

Advancing coordination control of multi-limb robotic tasks: application to bimanual dynamic manipulation and bipedal reactive locomotion

Présentée le 18 novembre 2022

Faculté des sciences et techniques de l'ingénieur
Laboratoire d'algorithmes et systèmes d'apprentissage
Programme doctoral en robotique, contrôle et systèmes intelligents

pour l'obtention du grade de Docteur ès Sciences

par

Michael Bosongo BOMBILE

Acceptée sur proposition du jury

Prof. H. Shea, président du jury
Prof. A. Billard, directrice de thèse
Prof. A. Kheddar, rapporteur
Prof. T. Asfour, rapporteur
Prof. S. Sakar, rapporteur

ABSTRACT

HUMANS have good bi-manual manipulation skills. They can effortlessly coordinate the motion of their two hands and execute successfully a given task. For instance, humans often resort to their bi-manual manipulation skills when moving a large or heavy object or when a task is too complex to be performed by a single hand. Bi-manual picking up of objects to place or toss them on a conveyor belt are manipulation activities generated daily in the industry, particularly in logistics and warehouses. Such repetitive and physically demanding tasks are still done largely by humans for lack of similarly fast, precise, and robust bi-manual robotic systems. With nowadays booming of e-commerce, the needs for faster package handling solutions continue to increase. Hence, automation and robotics is the only viable solution as the current workforce cannot keep up with the growing industry demands. In current applications, however, robots usually use quasi-static approaches (with near zero relative velocities) to grab and release objects, mainly to avoid impacts.

Thus, this thesis proposes dynamic alternative solutions to quasi-static manipulation approaches with the goal to accelerate object handling operations and improve their energy efficiency. It focuses on bi-manual (dual-arm) robotic manipulation for its potential to mimic human dexterity. The thesis tackles the bi-manual manipulation problem from a coordination perspective of both motion and interaction forces to enable robust reaching, stable grabbing and cooperative manipulation tasks. It also considers smooth and dynamic transitions between non-contact and contact motion phases. To validate its approach, the thesis considers different scenarios with two types of bi-manual systems: a biped humanoid robot and a pair of fixed base robot manipulators.

The first part of the thesis starts by endowing the humanoid robot with balance and locomotion abilities needed for dual-arm cooperative tasks. It proposes a reactive locomotion controller that exploits the capture point dynamics to generate on-the-fly adjustable omnidirectional walking patterns that are consistent with balance constraints. In the second part, the whole body of the humanoid robot is controlled to accomplish bi-manual motion coordination and cooperative compliant manipulation tasks. The proposed approach relies on dynamical systems and exploits a shrinkable virtual object concept with its associated constraints to achieve robust coordination of the robotic hands with smooth tran-

sitions between non-contact and contact motion phases. The framework uses quadratic programming (QP) to generate interaction wrenches that are consistent with the contact constraints in order to stabilize the grasp and perform the desired manipulation tasks.

The third part of the thesis considers more dynamic interactions of the dual-arm system with the objects by allowing the grabbing and releasing of objects with non-zero relative velocities. Such an approach, besides speeding up the task, offers the possibility to expand the robot’s reach beyond its physical boundaries. Thus, a unified coordination framework based on modulated dynamical systems is proposed for reaching, grabbing with impact, and tossing objects in one swipe. It is based on modulated dynamical systems that ensure motion continuity and robustness throughout the task. The last part of the thesis extends the dynamic capabilities of the framework by enabling the precise tossing of objects onto a moving target carried by a conveyor belt. It uses a learned inverse throwing map within a kinematic-based bi-level optimization to determine feasible tossing parameters required by the precise positioning task. Moreover, it proposes and uses a model of the tossable workspace (set of all positions reachable by an object if tossed by the robot) to determine intercept locations that yield high probability of success.

Finally, the proposed approaches are validated both in simulation and on real robotic platforms, namely: the humanoid robot iCub and a pair of KUKA LBR (IIWA7 and IIWA14) robots. Kinetic comparisons with the classical pick-and-place strategy are conducted and the results show that the proposed swift pick-and-toss reduces the task duration and the energy expenditure.

Keywords: bi-manual manipulation, dynamic manipulation, fast grabbing, tossing, pick-and-toss, dual-arm robot, dynamical systems

RÉSUMÉ

LES humains ont de bonnes capacités de manipulation bi-manuelle. Ils peuvent coordonner sans effort le mouvement de leurs deux mains et exécuter avec succès une tâche donnée. Par exemple, les humains ont souvent recours à leurs capacités de manipulation bimanuelle lorsqu'ils veulent déplacer un objet lourd ou volumineux ou lorsqu'une tâche est trop complexe pour être exécutée par une seule main. La prise bimanuelle d'objets pour les placer ou les lancer sur un tapis roulant est une activité de manipulation générée quotidiennement dans l'industrie, notamment dans la logistique et les entrepôts. Ce genre de tâches répétitives et physiquement exigeantes sont encore effectuées en grande partie par des humains, faute de systèmes robotisés bimanuels tout aussi rapides, précis et robustes. Cependant, avec l'essor actuel du commerce électronique, les besoins en solutions de manutention rapide des colis ne cessent d'augmenter. L'automatisation et la robotique sont donc la seule solution viable, car la main-d'œuvre actuelle ne peut pas répondre aux exigences croissantes du secteur. Cependant, dans les applications actuelles, les robots utilisent généralement des approches quasi-statiques (avec des vitesses relatives proches de zéro) pour saisir et libérer les objets, principalement pour éviter les impacts.

Ainsi, cette thèse propose une approche dynamique comme alternative aux approches de manipulation quasi-statiques dans le but d'accélérer les opérations de manipulation d'objets et d'améliorer leur efficacité énergétique. Elle se concentre sur la manipulation robotique bimanuelle (à deux bras) pour son potentiel à imiter la dextérité humaine. La thèse aborde le problème de la manipulation bimanuelle d'un point de vue de la coordination des mouvements et forces d'interaction pour permettre des tâches d'atteinte robustes, de saisie stable et de manipulation coopérative. Elle considère également les transitions douces et dynamiques entre les phases de mouvement sans contact et avec contact. Pour atteindre son objectif, la thèse considère différents scénarios avec deux types de systèmes bimanuels: un robot humanoïde bipède et une paire de robots manipulateurs à base fixe.

La première partie de la thèse commence par doter le robot humanoïde des capacités d'équilibre et de locomotion nécessaires aux tâches coopératives à deux bras. Elle propose un contrôleur de locomotion réactif qui exploite la dynamique du point de capture pour générer à la volée des trajectoires de marche omni-

directionnels et ajustables qui sont cohérents avec les contraintes d'équilibre. Dans la deuxième partie, le corps entier du robot humanoïde est contrôlé pour accomplir des tâches de coordination de mouvements bimanuels et de manipulation coopérative compliant. L'approche proposée s'appuie sur des systèmes dynamiques et exploite un concept d'objet virtuel rétrécissable avec ses contraintes associées pour obtenir une coordination robuste des mains robotiques avec des transitions douces entre les phases de mouvement sans contact et avec contact. Le cadre utilise la programmation quadratique (QP) pour générer des forces d'interaction qui sont cohérentes avec les contraintes de contact afin de stabiliser la prise et d'effectuer les tâches de manipulation souhaitées.

La troisième partie de la thèse considère des interactions plus dynamiques du système à deux bras avec les objets en permettant leur saisie et relâche avec des vitesses relatives non nulles. Une telle approche, en plus d'accélérer la tâche, offre la possibilité d'étendre la portée du robot au-delà de ses limites physiques. Ainsi, un cadre de coordination unifié basé sur des systèmes dynamiques modulés est proposé pour atteindre, saisir avec impact et lancer des objets en un seul mouvement. Il est basé sur des systèmes dynamiques modulés qui assurent la continuité et la robustesse du mouvement tout au long de la tâche. La dernière partie de la thèse étend le cadre dynamique en permettant le lancer précis d'objets sur une cible mobile transportée par un tapis roulant. Elle utilise une carte apprise de lancer inverse dans le cadre d'une optimisation à deux niveaux basée sur la cinématique pour déterminer les paramètres de lancer réalisables requis par la tâche. De plus, il propose et utilise un modèle de l'espace de travail lançable pour déterminer les positions d'interception ayant une plus grande probabilité de succès.

Enfin, les approches proposées sont validées à la fois en simulation et sur des plateformes robotiques réelles, notamment : le robot humanoïde iCub et une paire de robots KUKA LBR (IIWA7 et IIWA14). Des comparaisons cinétiques avec la stratégie classique de pick-and-place sont effectuées et les résultats montrent que le pick-and-toss rapide proposé réduit la durée de la tâche et la dépense énergétique.

Mots Clés: manipulation bimanuelle, manipulation dynamique, saisie rapide, lancer, prise-et-lancer, robot a deux bras, systemes dynamiques.

*To my wonderful family, my beloved Lydie and our son Zoe,
for their love and endless support*

ACKNOWLEDGMENTS

Words will never suffice to express my sincere gratitude to many people who in some way or another contributed to the realization of this thesis.

First of all, my sincere and deepest gratitude goes to my thesis supervisor Prof. Aude Billard for the unique opportunity she gave me to pursue my PhD under her supervision. Aude invited me for an internship in her prestigious laboratory. I was thus able to gain real experience with many state-of-art robotic platforms and have the exceptional opportunity to meet scientific leaders in the field of robotics. I thank her for the time she spent reading my articles and my thesis, for all the advice and all the judicious suggestions. I can hardly express my gratitude for being extremely patient with me. Even at the lowest of my form, she found ways to push me and help me achieve my goals. Beyond being my thesis supervisor, Aude has been a true destiny helper.

I would like to extend my gratitude to the members of my Jury : Prof. Abderrahmane Kheddar, Prof. Tamim Asfour, Prof. Selman Sakar, and Prof. Herbert Shea for kindly accepting to examine my thesis and for taking time to provide valuable comments and suggestions on the earlier version of this thesis.

I would also like to thank the people who assisted me in all the administrative aspects with which I was confronted before and during my PhD at EPFL. In particular, I thank the secretary of our laboratory (LASA) Joanna Erfani, and the secretary of the ERDS doctoral program Corinne Lebet who, additionally, has never ceased to advise me and encourages me during difficult moments of my PhD. I also thank the administrator of the doctoral school Sandra Roux and the HR assistant Patricia Neuhaus Lakasi.

I was blessed to be surrounded by wonderful colleagues who made my time in and out of the office more joyful. I would like to thank my officemates Kunpeng Yao and David Gonon for all the long and pleasant discussions touching both science and other areas of life such as philosophy, religion, culture, etc. Kunpeng, with his discipline and sense of responsibility, often reminded me to stay focused on what was important. As for David, he helped me relativize things when I was worrying about too many aspects. I also thank my former officemate Sébastien Deuve, with whom we shared pleasant moments both in the office and outside the office. His help during my first repairs of the robot iCub was very valuable.

I would like to thank Iason Batzianoulis who, when I arrived at LASA, helped

me to integrate better and also for the moral support during the dark times. I thank Iason for being a gentle soul.

I would like to thank my former colleagues Sina Mirrazavi and Nadia Figueroa whom I had joined in the EU project CogIMon and with whom we traveled several times to participate in various meetings of the Consortium. Through their work, Sina and Nadia have been a great source of inspiration for me. I also thank Salman Faraji, who later joined the project and provided me invaluable advices on the control of humanoid robots.

I would like to thank my other former and current colleagues from LASA: Mahdi Khoramshahi, Lucia Pais Ureche, Murali Krishna Karnam, Andrew Sutcliffe, Ilaria Lauzana, Brice Platerrier, Anaïs Betsabeh Haget, Leonardo Urbano, Walid Amanhoud, Jacob Hernandez, Farshad Khadivar, Bernado Fichera, Michael Koptev, Lucas Huber, Laura Cohen, Konstantinos Chatzilygeroudis, Baptiste Bush, Enrico Eberhard, Diego Paez-Granados, Ahalya Prabhakar, Xiao Gao, Aradhana, Harshit Khurana, Yang Liu, Sthithpragya Gupta, Dominic Reber. Loïc, Albéric de Lajarte, Gautier Maxime Dimitri. I thank them for their sympathy and the various exchanges and interesting discussions that we had.

I had the opportunity to meet and closely collaborate with people outside EPFL. My special thanks to Hsiu-chin Lin for introducing me to projected inverse dynamics during the CogIMon project. I also thank Niels Dehio with whom we collaborated both in CogIMon and I.A.M. EU projects.

I had the grace to meet wonderful people in Lausanne. I am particularly thinking of Michèle Massy, who welcomed me to Lausanne and offered me more than just accommodation, a home. She has become like a second mother to me. Thanks also to the M'lopo family for their friendship.

I am infinitely grateful to my parents Norbert Bombile and Merveille Amisi for their unfailing love, dedication, sacrifice and endless support. Having always encouraged me to reach heights, they did everything to give me the best learning conditions. My dad was a big inspiration to me. His great intelligence and wisdom were a reference. I thank him for instilling in me the value of hard work and discipline. Although he passed away when I was just starting my PhD, he would have been extremely happy to see me finish it. I also thank all my brothers and sisters Prosper, Marylin, Christian, Yves, Thierry, Joel, Daniel, Benedicte and Benjamin for their love, support, and encouragement.

Last but not least, from the bottom of my heart, I thank my dear wife Lydie who accepted to accompany me during this long journey and had to endure long moments of absence alone with our son Zoé. Thank you for her love, her efforts for the family and her continued support. Thanks also to my Zoé for the happiness he brings. Watching him grow and flourish is a source of joy for me.

TABLE OF CONTENTS

Acknowledgments	vii
1 Introduction	1
1.1 Motivation	1
1.2 Background	3
1.2.1 Bimanual coordination	3
1.2.2 Transitions between coordination phases	4
1.3 Research Problem and Objectives	5
1.4 Approach	7
1.5 Contributions	8
1.6 Thesis Outline	10
1.7 Publications and Source Codes	11
2 Background	13
2.1 Introduction	13
2.2 Modeling	14
2.2.1 Robot Dynamics	14
2.2.2 Object dynamics	16
2.2.3 Task constraints	16
2.3 Coordination Control Problem	19
2.4 Related Work	20
2.4.1 Reactive Omnidirectional walking controller	20
2.4.2 Bimanual coordinated compliant manipulation on humanoid robot	22
2.4.3 Bimanual coordinated grabbing with impact and tossing of objects	24
2.4.4 Bimanual dynamic grabbing and tossing of objects onto a moving target	27
3 Capture-Point Based Balance and Reactive Omnidirectional Walking Controller	33
3.1 Introduction	34
3.2 Modeling and Problem Formulation	35
3.2.1 CoM-CP System Dynamics	35
3.2.2 Problem Formulation	37
3.3 Capture Point Based Reactive Omnidirectional Walking Controller	38
3.3.1 CoM-CP Prediction Model	38
3.3.2 CoM Average Velocity	39
3.3.3 Robot's Orientation	39
3.3.4 Self-generated Capture-Point Reference Trajectories	40
3.3.5 Footstep Positions and Orientations	41

3.3.6	Global Objective Function	44
3.4	Empirical Validation	48
3.4.1	Velocity Driven Omnidirectional Walking	48
3.4.2	Interaction Force based Reactive Walking	50
3.5	Conclusion and Future Works	53
4	Bimanual coordinated motion and forces for cooperative compliant manipulation	55
4.1	Introduction	55
4.2	Problem definition	56
4.3	Proposed Approach	58
4.3.1	Bimanual coordinated reach-to-grasp motion	59
4.3.2	QP-based bimanual cooperative manipulation	62
4.4	Validation	63
4.4.1	Bimanual coordinated reaching and manipulation task	63
4.4.2	Coordinated reaching and grasping of a moving object	69
4.4.3	Whole-body and object-level compliant interaction	70
4.4.4	Real robot bimanual object reaching and grabbing based on whole-body inverse kinematics	72
4.5	Conclusion	74
5	Dual-arm control for coordinated fast grabbing and tossing of an object	75
5.1	Introduction	76
5.2	Problem statement	77
5.3	Proposed Approach	78
5.4	Modulation-based Coordinated Control	82
5.4.1	Stability and convergence to attractors	82
5.4.2	Generation of impact velocity	83
5.4.3	Manipulation task	83
5.4.4	Tossing task	84
5.4.5	QP-based contact force generation	85
5.5	Empirical Validation	85
5.5.1	Fast object grabbing with impact	86
5.5.2	Pick-and-place vs. proposed pick-and-toss	91
5.6	Conclusion	95
6	Bimanual dynamic grabbing and tossing of objects onto a moving target	97
6.1	Introduction	98
6.2	Problem Definition and Proposed Approach	100
6.2.1	Robot-object motion constraints	101
6.2.2	Problem statement	102
6.2.3	Overview of proposed approach	103
6.3	Estimation of Feasible Throwing States	104
6.3.1	Learning an inverse throwing map	104
6.3.2	Optimal feasible release state: Concept	110
6.3.3	Optimal feasible release state: Solution	110
6.3.4	Generation of best feasible release states	112
6.4	Learning Tossable Space	114
6.5	Dual-arm Throwing Task Control	117
6.5.1	Dual-arm coordinated control	117
6.5.2	Estimation of the target's state-to-go	117
6.5.3	Motion Adaptation	119

6.6	Empirical Validation	123
6.6.1	Tossing object to target moving at constant velocities . . .	125
6.6.2	Tossing object on target moving with changing velocities . . .	127
6.6.3	Comparison of placing and tossing object onto moving target	135
6.7	Discussion and Conclusion	137
7	General Conclusion	139
7.1	Main contributions	139
7.1.1	Reactive omnidirectional walking motion generator	139
7.1.2	Cooperative compliant manipulation on humanoid robot	140
7.1.3	Coordinated fast grabbing and tossing of objects	140
7.1.4	Grabbing and tossing of objects on moving target	141
7.2	Limitations and Future Works	142
	Appendices	145
A	Appendices for Chapter 2	147
A.1	Computation of CoM wrench maps	147
A.2	Lumped Contact constraints matrix and vector	148
B	Appendices for Chapter 3	151
B.1	Hessian matrix and gradient vector of the QP	151
C	Appendices for Chapter 4	153
C.1	Inequality constraints for the whole-body controller	153
C.2	Definition of Matrix \mathbf{L}_{jk}	153
C.3	Stability and Convergence of the reach-to-grasp	153
D	Appendices for Chapter 5	155
D.1	Orientation control	155
D.2	Determination of impact direction	156
D.3	Proof of Proposition 1	157
D.4	Proof of proposition 2	157
D.5	Proof of Proposition 3	158
D.6	Joint limits	158
E	Appendices for Chapter 6	161
E.1	Jacobian of Inverse throwing map	161
E.2	Throwing objectives	163
	References	165
	Curriculum Vitae	181

INTRODUCTION

1.1 Motivation

NOWADAYS, although still largely dominated by the manufacturing and electronics industries, the use of robots is experiencing a rapid growth in the field of logistics, particularly in warehouses. This growth is essentially driven by the rise of e-commerce, which has currently an estimate of 2.14 billion digital buyers or 27.6% of the world population ([Bigcommerce, 2022](#)). Further reinforced by the recent covid-19 pandemic, which has confined billions of people and pushed them towards online purchases, e-commerce represented in 2021 a worldwide market of approximately 4.9 trillion U.S. dollar and is projected to grow by 50% over the next four years to reach 7.4 trillion U.S. dollar by 2025 ([Chevalier, 2022](#)).

