final objective of finding a good trade-off between high precision of movement and low need of clutching. Finally, all subjects appreciated the presence of the mixed kinesthetic and vibrotactile haptic feedback to provide information about the manipulator's joint limits. This approach enabled them to always complete the task successfully, pushing them away from dangerous robot configurations in a very intuitive and non-obtrusive way. Subjects described the feeling due to the haptic feedback "as if the system was trying to nudge them toward a safer configuration of the robot."

Figure 4.14: The experimental setup showing the slave robotic arm on the left and the master haptic arm on the right.

4.3 Learning-Based Haptic Design in Shared Control Systems

In the approach described in Sec. 4.2, the human operator and autonomy act together in the same space. However, finding the proper balance between the level of human intervention and that of robot autonomy remains, in most cases, an open (and task-dependent) problem. This balance is indeed often tuned heuristically and, moreover, is not adapted during the task execution, a limitation that can lead to an increase of the operator's workload and a decrease of her/his efficiency. On the other hand, a shared control architecture could greatly benefit from the ability to adapt *online* the operator/autonomy balance as a function of the 'confidence' of the robot autonomy in realizing the shared task. This would (*i*) allow the operator to intervene only when strictly needed (when the robot autonomy is more likely to fail in fulfilling the task), and (*ii*) provide the operator with an informative feedback for guiding her/his actions only when the autonomous component is expected to perform well. However, to the best of our knowledge, no established way has yet been proposed for taking a step in this direction.

On the other hand, learning from demonstration (LfD) or programming by demonstration (PbD) have been proposed as an intuitive way to program robotic motions [207, 208]. In LfD methods, the distribution of the demonstrated trajectories is often modeled, and the learned distribution is leveraged for generalizing the demonstrated behaviors [209, 210]. Recent work on LfD showed that the variance of

the demonstrated trajectories can be used to adaptively control the robot behaviors [211]. Leveraging the distribution of the demonstrated trajectories into the design of shared control frameworks seems, therefore, a meaningful/promising approach for obtaining a shared control framework able to adjust *online* the balance between the robot autonomy and the human preference.

To this end, we hereby consider that the distribution of the demonstrated trajectories indicates the preference of an expert operator on the robot motion: a high variance in the demonstrated trajectories is assumed to indicate a weak preference of the operator, whereas a low variance is taken as a strong preference. Based on this assumption, we use the variance of the trajectory to control the balance between the controllers autonomy and the human intervention. In particular, when the variance of the demonstrated trajectories is low, the user is fed back with strong force cues meant to minimize any deviation from the 'nominal' (learned) trajectory. On the contrary, as the variance increases, the force cues are suitably attenuated, thus providing the operator with the possibility (and feeling) of freely moving the robotic system along any direction of interest. Through interactive task execution using our shared control framework, we can obtain additional trajectories that are executed under the supervision of the human operator. By aggregating the newly obtained data, we refine the learned trajectory distribution. Therefore, the performance of our shared control system improves through interactive task executions.

The proposed approach is applied on the pre-grasp approach phase providing an efficient LfD-based shared control architecture for trajectory following. The coming sections are organized as follows. In Sec. 4.3.1, an overview on related LfD work is presented. Sec 4.3.2, the general teleoperation framework is briefly introduced. Sec 4.3.2.1 details the algorithm for the generation of conditional trajectory distributions from demonstrations, while the details of the shared control architecture are presented in Sect. 4.3.3. Trajectory refinement from executions is described in Section 4.3.4 after which section 4.3.5 reports the experimental results.

4.3.1 Related Work

In LfD methods, the distribution of the demonstrated trajectory is often modeled with statistical methods. The framework called ProMP models the distribution of the trajectories in the parameter space [209] while the method in [210] modeled the distribution of the demonstrated trajectories at each time step using Gaussian Processes. These studies showed that the demonstrated behaviors can be generalized to new situation by modeling the distribution of the demonstrated trajectories. The learned distribution of demonstrations can then be used for designing control schemes. For instance, in [211] the robot learns collaborative skills and adapts its

impedance behavior from demonstrations. However, to the best of our knowledge, no prior work has been proposed for leveraging the distribution of the demonstrated trajectories for controlling the balance between the controllers autonomy and the human inputs.

Moreover, although we assume that the demonstrations from human experts are available, it is often expensive to obtain enough demonstrations for building an initial dataset rich enough for capturing all the possible conditions. However, in a shared control framework the trajectory obtained through the actual execution of the task can be also re-used to update the model of the trajectory distribution. Calinon et al. described the incremental learning of the trajectory distribution in [212]. The method presented in [212] uses kinesthetic teaching for modifying the trajectory, which can be cumbersome if a manipulator has many degrees of freedom. In contrast, we will use our shared control architecture so that human operator can intuitively modify the trajectory. Regarding online learning, Ross et al. proposed a data aggregation approach and clarified the no-regret property of their approach [213]. This approach can be used in various applications of imitation learning where additional data of experts' demonstration is available. Chernova et al. proposed to refine the policy by actively requesting additional demonstrations [214]. In this method, the confidence of the autonomous agent is computed and if the agent's confidence is low, additional demonstrations are requested. By using this kind of incremental aggregation of the demonstrated data, the policy for generating actions can be improved.

4.3.2 Modeling

Figure 4.14 illustrates the experimental setup at hand where the same bilateral master-slave teleoperation system described in 4.2.3.1 is used. However, we hereby define $\boldsymbol{x}_m = (\boldsymbol{p}_m, \boldsymbol{\phi}_m) \in \mathbb{R}^6$ as the position and a minimal representation of the orientation of the master end-effector in its base frames. A minimal representation of the orientation is actually necessary for the statistical methods used in the learning algorithm. While these methods could be generalized to non-singular representations of the orientation, like rotation matrices or quaternions, the work at hand focuses on the use of existing learning and statistical algorithms in developing shared control architectures. We also introduce a context vector $\boldsymbol{r}$ which characterizes the task. For example, the context vector could contain the position of the object we want to grasp. We will use the context vector to adapt the desired trajectories of the master-slave system.

In the next section, we discuss how to obtain representative statistics from a set of demonstrated master trajectories $\boldsymbol{x}_m(t)$ and how this statistics can then be

exploited for designing the force cues $\boldsymbol{\tau}$ in (4.19).

4.3.2.1 Demonstrated Trajectory Distributions

During the learning phase, we assume that a skilled human operator demonstrates a number of feasible trajectories for letting the slave robot approaching a target object of interest. During this phase the slave is commanded to follow the master position $\boldsymbol{x}_m(t)$ but no force feedback is provided to the user ($\boldsymbol{\tau} = \mathbf{0}$). We then model a distribution of these demonstration trajectories by using independent Gaussian distributions for each time step t, i.e.,

$$p(\boldsymbol{x}_m(t)) \sim \mathcal{N}(\bar{\boldsymbol{x}}_m(t), \boldsymbol{\Sigma}(t)) \tag{4.33}$$

where $\bar{\boldsymbol{x}}_m(t)$ and $\boldsymbol{\Sigma}(t)$ are the mean and variance of $\boldsymbol{x}_m(t)$. Assuming that the trajectory is given as a sequence of states of the system, i.e., $\boldsymbol{\gamma} = [\boldsymbol{x}_m(t_0), \ldots, \boldsymbol{x}_m(t_f)]$, the distribution of the trajectory $\boldsymbol{\gamma}$ can be modeled as

$$p(\boldsymbol{\gamma}) = \prod_{t=t_0}^{t=t_f} \mathcal{N}(\bar{\boldsymbol{x}}_m(t), \boldsymbol{\Sigma}(t)). \tag{4.34}$$

We assume that the demonstrations are available under various contexts $\boldsymbol{r}^i$. In this case, we can model the conditional distribution of the demonstrated trajectories given the context in order to generalize the demonstrated trajectories to new situations [210, 215, 216]. Here, we use Locally Weighted Regression (LWR) to model this distribution [217, 218]. Although we use LWR in this work, our approach is not limited to specific regression methods. Other regression methods such as Gaussian Mixture Regression can be also used to model the distribution.

We assume a dataset $\mathcal{D} = \{\boldsymbol{\gamma}^i, \boldsymbol{r}^i\}_{i=1}^N$ of trajectories and context vectors is available. Given a new query context $\boldsymbol{r}^{\text{test}}$, the locality weight for the ith sample can be computed as

$$w^i = \exp\left(-\frac{(\boldsymbol{r}^i - \boldsymbol{r}^{\text{test}})^T(\boldsymbol{r}^i - \boldsymbol{r}^{\text{test}})}{h}\right). \tag{4.35}$$

where h is a constant that determines the bandwidth of the locality kernel. As in LWR, we can now compute a local linear model by using a weighted linear ridge regression, i.e.,

$$\begin{bmatrix} \boldsymbol{a}_t^T \\ \boldsymbol{A}_t^T \end{bmatrix} = (\boldsymbol{\Gamma}^T \boldsymbol{W} \boldsymbol{\Gamma} + k\boldsymbol{I})^{-1} \boldsymbol{\Gamma}^T \boldsymbol{W} \boldsymbol{X}_t,$$

where the matrix $\boldsymbol{\Gamma} = [\tilde{\boldsymbol{r}}^1, \cdots, \tilde{\boldsymbol{r}}^N]^T$ contains all training context vectors $\tilde{\boldsymbol{r}}^i = [1; \boldsymbol{r}^i]$ which have been extended by a bias term, the matrix $\boldsymbol{W} = \text{diag}([w^i])$ is a diagonal matrix containing the weightings w^i and the matrix $\boldsymbol{X}_t = [\boldsymbol{x}_m^1(t), \cdots, \boldsymbol{x}_m^N(t)]^T$ contains all state samples obtained for time step t.

The estimated mean state for time t is then given by

$$\boldsymbol{E}[\boldsymbol{x}_m(t)|\boldsymbol{r}^{\text{test}}] = \boldsymbol{A}_t\boldsymbol{r}^{\text{test}} + \boldsymbol{a}_t. \tag{4.36}$$

Similarly, we can compute the conditional covariance over $\boldsymbol{x}_m(t)$ given our query context $\boldsymbol{r}^{\text{test}}$ using a weighted maximum likelihood estimate, i.e.,

$$\boldsymbol{\Sigma}_{\boldsymbol{x}_m|\boldsymbol{r}^{\text{test}}}(t) = \frac{\sum_{i=1}^{N} w^i(\boldsymbol{x}_m^i(t) - \bar{\boldsymbol{x}}_m^i)(\boldsymbol{x}_m^i(t) - \bar{\boldsymbol{x}}_m^i)^T}{\sum_{i=1}^{N} w^i}, \tag{4.37}$$

where $\bar{\boldsymbol{x}}_m^i = \boldsymbol{A}_t\boldsymbol{r}^i + \boldsymbol{a}_t$ is the estimated mean for sample i.

4.3.3 Shared Control Architecture

In the application at hand, we consider the position of the target object $\boldsymbol{p}_o = (x_o, y_o, z_o)$ as the context $\boldsymbol{r}$ upon which the conditional distribution of the trajectory is generated in (4.36) and (4.37). To simplify the notation, we will refer to $\boldsymbol{E}[\boldsymbol{x}_m(t)|\boldsymbol{r}_{\text{test}}]$ by $\boldsymbol{x}_{m,d}(t) = (\boldsymbol{p}_{m,d}(t), \boldsymbol{\phi}_{m,d}(t))$ in the following section. Moreover, we will denote by $\boldsymbol{\Sigma}_p(t) \in \mathbb{R}^{3\times3}$ and $\boldsymbol{\Sigma}_\phi(t) \in \mathbb{R}^{3\times3}$ the 3×3 block diagonal elements of the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{x}_m|\boldsymbol{r}_{\text{test}}}(t) \in \mathbb{R}^{6\times6}$ computed in (4.37).

Having estimated the distribution from demonstrations, we now present the design of the force cues $\boldsymbol{\tau}$ in (4.19). Our approach is as follows: we treat the mean of the conditional distribution, $\boldsymbol{x}_{m,d}(t)$, as a *desired* trajectory for the master device. Indeed, this trajectory represents, in some sense, the 'best/nominal' approaching trajectory (to the considered target location) from the expert user's demonstrations collected during the learning phase. The force cues $\boldsymbol{\tau}$ will then attempt to steer the master device along the desired $\boldsymbol{x}_{m,d}(t)$ with a 'stiffness', roughly speaking, inversely proportional to the variance of the generated conditional distribution. This will effectively tune the degree of maneuverability of the master device around the nominal trajectory $\boldsymbol{x}_{m,d}(t)$, and, thus, provide the operator with an increased situational awareness (more freedom to deviate from $\boldsymbol{x}_{m,d}(t)$ for large variances, and the converse for small variances).

Let $\boldsymbol{\tau} = (\boldsymbol{\tau}_p, \boldsymbol{\tau}_\phi)$, where $\boldsymbol{\tau}_p \in \mathbb{R}^3$ and $\boldsymbol{\tau}_\phi \in \mathbb{R}^3$ are the forces/torques acting on the position/orientation $\boldsymbol{p}_m$, $\boldsymbol{\phi}_m$ of the master end-effector, and define $\boldsymbol{e}_p = \boldsymbol{p}_{m,d} - \boldsymbol{p}_m$ and $\boldsymbol{e}_\phi = \boldsymbol{\phi}_{m,d} - \boldsymbol{\phi}_m$ as the position/orientation errors. Following the classical literature on task-space impedance control [219], we then design

$$\begin{aligned}\begin{bmatrix} \boldsymbol{\tau}_p \\ \boldsymbol{T}^T(\boldsymbol{\phi})\boldsymbol{\tau}_\phi \end{bmatrix} = &\begin{bmatrix} \boldsymbol{M}_p & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_\phi \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{p}}_{m,d} \\ \ddot{\boldsymbol{\phi}}_{m,d} \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_p & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B}_\phi \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{e}}_p \\ \dot{\boldsymbol{e}}_\phi \end{bmatrix} \\ &+ \begin{bmatrix} \boldsymbol{K}_p & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_\phi \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_p \\ \boldsymbol{e}_\phi \end{bmatrix}\end{aligned} \tag{4.38}$$

where $\boldsymbol{M}_p(\boldsymbol{p})$ and $\boldsymbol{M}_\phi(\boldsymbol{\phi})$ are the 3×3 block diagonal elements of the master inertia matrix $\boldsymbol{M}_m(\boldsymbol{x}_m)$ in (4.19) associated to the coordinates $\boldsymbol{p}$ and $\boldsymbol{\phi}$, ($\boldsymbol{B}_p$, $\boldsymbol{B}_\phi$) and ($\boldsymbol{K}_p$, $\boldsymbol{K}_\phi$) are 3×3 damping and stiffness matrix terms, and $\boldsymbol{T}(\boldsymbol{\phi})$ is the transformation matrix defined mapping the variation of the orientation $\dot{\boldsymbol{\phi}}_m$ to the angular velocity $\boldsymbol{\omega}_m$ such that

$$\boldsymbol{\omega}_m = \boldsymbol{T}(\boldsymbol{\phi}_m)\dot{\boldsymbol{\phi}}_m. \tag{4.39}$$

As mentioned earlier, a decrease in the variance of the generated distribution is assumed to indicate more confidence in the generated trajectory. This confidence is to be reflected as an increase in the stiffness of the virtual spring and the force cues fed to the human operator and vice versa. To this end, we consider the eigenvalue decomposition of the (symmetric and positive definite) covariance matrices

$$\begin{cases} \boldsymbol{\Sigma}_p &= \mathbf{V}_p \mathbf{S}_p \mathbf{V}_p^T, \\ \boldsymbol{\Sigma}_\phi &= \mathbf{V}_\phi \mathbf{S}_\phi \mathbf{V}_\phi^T, \end{cases} \tag{4.40}$$

with $\mathbf{S}_p = \text{diag}(\sigma_{p_i})$ and $\mathbf{S}_\phi = \text{diag}(\sigma_{\phi_i})$. The desired stiffness matrices ($\boldsymbol{K}_p$, $\boldsymbol{K}_\phi$) are then defined as

$$\begin{cases} \boldsymbol{K}_p &= \mathbf{V}_p \boldsymbol{K}_{p,0} e^{-\alpha_p \mathbf{S}_p} \mathbf{V}_p^T \\ \boldsymbol{K}_\phi &= \mathbf{V}_\phi \boldsymbol{K}_{\phi,0} e^{-\alpha_\phi \mathbf{S}_\phi} \mathbf{V}_\phi^T \end{cases} \tag{4.41}$$

where $\boldsymbol{K}_{p,0} = \text{diag}(k_{p,i}) > 0$, $\boldsymbol{K}_{\phi,0} = \text{diag}(k_{\phi,i}) > 0$, $\alpha_p > 0$, and $\alpha_\phi > 0$.

This stiffness design achieves the desired behavior: indeed, by focusing on the first position term (the second one being equivalent), the chosen $\boldsymbol{K}_p$ will implement a virtual spring of value $k_{p,i} e^{-\alpha \sigma_{p,i}}$ on each of the principal axes of $\boldsymbol{\Sigma}_p(t)$. The stifness will then range from the maximum values $k_{p,i}$ for small variances $\sigma_{p,i} \approx 0$, to negligible values for large variances $\sigma_{p,i}$, with the parameter α_p governing the decrease rate.

Finally, the damping terms are designed, as usual, in order to obtain a critically-damped closed-loop behavior [220]

$$\begin{cases} \boldsymbol{B}_p(t) &= 2(M_p^{-1/2} K_p(t) M_p^{-1/2})^{1/2} \\ \boldsymbol{B}_\phi(t) &= 2(M_\phi^{-1/2} K_\phi(t) M_\phi^{-1/2})^{1/2} \end{cases}. \tag{4.42}$$

4.3.4 Trajectory Refinement through Interactive Task Executions

By using the shared control architecture described in Section 4.3.3, we can obtain a new sample of the trajectory and the context. The obtained sample can be used to refine the model of the trajectory distribution by simply aggregating it to the dataset. When a new trajectory is obtained, we examine the information gain which

Algorithm 1 Learning Trajectory Distribution through Interactive Task Executions

Input: dataset of the trajectories demonstrated by experts and the contexts of each demonstration $\mathcal{D}$, information gain threshold $\mathcal{I}_0$
repeat
  Model the trajectory distribution $p(\boldsymbol{\gamma}|\boldsymbol{r})$
  Update the parameters of the shared control
  Perform the task under the context $\boldsymbol{r}^{\text{new}}$ using the shared control
  Record the obtained data $\mathcal{D}_{\text{new}} = \{\boldsymbol{\gamma}, \boldsymbol{r}^{\text{new}}\}$
  Evaluate the information gain from the new data $\mathcal{I}(\mathcal{D}, \mathcal{D}_{\text{new}})$
  if $\mathcal{I}(\mathcal{D}, \mathcal{D}_{\text{new}}) > \mathcal{I}_0$ **then**
    Aggregate the dataset $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_{\text{new}}$
  end if
until the trajectory distribution learned

can be obtained by adding it to the dataset. Using the dataset $\mathcal{D}$, we model the joint distribution of the context and the state of the system at time t as a Gaussian distribution

$$p_{\mathcal{D}}\left(\begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{x}_m(t) \end{bmatrix}\right) \sim \mathcal{N}_{\mathcal{D}}(\bar{\boldsymbol{x}}_{m,\mathcal{D}}(t), \boldsymbol{\Sigma}_{\mathcal{D}}(t)). \tag{4.43}$$

Let $\mathcal{D}'$ denote the dataset which can be obtained by aggregating the data as $\mathcal{D}' = \mathcal{D} \cup \mathcal{D}_{\text{new}}$. The information gain from the newly obtained data is given by Kullback-Leibler divergence [221] as

$$\begin{aligned} \mathcal{I}(\mathcal{D}, \mathcal{D}_{\text{new}}) =& \sum_{t=0}^{T} D_{\text{KL}}(p_{\mathcal{D}'} || p_{\mathcal{D}}) \\ =& \sum_{t=0}^{T} D_{\text{KL}}(\mathcal{N}_{\mathcal{D}'} || \mathcal{N}_{\mathcal{D}}) \\ =& \frac{1}{2} \sum_{t=0}^{T} \left(\log \left(\frac{\det \boldsymbol{\Sigma}_{\mathcal{D}}(t)}{\det \boldsymbol{\Sigma}_{\mathcal{D}'}(t)} \right) - n + \text{tr}(\boldsymbol{\Sigma}_{\mathcal{D}}^{-1}(t) \boldsymbol{\Sigma}_{\mathcal{D}'}(t)) \right. \\ & \left. + (\bar{\boldsymbol{x}}_{m,\mathcal{D}}(t) - \bar{\boldsymbol{x}}_{m,\mathcal{D}'(t)})^T \boldsymbol{\Sigma}_{\mathcal{D}}^{-1}(t) (\bar{\boldsymbol{x}}_{m,\mathcal{D}}(t) - \bar{\boldsymbol{x}}_{m,\mathcal{D}'}(t)) \right), \end{aligned} \tag{4.44}$$

where n is the dimension of $[\boldsymbol{r}^T, \boldsymbol{x}_m^T(t)]$. If the information gain $\mathcal{I}(\mathcal{D}, \mathcal{D}_{\text{new}})$ is larger than the threshold $\mathcal{I}_0$, the newly obtained data is aggregated to the dataset as $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_{\text{new}}$. Using the information criterion, we can keep the dataset as compact as possible. Our approach for learning shared control from interactive task executions is summarized in Algorithm 1.

Through interactive learning, the trajectory distribution is adapted by the operator's preference. If the operator's preference is stationary, the trajectory distribution induced by the learned model converges to the trajectory distribution induced by the operator as the number of the trajectory samples increase. Therefore, the required amount of control input from the operator is expected to decrease as the learned trajectory distribution is more and more refined.

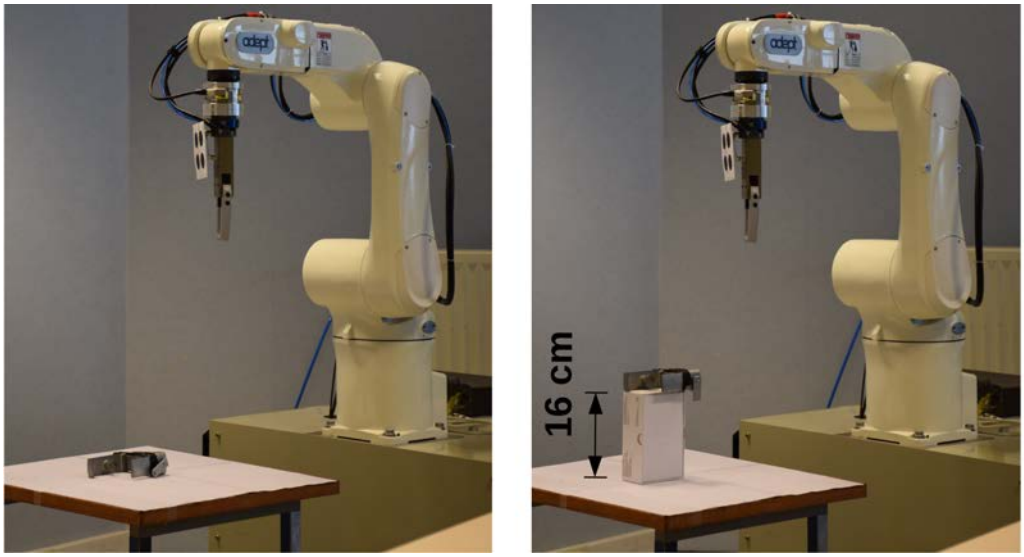


Figure 4.15: The experimental setup. The initial demonstrations were performed with an object on a table. The learned trajectory distribution was then tested with the object on a box.

4.3.5 Experimental Results

We evaluated the proposed approach using the experimental test-bed shown in Fig. 4.14. The slave side is an Adept Viper s850 6-dof serial manipulator equipped with a gripper whereas the master consists of the Haption VIRTUOSE 6D haptic device.

In the considered task, the human operator is required to steer the slave arm towards a target object with the intention of grasping it. The target object is assumed to be fixed and the slave and the master arms go back to their respective pre-defined initial positions after every iteration. A video showing the proposed approach, the test bed and the performed experiments is available here: https://youtu.be/hvzxmwqAH5s. A human operator demonstrated the described task 20 times using the master-slave system. During the demonstrations, the master manipulator was totally compliant and the user did not receive any force feedback.

To evaluate the algorithm, we used data from demonstrations performed under biased conditions. This bias leads to the generation of 'non-optimal' trajectory distributions given certain contexts which is essential in order to induce a significant analysable intervention from the human operator. To introduce the bias, initial demonstrations were performed with the target object placed 16 cm lower than in the testing phase (Fig. 4.15). A human operator performed the task using our shared control framework, and the learned trajectory distribution was refined by aggregating the executed trajectory to the dataset after every iteration. This

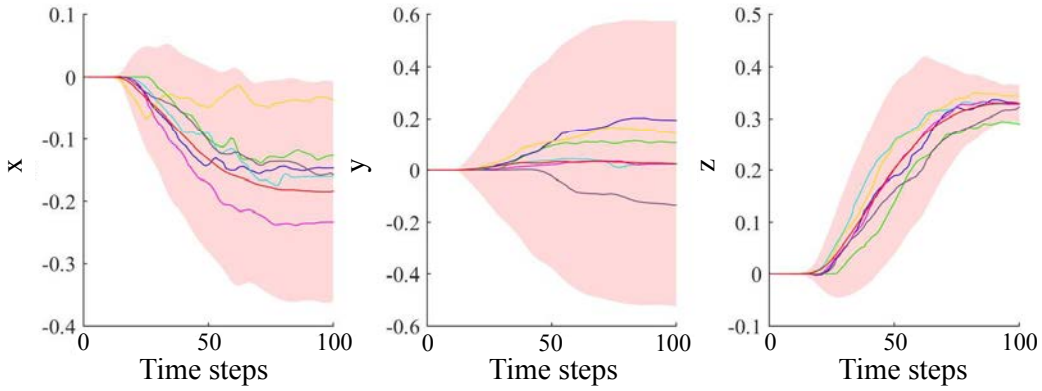


Figure 4.16: The three positional components of the pose of the end effector of the master arm at every time step t during each of the 20 demonstrations

procedure was repeated five times.

The initially demonstrated trajectories are shown in Fig. 4.16. To avoid redundancy, the displayed plots correspond only to the translational position of the end effector although both, positions and orientations of the trajectories, were planned and executed as described in the previous sections.

Fig. 4.17 shows the planned trajectory distribution (plotted in red with the shaded region reflecting the value of the variance) versus the executed trajectory (in blue) for the first, third and last iterations of the experiments. The difference between the planned trajectory distribution and the trajectory executed under the human operator's supervision was large in the first iteration as shown in Fig. 4.17a where the executed trajectory is mainly monotone while the planned distribution shifts from decreasing to increasing as in x and z. Moreover, the planned trajectory was jerky with lots of vibrations due to the prediction uncertainty. Meanwhile, Fig. 4.17b, which corresponds to the third iteration of the experiment, shows a notable improvement in the behavior of the planned trajectory distribution which is now more in line with the executed trajectory. The planned trajectories are, as expected, even better in the fifth iteration (Fig. 4.17c). Although the offset between the planned and executed trajectories still remained after five task executions, we think these are acceptable results. As long as the operator holds the master manipulator, an unintentional force is applied due to the inertia of the operator's hand. However, since the operator did not try to modify the trajectory in the final two iterations, we consider that the offset is acceptable. On the other hand, the smoothness of the trajectory has significantly improved when comparing the planned trajectory generated during the fifth iteration with the one generated in the first. This is mostly visible in x and z since the displacement in the y-axis was limited in the

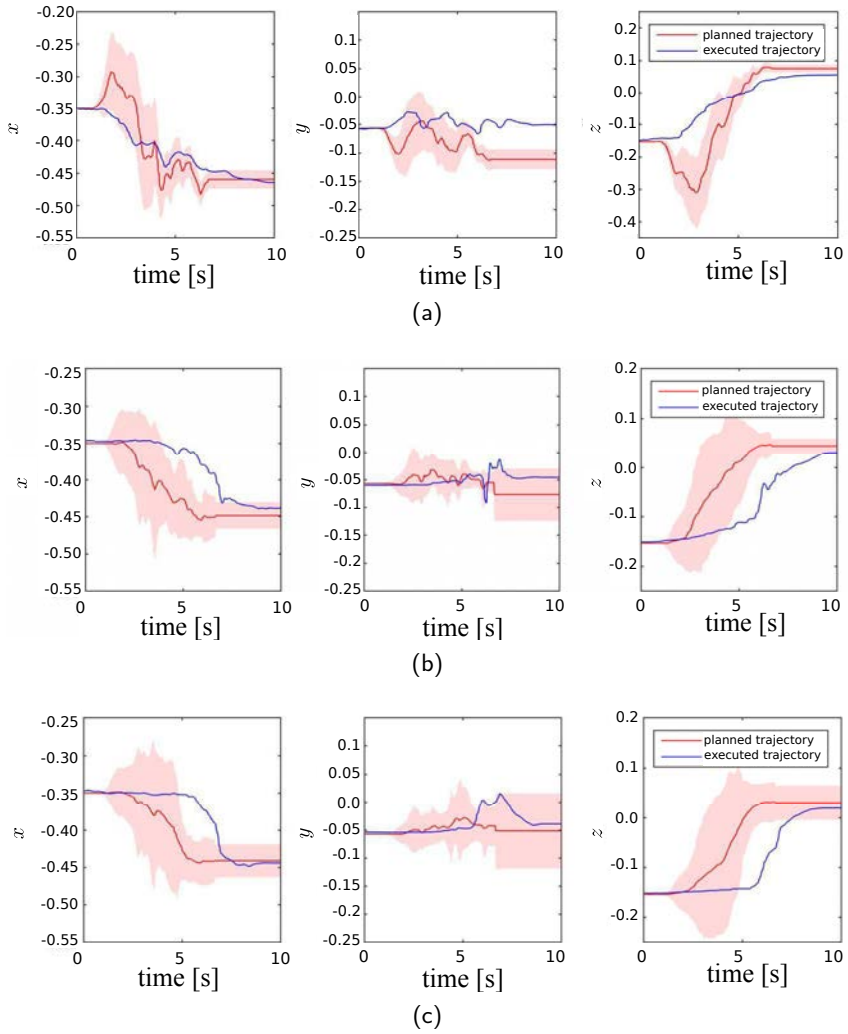


Figure 4.17: Improvement of the trajectory prediction. Blue lines represent the executed trajectory, and red lines represent the distribution of the predicted trajectory. (a)The executed trajectory and the predicted trajectory based on the initial dataset. (b)The executed trajectory and the predicted trajectory after the aggregation of three additional trajectories obtained from the interactive task executions. (c)The executed trajectory and the predicted trajectory after the aggregation of five additional trajectories obtained from the interactive task executions.

described experiment.

Metrics of the intervention of the human operator in each of the 5 iterations of the experiment are shown in Fig. 4.18. The magnitude of the forces and the torques exerted by the human operator at the master side is clearly decreasing with the progression of the experiment. However, as a result of the continuous interaction between the human operator and the end effector of the master arm, these forces will converge to a range but not to zero. The slight increase in the linear force for

iteration 4 is noise resulting from the mentioned interaction.

This result illustrates the efficiency of the shared control architecture in providing the human operator with the needed informative force cues reflecting her/his deviation from the generated trajectory distribution. Moreover, it reflects the adaptability of the learning routine and its swift compliance over iterations with the inputs dictated by the human operator resulting in less intervention from the human operator over subsequent task executions.

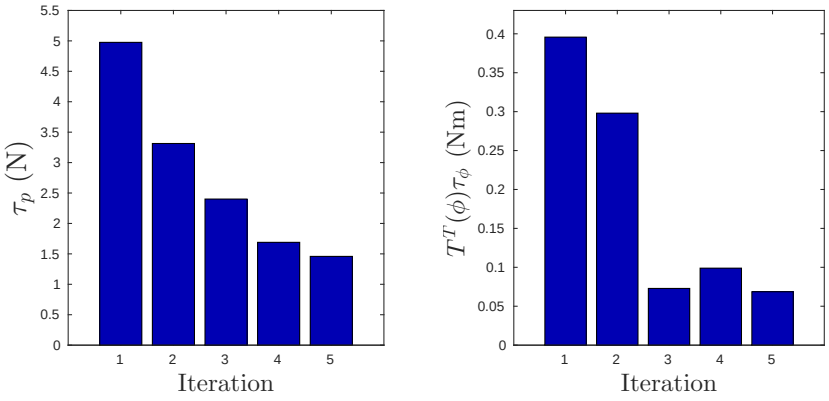


Figure 4.18: Metrics of the applied human intervention during task execution. The intervention from the human operator decreased after interactive task executions. (a)Mean force exerted at the master manipulator. (b)Mean torque exerted at the master manipulator.

In addition, the information gain from each executed trajectory (defined in (4.44)) was analyzed in this experiment as well to check its effectiveness as a token in deciding whether to aggregate the newly executed trajectory to the learning dataset or not. Fig. 4.19 recounts the behavior of the indicator over time showing a decrease in the information gain with every iteration. The horizontal line signifies the threshold above which an execution is aggregated to the dataset. However, for the sake of testing the impact of this aggregation, all the executions were exceptionally aggregated to the dataset in this experiment.

The mentioned figure shows that the most significant information gain was achieved during the first three iterations after which it dropped significantly. This result is in line with Fig. 4.17 which depicts the planned and executed trajectories during the first, third and fifth iterations. The figure shows that the planned trajectory changed significantly between the first and the third iterations. In contrast, the change was more limited between the third and the fifth. We can conclude that the aggregation of the last two executions (for which the information gain was

below the threshold in Fig. 4.19) into the dataset had only little impact on the final results. The observed behavior thus validates the effectiveness of the chosen measure for informative training samples.

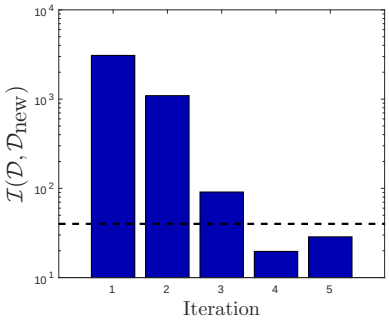


Figure 4.19: Information gain of the additional data. The information gain decreases as the number of the data samples increases.