However, this rapid growth imposes enormous challenges on the supply chain responsible for transporting the increasing number of products from sellers to consumers. More particularly, this rapid growth increases the need for faster and more flexible package handling solutions. In that regard, automation and robotics is the only viable option as the current workforce, although having better dexterity and flexibility, cannot keep up with the growing industry needs. Indeed, it is still common to find in the industry, for example in palletizing or sorting stations, humans carrying out bimanual picking up and placing or tossing of objects on conveyor belts or on pallets. Given the physical demands of these tasks associated with an aging population, particularly in the West and Asia-Pacific ([Nations et al., 2019](#)), manual operations are more and more giving way to robotic solutions.

However, current robots are far from showing the same level of dexterity as the humans they are supposed to replace or complement. Robots most often evolve in well-structured environments with precisely defined trajectories. Moreover, current robots use quasi-static manipulation approaches, whose interactions with objects happen at near zero relative contact and release velocities mainly to avoid impacts. In contrast, humans can operate in unstructured and dynamic environments. They can immediately adapt to perturbations or changes in their environment and effortlessly re-plan their motion to fulfill the desired task. Moreover, humans often use dynamic manipulation approaches and can

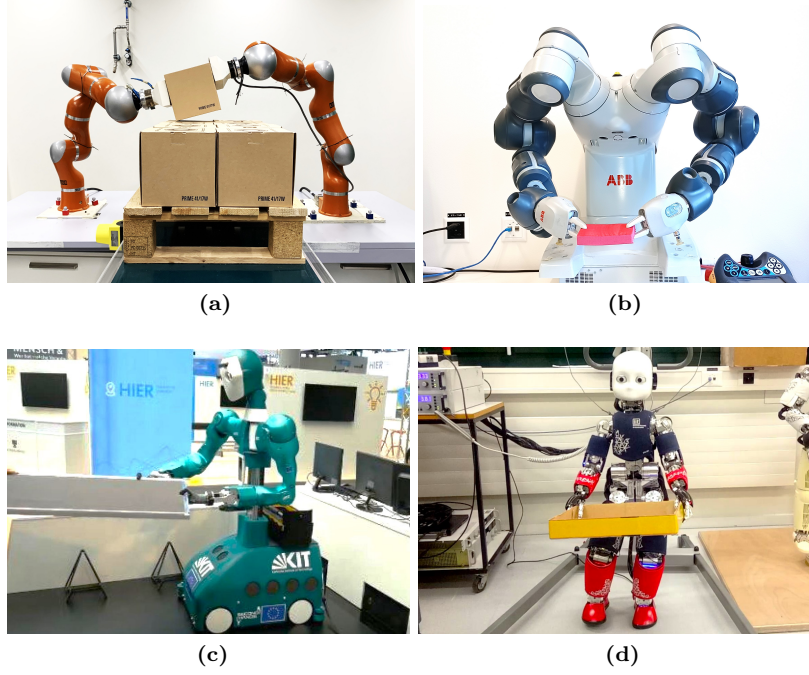


Figure 1.1: Examples of structural forms of the dual-arm robotic systems. (a) a pair of single-arm KUKA LWR robots as dual-arm system. (b) ABB YuMi robot as a semi-anthropomorphic dual-arm robot with fixed base. (c) Humanoid robot ARMAR-6 ([Asfour et al., 2018](#)) as a semi-anthropomorphic dual-arm robot with mobile base. (d) Humanoid robot iCub as full-anthropomorphic dual-arm robot

safely interact with objects at non-zero relative contact velocities thanks to their ability to predict the effects of impact. For instance, they can quickly grab an object by snatching it; they can pass it along by throwing it, etc. Furthermore, humans can effortlessly coordinate the motion of their two hands and execute successfully a given task that is too complex to be performed by a single hand. Having bimanual robots that exhibit such dynamic manipulation abilities with similar level of adaptability and robustness is important to address the increasing need for faster and more flexible package handling solutions. Moreover, if the robot manipulation skills are complemented with mobility, the application realm of the robots will be extended far beyond industrial settings.

Thus, this thesis aims to develop a framework that allows bimanual robotic systems to perform dynamic manipulation tasks and interact with objects at non-zero relative contact or release velocities. The purpose of this framework is to speed up and improve the energy efficiency of quasi-static manipulation approaches, beyond the mere optimization of the robot trajectories. It focuses on bimanual (dual-arm) robotic systems (see Figure 1.1) for their potential ability to perform manipulation tasks with human-like dexterity. An example of such a solution is the introduction of robotic bimanual fast grabbing with impact and tossing of parcels as an alternative to traditional pick-and-place operations.

1.2 Background

Devising a framework that can fulfill the bimanual manipulation objective outlined above is nontrivial. The dynamic manipulation feature requires the design of robot controllers that exploit dynamic characteristics of the tasks such as induced impacts or projectile dynamics to achieve the desired manipulation goals. Additionally, it requires a coordination control strategy as the robotic systems considered in the thesis are bimanual robots.

1.2.1 BIMANUAL COORDINATION

Bimanual manipulation is defined as a dual-arm coordination where the two arms are in physical interaction (Smith et al., 2012). In general, this physical interaction phase is preceded by a coordinated motion phase that ensures the reaching of the object before contact. The cooperative manipulation phase starts only once the contacts are established. During this phase, the interaction forces arising from the closed kinematic chain around the object need to be controlled to stabilize the grasp and to induce the desired motion of the object without exerting excessive stress on it.

Bimanual manipulation has been extensively studied; more detailed review on the subject can be found in (Smith et al., 2012; Wimböck et al., 2012; Caccavale and Uchiyama, 2016). Several methods have been proposed to address the motion coordination (Nakano et al., 1974; Uchiyama and Dauchez, 1992; Caccavale and Uchiyama, 2008) and the control of both motion and forces (Hsu and Su, 1992; Schneider and Cannon, 1992; Xi et al., 1996; Bonitz and Hsia, 1996a; Caccavale et al., 2008; Lin et al., 2018), etc. In the literature, the motion coordination during the reach-to-grasp and the cooperative manipulation phases are generally addressed separately, as the robotic system is characterized by two different dynamics: an unconstrained (free motion) and a constrained dynamics, respectively.

Motion coordination

Two main approaches have been proposed to address the motion coordination problem. The first known as master/slave (Nakano et al., 1974; Caccavale and Uchiyama, 2008) which later evolved to the leader/follower approach is where one robot, the master (leader), is aware of the desired motion and executes it independently. The other robot, the follower, generates its respective motion either from the state of the leader, from kinematic constraints of the task, or from the interaction forces through the object. The second is a symmetry type approach, where all robots have equal importance; they are all aware of the task's goal and execute their actions accordingly (Uchiyama and Dauchez, 1992).

The reach-to-grasp problem has been often addressed with motion planning. For instance, using algorithm based on rapidly-exploring randomized tree (RRT)

in (Gienger et al., 2008; Vahrenkamp et al., 2009, 2010), and (Lertkultanon and Pham, 2018) with a certified-complete planner. The bimanual reach-to-grasp of moving objects, which require spatial and temporal motion coordination, has been also addressed with dynamical systems-based approaches in (Salehian et al., 2016a; Mirrazavi Salehian et al., 2017a, 2018b), where a concept of virtual object was used to impose coordination constraints to the two robotic arms.

Cooperative manipulation

Works addressing the cooperative manipulation phase usually assumed an already grasped object and consider the interaction forces while trying to achieve a desired behavior of the object. To control simultaneously the robot interaction forces and motion during the cooperative manipulation task, hybrid force/position (Raibert et al., 1981) and impedance control (Hogan, 1984) and their variants have been widely used (Hsu and Su, 1992; Schneider and Cannon, 1992; Hsu, 1993; Khatib et al., 1996; Xi et al., 1996; Bonitz and Hsia, 1996a,b; Caccavale et al., 2008). In hybrid force/position, the manipulation task is decomposed, whenever possible, into two orthogonal subspaces where motion and forces are controlled independently. Whereas in impedance control, the interaction forces are controlled through a compliant motion governed by a virtual mass-damper-spring dynamics. Other frameworks that deals with constrained robotic systems such as projected inverse dynamics (Aghili, 2005) has also been employed for cooperative manipulation in (Lin et al., 2018). The dual-arm dynamics is projected onto a constrained and an unconstrained subspaces orthogonal to each other. This allows the motion of the object to be controlled without affecting the contact forces. Quadratic programming (QP) was used to compute joint torque commands that enforce explicitly the unilateral force constraints of the contacts. QP was also used in (Bouyarmane et al., 2017) to compute constraint-consistent contact forces in a bimanual cooperative manipulation by a humanoid robot.

In this thesis, we also consider bimanual coordination on a biped humanoid robot. Such a robot is characterized by a floating base and limited contacts with the ground. This increases the complexity of the control problem. Besides the previous dual-arm coordination problem, it additionally requires ensuring the balance of the robot and coordination between the robotic arms and the floating base.

1.2.2 TRANSITIONS BETWEEN COORDINATION PHASES

The body of works dealing independently with the free motion and the cooperative manipulation phases need to switch between controllers depending on the manipulation phases. This is problematic for applications that involve frequent transitions between these two motion phases and that require coordination at

contact to ensure a stable grasp. The transition can be quasi-static when it is smooth, or dynamic when it is abrupt, for instance, when grabbing with impact or when releasing objects by tossing them.

Quasi-static transitions

In the case of quasi-static transition, the integration problem of the two motion phases is less challenging. For instance, a smooth transition between the free motion phase (controlled by a dynamical system) and a cooperative manipulation phase (controlled by a projected inverse dynamic controller) was proposed in (Mirrazavi Salehian et al., 2018a). The approach uses a probabilistic estimate of the contact confidence to switch between the two controllers. In (Amanhoud et al., 2019), a smooth phase transition was achieved using a modulated dynamical system that unifies the free and constrained motion phases by projecting the contact forces into the motion space.

Dynamic transitions

Dynamic transitions, usually exploited in robotic locomotion dealing with hybrid dynamics, are now drawing an increasing interest in robotic manipulation. Their resulting impacts and tossing must be controlled not only to generate the desired object behavior but also to maintain the impact-induced velocity jumps and forces within safe limits for both the robot and the object. To tackle such a problem, an impact-aware controller applied to the bimanual swift grabbing of a box was presented in (Wang et al., 2020); and more recently a model predictive control-based extension applicable to deformable objects was proposed in (Dehio et al., 2022). These methods allow to control the impact by computing the limits of feasible contact velocities for rigid and deformable objects, respectively, and by providing ways to enforce these constraints for robot safety.

1.3 Research Problem and Objectives

Robust coordination of both motion and forces is critically important for faster and more flexible bimanual object manipulation. Although current bimanual approaches have made enormous progress, they are in general inadequate for faster object handling applications with dynamic transitions between motion phases. The few recent works, that have considered dynamic transitions, focused only on grabbing with impact and did not consider a dynamic release such as tossing. Moreover, none of these works leverage the induced bimanual impacts to speed up the post-grabbing process. Indeed, the directions of their applied bimanual impact velocities, normal to the contact surfaces of the object, are opposite to each other and therefore they have no resultant velocity applied to the object.

Our objective in this thesis is to develop a coordination control framework that unifies the control of the free and constrained motion phases. It should ensure robust quasi-static and more importantly dynamic transitions between the manipulation phases. Ultimately, the framework should speed up the robotic object grabbing and release process beyond the mere optimization of the robot trajectories. Optimizing the trajectories of the robot, for example by using a minimal time control strategy (Hassan et al., 2022), makes it possible to accelerate the desired task within the hardware limits of the robot. However, beyond such approach, we seek to accelerate the process by reducing the duration of the grabbing and release phases where, in traditional approaches, the robots often decelerate to reach near zero relative velocities. The acceleration strategy pursued in this thesis focuses essentially on the transitions between motion phases, when picking up and releasing objects, and seeks to shorten these phases by moving from quasi-static to dynamic transitions.

To evaluate and validate the proposed approach, this thesis considers two types of dual-arm robotic systems: a biped humanoid robot and a pair of fixed base robotic manipulators. Practically, the main objective can be divided into the following sub-objectives:

1. Given the structure of a bipedal humanoid robot, develop a balance and locomotion controller that easily integrates with bimanual cooperative tasks,
2. Develop, for a humanoid robot, a whole-body control-based bimanual cooperative compliant manipulation with smooth transitions between motion phases
3. Develop a dual-arm coordination framework that enables fast grabbing and tossing of objects, and
4. Extend the dynamic capabilities of the fast grabbing and tossing framework with the ability to precisely toss objects onto a moving target.

The first sub-objective is to equip the humanoid robot with balance and locomotion capabilities to support cooperative bimanual tasks with a human or another robot. The second allows to unify the free-motion and constrained phases with smooth transitions in a more complex coordination framework. Such a coordination, beyond the two arms, involves the whole body of the robot which must remain balanced throughout the task. The last two sub-objectives concern the coordination framework for dynamic manipulation and its application modality to the precise positioning of objects on a moving target.

1.4 Approach

In this thesis, we tackle the bimanual manipulation problem from a coordination perspective of both motion and interaction forces to enable robust reaching, stable grabbing, and cooperative manipulation tasks. We adopt a motion generation approach based on time-invariant dynamical systems to ensure robustness to perturbations and adaptability to changes in the environment. We adopt a force generation strategy using quadratic programming (QP) to optimally distribute the task effort between the two hands and to enforce explicitly the unilateral force constraints at the contacts.

Dynamical systems

The motion generation problem is generally addressed through motion planning. Consider the reach-to-grasp task, for instance, classical motion planning approaches address such problem by finding a collision-free path from the initial robot state to the goal state. Then, specifying the execution of the plan through the definition of the the robot state evolution along the path (Kavraki and LaValle, 2016). Although such an approach works well in static environment with no perturbations, it might become inadequate when there are uncertainties and changes in the environment as re-computation of a new plan might be required.

Thus, dynamical systems-based approaches have emerged as interesting alternative solutions to classical motion generation problem. They naturally unify the planning and execution phases (Selverston, 1980; Kelso, 1995; Schaal et al., 2000; Ijspeert et al., 2001). This feature allows to provide fast re-planning ability and adaptation to changes in the environment (Rimon and Koditschek, 1992; Feder and Slotine, 1997; Khansari-Zadeh and Billard, 2011; Kronander and Billard, 2016; Salehian et al., 2016b; Khoramshahi and Billard, 2019; Figueroa and Billard, 2022). Dynamical systems can be modulated to locally shape the generated motion flow for specific purposes. For example, to avoid obstacles (Khansari-Zadeh and Billard, 2012; Huber et al., 2019b), comply with external signals (Sommer et al., 2017) or avoid impacts at contact (Mirrazavi Salehian and Billard, 2018).

A motion generator based on DS represents a state-dependent map that instantly provides, for any given state in the operational domain, the desired direction and magnitude of the state evolution toward the goal state. Such motion generator inherits adaptability and robustness from the DS as they can swiftly adopt new plans toward the goal if the robot state is perturbed or the state of the environment changes. For these reasons, this thesis adopts motion generation approaches based on dynamical systems.

1.5 Contributions

The main contribution of this thesis lies essentially in providing a generic and unified dynamical systems-based motion generation framework that enables dynamic manipulation tasks with desired transitory states¹ such as desired impact and tossing. The proposed framework is generic because it can be formulated in task space as well as in joint space with first or second-order dynamics. Its unified nature stems from the framework’s ability to handle both free and constrained motions and to robustly transition between them in cooperative manipulation settings. For instance, when considering dynamic transitions, this thesis offers a novel approach to control a dual-arm robotic system for quickly grabbing and tossing packages onto a fixed or moving target. The considered application is inspired by existing challenges in logistics, particularly in depalletizing or sorting facilities, where bimanual tossing of packages onto a conveyor belt is common and mainly performed by humans. Given the physical demands of such tasks, the proposed dynamic grabbing and tossing can be a useful alternative for faster and more reliable package handling operations. We demonstrate the accuracy, repeatability, and robustness of the proposed framework. We also show that it is faster, and requires less energy than the traditional pick and place strategy.

Specifically, the thesis contributions can be described as follows:

- **Reactive omnidirectional walking motion generator**

We propose a motion generation algorithm to address the dynamic balance and locomotion problem of a biped dual-arm humanoid robot. We propose a capture-point-based walking controller that generates on-the-fly omnidirectional walking patterns for a biped robot and stabilizes the robot around them. The proposed algorithm, formulated within model predictive control (MPC) framework, exploits concurrently the center of mass (CoM) and capture-point (CP) dynamics. It allows the online generation of the CoM reference trajectory and the automatic generation of footstep positions and orientations in response to a given velocity to be tracked, or a disturbance to be rejected by the robot while accounting explicitly for different walking constraints. For instance, in order to cope with disturbance such as a push, the proposed controller not only adjusts the position of the Center of Pressure (CoP) within the support foot but can also induce at least one step with appropriate length allowing thus to maintain the stability of the robot. We validate the proposed algorithm through simulations and experiments on a real humanoid robot. We demonstrate the reactive generation and adaptation of the robot’s footsteps respectively in

¹Transitory states are states defined by desired position and velocity to be satisfied simultaneously. Unlike attractors where the system can converge to and settle, transitory states are not equilibrium points, and therefore the system can only transit through such states.

omnidirectional velocity tracking tasks and in a human-robot cooperative object transportation task.

- **Cooperative compliant manipulation on humanoid robot**

We propose an approach to achieve stable bimanual reach-to-grasp and compliant manipulation of an object by a humanoid robot. We use dynamical systems and exploit a concept of shrinkable virtual objects to achieve motion coordination by imposing virtual constraints on the robot’s hands. Moreover, the shrinkage of the virtual object ensures smooth transitions from virtual constraints in free motion to real constraints when the object is grasped. Also, the controller computes contacts-consistent optimal wrenches that stabilize the grasp and achieve desired manipulation tasks. We validate the proposed solution on the humanoid robot iCub.

- **Coordinated fast grabbing and tossing of objects**

We propose a unified motion generator enabling a dual-arm robotic system to grab and toss an object in one swipe. Unlike classical approaches that grab the object with quasi-zero contact relative, the proposed approach is able to grasp the object while in motion. The continuous coordinated control of reaching, grabbing, and tossing motion is achieved by combining a sequence of time-invariant dynamical systems in a single control framework. We control the contact forces following the impact so as to stabilize the dual-arm grip on the object. We demonstrate the effectiveness of the proposed framework in fast pick-and-toss tasks with a real dual-arm system. We show that such a dynamic grabbing and release of objects not only speeds up the pick-and-place process but also reduces energy expenditure. To the best of our knowledge, this is the first dual-arm system to demonstrate a fast grabbing and tossing of an object in one swipe.

- **Grabbing and tossing of objects on moving target**

We propose a control strategy that enables a dual-arm robotic system to pick up an object and throw it precisely on a moving target carried by a conveyor belt. Given the predicted trajectory of the moving target along the conveyor belt, we determine the best intercept position of the object and the target within the throwing workspace² of the robot. We also determine feasible throwing parameters (release position and release velocity) necessary to land on the intercept position. The proposed approach achieves this, by learning from the non-linear projectile dynamics of a thrown object an inverse throwing map that encodes a minimum release speed strategy. It then combines this inverse map with a robot-dependent optimization framework to generate, for the dual-arm system, kinematically feasible release states associated with a given tossing task. Moreover, building upon the obtained release states feasibility algorithm,

²set of all positions reachable by an object if tossed by the robot.

we propose a modeling approach for the throwing workspace (that we denoted by tossable workspace) of the dual-arm robot. Our approach yields a closed-form model of the tossable workspace, which allows fast prediction of the reachability of a given tossing target before initiating any action on the robot side. Besides the theoretical development, we provide experimental validation of the proposed framework both in simulation and on a real robotic system.

1.6 Thesis Outline

Chapter 2 Background

This chapter presents an overview of dual-arm manipulation. It introduces the fundamental concepts of dual-arm manipulation and presents modeling aspects of dual-arm systems from a motion and force perspective. Finally, it presents a brief review of works related to our thesis’s objectives and concludes with a summary of the thesis’s proposed approach to achieve its objectives.

Chapter 3 Balance and Reactive Omnidirectional Walking Controller

This chapter presents our method to generate on the fly omnidirectional stable walking patterns, for a biped robot, in response to velocity to be tracked, or a disturbance to be rejected by the robot. It describes our formulation that combines both the CoM and CP dynamics in the same MPC framework to generate the CoM motion, and the footstep positions and orientations consistent with the walking constraints. It validates our proposed approach by showcasing its application in human-robot cooperative tasks where the robot needs to react to human intentions.

Chapter 4 Cooperative compliant manipulation on humanoid robot

This Chapter presents an approach to achieve stable bimanual reach-to-grasp and compliant manipulation of an object by a humanoid robot. We use dynamical systems and exploits a concepts of shrinkable virtual object to achieve motion coordination by imposing virtual constraints to the robot’s hands. Moreover, through its shrinkage it ensures smooth transition from virtual constraints in free-motion to real constraints when the object is grasped. Also, the controller computes contacts-consistent optimal wrenches that stabilize the grasp and achieve desired manipulation tasks. We validate the proposed solution on the humanoid robot iCub.

Chapter 5 Dual-arm control for coordinated fast grabbing and tossing of an object

This chapter presents our method for generating coordinated motion for a dual-arm system to reach, grab and toss an object in one swipe. It provides a first-order formulation of the modulated DS that allows the coordination of a pair of robots and the local shaping of their motions to achieve desired transitory states (impact and tossing) when grabbing and releasing an object. It provides different coordinated applications along with their convergence proofs. It also describes the quadratic programming (QP)-based strategy to generate grasping forces that stabilize the object during the cooperative manipulation phase. Finally, it presents the experiments validating the proposed methods and discusses the results and their implications.

Chapter 6 Bimanual dynamic grabbing and tossing of objects onto a moving target

This chapter builds upon the DS introduced in Chapter 5 and presents our control strategy that now allows a dual-arm system to toss an object precisely on a moving target. It provides a model of the throwing situation that includes the free-flying dynamics of the object once released. It describes our approach to learning the inverse throwing map and its combination with the dual-arm kinematics to determine feasible release states associated with a given tossing task. It presents our proposed strategy to model the tossable workspace of the robot in closed form. It also shows how we complement the framework presented in Chapter 5 with an adaptation strategy to achieve interception of the tossed object with the moving target in the presence of motion perturbations. Finally, it presents the experimental results validating the proposed dynamic object positioning strategy.

Chapter 7 Conclusion

This chapter summarizes the main contributions of this thesis. It also discusses its assumptions and limitations, and provides possible research directions to improve the work of this thesis.

1.7 Publications and Source Codes

Most of the work of this thesis is extracted from our peer-reviewed conference and journal publications. The material of Chapter 3 was previously published in (Bombile and Billard, 2017b), the material of Chapter 5 was published in (Bombile and Billard, 2022) and the material of Chapter 6 is, at the time of publication of the thesis, under review in the Journal of Robotics and Autonomous

Systems. The videos of the experiments are available on-line following the links indicated in Table 1.1.

Table 1.1: Links of the videos of the experiments of each chapter

Chapter 3
https://youtu.be/1rPaMXguPDQ
Chapter 4
https://youtu.be/odoaZ80h7IA
Chapter 5
https://youtu.be/CeLoqXdPIOU
Chapter 6
https://youtu.be/pRSHH1866ug

The codes developed dual-arm robotic system and the humanoid robots are also available on line and their links are indicated in Table 1.2.

Table 1.2: Links to the source codes

Chapter 3
https://github.com/epfl-lasa/biped-walking-controller
Chapter 4
https://github.com/epfl-lasa/icub_whole_body_task_controller
Chapter 5 and Chapter 6
https://github.com/epfl-lasa/iam_dual_arm_control

BACKGROUND

This chapter presents an overview of dual-arm manipulation. The first Section introduces fundamental concepts of dual-arm manipulation. It is followed by the mathematical background Section, where modeling aspects of dual-arm manipulation are presented. These are namely: the dynamics of dual-arm systems (for both the fixed base and the humanoid robot), the object dynamics, and the model of the bimanual manipulation task with its related force and motion constraints. After defining the bimanual coordination control problem in the third Section, the Chapter presents a brief review of works related to our thesis objectives in its fourth Section. Finally, the Chapter concludes with a summary of our approaches proposed to achieve the thesis objectives.

2.1 Introduction

Bimanual manipulation is defined as a dual-arm coordination where the two arms are in physical interaction ([Smith et al., 2012](#)). In general, this physical interaction phase is preceded by a coordinated motion phase that ensures the reaching of the object before contact. The cooperative manipulation phase that ensures the grasping and handling of the object starts only once the contacts are established.

Depending on the type of interaction at the contact, two types of cooperative manipulation can be distinguished: 1) cooperative manipulation with fixed grasp points, where the object and the manipulator are rigidly linked by bidirectional contact constraints, and 2) cooperative manipulation with contact points or surface, where the contact constraints are unilateral, which makes rolling or sliding possible ([Mason and Salisbury Jr, 1985](#); [Bicchi and Kumar, 2000](#); [Okamura et al., 2000](#)).

Three typical models of contacts are often used to explain the transmission of forces to the object: frictionless contacts, frictional contacts and soft contacts. In the first case, there is no transmission of tangential force components, whereas in the second case, the transmission of tangential components is possible. In the last case, frictional torque around the normal component can be transmitted as the contact area could be large due to the softness of the contact ([Li and Kao, 2001](#)).

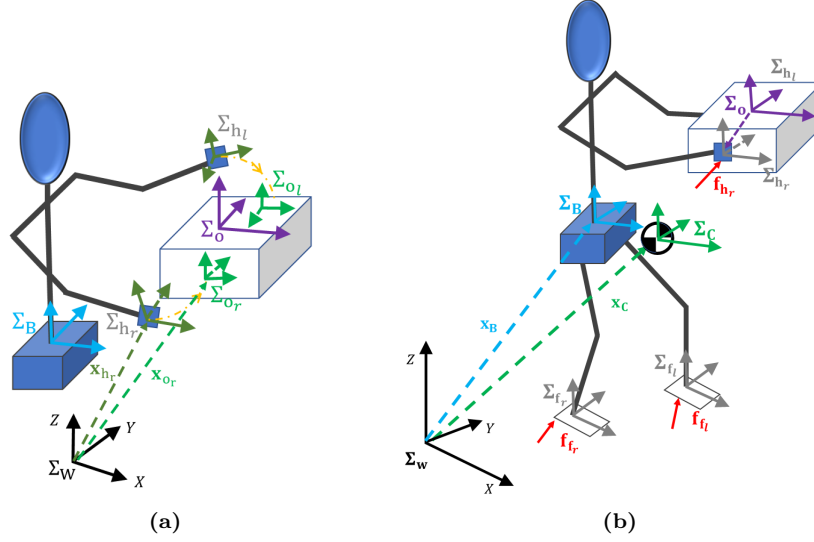


Figure 2.1: Illustration of the two situations of the considered during bimanual manipulation tasks. (a): Reaching-to-grasp with hands in free motion dynamics. (b): Cooperative manipulation with hands in constrained dynamics. Σ_w and Σ_o are the world and the object frame. Σ_{h_l} and Σ_{h_r} denote respectively the robot’s left and right hands frames, while Σ_{o_l} and Σ_{o_r} denote respectively their desired grasping configuration on the object side.

2.2 Modeling

Modeling a robot consists of deriving the set of mathematical expressions that describe the geometric and temporal properties of the robot’s motion (Murray et al., 1994). Thus, by deriving the dynamic model of a dual-arm robot, we seek to relate its motions to the forces/torques causing or resulting from these motions (Craig, 2005).

For dual-arm with fixed-base, the overall dynamics can be obtained by stacking individual robotic arms’s dynamics (Caccavale and Uchiyama, 2016). However, for a dual-arm system with a floating base, the dynamics is derived as a whole because of the dynamic coupling introduced by the floating base (Bouyarmane et al., 2017). The model can also be augmented with the dynamics of the manipulated object as in (Khatib, 1995) or (Dehio et al., 2018).

2.2.1 ROBOT DYNAMICS

Consider the bimanual robotic systems illustrated in Figure 2.1, which also shows a free-motion and cooperative manipulation tasks. The dynamic model of a robot interacting with its environment is generally expressed as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}_r^\top \mathbf{\Gamma} + \mathbf{J}_e^\top(\mathbf{q}) \mathbf{f}_e \quad (2.2.1)$$

where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n^D \times n^D}$ and $\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n^D}$ are the inertia matrix and the vector of centrifugal, Coriolis, and gravity forces of the robotic system. $\mathbf{q}, \dot{\mathbf{q}} \in \mathbb{R}^{n^D}$,

$\ddot{\mathbf{q}} \in \mathbb{R}^{n^D}$ denote the configuration (position), velocity and acceleration vectors, respectively. $\mathbf{J}_e(\mathbf{q}) \in \mathbb{R}^{n^e \times n^D}$ and $\mathbf{f}_e \in \mathbb{R}^{n^e}$ are respectively the Jacobian matrix of the interacting end-effector(s) and the vector of associated wrenches. $\mathbf{S} \in \mathbb{R}^{n^D \times n^D}$ is a selection matrix of the actuated components of the joint torques $\boldsymbol{\tau} \in \mathbb{R}^{n^D}$.

Depending on the type of dual-arm systems considered in this thesis, whether it is a biped humanoid robot or a pair of fixed-base arms, we have the following specificities:

Case of humanoid robot:

- $n^D = n^B + n^r$ where n^B and n^r denote the number of the floating base DoFs and robot actuated joints, respectively,
- $\mathbf{q} = \left[\begin{array}{c} \{\mathbf{x}_B^\top, \boldsymbol{\phi}_B^\top\} \\ \mathbf{q}_r^\top \end{array} \right]^\top$, where $\mathbf{x}_B \in \mathbb{R}^3$ and $\boldsymbol{\phi}_B \in SO(3)$ are the position and orientation vector of the robot floating base, respectively. $\mathbf{q}_r \in \mathbb{R}^{n^r}$ is the vector of actuated joint positions of the robot.
- $\mathbf{J}_e(\mathbf{q}) = \left[\begin{array}{cc} \mathbf{J}_h(\mathbf{q}) & \mathbf{J}_f(\mathbf{q}) \end{array} \right]$, where $\mathbf{J}_h(\mathbf{q}) \in \mathbb{R}^{12 \times n^D}$ and $\mathbf{J}_f(\mathbf{q}) \in \mathbb{R}^{12 \times n^D}$ are the Jacobian matrices of the robot hands and feet, respectively, $\mathbf{f}_e = \left[\begin{array}{c} \mathbf{f}_h \\ \mathbf{f}_f \end{array} \right]$ with $\mathbf{f}_h \in \mathbb{R}^{12}$ and $\mathbf{f}_f \in \mathbb{R}^{12}$ are the wrench vectors of the robot hand and feet, respectively.
- $\mathbf{S}_r \triangleq \left[\begin{array}{cc} \mathbf{0}_{n^B \times 6} & \mathbf{0}_{n^B \times n^r} \\ \mathbf{0}_{n^r \times n^B} & \mathbf{I}_{n^r \times n^r} \end{array} \right]$ and $\boldsymbol{\Gamma} = \left[\begin{array}{c} \mathbf{0}_{n^B} \\ \boldsymbol{\Gamma}_r \end{array} \right] \in \mathbb{R}^{n^D}$ where $\boldsymbol{\Gamma}_r$ is the actuated joint torque of the humanoid robot

Case of fixed base dual-arm robot

- $n^D = n^L + n^R$, where n^L and n^R denote the number of DoFs of the left and the right robot arm, respectively,
- $\mathbf{q} = \left[\begin{array}{cc} \mathbf{q}_L^\top & \mathbf{q}_R^\top \end{array} \right]^\top \in \mathbb{R}^{n^D}$ and $\mathbf{M}(\mathbf{q}) = \text{diag} \left\{ \mathbf{M}_L(\mathbf{q}_R) \quad \mathbf{M}_R(\mathbf{q}_R) \right\}$ is block-diagonal. $\mathbf{M}_L(\mathbf{q}_R) \in \mathbb{R}^{n^L \times n^L}$ and $\mathbf{M}_R(\mathbf{q}_R) \in \mathbb{R}^{n^R \times n^R}$ are inertial matrices of the left and right robotic arms, respectively.
- $\mathbf{J}_e(\mathbf{q}) = \mathbf{J}_h(\mathbf{q}) = \text{diag} \left\{ \mathbf{J}_{hL}(\mathbf{q}_R) \quad \mathbf{J}_{hR}(\mathbf{q}_R) \right\} \in \mathbb{R}^{12 \times n^D}$ corresponds to the block-diagonal Jacobian matrix of the dual-arm interacting end-effector(s) and $\mathbf{f}_e = \mathbf{f}_h$ is the vector of associated wrenches.
- $\mathbf{S}_r \triangleq \mathbf{I}_{n^D \times n^D}$ and $\boldsymbol{\Gamma} = \left[\begin{array}{c} \boldsymbol{\Gamma}_L \\ \boldsymbol{\Gamma}_R \end{array} \right]$, where $\boldsymbol{\Gamma}_L$ and $\boldsymbol{\Gamma}_R$ are joint torque of the left and the right robot arm, respectively,

2.2.2 OBJECT DYNAMICS

The dynamics of the object assumed to be rigid, of mass m_o and inertia \mathcal{I}_o can be written as

$$\mathbf{M}_o \dot{\mathbf{v}}_o + \mathbf{b}_o = \mathbf{f}_o \quad (2.2.2)$$

where $\mathbf{M}_o = \begin{bmatrix} m_o \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathcal{I}_o \end{bmatrix} \in \mathbb{R}^{6 \times 6}$ and $\mathbf{b}_o = \begin{bmatrix} -m_o \mathbf{g} \\ \boldsymbol{\omega}_o \times \mathcal{I}_o \boldsymbol{\omega}_o \end{bmatrix} \in \mathbb{R}^{6 \times 1}$ denote the inertia matrix and the Coriolis and gravity forces vector, respectively. $\mathbf{v}_o \triangleq \begin{bmatrix} \dot{\mathbf{x}}^o \\ \boldsymbol{\omega}^o \end{bmatrix} \in \mathbb{R}^6$ denotes the object velocity twist, and $\dot{\mathbf{x}}^o \in \mathbb{R}^3$ and $\boldsymbol{\omega}^o \in \mathbb{R}^3$ are the object's linear and angular velocity, respectively. $\mathbf{f}_o \in \mathbb{R}^6$ is the object effective wrench acting at Σ_o as a result of the applied bimanual wrenches.