4.4 Summary

This chapter is divided into three main sections in which we presented three shared-control architectures tackling different aspects of shared control. In Sec. 4.1, we presented a framework with haptic feedback, which enables the intuitive and effective teleoperation of a dual-arm robotic system. The human operator partially controls one arm (manipulator A), equipped with a gripper, through a 6-DoF grounded haptic interface. Haptic feedback provides the operator with information about joint and workspace limits as well as about the presence of singularity configurations and imminent collisions. The other robotic arm (manipulator B) is sharing the same environment with manipulator A and performs an autonomous task in the vicinity of the object to be grasped. To demonstrate the feasibility and effectiveness of our methods, we conducted a human subject experiment in simulated and real environments. Fifteen subjects used manual and assisted control architectures to control manipulator A and grasp an object placed on a conveyor belt. The results showed that the proposed system was efficient in avoiding collisions and increasing the safety of the system in addition to decreasing the overall task completion time.

On the other hand, Sec. 4.2 presented a haptic-enabled shared-control architecture, whose objective is to ease the manipulation of objects in cluttered environments. The architecture is a more general solution than that presented in chapter 3. It gives more freedom to the human operator while still guiding her/him to potential grasping poses over all 6 DoF. A point cloud scan of the environment is used to find potential grasp candidates for all the objects in the scene. These grasping poses are then used by the shared-control algorithm to provide haptic guidance to

the human operator. Dynamic active constraints gently steer the operator toward feasible grasping poses, enabling her/him to intuitively navigate the environment and safely switch between different grasp candidates placed on the same or different objects. Moreover, the algorithm uses the approach in Sec. 4.1 to also ensure that the operator complies with certain safety constraints of the system (e.g., workspace, joints, and singularity limitations) by introducing additional haptic cues. To enable the operator to differentiate between these two haptic information (guiding toward a grasping pose vs. proximity to unsafe configuration), we used kinesthetic and vibrotactile feedback. Active constraints providing grasping guidance are enforced by conveying kinesthetic feedback through a 6-DoF grounded haptic interface; active constraints enforcing the safety limits are conveyed via kinesthetic feedback provided through the same 6-DoF haptic interface *and* vibrotactile feedback provided by a custom haptic bracelet. We also presented an implementation for a 6-DoF robotic teleoperation system as well as a principled human subject evaluation in a real environment.

Finally, Sec. 4.3 presented a different framework which allows for adaptively controlling the balance between autonomy of the controller and the human intervention. The framework is guided by a trajectory distribution learned from human experts and exploits the variance of the demonstrated trajectories as an indicator of the human experts' preference. The haptic feedback exerted at the master system was controlled accordingly reflecting the confidence of autonomy in its choices and the preference of the expert operators. Experimental results show that the learned trajectory distribution was refined and the performance of our shared control improved through interactive task executions.

Chapter 5

Post-Grasping Phase

Contents

5.1 Related Works **150**
5.2 Problem Formulation **151**
5.3 Task-Oriented Velocity Manipulability **151**
5.3.1 TOV Definition 152
5.3.2 TOV Gradient 154
5.4 Haptic Feedback **156**
5.5 Experimental Results **157**
5.5.1 First Experiment 157
5.5.2 Second Experiment 158
5.6 Summary **160**

In chapters 3 and 4, we tackled the pre-grasp phase of manipulation proposing efficient shared-control architectures to facilitate the task on the operator and assist her/him during execution. However, another essential component to guarantee a successful manipulation is the feasibility of the post grasp task, that is, how easily the target object can be moved along some desired trajectory once the grasp has been completed. Indeed, besides grasping stability, the chosen grasping pose also impacts the trajectory which the manipulator should follow to achieve the desired post-grasp manipulation action. Surprisingly, while the post-grasp phase is the end-goal of grasping, it has been rarely tackled when designing assisted or autonomous control systems. This chapter focuses on this phase and presents a shared-control architecture which provides the operator with haptic cues that assist in selecting the "best" grasping poses for the post-grasp manipulative task. To this end, we introduce a task-oriented velocity manipulability cost function (TOV), which is used to quantify the kinematic capability of a manipulator during post-grasp motions, and exploit this metric for generating the haptic cues provided to

Figure 5.1: Left: the slave manipulator arm. The object to be manipulated is on the table in front of the slave robot. Right: The master device used by the operator for sending commands to the slave manipulator and receiving force cues.

the human operator. We show that grasps which minimise the TOV result in significantly reduced control effort of the manipulator, compared to other feasible grasps and demonstrate that by experiments on real and simulated robots. The work presented in this chapter has been done in cooperation with the University of Birmingham and is published in [25] and an illustrative video is available at https://youtu.be/JAFeKkb1EPY.

As discussed in the previous chapters, a variety of approaches for autonomous grasp planning have recently been proposed [222, 223, 224, 225]. However, such methods are still not accepted as being industrially robust, especially in safety-critical applications where human judgement is still considered the gold standard. On the other hand, autonomous trajectory planning has been widely studied for many years, and modern approaches, e.g. [226, 227], are sufficiently reliable for practical applications. For example, in the pick-and-place task shown in Fig. 5.1, a computer vision algorithm can be used to detect the initial and goal poses of the manipulated object, and a motion planner [226] can autonomously generate a trajectory for moving the object.

However, the pre-grasp and post-grasp phases of manipulation have predominantly been considered separately in previous literature. However, a stable grasp, selected by the human operator, may often result in a kinematically infeasible trajectory of the robot for the post-grasp motion of the grasped object. Therefore, it is important for the human to select a grasp pose that allows the manipulator to perform the desired post-grasp motions (or select a grasp which maximises the post-grasp manipulability of the robot more generally). Nonetheless, without appropriate cues for guidance, a human operator will likely be oblivious to the post-grasp consequences for manipulability, when choosing a particular grasp pose merely on the basis of perceived "graspability".

To overcome this problem, we first introduce a cost function (task-relevant velocity manipulability - TOV) for quantifying the kinematic capability of the manipulator over desired post-grasp motions of the manipulated object. However, such metrics are not sufficiently intuitive to be taken into account by the human operator during the reach-to-grasp approach phase. Therefore, secondly, we propose a human-in-the loop optimisation framework, in which the autonomous agent computes the gradient of the TOV, along which TOV decreases while the kinematic manipulability of the robot increases. Thirdly, by transforming this TOV gradient into haptic force cues, the human operator is encouraged to steer the slave robot towards grasp poses that are stable, while also being optimal in terms of maximising the post-grasp kinematic capability of manipulator.

Previously, functions such as the kinematic manipulability ellipsoid and manipulability value [228, 229] have been proposed for evaluating the kinematic capability of manipulation. The larger the kinematic manipulability value, the larger the capability that a manipulator has (at its present configuration) to move in arbitrary directions for future trajectories. Lee et. al [230] introduced a definition of the manipulability ellipsoid for a closed kinematic chain, comprising two arms holding an object in a bi-manual grasp. Zhang et. al [231] proposed a manipulability criterion only along the direction of linear velocity of the centre of mass of a Cricket robot. A task-oriented force manipulability ellipsoid was proposed in [232] which is the integral of the force manipulability along a proposed robot trajectory.

In contrast to previous works, in this paper we define *task-oriented velocity manipulability* (TOV) cost function to be the integral of the *inverse* of the velocity manipulability along the direction of movement over the post-grasp path. Our experimental results demonstrate that minimising TOV results in the minimum manipulator control effort, i.e. minimum norm of the manipulator's joint velocities over the post-grasp motion. Furthermore, by definition, configurations with singularities along the direction of movement cause a very large value of TOV. This means that minimising TOV corresponds to maximising kinematic capability.

Note that the TOV gradient is independent of the position of the grasped object, so that following the force cues along the TOV gradient may result in a gripper pose that is very far from the object. However, the human operator can still decide to compensate for poor post-grasp kinematic capability (communicated via force cues) by selecting a different feasible grasp pose.

The proposed approach can significantly reduce the workload of human operators during the approach-to-grasp phase, because the operator does not need to worry about the considerations of post-grasp manipulability. These are handled automatically by the autonomous agent, and communicated to the human opera-

tor intuitively, through haptic force cues which encourage the selection of highly manipulable grasp poses without engendering additional cognitive effort.

The remainder of the chapter is structured as follows. The related literature is presented in section 5.1. In section 5.3 the problem is formulated and *task-oriented manipulability* cost function (TOV) is introduced. Next, the derivative of the TOV is discussed, which is then used in section 5.4 to provide force cues for the haptic feedback control law. In section 5.5, the effectiveness of the proposed approach is demonstrated by simulation experiments with a 2 link manipulator, as well as several experiments with a real 6-dof robot in a pick-and-place task.

5.1 Related Works

Grasping and manipulative motion planning have been widely studied in the robotics literature [222, 224]. Most of the studies, however, focused either on the first phase of grasping and manipulation, namely approach phase [225], or on the second phase, namely autonomously manoeuvring the manipulator [227].

There are a few numbers of studies on jointly considering the problem of grasping an object, manipulating it and delivering it to the desired pose. For example, two-phase optimisations were used in [233] to generate the contact necessary for making a stable grasp on an object and to find the optimal object path that can be followed, given the optimal grasping configuration. In contrast, [234, 229] studied the optimal grasps resulting in a maximum manipulability at initial grasp configuration. Similarly, [232, 235] showed that different grasps can result in different task-oriented force manipulabilities as well as different torque efforts over the post-grasp motions. However, the main assumption of these works is that a planner can generate many stable grasp poses. Nonetheless, a reliable autonomous system has not yet developed that generates a stable grasp for an arbitrary object in a real world example. This is still an open research topic in autonomous robotics [222].

On one hand, a human operator can steer the slave robot to make the necessary contacts for a stable grasp. On the other hand, the operator does not have enough intuition and understanding about the kinematic capability of the manipulator during post-grasp manipulative motions while she/he is steering the slave arm to approach and grasp the object. Our approach allows a human operator to select a grasp by looking at the remote workspace and by using the force cues along the gradient of TOV cost function. Hence, a reduced TOV is obtained as our experimental results illustrate. The reduced TOV is equivalent to the increased kinematic capability and to decreased norm of joint velocities over post-grasp motion. We demonstrate the effectiveness of the approach by a series of experiments with an Adept Viper s850 6-dof serial manipulator. Our experimental results evidence that

the proposed control architecture eases the tele-operation by providing the force cues to the human operator via a master arm.

5.2 Problem Formulation

Similarly to the setup considered previously in Sec. 3.1 but excluding the manipulator with the camera, we consider three reference frames: $\mathcal{F}_g \in SE(3)$ attached to the robot end-effector, $\mathcal{F}_o \in SE(3)$ attached to the centre of mass (CoM) of the object to be grasped, and $\mathcal{W} \in SE(3)$ as a world frame. We also let ${}^o\boldsymbol{x}_g = \{{}^o\boldsymbol{p}_g, {}^o\boldsymbol{R}_g\} \in SE(3)$ be the relative pose between the gripper and the target object which, in our context, represents a possible grasping pose.

We assume that a trajectory for the object to be grasped in the world frame is *given*, for instance generated by any external planner/decision-making algorithm. In the common example of pick-and-place tasks, the trajectory of the grasped object could be, e.g., generated based on the initial location of the object and desired target position regardless of the manipulator actually performing the task. Let then ${}^w\boldsymbol{x}_o(s) = \{{}^w\boldsymbol{p}_o(s), {}^w\boldsymbol{R}_o(s)\} \in SE(3)$, $0 \leq s \leq 1$, be the object desired path in $\mathcal{W}$, with s being any parametrization such that $s = 0$ represents the starting point and $s = 1$ the endpoint of the path[1]. By standard kinematics, the corresponding path for the robot end-effector in $\mathcal{W}$ is then just

$$\begin{aligned} {}^w\boldsymbol{R}_g(s) &= {}^w\boldsymbol{R}_o(s){}^o\boldsymbol{R}_g \\ {}^w\boldsymbol{p}_g(s) &= {}^w\boldsymbol{p}_o(s) + {}^w\boldsymbol{R}_o(s){}^o\boldsymbol{p}_g \end{aligned}. \tag{5.1}$$

The main goal of this paper is to generate force cues for a human operator able to inform about the optimality of a candidate grasping pose ${}^o\boldsymbol{x}_g$ w.r.t. the TOV manipulability index evaluated over the whole robot path (5.1). The latter is, indeed, function of the object path ${}^w\boldsymbol{x}_o(s)$ (a given quantity) *and* of the grasping pose ${}^o\boldsymbol{x}_g$ that will then act as an 'optimisation variable' for the proposed optimality criterion. We now proceed to detail the chosen cost function and the expression of its gradient w.r.t. the optimisation variable ${}^o\boldsymbol{x}_g$.

5.3 Task-Oriented Velocity Manipulability

In this work, we are interested in cueing about the location of the grasping pose ${}^o\boldsymbol{x}_g$ that optimises a particular index related to the classical notion of (kinematic) manipulability evaluated along the whole object/robot path. Towards this end, and

[1]The actual trajectory executed by the robot can be obtained by choosing any suitable timing law $s(t)$ for travelling along the desired path. Since the optimisation problem considered in the next sections is only function of the path geometry, we prefer to decouple the geometric component of the problem from its temporal component.

similarly to the previous chapters (see Sec. 3.1), let $\boldsymbol{q} \in \mathbb{R}^6$ be the joint vector of the considered 6-DoF manipulator arm and

$$\boldsymbol{v}_g = \begin{bmatrix} \dot{\boldsymbol{p}}_g \\ \boldsymbol{\omega}_g \end{bmatrix} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{5.2}$$

be the geometric Jacobian relating joint velocities to the end-effector linear/angular velocities $\boldsymbol{v}_g = (\dot{\boldsymbol{p}}_g, \boldsymbol{\omega}_g) \in \mathbb{R}^6$ expressed in the end-effector frame $\mathcal{F}_g$.

5.3.1 TOV Definition

As well-known, the classical (kinematic) manipulability ellipsoid, which for non-redundant manipulators is defined by the equation

$$\boldsymbol{v}_g^T(\boldsymbol{J}\boldsymbol{J}^T)^{-1}\boldsymbol{v}_g = 1 \tag{5.3}$$

represents the capability of the robot manipulator in generating task space velocities for a given norm of joint velocities (thus, representing some sort of dexterity of the robot arm). In this work we are interested in maximising (in an integral sense) a particular *task-oriented* manipulability measure derived from (5.3): the radius of the manipulability ellipsoid along the tangent vector to the desired path in task space. This is meant to ease as much as possible the execution of the desired trajectory (5.1) by the manipulator arm with the smallest possible control effort (norm of the joint velocities).

Let then $\boldsymbol{q}(s)$ be the path in joint space (generated by the robot inverse kinematics) associated to the end-effector trajectory (5.1), and $\boldsymbol{v}_g(s)$ the corresponding linear/angular end-effector velocity for some $0 \leq s \leq 1$. Decompose $\boldsymbol{v}_g(s)$ as $\boldsymbol{v}_g(s) = a(s)\bar{\boldsymbol{v}}_g(s)$, with $a(s)$ representing the norm of $\boldsymbol{v}_g(s)$ and $\bar{\boldsymbol{v}}_g(s)$ its (unit-norm) direction. From (5.3) it follows that, along the planned path,

$$a^2(s)\bar{\boldsymbol{v}}_g^T(s)(\boldsymbol{J}(\boldsymbol{q}(s))\boldsymbol{J}^T(\boldsymbol{q}(s)))^{-1}\bar{\boldsymbol{v}}_g(s) = 1. \tag{5.4}$$

It is easy to verify that the quantity $a(s)$ solution of (5.4) represents the length of the ellipsoid radius along the direction $\bar{\boldsymbol{v}}_g(s)$, see also the illustrative example in Figs. 5.2–5.3. Since our aim is to *maximise* the quantity $a(s)$ along the whole path, exploiting the relationship (5.4), we can define the following integral cost function to be *minimised*

$$H = \int_0^1 \frac{1}{a^2(s)}\mathrm{d}s = \int_0^1 \bar{\boldsymbol{v}}_g^T(s)(\boldsymbol{J}(\boldsymbol{q}(s))\boldsymbol{J}^T(\boldsymbol{q}(s)))^{-1}\bar{\boldsymbol{v}}_g(s)\mathrm{d}s, \tag{5.5}$$

which we then denote as Task-oriented velocity manipulability (TOV).

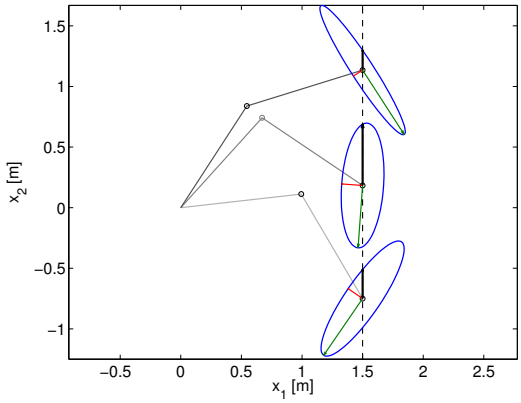


Figure 5.2: A 2-D manipulator follows a vertical line from bottom to top shown with dashed blue line. The manipulability ellipsoids are also depicted at several configurations. Red and green arrows represent the ellipsoid major/minor axes. The proposed TOV measure (black arrow) is obtained by evaluating the radius of the manipulability ellipsoid along the desired end-effector path.

Note that, because of (5.1), the various terms in the integrand of (5.5) are ultimately function of s and of the grasping pose ${}^o\boldsymbol{x}_g$ (our optimisation variable). Therefore, the cost (5.5) can be expressed as

$$H({}^o\boldsymbol{x}_g) = \int_0^1 h({}^o\boldsymbol{x}_g,\, s)\mathrm{d}s, \tag{5.6}$$

which highlights the dependency on the ${}^o\boldsymbol{x}_g$ (as desired).

We now show an illustrative example of the introduced TOV index: the example is obtained for a 2-D link manipulator arm following a vertical line as depicted in Fig. 5.2. Therein, the manipulability ellipsoid is shown at the end-effector during motion, with the green and red lines representing the major/minor ellipse axes. The black line is the radius of the ellipsoid along the tangent to the current path, that is, the previously introduced quantity $a(s)$. By minimising $H({}^o\boldsymbol{x}_g)$ we aim at maximising $a(s)$ along the whole planned path.

Furthermore, we simulate a possible grasping task in order to show how the proposed measures change as a function of the grasping pose (which is the optimisation variable). In the example of Fig. 5.3a, the 2-D manipulator must grasp a rectangular object (red rectangle) from the top edge and place it at the target position shown by brown rectangle. The object must follow the path represented by the dashed line in the picture. In Fig. 5.3b, the average ellipsoid radius $a(s)$ (top) and the norm of the joint velocities integrated over the whole trajectory (bottom) are shown for different grasp candidates. As shown in these figures, the grasp location

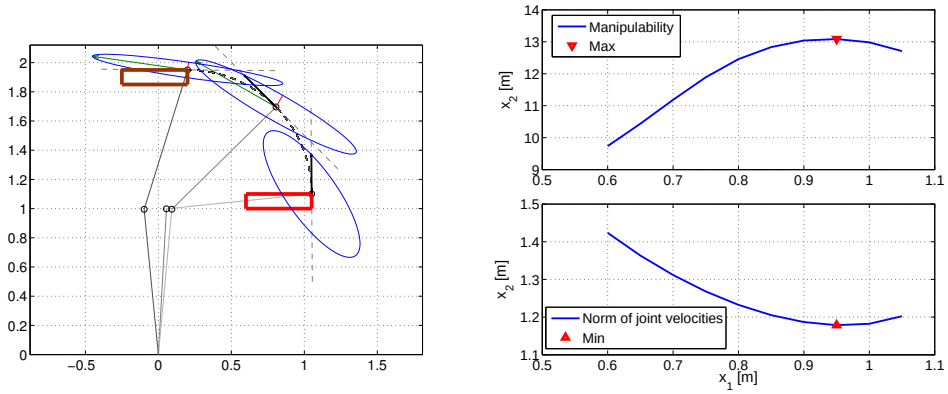


Figure 5.3: (a) shows a 2-D planar manipulator following a half circle curve (black dashed line). The manipulability ellipsoid is shown at several configurations, with the proposed TOV measure highlighted by a black arrow. An object to be grasped is assumed to be located at the red rectangle. The robot must grasp the object from the top edge and place it at the target position (brown rectangle). (b) top reports the value of the average ellipsoid radius along the trajectory $a(s)$ (5.4) as a function of different grasping poses, while the bottom shows the behaviour of the integral of the joint velocity norm over the path. As expected (and desired) the latter quantity has a minimum in correspondence of the maximum of average $a(s)$ (attained for a particular 'optimal' grasping pose ${}^o\boldsymbol{x}_g$).

has a clear influence on $a(s)$ and, more importantly, on the resulting (integral) joint velocity norm. In particular, the maximum of $a(s)$ corresponds to the minimum of the joint velocity norm (as expected and desired).

5.3.2 TOV Gradient

We now proceed to detail an explicit expression for the gradient of $H({}^o\boldsymbol{x}_g)$ w.r.t. the grasping pose ${}^o\boldsymbol{x}_g$: this gradient information will in fact be used for cueing the human operator about which directions to move in order to minimise the TOV index over the planned path. For the sake of the gradient computation, we choose to represent the orientation component of ${}^o\boldsymbol{x}_g$ with a quaternion parametrization. Therefore, in the following ${}^o\boldsymbol{\rho}_g \in \mathbb{S}^3$ will represent the unit-quaternion associated to the rotation matrix ${}^o\boldsymbol{R}_g$, and $\boldsymbol{T}({}^o\boldsymbol{\rho}_g) \in \mathbb{R}^{4\times 3}$ the usual mapping matrix from angular velocities to quaternion rates, i.e., such that

$$ {}^o\dot{\boldsymbol{\rho}}_g = \boldsymbol{T}({}^o\boldsymbol{\rho}_g){}^o\boldsymbol{\omega}_g. \tag{5.7} $$

From (5.6) one has

$$\frac{\partial H({}^o\boldsymbol{x}_g)}{\partial {}^o\boldsymbol{x}_g} = \int_0^1 \frac{\partial h({}^o\boldsymbol{x}_g,\, s)}{\partial {}^o\boldsymbol{x}_g} \mathrm{d}s \tag{5.8}$$

whose integrand, exploiting (5.5) (and omitting the dependence on s and ${}^o\boldsymbol{x}_g$ for notational sake), can be expanded as

$$\begin{aligned}\frac{\partial h}{\partial {}^o\boldsymbol{x}_g} = \frac{\partial \bar{\boldsymbol{v}}_g^T}{\partial {}^o\boldsymbol{x}_g}(\boldsymbol{J}\boldsymbol{J}^T)^{-1}\bar{\boldsymbol{v}}_g + \bar{\boldsymbol{v}}_g^T \frac{\partial (\boldsymbol{J}\boldsymbol{J}^T)^{-1}}{\partial {}^o\boldsymbol{x}_g}\bar{\boldsymbol{v}}_g + \\ \bar{\boldsymbol{v}}_g^T (\boldsymbol{J}\boldsymbol{J}^T)^{-1}\frac{\partial \bar{\boldsymbol{v}}_g}{\partial {}^o\boldsymbol{x}_g}.\end{aligned} \tag{5.9}$$

We now provide an explicit expression of the various terms in (5.9). Let us first focus on the term $\partial \bar{\boldsymbol{v}}_g / \partial {}^o\boldsymbol{x}_g$: we recall that $\bar{\boldsymbol{v}}_g = \boldsymbol{v}_g / \|\boldsymbol{v}_g\|$, and $\boldsymbol{v}_g = (\dot{\boldsymbol{p}}_g, \boldsymbol{\omega}_g)$. Therefore, the evaluation of $\partial \bar{\boldsymbol{v}}_g / \partial {}^o\boldsymbol{x}_g$ requires an explicit expression for $\partial \dot{\boldsymbol{p}}_g / \partial {}^o\boldsymbol{x}_g$ and $\partial \boldsymbol{\omega}_g / \partial {}^o\boldsymbol{x}_g$. Since the following relationship holds

$$\begin{bmatrix} \dot{\boldsymbol{p}}_g \\ \boldsymbol{\omega}_g \end{bmatrix} = \begin{bmatrix} {}^g\boldsymbol{R}_o\, {}^o\dot{\boldsymbol{p}}_g \\ {}^g\boldsymbol{R}_o\, {}^o\boldsymbol{\omega}_g \end{bmatrix} = \begin{bmatrix} {}^g\boldsymbol{R}_o\, ({}^o\dot{\boldsymbol{p}}_o + [{}^o\boldsymbol{\omega}_o]_\times\, {}^o\boldsymbol{p}_g) \\ {}^g\boldsymbol{R}_o\, {}^o\boldsymbol{\omega}_g \end{bmatrix}, \tag{5.10}$$

with $[\cdot]_\times$ being the usual skew-symmetric operator, one simply has

$$\frac{\partial \dot{\boldsymbol{p}}_g}{\partial {}^o\boldsymbol{x}_g} = \begin{bmatrix} \dfrac{\partial \dot{\boldsymbol{p}}_g}{\partial {}^o\boldsymbol{p}_g} \\ \dfrac{\partial \dot{\boldsymbol{p}}_g}{\partial {}^o\boldsymbol{\rho}_g} \end{bmatrix} = \begin{bmatrix} {}^g\boldsymbol{R}_o[{}^o\boldsymbol{\omega}_o]_\times \\ \dfrac{\partial {}^g\boldsymbol{R}_o}{\partial {}^o\boldsymbol{\rho}_g}\, ({}^o\dot{\boldsymbol{p}}_o + [{}^o\boldsymbol{\omega}_o]_\times\, {}^o\boldsymbol{p}_g) \end{bmatrix}. \tag{5.11}$$

Here, $\partial {}^g\boldsymbol{R}_o / \partial {}^o\boldsymbol{\rho}_g$ is the partial derivative of a rotation matrix w.r.t. its quaternion representation, which can be directly obtained from the analytic expression of ${}^g\boldsymbol{R}_o$ in terms of ${}^o\boldsymbol{\rho}_g$. Analogously, it also follows

$$\frac{\partial \boldsymbol{\omega}_g}{\partial {}^o\boldsymbol{x}_g} = \begin{bmatrix} \dfrac{\partial \boldsymbol{\omega}_g}{\partial {}^o\boldsymbol{p}_g} \\ \dfrac{\partial \boldsymbol{\omega}_g}{\partial {}^o\boldsymbol{\rho}_g} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \dfrac{\partial {}^g\boldsymbol{R}_o}{\partial {}^o\boldsymbol{\rho}_g}\, {}^o\boldsymbol{\omega}_g \end{bmatrix}. \tag{5.12}$$

The expressions (5.11–5.12) then allow the evaluation of the term $\partial \bar{\boldsymbol{v}}_g / \partial {}^o\boldsymbol{x}_g$ and, thus, of the first and third terms of (5.9).

As for the second term of (5.9), we note that (see, e.g., [236])

$$\begin{aligned}\frac{\partial (\boldsymbol{J}\boldsymbol{J}^T)^{-1}}{\partial {}^o\boldsymbol{x}_g} = -(\boldsymbol{J}\boldsymbol{J}^T)^{-1}\frac{\partial (\boldsymbol{J}\boldsymbol{J}^T)}{\partial {}^o\boldsymbol{x}_g}(\boldsymbol{J}\boldsymbol{J}^T)^{-1} \\ = -(\boldsymbol{J}\boldsymbol{J}^T)^{-1}\left(\frac{\partial \boldsymbol{J}}{\partial {}^o\boldsymbol{x}_g}\boldsymbol{J}^T + \boldsymbol{J}\frac{\partial \boldsymbol{J}^T}{\partial {}^o\boldsymbol{x}_g}\right)(\boldsymbol{J}\boldsymbol{J}^T)^{-1}\end{aligned} \tag{5.13}$$

Exploiting the chain rule, we can decompose $\partial \boldsymbol{J}/\partial {}^o\boldsymbol{x}_g$ as

$$\frac{\partial \boldsymbol{J}}{\partial {}^o\boldsymbol{x}_g} = \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{q}} \frac{\partial \boldsymbol{q}}{\partial {}^o\boldsymbol{x}_g} \tag{5.14}$$

The term $\partial \boldsymbol{J}/\partial \boldsymbol{q}$ can clearly be computed from the (explicit) expression of the geometric Jacobian $\boldsymbol{J}$. As for $\partial \boldsymbol{q}/\partial {}^o\boldsymbol{x}_g$ we exploit the relationships

$$\begin{cases} {}^o\dot{\boldsymbol{p}}_g &= {}^o\dot{\boldsymbol{p}}_g = {}^o\boldsymbol{R}_g\dot{\boldsymbol{p}}_g = {}^o\boldsymbol{R}_g\boldsymbol{J}_{\dot{\boldsymbol{p}}}\dot{\boldsymbol{q}} \\ {}^o\dot{\boldsymbol{\rho}}_g &= \boldsymbol{T}({}^o\boldsymbol{\rho}_g){}^o\boldsymbol{\omega}_g = \boldsymbol{T}({}^o\boldsymbol{\rho}_g){}^o\boldsymbol{R}_g\boldsymbol{\omega}_g \\ & \qquad = \boldsymbol{T}({}^o\boldsymbol{\rho}_g){}^o\boldsymbol{R}_g\boldsymbol{J}_{\boldsymbol{\omega}}\dot{\boldsymbol{q}} \end{cases} \tag{5.15}$$

where $\boldsymbol{J}_{\dot{\boldsymbol{p}}}$ and $\boldsymbol{J}_{\boldsymbol{\omega}}$ are the 3×6 block rows of the geometric Jacobian $\boldsymbol{J}$ associated to the linear and angular velocities, respectively. From (5.15) it then follows

$$\begin{cases} \dfrac{\partial \boldsymbol{q}}{\partial {}^o\boldsymbol{p}_g} &= \boldsymbol{J}_{\dot{\boldsymbol{p}}}^{\dagger}{}^o\boldsymbol{R}_g^T \\ \dfrac{\partial \boldsymbol{q}}{\partial {}^o\boldsymbol{\rho}_g} &= \boldsymbol{J}_{\boldsymbol{\omega}}^{\dagger}\,{}^o\boldsymbol{R}_g^T\;\boldsymbol{T}^{\dagger}({}^o\boldsymbol{\rho}_g). \end{cases}, \tag{5.16}$$

which, when plugged in (5.14), allows evaluation of (5.13) and, thus, of (5.8–5.9).

We then now proceed to describe the design of the force cues provided to the operator, which are generated by exploiting the gradient (5.8).

5.4 Haptic Feedback

We consider the same bilateral force-feedback system described previously in Sec. 3.2.3. In this section, we consider a velocity-to-velocity coupling between the master and the slave (Sec. 3.2.3.1) such that

$${}^w\boldsymbol{v}_g = {}^w\boldsymbol{v}_m. \tag{5.17}$$

As for the force cues, we recall that the goal of the haptic feedback in this scenario is to inform the operator about which direction to move in order to minimise the proposed (integral) TOV measure. The variation of the cost H is related to the velocity of the master end-effector by

$$\dot{H} = \frac{\partial H}{\partial {}^o\boldsymbol{x}_g}\boldsymbol{Q}({}^w\boldsymbol{R}_o, {}^o\boldsymbol{\rho}_g){}^w\boldsymbol{v}_m, \tag{5.18}$$

where

$$\boldsymbol{Q}({}^w\boldsymbol{R}_o, {}^o\boldsymbol{\rho}_g) = \begin{bmatrix} {}^o\boldsymbol{R}_w & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T}({}^o\boldsymbol{\rho}_g){}^o\boldsymbol{R}_w \end{bmatrix} \tag{5.19}$$

rotates the velocity vector from the world frame to the object frame and maps the resulting angular velocity to the corresponding quaternion rate (see (5.7)).

The desired behaviour can then be obtained by implementing a force cue $\boldsymbol{f} \in \mathbb{R}^6$ directed along the negative gradient given in eq. (5.18), i.e.,

$$\boldsymbol{f} = -\left(\frac{\partial H}{\partial {}^o\boldsymbol{x}_g}\boldsymbol{Q}({}^w\boldsymbol{R}_o, {}^o\boldsymbol{\rho}_g)\right)^T. \tag{5.20}$$

Finally, $\boldsymbol{f}$ is plugged in (3.22) and the resulting force feedback signal

$$\boldsymbol{\tau} = -\mathbf{B}_m\dot{\boldsymbol{x}}_m + \boldsymbol{f}. \tag{5.21}$$

will then cue the operator about which direction to move in order to minimise the proposed cost function $H({}^o\boldsymbol{x}_g)$ and, as consequence, maximise $a(s)$ over the path.

5.5 Experimental Results

Several experiments were conducted in order to test the described architecture. The experimental test-bed used is the same test-bed used in the experiments in the previous chapters and is shown in Fig. 5.1. The slave side consists of an Adept Viper s850 6-DoF serial manipulator equipped with a linear pneumatic gripper whereas the master device is a Haption VIRTUOSE 6-DoF haptic device. A video of the experiments is available at https://youtu.be/JAFeKkb1EPY.

The user is required to grasp the object for performing a pick and place task. As explained, the pick and place task is pre-defined, i.e., the path that the object is required to follow is fixed and known beforehand. By acting on the master device, however, the user can choose the grasping pose that seems to her/him the most convenient. While approaching the object, the user is provided with a force cue informing about where to move the gripper in order to minimise the proposed (integral) TOV measure. It is ultimately up to the user's decision where to grasp the object, but this decision is an informed one thanks to the force cues. The user will indeed have the possibility of weighting between the most suitable grasping position (according to the her/his preference) and the minimisation of the proposed optimality index which implies a reduced control effort (and better overall dexterity) for the subsequent autonomous pick-and-place task.

5.5.1 First Experiment

The goal of the first experiment is to verify that the provided force cues are indeed able to guide the user towards a pose that can minimise the TOV measure and, as a consequence, minimise the integral norm of the joint velocities over the planned

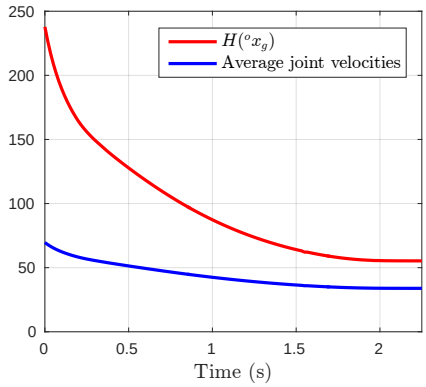


Figure 5.4: Behaviour of the cost function $H(^o\boldsymbol{x}_g)$ and of the average joint velocity: note how $H(^o\boldsymbol{x}_g)$ monotonically decreases as the user follows the provided force cues (as expected). Consequently, also the integral joint velocity norm decreases as well, as the gripper reaches an optimal pose.

path. The user in this case was thus simply asked to passively follow the force cues during the approaching phase.