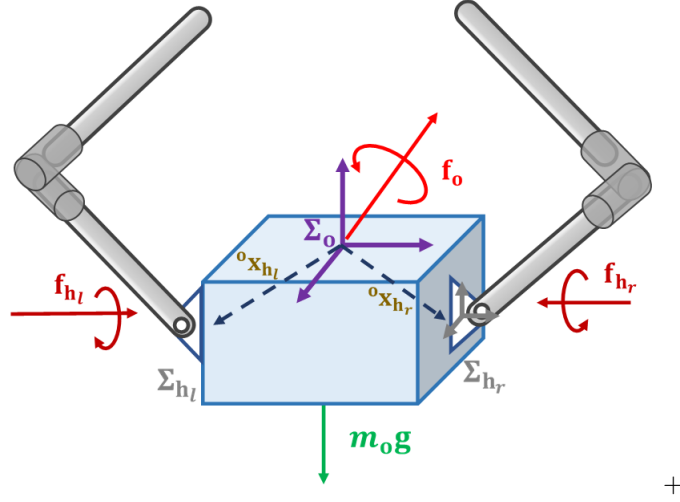


Figure 2.2: Grasp situation in a bimanual manipulation setting of an object

2.2.3 TASK CONSTRAINTS

BIMANUAL COOPERATIVE CONSTRAINTS

When the object is stably grasped, the dynamic coupling between the robot and the object can be written as

$$\mathbf{f}_o = \mathbf{G}_o \mathbf{f}_h \quad (2.2.3)$$

where $\mathbf{G}_o \in \mathbb{R}^{6 \times 12}$ denotes the bimanual grasp matrix (Murray et al., 1994; Caccavale and Uchiyama, 2016). In our case, it is given by

$$\mathbf{G}_o = \begin{bmatrix} \mathbf{G}_{o_L} & \mathbf{G}_{o_R} \end{bmatrix} \quad \text{with} \quad \mathbf{G}_{o_i} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \\ [\mathbf{x}_{oo_i}]_{\times} & \mathbf{I}_3 \end{bmatrix} \quad (2.2.4)$$

where $\mathbf{x}_{oo_i} \in \mathbb{R}^3$ is the relative position between the grasping point located at Σ_{o_i} and the object frame Σ_o . The vector $\mathbf{f}_h \triangleq [\mathbf{f}_{h_L}^\top \mathbf{f}_{h_R}^\top]^\top \in \mathbb{R}^{12}$, where $\mathbf{f}_{h_L} \in \mathbb{R}^6$ and $\mathbf{f}_{h_R} \in \mathbb{R}^6$ are the wrenches of the left and right robot hands, respectively.

Force decomposition

The bimanual grasp creates a redundancy in actuation ($\mathbf{f}_h = [\mathbf{f}_{h_L}^\top \mathbf{f}_{h_R}^\top]^\top \in \mathbb{R}^{12}$ while $\mathbf{f}_o \in \mathbb{R}^6$), which leads to infinite combinations of \mathbf{f}_{h_L} and \mathbf{f}_{h_R} that can produce the same value of \mathbf{f}_o . This can be seen by solving Eq. (2.2.3) for the wrench $\mathbf{f}_h^* = [\mathbf{f}_{h_L}^{*\top} \mathbf{f}_{h_R}^{*\top}]^\top$ that the dual-arm ought to apply to produce a desired object wrench \mathbf{f}_o^* . Hence, we obtain

$$\mathbf{f}_h^* = \mathbf{G}_o^+ \mathbf{f}_o^* + (\mathbf{I} - \mathbf{G}_o^+ \mathbf{G}_o) \mathbf{f}_{h_o} \quad (2.2.5)$$

where the matrix $\mathbf{G}_o^+ \in \mathbb{R}^{12 \times 6}$ is a generalized pseudo-inverse of \mathbf{G}_o and $\mathbf{f}_{h_o} \in \mathbb{R}^{12 \times 1}$ represents the end-effector wrenches that lie in the null-space of \mathbf{G}_o . The term $\mathbf{G}_o^+ \mathbf{f}_o^*$ represents the component of \mathbf{f}_h^* that induces the object's motion and it is known as the effective (external) wrench applied on the object. The term $(\mathbf{I} - \mathbf{G}_o^+ \mathbf{G}_o) \mathbf{f}_{h_o}$, with $\mathbf{f}_{h_o} \in \mathbb{R}^{12 \times 1}$ lying in the null-space of \mathbf{G}_o , represents the internal wrench components of \mathbf{f}_h applied on the object. \mathbf{f}_{h_o} does not induce object motion but exerts stress on the object. However, it can create deformation if the object is non-rigid. \mathbf{f}_{h_o} is the applied wrench component responsible to maintain the grasp; it must be consistent with the contact constraints (unilateral or bilateral). The analysis of internal forces has also been studied based on other models such as virtual truss (Chiacchio et al., 1991) or virtual linkage (Williams and Khatib, 1993).

Motion decomposition

The bimanual grasp also creates motion constraints between the dual-arm and the object. The mapping between their respective velocities can be obtained by applying the virtual work principle to (2.2.5) and exploiting the duality between wrench and velocity twist. Thus, we obtain

$$\begin{cases} \mathbf{v}_o &= (\mathbf{G}_o^+)^T \mathbf{v}_h \\ \mathbf{v}_{o_{int}} &= (\mathbf{I} - \mathbf{G}_o^+ \mathbf{G}_o)^T \mathbf{v}_h \end{cases} \quad (2.2.6)$$

where $\mathbf{v}_h \triangleq [\mathbf{v}_{h_L}^\top \mathbf{v}_{h_R}^\top]^\top \in \mathbb{R}^{12}$ is a vector gathering the velocity of the left and right robotic hands. $\mathbf{v}_o = (\mathbf{G}_o^+)^T \mathbf{v}_h \in \mathbb{R}^6$ represents the effective object velocity twist describing the global displacement of the object (velocity twist of Σ_o), and $\mathbf{v}_{o_{int}} \triangleq (\mathbf{I} - \mathbf{G}_o^+ \mathbf{G}_o)^T \mathbf{v}_h \in \mathbb{R}^{12}$ represents the velocity of the object's

contact frames (Σ_{o_i}) relative to the object's frame Σ_o . For non-rigid object, the components of $\mathbf{v}_{o_{\text{int}}}$ are the velocities that describing the object deformation as they result from contact wrenches that stress the object.

BALANCE OF HUMANOID ROBOT

The balance task is achieved when the dynamic wrench is equal to the sum of the gravity and contact wrenches. The overall dynamic effect of the multi-body humanoid robot can be deduced from the linear and angular momentum of its center of mass (CoM). This yields a reduced dynamics known as centroidal dynamics (Orin et al., 2013). Thus, assuming the humanoid robot interacts with its environment only through the hands and feet, the balance task can be written at the motion and forces level as

$$\dot{\mathbf{h}} = \mathbf{A}_q \ddot{\mathbf{q}} + \dot{\mathbf{A}}_q \dot{\mathbf{q}} = \mathbf{G}_{c_f} \mathbf{f}_f + \mathbf{G}_{c_h} \mathbf{f}_h + \mathbf{f}_{\text{gvt}} \quad (2.2.7)$$

where $\mathbf{h} = [\dot{\mathcal{P}}^\top \ \dot{\mathcal{L}}^\top]^\top$ represents the centroidal momentum, with $\mathcal{P} \in \mathbb{R}^3$ and $\mathcal{L} \in \mathbb{R}^3$ denoting the linear and angular momentum of the entire robot, respectively. $\mathbf{A}_q \in \mathbb{R}^{6 \times n}$ is the centroidal moment matrix (Orin and Goswami, 2008). $\mathbf{G}_{c_f} \in \mathbb{R}^{6 \times 6}$ and $\mathbf{G}_{c_h} \in \mathbb{R}^{6 \times 6}$ represent wrench transformation matrices that map respectively the feet and hands contact wrenches to the frame attached to the CoM (see Appendix A.1 for their derivation). $\mathbf{f}_{\text{gvt}} \triangleq [0, 0, -m_r g, 0, 0, 0]^\top \in \mathbb{R}^6$ denotes the gravity forces acting on the CoM frame.

Contact constraints

When a biped humanoid robot is standing on its feet, the force components of \mathbf{f}_{f_l} and \mathbf{f}_{f_r} that are normal to the ground (contact surface) must be unilateral (i.e. forces that can only push against and not pull the ground). Moreover, if the contacts are assumed to be static (non sliding) and frictional, all force components of \mathbf{f}_{f_l} and \mathbf{f}_{f_r} that are tangent to the ground must stay within the friction cone. For balance, the center of pressure of all contact points must remain within the convex hull formed by the contacts.

These contact constraints are non-linear mainly because of the friction cone. However, when linearized, these constraints can be lumped and written as (see Appendix A.2 for more details)

$$\mathbf{C}_f \mathbf{f}_f \leq \mathbf{d}_f \quad (2.2.8)$$

where $\mathbf{C}_f \in \mathbb{R}^{n^f \times 12}$ and $\mathbf{d}_f \in \mathbb{R}^{n^f}$ denote the matrix and vector of the linearized constraints, respectively. n^f is the number of linearized contact constraints of the robot feet.

2.3 Coordination Control Problem

After modeling the dual-arm system, the object, and analyzing the force and motion constraints resulting from the grasp, the next step is to address the coordination control problem. This problem consists of computing the actuated torque $\mathbf{\Gamma}$ such that:

- the two robot hands reach the object to grasp in a coordinated manner,
- once the object reached, the robot applies wrenches that achieve stable grasp and the desired manipulation task,
- the dual-arm system complies with the hardware and task constraints throughout the task.

The first two points concern the coordinated reach-to-grasp and the cooperative manipulation control problems. As stated in Section 1.2.1, the reach-to-grasp problem is traditionally solved using motion planning. The cooperative control problem is solved by computing (directly or indirectly) bimanual wrenches that the robot should apply to produce stable grasps and perform the desired tasks without excessive internal wrenches. How well the object is grasped depends on the internal wrenches. Also, too high internal wrenches could damage the robot or the object.

The third point ensures that the computed values of the torque $\boldsymbol{\tau}$ yield motion and forces that are feasible for the robot. This implies satisfying constraints such as the actuator and joints limits, collision avoidance. Moreover, there is an additional challenge when considering a humanoid robot as in this thesis. Unlike fixed-base robots, a biped humanoid robot is characterized by a floating base and unilateral contact constraints with the ground. Such a robot can successfully perform the desired manipulation task only if the robot is balanced or has stable gait. Thus, cooperative applications of humanoid robots are often presented along with walking or whole-body balance controllers.

In the next Sections, we will briefly review related work on bimanual manipulation. We start by reviewing works on balance and locomotion with an emphasis on reactive walking generators or controllers. Afterward, we present works relevant to bimanual coordination control and compliant manipulation on a humanoid robot. Then, we review bimanual coordination studies that are related to dynamic transitions between manipulation phases, especially when grabbing and releasing objects. In light of this review, we summarize our approach to conclude the Chapter.

2.4 Related Work

2.4.1 REACTIVE OMNIDIRECTIONAL WALKING CONTROLLER

The balance and gait stability of a bipedal humanoid robot refer to its ability to stand and walk without falling (Kajita and Espiau, 2008). As shown in Eq. (2.2.7), the balance of bipedal robots arises from the balance of forces and moments. In the literature, different concepts have been proposed to ensure balance and gait stability of bipedal robots. The most widely used is the Zero Moment Point (ZMP), defined as the point of the ground where horizontal moments (tipping moments) are zero (Vukobratović and Stepanenko, 1972). The ZMP assumes a horizontal ground where it corresponds to the centre of pressure (CoP). The ZMP must stay within the support polygon for the robot to be balanced.

In ZMP-based approaches, the footsteps of the robot are generally planned in advance. This reduces the walking control problem to the generation of walking patterns: CoM trajectories that are consistent with the robot dynamics and that satisfy the desired ZMP associated with the footsteps (Kajita et al., 2003; Pratt et al., 2006). For instance, walking patterns were generated in (Kajita et al., 2003) using preview control that is able to reproduce the anticipatory motion of the CoM over the footstep placements. Wieber (2006) proposed instead a linear model predictive controller (LMPC) to enforce explicitly the ZMP constraints.

To deal with large perturbations such as pushes, collisions, or tripping, various reactive stepping approaches have been proposed in the humanoid community. Besides moving the ZMP within the support polygon as in (Kajita et al., 2006; Diedam et al., 2008; Morisawa et al., 2007), the robot could either accelerate its angular momentum (through trunk or upper limbs motions), or take a step in order to prevent a fall or to come to a stop (Pratt et al., 2006; Stephens and Atkeson, 2010; Urata et al., 2011). In such a case, the “*capture-point*” (CP) (Pratt et al., 2006) also called “*extrapolated center of mass*” (Hof, 2008), defined as the point on the ground where a biped robot should step to in order to come to a complete stop, has proven to be very effective. In (Koolen et al., 2012), the CP was also used to define a concept for the stability analysis of legged locomotion, namely the “*N-step capturability*”, which was validated in (Pratt et al., 2012).

Apart from push recovery, Engelsberger et al. (2011) proposed a control approach of the unstable dynamics of the CP to facilitate the generation of walking patterns. A backward recursive method to generate dynamically-consistent reference trajectories of the CP required by the task was proposed in (Engelsberger and Ott, 2012b). However, the stability constraints were not always satisfied by the computed ZMP. The latter needed to be projected onto the support polygon, yielding thus some discontinuities in the controller output.

To address this problem, [Krause et al. \(2012\)](#) proposed a formulation of the CP controller within an MPC framework. In [\(Griffin and Leonessa, 2016\)](#), the 3D extension of the CP called “*divergent component of motion*”(DCM) was controlled using MPC for the adjustment of footstep positions and orientations in the presence of disturbances. In [\(Shafiee-Ashtiani et al., 2016\)](#), a CP-based MPC approach is used for push recovery. The ZMP is controlled to stay within the support polygon while the CP is steered towards its desired location by modulating the Centroidal Moment Pivot (CMP) through the angular momentum. This approach was extended in [\(Shafiee-Ashtiani et al., 2017\)](#) in order to include the adjustment of footstep positions while walking.

Although benefiting from the robustness inherent to the CP control, the above MPC based works still rely on predefined footstep placements. Consequently, they are not suited for reactive walking situations, where the footsteps cannot be planned beforehand but have to be determined on-the-fly. To address the reactive walking problem, inspired by the formulation of continuous adaptation of the footsteps strategy in [\(Diedam et al., 2008\)](#), [Herdt et al. \(2010b\)](#) proposed a ZMP-based walking algorithm with automatic generation of footsteps placements in response to translational velocities. The main idea is to formulate the footstep placements as decision variables to be determined by the optimization process based on velocity requirements. An extension to rotational velocity was proposed in [\(Herdt et al., 2010a\)](#). However, because of the non-linearity introduced by the rotation in that formulation, the authors suggested a predetermination of the orientation, hence, limiting the reactive ability to the translation. To overcome this challenge, a quasi-linear ZMP-based approach to automatic generation of the orientation was proposed in [\(Bombile, 2015a\)](#). In [\(Naveau et al., 2017\)](#), nonlinear model predictive control (NMPC) was proposed for the automatic generation of both footstep positions and orientations.

SUMMARY AND THESIS’S RELATED CONTRIBUTION

In this Section [2.4.1](#), we have presented a brief overview of reactive walking pattern generators with emphasizes on ZMP and MPC-based controllers for their ability to enforce explicitly walking constraints. As outlined above, despite successful achievement of reactive omnidirectional walking, the works outlined above were based on the ZMP control. In such approaches, the generation of walking pattern generators generally operate in open loop as tracking the ZMP with the CoM’s acceleration as one of the state variables often leads to problematic feedback systems.

Thus, in Chapter [3](#), this thesis proposes instead an approach to generate reactive omnidirectional walking patterns based on the control of the CP to leverage its inherent robustness. With the CP being a linear combination of CoM’s position and velocity, implementing state feedback systems based on the CP is simpler. Moreover, to achieve reactive omnidirectionality, unlike the NMPC approach [\(Naveau et al., 2017\)](#), we formulated and solve our problem

as a piece-wise linear problem by focusing on the discrete orientation of the footsteps instead of the continuous rotation of the CoM (hip).

2.4.2 BIMANUAL COORDINATED COMPLIANT MANIPULATION ON HUMANOID ROBOT

Humanoid robots with their anthropomorphic structure have the potential ability to reproduce humans bimanual manipulation skills. Their combined manipulation and mobility capabilities extends largely their workspace and make them suitable to operate in various type of environments and numerous applications. As stated in Section 1.3, we seek to realize versatile bimanual manipulation tasks with a humanoid robot. More specifically, we consider a coordinated reach to grasp task and compliant manipulation of an object. To accomplish this, considering the humanoid robot particular dynamics, the robot must be balanced and the two arms must be coordinated both in motion and force to ensure compliant interaction.

As outlined in Section 1.2.1, several methods have been proposed to address the motion coordination (Nakano et al., 1974; Uchiyama and Dauchez, 1992; Caccavale and Uchiyama, 2008) and the control of both motion and forces (Hsu and Su, 1992; Schneider and Cannon, 1992; Xi et al., 1996; Bonitz and Hsia, 1996a; Caccavale et al., 2008; Lin et al., 2018), etc. Most of these approaches, however, consider dual-arm robotic systems with fixed bases. A biped humanoid robot is characterized by a floating base and limited contacts with the ground. While the mobility of the floating base enlarges the robot’s workspace, it also increases the complexity of the control. Besides the previous dual-arm coordination problem, it additionally requires the coordination between the robotic arms and the floating base.