The results of this experiment are reported in Fig. 5.4. It is worth noting the monotonic decrease of the TOV measure ($H\,(^o\boldsymbol{x}_g)$; red line) as the user follows the received force cues, thus confirming that they are actually steering the user (and the gripper pose) along the negative gradient of $H\,(^o\boldsymbol{x}_g)$. In addition, the green line depicts the behaviour of the integral norm of the joint velocities that would be required to travel along the path: as the grasping pose reaches its final optimal location, the joint velocity norm reaches a minimum as expected.

5.5.2 Second Experiment

The algorithm was then tested in a more realistic scenario. In this experiment, three different configurations, i.e., three different post-grasp pick-and-place paths for the object, were chosen (in translation and rotation). The user was first asked to approach and grasp the object *without* being fed with force cues. Subsequently, the haptic feedback was activated and the user was asked to reach again a grasping location, but this time while being assisted by the haptic feedback. The experiment was repeated six times for each configuration (i.e., each pick-and-place path), for then a total of six times with haptic guidance and six times without haptic guidance.

The three object trajectories in the robot base frame were chosen as follows:

- Trajectory 1: A pure translation of 35 cm along the y-axis and 15 cm along the z-axis.

- Trajectory 2: A translation of 25 cm along the y-axis, 15 cm along the z-axis and a rotation of 90 degrees around the y-axis applied at the centre of gravity of the object.
- Trajectory 3: A translation of 5 cm along the x-axis, 25 cm along the y-axis, 15 cm along the z-axis and a rotation of 90 degrees around the z-axis applied at the centre of gravity of the object.

Fig. 5.5 shows the results. The left figure shows the mean and variance of the average joint velocities over the post-grasp trajectory whereas the right one shows the mean and variance of the cost function $H\left({}^{o}\boldsymbol{x}_{g}\right)$.

For the first tested configuration, the haptic guidance helped in decreasing $H\left({}^{o}\boldsymbol{x}_{g}\right)$ by 40% w.r.t. the case without haptic guidance. On the other hand, the average joint velocities decreased by 25%. This impact is, however, much larger for the second configuration where $H\left({}^{o}\boldsymbol{x}_{g}\right)$ decreased by a factor of 87% (w.r.t. the case without force cues), while joint velocities decreased by a factor of 63%. Finally, a similar behaviour, with guidance and without guidance, was observed for the third configuration where the force was actually guiding the user towards the same intuitive position that she/he would have chosen also without the guidance.

To have a better understanding of the results, Fig. 5.6 shows the mean grasping position for the first configuration both with and without guidance. Taking into account the considered shape of the object, the user (who in general is not a robotics expert and has, thus, a limited understanding of the kinematic capabilities of the robot), would just grasp from the easiest/most intuitive grasping position, which is the one shown in Fig. 5.6(left). This behaviour was indeed reflected in the results. However, when guided by the force cues, the user was successfully capable of steering the robot towards a different grasping position (Fig. 5.6(right)) which is much more convenient in view of the post-grasp planned path, but still guarantees a proper gripper-object contact for a successful grasping.

It was also interesting to analyse the reasons behind the significant impact of the different configurations (i.e. object paths) on the results. As described before, the resulting trajectory of the end-effector depends on both the planned trajectory of the object and the chosen grasping pose. We observed that in the second configuration, and when the user was not receiving any guiding force cues, the resulting post-grasp trajectory of the robot was always passing very close to a kinematic singularity (thus, leading to a large joint norm velocity). This behaviour significantly changed when the haptic feedback was activated, since the force cues guided the operator towards a grasping pose that would result in a gripper trajectory much further away from singularities thanks to the minimisation of the proposed TOV index. However, this effect was not present in the third configuration, since in this case the 'intuitive'

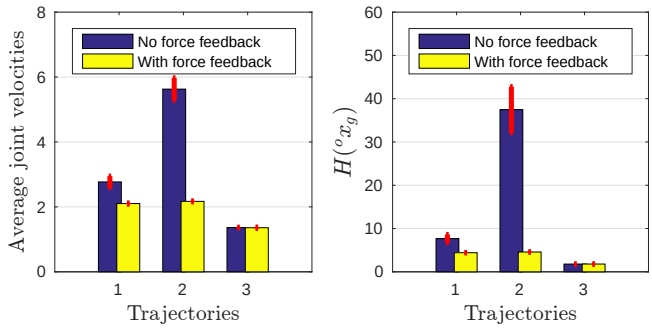


Figure 5.5: The results of the experiment with three different post-grasp trajectories over six trials. Left: the mean and variance of the average of joint velocities during post-grasp motions over six trials. Right: the mean and variance of the cost function $H\left({}^o\boldsymbol{x}_g\right)$ over six trials.

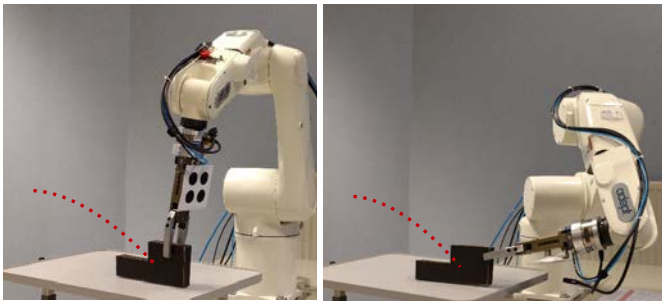


Figure 5.6: The object to be manipulated in the remote workspace and the given post-grasp trajectory (dotted red line). This figure shows the position from which the user grasped the object without haptic guidance (left) and with haptic guidance (right).

grasping pose chosen without any force guidance was already optimal w.r.t. the TOV index. Therefore, the impact of delivered force cues was not significant in this case.

5.6 Summary

In this chapter we presented an approach to assist a human operator in selecting a grasp pose while accounting for the post-grasp task. The force cues provided to the operator inform about the optimality of the current grasp pose w.r.t. an optimality index able to account for the kinematic capabilities of the slave arm in performing the needed manipulative task (a pick-and-place) after the grasping has been performed. This way, the operator can easily balance her/his preference between an (intuitively) stable grasp and an optimised trajectory for the slave arm during the subsequent pick-and-place task. Instead of using the classical manipulability

ellipsoid as a measure, we propose a task-relevant manipulability measure which accounts explicitly for the manipulative capabilities of the robot along the particular directions dictated by the post-grasp trajectory. Several experiments have been run in order to demonstrate the effectiveness of our approach in a real example of pick-and-place task with a 6-dof serial manipulator. The obtained results showed that the proposed architecture significantly reduced the control effort needed by the manipulator while assuring the fulfilment of the desired manipulative task.

Part III

Shared Control and Balancing of Humanoid Robots

Chapter 6

Humanoid Teleoperation

Contents

6.1 State of the Art **166**
6.2 Background **168**
6.2.1 Dynamic Model 168
6.2.2 Underlying Compliance Controller 169
6.3 Shared-Control Architecture **171**
6.3.1 Whole-Body Motion 171
6.3.2 Master Side 172
6.3.3 Balancing and Physical Constraints 173
6.3.4 CoM Control Law 175
6.3.5 Haptic Feedback 176
6.4 Experiments and Results **177**
6.4.1 Experiment I 178
6.4.2 Experiment II 179
6.4.3 Experiment III 179
6.5 Summary **183**

The work presented in part II focused on single and dual-arm serial manipulators. It proposed different shared-control strategies to allow a human operator to control these kinds of systems in a simple yet effective way. However, while fixed-base manipulators are very efficient in controlled environments, the need for a mobile robot capable of supporting and replacing humans in dangerous and physically demanding tasks is also crucial. There is a need for a robot that is able to navigate an environment designed for humans, e.g. climb the stairs or open doors, operate tools or interact with interfaces designed for a human operator. Humanoids have been proposed as a promising solution in this

regard but are still far from demonstrating their full potential. As an attempt to improve the usability and effectiveness of humanoid robots in complex unstructured tasks, we hereby apply and extend the previous approaches to the shared control of a humanoid robot.

The work presented in this chapter was conducted during the author's six-month visit to the German Aerospace Center (DLR), Oberpfaffenhofen, and was tested on DLR's hardware. It is published in [33] and was a finalist for the best paper award on safety, security and rescue robotics in the International Conference on Intelligent Robots and Systems (IROS 2018) in Madrid. An illustrative video of the performed experiments is available under https://youtu.be/PT2Jx6ULIHQ.

The rest of the chapter is divided as follows: Sec. 6.1 gives a brief introduction to the state of the art in humanoid teleoperation, Sec. 6.2 describes briefly the modeling of the humanoid and the employed compliance controller, Sec. 6.3 illustrates the proposed shared-control architecture commanding the slave's motion and the haptic interface, Sec. 6.4 comments on the experiments and results, and finally, Sec. 6.5 concludes the chapter.

6.1 State of the Art

Teleoperated humanoids have been studied for applications ranging from space manipulation [237] to substituting humans in nuclear environments [238], driving a lift truck [239] or piloting an aircraft [240]. A major challenge in this regard is designing a convenient hardware-software human machine interface (HMI) capable of controlling such highly versatile robots [241, 242]. The complexity of designing a teleoperation architecture for a humanoid lies in simultaneously giving the user an immersive task-oriented experience while informing her/him at the same time about the robot's balance and physical constraints. Task-relevant balance feedback is actually a topic that has not been studied thoroughly in literature, and is at the core of the work at hand. Most of the literature either provides the operator with full immersion in the task itself through haptic feedback from interactions with the environment, or gives her/him control over the posture of the humanoid while maintaining its balance. The need for bridging this gap between the robot's balance and the task at hand is indeed essential.

The potentiality of using a humanoid robot as a replacement for humans in a disaster scenario was lately demonstrated in the DARPA Robotics Challenge (DRC), where robots where tested for simple tasks such as driving a car, opening doors, walking on rough terrains, and using simple tools. In [243], Zucker et al. present the teleoperation system used for controlling the DRC-HUBO humanoid robot at trials. In this work, three operators were in charge of fully controlling the

humanoid through a software interface for navigation and manipulation purposes. One of the operators was explicitly responsible of ensuring stability and avoiding collisions by commanding the robot through that interface.

A different approach is described in [244], where a single human operator commands a humanoid by acting on a 3 DoF master device. The operator selects a certain point of the robot's body for manipulation instead of simultaneously controlling all its DoF. An autonomous controller integrates the operator's commands for producing the required whole-body motion while respecting postural stability constraints. In [245], the operator is given control over the robot's posture via a Phantom Omni master device, over which she/he receives haptic cues, reflecting sensory data from the load cells installed at the robot's feet. The user's actions are logged and used to teach the robot how to keep balance using a learning-from-demonstration framework. A bimanual master interface using two 6 DoF master devices to give the operator command over the humanoid's legs was proposed in [246]. The operator was fed back with force cues informing her/him about the position of the Zero-Moment Point (ZMP) within the support polygon. The same measure was used to give the operator a sense of the robot's balance using a vibrotactile belt for providing cutaneous haptic balance feedback [247]. This cutaneous belt was replaced with a kinesthetic system applying forces on the operator's waist to inform her/him about their proximity to the edges of the support polygon in [248].

The focus of the described literature is either on the manipulation task itself or on the balance/stability of the robot. In the former case, the operator is given command over the DoF of interest while an autonomous algorithm generates the required whole-body motion and enforces stability constraints. She/he is fed with haptic information related to the task at hand, e.g., interactions with the environment, without any insight on the balance of the robot. In the latter case, the user is given the task of maintaining the robot's balance by acting on a specific HMI to control the lower body (posture, legs, ...) of the humanoid.

The work in [247] is one of few that tackles both aspects simultaneously. Here, the operator is commanding the hands of the humanoid while being informed with haptic cues of its stability through a vibrotactile belt. However, she/he is still not informed about the impact of her/his manipulative actions on stability itself. In fact, manipulation actions have a crucial impact on stabilizing/destabilizing the robot. The operator, who is given command over some of the robot's DoF, can be oblivious as to how these DoF can be employed for improving the robot's balance. To this end, we propose in this chapter a balance-feedback human machine interface that closes the gap between the manipulative actions and their impact on the stability of a humanoid. A haptic interface establishes the direct association between the two by providing the operator with cues informing her/him of the impact of one

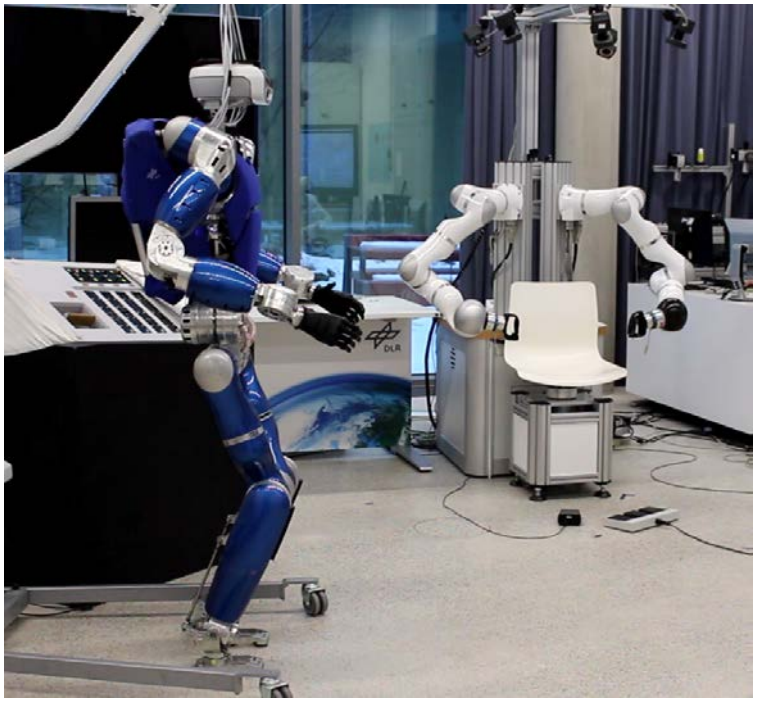


Figure 6.1: The Experimental Setup. The haptic user interface system HUG (right) is used as master device, while the torque-controlled humanoid robot TORO (left) works as slave device.

on the other. These cues are meant to provide the user with potential solutions to assure a successful task completion while accounting for the different constraints of the system.

On the other hand, the null space of the operator's commands is also employed to maintain the balance of the robot through an autonomous controller acting on an underlying impedance control architecture previously presented in [249]. In literature, a large variety of balancing controllers can be found, most of them based on inverse dynamics [250, 251, 252]. The approach presented in [249] is passivity-based and allows to stabilize the Center of Mass and the end-effectors that are not used for supporting the robot (the hands, in this context) in Cartesian space with respect to the world. This controller is then suitable as a tool for the teleoperation approach presented in this work.

6.2 Background

6.2.1 Dynamic Model

In legged humanoid robotics the use of dynamic models with a free-floating base is widespread because they feature a higher flexibility regarding contact changes

compared to dynamic models with a fixed base. In general, a central body within the kinematic structure of the robot is chosen as a base link, such as the hip or the trunk. Some works also utilize the center of mass (CoM) as a base, since it represents an essential quantity for balancing. Here, we will follow the proposition of [249] by defining a CoM frame $\mathcal{C}$, which is located at the CoM and has the same orientation of the hip. Let $\boldsymbol{p}_c \in \mathbb{R}^3$ and $\boldsymbol{R}_c \in \mathcal{SO}(3)$ denote the position and orientation of the frame $\mathcal{C}$ with respect to the world frame $\mathcal{W}$. The corresponding translational and rotational velocities are $\dot{\boldsymbol{p}}_c$ and $\boldsymbol{\omega}_c$, respectively. Based on the n joint angles $\boldsymbol{q} \in \mathbb{R}^n$ and $\boldsymbol{v}_c = (\dot{\boldsymbol{p}}_c^T, \boldsymbol{\omega}_c^T)^T$, the dynamics of the humanoid robot is given by

$$\boldsymbol{M} \underbrace{\begin{pmatrix} \dot{\boldsymbol{v}}_c \\ \ddot{\boldsymbol{q}} \end{pmatrix}}_{\dot{\boldsymbol{\nu}}} + \boldsymbol{C} \underbrace{\begin{pmatrix} \boldsymbol{v}_c \\ \dot{\boldsymbol{q}} \end{pmatrix}}_{\boldsymbol{\nu}} + \underbrace{\begin{pmatrix} m\boldsymbol{g}_0 \\ \boldsymbol{0} \end{pmatrix}}_{\boldsymbol{g}} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{\tau} \end{pmatrix} + \boldsymbol{\tau}_{\text{ext}}. \tag{6.1}$$

Herein, $\boldsymbol{M} \in \mathbb{R}^{(6+n)\times(6+n)}$ and $\boldsymbol{C} \in \mathbb{R}^{(6+n)\times(6+n)}$ denote the inertia and Coriolis/-centrifugal matrix, respectively. The gravitational torques are given by $\boldsymbol{g} \in \mathbb{R}^{6+n}$ with m denoting the overall mass of the robot and $\boldsymbol{g}_0 \in \mathbb{R}^6$ the gravitational acceleration[1]. The joint torques are given by $\boldsymbol{\tau} \in \mathbb{R}^n$. The influence of external wrenches acting on the robot is taken into account by the generalized torque vector $\boldsymbol{\tau}_{\text{ext}} \in \mathbb{R}^{6+n}$.

Let us divide the Ψ end-effectors into two subgroups [249]: The first one is referred to as "balancing end-effector" (*bal*) and contains the ψ end-effectors that are used by the robot to support itself (usually the feet). The remaining end effectors are called "interaction end-effectors" (*int*), as they are still free to be used in a manipulation or interaction task (usually the hands). Based on this definition, the Cartesian velocities of the end effectors $\boldsymbol{v} \in \mathbb{R}^{6\Psi}$ are given by

$$\boldsymbol{v} = \begin{pmatrix} \boldsymbol{v}_{\text{bal}} \\ \boldsymbol{v}_{\text{int}} \end{pmatrix} = \begin{bmatrix} \boldsymbol{J}_{\text{bal}} \\ \boldsymbol{J}_{\text{int}} \end{bmatrix} \boldsymbol{\nu} = \boldsymbol{J}\boldsymbol{\nu}. \tag{6.2}$$

with the Jacobian matrix $\boldsymbol{J} \in \mathbb{R}^{6\Psi\times(6+n)}$ and $\boldsymbol{v}_{\text{bal}} \in \mathbb{R}^{6\psi}$, $\boldsymbol{v}_{\text{int}} \in \mathbb{R}^{6(\Psi-\psi)}$, $\boldsymbol{J}_{\text{bal}} \in \mathbb{R}^{6\psi\times(6+n)}$, $\boldsymbol{J}_{\text{int}} \in \mathbb{R}^{6(\Psi-\psi)\times(6+n)}$. In the case where all external disturbances act solely at the end-effectors, $\boldsymbol{\tau}_{\text{ext}}$ simplifies to

$$\boldsymbol{\tau}_{\text{ext}} = \boldsymbol{J}^T \boldsymbol{F}_{\text{ext}} \tag{6.3}$$

6.2.2 Underlying Compliance Controller

This section gives a brief recapitulation of the balancing controller presented in [249]. The controller stabilizes the CoM by a Cartesian compliance, which applies

[1] Note that $\boldsymbol{g}_0$ is six-dimensional by containing also the rotational DoF.

a wrench $\boldsymbol{F}_c \in \mathbb{R}^6$ at the CoM frame $\mathcal{C}$. Each one of the interaction end-effectors is stabilized by another Cartesian compliance, with the resulting wrenches stacked into $\boldsymbol{F}_{\text{int}} \in \mathbb{R}^{6(\Psi-\psi)}$. In order to support the robot, the control algorithm computes a suitable set of balancing wrenches $\boldsymbol{F}_{\text{bal}} \in \mathbb{R}^{6\Psi}$ by solving the following quadratic optimization problem

$$\boldsymbol{F}_{\text{bal}}^{\text{opt}} = \underset{\boldsymbol{F}_{\text{bal}}}{\text{argmin}} \left(\boldsymbol{F}_{\text{bal}} - \boldsymbol{F}_{\text{bal}}^{\text{def}}\right)^T \boldsymbol{Q} \left(\boldsymbol{F}_{\text{bal}} - \boldsymbol{F}_{\text{bal}}^{\text{def}}\right) \tag{6.4}$$

with respect to

$$\boldsymbol{Ad}_{\text{bal}}^T \boldsymbol{F}_{\text{bal}} + \boldsymbol{Ad}_{\text{int}}^T \boldsymbol{F}_{\text{int}} = m\boldsymbol{g}_0 - \boldsymbol{F}_c \tag{6.5}$$

and

$$\begin{aligned} f_{k,z} &\geq f_{k,z}^{\min} \,, \\ \delta_{k,x/y}^{\min} \leq \delta_{k,x/y} &\leq \delta_{k,x/y}^{\max} \,, \\ \left|f_{k,x/y}\right| &\leq \tilde{\mu}_k f_{k,z} \,. \end{aligned} \tag{6.6}$$

The cost function (6.4) minimizes the deviation of $\boldsymbol{F}_{\text{bal}}$ from a default wrench distribution $\boldsymbol{F}_{\text{bal}}^{\text{def}}$ considering the positive definite weighting matrix $\boldsymbol{Q} \in \mathbb{R}^{6\Psi \times 6\Psi}$. The default distribution $\boldsymbol{F}_{\text{bal}}^{\text{def}}$ is a tuning parameter, which can be set to half of the weight of the robot if the stance is symmetric. The equality constraint (6.5) represents the underactuation of the base by demanding that the influence of all commanded end-effector wrenches ($\boldsymbol{F}_{\text{bal}}$, $\boldsymbol{F}_{\text{int}}$) on the CoM must sum up to the compliance wrench $\boldsymbol{F}_c$ plus gravity. For this, the Jacobian matrices $\boldsymbol{J}_{\text{bal}} = [\, \boldsymbol{Ad}_{\text{bal}} \; \bar{\boldsymbol{J}}_{\text{bal}} \,]$ and $\boldsymbol{J}_{\text{int}} = [\, \boldsymbol{Ad}_{\text{int}} \; \bar{\boldsymbol{J}}_{\text{int}} \,]$ are partitioned into $\boldsymbol{Ad}_{\text{bal}} \in \mathbb{R}^{6\psi \times 6}$, $\boldsymbol{Ad}_{\text{int}} \in \mathbb{R}^{6(\Psi-\psi)\times 6}$, $\bar{\boldsymbol{J}}_{\text{bal}} \in \mathbb{R}^{6\psi \times n}$, and $\bar{\boldsymbol{J}}_{\text{int}} \in \mathbb{R}^{6(\Psi-\psi)\times n}$. The first two are the stacked adjoint matrices of each end-effector, and relate a motion of the CoM frame $\mathcal{C}$ with a motion of the end-effectors. The matrices $\bar{\boldsymbol{J}}_{\text{bal}}$ and $\bar{\boldsymbol{J}}_{\text{int}}$ describe the influence of a joint motion on the end-effectors. The inequality constraints (6.6) represent the contact model to which $\boldsymbol{F}_{\text{bal}}$ is subjected to in order to account for unilaterality, the position of the Center of Pressure (CoP), and for friction of the balancing contacts. For each wrench within $\boldsymbol{F}_{\text{bal}}$, the force perpendicular to the contact surface $\mathcal{S}$ is bounded from below by the minimum contact force $f_{k,z}^{\min}$ in order to prevent the end-effector from lifting off[2]. Slippage is prevented by constraining the tangential forces $f_{k,x/y}$ to the friction cone given by $\tilde{\mu}_k$. The CoP δ_k is constrained to the interior of the contact surface via $\delta_{k,x/y}^{\min/\max}$ to prevent the end-effector from tilting. From the contact model (6.6) and a given contact configuration, one can compute an equivalent support polygon, as shown in [253]. In order to achieve a static and stable balancing, the CoM projected to the ground floor must stay within the support polygon as well. This allows us to use the support polygon as a stability criterion in Sec. 6.3.3.

[2]For the conducted experiments (see Sec. 6.4), $f_{k,z}^{\min}$ was set to 50 N in order to account for joint friction.

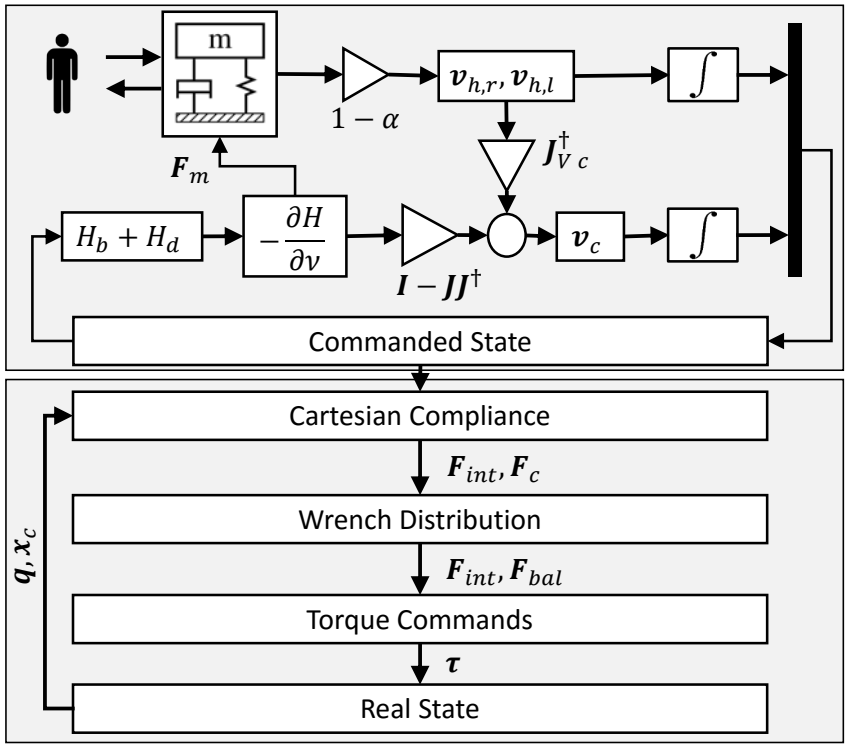


Figure 6.2: A scheme of the proposed shared-control framework. The lower block represents the underlying compliance controller, while the upper block is the high-level teleoperation framework.

After computing a suitable wrench distribution $\boldsymbol{F}_{\text{bal}}^{\text{opt}}$, the end effector wrenches are mapped to joint space via

$$\boldsymbol{\tau} = -\begin{bmatrix} \bar{\boldsymbol{J}}_{\text{bal}}^{T} & \bar{\boldsymbol{J}}_{\text{int}}^{T} \end{bmatrix} \begin{pmatrix} \boldsymbol{F}_{\text{bal}}^{\text{opt}} \\ \boldsymbol{F}_{\text{int}} \end{pmatrix}. \tag{6.7}$$

6.3 Shared-Control Architecture

6.3.1 Whole-Body Motion

While it is habitual to have the high-level 'manipulation' controller incorporated into the compliance controller itself, a different approach is opted for in this work. Indeed, the compliance controller described in the previous section is treated as a 'packaged' controller and topped with the needed higher level position/velocity whole-body motion controller, as shown in Fig. 6.2. This approach is remunerative as it allows the user to design a controller for a force/torque compliance-controlled robot from a pre-designed and tuned compliance controller without having to deal with its specificities.

Going back to Fig. 6.2, the shared-control teleoperation architecture is the upper gray zone. The user acts on an HMI sending velocity commands $(\boldsymbol{v}_{m,r}, \boldsymbol{v}_{m,l})$ through the right and left arms of the master device while receiving haptic cues $(\boldsymbol{F}_{m,r}, \boldsymbol{F}_{m,l})$ from potentials informing about the proximity to the edge of the support polygon and other physical constraints of the system. As the user gets closer to hitting these constraints, her/his commands are gradually scaled down to zero in order to ensure the stability of the system. On the other hand, the mentioned potentials are also used to generate a CoM motion $\boldsymbol{v}_c$ in the null-space of the motion of the hands, which are commanded by the user, and the feet (stable on the ground) in order to ensure that the system remains as far as possible from the constraints. The resulting velocities are integrated to generate new positions, which are then fed to the underlying compliance controller.

6.3.2 Master Side

Similarly to the previous chapters, and following the classical bilateral force-feedback teleoperation framework, we assume the presence of a dual-arm master device through which the user sends velocity commands and receives force-feedback. The two master arms are modeled as generic (gravity pre-compensated) mechanical systems

$$\boldsymbol{M}_m(\boldsymbol{x}_m) \begin{bmatrix} \dot{\boldsymbol{v}}_{m,r} \\ \dot{\boldsymbol{v}}_{m,l} \end{bmatrix} + C_m(\boldsymbol{x}_m, \boldsymbol{v}_{m,r}, \boldsymbol{v}_{m,l}) = \begin{bmatrix} \boldsymbol{F}_{m,r} \\ \boldsymbol{F}_{m,l} \end{bmatrix} + \boldsymbol{F}_h \tag{6.8}$$

where $\boldsymbol{x}_m \in \mathbb{R}^{3+3} \times \mathbb{S}^{3+3}$ is the device cartesian configuration vector containing the pose of the end-effectors of the right and left master arms, $\boldsymbol{v}_{m,r}, \boldsymbol{v}_{m,l} \in \mathbb{R}^6$ are the linear and angular velocities of each of the right and left master end-effectors respectively, $\boldsymbol{M}_m(\boldsymbol{x}_m) \in \mathbb{R}^{12\times12}$ is the positive-definite symmetric mass matrix, $C_m(\boldsymbol{x}_m, \boldsymbol{v}_{m,r}, \boldsymbol{v}_{m,l}) \in \mathbb{R}^{12\times12}$ accounts for Coriolis/centrifugal terms, $\boldsymbol{F}_h \in \mathbb{R}^{12}$ account for the forces applied by the human operator, and $\boldsymbol{F}_{m,r}, \boldsymbol{F}_{m,l} \in \mathbb{R}^6$ are the control forces on the right and the left arm, respectively.

The operator is given control over the hands of the robot through a direct Cartesian coupling with the end effectors of the master arms. As the workspace of the human operator and the master arms is close to that of the humanoid, no scaling is implemented. However, the user can always "clutch" to move the master arms to a more convenient position without moving the slave. A velocity-velocity master-slave coupling is employed such that

$$\begin{cases} \boldsymbol{v}_{h,r} = \tilde{\boldsymbol{R}}_k \ \boldsymbol{v}_{m,r} \\ \boldsymbol{v}_{h,l} = \tilde{\boldsymbol{R}}_k \ \boldsymbol{v}_{m,l} \end{cases} \tag{6.9}$$

where $\boldsymbol{v}_{h,r}$ and $\boldsymbol{v}_{h,l} \in \mathbb{R}^6$ are the cartesian velocity commands of the right and left hands of the slave respectively, and $\tilde{\boldsymbol{R}}_k$ is the rotation map between the master and the slave velocities.

6.3.3 Balancing and Physical Constraints

In addition to executing the operator's commands, it is crucial to maintain the stability of the humanoid by ensuring its CoM remains within the support polygon[3] (see Fig. 6.3). To this end, we define a cost function $H_b : \mathbb{R}^n \mapsto \mathbb{R}$ as a measure of the 'balance' of the humanoid, which goes to infinity as the user approaches the edges of the support polygon such that

$$H_b(\boldsymbol{p}_c) = \sum_i h_{i,b}(\boldsymbol{p}_c), \tag{6.10}$$

where $\boldsymbol{p}_c \in \mathbb{R}^3$ is the position of the center of mass, and $h_{i,b}$ is the cost function attached to each edge i of the support polygon and defined by

$$h_{i,b}(\boldsymbol{p}_c) = \begin{cases} \mu \; tan\left(\frac{\pi}{2}\left(1 - \frac{d_i}{d_{l,i}}\right)\right)^2 & d_i < d_{l,i}, \\ 0 & otherwise \end{cases} \tag{6.11}$$

with μ being a regulation gain ($\mu = 1$ in the performed experiments), d_i is the horizontal distance from the center of mass to each edge i of the support polygon (which is known from the fixed posture of the feet), and $d_{l,i}$ is the distance from the center of the support polygon to its i-th edge (see Fig. 6.3).

The time derivative of H_b can then be defined by

$$\dot{H}_b = \frac{\partial H_b}{\partial \boldsymbol{p}_c}\dot{\boldsymbol{p}}_c = \frac{\partial H_b}{\partial \boldsymbol{p}_c}\boldsymbol{S}_{\boldsymbol{p}_c}\boldsymbol{\nu}. \tag{6.12}$$

where $\boldsymbol{S}_{\boldsymbol{p}_c} = [\boldsymbol{I} \; \boldsymbol{0}] \in \mathbb{R}^{3\times 6+n}$ is a selection matrix extracting $\dot{\boldsymbol{p}}_c$ from the state velocity vector $\boldsymbol{\nu}$.

While the physical constraints of a humanoid robot range from workspace limits to joint limits, singularities and torque constraints, a representative workspace constraint was considered in this work consisting of the distance between the hands of the robot and its center of mass. Constraining this distance ensures that the robot maintains a "safe" posture and avoids a "full stretch" configuration that could push the robot to singularities and joint limits. While this constraint does not cover the various limitations of a humanoid, it is meant as a proof of concept and a representative constraint. The exploration of further constraints is a future work.

[3]Note that this condition is only valid for evaluating static stability. In the dynamic case, a condition could be used based e.g. on the capture point [254].