In general, for a robotic arm on a mobile-base, the coordination problem between the arm and its base is usually handled with redundancy resolution approaches, for instance by creating hierarchy between the arm and the mobile-base tasks (Khatib et al., 1996, 1999), or through admittance coupling between the base and the arm’s end-effector (Erhart et al., 2013). Unlike robots on wheeled mobile bases, legged humanoid robots require active control of their balance due to their limited and unilateral contact with the ground. Thus, the bimanual approaches outlined above could be used on a legged humanoid robot provided that the robot is balanced and there is coordination between its arms and the floating base.

In that regards, whole-body control approaches can meet simultaneously these additional requirements (Khatib et al., 2004; Sentis and Khatib, 2006). In (Garcia-Haro et al., 2019) a dual-arm manipulation task was integrated in a passivity-based whole-body torque controller on the humanoid robot TORO. The sum of the CoM and interaction wrenches is distributed between the feet end-effector for the balance task, while the manipulation wrench is distributed

between the two robotic hands. Although compliant behavior with good tracking performance has been achieved, this framework does not explicitly enforce inequality constraints such as the joint limits and the unilateral contact constraints necessary for the balance. Such a limitation is addressed when whole-body control is formulated as optimization problem, in particular quadratic programming (QP) (Salini et al., 2010; Kanoun et al., 2011; Saab et al., 2013; Escande et al., 2014; Dai et al., 2014; Feng et al., 2015; Herzog et al., 2016). It allows multiple task constraints including balance and contacts stability to be explicitly enforced. For instance, QP-based whole-body control was used in (Bouyarmane et al., 2017) to perform bimanual manipulation of an object with a humanoid robot HRP-4. The task-space force constraints and the hardware constraints were explicitly enforced. In (Hoffman et al., 2018), a QP-based whole-body inverse kinematics was used to balance a humanoid robot iCub performing a bimanual manipulation task of a tray to stabilize an empty cup on it.

Most of the above works, however, assume an object already grasped by the robot and focus on the post-contact manipulation phase. The free motion phase and its transition towards the contact phase has been rarely considered. In (Mirrazavi Salehian et al., 2018a), such transition was considered. The approach uses DS-based multi-arm coordination framework (Mirrazavi Salehian et al., 2017a) that employs a virtual object to achieve robust coordinated reaching of the object by the robotic arms. Upon contact, the system smoothly switches to the projected inverse dynamics controller (Lin et al., 2018) to handle the constrained manipulation scheme. In (Amanhoud et al., 2019), a dual-arm coordination using DS with force modulation was proposed to smoothly integrate the motion phases without switching between controllers. Although these approaches have successfully integrated both free and constrained manipulation phases, they were designed for fixed-base robots and were not concerned with the balance problem. In (Rakita et al., 2019), transitions between bimanual free and constrained motion phases were considered in a share control context with the humanoid robot HUBO. The dual-arm robot follows the human operator’s movements with the goal to assist and ease the user’s task completion. Although validated on the humanoid robot, this framework was limited to the robot’s kinematics and consider neither the control of interaction forces nor the balance task.

SUMMARY AND THESIS’S RELATED CONTRIBUTION

In this Section 2.4.2, we presented bimanual coordination methods for both free and constrained motion with a focus on quasi-static (smooth) transitions between them. We narrowed our focus on bimanual manipulation by a humanoid robot and highlighted the fact that previous works, in general, have only partially addressed the dual-arm coordination problem on such robots. On the one hand, works that considered the balance in addition to the coordination problem assumed already grasped objects and focused on the cooperative manipulation task. On the other hand, works that considered the reach-to-grasp and the co-

operative manipulation phases including the transitions between them were not concerned with the balance problem.

Thus, in Chapter 4 of this thesis, we propose an approach that considers the whole-body control to achieve bimanual motion coordination and cooperative compliant manipulation on a humanoid robot while enforcing the robot’s balance. Our approach considers all bimanual motion phases including the transitions between them. The motion generation is based on DS and the force generation uses QP to ensure, respectively, robust coordination and constraint-consistency of interaction forces throughout the task.

2.4.3 BIMANUAL COORDINATED GRABBING WITH IMPACT AND TOSSING OF OBJECTS

One of the objectives of the thesis is to develop a bimanual dynamic manipulation framework to speed up object-handling operations. We thus propose a bimanual coordination framework for grabbing with impact and tossing objects in one swipe. To achieve this objective, in addition to generating the desired impact and tossing motion, the framework must enforce the coordination of the two robotic arms throughout the task. A poorly coordinated system would lead to uncontrolled impact or tossing. The dual-arm coordination methods described in the previous Section, namely (Mirrazavi Salehian et al., 2018a) or (Amanhoud et al., 2019), can be extended to this case with dynamic transitions between motion phases. However, the problem of achieving grabbing with impact or tossing of objects is particular in the sense that these tasks require that desired transitory states expressed in terms of desired position and velocity be simultaneously satisfied (during contact or at the release time, respectively). Moreover, the control of impacts and tossing is necessary not only to generate the desired object behavior but also to guarantee the feasibility of the task and safety of the robotic system. For instance, the impact-induced velocity jumps and forces must remain within safe limits for both the robot and object.

ROBOTIC IMPACT GENERATION

In the literature, besides robotic applications such as batting or hitting balls, the generation of intentional impacts to achieve manipulation goals has also been reported. For instance, in (Konno et al., 2011) intentional impacts were generated using sequential quadratic programming (SQP) to induce high impulsive forces to break an object with a humanoid robot HOAP. In (Rijnen et al., 2019), the control problem of a robotic system subject to inelastic impact was considered. The reference spreading control law was proposed to mitigate the effects of impact-induced state jumps in the feedback command. The problem of constraining within the robot’s safe limits the state jumps induced by hard impacts was considered in (Wang and Kheddar, 2019) and addressed using a QP-based approach. A similar problem considering soft impact was addressed

in (Dehio and Kheddar, 2021). Although the above-mentioned works offer effective approaches to generating controlled impact, they were not concerned by the coordination problem as they considered the motion generation of a single robotic arm. Few studies have considered the dual-arm impact generation problem. The work (Wang and Kheddar, 2019) was extended to dual-arm systems and an impact-aware controller applied to the bimanual swift grabbing of a box was presented in (Wang et al., 2020). More recently a model predictive control-based extension applicable to deformable objects was proposed in (Dehio et al., 2022). These methods allow controlling the impact by computing the limits of feasible contact velocities for rigid and deformable objects, respectively, and by providing ways to enforce these constraints for robot safety.

While these methods allow the safe generation of dual-arm impacts necessary for the dynamic grabbing, they do not address the dual-arm arm tossing problem that is considered in this thesis to achieve the dynamic release of objects.

ROBOTIC THROWING

Indeed, robotic throwing or tossing is a dynamic manipulation task that offers the possibility of positioning objects within or outside the physical workspace of a robot, and saving time and energy when compared to non-dynamic manipulation methods (Mason and Lynch, 1993). Since the work on this subject by Hove and Slotine (1991), several researchers have investigated robotic throwing. Apart from the dynamic aspects, research on robotic throwing is also motivated by potential applications in industry. As first suggested by Frank et al. (2006), robotic throwing can serve as a more flexible alternative transportation method for certain types of products.

Robotic throwing has been demonstrated in the literature using different types of robotic systems that can be classified into three main categories: throwing with specialized devices such as 1-DoF or 2-DoF (degrees of freedom) launching mechanisms (Mason and Lynch, 1993; Frank et al., 2006; Frank, 2008a,b; Senoo et al., 2008; Ichinose et al., 2008; Frank et al., 2009; Mori et al., 2010), throwing by industrial robots (August et al., 2010; Zhang et al., 2012; Zeng et al., 2020; Gallant, 2020; Raptopoulos et al., 2020) and throwing by humanoid robots (Kim et al., 2008; Satıcı et al., 2016). In the case of industrial robots, August et al. (2010), for instance, used the KUKA KR-16 robot to demonstrate accurate throwing of a tennis ball to a target located approximately 2.5 m away. More recently, the TossingBot (Zeng et al., 2020) used a UR5 robot to throw various objects with different shapes.

From the motion generation perspective, there are three main phases characterizing robotic throwing: an acceleration phase, a release phase, and a free-flight phase (Ruggiero et al., 2018). The first two phases are directly controlled by the robotic system, which must transport the object in a prehensile or non-prehensile manner to the desired release state (defined in terms of position and velocity) before releasing the object. This represents a trajectory generation problem with

desired intermediate or transitory states, often addressed through motion planning (Sintov and Shapiro, 2015; Zhang et al., 2012; August et al., 2010), where a deceleration phase after release is often included to ensure the feasibility of the movement throughout the task until the robot stops. There are also methods based on trajectory optimization (Okada et al., 2015; Gallant, 2020; Hassan et al., 2022) and methods based on optimal control (Asgari and Nikoobin, 2021), where the torques necessary to bring the robot to the desired state are directly computed, ensuring the dynamic feasibility of the task. During the release phase following the acceleration phase, the time required for the robot to release the object may create uncertainty concerning the actual release state. To address this problem, August et al. (2010), for example, proposed a solution based on extending the movement of the robot beyond the release position and following a parabolic trajectory to model the projectile (thrown object); (Okada et al., 2015) optimized the release position such that the sensitivity of the landing position to changes of the release position was minimized.

SUMMARY AND THESIS’S RELATED CONTRIBUTION

In this Section 2.4.3, we have presented an overview of approaches related to dynamic bimanual grabbing and tossing of objects and highlighted the challenges associated with the generation of controlled impacts and tossing motion. We have discussed the state-of-art solutions that have been proposed to address them. We have outlined the fact that despite their ability to generate safe impacts for the robot, works dealing with bimanual grabbing with impacts have only been limited to this task. They did not consider combining it with other bimanual manipulation tasks such as placing or tossing objects. Also, works that considered throwing were not confronted with the problem of coordination since they used special robotic devices, single-arm robots or parallel robots.

Thus, in Chapter 5, this thesis proposes a manipulation task that goes beyond dynamic grabbing and includes a dynamic release (throwing) of the objects by a bimanual robotic system. For motion generation, unlike planning-based methods which are less reactive and prone to spatial and temporal perturbations, we adopt a DS-based approach. More specifically, we use modulated dynamical systems that allow local shaping of the motion flow to achieve the desired task objectives. The idea of locally modulating DS has been previously used for instance in (Khansari-Zadeh and Billard, 2012) and (Huber et al., 2019a) to avoid obstacles, in (Kronander et al., 2015) for incremental learning, or in (Salehian and Billard, 2018) to avoid impact by achieving stable contact with a surface. In Chapter 5, however, we use the local modulation to intentionally generate, in a coordinated manner, specified impact velocities at desired locations on the grabbing surfaces of an object. Moreover, we show how to adapt the local modulation to achieve other coordinated manipulation tasks such as object lifting and tossing.

2.4.4 BIMANUAL DYNAMIC GRABBING AND TOSSING OF OBJECTS ONTO A MOVING TARGET

The last objective of this thesis is to extend the dynamic grabbing and tossing framework with the ability to precisely toss objects onto a moving target. Tossing an object onto a moving target using a robotic system is challenging. It requires a solution to the following main sub-problems: 1) Finding a feasible intercept position for the thrown object to meet the moving target. 2) Finding feasible throwing parameters (release position, release speed, and direction). 3) Generating the motion of the robot to pick up the object and successfully reach the desired release state on time, such that the thrown object intercepts the target at the desired location.

Achieving such an objective requires solving an interception problem between the moving target and the thrown object. Finding a valid intercept point is straightforward; such a point lies along the path of the moving target within the throwable workspace of the robot (the robot’s extended reachable workspace when throwing objects). As this point must be determined beforehand, on the target side, it requires estimation and prediction of the trajectory. On the robot side, it requires learning the throwable workspace.

In general, an interception task, whether performed by a single arm or a multi-arm system shares similar challenges with robotic catching (Burridge et al., 1995; Lynch and Mason, 1999; Bätz et al., 2010; Kim et al., 2014; Schill and Buss, 2018; Dong et al., 2020; Satici et al., 2022), batting or hitting of flying objects (Anderson, 1988; Acosta et al., 2003; Senoo et al., 2006; Lai and Tsay, 2011; Mülling et al., 2013; Serra et al., 2016; Jia et al., 2019), or juggling (Aboaf et al., 1989; Buhler et al., 1990; Schaal and Atkeson, 1993; Lynch and Black, 2001; Akbarimajd and Ahmadabadi, 2007; Reist and D’Andrea, 2012; Serra et al., 2017; Poggensee et al., 2020). Unlike these tasks, in this thesis, the interceptor is not the robot but rather the thrown object, whose final motion phase is governed by projectile dynamics. For the thrown object to reach its target, the throwing parameters must be appropriate as the thrower can no longer correct the trajectory of the object once it is released.

Determining throwing parameters is not a trivial problem. The object is subjected to gravity, and to nonlinear aerodynamic forces and phenomena that depend on its shape, its speed, and the environment (air density, pressure, etc.). The object can have a complex movement combining translation and rotation around its center of mass, which may not be at its geometric center, depending on the mass distribution. Moreover, following the determination of the feasible throwing parameters corresponding to the desired intercept point, the motion of the robot must be generated and adapted such that the object is not thrown too early or too late to achieve the desired interception.

Thus, in addition to robotic throwing literature, we briefly review a body of work on robotic interception tasks, with a focus on how the intercept position

(configuration) was determined and how the robot motion was generated.

ROBOTIC THROWING

In the previous Section, we mentioned that robotic throwing is characterized by an acceleration phase, a release phase, and a free-flight phase (Ruggiero et al., 2018). Focusing on the motion generation, we reviewed a number of works related to the first two phases. We here complement the review with an emphasis free-flying motion phase which is governed by projectile dynamics.

Given that no corrective action is possible after release of the object, the throwing task accuracy depends on how well the throwing parameters are determined, which in turn depend on the modeling approach used for the free-flying object. In the literature, there are two main modeling approaches for throwing motion, with a third between them. The first approach relies on an analytical model of the object’s free-flying dynamics, with or without nonlinear aerodynamic phenomena; the second approach is data-based. For instance, the authors of (August et al., 2010) used a simple ballistic motion to determine the throwing parameters, neglecting aerodynamic forces. Example of approaches that considered additionally the Newton drag forces can be found in (Frank, 2008b). In a batting application of free-flying objects, which can be seen as an simultaneous catching and throwing task, Jia et al. (2019) considered the Magnus effect Sturek et al. (1978) and proposed a closed-form expression approximating the solution of the resulting nonlinear dynamics. Although physics-based analytical approaches can easily generalize to different conditions and objects, their prediction accuracy depends on knowledge of the object’s physical properties and aerodynamic phenomena that occur. The latter are generally difficult to estimate, as discussed in (Mason and Lynch, 1993).

Estimation of throwing parameters has also been addressed from a learning perspective. In an early work along this line, Aboaf et al. (1988) directly exploited the object–target landing error in throwing tasks to learn the correct throwing parameters without modeling the underlying dynamics. Other example of learning-based approach can be found in (Kober et al., 2011; Kim and Doncieux, 2017). Although these approaches provide good accuracy, they do not generalize well to conditions other than those they were trained for. Approaches combining both physics and learning-based solutions have also been proposed to leverage their respective strengths. Such a hybrid approach was proposed for instance in (Zeng et al., 2020); to improve the success of throwing tasks, solution of a physics-based model was complemented with data-based components that learned the “residual physics” (unknown and unmodeled dynamics not captured by the physics-based model). The authors used deep neural networks to directly learn the residual physics in the control space, considering the synergy between grasping and tossing.

Although, this approach was successfully applied in tossing various small objects, the feasibility problem of the tossing parameters has only been partially

considered with the release speed as main throwing parameter. In (Zeng et al., 2020) for instance, the release angle was set to 45 degrees and the release position was cylindrically constrained at a fixed height and fixed distance relative to the robot’s base. Thus, there a need to consider the full release state particularly for dual-arm systems as considered in this thesis.

ROBOTIC INTERCEPTION

Robotic interception consists of approaching a moving target to match its position and velocity in the shortest possible time (Mehrandezh et al., 1999). This problem has been widely investigated from different perspectives in the literature. The cited works on catching, batting, and juggling addressed the interception problem. For slowly moving targets or targets with long-term predictable trajectories such as objects moving on a conveyor belt, the solution can be cast into the general prediction planning execution (PPE) framework (Sharma et al., 1992; Mikesell and Cipra, 1994). In this framework, the motion of the target is predicted. The robot’s motion to an intercept or rendezvous location along the target trajectory is planned and the robot’s motion is executed. To address uncertainty and prediction errors, these steps can be repeated until interception, leading to an active PPE process (APPE). Interception of objects on conveyor belts was addressed with such a framework in (Holland et al., 1979) or (Mo and Liu, 1985) using a vision system. Allen et al. (1993) demonstrated a more reactive vision-based hand–eye system with movement rates similar to human movement to track and grasp a moving model train. Over four decades, this topic has been addressed from different perspectives. Optimal control of tasks with robot dynamics and constraints was considered in (Park and Lee, 1992), the optimal choice of interception point in (Croft et al., 1998), time-optimal considerations of tasks in (Croft et al., 1995), or the grasp reachability of the target in (Akinola et al., 2021). These works considered the interception problem using a single robot. In this thesis, however, the joint motion of two robotic systems must be considered when addressing the interception problem.

Robotic interception with dual-arm systems has been investigated. Previous work by our group (Salehian et al., 2016a) addressed the problem of robustly reaching moving objects with multi-arm systems. In that approach, generation of coordinated robot motion to perform reaching was based on dynamical systems and used a virtual object. A virtual object was also used in determination of intercept points by predicting object progress using a forward model. To ensure kinematic feasibility for the robots, the intercept points were determined along the intersection between the predicted object motion and the robot reachable space modeled with a Gaussian mixture model (GMM). Using this approach, catching a flying rod was demonstrated in (Mirrazavi Salehian et al., 2017b) and reaching for various car parts was demonstrated in (Mirrazavi Salehian et al., 2018b).

Although dual-arm coordinated interception tasks including grabbing and

catching flying objects have been successfully performed, a post-grabbing or post-catching task such as a placing or tossing task has not been considered. Moreover, unlike previous works where the interceptor (robot) was fully controllable throughout the task, this thesis considers a task where the interceptor (the thrown object) is only partially controllable.