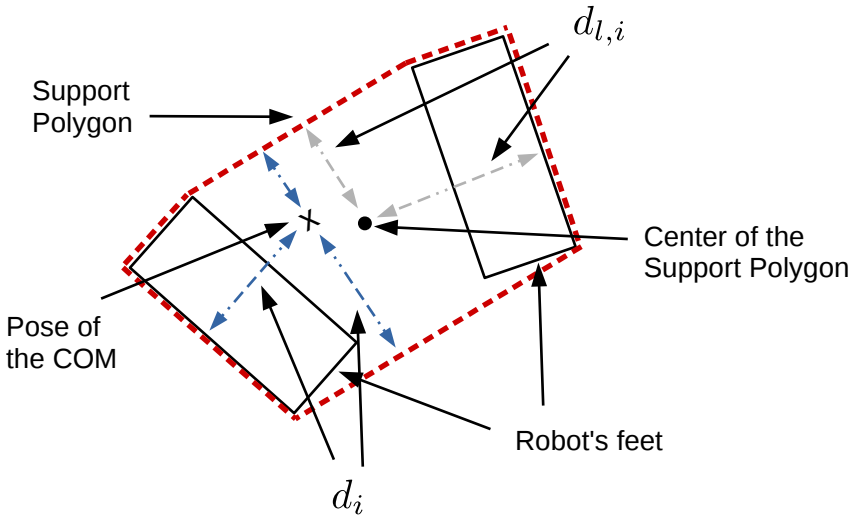


Figure 6.3: An illustration of the support polygon for a robot standing on horizontal ground.

The constraint hereby considered is then to maintain the distance between each of the hands of the humanoid and its CoM within a predefined threshold d_{th}. The corresponding cost function H_d accounting for this physical constraint is defined as

$$H_d\left(\boldsymbol{p}_c, \boldsymbol{p}_{h,r}, \boldsymbol{p}_{h,l}\right) = h_{c,r}\left(\boldsymbol{p}_c, \boldsymbol{p}_{h,r}\right) + h_{c,l}\left(\boldsymbol{p}_c, \boldsymbol{p}_{h,l}\right) \tag{6.13}$$

where the cost function of each hand, $h_{c,x}\left(\boldsymbol{p}_c, \boldsymbol{p}_{h,x}\right)$, is

$$h_{c,x} = \begin{cases} \rho\ tan\left(\frac{\pi}{2}\left(1 - \frac{d_{h,x} - d_s}{d_{th} - d_s}\right)\right)^2 & d_s < d_i < d_l, \\ 0 & otherwise \end{cases}. \tag{6.14}$$

$d_{h,x}$ in the above equation is the distance from each hand x to the center of mass, while d_s is the minimum distance after which the potential starts, and ρ is a positive gain ($\rho = 1$ in the performed experiments). The derivative of the potential H_d is then

$$\begin{aligned} \dot{H}_d &= \begin{bmatrix} \frac{\partial H_d}{\partial \boldsymbol{p}_c} & \frac{\partial H_d}{\partial \boldsymbol{p}_{h,r}} & \frac{\partial H_d}{\partial \boldsymbol{p}_{h,l}} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{p}}_c \\ \dot{\boldsymbol{p}}_{h,r} \\ \dot{\boldsymbol{p}}_{h,l} \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial H_d}{\partial \boldsymbol{p}_c} & \frac{\partial H_d}{\partial \boldsymbol{p}_{h,r}} & \frac{\partial H_d}{\partial \boldsymbol{p}_{h,l}} \end{bmatrix} \begin{bmatrix} \boldsymbol{S}_{\boldsymbol{p}_c} \\ \boldsymbol{J}_{\boldsymbol{p}_{h,r}} \\ \boldsymbol{J}_{\boldsymbol{p}_{h,l}} \end{bmatrix} \boldsymbol{\nu}, \end{aligned} \tag{6.15}$$

where $\boldsymbol{J}_{\boldsymbol{p}_{h,r}}, \boldsymbol{J}_{\boldsymbol{p}_{h,l}} \in \mathbb{R}^{3\times 6+n}$ are the linear part of the geometric jacobians linking the cartesian velocities of the right and left hand to the state velocity $\boldsymbol{\nu}$, respectively.

The desired state velocity $\boldsymbol{\nu}_{des}$ which minimizes the potentials and makes sure the system constraints are respected can then be defined as the negative of the transpose of the gradient of potentials H_b and H_d (see (6.12) and (6.15)) such that

$$\boldsymbol{\nu}_{des} = -\left(\begin{bmatrix} \frac{\partial H_d}{\partial \boldsymbol{p}_c} & \frac{\partial H_d}{\partial \boldsymbol{p}_{h,r}} & \frac{\partial H_d}{\partial \boldsymbol{p}_{h,l}} \end{bmatrix} \begin{bmatrix} \boldsymbol{S}_{\boldsymbol{p}_c} \\ \boldsymbol{J}_{\boldsymbol{p}_{h,r}} \\ \boldsymbol{J}_{\boldsymbol{p}_{h,l}} \end{bmatrix} \right)^T - \left(\frac{\partial H_b}{\partial \boldsymbol{p}_c} \boldsymbol{S}_{\boldsymbol{p}_c} \right)^T \tag{6.16}$$

6.3.4 CoM Control Law

The motion of the center of mass is divided into two hierarchical subtasks. The primary task consists of following the commands of the human operator while fixing the feet of the robot on the ground. The secondary task, executed in the nullspace of the first, is to maintain the balance of the robot keeping its CoM at the center of the support polygon. We commence now by explaining the autonomous balancing algorithm governing this behavior.

The robot's state velocity vector is mapped to the velocities of the hands (interacting end-effectors) and the feet (balancing end-effectors) as described in Sec. 6.2.2 where

$$\begin{bmatrix} \boldsymbol{v}_{h,r} \\ \boldsymbol{v}_{h,l} \\ \boldsymbol{v}_{f,r} \\ \boldsymbol{v}_{f,l} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Ad}_{\text{int}} & \bar{\boldsymbol{J}}_{\text{int}} \\ \boldsymbol{Ad}_{\text{bal}} & \bar{\boldsymbol{J}}_{\text{bal}} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_c \\ \dot{\boldsymbol{q}} \end{bmatrix}. \tag{6.17}$$

The impact of the velocity inputs from the human operator on the velocity of the CoM, dubbed as its primary task, can then be retrieved by inverting the previous equation such that

$$\boldsymbol{v}_c = \boldsymbol{Ad}^{\dagger} \begin{bmatrix} \boldsymbol{v}_{h,r} \\ \boldsymbol{v}_{h,l} \\ \boldsymbol{v}_{f,r} \\ \boldsymbol{v}_{f,l} \end{bmatrix} \tag{6.18}$$

with $\boldsymbol{Ad}^{\dagger}$ being the Moore pseudo-inverse of $\boldsymbol{Ad} = \begin{bmatrix} \boldsymbol{Ad}_{\text{int}}^T & \boldsymbol{Ad}_{\text{bal}}^T \end{bmatrix}^T$, which is the upper part of Jacobian $\boldsymbol{J}$ corresponding to the velocity of the center of mass $\boldsymbol{v}_c$. The nullspace balancing motion is then added to (6.18), and the resulting full control law becomes

$$\boldsymbol{v}_c = \boldsymbol{Ad}^{\dagger} \begin{bmatrix} \boldsymbol{v}_{h,r} \\ \boldsymbol{v}_{h,l} \\ \boldsymbol{v}_{f,r} \\ \boldsymbol{v}_{f,l} \end{bmatrix} + \boldsymbol{N} \begin{bmatrix} \boldsymbol{S}_{\boldsymbol{p}_c} \boldsymbol{\nu}_{des} \\ \boldsymbol{0} \end{bmatrix} \tag{6.19}$$

where $\boldsymbol{N} = (\boldsymbol{I} - \boldsymbol{Ad}\ \boldsymbol{Ad}^{\dagger})$ is the null-space projector.

6.3.5 Haptic Feedback

In case of proximity to a constraint, force-feedback is provided to the operator to guide her/him away from the undesired configuration. Moreover, if the operator keeps pushing towards the constraint, her/his velocity commands are tuned down to prevent the system from hitting the constraint.

The motion direction of the hands, $[\tilde{\boldsymbol{v}}_{h,r},\ \tilde{\boldsymbol{v}}_{h,l}]$, which ensure the minimization of the potentials H_b and H_d can be retrieved as

$$\begin{bmatrix} \tilde{\boldsymbol{v}}_{h,r} \\ \tilde{\boldsymbol{v}}_{h,l} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{J}_{h,r} \\ \boldsymbol{J}_{h,l} \end{bmatrix} \boldsymbol{N}_{bal} \boldsymbol{\nu}_{des}, \tag{6.20}$$

where $\boldsymbol{N}_{bal} = (\boldsymbol{I} - \boldsymbol{J}_{bal} \boldsymbol{J}_{bal}^{\dagger})$ is the null-space projector ensuring that the generated motion does not impact the position of the feet.

However, while the constraint potentials H_b and H_d are always active, the user is to receive feedback only when in the proximity of a constraint. To this end, we define a saturation function, α where $\alpha : 0 \mapsto 1$ as the total cost function $H : H_0 \mapsto H_{\max}$. Here H is defined as $H = H_b + H_c$ whereas H_0 and $H_{\max}$ are pre-defined thresholds. The input velocities of the human operator are then tuned down if the robot is in the proximity of a constraint, i.e. $\alpha > 0$, and the direction commanded by the operator is opposing the desired motion directions $[\tilde{\boldsymbol{v}}_{h,r},\ \tilde{\boldsymbol{v}}_{h,l}]$

$$\begin{cases} \boldsymbol{v}_{h,r} = (1-\alpha) \tilde{\boldsymbol{R}}_k\ \boldsymbol{v}_{m,r} & if\ \tilde{\boldsymbol{v}}_{h,r}^T \tilde{\boldsymbol{R}}_k\ \boldsymbol{v}_{m,r} < 0 \\ \boldsymbol{v}_{h,l} = (1-\alpha) \tilde{\boldsymbol{R}}_k\ \boldsymbol{v}_{m,l} & if\ \tilde{\boldsymbol{v}}_{h,l}^T \tilde{\boldsymbol{R}}_k\ \boldsymbol{v}_{m,l} < 0 \end{cases}. \tag{6.21}$$

The forces received by the operator on the right and left arms are also defined in function of α such that

$$\begin{cases} \boldsymbol{F}_{m,r} = F_{max}\ \alpha\ \tilde{\boldsymbol{v}}_{h,r} / ||\tilde{\boldsymbol{v}}_{h,r}|| + \boldsymbol{B}\ \boldsymbol{v}_{m,r} \\ \boldsymbol{F}_{m,l} = F_{max}\ \alpha\ \tilde{\boldsymbol{v}}_{h,l} / ||\tilde{\boldsymbol{v}}_{h,l}|| + \boldsymbol{B}\ \boldsymbol{v}_{m,l} \end{cases} \tag{6.22}$$

where F_{max} is a design parameter defining the maximum force expected from the operator, and $\boldsymbol{B} \in \mathbb{R}^{6\times6}$ is a damping matrix.

This proposed design of the haptic interface follows the same spirit of the previous chapters in that it is not a pure resistive force stopping the user as she/he approaches a constraint or another system limit. On the contrary, the described force cues represent an active guidance providing the user with several solutions for avoiding the system constraints over the 12 DoF she/he is commanding.

Note that the behavior of the force feedback in (6.22) is akin to a spring-damper system, as α is directly linked to the distances from the CoM to the support polygon boundaries and to the hands of the robot. While the potentials H_b and H_d do go to infinity in the proximity of the edges of the support polygon, the haptic feedback given to the operator adopts the direction of the gradient of these potentials (but not its magnitude). While this is not needed for passivity, as the potential can be proven passive in itself, it allows for a better design of the haptic interface, since any haptic device has limited capabilities for force generation and the design of the haptic interface must account for these limitations.

6.4 Experiments and Results

This section describes the experiments conducted to validate the described approach, and discusses the obtained results.

the employed experimental setup uses HUG on the master side. HUG is a haptic user interface system composed by two light-weight robotic arms with a workspace of 1.1 m and a nominal payload of 20 kg each [255]. On the slave side, we use TORO, a humanoid robot developed at DLR [256]. It has a height of 1.74 m, a weight of 76.4 kg, and 25 joints in total (not counting the neck and the hands). In the presented experiments, the feet of TORO are in contact with the ground floor in order to support the robot (balancing end-effectors). The hands (interaction end-effectors) were directly commanded by the operator via HUG. The joints in the neck and hands were not used. Due to the feet contacts ($2 \times 6 = 12$ DoF) and the user input for the hands ($2 \times 6 = 12$ DoF), the robot can only move 7 DoF in order to maintain balance (6 DoF attached to the floating-base (frame $\mathcal{C}$) plus 1 remaining DoF of the 25 joints).

In order to test the behavior of the different parts of the system, we distinguish two major components: the null-space autonomous balancer acting on the CoM to keep the system as far as possible from constraints, and the haptic guidance informing the user of the proximity to any of these constraints and the possible directions to avoid them. Three experiments were performed while activating or deactivating these two components, to asses their impact on the general behaviour of the system. A video available at https://youtu.be/PT2Jx6ULIHQ provides further insight on the performed experiments.

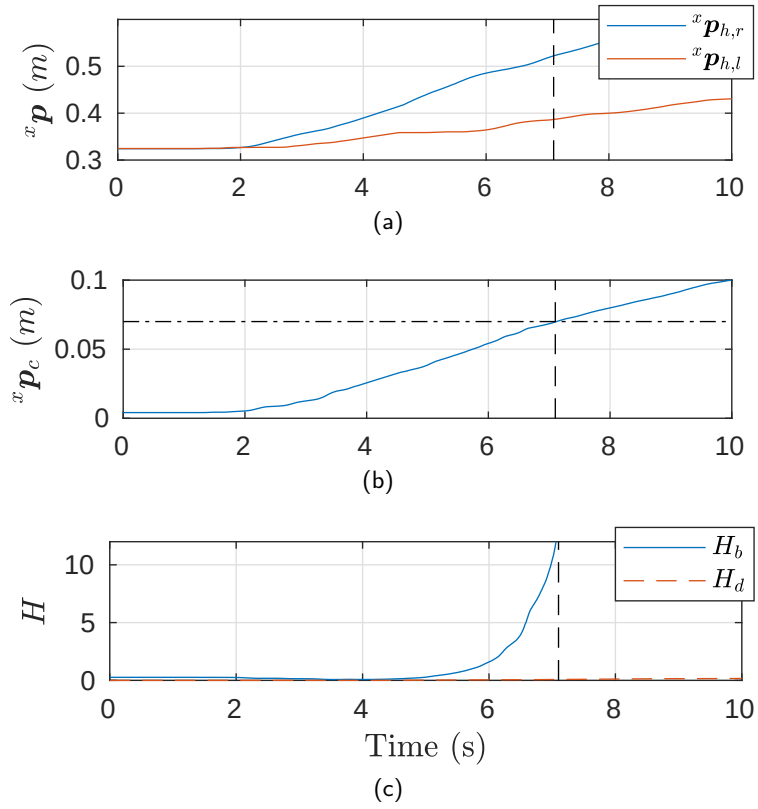


Figure 6.4: Experiment I: No haptic guidance is provided, and the null space balancer is deactivated. (a) and (b) show the position of the hands and CoM along the x-axis, respectively, while (c) shows the system potentials H_b and H_d.

6.4.1 Experiment I

In the first experiment, the haptic guidance and the null-space balancer were both deactivated, and the user was given unrestricted control over the hands of the humanoid.

The user was asked to reach as far as possible with the hands of the humanoid along the x-axis (which is pointing forward). Fig. 6.4 reports on the obtained results. Fig. 6.4a and Fig. 6.4b show the x-position of the hands and the CoM respectively throughout the experiment, while Fig. 6.4c shows the variation of the potentials H_b and H_d. Note that this is just a visualization of the potentials, as they were not active in this experiment. On the other hand, in Fig. 6.4b the horizontal dotted line represents the edge of the support polygon along the positive x-direction. The center of the support polygon was at 1 cm from the world frame in this scenario, with its edges at -3 cm and 7 cm along the x-axis.

Observing the figures, the maximum reach of the hands of the humanoid before hitting the support polygon's edge was 52.7 cm for the right arm (the vertical

dotted line denotes the moment at which the CoM crossed the edge of the support polygon). In fact, the humanoid then started to tip over, losing contact on its right foot, and the experiment was called to a halt. Note that the potential H_b increased significantly as the edge of the support polygon was closer.

The results of this experiment are also depicted in Fig. 6.7, where Fig. 6.7a shows the trajectories followed by the right hand (in blue) and the CoM (in red), and Fig. 6.7d depicts the final posture of the robot after the right foot lost grip.

6.4.2 Experiment II

Following on the previous experiment, haptic guidance was activated along with the restriction applied on the operator's commands when approaching a constraint (by exploiting the proximity measure α, check (6.21)). The null-space autonomous balancer was, however, still not active. The same experiment was repeated and the user was asked again to reach as far as possible. The results are reported in Fig. 6.5. Fig. 6.5b plots the forces fed to the operator (along the x-axis) on the end-effectors of the master device. On the other hand, Fig. 6.5a and Fig. 6.5c show the x-position of the slave hands (right and left) and the center of mass, respectively, as the experiment proceeds. Finally Fig. 6.5d depicts the two potential functions describing the stability of the robot and its physical constraints.

An interesting behavior is observed in this experiment. As the user moved the hands of the robot forward, the CoM started approaching the edge of the support polygon, and this was reflected as an increase in the cost functions and an active force on both hands along the negative x direction (see t=10 sec). The user then reactively started moving the left hand backwards, following the haptic guidance. This decreased the cost function, allowing him to further push the right hand forward. In fact, this strategy, inspired by the informative haptic guidance, helped the user to reach 34.9 cm farther than the first experiment (t=45 sec), without impacting the stability of the robot.

The followed trajectories and final posture of the robot for experiment II are reported in Fig. 6.7b and Fig. 6.7e.

6.4.3 Experiment III

The null-space autonomous balancer was activated for the third experiment, in addition to having the haptic guidance active as well. The experiment followed the same procedure as before, and the results are reported in Fig. 6.6.

An interesting figure to observe in this experiment is Fig. 6.6c, which shows the evolution of the x-position of the CoM. In contrast to the previous two experiments, the CoM remained close to the center of the support polygon for a significant portion

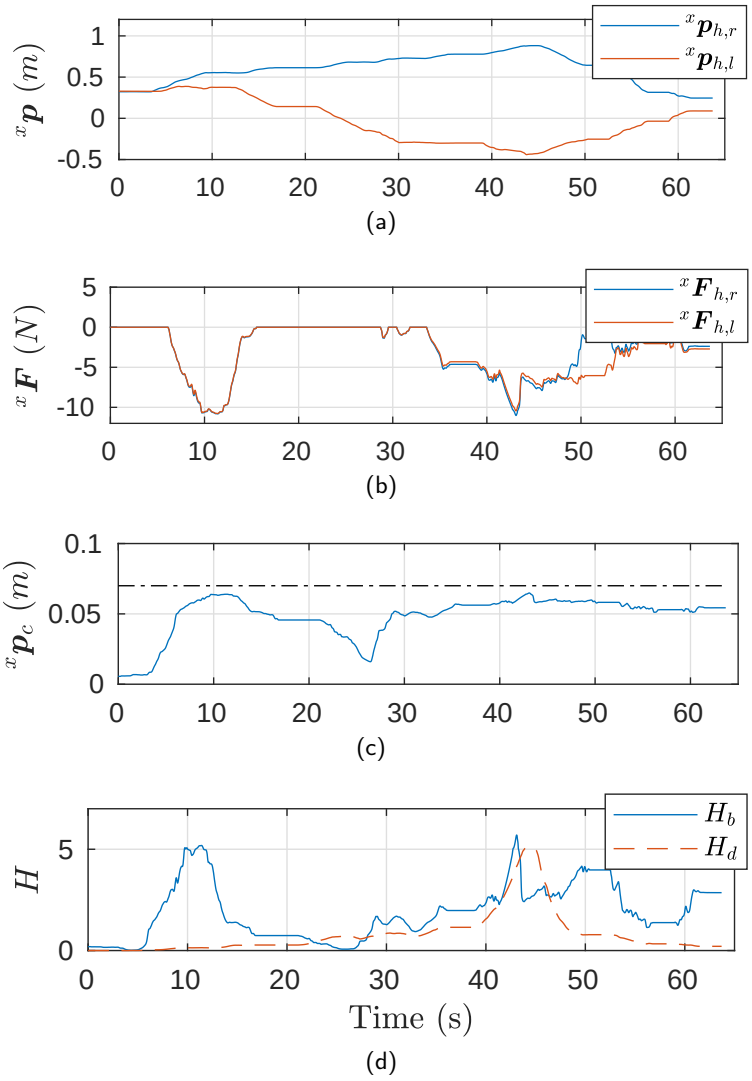


Figure 6.5: Experiment II: Haptic guidance is provided to the operator, but the null-space balancer is deactivated. (b) shows the force cues fed to the operator along the x-axis on the right and left hand, respectively, while (a) and (c) report on the x-position of the hands and the CoM. (d) plots the potentials H_b and H_d.

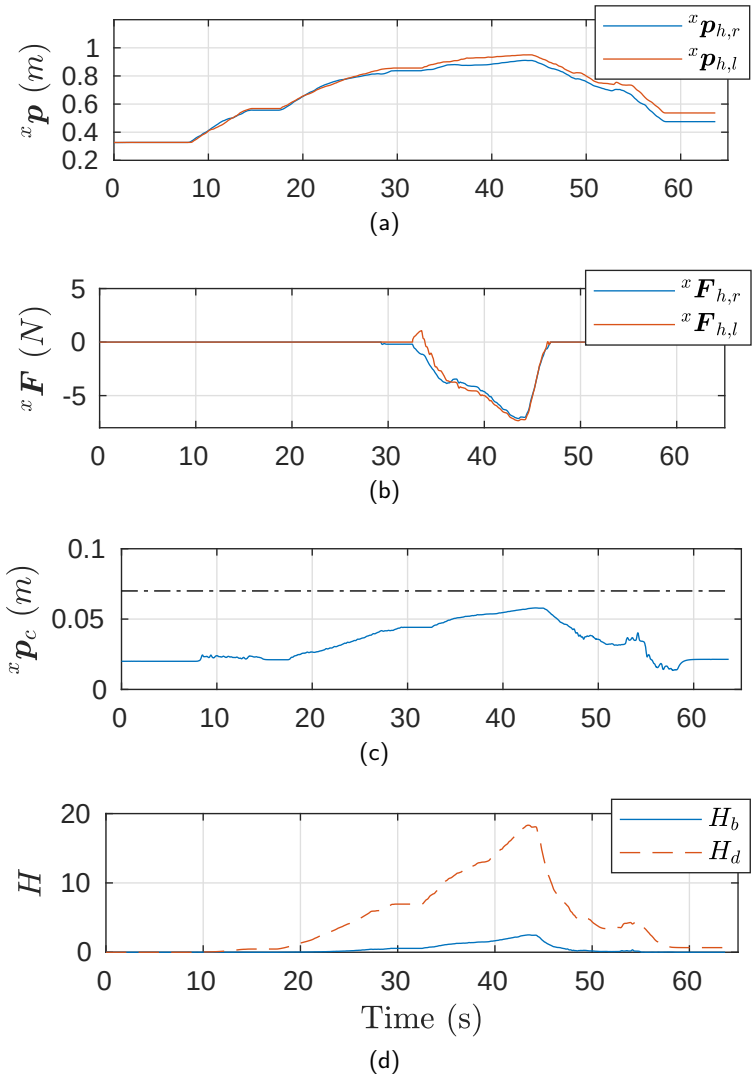


Figure 6.6: Experiment III: Haptic guidance is provided to the operator, and the null-space balancer is active. (b) shows the force cues fed to the operator along the x-axis on the right and left hand, respectively, while (a) and (c) show the x-position of the hands and the CoM. (d) depicts the potentials H_b and H_d.

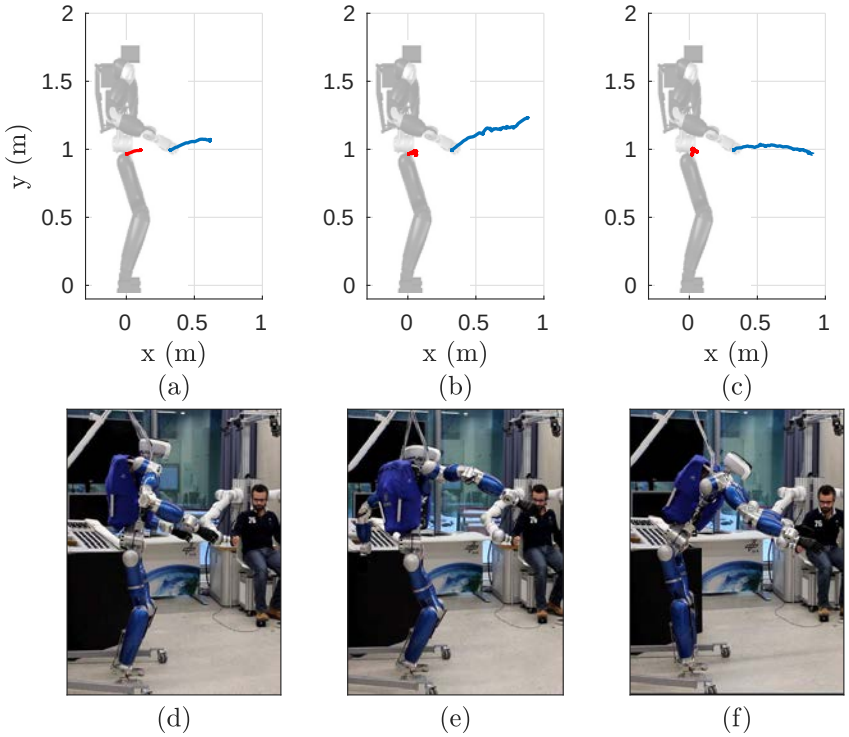


Figure 6.7: Comparison of the conducted experiments: The top row shows the trajectory of the right hand and the CoM in the saggital plane of the robot. The bottom row shows the final pose of the robot with the largest achievable reach of the hands. (a) and (d) report on experiment I, (b) and (e) on experiment II, and (c) and (f) on experiment III

of the experiment. In fact, both the right and left hands were close to a 60 cm reach before the CoM started moving forward. It was then pushed forward as the hands were being moved farther ahead, thus driving the robot to be more stretched and increasing the potential H_d, as can be observed in Fig. 6.6d. The resulting maximum reach was 91.4 cm, a bit higher than in experiment II. However, the user was able to push both hands forward instead of one, thus allowing for more manipulation options.

On the other hand, Fig. 6.6b shows that the forces received throughout this experiment were negligible, except for t=[33,48] s when the hands were stretched at more than 85 cm. This is important in that autonomy, while accounting for the operator's commands, was able to successfully command the null-space and provide the operator with the maximum workspace possible with minimal disturbance.

The followed trajectories and final posture of the robot in this experiment are reported in Fig. 6.7c, and Fig. 6.7f.

6.5 Summary

While fixed manipulators are very efficient in controlled environments, mobile robots, and specifically humanoids, do offer a much bigger potential. However, this comes at the cost of a more complex system and issues like stability and balance. This chapter presented a shared-control architecture for teleoperating a torque-controlled humanoid robot. An operator was given control over the hands of the humanoid, while the whole-body motion was governed by a null-space balancer acting in the null-space of the operator's commands. A novel approach for providing haptic feedback is introduced where the user is fed with high-level informative haptic cues informing her/him about the impact of her/his potential actions on the robot's balance. This approach bridges the gap between the task itself and the different constraints of the system, thus allowing the operator to adapt her/his approach for a successful task execution within the constraints of the system. We finally present several experiments performed on real robots to validate the proposed architecture along with a discussion of the obtained results.

Chapter 7

Humanoid Balancing while Performing High-Force Interaction Tasks

Contents

7.1 Background and State of the Art **186**
7.2 Controller Design **187**
7.2.1 Force Polyhedron and Support Polygon 188
7.2.2 Proposed Controller 189
7.3 Contact Switching **191**
7.4 Experiments and Results **193**
7.4.1 Experimental Setup 193
7.4.2 Experimental Results: Balancing 194
7.4.3 Experimental Results: Automatic Contact Switching . . 197
7.5 Summary **199**

In the previous chapter, we presented an assisted control architecture for teleoperating humanoid robots. However, the proposed architecture was designed for quasi-static situations and did not account for interaction forces between the robot and the environment. This chapter extends the underlying passivity-based whole-body balancing framework used in the previous chapter (see Sec. 6.2.2) to guarantee the equilibrium of a humanoid robot while performing different interaction tasks when the (high) task forces acting on the robot are difficult to predict. In the context of this thesis, the controller presented in this chapter serves as an essential building block toward a shared-control architecture which allows the user to exert the needed forces when teleoperating a humanoid robot. The work presented in this chapter is

under consideration for publication in [34] and a corresponding descriptive video is available at https://youtu.be/TBl49-XvnRE.

In the rest of this chapter, Sec. 7.1 presents the relevant background while Sec. 7.2 introduces the proposed balancing controller. Sec 7.3 details the proposed automatic contact-switch mechanism after which Sec. 7.4 describes the performed experiments and comments on the results. Finally Sec. 7.5 concludes the chapter.

7.1 Background and State of the Art

While humanoid robots have been a research topic for a few decades, they are yet to attain their expected potential. Most of the applications considered in literature are 'modest' compared to what a humanoid robot is expected to accomplish in a real-life scenario. While usual applications range from driving a car to opening doors, walking on rough terrains, or using simple tools, a real disaster scenario is much more demanding. One would expect the robot to be capable of moving heavy objects (rocks, debris), operating heavy machinery, and employing tools and devices that require both skill and strength, such as hydraulic rescue tools. Performing such tasks requires suitable control frameworks for dealing with the interaction forces generated during the execution of the task, while still accounting for the balance of the robotic platform.

Accounting for interaction forces of the robot with the environment can be achieved mainly in two different ways. A straightforward approach is to seperate interaction and balancing tasks, which usually translates into considering two independent controllers for the lower and upper body of the robot [257, 258]. Forces coming from interactions with the environment using the upper body are considered as a disturbance input by the underlying balancing control, which guarantees the robot balance using only the joints in the lower body. The consideration of force distribution across lower and upper body at the same time can also be achieved through whole-body control frameworks, which exploit the capabilities of redundant robots to deal with multiple tasks [252, 259, 249, 260, 261]. Different whole-body balancing controllers have been proposed based mainly on two approaches: solving the inverse kinematics or dynamics of the robot [262, 252, 251, 250], or using passivity-based approaches [263, 264, 249]. A subset of the whole-body control frameworks feature a hierarchical architecture that allows for multiple control objectives [252, 260]. In [261], a hierarchical approach was used to balance on contacts scattered over the whole body of the robot.

When it is foreseen that high forces could arise in the interaction of the robot with the environment, the robot can plan in advance the best posture and force distribution required to deal with the task [265]. A recent method to achieve this

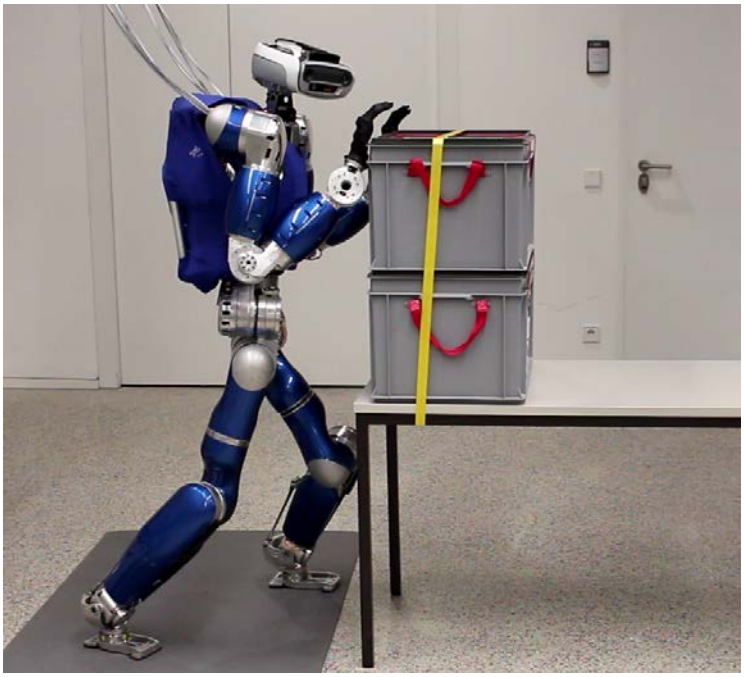


Figure 7.1: TORO pushing a table with a weight of 50 kg on top.

goal is based on the computation of the so-called Gravito-Inertial Wrench Cone (GIWC), which provides the maximum perturbations that the robot can resist at a given configuration, and/or the maximum interaction force that the robot can generate at a given posture [253, 266, 267]. However, this polyhedron has been so far used for (offline) planning and not for instantaneous control of the robot posture. In this context, the main goal of the architecture presented in this chapter is to leverage the concept of the GIWC and apply it to instantaneous control of the robot posture. A whole-body controller usually requires an explicit goal provided for the center of mass (CoM) trajectory. The proposed approach does not specify an explicit location for the CoM; instead, the CoM is moved such that the contact wrenches required for balancing the interaction force are feasible for each contact configuration. In that way, the polyhedron of feasible balancing wrenches or GIWC is used to ensure the online stability of the humanoid while interacting with an unknown environment. Moreover, the same polyhedron is employed to allow the robot to automatically switch between different contact configurations without the need to explicitly plan the trajectory of the CoM.

7.2 Controller Design

In the architecture described in Sec. 6.2, a Cartesian compliance defines the wrench $\boldsymbol{F}_c$ to be applied at the CoM in order to stabilize it at a desired configuration or

move it over a desired trajectory. This approach proved efficient for balancing, and showed significant robustness to noise and external disturbances [249]. However, the amount of force and torque the robot can apply with the interaction end-effectors is limited because the balancing contacts must provide an appropriate counter-force constrained by the contact model (6.6). With that in mind, we hereby propose a new approach for controlling the wrench $\boldsymbol{F}_c$ on the CoM while accounting for the interaction forces between the robot and the environment. The proposed architecture is based on the controller presented in [249] and aims at ensuring the feasibility of the needed balancing wrenches as a function of the contact configuration of the balancing end-effectors by moving the CoM accordingly. The control is based on the polyhedron of feasible balancing wrenches calculated from the current contact configuration, with no need for calculating the Zero Moment Point (ZMP) support area.

7.2.1 Force Polyhedron and Support Polygon

The support polygon of a humanoid robot is the region in which the vertical projection of the CoM must lie so that the balancing end-effectors can carry, statically, the robot's weight. In case the humanoid is on a horizontal plane, the support polygon is the convex hull of the contact areas with the ground, mainly corresponding to the feet of the humanoid (Fig. 6.3). While the support polygon is restricted to static scenarios, the zero-moment point (ZMP) support area is the generalization of the support polygon to the dynamic level. However, in a multi-contact scenario where other parts of the humanoid's body are used for balancing (e.g. the hands), the calculation of the ZMP support area gets more complicated [268]. The same holds in the case of non-horizontal or non-planar contact surfaces (e.g. rough terrain).

The ZMP support area is calculated from the GIWC, which is the set of wrenches that the balancing end-effectors can apply at a particular contact configuration. The contact configuration, referred to hereafter by $\boldsymbol{\Upsilon}$, is the pose of the end-effectors used for balancing in the world frame. The GIWC is dependent on the contact configuration itself and the contact model (see (6.6)) specifying the maximum load that each end-effector can carry. The contact model can be re-written into a polyhedron $\boldsymbol{\zeta}_i = \{\boldsymbol{F}_{\text{bal,i}} : \boldsymbol{A}_i\boldsymbol{F}_{\text{bal,i}} \leq \boldsymbol{b}_i\}$ representing the set of feasible wrenches for each balancing end-effector (see [266] for details). The GIWC can be calculated via the Minkowski-Sum of $\boldsymbol{\zeta}_i$, resulting in the wrench polyhedron $\boldsymbol{\zeta}$ defined as

$$\boldsymbol{\zeta} = \{{}^{w}\boldsymbol{F}_{\text{bal}} : \boldsymbol{A}\,{}^{w}\boldsymbol{F}_{\text{bal}} \leq \boldsymbol{b}\} \tag{7.1}$$

where $\boldsymbol{A} = [\,\boldsymbol{A}_1 \ldots \boldsymbol{A}_n\,]^T \in \mathbb{R}^{6\times n}$ is a constant matrix, ${}^{w}\boldsymbol{F}_{\text{bal}} \in \mathbb{R}^6$ is a wrench defined in the world frame $\mathcal{W}$, and $\boldsymbol{b} = (\,b_1 \ldots b_n\,)^T \in \mathbb{R}^n$ is a constant vector. Any wrench ${}^{w}\boldsymbol{F}$

satisfying (7.1) is feasible and can be applied by the balancing end-effectors in the current contact configuration $\boldsymbol{\Upsilon}$. If $\boldsymbol{\Upsilon}$ changes, $\boldsymbol{\zeta}$ can be re-computed accordingly. We refer the reader to [266] for more details on the calculation of the GIWC.

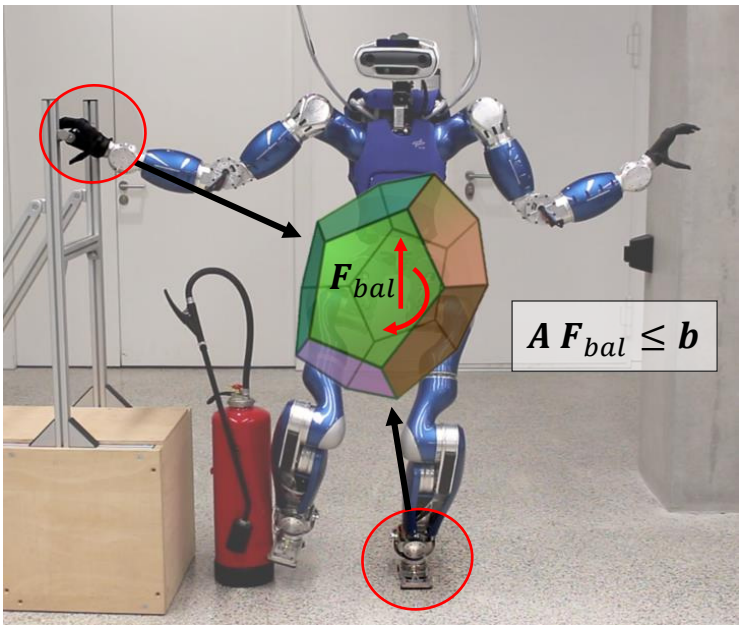


Figure 7.2: TORO balancing with its right hand and left foot while interacting with the environment using its right foot. The two balancing end-effectors can exert a set of forces estimated by the polyhedron $\boldsymbol{AF}_{\text{bal}} \leq \boldsymbol{b}$.

The ZMP support area is calculated at a particular pre-defined point by projecting $\boldsymbol{\zeta}$ on a certain plane of interest [267]. For instance, in the case of a robot standing on horizontal ground, $\boldsymbol{\zeta}$ is projected on the horizontal plane at a point located on the ground to retrieve the ZMP support area (or consequently, the support polygon). This solution is, however, clearly suboptimal since $\boldsymbol{\zeta}$ could span the whole polyhedron (7.1) and not just be restricted to a planar projection. Therefore we propose to define a control law at the level of the balancing wrench itself, without the need of calculating the ZMP support area.

7.2.2 Proposed Controller

Assuming a quasi-static scenario, the dynamic model (see (6.1)) describing the forces at the CoM and defined at the origin of the world frame $\mathcal{W}$ can be written as

$$ {}^{w}\boldsymbol{F}_{\text{bal,t}} + {}^{w}\boldsymbol{Ad}_{\text{int}}(\boldsymbol{p}_{\text{int}})^{T}\boldsymbol{F}_{\text{int}} + {}^{w}\boldsymbol{Ad}_{c}(\boldsymbol{p}_{c})^{T} m\boldsymbol{g}_{0} = 0, \tag{7.2} $$

where ${}^{w}\boldsymbol{F}_{\text{bal,t}} = {}^{w}\boldsymbol{Ad}_{\text{bal}}\boldsymbol{F}_{\text{bal}}$.

On the other hand, from the definition of $\boldsymbol{\zeta}$ in (7.1), the balancing wrenches are feasible if

$$\boldsymbol{A}\,{}^{w}\boldsymbol{F}_{\text{bal,t}} \leq \boldsymbol{b}. \tag{7.3}$$

Assuming that the robot starts from a feasible configuration, we define a "distance" measure d_i from the current balancing wrench ${}^{w}\boldsymbol{F}_{\text{bal,t}}$ to the i_{th} face of the polyhedron $\boldsymbol{\zeta}$ given by $\{\boldsymbol{F}_{\text{bal,i}} : \boldsymbol{A}_i\boldsymbol{F}_{\text{bal,i}} = \boldsymbol{b}_i\}$, such that

$$d_i(\boldsymbol{\zeta}, {}^{w}\boldsymbol{F}_{\text{bal,t}}) = \frac{\boldsymbol{b}_i - \boldsymbol{A}_i\,{}^{w}\boldsymbol{F}_{\text{bal,t}}}{||\boldsymbol{A}_i||}. \tag{7.4}$$

The balancing end-effectors can apply the required balancing wrench ${}^{w}\boldsymbol{F}_{\text{bal,t}}$ if and only if $d_i > 0\ \forall i$. We then define a potential H to encode the proximity of the balancing wrench to the limits of $\boldsymbol{\zeta}$ as

$$H(\boldsymbol{\zeta}, {}^{w}\boldsymbol{F}_{\text{bal,t}}) = \sum_i \frac{1}{d_i\,(\boldsymbol{\zeta}, {}^{w}\boldsymbol{F}_{\text{bal,t}})}. \tag{7.5}$$

Since $\boldsymbol{\zeta}$ is convex, the balance of the humanoid robot can be ensured by defining a gradient-descent control law on H for attaining its minimum. To simplify the notation, we will refer to $H(\boldsymbol{\zeta}, {}^{w}\boldsymbol{F}_{\text{bal,t}})$ and $d_i(\boldsymbol{\zeta}, {}^{w}\boldsymbol{F}_{\text{bal,t}})$ by H and d_i, respectively. The time derivative of H then takes the form

$$\dot{H} = \sum_i -\frac{1}{d_i^2}\dot{d}_i = \left(\sum_i \frac{\boldsymbol{A}_i}{d_i^2\ ||\boldsymbol{A}_i||}\right)\ \frac{d\,({}^{w}\boldsymbol{F}_{\text{bal,t}})}{dt} \tag{7.6}$$

On the other hand, from (7.2) we know that

$${}^{w}\boldsymbol{F}_{\text{bal,t}} = -{}^{w}\boldsymbol{A}\boldsymbol{d}_{\text{int}}(\boldsymbol{p}_{\text{int}})^T\boldsymbol{F}_{\text{int}} - {}^{w}\boldsymbol{A}\boldsymbol{d}_c(\boldsymbol{p}_c)^T m\boldsymbol{g}_0 \tag{7.7}$$

As no previous knowledge of the environment is assumed, $\boldsymbol{F}_{\text{int}}$ in the above equation is unknown. Substituting (7.7) in (7.6) we get

$$\dot{H} = -\left(\sum_i \frac{\boldsymbol{A}_i}{d_i^2\ ||\boldsymbol{A}_i||}\right)\ \frac{d({}^{w}\boldsymbol{A}\boldsymbol{d}_c(\boldsymbol{p}_c)^T m\boldsymbol{g}_0)}{dt} + \vartheta, \tag{7.8}$$

where ϑ is an unknown variable that cannot be controlled and represents the effect of $\boldsymbol{F}_{\text{int}}$. The term ${}^{w}\boldsymbol{A}\boldsymbol{d}_c(\boldsymbol{p}_c)^T m\boldsymbol{g}_0$ can be expanded as

$$\begin{aligned} {}^{w}\boldsymbol{A}\boldsymbol{d}_c(\boldsymbol{p}_c)^T m\boldsymbol{g}_0 &= \begin{bmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\ [\boldsymbol{p}_c]_x & \boldsymbol{I}_{3\times3} \end{bmatrix} m\boldsymbol{g}_0 \\ &= \begin{bmatrix} m\boldsymbol{g}_{0,x} \\ \boldsymbol{0}_{3\times3} \end{bmatrix} - \begin{bmatrix} \boldsymbol{0}_{3\times3} \\ [m\boldsymbol{g}_{0,x}]_x \end{bmatrix} \boldsymbol{p}_c, \end{aligned} \tag{7.9}$$

where $[\cdot]_x$ denotes the skew-symmetric operator, and $\boldsymbol{g}_{0,x} \in \mathbb{R}^3$ is the linear component of $\boldsymbol{g}_0 = (\, \boldsymbol{g}_{0,x}^T \; \mathbf{0}_3 \,)^T$. Substituting in (7.8), the variations of H and $\boldsymbol{p}_c$ can be related by

$$\dot{H} = -\left(\sum_i \frac{\boldsymbol{A}_i}{d_i^2 \; ||\boldsymbol{A}_i||} \right) \begin{bmatrix} \mathbf{0}_{3\times 3} \\ [m\boldsymbol{g}_{0,x}]_x \end{bmatrix} \dot{\boldsymbol{p}}_c + \vartheta. \tag{7.10}$$

From the equation above, we note that the variation of z_c, the vertical component of $\boldsymbol{p}_c = [x_c \; y_c \; z_c]^T$, has no impact on H since $m\boldsymbol{g}_{0,x}$ has the form $[0 \; 0 \; mg]^T$. $\dot{H}$ can then be further decomposed as

$$\begin{aligned} \dot{H} &= -\left(\sum_i \frac{\boldsymbol{A}_i}{d_i^2 \; ||\boldsymbol{A}_i||} \right) \begin{bmatrix} \mathbf{0}_{3\times 1} \\ -mg \; \dot{y}_c \\ mg \; \dot{x}_c \\ 0 \end{bmatrix} + \vartheta \\ &= -mg \left(\sum_i \frac{[A_{i,\omega_x}, A_{i,\omega_y}]}{d_i^2 \; ||\boldsymbol{A}_i||} \right) \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \end{bmatrix} + \vartheta. \end{aligned} \tag{7.11}$$

where A_{i,ω_x} and A_{i,ω_y} are the fourth and fifth components of $\boldsymbol{A}_i$ respectively.

As the only CoM parameters impacting H are x_c and y_c, the control wrench $\boldsymbol{F}_c$ to be applied at the CoM (6.5) can be divided into two parts $\boldsymbol{F}_c = [\boldsymbol{F}_{c,b} \; \boldsymbol{F}_{c,i}]^T$, where $\boldsymbol{F}_{c,b} = [F_{c,x} \; F_{c,y}]^T \in \mathbb{R}^2$ ensures the balance of the robot by acting on the x and y components of $\boldsymbol{p}_c$, while $\boldsymbol{F}_{c,i} = [F_{c,z} \; \boldsymbol{F}_{c,\boldsymbol{\omega}}]^T \in \mathbb{R}^4$ commands the z component of $\boldsymbol{p}_c$ and the orientation $\boldsymbol{R}_c$ of the CoM. Consequently, we define $\boldsymbol{F}_{c,b}$ as

$$\begin{aligned} \boldsymbol{F}_{c,b} &= -k_b \left(\frac{\partial H}{\partial [x_c \; y_c]^T} \right)^T - \boldsymbol{D}_{c,b} \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \end{bmatrix} \\ &= \left(mg \left(\sum_i \frac{[A_{i,\omega_x}, A_{i,\omega_y}]}{d_i^2 \; ||\boldsymbol{A}_i||} \right) \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \right)^T - \boldsymbol{D}_{c,b} \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \end{bmatrix}, \end{aligned} \tag{7.12}$$

where $\boldsymbol{D}_{c,b} \in \mathbb{R}^{2\times 2}$ is a damping matrix and k_b is a control gain. On the other hand, $\boldsymbol{F}_{c,i}$ is defined as an impedance-based control task to allow for the control of z_c and $\boldsymbol{R}_c$ as described in sec. 6.2.2.

7.3 Contact Switching

While a humanoid can balance on its feet, different contact configurations may be required in cases where, for example, the ground is not planar or where the robot needs to use its feet to interact with the environment. Switching from a particular contact configuration to another requires shifting the supporting forces

from some end-effectors to others, depending on the given contact configuration. In this section, we describe an autonomous contact-switch algorithm that handles this 'weight shifting' process in the wrench space when the end-effectors are already in position. A typical situation happens when the two feet of the robot are on the ground and one hand is in contact with a wall. The robot can switch contacts to balance using any possible combination of the three end-effectors, i.e. the two feet, the two feet and the hand, or one feet and the hand. The three end-effectors are assumed to be static during the switching process. In order to handle cases where the balancing end-effectors should be moved (e.g. for stepping), this algorithm needs to be topped with a suitable planner.

Assume the robot is balancing in a particular contact configuration, $\boldsymbol{\Upsilon}_n$, with n end-effectors, and needs to switch to a different contact configuration, $\boldsymbol{\Upsilon}_m$, with m different end-effectors. Two GIWC, $\boldsymbol{\zeta}_n$ and $\boldsymbol{\zeta}_m$, can be defined corresponding to contact configurations $\boldsymbol{\Upsilon}_n$ and $\boldsymbol{\Upsilon}_m$ respectively. The goal of the contact switch is to change ${}^w\boldsymbol{F}_{\text{bal},t}$ from polyhedron $\boldsymbol{\zeta}_n$ to polyhedron $\boldsymbol{\zeta}_m$. To this end, and assuming that no external forces are applied on the interacting end-effectors during the contact switch, we define an optimal balancing wrench ${}^w\boldsymbol{F}_{\text{opt},x}$ for each GIWC, $\boldsymbol{\zeta}_x$, in the following way

$$\begin{aligned} &\underset{{}^w\boldsymbol{F}_{\text{opt},x}}{\text{minimize}} \quad H(\boldsymbol{\zeta}_x, {}^w\boldsymbol{F}_{\text{opt},x}) \\ &\text{subject to} \quad \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} {}^w\boldsymbol{F}_{\text{opt},x} = \begin{bmatrix} 0 \\ 0 \\ mg \\ 0 \end{bmatrix}. \end{aligned}$$

Note that when the robot is in contact configuration $\boldsymbol{\zeta}_x$, the gradient descent controller defined in sec. 6.2.2 maintains ${}^w\boldsymbol{F}_{\text{bal},t}$ at ${}^w\boldsymbol{F}_{\text{opt},x}$ if ${}^w\boldsymbol{F}_{\text{int},t} = \mathbf{0}$ (which is assumed to be the case during the contact switch). The contact switch controller aims then to drive ${}^w\boldsymbol{F}_{\text{bal},t}$ from ${}^w\boldsymbol{F}_{\text{opt},n}$ to ${}^w\boldsymbol{F}_{\text{opt},m}$ (i.e. from polyhedron $\boldsymbol{\zeta}_n$ to polyhedron $\boldsymbol{\zeta}_m$). To this end, we define a quadratic contact-switch potential H_{cs} as

$$H_{cs} = ({}^w\boldsymbol{F}_{\text{bal},d}(t) - {}^w\boldsymbol{F}_{\text{bal}})^T({}^w\boldsymbol{F}_{\text{bal},d}(t) - {}^w\boldsymbol{F}_{\text{bal}}) \tag{7.13}$$

where ${}^w\boldsymbol{F}_{\text{bal},d}(t) = (1-\kappa)\ {}^w\boldsymbol{F}_{\text{opt},n} + \kappa\ {}^w\boldsymbol{F}_{\text{opt},m}$ is the desired balancing wrench that shifts smoothly from ${}^w\boldsymbol{F}_{\text{opt},n}$ to ${}^w\boldsymbol{F}_{\text{opt},m}$ as $\kappa : 0 \to 1$. Since the contacts corresponding to both $\boldsymbol{\zeta}_n$ and $\boldsymbol{\zeta}_m$ are established during the switch, and as all the computed wrench polyhedrons are convex, ${}^w\boldsymbol{F}_{\text{bal},d}$ necessarily lies within the Minkowski-Sum of $\boldsymbol{\zeta}_n$ and $\boldsymbol{\zeta}_m$, which ensures its feasibility.

Finally, and similarly to (7.12), the wrench $\boldsymbol{F}_{c,cs}$ to be applied at the CoM is defined as

$$\boldsymbol{F}_{c,cs} = -k_{cs} \left(\frac{\partial H_{cs}}{\partial [x_c \; y_c]^T} \right)^T - \boldsymbol{D}_{c,b} \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \end{bmatrix}, \tag{7.14}$$

where k_{cs} is a control gain. The resulting overall algorithm is detailed in Alg. 2. For this algorithm, ϵ_κ and $\epsilon_{H_{cs}}$ are control variables to be tuned.

Algorithm 2 Contact-Switch algorithm.

```
^w F_bal,d = ^w F_opt,n
Switch from controller (7.12) to controller (7.14) (F_c = F_c,cs)
repeat
    F_c = F_c,cs(^w F_bal,d)
until H_cs < ε_Hcs
κ = 0
repeat
    ^w F_bal,d = (1 − κ) ^w F_opt,n + κ ^w F_opt,m
    F_c = F_c,cs(^w F_bal,d)
    κ = κ + ε_κ
until κ ≥ 1
^w F_bal,d = ^w F_opt,m
repeat
    F_c = F_c,cs(^w F_bal,d)
until H_cs < ε_Hcs
Switch back from controller (7.14) to controller (7.12) (F_c = F_c,b)
```

7.4 Experiments and Results

7.4.1 Experimental Setup

The proposed approach was implemented on TORO, the torque-controlled humanoid robot developed at DLR [256]. The robot has 27 DoF (plus 12 DoF at the hands), a total height of 1.74 m and a weight of 76.4 kg. 25 of the joints, located at the arms, legs and hip, are based on the DLR-KUKA LBR (Lightweight robot arm), and allow for both position and torque control modes. The 2 DoF at the neck are based on Dynamixel servos and can only be controlled in position mode. Besides position and torque sensing at the LBR-based joints, TORO has an IMU at the trunk and 6 DoF Force-Torque sensors at each foot.

Two types of experiments were performed to validate the proposed architecture. The first is a set of three experiments and aims at testing the balancing controller under high-force interaction tasks. The second demonstrates the automatic contact-switch mechanism and elaborates on the correlation between of the behaviour of the used cost functions and the balance of the robot. The experiments are also demonstrated in a video available at https://youtu.be/TBl49-XvnRE.

7.4.2 Experimental Results: Balancing

7.4.2.1 Experiment 1. Carrying a Heavy Box

This experiment consists of a person passing to the robot a box weighing 25 kg (nearly one third of the robot's weight), and then taking it back. The hands of the robot are controlled in impedance mode and are subject to the Cartesian end-effector compliance described in sec. 6.2.2. No feed-forward forces were fed to the arms to counteract the weight of the box, which is totally unknown to the algorithm. Fig. 7.3 shows the results of the experiment. The two dashed lines denote the instances when the robot takes the box from the human (t= 6.5 s) and when it gives it back (t=19.5 s). Fig. 7.3a and Fig. 7.3b show the commanded linear compliance forces on the right and left hand respectively, which reach a maximum of 142 N on the right and 134 N on the left hand. Fig. 7.3c shows the x and y coordinates of the CoM, which is autonomously shifted backwards due to (7.11) to compensate for the weight of the box. This is also reflected in Fig. 7.3d where the ZMP, measured from sensors on the feet of the robot, is rapidly stabilized after the disruptions during the application and the release of the weight. The sensors used for measuring the ZMP position are relatively noisy, and this is visible in the initial ($t < 3$ s) and final ($t > 20$ s) stages of the experiment. The estimation of the CoM position (calculated from joint encoders as in [249]) is, on the other hand, more smooth and accurate. The trajectories of both ZMP and CoM throughout the experiment are visualized in Fig. 7.3e over the support polygon. Note that the ZMP remains in the support area throughout the whole experiment. Despite the disruptions during the box handover, the ZMP remains almost stationary while the CoM is shifted to the back in order to compensate for the weight of the box. It is worth noting here that the maximum weight of the box was limited by the maximum torques that the joints can deliver (in particular, the limit of 40 Nm at the elbow was critical). In the video (https://youtu.be/TBl49-XvnRE), it is visible that the elbows were stretched when carrying the whole mass of 25 kg.

7.4.2.2 Experiment 2. Table Pushing

In the second experiment, the robot is placed in front of a table and is required to push it forward. The weight of the table and the friction parameters with the ground are unknown (as it is usually the case in such situations), the push-forward motion is planned as a forward Cartesian trajectory for the two hands with no feed-forward forces. Fig. 7.4a and Fig. 7.4b show the commanded linear compliance forces, which reach an overall magnitude of 79.5 N and 60.8 N on the right and left hands, respectively. Fig. 7.4c shows the x and y coordinates of the CoM, which shifts significantly forward to compensate for the forces applied by the interacting

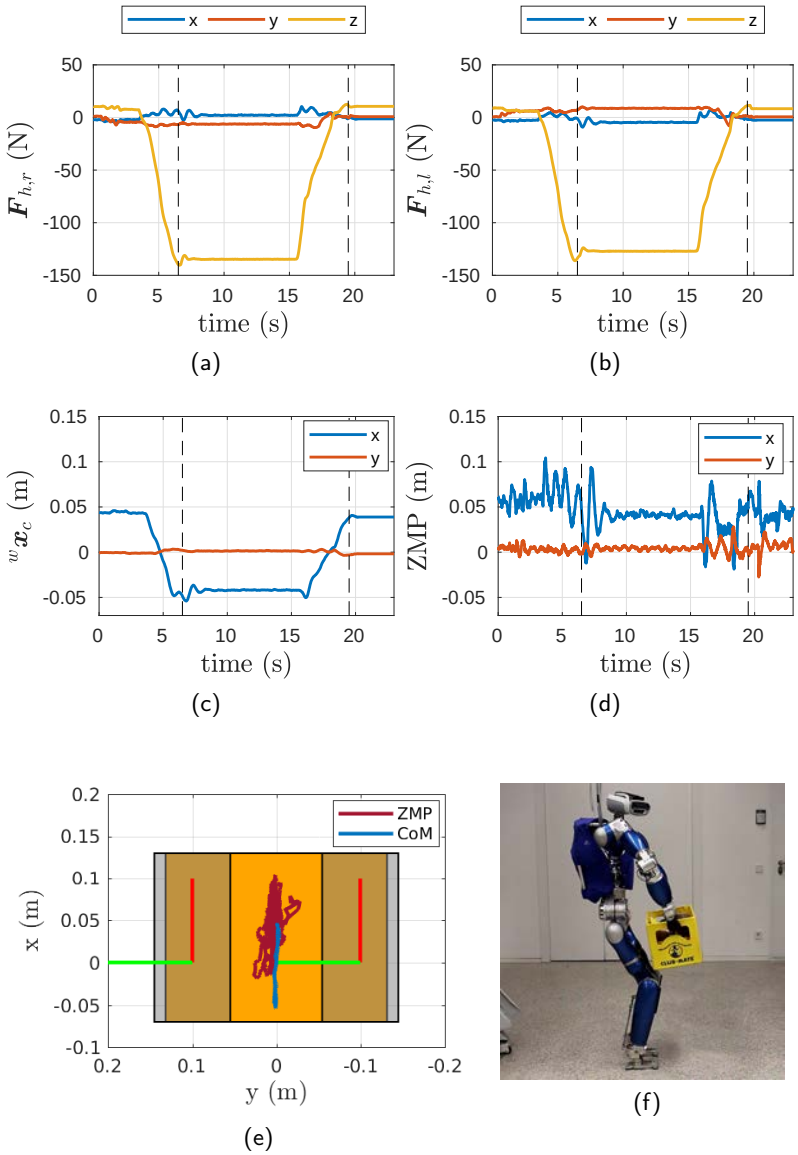


Figure 7.3: Results for experiment 1: lifting a box of 25 kg. (a) Force on the right hand. (b) Force on the left hand. (c) CoM position. (d) ZMP position. (e) CoM and ZMP over the support polygon. (f) Experimental setup.

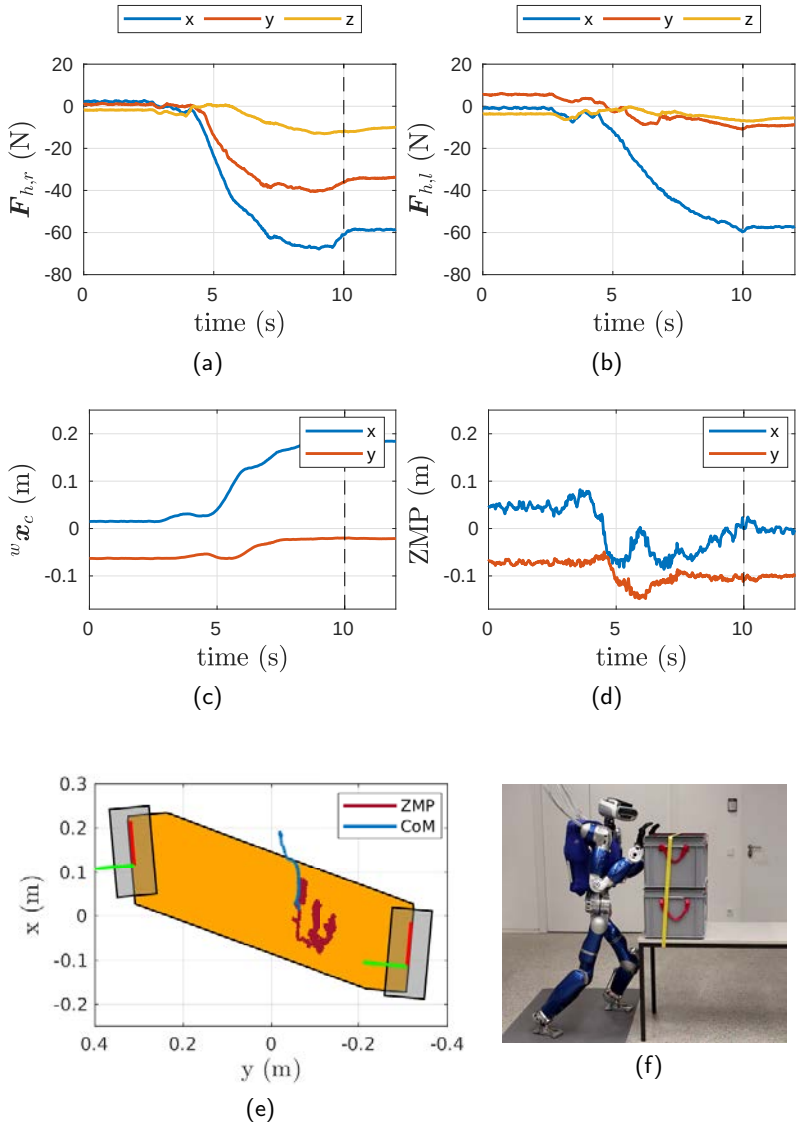


Figure 7.4: Results for experiment 2: pushing a table weighing 50 kg. (a) Force on the right hand. (b) Force on the left hand. (c) CoM position. (d) ZMP position. (e) CoM and ZMP over the support polygon. (f) Experimental setup.

end-effectors. In fact, the CoM leaves the ground support area toward the front, which would result in an immediate fall without the interaction forces. This is reflected in Fig. 7.4d and Fig. 7.4e, which depict the trajectory of the ZMP. Note that the robot had a different contact configuration in this experiment, resulting in a different ZMP support area from that of the first experiment. Fig. 7.4e shows that the ZMP is significantly far from the boundaries of its support area throughout the experiment. It is slightly shifted from the center due to noisy measurements and minor uncertainties in the model of the robot.

7.4.2.3 Experiment 3. Interacting with the Right Foot

The robot in this experiment is balancing using its left foot and right hand, while the right foot and the left hand are free end-effectors available for interaction. The robot uses its right foot to push an object (a fire extinguisher of mass 19.8 kg). Again, the environment and its properties are unknown to the controller, and the motion of the foot is simply planned in the Cartesian space. Fig. 7.5 describes the results of the performed experiments. Fig. 7.5a shows the commanded compliance forces on the right foot, which increase in magnitude up to 70.3 N. The different phases of the experiment are clearly distinguishable on the graph as the foot establishes contact with the object (force starts to increase), starts pushing it (the force is at its maximum) and finally retreats back to its initial position (the force goes back to zero). Fig. 7.5b shows the forces applied by the right hand, which is used for balancing. Note that the applied force maintains a stable behavior as the right foot pushes and releases the object. The threshold for the maximum balancing forces to be applied on the right hand is 100 N; this threshold is never reached during the experiment. The motion of the CoM as it counteracts the force applied on the foot by moving slightly to the right is depicted in Fig. 7.5c and Fig. 7.5e. The latter shows the trajectory of the CoM on the ZMP support area, which is calculated from the contact configuration of the balancing end-effectors (the left foot and the right hand in this case). The left foot, which maintains contact with the ground, is shown as well. The ZMP is not plotted over the ZMP support area as in previous experiments, since there is no information on the forces applied at the right hand (there are no force/pressure sensors on the right hand). However, the behavior of the ZMP of the left foot (measured from the force/torque sensor at the left foot only) is plotted in Fig. 7.5d, and its trajectory is overlayed on the foot itself in Fig. 7.5f. The mentioned ZMP keeps a relatively constant position throughout the experiment, regardless of the forces applied by the right foot.

7.4.3 Experimental Results: Automatic Contact Switching

Unlike the previous experiments, this experiment aims at validating the automatic contact switching algorithm described in sec. 7.3. The robot starts the experiment balancing on its two feet, as shown in Fig. 7.6. It establishes contact with the structure using its right hand, and shifts its weight to balance using its feet and the right hand (see Fig. 7.6b). It then switches again to balance using the right hand and the left foot only, thus liberating the right foot (Fig. 7.6c). Fig. 7.6d shows the change in the position of the center of mass as the robot shifts from one balancing configuration to another. The three contact configurations are respectively referred to by $\boldsymbol{\Upsilon}_{ff}$ for balancing on two feet, $\boldsymbol{\Upsilon}_{ffh}$ for balancing on both feet and the right

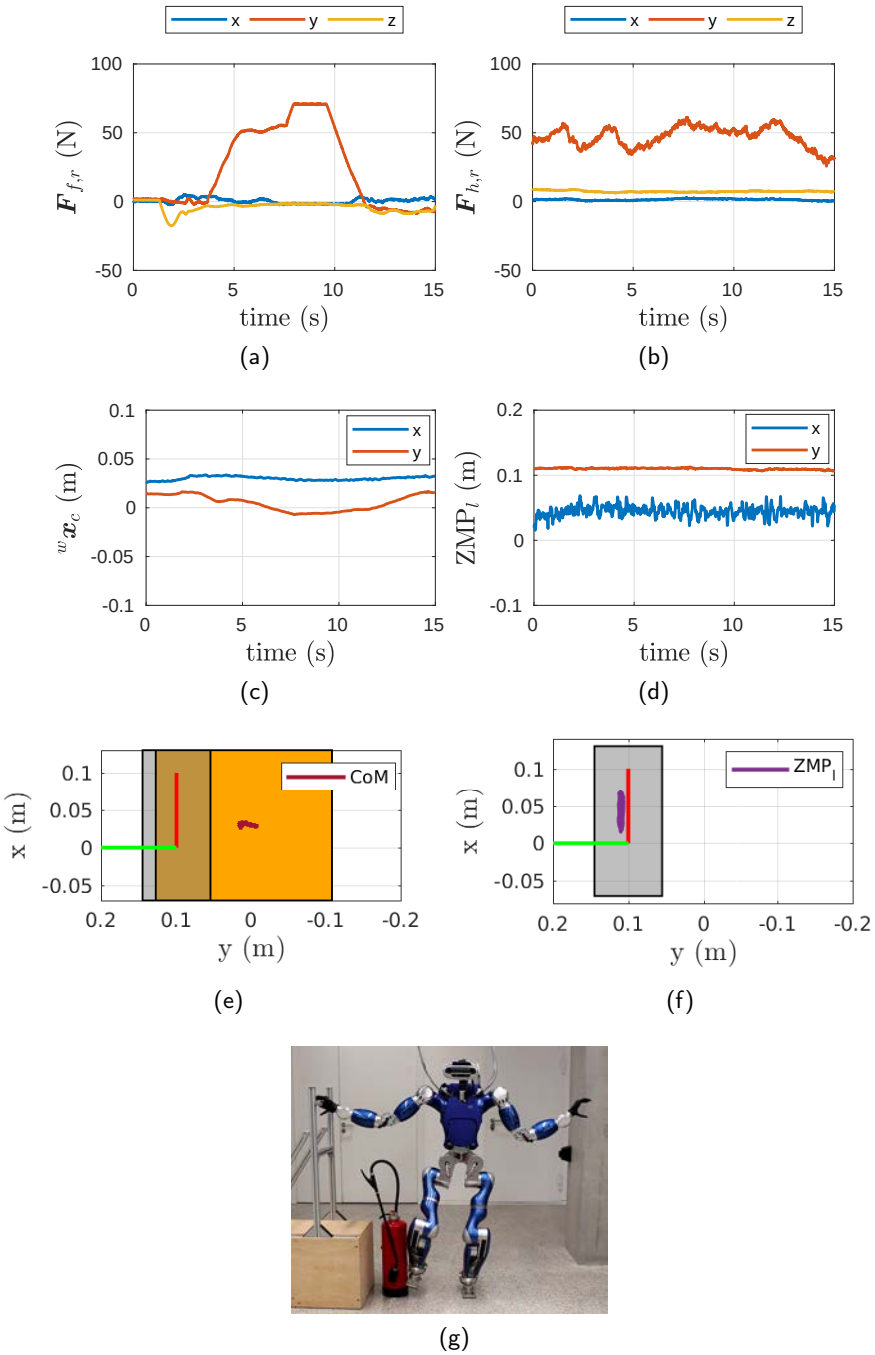


Figure 7.5: Results for experiment 3: using the right foot to move a heavy fire extinguisher (19.8 kg). (a) Force on the right hand. (b) Force on the left hand. (c) CoM position. (d) ZMP position. (e) CoM over the support area. (f) ZMP over the left foot. (g) Experimental setup.

hand, and $\boldsymbol{\Upsilon}_{fh}$ for balancing on the left foot and the right hand. Figures 7.6a-7.6c show the evolution of the potential H corresponding respectively to each one of the three contact configurations $\boldsymbol{\Upsilon}_{ff}$, $\boldsymbol{\Upsilon}_{ffh}$ and $\boldsymbol{\Upsilon}_{fh}$ throughout the experiment. The regions shaded in gray represent the phases during which the contact switch takes place. During the first phase (t=1→4.4 s) the robot, which is balancing on its two feet, incorporates its right hand as an additional balancing end-effector. During the second phase (t=7.3→12 s) the robot shifts from balancing on the two feet and the right hand to balancing on the left foot and the right hand, leaving only the right foot free to move and interact with the environment as needed.