TOSSABLE WORKSPACE

Robotic throwing extends the robot workspace beyond its physical boundary. The tossable workspace is the set of all positions (within and outside the boundaries of the physical workspace) reachable by a given object if thrown by the robot. Unlike the normal robot reachable workspace, which depends only on robot joint configuration, the tossable workspace also depends on the kinematic and dynamic characteristics of the robot, and the properties (inertia, size, aerodynamic characteristics) and desired landing orientation of the object, making modeling difficult. Few studies have tackled the estimation problem of the tossable workspace of a robot. For instance, [Gallant \(2020\)](#) proposed an approach for determining the maximum throwing reach of robots with kinematic and dynamic feasibility constraints. A trajectory optimization-based solution was proposed, with parameterized trajectories including cubic splines, polynomial functions, and Fourier series. The maximum throwing reach of 2-DoF, 3-DoF, and 5-DoF robots was determined. Although this approach can be a step toward estimating the tossable workspace, it is not explicitly determined. More recently, [Asgari and Nikoobin \(2021\)](#) proposed an indirect solution-based optimal control approach to estimate the maximum set of points to which robotic manipulators can throw an object. They called such set the "maximum throw-able" workspace. They modeled the throwing trajectory using the simple ballistic motion and applied it as a moving boundary conditions to optimize the release speed and angle. While this approach could successfully estimate the feasible throw-able workspace of 2 DoF planar and spatial robots, as highlighted by the author themselves, it comes with high computation burden which makes it difficult for online usage.

SUMMARY AND THESIS'S RELATED CONTRIBUTION

In this Section [2.4.4](#), we reviewed works related to robotic interception of moving targets by thrown objects. We discussed works robotic throwing with an emphasis on modeling the free-flying dynamics of a thrown object for the determination of the throwing parameters. We saw that physics-based models provide good generalization but have limited accuracy, whereas data-driven models have good accuracy but limited generalization. Hence, to leverage the strengths of both approaches, in Chapter [6](#) of this thesis, we propose to use a hybrid modeling approach where a throwing map is learned from a parameterized physics-based model of the free-flying dynamics. Unlike works that consider

only the release speed as the main throwing parameter, our approach considers the full release state (position and velocity) and ensures their kinematic feasibility. The optimality of the release velocity is encoded in a learned model of the throwing maps used in a bi-level optimization to compute the release states of both the object and robots. Moreover, while previous robotic interception works whether with a single arm or dual arm system usually consider a fully controllable interceptor (the robot), we consider an interceptor that is only partially controllable (the thrown object is only controllable up to the release). Furthermore, unlike (Asgari and Nikoobin, 2021), we go beyond the estimation of the throwable points, we derive a probabilistic model of their distribution and use it to predict landing (intercept) positions that yield a high probability of success.

CAPTURE-POINT BASED BALANCE AND REACTIVE OMNIDIRECTIONAL WALKING CONTROLLER

This Chapter presents our method to endow a biped humanoid robot with balance and reactive locomotion abilities necessary to support cooperative manipulation tasks. It briefly reviews works on reactive walking controllers and proceeds to describe our formulation that combines the capture-point (CP) and center of mass (CoM) dynamics within a model predictive control (MPC) framework to generate on-the-fly adjustable omnidirectional walking patterns that are consistent with the walking constraints. It validates our proposed approach by showcasing its application in human-robot cooperative tasks.

Publication note: The material presented in this Chapter was adopted from:

- Bombile, M. and Billard, A., 2017, November. Capture-point based balance and reactive omnidirectional walking controller. In 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids) (pp. 17-24). IEEE.

Source codes:

- Reactive bipedal walking controller
<https://github.com/epfl-lasa/biped-walking-controller>

Supplementary Video:

- Capture-point based reactive omnidirectional walking controller
<https://youtu.be/1rPaMXguPDQ>

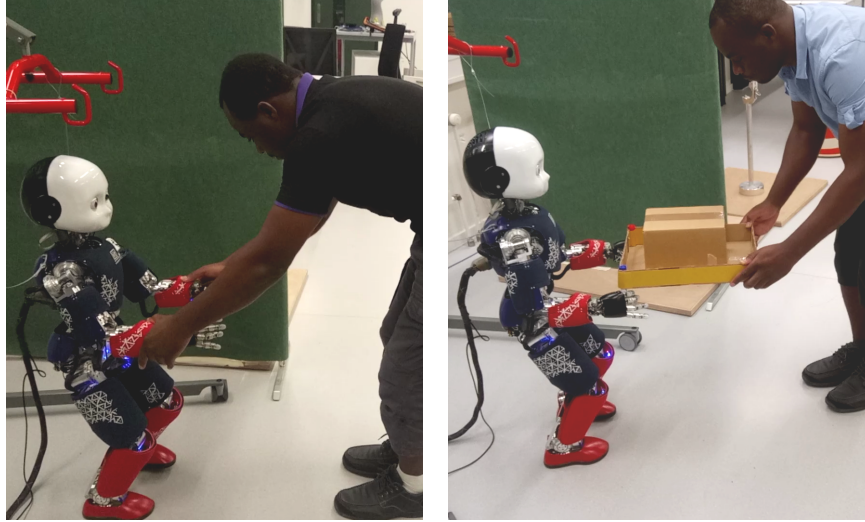


Figure 3.1: iCub walking: left- in reaction to persistent pulling force, and right - in reaction to human's intention in a cooperative transporting task

3.1 Introduction

THE potential ability of humanoid robots to operate in unstructured environments with narrow passages and limited support areas renders them very useful for service robotics. They could be employed for underground and planetary explorations, rescue operations after disasters, etc. Moreover, their anthropomorphic structure enables them to perform tasks in environments designed for humans and potentially to better collaborate with humans. However, working in such an environment is really challenging; the robots might be pulled or pushed while interacting with humans. They could possibly bump into obstacles, step on small objects lying on the floor or walk on different surfaces, etc. Consequently, they must be balanced and have a stable and robust walking ability to successfully complete their missions and cope with perturbations. The walking control problem, as outlined in Section 2.4.1, has been generally addressed through the generation of walking patterns: CoM, ZMP trajectories, and footstep positions that are often planned in advance.

Thus, this Chapter focuses on cases where the footsteps cannot be planned beforehand, but have to be automatically determined on-the-fly. Such a reactive behavior is particularly important when the humanoid robot is driven by a high-level task objective which provides reference commands in form of velocities to be followed by the walking robot (e.g. while cooperatively transporting an object (see Figure 3.1)) or tasks which could generate a significant change of the robot's momentum (e.g. when handing over or throwing a heavy object). Moreover, such a reactive behavior is suitable for the mitigation of the effects of other disturbances such as pushing or pulling forces, ground unevenness, etc., that the robot could be subjected to while performing its prescribed task.

Unlike reactive walking pattern generators (Herdt et al., 2010b,a; Bombile, 2015b; Naveau et al., 2017) discussed in Section 2.4.1, this Chapter proposes a reactive omnidirectional walking algorithm based on the control of the capture-point (CP). Our approach draws inspiration from these reactive walking algorithms, designed for the ZMP tracking, but extends them to the tracking of the CP to leverage its inherent robustness. Our walking controller has the ability to automatically generate reference trajectories of the CP and their related foot-step positions and orientations in response to desired locomotion velocities of the robot, and adapt the position of the CoP within the support polygon in reaction to perturbations. Moreover, in achieving reactive omnidirectionality, to tackle the nonlinearity challenges introduced by the orientation, our approach focuses on the discrete orientation of the footsteps instead of the continuous rotation of the CoM (hip). As a result, this yields a piece-wise linear problem solved with linear MPC. In addition to the theoretical development, the experimental validation of the proposed controller is provided.

This Chapter is organized as follows. Section 3.2 briefly introduces the model of the CoM-CP simplified dynamics and states the problem. Section 3.3 presents the proposed reactive omnidirectional balance and walking controller. In Section 3.4, experimental results validating the proposed controller are presented and briefly discussed. Finally, Section 3.5 summaries the Chapter and provides some future perspectives.

3.2 Modeling and Problem Formulation

3.2.1 CoM-CP SYSTEM DYNAMICS

Consider the illustration of Figure 3.2, which shows an inverted pendulum superimposed on the stance leg of a standing humanoid robot executing a step.

The balance of a humanoid robot, as shown in Eq.(2.2.7), can be modeled by the linear and angular momentum dynamics of the entire robot about the CoM (centroidal dynamics (Orin et al., 2013)). Assuming that the robot is standing on one foot and performing a step, the centroidal dynamics can be re-written as

$$\dot{\mathbf{h}} = \begin{cases} \dot{\mathcal{P}} = m\ddot{\mathbf{c}} & = \mathbf{f}_r + m\mathbf{g} \\ \dot{\mathcal{L}} = \mathbf{c} \times m\ddot{\mathbf{c}} & = \mathbf{c} \times m\mathbf{g} + \mathbf{r} \times \mathbf{f}_r + \boldsymbol{\tau}_r \end{cases} \quad (3.2.1)$$

$s.t \quad \mathbf{f}_r \subset \text{friction cone}$

where m is the total mass of the robot. $\mathbf{c} = [c_x \ c_y \ c_z]^T$ is the position of CoM and $\mathbf{r} = [r_x \ r_y \ r_z]^T$ denotes the position foot ankle (footstep position) with respect to the inertial frame \mathcal{W} . $\mathbf{f}_r = [f_r^x \ f_r^y \ f_r^z]^T$ and $\boldsymbol{\tau}_r = [\tau_r^x \ \tau_r^y \ \tau_r^z]$ denotes the reaction force and torque at the robot's ankle.

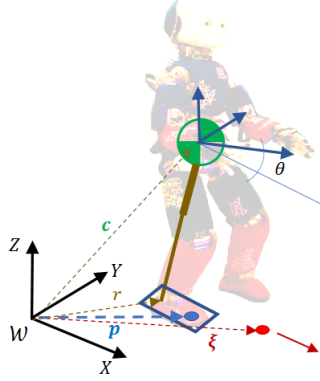


Figure 3.2: 3D Linear inverted pendulum model (LIPM) with finite-sized foot as simplified model of a standing and stepping humanoid robot. $\mathbf{c} \in \mathbb{R}^3$, $\mathbf{r} \in \mathbb{R}^3$, $\mathbf{p} \in \mathbb{R}^3$ and $\boldsymbol{\xi} \in \mathbb{R}^3$ denote the positions of the CoM, the footstep, the ZMP and the CP, respectively.

3D-LIPM WITH POINT MASS AND FINITE-SIZED FOOT

Combining the rows of Eq.(3.2.1) reduces the moment balance expression to

$$\mathbf{f}_r \times (\mathbf{c} - \mathbf{r}) + \boldsymbol{\tau}_r = 0 \quad (3.2.2)$$

Developing Eq.(3.2.2) gives the expression of the CoM dynamics. Additionally, if we assume that the CoM is constrained on a horizontal plane such that $c_z - r_z = c_{z0}$ and $\ddot{c}_z = 0$, we obtain the famous equations of the 3D linear inverted pendulum model (3D-LIPM (Kajita et al., 2001, 2003)) with finite-sized foot as

$$\ddot{c}_x = \omega^2(c_x - p_x) \quad \text{and} \quad \ddot{c}_y = \omega^2(c_y - p_y) \quad (3.2.3)$$

where $\omega \triangleq \sqrt{g/c_{z0}}$ denotes the natural frequency of the inverted pendulum. $p_x \triangleq \left(r_x - \frac{\tau_y^u}{m(\ddot{c}_z + g)} \right)$ and $p_y \triangleq \left(r_y + \frac{\tau_x^u}{m(\ddot{c}_z + g)} \right)$ represent the horizontal coordinates of the ZMP.

CAPTURE-POINT DYNAMICS

Mathematically, the CP can be derived from the orbital energy of the inverted pendulum. Alternatively, it can be determined from the solution of Eq. (3.2.3) which asymptotically tends toward the ZMP. Its expression is

$$\xi_x \triangleq c_x + \frac{1}{\omega} \dot{c}_x \quad \text{and} \quad \xi_y \triangleq c_y + \frac{1}{\omega} \dot{c}_y \quad (3.2.4)$$

The dynamics of the CP can be determined from (3.2.4) and (3.2.3) to give

$$\dot{\xi}_x = \omega \xi_x - \omega p_x \quad \text{and} \quad \dot{\xi}_y = \omega \xi_y - \omega p_y \quad (3.2.5)$$

Eq.(3.2.5) describes the evolution of the CP as function of the ZMP.

Finally, combining Eqs.(3.2.5) and (3.2.3) gives the CoM-CP joint dynamics, which can be expressed as (Engelsberger et al., 2011)

$$\begin{bmatrix} \dot{c}_h \\ \dot{\xi}_h \end{bmatrix} = \begin{bmatrix} -\omega & \omega \\ 0 & \omega \end{bmatrix} \begin{bmatrix} c_h \\ \xi_h \end{bmatrix} + \begin{bmatrix} 0 \\ -\omega \end{bmatrix} p_h \quad (3.2.6)$$

where the subscript $h \equiv x, y$ is associated with the x and y coordinates. This the dynamics governs the translational motions of the robot and it will be used to propagate the states over the MPC's prediction horizon.

3.2.2 PROBLEM FORMULATION

The main idea to generate walking patterns on the fly based on CP control, as in ZMP-based approaches (Herdt et al., 2010b,a; Naveau et al., 2017), consists to formulate an optimization problem where the walking patterns including the footstep positions and orientations are defined as decision variables. Then, let the algorithm compute their values that satisfy desired high-level walking task objectives (e.g. desired walking velocity) and constraints. Hence, the problem can be mathematically synthesized as

$$\begin{aligned} r_h^*, \theta_f^*, p_h^*, \ddot{\theta}^* = \underset{r_h, \theta_f, p_h, \ddot{\theta}}{\operatorname{argmin}} \quad & (J_1(\dot{c}_h, \dot{\theta}) + J_2(\xi_h, r_h, \theta_f) + J_3(p_h, \ddot{\theta})) \quad (3.2.7) \\ \text{s.t.} \quad & \text{balance \& walking constraints} \end{aligned}$$

where r_h and θ_f represent footstep position and orientation, respectively. subjected to balance and walking constraints. p_h is the ZMP and $\ddot{\theta}$ denotes the angular jerk of the CoM's frame (more details in Section 3.3.3). $J_1 - J_3$ are cost functions with the following purposes:

- $J_1(\dot{c}_h, \dot{\theta}) \triangleq J_{V_{CoM}}$ must ensure that the actual velocity of the frame attached to the CoM tracks its reference velocity (translation + rotation),
- $J_2(\xi_h, f_h, \theta_f) \triangleq J_{f_{Pose}}$ must ensure that the actual CP position and footstep orientation follow their auto-generated references,
- $J_3(p_h, \ddot{\theta}) \triangleq J_{ctrl}$ must ensure that the control effort to achieve the previous objectives is kept minimal,

The individual cost functions need to be defined to solve our walking patterns generation problem, which will be formulated within MPC framework to handle explicitly the balance and walking constraints. To that end, we will require: 1) the CoM-CP dynamics and a rotational dynamics of the robot that will govern translational and rotational displacements of the robot, 2) a model for the auto-generation of the CP reference trajectory, and 3) a formulation of walking constraints that have to be respected by the robot.

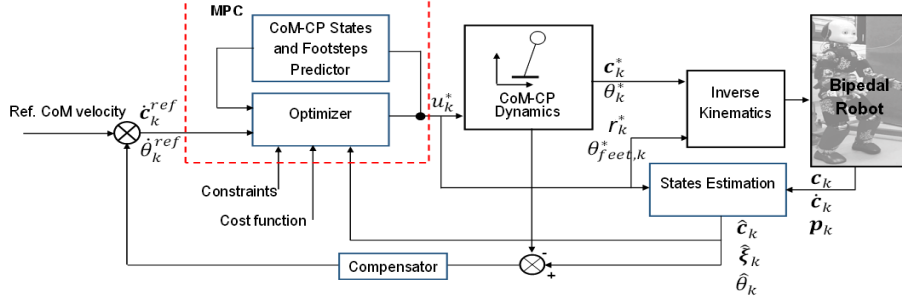


Figure 3.3: Proposed MPC based reactive walking controller. Given the estimated CoM and CP states $(\hat{c}_k, \hat{\xi}_k, \hat{\theta}_k)$ of the robot (under perturbation or not), the controller generates motions of the CoM (c_k^*, θ_k^*) and of the feet $(r_k^*, \theta_{feet,k}^*)$ to steer the CoM translational and rotational velocities towards their reference $(\dot{c}_k^{ref}, \dot{\theta}_k^{ref})$ (feedforward + feedback coming from the Compensator block). u_k^* denotes the optimized decision variables and p_k the ZMP.

3.3 Capture Point Based Reactive Omnidirectional Walking Controller

This section presents the development of the proposed walking controller able to generate reactively not only the footstep positions, but also their orientations. A block diagram of the proposed MPC-based solution is shown in Figure 3.3.

3.3.1 CoM-CP PREDICTION MODEL

After discretizing Eq. (3.2.6) with the state vector defined as $\mathbf{x}_k \triangleq \begin{bmatrix} c_k & \xi_k \end{bmatrix}^\top$ (index h dropped to simplify notation), the prediction model over a horizon N of each state of Eq. (3.2.6) can be written as

$$\begin{aligned} \underline{c}_k &= S_c \mathbf{x}_k + U_c \underline{p}_{k-1} \\ \underline{\xi}_k &= S_\xi \mathbf{x}_k + U_\xi \underline{p}_{k-1} \end{aligned} \quad (3.3.1)$$

where the notation $\underline{\cdot}_k$ means a stack of N *predicted* values of the considered variable, starting from but not including the time index k . Here $S_j \in \mathbb{R}^{N \times 2}$ and $U_j \in \mathbb{R}^{N \times N}$ ($j \equiv c, \xi$) represent respectively the sub-matrices associated with the state c or ξ of the matrices \mathbf{S}_s and \mathbf{U}_p given by

$$\mathbf{S} \triangleq \begin{bmatrix} \mathbf{A} \\ \vdots \\ \mathbf{A}^N \end{bmatrix}, \quad \mathbf{U} \triangleq \begin{bmatrix} \mathbf{A}^0 \mathbf{B} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{A}^{N-1} \mathbf{B} & \dots & \mathbf{A}^0 \mathbf{B} \end{bmatrix}$$

where the state transition matrix \mathbf{A} and the control vector \mathbf{B} of the discrete model of (3.2.6) are given by

$$\mathbf{A} = \begin{bmatrix} e^{-\omega T} & \frac{1}{2} e^{\omega T} (1 - e^{-2\omega T}) \\ 0 & e^{\omega T} \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 1 - \frac{1}{2} e^{\omega T} (1 + e^{-2\omega T}) \\ 1 - e^{\omega T} \end{bmatrix} \quad (3.3.2)$$

with T is the sampling time.