The results of the experiment showed several interesting characteristics of the architecture and the proposed potential H. First, note that the value of the balancing potential H is lowest for contact configuration $\boldsymbol{\Upsilon}_{ffh}$ at ≈ 3.8e-3 (see Fig. 7.6b), higher for contact configuration $\boldsymbol{\Upsilon}_{ff}$ where it ranges between 4.8e-3 and 5.5e-3 (see Fig. 7.6a), and highest for contact configuration $\boldsymbol{\Upsilon}_{fh}$, where it reaches 7.8e-3 at its maximum (see Fig. 7.6c). Recalling from the definition of H in eq. (7.5) that the lower the potential, the better the balance, the results go in-line with the intuition that the robot is best balanced using all three end-effectors, and worst balanced when balancing on one foot and a hand only. On the other hand, the contact switch is reflected as the robot shifts from configuration $\boldsymbol{\Upsilon}_{ff}$ to $\boldsymbol{\Upsilon}_{ffh}$ where the potential corresponding to $\boldsymbol{\Upsilon}_{ff}$ increases while that of $\boldsymbol{\Upsilon}_{ffh}$ decreases to its minimum. A similar behavior is observed when switching from $\boldsymbol{\Upsilon}_{ffh}$ to $\boldsymbol{\Upsilon}_{fh}$ where the potential corresponding to the former increases while that of the latter decreases to its minimum. On the other hand, the degree of variation of each potential reflect as well the robustness of each balancing configuration, where we notice that balancing on one foot and one hand is quite sensitive to changes in the posture of the robot, whereas balancing on both feet and the hand altogether is much more robust.

7.5 Summary

In this chapter, we presented a controller for maintaining the balance of torque-controlled humanoid robots in the presence of unknown (and high) external forces. In the context of this thesis, the controller serves as an essential building block for allowing a human operator teleoperating a humanoid to exert the needed forces for a proper interaction with the environment. The controller acts at the wrench level to ensure that the needed balancing forces lie within the Gravito-Inertial Wrench Cone (GIWC). The same approach is applied to allow for automatic switching between different contact configurations by acting on the GIWC itself. The efficiency and robustness of the approach was demonstrated by several experiments that tested the robot hardware with forces up to the order of 250 N (≈ 1/3 of the robot's

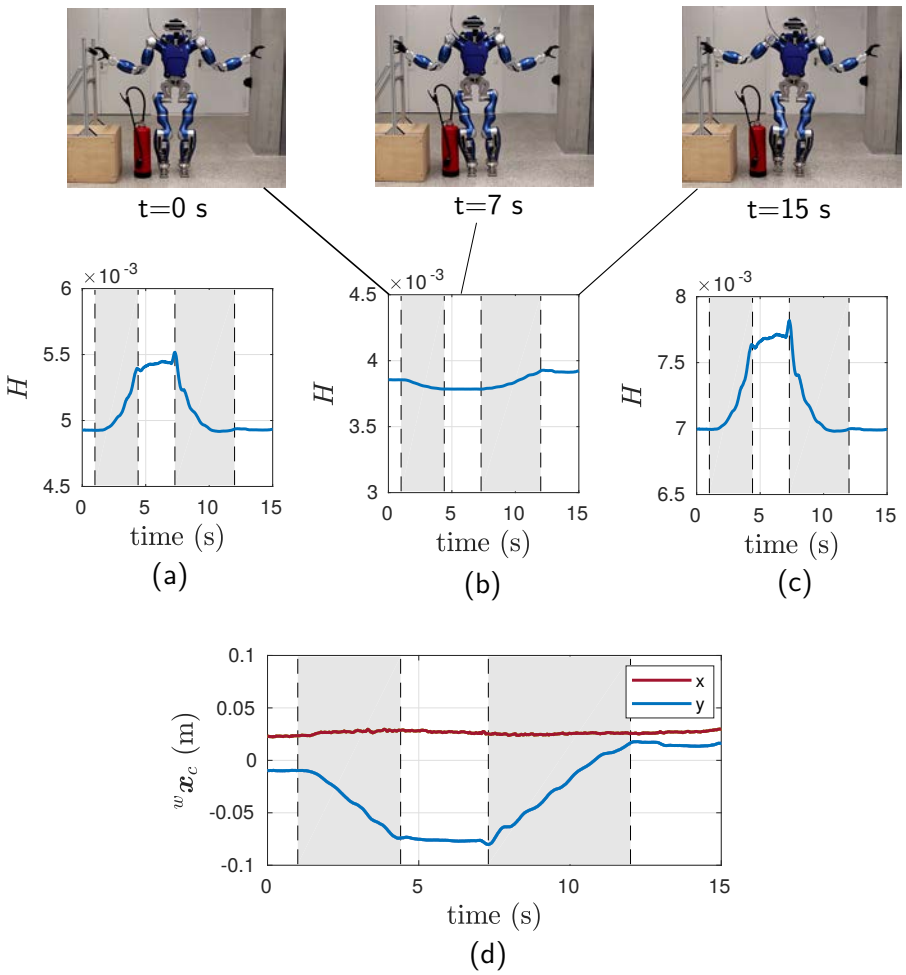


Figure 7.6: Results for experiment 4: Automatic Contact Switching for three different configurations. (a) cost function H corresponding to the two feet being the balancing end-effectors. (b) H corresponding to the two feet and the right hand being the balancing end-effectors. (c) H corresponding to the left foot and the right hand being the balancing end-effectors. (d) Change in position for the center of mass.

weight). The robot interacted with and manipulated the environment using one hand, both hands and even one foot, while the controller was handling the different force directions and contact configurations.

Part IV

Conclusions and Future Work

Chapter 8

Conclusions

This thesis focused on shared-control architectures for assisted manipulation. Besides the scientific interest in advancing the state of the art in shared-control systems, the proposed work was also inspired by the needs of the RoMaNS H2020 European project, whose objective is to design more intuitive and effective ways to sort, segregate and, in general, manipulate nuclear waste by using (semi-)autonomous robot arms. A faster and more efficient sort and segregation of the stored waste is, in fact, the first step toward the decommissioning of old nuclear sites and can have significant economical and societal impacts.

After a brief introduction of the state of the art of HRI in Part I, with a focus on assisted-control systems, we proceeded to detail the contributions of the thesis. Two robotic systems of interest for remote telemanipulation are considered: (*i*) single and dual-arm fixed-base manipulators (in Part II) and (*ii*) humanoid robots (in Part III). While fixed-base manipulators provide a robust and efficient system for manipulation tasks in industrial settings, mobile platforms are essential for a variety of other scenarios where an unforeseen human intervention may be necessary in dangerous, remote or physically demanding environments and humanoids have been proposed as a potential solution.

8.1 Part II: Shared Control for Serial Manipulators

Part II of the thesis tackled three important axes of shared-control modalities: (*i*) passive guidance, where autonomy assists the user to prevent potential system failures, or enforce particular fixtures without interfering in the task itself, (*ii*) active assistance, where autonomy is actively assisting in the completion of the task at hand (e.g. attracting the user toward a grasp pose), and (*iii*) the tuning of the operator/autonomy balance for deciding the division of roles/responsibilities between the human operator and the system autonomy.

Passive Guidance: We introduced a haptic interface which informs the human operator about the proximity of single and dual arm robotic systems to potential constraints like joint limits, singularities and imminent collisions with the environment. While a human operator may be able to independently account for such limitations when commanding a simple robotic system, the complexity grows exponentially as more components are added. This leads to an increase in the cognitive load on the operator, higher risks of failure (e.g. collisions) and large delays in task execution. The proposed haptic guidance effectively steers the user toward 'safe zones' in constrained workspaces decreasing the risk of failure (reflected in the number of collisions) and limiting the cognitive load on the operator while allowing for a faster task execution (reflected in the time-to-completion of the task).

Active Assistance: Two active assistance algorithms for grasping applications are also introduced along with a discussion of the performed user subject tests. The first is the algorithm presented in chap. 3 where the DoF of the system are distributed between autonomy (controlling 2-DoF) and the human operator (controlling the remaining 4-DoF). The control of two DoF subsets is completely decoupled. In the second approach detailed in chap. 4 (specifically Sec. 4.2), autonomy and the operator are both acting in the same space (all 6-DoF of the end-effector). Active haptic cues gently steer the operator toward feasible grasping poses, allowing her/him to intuitively navigate the environment and safely switch between different potential grasp candidates placed on the same or different objects. In order to let the operator distinguish the source of the received haptic cues (active cues guiding toward a grasping pose vs. passive cues informing about proximity to unsafe configuration), we employed vibrotactile feedback besides kinesthetic cues. Active constraints are enforced by conveying kinesthetic feedback through a 6-DoF grounded haptic interface; Passive cues, enforcing the safety limits, are conveyed via kinesthetic feedback provided through the same 6-DoF haptic interface *and* vibrotactile feedback provided by a custom haptic bracelet. Results showed that the proposed shared-control techniques are viable and effective approaches to robotic telemanipulation.

While the first approach simplifies the space for the user from 6-DoF to 4-DoF, it is limited to one target object at a time. Moreover, the operator cannot impact the DoF governed by autonomy which may create issues if the robustness and accuracy of autonomy are questionable. The second approach is a more general and flexible approach which is applicable to multiple objects in cluttered environments. Moreover, it is more robust to autonomy errors as

the human operator can always impact (or simply disregard) the received cues. While both approaches outperformed classical teleoperation in the different considered metrics, picking one over the other is highly dependent on several factors including the nature of the task itself, the reliability of the autonomous controller and its sensing capabilities, the tolerated error margin, and the level of control desired by the human operator.

Operator/Autonomy Balance: The level of control of the human operator versus that of autonomy is always an open question in shared control architectures. In order to tackle this issue, we presented a shared control framework guided by a trajectory distribution learned from human experts. By exploiting the learned distributions, we can adaptively control the balance between autonomy of the controller and the human intervention. We used the variance of the demonstrated trajectories as an indicator of the human experts' preference, and the haptic feedback exerted at the master system was controlled accordingly. Experimental results show that the learned trajectory distribution was refined and the performance of our shared control improved through interactive task executions. However, the stability of the teleoperation system in the presence of a time-varying stiffness needs to be analysed even though no instabilities were reported during the experiments.

The described shared-control approaches, albeit effective, suffer however from a locality issue since the operator can only provide instantaneous velocity commands (in a suitable task space), and receive instantaneous force feedback cues. On the other hand, the ability to 'steer' a whole future trajectory in task space, and to receive a corresponding integral force feedback along the whole planned trajectory (because of any constraint of the considered system), could significantly enhance the operator's performance, especially when dealing with complex manipulation tasks. This aspect has also been considered in Sec. 3.3 as we proposed a shared-planning framework where the human operator acts on some DoF of the future position and orientation trajectories of the manipulators while an autonomous agent optimises the remaining null-space against pre-defined constraints. The user is informed by force cues about the feasibility of her/his actions and the proximity to any of the system's kinematic or visibility constraints.

While the described contributions tackled some aspects of shared-control architectures several questions are yet to be addressed. The design of the cost functions describing the constraints, for example, and the tuning of their different parameters (which was mostly done heuristically) is an important topic which can have a significant impact on the experience of the human operator. However, the main open issue remains to be the operator/autonomy balance. Questions include the

number of DoF which the human operator can optimally control, the DoF which can be reliably controlled by autonomy or the effect which the strength/stiffness of haptic guidance has on the general performance. A system could, for example, use a stiff guidance approach (i.e., less freedom for the operator) when it is operated by novices, while it could implement a soft guidance approach (i.e., more freedom for the operator) when it is operated by experts. The human factor is of course central when considering a shared-control architecture. These questions are briefly tackled in the learning-based framework presented in Sec. 4.3 which is, up to our knowledge, the first attempt to propose an 'established' way for tuning the human/autonomy balance. However, the framework addressed only a single task (e.g. approaching objects). It is interesting to address the generalisation of the architecture to more complicated tasks consisting, for example, of a sequence of primitive motions. Moreover, a human-subject experiment studying the operator's performance under such a framework is also essential.

Finally, Part II also addressed the post-grasp phase of manipulation or the feasibility of the manipulative task itself. Surprisingly, although the task to be performed after a grasp is arguably the core of the manipulation action, the feasibility of this task for a given grasp is mostly ignored in assisted-control literature which focuses more on approaching and grasping. In this regard, we propose in chap. 5 a shared-control architecture which gives the human operator insights on the impact of her/his actions on the (velocity) manipulability of the manipulator over the post-grasp manipulative task.

Although we considered only the geometrical properties of the post-grasp path to be executed by the slave manipulator arm, it would be of course very interesting to also take into account the dynamic properties of the post-grasp task in the grasp selection criteria. Indeed, the mass/inertia of the object, the dynamics of the manipulator and the temporal component of the pick-and-place trajectory, ultimately determine the torque-level control efforts for the slave arm which should also be taken into account in the grasp optimisation procedure. Moreover, it would also be interesting to extend our approach to redundant manipulators as well (thus, with the possibility of exploiting the over-actuation in order to further help the pick-and-place execution).

Furthermore, in our current implementation, the provided force cues (which are generated from the gradient of the proposed TOV optimality index), may move the end-effector towards an optimal pose which is, however, far from the object since no real grasping constraint is included in the optimisation procedure. Indeed, in the reported experiments the operator is responsible for weighting the optimisation action (the force cues) and the feasibility of a possible grasp. However, it would be clearly important to embed in our optimisation procedure a grasping constraint, so

that the cost function gradient would (by construction) drive the user towards an optimal and admissible grasping pose. All these open points are the subject of the author's ongoing research activity.

8.2 Part III: Humanoid Teleoperation and Balancing

In Part III, we presented a shared control strategy for teleoperating a humanoid robot. The user is given control over the two hands of the humanoid while a passivity-based whole-body controller maintains its balance. She/He is informed about the impact of her/his actions on the balance of the robot through informative haptic cues. The underlying balancing controller, however, did not account for external forces and interactions with the environment. To this end, we also extend the balancing framework to guarantee the equilibrium of the humanoid robot while performing different interaction tasks where the (high) task forces acting on the robot are difficult to foresee. Instead of controlling the center of mass, the proposed controller directly uses information from the Gravito-Inertial Wrench Cone to guarantee the feasibility of the balancing forces.

Several experiments were performed on real robots to validate the proposed architectures. in the teleoperation scenario, the provided haptic feedback and the autonomous balancing null-space controller proved efficient in maintaining the balance of the robot. However, some issues were observed as the robot approached its physical limits which had to be taken into account in the design.

On the other hand, the described teleoperation framework considered a particular stance where the robot was standing on its two feet on a horizontal ground. However, it is interesting to employ the algorithm proposed in chap. 7 in the teleoperation controller to account for variable posture/configuration of the robot. This would allow the human operator to change the contact configuration of the robot on the fly which can be essential for proper interaction with the environment. For example, the operator can employ one of the robot hand as an additional balancing end-effector (in addition to the two feet) to allow for more stability and a larger manipulative space of the other hand. Finally, incorporating dynamics into the approach to provide the operator with a meaningful instantaneous guidance during a dynamic behavior, like walking, is another future challenge. A more exhaustive analysis of the passivity and stability of the proposed architecture should also be performed.

8.3 Final Considerations

While a lot of work has already been done on shared control, these architectures are yet to reach the market. This raises some questions on the issues facing the field and the potential future directions which can be adopted to tackle them.

In the current state of the art, most of the considered shared-control architectures are rigid and do not adapt to the human operator using them. We believe however that an autonomous assistant must act in a way similar to a human assistant with wider capabilities and closer partnership with the operator. On one hand, the autonomous assistant needs to be versatile to adapt to the way by which the human operator prefers to approach the problem. On the other, it can also push the human operator towards a different behaviour if this can improve the efficiency of the task execution. To achieve such a level of assisted control, the controller must have an awareness of the task for deciding whether to follow or not the human operator commands, depending on its judgement of the situation. While such "awareness" can be modelled a priori in some cases (by imposing a variable stiffness, for example), it can ideally be learned online. For this, we believe that more use of learning-based techniques in shared control is needed. This is a topic which we briefly touched on in Sec 4.3 and more work is needed in this direction. For example, learning techniques can also be used to vary the assistance level depending on the expertise of the operator. Having learned from demonstrations by expert operators, the algorithm can then assist novices in acquiring the skills by augmenting assistance over some directions or others depending on the contrast between the performance of the novice operator and the demonstrations previously performed by experts. Learning can also be used to aggregate the human operator's preference over time by modifying the learned model with new input or overriding it completely.

Another important point for enhancing shared-control architectures is allowing the operator to switch smoothly between different control architectures through a convenient interface. Different forms of interaction between the human and the robot have been proposed to allow for such an interface based on gesture recognition, intent recognition, and even speech recognition. However, we believe that tailoring such an interface to the particular operator commanding the system is crucial. The attitude of different human operators varies greatly for the same task and autonomy would need to adapt its approach depending on the attitude of the operator. For example, while some operators are more aggressive and increase their stiffness to assure control when conflicting with an autonomous controller, others tend to avoid conflict by allowing autonomy to take lead. Providing the same level of assistance to all operators can lead to varied (and undesirable) results as their reaction is not

always the same. Moreover, the intent of different operators is to be interpreted differently even if the same reaction was observed. The adaptability of the assistance to each user may thus be essential for maintaining the desired operator/autonomy balance and understanding the operator's intent. Autonomy can, for example, start from a nominal level of assistance and adapt to the particular user by learning her/his attitude over time or through a pre-defined set of sample task executions.

On the other hand, shared control is highly dependent on the robustness of the autonomous part of the architecture. If the autonomous controller is prone to error, this can have significant impacts on the task execution. Detecting potential failures of autonomy and reacting accordingly is essential especially if it is acting in the space orthogonal to that of the operator. If a failure of the autonomous part is detected/expected, the control can be delegated back to the human operator who must also be informed about this switch. Moreover, even in the case where the operator and autonomy are acting in the same space, autonomy would ideally inform the operator about its uncertainty to allow her/him to react accordingly. However, detecting such failures or having a measure of the "confidence" of autonomy is not an easy task. Errors can result from issues in the used sensors or weaknesses in the algorithms themselves. Both these cases need to be treated differently. In fact, some measures of quality/certainty of autonomous algorithms have been already proposed. Foe example, we can have a measure of the quality of a reconstructed 3D map, or a score for a potential grasp candidate. However, using such information in designing the shared control architecture still needs farther investigation. Moreover, historical data can also be collected and used in learning-based architectures to develop such measures over time. Demonstrations by experts can be another source of such information.

The nature of the interface which the human is using is another area which can be improved. In this thesis, we concentrated mainly on kinesthetic feedback neglecting other forms of haptic feedback. However, mixing kinesthetic and cutaneous haptic interfaces can be interesting. We marginally touched on this topic in Sec. 4.2, but a lot is to be explored in this direction. While using cutaneous feedback for guiding the operator is a topic which has been discussed in the literature, we are not aware of a thorough investigation of mixing both types of haptic feedback. Cutaneous feedback can be efficient for informing the user about the source or nature of the provided kinesthetic feedback in the case where the user is simultaneously receiving kinesthetic feedback from different sources such as repulsive fixtures, attractive fixtures, or interactions with the environment. On the other hand, visual interfaces are to be exploited for informing the user about similar information as well. For example, when receiving a repulsive haptic cue due to hitting a constraint, the user may be interested in understanding the nature of this constraint (e.g., which

singular configuration is being approached or which joint is about to hit its limit). Such type of information cannot be delivered over the haptic interface alone and this highlights the need for an efficient visual interface such as virtual and augmented reality. A virtual-reality interface have been proposed in Sec. 3.3 where colors and markers were used to transmit information to the user. However, more work is needed in this direction. Moreover, augmented reality is to be explored especially with vision based systems where a camera is already present in the scene.

Moreover, autonomy is to be equipped with improved planning and reasoning capabilities. Apart from the trajectory-based shared planning approach described in Sec. 3.3, the autonomous components of the approaches presented in this Thesis have always been "reactive/instantaneous". But it may be of much interest to equip the robot with the capability of reasoning about the future by predicting the consequence of the user's actions on the task and then give feedback accordingly. In fact, the user is rarely interested in the very details of the task but would still be interested in an immersive interface rather than pure supervisory control. Pushing the limits of shared control towards "shared planning" approaches may be interesting for the operator as it allows her/him to focus on the important aspects of the task. It can also provide her/him with an insight into future issues which the system may face before they occur.

Finally, more interaction with expert operators who are the end-users of industrial-oriented shared-control systems is essential. While this point is highly stressed in medical robotics where surgeons are heavily involved, it is nearly absent in industrial-oriented applications. Involving expert operators in designing shared-control modalities can be highly beneficial for several reasons. It surely can allow for more robust and 'grounded' architectures. But it can also highlight issues which the research community rarely considers like fatigue, boredom, the variation in concentration level, and others. Accounting for such factors in designing shared-control architectures can be key for pushing them closer to the market.

Appendix A

Additional technical details associated to chapter 3

Contents

A.1 Proving the Validity and Orthogonality of the Basis Proposed in (3.7) 213

A.2 Derivation of the Orientation Jacobians in Sec. 3.3.2.2 214

A.2.1 The Orientation Jacobian w.r.t. Control Points Velocities 214

A.2.2 The Orientation Jacobian w.r.t. the Traveling Speed Along the Trajectory 217

THIS appendix includes additional technical details related to the thesis. The mathematical details presented here are not essential for the understanding of the architectures but give to the interested reader an insight on how we proceeded to arrive to certain results.

A.1 Proving the Validity and Orthogonality of the Basis Proposed in (3.7)

We can prove that vectors $\boldsymbol{n}_i$ in (3.7) span the null-space of $\boldsymbol{L}_s$ in (3.5) as follows:

- $\boldsymbol{L}_s\boldsymbol{n}_1 = \begin{bmatrix} \frac{1}{d}\boldsymbol{P}_s & [\boldsymbol{s}]_\times \end{bmatrix} \begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{0} \end{bmatrix} = \boldsymbol{P}_s\boldsymbol{s}/d = \boldsymbol{0}$ because of the properties of $\boldsymbol{P}_s$.
- $\boldsymbol{L}_s\boldsymbol{n}_2 = \begin{bmatrix} \frac{1}{d}\boldsymbol{P}_s & [\boldsymbol{s}]_\times \end{bmatrix} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{s} \end{bmatrix} = [\boldsymbol{s}]_\times\boldsymbol{s} = \boldsymbol{0}$.

- $\boldsymbol{L}_s\boldsymbol{n}_3 = \begin{bmatrix}\frac{1}{d}\boldsymbol{P}_s & [\boldsymbol{s}]_\times\end{bmatrix}\begin{bmatrix}-[\boldsymbol{s}]_\times\boldsymbol{e}_y \\ -\boldsymbol{P}_s\boldsymbol{e}_y/d\end{bmatrix} = (\boldsymbol{P}_s[\boldsymbol{s}]_\times - [\boldsymbol{s}]_\times\boldsymbol{P}_s)\boldsymbol{e}_y/d = \boldsymbol{0}$ since $\boldsymbol{P}_s[\boldsymbol{s}]_\times = [\boldsymbol{s}]_\times\boldsymbol{P}_s = [\boldsymbol{s}]_\times$ because $\boldsymbol{P}_s$ is the orthogonal projector onto the plane orthogonal to $\boldsymbol{s}$ which is spanned by $[\boldsymbol{s}]_\times$.

- $\boldsymbol{L}_s\boldsymbol{n}_4 = \begin{bmatrix}\frac{1}{d}\boldsymbol{P}_s & [\boldsymbol{s}]_\times\end{bmatrix}\begin{bmatrix}[\boldsymbol{s}]_\times\boldsymbol{e}_x \\ \boldsymbol{P}_s\boldsymbol{e}_x/d\end{bmatrix} = ([\boldsymbol{s}]_\times\boldsymbol{P}_s - \boldsymbol{P}_s[\boldsymbol{s}]_\times)\boldsymbol{e}_x/d = \boldsymbol{0}$ because of the same reasons as above.

On the other hand, the null-space vectors exhibit interesting orthogonality properties between one another since $\boldsymbol{n}_1^T\boldsymbol{n}_2 = 0$ by inspection, $\boldsymbol{n}_1^T\boldsymbol{n}_3 = -\boldsymbol{s}^T[\boldsymbol{s}]_\times\boldsymbol{e}_y = 0$, $\boldsymbol{n}_1^T\boldsymbol{n}_4 = \boldsymbol{s}^T[\boldsymbol{s}]_\times\boldsymbol{e}_x = 0$, $\boldsymbol{n}_2^T\boldsymbol{n}_3 = -\boldsymbol{s}^T\boldsymbol{P}_s\boldsymbol{e}_y/d = 0$, $\boldsymbol{n}_2^T\boldsymbol{n}_4 = \boldsymbol{s}^T\boldsymbol{P}_s\boldsymbol{e}_x/d = 0$. $\boldsymbol{n}_3$ and $\boldsymbol{n}_4$ however are only orthogonal when $\boldsymbol{s}$ is orthogonal to $\boldsymbol{e}_x$ and $\boldsymbol{e}_y$ (In the performed experiments $\boldsymbol{s}_d$ is set as $\boldsymbol{e}_z = [1\ 0\ 0]^T$ which satisfies the condition).

A.2 Derivation of the Orientation Jacobians in Sec. 3.3.2.2

A.2.1 The Orientation Jacobian w.r.t. Control Points Velocities

Let us first derive the map between variations of a single control point $\mathrm{d}\,\boldsymbol{\rho}_k$ and variations of the spline $\mathrm{d}\boldsymbol{\eta}$

$$\begin{aligned}
\mathrm{d}\boldsymbol{\eta} &= \bigotimes_{i=1}^{k-1}\widetilde{\boldsymbol{\rho}}_i \otimes \mathrm{d}(\widetilde{\boldsymbol{\rho}}_k \otimes \widetilde{\boldsymbol{\rho}}_{k+1}) \otimes \bigotimes_{i=k+2}^{l}\widetilde{\boldsymbol{\rho}}_i \\
&= \bigotimes_{i=1}^{k-1}\widetilde{\boldsymbol{\rho}}_i \otimes (\mathrm{d}\widetilde{\boldsymbol{\rho}}_k \otimes \widetilde{\boldsymbol{\rho}}_{k+1} + \widetilde{\boldsymbol{\rho}}_k \otimes \mathrm{d}\widetilde{\boldsymbol{\rho}}_{k+1}) \otimes \bigotimes_{i=k+2}^{l}\widetilde{\boldsymbol{\rho}}_i \\
&= \bigotimes_{i=1}^{k-1}\widetilde{\boldsymbol{\rho}}_i \otimes \mathrm{d}\widetilde{\boldsymbol{\rho}}_k \otimes \bigotimes_{i=k+1}^{l}\widetilde{\boldsymbol{\rho}}_i + \bigotimes_{i=1}^{k}\widetilde{\boldsymbol{\rho}}_i \otimes \mathrm{d}\widetilde{\boldsymbol{\rho}}_{k+1} \otimes \bigotimes_{i=k+2}^{l}\widetilde{\boldsymbol{\rho}}_i.
\end{aligned}$$

Since $\widetilde{\boldsymbol{\rho}}_k$ and $\widetilde{\boldsymbol{\rho}}_{k+1}$ are unit quaternions, we can write their differentials in the following form

$$\mathrm{d}\widetilde{\boldsymbol{\rho}}_k = \frac{1}{2}\widetilde{\boldsymbol{\rho}}_k \otimes \begin{bmatrix}0 \\ \boldsymbol{\xi}_k^+\mathrm{d}t\end{bmatrix} \tag{A.1}$$

$$\mathrm{d}\widetilde{\boldsymbol{\rho}}_{k+1} = \frac{1}{2}\begin{bmatrix}0 \\ \boldsymbol{\xi}_k^-\mathrm{d}t\end{bmatrix} \otimes \widetilde{\boldsymbol{\rho}}_{k+1} \tag{A.2}$$

$$\tag{A.3}$$

We then have

$$\begin{aligned}
\mathrm{d}\boldsymbol{\eta} &= \frac{1}{2}\bigotimes_{i=1}^{k}\widetilde{\boldsymbol{\rho}}_i \otimes \begin{bmatrix} 0 \\ \left(\boldsymbol{\xi}_k^+ + \boldsymbol{\xi}_k^-\right)\mathrm{d}t \end{bmatrix} \otimes \bigotimes_{i=k+1}^{l}\widetilde{\boldsymbol{\rho}}_i \\
&= \frac{1}{2}\boldsymbol{\eta} \otimes \left(\bigotimes_{i=k+1}^{l}\widetilde{\boldsymbol{\rho}}_i\right)^* \otimes \begin{bmatrix} 0 \\ \left(\boldsymbol{\xi}_k^+ + \boldsymbol{\xi}_k^-\right)\mathrm{d}t \end{bmatrix} \otimes \bigotimes_{i=k+1}^{l}\widetilde{\boldsymbol{\rho}}_i \\
&= \frac{1}{2}\boldsymbol{\eta} \otimes \begin{bmatrix} 0 \\ {}^{k+1}\boldsymbol{R}_l^T\left(\boldsymbol{\xi}_k^+ + \boldsymbol{\xi}_k^-\right)\mathrm{d}t \end{bmatrix} = \frac{1}{2}\boldsymbol{\eta} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\Lambda}_k \mathrm{d}t \end{bmatrix}
\end{aligned}$$

where ${}^{k+1}\boldsymbol{R}_l$ is the rotation matrix corresponding to the quaternion $\bigotimes_{i=k+1}^{l}\widetilde{\boldsymbol{\rho}}_i$ and we defined

$$\boldsymbol{\Lambda}_k = {}^{k+1}\boldsymbol{R}_l^T\left(\boldsymbol{\xi}_k^+ + \boldsymbol{\xi}_k^-\right). \tag{A.4}$$

To calculate $\boldsymbol{\xi}_k^+$ we can apply the chain rule to the expression of $\widetilde{\boldsymbol{\rho}}_k$. We have

$$\mathrm{d}\widetilde{\boldsymbol{\rho}}_k = \widetilde{B}_k \boldsymbol{D}_k^+ \mathrm{d}({}^{k-1}\boldsymbol{\rho}_k) = \frac{\widetilde{B}_k}{2}\boldsymbol{D}_k^+ \left({}^{k-1}\boldsymbol{\rho}_k \otimes \begin{bmatrix} 0 \\ \boldsymbol{\xi}_k \mathrm{d}t \end{bmatrix}\right)$$

where we exploited (3.50) and we defined

$$\boldsymbol{D}_k^+ = \left.\frac{\mathrm{d}\exp(\boldsymbol{v})}{\mathrm{d}\boldsymbol{v}}\right|_{\log({}^{k-1}\boldsymbol{\rho}_k)\widetilde{B}_k} \left.\frac{\mathrm{d}\log(\boldsymbol{\rho})}{\mathrm{d}\,\boldsymbol{\rho}}\right|_{{}^{k-1}\boldsymbol{\rho}_k} \in \mathbb{R}^{4\times 4}$$

Comparing the above expression with (A.1) we conclude

$$\begin{aligned}
\begin{bmatrix} 0 \\ \boldsymbol{\xi}_k^+ \end{bmatrix} &= \widetilde{B}_k \widetilde{\boldsymbol{\rho}}_k^* \otimes \left[\boldsymbol{D}_k^+ \left({}^{k-1}\boldsymbol{\rho}_k \otimes \begin{bmatrix} 0 \\ \boldsymbol{\xi}_k \end{bmatrix}\right)\right] \Rightarrow \\
\boldsymbol{\xi}_k^+ &= \widetilde{B}_k \boldsymbol{Q}(\widetilde{\boldsymbol{\rho}}_k^*)\boldsymbol{D}_k^+ \boldsymbol{W}({}^{k-1}\boldsymbol{\rho}_k)\boldsymbol{\xi}_k = \boldsymbol{J}_k^+ \boldsymbol{\xi}_k
\end{aligned} \tag{A.5}$$

where

$$\begin{aligned}
\boldsymbol{W}(\boldsymbol{\rho}) : \boldsymbol{\rho} = (\rho_0, \boldsymbol{\rho}_v) \in \mathbb{S}^3 \mapsto \begin{bmatrix} -\boldsymbol{\rho}_v^T \\ \rho_0 \boldsymbol{I}_3 + [\boldsymbol{\rho}_v]_\times \end{bmatrix} \in \mathbb{R}^{4\times 3} \\
\boldsymbol{Q}(\boldsymbol{\rho}) : \boldsymbol{\rho} = (\rho_0, \boldsymbol{\rho}_v) \in \mathbb{S}^3 \mapsto \begin{bmatrix} \boldsymbol{\rho}_v & \rho_0 \boldsymbol{I}_3 + [\boldsymbol{\rho}_v]_\times \end{bmatrix} \in \mathbb{R}^{3\times 4}
\end{aligned}$$

with $[\boldsymbol{v}]_\times$ representing the skew symmetric matrix built with the components of the 3D vector $\boldsymbol{v}$.