3.3.2 CoM AVERAGE VELOCITY

The average CoM velocity of a walking biped robot would be conveniently computed between every two footsteps because of the sway motion. If, for instance, four footsteps are considered within the MPC horizon N , the average velocity can be written as

$$\dot{\underline{\mathbf{c}}}_k = E \underline{\mathbf{c}}_k \quad \text{with} \quad E = \frac{1}{2T_{sp}} \begin{bmatrix} -I & I \\ -I & I \end{bmatrix} \quad (3.3.3)$$

Here $I \in \mathbb{R}^{\frac{N}{2} \times \frac{N}{2}}$ is a unit matrix and T_{sp} denotes the duration of a step.

3.3.3 ROBOT'S ORIENTATION

ANGULAR TRAJECTORY OF THE ROBOT

In order to follow a rotational velocity, we attach a frame to the CoM and consider only the yaw angle θ (assuming an upright posture). Using the jerk $\ddot{\theta}_k$ as a control variable, the rotational motion of the CoM frame can be described by (Herdt et al., 2010a)

$$\begin{bmatrix} \theta_{k+1} \\ \dot{\theta}_{k+1} \\ \ddot{\theta}_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}_\theta} \underbrace{\begin{bmatrix} \theta_k \\ \dot{\theta}_k \\ \ddot{\theta}_k \end{bmatrix}}_{\boldsymbol{\theta}_k} + \underbrace{\begin{bmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{bmatrix}}_{\mathbf{B}_\theta} \ddot{\theta}_k \quad (3.3.4)$$

Similarly to (3.3.1), the prediction model related to the first two states of (3.3.4) over the horizon N can be written as

$$\begin{aligned} \underline{\boldsymbol{\theta}}_k &= S_\theta \boldsymbol{\theta}_k + U_\theta \ddot{\underline{\boldsymbol{\theta}}}_{k-1} \\ \dot{\underline{\boldsymbol{\theta}}}_k &= S_{\dot{\theta}} \boldsymbol{\theta}_k + U_{\dot{\theta}} \ddot{\underline{\boldsymbol{\theta}}}_{k-1} \end{aligned} \quad (3.3.5)$$

Computed as in (3.3.1) but with $(\mathbf{A}_\theta, \mathbf{B}_\theta)$ instead of (\mathbf{A}, \mathbf{B}) , $S_j \in \mathbb{R}^{N \times 3}$ and $U_j \in \mathbb{R}^{N \times N}$ ($j \equiv \theta, \dot{\theta}, \ddot{\theta}$) represent the sub-matrices associated with the state θ , $\dot{\theta}$ and $\ddot{\theta}$ over the horizon N . Thus, from (3.3.3) and (3.3.5) $J_{V_{CoM}}$ can now be written as

$$J_{V_{CoM}} = \sum_{h=x,y} \frac{\beta}{2} \left\| \dot{\underline{\mathbf{c}}}_k - \dot{\underline{\mathbf{c}}}_k^{ref} \right\|^2 + \frac{\alpha_\theta}{2} \left\| \dot{\underline{\boldsymbol{\theta}}}_k - \dot{\underline{\boldsymbol{\theta}}}_k^{ref} \right\|^2 \quad (3.3.6)$$

where $\dot{\underline{\mathbf{c}}}_k^{ref} \in \mathbb{R}^{N \times 1}$ and $\dot{\underline{\boldsymbol{\theta}}}_k^{ref} \in \mathbb{R}^{N \times 1}$ are the translational and rotational reference velocity of the CoM frame. β and α_θ are weights of the cost function.

3.3.4 SELF-GENERATED CAPTURE-POINT REFERENCE TRAJECTORIES

The reference trajectory of the CP, $\xi^*(t)$, is derived from the solution of the CP dynamics (3.2.6). Hence, for constant ZMP position, $\xi^*(t)$ can be written as

$$\xi^*(t) = e^{\omega t} \xi_{ini} + (1 - e^{\omega t}) r \quad (3.3.7)$$

where ξ_{ini} and r are respectively the initial CP and the footstep position (fixed ZMP). Hence, for a sequence of m footsteps, if the CP at the end of the step i is denoted by $\xi_{eos,i}$, the dynamically consistent initial CP, $\xi_{ini,i}$, can be computed with a backward recursion as (Englsberger and Ott, 2012a; Krause et al., 2012)

$$\xi_{ini,i} = \xi_{eos,i} e^{-\omega T_{sp}} + (1 - e^{-\omega T_{sp}}) r_i \quad (3.3.8)$$

$$\xi_{eos,i-1} = \xi_{ini,i} \quad (3.3.9)$$

Thus, starting from the final preplanned footstep (r_m and $\xi_{eos,m}$), the $\xi_{ini,i}$ are computed down to the current footstep.

Unlike in (Englsberger and Ott, 2012a; Krause et al., 2012; Griffin and Leonessa, 2016; Shafiee-Ashtiani et al., 2017) where the reference footsteps were predetermined, in this work, they are formulated as variables such that they can be generated automatically on-line. To that end, let us consider that there are only four ($m = 4$) footsteps (r_1, r_2, r_3, r_4) falling within the receding horizon N (it only eases the computation of the CoM's average velocity in the rest of these developments). Thus, starting from the fourth footstep and using (3.3.7), it can be shown that all $\xi_{ini,i}$ will be given by

$$\xi_{ini,1:4} = N_e \xi_{eos,4} + \mathbf{M}_e \mathbf{r}_{1:4} \quad (3.3.10)$$

where $\xi_{eos,4} = \xi_{eos,m} = \xi_N$ is the end of horizon CP and

$$\begin{aligned} \xi_{ini,1:4} &\triangleq [\xi_{ini,1} \quad \xi_{ini,2} \quad \xi_{ini,3} \quad \xi_{ini,4}]^\top \\ \mathbf{r}_{1:4} &\triangleq [r_1 \quad r_2 \quad r_3 \quad r_4]^\top \\ N_e &\triangleq \begin{bmatrix} e^{-4\omega T_{sp}} & e^{-3\omega T_{sp}} & e^{-2\omega T_{sp}} & e^{-\omega T_{sp}} \end{bmatrix}^\top \\ \mathbf{M}_e &\triangleq \delta_{e_\omega} \begin{bmatrix} 1 & e^{-\omega T_{sp}} & e^{-2\omega T_{sp}} & e^{-3\omega T_{sp}} \\ 0 & 1 & e^{-\omega T_{sp}} & e^{-2\omega T_{sp}} \\ 0 & 0 & 1 & e^{-\omega T_{sp}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

with $\delta_{e_\omega} \triangleq (1 - e^{-\omega T_{sp}})$. Substituting now (3.3.10) in (3.3.7), the four CP reference trajectories can be shown to be given by

$$\xi^*(t) = \underbrace{\begin{bmatrix} e^{\omega t} \mathbf{M}_e + (1 - e^{\omega t}) \mathbf{I}_m \end{bmatrix}}_{\Xi_{r_m}} \underbrace{\begin{bmatrix} e^{\omega t} N_e \end{bmatrix}}_{\Xi_{\xi_N}} \begin{bmatrix} \mathbf{r}_{1:m} \\ \xi_N \end{bmatrix} \quad (3.3.11)$$

Separating the known current footstep position r_1 of $\mathbf{r}_{1:m}$ from the unknown future footsteps $\mathbf{r}_{2:m}$ (to be determined together with ξ_N), the reference CP for the m footsteps can now be written as

$$\xi^*(t) = \Xi_{\mathbf{r}_m} V_c r_1 + \Xi_{\mathbf{r}_m} \mathbf{V}_f \mathbf{r}_{2:m} + \Xi_{\xi_N} \xi_N \quad (3.3.12)$$

where the substitution $\mathbf{r}_{1:m} = V_c r_1 + \mathbf{V}_f \mathbf{r}_{2:m}$ was used and where $V_c \in \mathbb{R}^{m \times 1}$ and $\mathbf{V}_f \in \mathbb{R}^{m \times (m-1)}$ are constant selection vector and matrix given by

$$V_c \triangleq \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \mathbf{V}_f \triangleq \begin{bmatrix} 0 & \cdots & 0 \\ 1 & \ddots & \vdots \\ 0 & \ddots & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3.13)$$

Note that (3.3.12) defines a piecewise continuous trajectory whose discontinuity stems from the discrete footsteps. Between two successive footsteps, this trajectory is continuous and can be obtained for each footstep (each row of (3.3.12)) by varying the time t between $[0, T_{sp}]$ (reinitialized due to change of initial condition for each footsteps). Over the prediction horizon N , the overall CP reference trajectory $\underline{\xi}_k^* \in \mathbb{R}^{N \times 1}$ will be obtained by superimposing the discretized CP reference trajectories of each of the m footsteps.

3.3.5 FOOTSTEP POSITIONS AND ORIENTATIONS

FOOTSTEP ORIENTATIONS

During bipedal locomotion, the footstep positions are discrete over time and so are their orientations. If it is assumed that there is no slippage between the foot and the ground, then the current footstep orientation denoted $\theta_{f,1}^w$ is fixed and known with respect to the inertial frame \mathcal{W} . In such a case, only the future footstep orientations denoted $\theta_{f,2:m}^w$ have to be determined. Hence, the footstep orientations over the prediction horizon can be written as

$$\underline{\theta}_k^{*w} = H_{k+1}^c \theta_{f,1}^w + \mathbf{H}_{k+1}^f \theta_{f,2:m}^w \quad (3.3.14)$$

where $H_{k+1}^c \in \mathbb{R}^{N \times 1}$ and $\mathbf{H}_{k+1}^f \in \mathbb{R}^{N \times (m-1)}$ are cyclic vector and matrix associating each sample instant with a footstep. They are the same as those used for translations in (Herdt et al., 2010b,a). The superscript w indicates a reference to the frame \mathcal{W} . Thus, in order to ensure the desired orientation of the robot, the decision variables to be optimized are the future footstep orientations $\theta_{f,2:m}^w$ in (3.3.14) and the jerk $\ddot{\underline{\theta}}_{h,k-1}$ in (3.3.5).

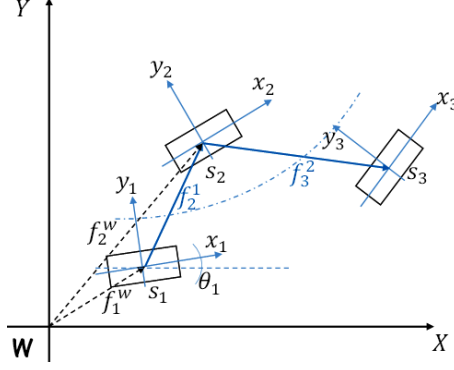


Figure 3.4: Example of footstep positions when the robot performs a rotation. $r_1^w = [x_1^w \ y_1^w]^\top$ and $r_2^w = [x_2^w \ y_2^w]^\top$ denote respectively the positions of the footsteps s_1 and s_2 with respect to the fixed inertial frame \mathcal{F}_w (the world frame), while $r_j^i = [x_j^i \ y_j^i]^\top$ represents the position of the footstep s_j relative to the footstep s_i .

FOOTSTEP POSITIONS

When accounting for the rotational motion of the robot, the X and Y translations which were previously independent will now be coupled by the nonlinear orientation mapping. To illustrate that, consider Figure 3.4 depicting a sequence of three footsteps. If r_1^w has an orientation θ_1 with respect to the inertial frame \mathcal{W} , the position of r_2^w with respect to \mathcal{W} is given by

$$\begin{aligned} x_2^w &= x_1^w + \cos\theta_1 x_2^1 - \sin\theta_1 y_2^1 \\ y_2^w &= y_1^w + \sin\theta_1 x_2^1 + \cos\theta_1 y_2^1 \end{aligned} \quad (3.3.15)$$

Equation (3.3.15) shows clearly the introduced nonlinearity. However, keeping the MPC linear or reducing the induced non-linearity will be beneficial for a real-time implementation of the controller. To that end, let us rewrite (3.3.15) as follows

$$r_2^w = r_1^w + R_{s_1}^w r_2^1 \quad (3.3.16)$$

The absolute footstep position (with respect to fixed inertial frame) r_2^w is written as affine transformation of the relative footstep r_2^1 . Thus, considering four footsteps ($m = 4$) over the horizon N , starting from the i^{th} footstep denoted s_i and following (3.3.16), the positions of the four footsteps $r_{i:i+3}^w$ and the end of horizon CP (end of the $i^{th} + 3$ step), ξ_N^w , can be written as

$$r_{i:i+3}^w = \mathbf{1}_4 r_{s_i}^w + \mathbf{R}_{xy} \Delta r_{i+1:3} \quad (3.3.17)$$

$$\xi_N^w = r_{s_i}^w + \mathbf{R}_{xy(3:)} \Delta r_{i+1:3} + R_{s_{i+2}}^w \xi_N^{s_{i+2}} \quad (3.3.18)$$

with

$$\Delta r_{i+1:3} \triangleq \begin{bmatrix} r_{s_{i+1}}^{s_i} \\ r_{s_{i+2}}^{s_i} \\ r_{s_{i+3}}^{s_i} \end{bmatrix}, \quad R_{s_i}^w \triangleq \begin{bmatrix} \cos\theta_{s_i} & -\sin\theta_{s_i} \\ \sin\theta_{s_i} & \cos\theta_{s_i} \end{bmatrix} \quad (3.3.19)$$

$$\mathbf{r}_{i:i+3}^w \triangleq \begin{bmatrix} \mathbf{r}_{s_i}^w \\ \mathbf{r}_{s_{i+1}}^w \\ \mathbf{r}_{s_{i+2}}^w \\ \mathbf{r}_{s_{i+3}}^w \end{bmatrix}, \mathbf{R}_{xy} \triangleq \begin{bmatrix} 0 & 0 & 0 \\ R_{s_i}^w & 0 & 0 \\ R_{s_i}^w & R_{s_{i+1}}^w & 0 \\ R_{s_i}^w & R_{s_{i+1}}^w & R_{s_{i+2}}^w \end{bmatrix}, \quad (3.3.20)$$

where $\mathbf{1}_4 \in \mathbb{R}^{4 \times 1}$ is a unit vector and in (3.3.18) $\mathbf{R}_{xy(3:)} \in \mathbb{R}^{2 \times 6}$ is the fourth row of \mathbf{R}_{xy} . $\mathbf{r}_{s_{i+1}}^{s_i} \triangleq [\mathbf{r}_{x,s_{i+1}}^{s_i} \quad \mathbf{r}_{y,s_{i+1}}^{s_i}]^\top$ and $\boldsymbol{\xi}_N^{s_i} \triangleq [\boldsymbol{\xi}_{x,N}^{s_i} \quad \boldsymbol{\xi}_{y,N}^{s_i}]^\top$ are respectively the relative positions of the step s_{i+1} and the end of horizon CP with respect to the step s_i .

Using equations (3.3.17) and (3.3.18) in conjunction with the definitions (3.3.19)-(3.3.20), and separating the x from the y components of the future steps $(\mathbf{r}_{s_{i+1}}^w, \mathbf{r}_{s_{i+2}}^w, \mathbf{r}_{s_{i+3}}^w)$, it can be shown that the considered four footstep positions expressed in a fixed inertial frame can be written as

$$\begin{cases} \mathbf{r}_{x,i:i+3}^w &= (V_c + \mathbf{V}_f \mathbf{1}_3) \mathbf{r}_{x,i}^w + V_f \mathbf{R}_x \Delta \mathbf{r}_{i+1:3} \\ \mathbf{r}_{y,i:i+3}^w &= (V_c + \mathbf{V}_f \mathbf{1}_3) \mathbf{r}_{y,i}^w + V_f \mathbf{R}_y \Delta \mathbf{r}_{i+1:3} \end{cases} \quad (3.3.21)$$

The matrices \mathbf{R}_x and $\mathbf{R}_y \in \mathbb{R}^{3 \times 6}$ are defined as

$$\mathbf{R}_x \triangleq \begin{bmatrix} c\theta_{s_i} & -s\theta_{s_i} & 0 & 0 & 0 & 0 \\ c\theta_{s_i} & -s\theta_{s_i} & c\theta_{s_{i+1}} & -s\theta_{s_{i+1}} & 0 & 0 \\ c\theta_{s_i} & -s\theta_{s_i} & c\theta_{s_{i+1}} & -s\theta_{s_{i+1}} & c\theta_{s_{i+2}} & -s\theta_{s_{i+2}} \end{bmatrix},$$

$$\mathbf{R}_y \triangleq \begin{bmatrix} s\theta_{s_i} & c\theta_{s_i} & 0 & 0 & 0 & 0 \\ s\theta_{s_i} & c\theta_{s_i} & s\theta_{s_{i+1}} & c\theta_{s_{i+1}} & 0 & 0 \\ s\theta_{s_i} & c\theta_{s_i} & s\theta_{s_{i+1}} & c\theta_{s_{i+1}} & s\theta_{s_{i+2}} & c\theta_{s_{i+2}} \end{bmatrix},$$

where $s\theta_{s_i}$ and $c\theta_{s_i}$ stand for $\sin\theta_{s_i}$ and $\cos\theta_{s_i}$, respectively.