Similarly, for $\boldsymbol{\xi}_k^-$ we have

$$\mathrm{d}\widetilde{\boldsymbol{\rho}}_{k+1} = \widetilde{B}_{k+1}\boldsymbol{D}_k^- \mathrm{d}({}^k\boldsymbol{\rho}_{k+1}) = -\frac{\widetilde{B}_{k+1}}{2}\boldsymbol{D}_k^- \left(\begin{bmatrix} 0 \\ \boldsymbol{\xi}_k \mathrm{d}t \end{bmatrix} \otimes {}^k\boldsymbol{\rho}_{k+1}\right)$$

with $\boldsymbol{D}_k^- = \boldsymbol{D}_{k+1}^+$. Comparing this to (A.2) we conclude that

$$\begin{aligned}
\begin{bmatrix} 0 \\ \boldsymbol{\xi}_k^- \end{bmatrix} &= -\widetilde{B}_{k+1}\left[\boldsymbol{D}_k^- \left(\begin{bmatrix} 0 \\ \boldsymbol{\xi}_k \mathrm{d}t \end{bmatrix} \otimes {}^k\boldsymbol{\rho}_{k+1}\right)\right] \otimes \widetilde{\boldsymbol{\rho}}_{k+1}^* \Rightarrow \\
\boldsymbol{\xi}_k^- &= -\widetilde{B}_{k+1}\overline{\boldsymbol{Q}}(\widetilde{\boldsymbol{\rho}}_{k+1}^*)\boldsymbol{D}_k^- \overline{\boldsymbol{W}}({}^k\boldsymbol{\rho}_{k+1})\boldsymbol{\xi}_k = \boldsymbol{J}_k^- \boldsymbol{\xi}_k
\end{aligned} \tag{A.6}$$

where

$$\overline{\boldsymbol{W}}(\boldsymbol{\rho}) : \boldsymbol{\rho} = (\rho_0, \boldsymbol{\rho}_v) \in \mathbb{S}^3 \mapsto \begin{bmatrix} -\boldsymbol{\rho}_v^T \\ \rho_0 \boldsymbol{I}_3 - [\boldsymbol{\rho}_v]_\times \end{bmatrix} \in \mathbb{R}^{4\times 3}$$

$$\overline{\boldsymbol{Q}}(\boldsymbol{\rho}) : \boldsymbol{\rho} = (\rho_0, \boldsymbol{\rho}_v) \in \mathbb{S}^3 \mapsto \begin{bmatrix} \boldsymbol{\rho}_v & \rho_0 \boldsymbol{I}_3 - [\boldsymbol{\rho}_v]_\times \end{bmatrix} \in \mathbb{R}^{3\times 4}$$

Substituting (A.5) and (A.6) in (A.4) we can finally obtain

$$\boldsymbol{\Lambda}_k = {}^{k+1}\boldsymbol{R}_l^T \left(\boldsymbol{J}_k^+ + \boldsymbol{J}_k^- \right) \boldsymbol{\xi}_k = \boldsymbol{J}_k \boldsymbol{\xi}_k. \tag{A.7}$$

As for the calculation of $\boldsymbol{D}$, defining $\frac{\theta}{2} = \operatorname{atan2}(\|\boldsymbol{\rho}_v\|, \rho_0)$ and differentiating (3.45) we obtain

$$\frac{\mathrm{d} \log \boldsymbol{\rho}}{\mathrm{d}\, \boldsymbol{\rho}} = \begin{bmatrix} \frac{\frac{\theta}{2}\cos(\frac{\theta}{2}) - \sin(\frac{\theta}{2})}{\sin^3(\frac{\theta}{2})} \boldsymbol{\rho}_v & \frac{1}{\operatorname{sinc}(\frac{\theta}{2})} \boldsymbol{I}_3 \end{bmatrix}. \tag{A.8}$$

Note that the above expression is well defined for $\frac{\theta}{2} \to 0$, indeed one can easily compute the following truncated Taylor expansion for the first term

$$\frac{\frac{\theta}{2}\cos\left(\frac{\theta}{2}\right) - \sin\left(\frac{\theta}{2}\right)}{\sin^3\left(\frac{\theta}{2}\right)} \approx -\frac{1}{3} - \frac{2}{15}\left(\frac{\theta}{2}\right)^2 - \frac{2}{63}\left(\frac{\theta}{2}\right)^4 - \frac{4}{675}\left(\frac{\theta}{2}\right)^6$$

which can be used to evaluate (A.8) for $\frac{\theta}{2} \approx 0$.

Similarly, differentiating (3.46), one has

$$\frac{\mathrm{d} \exp \boldsymbol{v}}{\mathrm{d}\boldsymbol{v}} = \begin{bmatrix} -\operatorname{sinc}(\|\boldsymbol{v}\|)\boldsymbol{v}^T \\ \left(\frac{\cos(\|\boldsymbol{v}\|) - \operatorname{sinc}(\|\boldsymbol{v}\|)}{\|\boldsymbol{v}\|^2}\right)\boldsymbol{v}\boldsymbol{v}^T + \operatorname{sinc}(\|\boldsymbol{v}\|)\boldsymbol{I}_3 \end{bmatrix}. \tag{A.9}$$

Also in this case, the following well defined truncated Taylor expansion can be used for $\|\boldsymbol{v}\| \approx 0$

$$\frac{\cos(\|\boldsymbol{v}\|) - \operatorname{sinc}(\|\boldsymbol{v}\|)}{\|\boldsymbol{v}\|^2} \approx -\frac{1}{3} + \frac{1}{30}\|\boldsymbol{v}\|^2 - \frac{1}{840}\|\boldsymbol{v}\|^4 + \frac{1}{45360}\|\boldsymbol{v}\|^6.$$

The resulting orientation jacobian $\boldsymbol{J}_{\boldsymbol{\eta},\boldsymbol{\xi}}$ can then be retrieved by stacking the jacobians $\boldsymbol{J}_k$ from (A.7) such that

$$\boldsymbol{J}_{\boldsymbol{\eta},\boldsymbol{\xi}} = \begin{bmatrix} \dots & \boldsymbol{J}_k & \dots \end{bmatrix} \tag{A.10}$$

and the map between the variation of the spline $\mathrm{d}\boldsymbol{\eta}$ and the angular velocities of all control points $\boldsymbol{\rho}_k$ stacked in $\boldsymbol{\xi}$ would be

$$\mathrm{d}\boldsymbol{\eta} = \frac{1}{2}\boldsymbol{\eta} \otimes \begin{bmatrix} 0 \\ \boldsymbol{J}_{\boldsymbol{\eta},\boldsymbol{\xi}}\boldsymbol{\xi}\mathrm{d}t \end{bmatrix} \tag{A.11}$$

For more details about these relations, we also refer the reader to [269].

A.2.2 The Orientation Jacobian w.r.t. the Traveling Speed Along the Trajectory

We now derive the map between the traveling speed along the trajectory σ and variations of the spline $\mathrm{d}\boldsymbol{\eta}$. Operating in a similar way to what done in A.2 we can write

$$\mathrm{d}\boldsymbol{\eta} = \sum_{k=1}^{l} \left[\bigotimes_{i=1}^{k-1} \widetilde{\boldsymbol{\rho}}_i \otimes \mathrm{d}\widetilde{\boldsymbol{\rho}}_k \otimes \bigotimes_{i=k+1}^{l} \widetilde{\boldsymbol{\rho}}_i \right].$$

From [270] we also know that

$$\frac{\mathrm{d}}{\mathrm{d}s}\,\boldsymbol{\rho}^{f(s)} = \frac{\mathrm{d}f}{\mathrm{d}s}\,\boldsymbol{\rho}^{f(s)} \otimes \begin{bmatrix} 0 \\ \log(\boldsymbol{\rho}) \end{bmatrix}.$$

Applying the above relation to $\mathrm{d}\widetilde{\boldsymbol{\rho}}_k$ we can write

$$\begin{aligned}
\mathrm{d}\boldsymbol{\eta} &= \frac{1}{2}\sum_{k=1}^{l} \left[\frac{\mathrm{d}\widetilde{B}_i}{\mathrm{d}s} \bigotimes_{i=1}^{k} \widetilde{\boldsymbol{\rho}}_i \otimes \begin{bmatrix} 0 \\ \log({}^{k-1}\boldsymbol{\rho}_k) \end{bmatrix} \otimes \bigotimes_{i=k+1}^{l} \widetilde{\boldsymbol{\rho}}_i \right] \sigma dt \\
&= \frac{1}{2}\sum_{k=1}^{l} \left[\frac{\mathrm{d}\widetilde{B}_i}{\mathrm{d}s} \boldsymbol{\eta} \left(\bigotimes_{i=k+1}^{l} \widetilde{\boldsymbol{\rho}}_i \right)^{*} \otimes \begin{bmatrix} 0 \\ \log({}^{k-1}\boldsymbol{\rho}_k) \end{bmatrix} \otimes \bigotimes_{i=k+1}^{l} \widetilde{\boldsymbol{\rho}}_i \right] \sigma dt \\
&= \frac{1}{2}\boldsymbol{\eta} \otimes \sum_{k=1}^{l} \begin{bmatrix} 0 \\ \frac{\mathrm{d}\widetilde{B}_i}{\mathrm{d}s}\, {}^{k+1}\boldsymbol{R}_l^T \log({}^{k-1}\boldsymbol{\rho}_k)\sigma dt \end{bmatrix}
\end{aligned}$$

and comparing this expression with (3.56) we conclude that

$$\boldsymbol{J}_{\eta,\sigma} = \sum_{k=1}^{l} \frac{\mathrm{d}\widetilde{B}_i}{\mathrm{d}s}\, {}^{k+1}\boldsymbol{R}_l^T \log({}^{k-1}\boldsymbol{\rho}_k).$$

Finally, the full orientation trajectory dynamics would be

$$\dot{\boldsymbol{\eta}}(s) = \frac{1}{2}\boldsymbol{\eta} \otimes \begin{bmatrix} 0 \\ \boldsymbol{J}_{\eta,\xi}\boldsymbol{\xi} + \boldsymbol{J}_{\eta,\sigma}\sigma \end{bmatrix}. \tag{A.12}$$

Bibliography

[1] T. B. Sheridan, "Human-robot interaction: Status and challenges," *Human Factors*, vol. 58, no. 4, pp. 525–532, 2016, pMID: 27098262. [Online]. Available: https://doi.org/10.117/0018720816644364

[2] R. Goertz, "Master-slave manipulator," Argonne National Laborator, USA, Tech. Rep., 1949.

[3] "1954 electromechanical manipulator by ray goertz," http://cyberneticzoo.com/teleoperators/1954-electromechanical-manipulator-ray-goertz-american/, accessed: 2018-08-01.

[4] J. Vertut, "New types of heavy manipulators," *Proc. Hot Lab. Equip. Conf., 10th, Washington, D.C., 1962*, 1 1962.

[5] J. Vertut, G. Lafort, J. Rouillard, and J. Cazalis, "Master-slave manipulator model through the ceiling with index motion and alpha-beta-gamma shielding," *Proc. Conf. Hot Lab. Equip.*, vol. Vol: 11th, 1 1963.

[6] J. Vertut, *Teleoperation and Robotics: Applications and Technology*. Springer Science & Business Media, Mar. 2013, google-Books-ID: ZEfqCAAAQBAJ.

[7] C. Flatau, "Possible handling methods for radioactivated accelerator components," Brookhaven National Lab., Upton, N.Y., Tech. Rep., 1963.

[8] ——, "Proposed remote handling methods for a modified ags," Brookhaven National Lab., Upton, N.Y., Tech. Rep., 1964.

[9] C. Flatau, J. Vertut, J. Guilbaud, J. Germond, and C. Glachet, "Ma22: a compact bilateral servo master-slave manipulator." *Conf. Remote Syst. Technol., Proc. 20: 296-302(1972).*, 1 1972.

[10] A. Chapuis and E. Droz, *Les Automates: Figures Artificielles D'Hommes Et D'Animaux*. Editions du Griffon, Neuchâtel, 1958.

[11] "Videos present almost 250-year old intricate automatons in action," https://www.realmofhistory.com/2017/03/31/videos-250-year-automatons-jaquet-droz/, accessed: 2018-08-01.

[12] "This mechanical boy from 1770 is the world's first computer," https://sobadsogood.com/2016/01/21/mechanical-boy-1770-worlds-first-computer/, accessed: 2018-08-01.

[13] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: A survey," *Foundations and Trends in Human-Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2007. [Online]. Available: http://dx.doi.org/10.1561/1100000005

[14] H. A. Ernst, "Mh-1, a computer-operated mechanical hand," Ph.D. dissertation, Massachusetts Institute of Technology. Dept. of Electrical Engineering, United States, 1962.

[15] W. R. Ferrell and T. B. Sheridan, "Supervisory control of remote manipulation," *IEEE Spectrum*, vol. 4, no. 10, pp. 81–88, Oct 1967.

[16] A. M. Madni, Y. y. Chu, and A. Freedy, "Intelligent interface for remote supervision and control of underwater manipulation." Institute of Electrical and Electronics Engineers, 1983, pp. pp. 149–155. [Online]. Available: http://www.ieee.org/

[17] F. Miyazaki, S. Matsubayashi, T. Yoshimi, and S. Arimoto, "A new control methodology toward advanced teleoperation of master-slave robot systems," in *Proceedings. 1986 IEEE Int. Conf. on Robotics and Automation*, vol. 3, April 1986, pp. 997–1002.

[18] B. Hannaford, "A design framework for teleoperators with kinesthetic feedback," vol. 5, no. 4, Aug 1989, pp. 426–434.

[19] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035 – 2057, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109806002871

[20] "Radioactive wastes in the uk: A summary of the 2016 inventory," Department for Business, Energy & Industrial Strategy (BEIS) and the Nuclear Decommissioning Authority (NDA), United Kingdom, Tech. Rep., 2016.

[21] K. HASEGAWA, "Facing nuclear risks: Lessons from the fukushima nuclear disaster," *Int. Journal of Japanese Sociology*, vol. 21, no. 1, pp. 84–91, 2012. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1475-6781.2012.01164.x

[22] F. Abi-Farraj, N. Pedemonte, and P. Robuffo Giordano, "A visual-based shared control architecture for remote telemanipulation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2016.

[23] F. Abi-Farraj, T. Osa, N. Pedemonte, J. Peters, G. Neumann, and P. Robuffo Giordano, "A learning-based shared control architecture for interactive task execution," in *IEEE Int. Conf. Robotics and Automation*, 2017, pp. 329–335.

[24] N. Pedemonte, F. Abi-Farraj, and P. Robuffo Giordano, "Visual-based shared control for remote telemanipulation with integral haptic feedback," in *IEEE Int. Conf. Robotics and Automation*, 2017, pp. 5342–5349.

[25] A. Ghalamzan, F. Abi-Farraj, P. Robuffo Giordano, and R. Stolkin, "Human-in-the-loop optimisation: mixed initiative grasping for optimally facilitating post-grasp manipulative actions," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2017, pp. 3386–3393.

[26] F. Abi-Farraj, C. Pacchierotti, and P. Robuffo Giordano, "User evaluation of a haptic-enabled shared-control approach for robotic telemanipulation," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Madrid, Spain, 2018. [Online]. Available: https://hal.inria.fr/hal-01851710

[27] M. Selvaggio, F. Abi-Farraj, C. Pacchierotti, P. R. Giordano, and B. Siciliano, "Haptic-based shared-control methods for a dual-arm system," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4249–4256, Oct 2018.

[28] F. Abi-Farraj, C. Pacchierotti, O. Arenz, G. Neumann, and P. R. Giordano, "A haptic shared-control architecture for guided robotic grasping in unknown environments," *IEEE Transactions on Haptics*, Under consideration for publication.

[29] F. Abi-Farraj, R. Spica, and P. Robuffo Giordano, "Trajectory-based shared control with integral haptic feedback," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems Workshop: Human in-the-loop robotic manipulation: on the influence of the human role*, Vancouver, Canada, Oct. 2017.

[30] ——, "Trajectory-based shared control with integral haptic feedback," in *10th Int. Workshop on Human Friendly Robotics*, Naples, Italy, Nov. 2017.

[31] F. Abi-Farraj, C. Pacchierotti, and P. Robuffo Giordano, "Human-subject evaluation of shared-control approaches for robotic telemanipulation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems Workshop: Human in-the-loop robotic manipulation: on the influence of the human role*, Vancouver, Canada, Oct. 2017.

[32] ——, "Human-subject evaluation of shared-control approaches for robotic telemanipulation," in *10th Int. Workshop on Human Friendly Robotics*, Naples, Italy, Nov. 2017.

[33] F. Abi-Farraj, B. Henze, A. Werner, M. Panzirsch, C. Ott, and M. A. Roa, "Humanoid Teleoperation using Task-Relevant Haptic Feedback," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Madrid, Spain, Oct. 2018, pp. 1–8. [Online]. Available: https://hal.inria.fr/hal-01853934

[34] F. Abi-Farraj, B. Henze, , C. Ott, P. Robuffo Giordano, and M. A. Roa, "Torque-based balancing for a humanoid robot performing high-force interaction tasks," in *IEEE Int. Conf. Robotics and Automation*, Montereal, Canada, May Under consideration for publication, pp. 1–8.

[35] T. B. Sheridan and W. L. Verplank, "Human and computer control of undersea teleoperators," Man-Machine Systems Lab, Massachusetts Institute of Technology, Cambridge Massachusetts, Tech. Rep., 1978.

[36] D. B. Kaber and M. R. Endsley, "The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task," *Theoretical Issues in Ergonomics Science*, vol. 5, no. 2, pp. 113–153, 2004. [Online]. Available: https://doi.org/10.1080/1463922021000054335

[37] C. A. Miller and R. Parasuraman, "Beyond levels of automation: An architecture for more flexible human-automation collaboration," *Proc. the Human Factors and Ergonomics Society Annual Meeting*, vol. 47, no. 1, pp. 182–186, 2003. [Online]. Available: https://doi.org/10.1177/154193120304700138

[38] "5 things humans do every day that robots will soon take over," https://www.inverse.com/article/30583-home-robots-help-with-chores, accessed: 2018-08-01.

[39] "Exhibit 04: Mobile and multipurpose coworker," http://www.smerobotics.org/AUTOMATICA/exhibit-04-2016.html, accessed: 2018-08-01.

[40] J. Artigas, R. Balachandran, C. Riecke, M. Stelzer, B. Weber, J. Ryu, and A. Albu-Schaeffer, "Kontur-2: Force-feedback teleoperation from the interna-

tional space station," in *Proc. IEEE Int. Conf. on Robotics and Automation*, May 2016, pp. 1166–1173.

[41] R. O. Ambrose, H. Aldridge, R. S. Askew, R. R. Burridge, W. Bluethmann, M. Diftler, C. Lovchik, D. Magruder, and F. Rehnmark, "Robonaut: Nasa's space humanoid," *IEEE Intelligent Systems and their Applications*, vol. 15, no. 4, pp. 57–63, July 2000.

[42] L. L. Whitcomb, "Underwater robotics: out of the research laboratory and into the field," in *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 1, April 2000, pp. 709–716 vol.1.

[43] T. Fong and C. Thorpe, "Vehicle teleoperation interfaces," *Autonomous Robots*, vol. 11, no. 1, pp. 9–18, Jul 2001. [Online]. Available: https://doi.org/10.1023/A:1011295826834

[44] S. Lichiardopol, *A survey on teleoperation*, ser. DCT rapporten. Technische Universiteit Eindhoven, 2007, dCT 2007.155.

[45] V. Hayward, O. R. Astley, M. Cruz?Hernandez, D. Grant, and G. Robles?De?La?Torre, "Haptic interfaces and devices," *Sensor Review*, vol. 24, no. 1, pp. 16–29, 2004. [Online]. Available: https://doi.org/10.1108/02602280410515770

[46] B. Edin and N. Johansson, "Skin strain patterns provide kinaesthetic information to the human central nervous system," *The Journal of Physiology*, vol. 487, no. 1, pp. 243–251, 1995.

[47] C. Pacchierotti, L. Meli, F. Chinello, M. Malvezzi, and D. Prattichizzo, "Cutaneous haptic feedback to ensure the stability of robotic teleoperation systems," *Int. Journal of Robotics Research*, vol. 34, no. 14, pp. 1773–1787, 2015. [Online]. Available: https://doi.org/10.1177/0278364915603135

[48] K. O. Johnson, "The roles and functions of cutaneous mechanoreceptors," *Current Opinion in Neurobiology*, vol. 11, no. 4, pp. 455 – 461, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0959438800002348

[49] S. Follmer, D. Leithinger, A. Olwal, A. Hogge, and H. Ishii, "inform: Dynamic physical affordances and constraints through shape and object actuation," in *Proc. the 26th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '13. New York, NY, USA: ACM, 2013, pp. 417–426. [Online]. Available: http://doi.acm.org/10.1145/2501988.2502032

[50] B. Long, S. A. Seah, T. Carter, and S. Subramanian, "Rendering volumetric haptic shapes in mid-air using ultrasound," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 181:1–181:10, Nov. 2014. [Online]. Available: http://doi.acm.org/10.1145/2661229.2661257

[51] H. Culbertson, S. B. Schorr, and A. M. Okamura, "Haptics: The present and future of artificial touch sensation," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 385–409, 2018. [Online]. Available: https://doi.org/10.1146/annurev-control-060117-105043

[52] "Dlr - automatica 2014," https://www.dlr.de/rm/en/desktopdefault.aspx/tabid-3755/17612_read-40216/, accessed: 2018-08-01.

[53] "Haption," https://www.haption.com/en/, accessed: 2018-08-01.

[54] J. Biggs and M. A. Srinivasan, "Tangential versus normal displacements of skin: Relative effectiveness for producing tactile sensations," in *Proc. the 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, ser. HAPTICS '02. Washington, DC, USA: IEEE Computer Society, 2002, pp. 121–. [Online]. Available: http://dl.acm.org/citation.cfm?id=795682.797528

[55] M. Aggravi, F. Pausé, P. R. Giordano, and C. Pacchierotti, "Design and evaluation of a wearable haptic device for skin stretch, pressure, and vibrotactile stimuli," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2166–2173, July 2018.

[56] S. Ertan, C. Lee, A. Willets, H. Tan, and A. Pentland, "A wearable haptic navigation guidance system," in *Digest of Papers. Second Int. Symposium on Wearable Computers (Cat. No.98EX215)*, Oct 1998, pp. 164–165.

[57] T. Lisini Baldi, G. Paolocci, and D. Prattichizzo, "Human guidance: Suggesting walking pace under manual and cognitive load," in *Haptics: Science, Technology, and Applications*, D. Prattichizzo, H. Shinoda, H. Z. Tan, E. Ruffaldi, and A. Frisoli, Eds. Cham: Springer International Publishing, 2018, pp. 416–427.

[58] C. Pacchierotti, D. Prattichizzo, and K. J. Kuchenbecker, "Cutaneous feedback of fingertip deformation and vibration for palpation in robotic surgery," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 2, pp. 278–287, Feb 2016.

[59] "Dp and sirslab's folks blog - claudio pacchierotti," https://prattichizzoblog.wordpress.com/author/cpacchierotti/, accessed: 2018-08-01.

[60] I. Jang and D. Lee, "On utilizing pseudo-haptics for cutaneous fingertip haptic device," in *2014 IEEE Haptics Symposium (HAPTICS)*, Feb 2014, pp. 635–639.

[61] A. Lecuyer, J. Burkhardt, J. L. Biller, and M. Congedo, ""a/sup 4/": a technique to improve perception of contacts with under-actuated haptic devices in virtual reality," in *First Joint Eurohaptics Conf. and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conf.*, March 2005, pp. 316–322.

[62] Y. Ban, T. Narumi, T. Tanikawa, and M. Hirose, "Displaying shapes with various types of surfaces using visuo-haptic interaction," in *Proc. the 20th ACM Symposium on Virtual Reality Software and Technology*, ser. VRST '14. New York, NY, USA: ACM, 2014, pp. 191–196. [Online]. Available: http://doi.acm.org/10.1145/2671015.2671028

[63] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.

[64] C. Passenberg, A. Peer, and M. Buss, "A survey of environment-, operator-, and task-adapted controllers for teleoperation systems," *Mechatronics*, vol. 20, no. 7, pp. 787–801, 2010.

[65] R. J. Anderson and M. W. Spong, "Asymptotic stability for force reflecting teleoperators with time delay," *Int. Journal of Robotics Research*, vol. 11, no. 2, pp. 135–149, 1992. [Online]. Available: https://doi.org/10.1177/027836499201100204

[66] G. Niemeyer and J. . E. Slotine, "Stable adaptive teleoperation," *IEEE Journal of Oceanic Engineering*, vol. 16, no. 1, pp. 152–162, Jan 1991.

[67] J. E. Colgate and G. Schenkel, "Passivity of a class of sampled-data systems: application to haptic interfaces," in *Proc. American Control Conf. - ACC*, vol. 3, June 1994, pp. 3236–3240 vol.3.

[68] Y. Yokokohji, T. Imaida, and T. Yoshikawa, "Bilateral control with energy balance monitoring under time-varying communication delay," in *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 3, April 2000, pp. 2684–2689 vol.3.

[69] D. Lee and P. Y. Li, "Passive coordination control of nonlinear bilateral teleoperated manipulators," in *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 3, May 2002, pp. 3278–3283 vol.3.

[70] A. Franchi, C. Secchi, H. I. Son, H. H. Bulthoff, and P. R. Giordano, "Bilateral teleoperation of groups of mobile robots with time-varying topology," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1019–1033, Oct 2012.

[71] C. Schindlbeck and S. Haddadin, "Unified passivity-based cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks," in *Proc. IEEE Int. Conf. on Robotics and Automation*, May 2015, pp. 440–447.

[72] F. Ferraguti, N. Preda, A. Manurung, M. Bonfè, O. Lambercy, R. Gassert, R. Muradore, P. Fiorini, and C. Secchi, "An energy tank-based interactive control architecture for autonomous and teleoperated robotic surgery," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1073–1088, Oct 2015.

[73] B. Hannaford and J.-H. Ryu, "Time-domain passivity control of haptic interfaces," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 1–10, Feb 2002.

[74] J.-H. Ryu, D.-S. Kwon, and B. Hannaford, "Stable teleoperation with time-domain passivity control," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 365–373, April 2004.

[75] J.-H. Ryu, Y. S. Kim, and B. Hannaford, "Sampled- and continuous-time passivity and stability of virtual environments," *IEEE Transactions on Robotics*, vol. 20, no. 4, pp. 772–776, Aug 2004.

[76] J.-H. Ryu, C. Preusche, B. Hannaford, and G. Hirzinger, "Time domain passivity control with reference energy following," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 5, pp. 737–742, Sept 2005.

[77] D. Lee and K. Huang, "Passive-set-position-modulation framework for interactive robotic systems," *Robotics, IEEE Transactions on*, vol. 26, no. 2, pp. 354–369, 2010.

[78] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl, "Rotex-the first remotely controlled robot in space," in *Proc. IEEE Int. Conf. on Robotics and Automation*, May 1994, pp. 2604–2611 vol.3.

[79] B. Wilcox and T. Nguyen, "Sojourner on mars and lessons learned for future planetary rovers," in *SAE Int. Conf. on Environmental Systems*, Jul 1998.

[80] P. C. Leger, A. Trebi-Ollennu, J. R. Wright, S. A. Maxwell, R. G. Bonitz, J. J. Biesiadecki, F. R. Hartman, B. K. Cooper, E. T. Baumgartner, and M. W. Maimone, "Mars exploration rover surface operations: driving spirit at gusev crater," in *IEEE Int. Conf. on Systems, Man and Cybernetics*, vol. 2, Oct 2005, pp. 1815–1822 Vol. 2.

[81] T. Sato and S. Hirai, "Language-aided robotic teleoperation system (larts) for advanced teleoperation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 5, pp. 476–481, October 1987.

[82] A. K. Bejczy, W. S. Kim, and S. C. Venema, "The phantom robot: predictive displays for teleoperation with time delay," in *Proc. IEEE Int. Conf. on Robotics and Automation*, May 1990, pp. 546–551 vol.1.

[83] P. Schmaus, D. Leidner, T. Krueger, A. Schiele, B. Pleintinger, R. Bayer, and N. Y. Lii, "Preliminary insights from the meteron supvis justin space-robotics experiment," *IEEE Robotics and Automation Letters*, pp. 1–1, 2018.

[84] K. H. Goodrich, P. C. Schutte, F. O. Flemisch, R. Williams *et al.*, "Application of the h-mode, a design and interaction concept for highly automated vehicles, to aircraft," in *25th Digital Avionics Systems Conf., 2006 IEEE/AIAA*. IEEE, 2006, pp. 1–13.

[85] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.

[86] N. Turro, O. Khatib, and E. Coste-Maniere, "Haptically augmented teleoperation," in *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 1. IEEE, 2001, pp. 386–392.

[87] J. C. Gerdes and E. J. Rossetter, "A unified approach to driver assistance systems based on artificial potential fields," *Journal of Dynamic Systems, Measurement, and Control*, vol. 123, no. 3, pp. 431–438, 2001.

[88] S. Park, R. D. Howe, and D. F. Torchiana, "Virtual fixtures for robotic cardiac surgery," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2001*. Springer, 2001, pp. 1419–1420.

[89] S. Payandeh and Z. Stanisic, "On application of virtual fixtures as an aid for telemanipulation and training," in *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002. Proceedings. 10th Symposium on*. IEEE, 2002, pp. 18–23.

[90] A. Bettini, P. Marayong, S. Lang, A. M. Okamura, and G. D. Hager, "Vision-assisted control for manipulation using virtual fixtures," *Robotics, IEEE Transactions on*, vol. 20, no. 6, pp. 953–966, 2004.

[91] P. Marayong and A. M. Okamura, "Speed-accuracy characteristics of human-machine cooperative manipulation using virtual fixtures with variable admittance," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 46, no. 3, pp. 518–532, 2004.

[92] A. D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," *Int. Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.

[93] K. Hauser, "Recognition, prediction, and planning for assisted teleoperation of freeform tasks," *Autonomous Robots*, vol. 35, no. 4, pp. 241–254, 2013.

[94] A. Ajoudani, N. Tsagarakis, and A. Bicchi, "Tele-impedance: Teleoperation with impedance regulation using a body?machine interface," *Int. Journal of Robotics Research*, vol. 31, no. 13, pp. 1642–1656, 2012. [Online]. Available: https://doi.org/10.1177/0278364912464668

[95] A. E. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *Proc. the seventh annual ACM/IEEE Int. Conf. on Human-Robot Interaction*. ACM, 2012, pp. 1–8.

[96] H. Boessenkool, D. Abbink, C. J. Heemskerk, F. C. van der Helm, J. G. Wildenbeest *et al.*, "A task-specific analysis of the benefit of haptic shared control during telemanipulation," *Haptics, IEEE Transactions on*, vol. 6, no. 1, pp. 2–12, 2013.

[97] H. Boessenkool, D. Abbink, C. J. Heemskerk, F. C. van der Helm *et al.*, "Haptic shared control improves tele-operated task performance towards performance in direct control," in *World Haptics Conf. (WHC), 2011 IEEE*. IEEE, 2011, pp. 433–438.

[98] Y. S. Park, H. Kang, T. F. Ewing, E. L. Faulring, J. E. Colgate, and M. A. Peshkin, "Enhanced teleoperation for d and d," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 4, 2004, pp. 3702–3707 Vol.4.

[99] M. W. Noakes, "Telerobotic sensor-based tool control derived from behavior-based robotics concepts," Ph.D. dissertation, University of Tennessee - Knoxville, 2011.

[100] L. B. Rosenberg, "Virtual fixtures: Perceptual tools for telerobotic manipulation," in *Virtual Reality Annual Int. Symposium, 1993., 1993 IEEE*. IEEE, 1993, pp. 76–82.

[101] A. D. Dragan and S. S. Srinivasa, *Formalizing assistive teleoperation*. MIT Press, July, 2012.

[102] X. Wang, C. Yang, and H. Ma, "Automatic obstacle avoidance using redundancy for shared controlled telerobot manipulator," in *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2015 IEEE Int. Conf. on*. IEEE, 2015, pp. 1338–1343.

[103] D. A. Bell, S. P. Levine, Y. Koren, L. A. Jaros, and J. Borenstein, "Design criteria for obstacle avoidance in a shared-control system," in *Proc. the RESNA*, vol. 94, 1994, pp. 17–24.

[104] M. Babel, F. Pasteau, S. Guégan, P. Gallien, B. Nicolas, B. Fraudet, S. Achille-Fauveau, and D. Guillard, "Handiviz project: clinical validation of a driving assistance for electrical wheelchair," in *IEEE Workshop On Advanced Robotics And Its Social Impacts (ARSO)*, 2015.

[105] H. Yu, M. Spenko, and S. Dubowsky, "An adaptive shared control system for an intelligent mobility aid for the elderly," *Autonomous Robots*, vol. 15, no. 1, pp. 53–66, 2003.

[106] S. Payandeh, "Application of shared control strategy in the design of a robotic device," in *Proc. American Control Conf.*, vol. 6. IEEE, 2001, pp. 4532–4536.

[107] S. S. Nudehi, R. Mukherjee, and M. Ghodoussi, "A shared-control approach to haptic interface design for minimally invasive telesurgical training," *Control Systems Technology, IEEE Transactions on*, vol. 13, no. 4, pp. 588–592, 2005.

[108] K. Lindgren, K. Huang, and B. Hannaford, "Towards real-time surface tracking and motion compensation integration for robotic surgery," in *2017 IEEE/SICE Int. Symposium on System Integration (SII)*, Dec 2017, pp. 450–456.

[109] M. Steele and R. B. Gillespie, "Shared control between human and machine: Using a haptic steering wheel to aid in land vehicle guidance," in *Proc. the human factors and ergonomics society annual meeting*, vol. 45, no. 23. SAGE Publications, 2001, pp. 1671–1675.

[110] S. Anderson, S. Peters, K. Iagnemma, and J. Overholt, "Semi-autonomous stability control and hazard avoidance for manned and unmanned ground vehicles," DTIC Document, Tech. Rep., 2010.

[111] P. G. Griffiths and R. B. Gillespie, "Sharing control between humans and automation using haptic interface: primary and secondary task performance benefits," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 47, no. 3, pp. 574–590, 2005.

[112] A. Franchi, C. Secchi, M. Ryll, H. H. Bülthoff, and P. R. Giordano, "Shared control: Balancing autonomy and human assistance with a group of quadrotor uavs," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 3, pp. 57–68, 2012.

[113] C. Masone, A. Franchi, H. H. Bülthoff, and P. R. Giordano, "Interactive planning of persistent trajectories for human-assisted navigation of mobile robots," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. IEEE, 2012, pp. 2641–2648.

[114] C. Masone, P. R. Giordano, H. H. Bulthoff, and A. Franchi, "Semi-autonomous trajectory generation for mobile robots with integral haptic shared control," in *Proc. IEEE Int. Conf. on Robotics and Automation*. IEEE, 2014, pp. 6468–6475.

[115] F. Janabi-Sharifi and I. Hassanzadeh, "Experimental analysis of mobile-robot teleoperation via shared impedance control," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 41, no. 2, pp. 591–606, 2011.

[116] N. Yu, K. Wang, Y. Li, C. Xu, and J. Liu, "A haptic shared control approach to teleoperation of mobile robots," in *IEEE Int. Conf. on Cyber Technology in Automation, Control, and Intelligent Systems*. IEEE, 2015, pp. 31–35.

[117] L. Kaufman, *Sight and mind: An introduction to visual perception*. Oxford University Press, 1974.

[118] C. Ware and J. Rose, "Rotating virtual objects with real handles," *ACM Trans. Computer-Human Interaction*, vol. 6, no. 2, pp. 162–180, 1999.

[119] S. Zhai and P. Milgram, "Quantifying coordination in multiple dof movement and its application to evaluating 6 dof input devices," in *Proc. SIGCHI Conf. Human Factors in Computing Systems*, 1998, pp. 320–327.

[120] L. M. Parsons, "Inability to reason about an object's orientation using an axis and angle of rotation," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 21, pp. 1259–1277, 1995.

[121] K. Hertkorn, M. A. Roa, M. Brucker, P. Kremer, and C. Borst, "Virtual reality support for teleoperation using online grasp planning," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Nov 2013, pp. 2074–2074.

[122] J. Vogel, K. Hertkorn, R. U. Menon, and M. A. Roa, "Flexible, semi-autonomous grasping for assistive robotics," in *Proc. IEEE Int. Conf. on Robotics and Automation*, May 2016, pp. 4872–4879.

[123] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D, J. Vogel, S. Haddadin, J. Liu, S. S. Cash, P. V. Der, and J. P. Donoghue, "Reach and grasp by people with tetraplegia using a neurally controlled robotic arm." nature 485," *Donoghue*, p. 243, 2012.

[124] P. Aigner and B. McCarragher, "Human integration into robot control utilising potential fields," in *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 1. IEEE, 1997, pp. 291–296.

[125] J. W. Crandall, M. Goodrich *et al.*, "Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation," in *Intelligent Robots and Systems, 2002. IEEE/RSJ Int. Conf. on*, vol. 2. IEEE, 2002, pp. 1290–1295.

[126] P. Marayong, M. Li, A. M. Okamura, and G. D. Hager, "Spatial motion constraints: Theory and demonstrations for robot guidance using virtual fixtures," in *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 2. IEEE, 2003, pp. 1954–1959.

[127] J. I. Bbott and A. M. Okamura, "Virtual fixture architectures for telemanipulation," in *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 2. IEEE, 2003, pp. 2798–2805.

[128] D. A. Abbink, M. Mulder, and E. R. Boer, "Haptic shared control: smoothly shifting control authority?" *Cognition, Technology & Work*, vol. 14, no. 1, pp. 19–28, 2012.

[129] D. A. Abbink and M. Mulder, "Neuromuscular analysis as a guideline in designing shared control," 2010.

[130] J. Wildenbeest, H. Boessenkool, C. Heemskerk, M. de Baar, F. van der Helm, D. Abbink *et al.*, "Haptic shared control in tele-manipulation: Effects of inaccuracies in guidance on task execution," 2015.

[131] A. de Jonge, J. Wildenbeest, H. Boessenkool, and D. Abbink, "The effect of trial-by-trial adaptation on conflicts in haptic shared control for free-air teleoperation tasks," *IEEE transactions on haptics*, no. 1, pp. 1–1, 2015.

[132] J. van Oosterhout, D. Abbink, J. Koning, H. Boessenkool, J. Wildenbeest, and C. Heemskerk, "Haptic shared control improves hot cell remote handling despite controller inaccuracies," *Fusion Engineering and Design*, vol. 88, no. 9, pp. 2119–2122, 2013.

[133] K. Hauser and V. Ng-Thow-Hing, "Fast smoothing of manipulator trajectories using optimal bounded-acceleration shortcuts," in *Proc. IEEE Int. Conf. on Robotics and Automation*. IEEE, 2010, pp. 2493–2498.

[134] S. J. Lederman and R. L. Klatzky, "Haptic perception: A tutorial," *Attention, Perception, & Psychophysics*, vol. 71, no. 7, pp. 1439–1459, 2009.

[135] D. E. Whitney, "Force feedback control of manipulator fine motions," *Journal of Dynamic Systems, Measurement, and Control*, vol. 99, no. 2, pp. 91–97, 1977.

[136] N. Pedemonte, T. Laliberté, and C. Gosselin, "A bidirectional haptic device for the training and assessment of handwriting capabilities," in *World Haptics Conf. (WHC), 2013*. IEEE, 2013, pp. 599–604.

[137] W. B. Griffin, W. R. Provancher, and M. R. Cutkosky, "Feedback strategies for telemanipulation with shared control of object handling forces," *Presence: Teleoperators and Virtual Environments*, vol. 14, no. 6, pp. 720–731, 2005.

[138] A. Kheddar, C. Tzafestas, and P. Coiffet, "The hidden robot concept-high level abstraction teleoperation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 3. IEEE, 1997, pp. 1818–1825.

[139] T. Inagaki, "Adaptive automation: Sharing and trading of control," *Handbook of cognitive task design*, vol. 8, pp. 147–169, 2003.

[140] T. Debus, J. Stoll, R. D. Howe, and P. Dupont, "Cooperative human and machine perception in teleoperated assembly," in *Experimental Robotics VII*. Springer, 2001, pp. 51–60.

[141] E. You and K. Hauser, "Assisted teleoperation strategies for aggressively controlling a robot arm with 2d input," in *Robotics: science and systems*, vol. 7, 2012, p. 354.

[142] D.-J. Kim, R. Hazlett-Knudsen, H. Culver-Godfrey, G. Rucks, T. Cunningham, D. Portee, J. Bricout, Z. Wang, and A. Behal, "How autonomy impacts performance and satisfaction: Results from a study with spinal cord injured subjects using an assistive robot," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 42, no. 1, pp. 2–14, 2012.

[143] R. Parasuraman and V. Riley, "Humans and automation: Use, misuse, disuse, abuse," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 39, no. 2, pp. 230–253, 1997.

[144] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, "A model for types and levels of human interaction with automation," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 30, no. 3, pp. 286–297, 2000.

[145] W. B. Rouse, "Adaptive allocation of decision making responsibility between supervisor and computer," in *Monitoring behavior and supervisory control*. Springer, 1976, pp. 295–306.

[146] D. Kortenkamp, D. Keirn-Schreckenghost, and R. P. Bonasso, "Adjustable control autonomy for manned space flight," in *Aerospace Conf. Proceedings, 2000 IEEE*, vol. 7. IEEE, 2000, pp. 629–640.

[147] M. Li and A. M. Okamura, "Recognition of operator motions for real-time assistance using virtual fixtures," in *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003. Proceedings. 11th Symposium on.* IEEE, 2003, pp. 125–131.

[148] J. Kofman, X. Wu, T. J. Luu, and S. Verma, "Teleoperation of a robot manipulator using a vision-based human-robot interface," *Industrial Electronics, IEEE Transactions on*, vol. 52, no. 5, pp. 1206–1219, 2005.

[149] C. Passenberg, R. Groten, A. Peer, and M. Buss, "Towards real-time haptic assistance adaptation optimizing task performance and human effort," in *IEEE World Haptics Conf.* IEEE, 2011, pp. 155–160.

[150] M. Desai, H. Yanco *et al.*, "Blending human and robot inputs for sliding scale autonomy," in *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE Int. Workshop on.* IEEE, 2005, pp. 537–542.

[151] C. Passenberg, A. Glaser, and A. Peer, "Exploring the design space of haptic assistants: the assistance policy module," *Haptics, IEEE Transactions on*, vol. 6, no. 4, pp. 440–452, 2013.

[152] S. M, "Adaptive automation," *Karwowski W (ed) Int. encyclopedia of ergonomics and human factors*, p. 1077?1079, 2001.

[153] S. G. Loizou and V. Kumar, "Mixed initiative control of autonomous vehicles," in *Proc. IEEE Int. Conf. on Robotics and Automation.* IEEE, 2007, pp. 1431–1436.

[154] A. Enes and W. Book, "Blended shared control of zermelo's navigation problem," in *American Control Conf. (ACC), 2010.* IEEE, 2010, pp. 4307–4312.

[155] D. Aarno, S. Ekvall, and D. Kragić, "Adaptive virtual fixtures for machine-assisted teleoperation tasks," in *Proc. IEEE Int. Conf. on Robotics and Automation.* IEEE, 2005, pp. 1139–1144.

[156] W. Yu, R. Alqasemi, R. Dubey, and N. Pernalete, "Telemanipulation assistance based on motion intention recognition," in *Proc. IEEE Int. Conf. on Robotics and Automation.* IEEE, 2005, pp. 1121–1126.

[157] M. R. Endsley and E. O. Kiris, "The out-of-the-loop performance problem and level of control in automation," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 37, no. 2, pp. 381–394, 1995.

[158] C. E. Billings, *Aviation automation: The search for a human-centered approach*, 1997.

[159] C. J. Winstein, P. S. Pohl, and R. Lewthwaite, "Effects of physical guidance and knowledge of results on motor learning: support for the guidance hypothesis," *Research quarterly for exercise and sport*, vol. 65, no. 4, pp. 316–323, 1994.

[160] J. D. Lee, "Human factors and ergonomics in automation design," *Handbook of Human Factors and Ergonomics, Third Edition*, pp. 1570–1596, 2006.

[161] J. C. de Winter and D. Dodou, "Preparing drivers for dangerous situations: A critical reflection on continuous shared control," in *Systems, Man, and Cybernetics (SMC), 2011 IEEE Int. Conf. on.* IEEE, 2011, pp. 1050–1056.

[162] K. Lee, V. Pruks, and J. Ryu, "Development of shared autonomy and virtual guidance generation system for human interactive teleoperation," in *2017 14th Int. Conf. on Ubiquitous Robots and Ambient Intelligence (URAI)*, June 2017, pp. 457–461.

[163] Y. Hirata and K. Kosuge, "Distributed robot helpers handling a single object in cooperation with a human," in *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 1, April 2000, pp. 458–463 vol.1.

[164] K. Kosuge and Y. Hirata, "Human-robot interaction," in *2004 IEEE Int. Conf. on Robotics and Biomimetics*, Aug 2004, pp. 8–11.

[165] M. Lawitzky, A. Mörtl, and S. Hirche, "Load sharing in human-robot cooperative manipulation," in *19th Int. Symposium in Robot and Human Interactive Communication*, Sept 2010, pp. 185–191.

[166] B. Corteville, E. Aertbelien, H. Bruyninckx, J. D. Schutter, and H. V. Brussel, "Human-inspired robot assistant for fast point-to-point movements," in *Proceedings 2007 IEEE Int. Conf. on Robotics and Automation*, April 2007, pp. 3639–3644.

[167] N. Jarrasse, J. Paik, V. Pasqui, and G. Morel, "How can human motion prediction increase transparency?" in *Proc. IEEE Int. Conf. on Robotics and Automation*, May 2008, pp. 2134–2139.

[168] P. Evrard and A. Kheddar, "Homotopy switching model for dyad haptic interaction in physical collaborative tasks," in *World Haptics 2009 - Third Joint EuroHaptics Conf. and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, March 2009, pp. 45–50.

[169] J. R. Medina, M. Lawitzky, A. Mörtl, D. Lee, and S. Hirche, "An experience-driven robotic assistant acquiring human knowledge to improve haptic cooperation," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Sept 2011, pp. 2416–2422.

[170] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human?robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957415818300321

[171] C. Masone, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano, "Interactive Planning of Persistent Trajectories for Human-Assisted Navigation of Mobile Robots," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012, pp. 2641–2648.

[172] C. Masone, P. Robuffo Giordano, H. H. Bülthoff, and A. Franchi, "Semi-autonomous Trajectory Generation for Mobile Robots with Integral Haptic Shared Control," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2014, pp. 6468–6475.

[173] E. Marchand, F. Spindler, and F. Chaumette, "ViSP for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics and Automation Magazine*, vol. 12, no. 4, pp. 40–52, 2005.

[174] F. Chaumette and S. Hutchinson, "Visual servo control, Part I: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[175] N. Guenard, T. Hamel, and R. Mahony, "A practical visual servo control for an unmanned aerial vehicle," *tr*, vol. 2, no. 24, pp. 331–340, 2008.

[176] R. Spica, P. Robuffo Giordano, and F. Chaumette, "Active Structure from Motion: Application to Point, Sphere and Cylinder," *tr*, vol. 30, no. 6, pp. 1499–1513, 2014.

[177] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *IEEE Transactions on Robotics*, vol. 20, no. 4, pp. 713–723, Aug 2004.

[178] N. Bourbaki, *Topological Vector Spaces*. Springer-Verlag Berlin Heidelberg, 1981.

[179] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," vol. 42, no. 12, pp. 2035–2057, 2006.

[180] E. Nuño, L. Basañez, and R. Ortega, "Passivity-based control for bilateral teleoperation: A tutorial," vol. 47, no. 3, pp. 485–495, 2011.

[181] A. Franchi, C. Masone, V. Grabe, M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano, "Modeling and Control of UAV Bearing-Formations with Bilateral High-Level Steering," vol. 12, no. 31, pp. 1504–1525, 2012.

[182] D. Lee, A. Franchi, H. I. Son, H. H. Bülthoff, and P. Robuffo Giordano, "Semi-Autonomous Haptic Teleoperation Control Architecture of Multiple Unmanned Aerial Vehicles," *IEEE/ASME Trans. on Mechatronics*, vol. 4, no. 18, pp. 1334–1345, 2013.

[183] T. YOSHIKAWA, "Analysis and control of robot manipulators with redundancy," *The First Int. Symposium on Robot Research*, 1984.

[184] D. Lee and M. W. Spong, "Passive bilateral teleoperation with constant time delay," *IEEE Trans. Robot.*, vol. 22, no. 2, pp. 269–281, 2006.

[185] R. Ortega, A. van der Schaft, F. Castanos, A. Astolfi, and A. Astolfi, "Control by interconnection and standard passivity-based control of port-hamiltonian systems," *IEEE Transactions on Automatic Control*, vol. 53, no. 11, pp. 2527–2542, Dec 2008.

[186] D. J. Lee and K. Huang, "Passive-set-position-modulation framework for interactive robotic systems," vol. 26, no. 2, pp. 354–369, 2010.

[187] L. Biagiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*. Berlin Heidelberg: Springer, 2008.

[188] M.-J. Kim, M.-S. Kim, and S. Y. Shin, "A general construction scheme for unit quaternion curves with simple high order derivatives," in *22nd annual conf. on Computer graphics and interactive techniques*, Los Angeles, CA, USA, Aug. 1995, pp. 369–376.

[189] L. Piegl and W. Tiller, *The NURBS book*, 2012.

[190] M.-J. Kim, M.-S. Kim, and S. Y. Shin, "A compact differential formula for the first derivative of a unit quaternion curve," *The Journal of Visualization and Computer Animation*, vol. 7, no. 1, pp. 43–57, 1996.

[191] A. Dietrich, T. Wimbock, A. Albu-Schaffer, and G. Hirzinger, "Integration of reactive, torque-based self-collision avoidance into a task hierarchy," *IEEE Trans. Robot.*, vol. 28, no. 6, pp. 1278–1293, 2012.

[192] C. Dube, M. Tsoeu, and J. Tapson, "A model of the humanoid body for self collision detection based on elliptical capsules," in *IEEE Int. Conf. on Robotics and Biomimetics*, 2011, pp. 2397–2402.

[193] K. Steinbach, J. Kuffner, T. Asfour, and R. Dillmann, "Efficient collision and self-collision detection for humanoids based on sphere trees hierarchies," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2006, pp. 560–566.

[194] F. Seto, K. Kosuge, and Y. Hirata, "Self-collision avoidance motion control for human robot cooperation system using robe," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2005, pp. 3143–3148.

[195] B. Dariush, G. B. Hammam, and D. Orin, "Constrained resolved acceleration control for humanoids," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2010, pp. 710–717.

[196] T. F. Chan and R. V. Dubey, "A weighted least-norm solution based scheme for avoiding joint limits for redundant manipulators," in *IEEE Int. Conf. on Robotics and Automation*, 1993, pp. 395–402.

[197] M. Selvaggio, S. Grazioso, G. Notomista, and F. Chen, "Towards a self-collision aware teleoperation framework for compound robots," in *IEEE World Haptics Conf.*, 2017, pp. 460–465.

[198] J. Cascio, M. Karpenko, Q. Gong, P. Sekhavat, and I. M. Ross, "Smooth proximity computation for collision-free optimal control of multiple robotic manipulators," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2009, pp. 2452–2457.

[199] B. Siciliano and J. J. E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Fifth Int. Conf. on Advanced Robotics*, 1991, pp. 1211–1216 vol.2.

[200] C. Secchi, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano, "Bilateral teleoperation of a group of uavs with communication delays and switching topology," in *IEEE Int. Conf. on Robotics and Automation*, 2012, pp. 4307–4314.

[201] E. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robot. Autom. Mag.*, vol. 12, no. 4, pp. 40–52, 2005.

[202] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor Fusion IV: Control Paradigms and Data Structures*, vol. 1611, 1992.

[203] M. Gualtieri, A. ten Pas, K. Saenko, and R. Platt, "High precision grasp pose detection in dense clutter," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. IEEE, 2016.

[204] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *Int. Journal of Robotics Research*, 2017.

[205] M. Selvaggio, G. Notomista, F. Chen, B. Gao, F. Trapani, and D. Caldwell, "Enhancing bilateral teleoperation using camera-based online virtual fixtures generation," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2016, pp. 1483–1488.

[206] J. Bimbo, C. Pacchierotti, M. Aggravi, N. Tsagarakis, and D. Prattichizzo, "Teleoperation in cluttered environments using wearable haptic feedback," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2017.

[207] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, *Springer Handbook of Robotics*. Springer Berlin Heidelberg, 2008, ch. Robot Programming by Demonstration, pp. 1371–1394.

[208] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[209] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Probabilistic movement primitives," in *Proc. Advances in Neural Information Processing Systems 26*, 2013.

[210] T. Osa, N. Sugita, and M. Mitsuishi, "Online trajectory planning in dynamic environments for surgical task automation," in *Proc. Robotics: Science and Systems*, Berkeley, USA, July 2014.

[211] L. Rozo, S. Calinon, D. G. Caldwell, P. Jimenéz, and C. Torras, "Learning physical collaborative robot behaviors from human demonstrations," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 513–527, 2016.

[212] S. Calinon and A. Billard, "Incremental learning of gestures by imitation in a humanoid robot," in *Proc. the ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI)*, 2007.

[213] S. Ross, G. J. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proc. Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2011, pp. 627–635.

[214] S. Chernova and M. Veloso, "Interactive policy learning through confidence-based autonomy," *Journal of Artificial Intelligence Research*, vol. 34, no. 1, pp. 1–25, 2009.

[215] B. C. da Silva, G. Baldassarre, G. Konidaris, and A. Barto, "Learning parameterized motor skills on a humanoid robot," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2014.

[216] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent Service Robotics*, 2016.

[217] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, pp. 129–145, 1996.

[218] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning for control," *Artificial Intelligence Review*, vol. 11, no. 1, pp. 75–113, 1997.

[219] F. Caccavale, C. Natale, B. Siciliano, and L. Villani, "Six-dof impedance control based on angle/axis representations," *IEEE Trans. on Robotics*, vol. 15, no. 2, pp. 289–300, 1999.

[220] J. A. Gray and A. N. Andry, "A simple calculation for the critical damping matrix of a linear multidegree of freedom system," vol. 9, no. 6, pp. 379–380. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0093641382900350

[221] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[222] Y. Zhou, W. Wang, W. Guan, Y. Wu, H. Lai, T. Lu, and M. Cai, "Visual robotic object grasping through combining rgb-d data and 3d meshes," in *Int. Conf. on Multimedia Modeling*. Springer, 2017, pp. 404–415.

[223] M. Kopicki, R. Detry, M. Adjigble, R. Stolkin, A. Leonardis, and J. L. Wyatt, "One-shot learning and generation of dexterous grasps for novel objects," *Int. Journal of Robotics Research*, vol. 35, no. 8, pp. 959–976, 2016.

[224] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *arXiv preprint arXiv:1603.02199*, 2016.

[225] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[226] T. Osa, A. M. Ghalamzan E., R. Stolkin, R. Lioutikov, J. Peters, and G. Neumann, "Guiding trajectory optimization by demonstrated distributions," *Robotics and Automation Letters*, vol. 2, no. 2, pp. 819–826, 2017.

[227] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *Proc. IEEE Int. Conf. on Robotics and Automation*. IEEE, 2009, pp. 489–494.

[228] T. Yoshikawa, "Manipulability of robotic mechanisms," *Int. journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[229] N. Vahrenkamp, T. Asfour, G. Metta, G. Sandini, and R. Dillmann, "Manipulability analysis," in *12th IEEE/RAS Int. Conf. on Humanoid Robots*, 2012, pp. 568–573.

[230] S. Lee and J. M. Lee, "Task-oriented dual-arm manipulability and its application to configuration optimization," in *Proc. the 27th Conf. on Decision and Control*. IEEE, 1988, pp. 2253–2260.

[231] Q. Zhang and A. Chen, "Kinematics and task-oriented manipulability measure for the bionic cricket hopping robot," in *Int. Conf. on Intelligent Robotics and Applications*. Springer, 2008, pp. 342–351.

[232] A. M. Ghalamzan E., N. Mavrakis, M. Kopicki, R. Stolkin, A. Leonardis *et al.*, "Task-relevant grasp selection: A joint solution to planning grasps and manipulative motion trajectories," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2016, pp. 907–914.

[233] M. B. Horowitz and J. W. Burdick, "Combined grasp and manipulation planning as a trajectory optimization problem," in *Proc. Int. Conf. on Robotics and Automation*. IEEE, 2012, pp. 584–591.

[234] N. Vahrenkamp, M. Przybylski, T. Asfour, and R. Dillmann, "Bimanual grasp planning," in *IEEE/RAS Int. Conf. on Humanoid Robots*, 2011, pp. 493–499.

[235] N. Mavrakis, A. M. Ghalamzan E., R. Stolkin, L. Baronti, M. Kopicki, and M. Castellani, "Analysis of the inertia and dynamics of grasped objects, for choosing optimal grasps to enable torque-efficient post-grasp manipulations," in *IEEE/RAS Int. Conf. on Humanoid Robots*, 2016, pp. 171–178.

[236] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*. Technical University of Denmark, 2012.

[237] R. O. Ambrose, H. Aldridge, R. S. Askew, R. R. Burridge, W. Bluethmann, M. Diftler, C. Lovchik, D. Magruder, and F. Rehnmark, "Robonaut: NASA's space humanoid," *IEEE Intelligent Systems and their Applications*, vol. 15, no. 4, pp. 57–63, 2000.

[238] J. Iqbal, A. M. Tahir, R. ul Islam, and R. un Nabi, "Robotics for nuclear power plants - challenges and future perspectives," in *Int. Conf. Applied Robotics for the Power Industry*, 2012, pp. 151–156.

[239] H. Hasunuma, M. Kobayashi, H. Moriyama, T. Itoko, Y. Yanagihara, T. Ueno, K. Ohya, and K. Yokoil, "A tele-operated humanoid robot drives a lift truck," vol. 3, pp. 2246–2252, 2002.

[240] H. Song, H. Shin, H. You, J. Hong, and D. H. Shim, "Toward autonomous aircraft piloting by a humanoid robot: Hardware and control algorithm design," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2016, pp. 398–403.

[241] S. Tachi, K. Komoriya, K. Sawada, T. Nishiyama, T. Itoko, M. Kobayashi, and K. Inoue, "Telexistence cockpit for humanoid robot control," *Advanced Robotics*, vol. 17, no. 3, pp. 199–217, 2003.

[242] K. Yokoi, K. Nakashima, M. Kobayashi, H. Mihune, H. Hasunuma, Y. Yanagihara, T. Ueno, T. Gokyuu, and K. Endou, "A tele-operated humanoid operator," *Int. J. Robotics Research*, vol. 25, no. 5-6, pp. 593–602, 2006.

[243] M. Zucker, S. Joo, M. X. Grey, C. Rasmussen, E. Huang, M. Stilman, and A. Bobick, "A general-purpose system for teleoperation of the DRC-HUBO humanoid robot," *J. Field Robotics*, vol. 32, no. 3, pp. 336–351, 2015.

[244] E. S. Neo, K. Yokoi, S. Kajita, and K. Tanie, "Whole-body motion generation integrating operator's intention and robot's autonomy in controlling humanoid robots," *IEEE Trans. on Robotics*, vol. 23, no. 4, pp. 763–775, 2007.

[245] E. G. Estrelinha, *Tele-operation of a humanoid robot using haptics and load sensors*. Ph.D. dissertation, Universidade de Aveiro, 2013.

[246] J. Oliveira, V. Ferreira, and F. Teixeira, "Bimanual haptics for humanoid robot teleoperation using ROS and V-REP," in *IEEE Int. Conf. Autonomous Robot Systems and Competitions*, 2015, pp. 174–179.

[247] A. Brygo, I. Sarakoglou, N. Tsagarakis, and D. G. Caldwell, "Tele-manipulation with a humanoid robot under autonomous joint impedance regulation and vibrotactile balancing feedback," in *IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 862–867.

[248] J. Ramos, A. Wang, and S. Kim, "A balance feedback human machine interface for humanoid teleoperation in dynamic tasks," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2015, pp. 4229–4235.

[249] B. Henze, M. A. Roa, and C. Ott, "Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios," *Int. J. Robotics Research*, vol. 35, no. 12, pp. 1522 – 1543, 2016.

[250] M. Mistry, J. Buchli, and S. Schaal, "Inverse dynamics control of floating base systems using orthogonal decomposition," in *IEEE Int. Conf. Robotics and Automation*, 2010, pp. 3406 – 3412.

[251] L. Righetti, J. Buchli, M. Mistry, M. Kalakrishnan, and S. Schaal, "Optimal distribution of contact forces with inverse dynamics control," *Int. J. Robotics Research*, vol. 32, no. 3, pp. 280 – 298, 2013.

[252] L. Sentis, "Compliant control of whole-body multi-contact behaviors in humanoid robots," in *Motion Planning for Humanoid Robots*, K. Harada, E. Yoshida, and K. Yokoi, Eds. Springer, 2010, pp. 29–66.

[253] S. Caron and A. Kheddar, "Multi-contact walking pattern generation based on model preview control of 3D COM accelerations," in *IEEE-RAS Int. Conf. Humanoid Robots*, 2016, pp. 550 – 557.

[254] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *IEEE-RAS Int. Conf. Humanoid Robots*, 2006, pp. 200 – 207.

[255] M. Sagardia, T. Hulin, K. Hertkorn, P. Kremer, and S. Schätzle, "A platform for bimanual virtual assembly training with haptic feedback in large multi-object environments," in *ACM Conf. Virtual Reality Software and Technology*, 2016, pp. 153–162.

[256] J. Englsberger, A. Werner, C. Ott, B. Henze, M. A. Roa, G. Garofalo, R. Burger, A. Beyer, O. Eiberger, K. Schmid, and A. Albu-Schäffer, "Overview of the torque-controlled humanoid robot TORO," in *IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 916–923.

[257] A. Ibanez, P. Bidaud, and V. Padois, "Unified preview control for humanoid postural stability and upper-limb interaction adaptation," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2012, pp. 1801 – 1808.

[258] C. Ott, B. Henze, and D. Lee, "Kinesthetic teaching of humanoid motion based on whole-body compliance control with interaction-aware balancing," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2013, pp. 4615–4621.

[259] B. Henze, C. Ott, and M. A. Roa, "Posture and balance control for humanoid robots in multi-contact scenarios based on model predictive control," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, September 2014, pp. 3253–3258.

[260] B. Henze, A. Dietrich, and C. Ott, "An approach to combine balancing with hierarchical whole-body control for legged humanoid robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 700 – 707, July 2016, date of Publication: 29 December 2015.

[261] B. Henze, A. Dietrich, M. A. Roa, and C. Ott, "Multi-contact balancing of humanoid robots in confined spaces: Utilizing knee contacts," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, September 2017, pp. 697–704.

[262] B. J. Stephens and C. G. Atkeson, "Push recovery by stepping for humanoid robots with force controlled joints," in *IEEE-RAS Int. Conf. Humanoid Robots*, December 2010, pp. 52 –59.

[263] S. Hyon, J. Hale, and G. Cheng, "Full-body compliant human-humanoid interaction: Balancing in the presence of unknown external forces," *IEEE Trans. Robotics*, vol. 23, no. 5, pp. 884–898, Oct 2007.

[264] C. Ott, M. A. Roa, and G. Hirzinger, "Posture and balance control for biped robots based on contact force optimization," in *IEEE-RAS Int. Conf. Humanoid Robots*, 2011, pp. 26–33.

[265] M. Murooka, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Whole-body pushing manipulation with contact posture planning of large and heavy object for humanoid robot," in *IEEE Int. Conf. Robotics and Automation*, 2015, pp. 5682–5689.

[266] S. Caron, Q.-C. Pham, and Y. Nakamura, "Leveraging cone double description for multi-contact stability of humanoids with applications to statics and dynamics," in *Robotics: Science and System*, Jul. 2015.

[267] S. Caron, Q. Pham, and Y. Nakamura, "Zmp support areas for multicontact mobility under frictional constraints," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 67–80, Feb 2017.

[268] H. Hirukawa, S. Hattori, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, and M. Morisawa, "A universal stability criterion of the foot contact of legged robots - adios ZMP," in *IEEE Int. Conf. Robotics and Automation*, 2006, pp. 1976–1983.

[269] N. Dantam, "Quaternion computation," http://www.neil.dantam.name/note/dantam-quaternion.pdf, 2014.

[270] D. Xu and D. P. Mandic, "The theory of quaternion matrix derivatives," *IEEE Transactions on Signal Processing*, vol. 63, no. 6, pp. 1543–1556, March 2015.

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

Titre : Contributions aux architectures de contrôle partagé pour la télémanipulation avancée.

Mots clés : Robotique, Contrôle Partagé, Interfaces Haptiques, Téléopération, Humanoïdes, Manipulation

Résumé : Bien que la pleine autonomie dans des environnements inconnus soit encore loin, les architectures de contrôle partagé où l'humain et un contrôleur autonome travaillent ensemble pour atteindre un objectif commun peuvent constituer un « terrain intermédiaire » pragmatique. Dans cette thèse, nous avons abordé les différents problèmes des algorithmes de contrôle partagé pour les applications de saisie et de manipulation. En particulier, le travail s'inscrit dans le projet H2020 Romans dont l'objectif est d'automatiser le tri et la ségrégation des déchets nucléaires en développant des architectures de contrôle partagées permettant à un opérateur humain de manipuler facilement les objets d'intérêt.

La thèse propose des architectures de contrôle partagé différentes pour manipulation à double bras avec un équilibre opérateur / autonomie différent en fonction de la tâche à accomplir. Au lieu de travailler uniquement sur le contrôle instantané du manipulateur, nous proposons des architectures qui prennent en compte automatiquement les tâches de pré-saisie et de post-saisie permettant à l'opérateur de se concentrer uniquement sur la tâche à accomplir.
La thèse propose également une architecture de contrôle partagée pour contrôler un humanoïde à deux bras où l'utilisateur est informé de la stabilité de l'humanoïde grâce à un retour haptique. En plus, un nouvel algorithme d'équilibrage permettant un contrôle optimal de l'humanoïde lors de l'interaction avec l'environnement est également proposé.

Title: Contributions to Shared Control Architectures for Advanced Telemanipulation.

Keywords: Robotics, Shared Control, Haptics, Teleoperation, Humanoids, Manipulation

Abstract: While full autonomy in unknown environments is still in far reach, shared-control architectures where the human and an autonomous controller work together to achieve a common objective may be a pragmatic "middle-ground". In this thesis, we have tackled the different issues of shared-control architectures for grasping and sorting applications. In particular, the work is framed in the H2020 RoMaNS project whose goal is to automatize the sort and segregation of nuclear waste by developing shared control architectures allowing a human operator to easily manipulate the objects of interest.

The thesis proposes several shared-control architectures for dual-arm manipulation with different operator/autonomy balance depending on the task at hand. While most of the approaches provide an instantaneous interface, we also propose architectures which automatically account for the pre-grasp and post-grasp trajectories allowing the operator to focus only on the task at hand (ex., grasping).
The thesis also proposes a shared control architecture for controlling a force-controlled humanoid robot in which the user is informed about the stability of the humanoid through haptic feedback. A new balancing algorithm allowing for the optimal control of the humanoid under high interaction forces is also proposed.