Now, in order to generate automatically the reference trajectory of the CP, the positions (3.3.21) can in turn be substituted in (3.3.12). Thus, the decision variables to be determined in this case are now the relative footstep positions given by

$$\Delta \mathbf{r}_{i+1:3} = \begin{bmatrix} \mathbf{r}_{x,s_{i+1}}^{s_i} & \mathbf{r}_{y,s_{i+1}}^{s_i} & \mathbf{r}_{x,s_{i+2}}^{s_{i+1}} & \mathbf{r}_{y,s_{i+2}}^{s_{i+1}} & \mathbf{r}_{x,s_{i+3}}^{s_{i+2}} & \mathbf{r}_{y,s_{i+3}}^{s_{i+2}} \end{bmatrix}^\top$$

instead of their absolute values $\mathbf{r}_{i:i+3}^w$ as in (Herdt et al., 2010b)(Herdt et al., 2010a) or (Naveau et al., 2017). Similar reasoning applies also for $\boldsymbol{\xi}_N$.

Thus, J_{fPose} and J_{ctrl} can now be written as

$$J_{fPose} = \sum_{h=x,y} \frac{\gamma}{2} \left\| \underline{\boldsymbol{\xi}}_{\rightarrow h}^k - \underline{\boldsymbol{\xi}}_{\rightarrow h}^{*k} \right\|^2 + \frac{\gamma_\theta}{2} \left\| \underline{\boldsymbol{\theta}}_{\rightarrow}^k - \underline{\boldsymbol{\theta}}_{\rightarrow}^{*w} \right\|^2 \quad (3.3.22)$$

$$J_{ctrl} = \sum_{h=x,y} \frac{\kappa}{2} \left\| \underline{\Delta \mathbf{p}}_{\rightarrow h}^{k-1} \right\|^2 + \frac{\kappa_\theta}{2} \left\| \underline{\ddot{\boldsymbol{\theta}}}_{\rightarrow}^{k-1} \right\|^2 \quad (3.3.23)$$

where $\underline{\Delta \mathbf{p}}_{\rightarrow h}^{k-1}$ is the variation of $\underline{\mathbf{p}}_{\rightarrow h}^{k-1}$. γ , γ_θ , κ and κ_θ are weights of the cost function.

3.3.6 GLOBAL OBJECTIVE FUNCTION

From (3.3.6), (3.3.22) and (3.3.23), the global objective function J to be minimized so as to generate the CoM trajectory and the footstep positions and orientations can now be written as

$$\left\{ \begin{aligned} J &\triangleq \sum_{h=x,y} \frac{\beta}{2} \left\| \dot{\vec{c}}_k - \dot{\vec{c}}_k^{ref} \right\|^2 + \frac{\kappa}{2} \left\| \Delta \vec{p}_{h,k-1} \right\|^2 \\ &+ \frac{\gamma}{2} \left\| \vec{\xi}_{h,k} - \vec{\xi}_{h,k}^* \right\|^2 \\ &+ \frac{\alpha_\theta}{2} \left\| \vec{\theta}_k - \vec{\theta}_k^{ref} \right\|^2 + \frac{\kappa_\theta}{2} \left\| \ddot{\vec{\theta}}_{k-1} \right\|^2 \\ &+ \frac{\gamma_\theta}{2} \left\| \vec{\theta}_k - \vec{\theta}_k^{*w} \right\|^2 \end{aligned} \right. \quad (3.3.24)$$

Writing this objective as a QP problem leads to

$$\mathbf{u}_k^* = \underset{\mathbf{u}_k}{\operatorname{argmin}} \left(\frac{1}{2} \cdot \mathbf{u}_k^\top \mathbf{Q}_k \mathbf{u}_k + \mathbf{p}_k^\top \mathbf{u}_k \right) \quad (3.3.25)$$

$$\text{with } \mathbf{u}_k \triangleq \left[\vec{p}_{x,k-1} \ \vec{p}_{y,k-1} \ \Delta \mathbf{r}_{i+1:3} \ \xi_N^{s_i+2} \ \ddot{\vec{\theta}}_{k-1} \ \vec{\theta}_{f,i+1:3}^w \right]^\top$$

$$s.t \quad |\Delta \mathbf{r}_{x,i}| \leq l_x \quad (3.3.26)$$

$$l_{yi} \leq |\Delta \mathbf{r}_{y,i}| \leq l_{yo} \quad (3.3.27)$$

$$|\theta_{s_{i+1}} - \theta_{s_i}| \leq l_\theta, \quad \text{with } i = 1 \dots 3 \quad (3.3.28)$$

$$\mathbf{N}_k(\vec{\theta}_k^{*w}) \begin{bmatrix} \vec{p}_{x,k-1} - \vec{r}_{x,k-1}^w \\ \vec{p}_{y,k-1} - \vec{r}_{y,k-1}^w \end{bmatrix} \leq \mathbf{b}_k \quad (3.3.29)$$

with $\vec{r}_{h,k-1}^w = (H_{k+1}^c + \mathbf{H}_{k+1}^f \mathbf{1}_3) \mathbf{r}_{h,i,k}^w + \mathbf{H}_{k+1}^f \mathbf{R}_h \Delta \mathbf{r}_{i+1:3,k}$ and where l_x and l_θ represent respectively the upper bounds of the relative footstep longitudinal and angular displacements, while l_{yi} and l_{yo} represent respectively the lower and upper bounds of the relative footstep position in the lateral direction. $\mathbf{N}_k(\vec{\theta}_k^{*w})$ and \mathbf{b}_k represent respectively the matrix gathering the x and y components of the normals to the edges of the support polygon and the bounds of the latter in the direction of the normals. Also, in (3.3.25) we have

$$\mathbf{Q}_k = \begin{bmatrix} Q_k^{p_x} & 0 & Q_k^{p_x \Delta \mathbf{r}} & Q_k^{p_x \xi_N} & 0 \\ 0 & Q_k^{p_y} & Q_k^{p_y \Delta \mathbf{r}} & Q_k^{p_y \xi_N} & 0 \\ Q_k^{\Delta \mathbf{r}, p_x} & Q_k^{\Delta \mathbf{r}, p_y} & Q_k^{\Delta \mathbf{r}} & Q_k^{\Delta \mathbf{r} \xi_N} & 0 \\ Q_k^{\xi_N p_x} & Q_k^{\xi_N p_y} & Q_k^{\xi_N \Delta \mathbf{r}} & Q_k^{\xi_N} & 0 \\ 0 & 0 & 0 & 0 & Q_k^\theta \end{bmatrix} \quad \text{and } \mathbf{p}_k = \begin{bmatrix} p_k^{p_x} \\ p_k^{p_y} \\ p_k^{\Delta \mathbf{r}} \\ p_k^{\xi_N} \\ p_k^{\ddot{\theta}} \\ p_k^{\theta_r} \end{bmatrix} \quad (3.3.30)$$

with the $Q_{h,ij}$ and $p_{k,i}$ elements given in Appendix B.1.

Note that the use in the proposed MPC formulation of relative instead of absolute footstep positions as decision variables has the advantage to transform a global problem into a local one. It allows us, unlike in (Herdt et al., 2010b) (Naveau et al., 2017), to keep the MPC's footstep feasibility constraints linear ((3.3.26) and (3.3.27)) except the constraints on the CoP (3.3.29). However,

based on (3.3.30) and the consideration that there is no obstacle in the biped's workspace, the angular variable (θ) of the foot is free to reach, independently from the x and y variables, its prescribed value dictated by the desired rotational velocity (provided that the latter is within the robot capabilities). Thus, the orientation can be solved in a separated MPC and the obtained values at each iteration substituted as parameters for the x and y variables. As a result, this MPC problem which is non-linear in nature due to the rotation can be solved sequentially as a linear problem (Bombile, 2015a,b).

Examples: To illustrate the behavior of the proposed controller, we simulate the generation of omnidirectional walking patterns in response to desired translational and rotational velocities of the frame attached to the robot's CoM.

We first consider a pure translation along the longitudinal direction (X) as illustrated in Figure 3.5, which shows the associated walking patterns. The values of the desired (blue line) and current (green line) velocities of the robot model (3D-LIPM) are shown in Figure 3.5(a). The positions of the CP and the automatically generated CP and ZMP references are shown in Figure 3.5(b). The ZMP references correspond to the footstep positions. Figure 3.5(c) shows additionally the trajectories of the CoM and the current ZMP acting as the control variable of the CoM-CP dynamics (see Eq.(3.2.6)). The ZMP clearly remains within its limits as can be seen in Figure 3.5(d), which shows the 2D walking trajectories and the footstep position along with their support polygons.

Similarly, we simulate a pure lateral translation (Y direction) as illustrated in Figure 3.6. The desired and current velocities of the 3D-LIPM are shown in Figure 3.6(a). The velocity of the 3D-LIPM tracks on average its desired value, although alternating between negative and positive values because of the sway motion. Figure 3.6(b) shows how closely the position of CP follows its generated reference (CP_{ref}). All walking patterns including the CoM and the ZMP trajectories are shown in Figure 3.6(c). The 2D footstep positions during this pure lateral translation are shown in Figure 3.6(d).

Finally, we simulate a pure planar rotation of the robot by specifying desired angular velocities around the vertical axis (Z). The successive variations of the desired angular velocity along with the angular velocity of the robot model are shown in Figure 3.7(a). The discrete footstep orientations along with the continuous angular positions of the frame attached to the CoM are shown in Figure 3.7(b). Figure 3.7(c) and Figure 3.7(d) show the walking patterns in the X and Y directions, respectively. It can be seen that each forward step of one foot is followed by a backward step of the other foot, both with the desired orientation. As a result, the robot performs in discrete steps a rotation without translation, as can be seen in Figure 3.8, which shows a rotation of about $\pi(rad)$ with the footstep positions similar to their initial values.

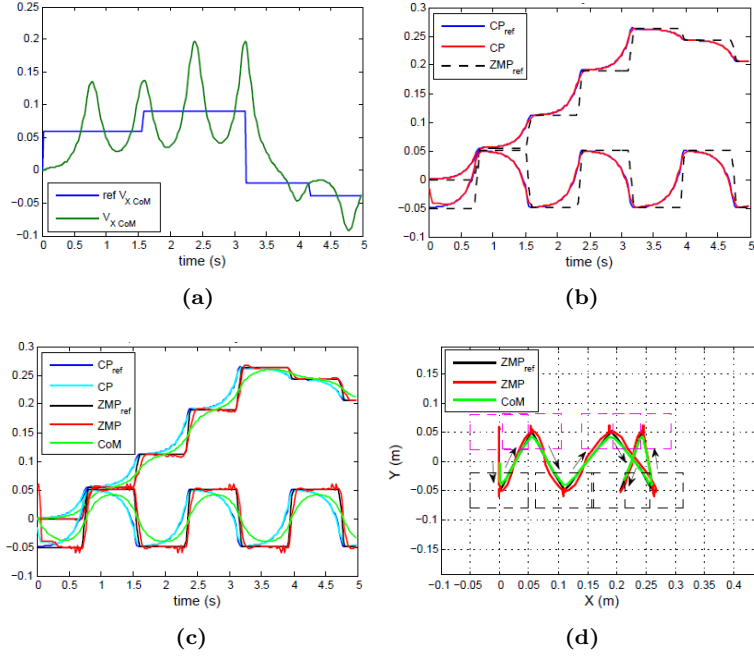


Figure 3.5: Example of generated walking trajectories for a biped robot in pure translation along the **longitudinal** direction. (a) The reference and current velocity of the CoM. (b) Trajectories of the CP and the footstep positions (ZMP_{ref}). (c) Trajectories of the CP, the ZMP and CoM of the robot model. (d) 2D trajectories of the footsteps, ZMP and CoM.

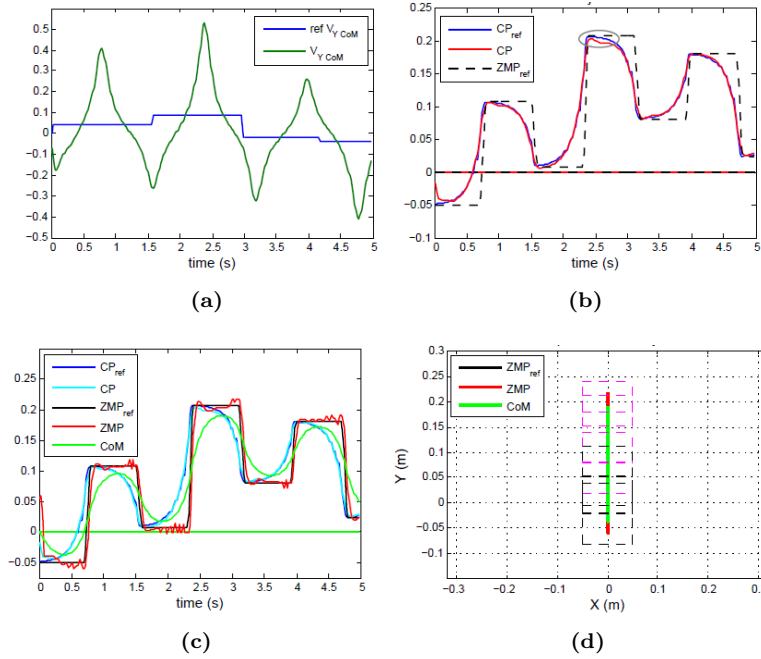


Figure 3.6: Example of generated walking trajectories for a biped robot in pure translation along the **lateral** direction. (a) The reference and current velocity of the CoM. (b) Trajectories of the CP and the footstep positions (ZMP_{ref}). (c) Trajectories of the CP, the ZMP and CoM of the robot model. (d) 2D trajectories of the footsteps, ZMP and CoM.

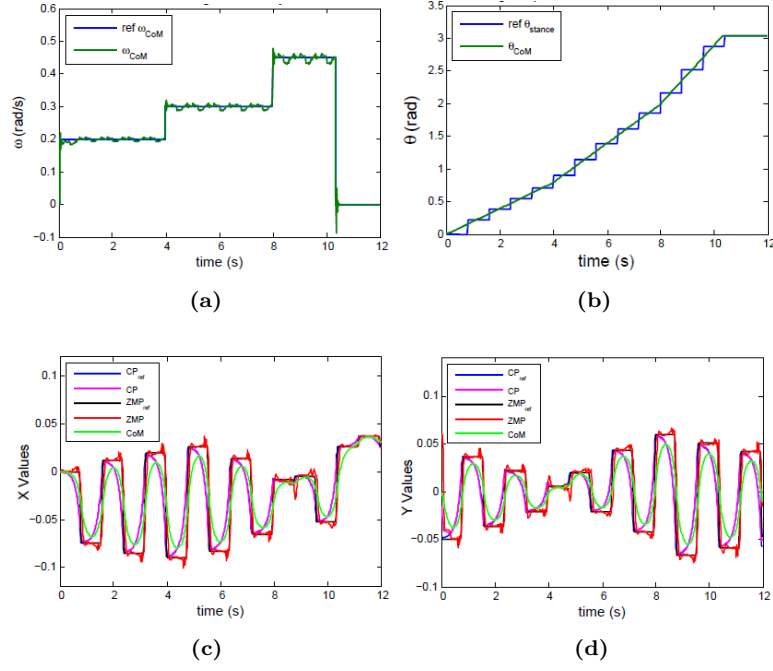


Figure 3.7: Example of generated walking trajectories for a biped robot in pure rotation around the **vertical** axis. (a) The reference and current angular velocity of the frame attached to the CoM. (b) Discrete footstep orientations and continuous orientation of CoM's frame. (c)-(d) Trajectories of the CP, ZMP, CoM and footsteps along the X and Y directions.

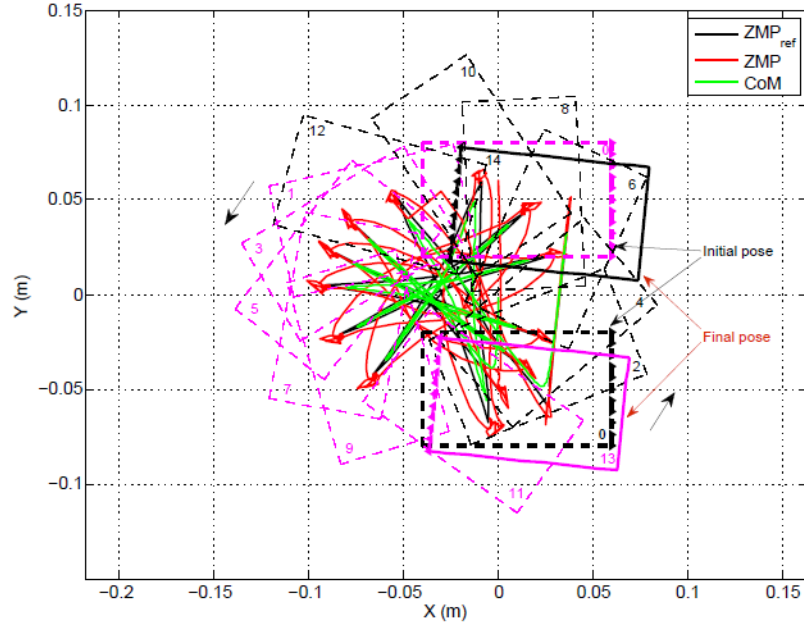


Figure 3.8: Example of generated 2D trajectories of the footsteps, ZMP and CoM of a robot model during a pure rotation. The magenta and black dashed rectangles indicate the support polygons of the left and right foot, respectively. Their initial and final positions after a rotation of about π (rad) are indicated the numbers 0, and 13 and 14, respectively.

3.4 Empirical Validation

This section presents the results of two kinds of reactive walking experiments carried out on the humanoid iCub (version 2.5). The first experiment is about velocity driven reactive walking task whereas the second is about interaction force based walking tasks. A video (Bombile and Billard, 2017a) of all experiments is provided as supplementary material.

From input velocity of the robot, the controller generates in real-time the reference trajectories of the CoM, the pose of the next footstep, which are sent to the inverse kinematics module. However, because the controller continuously adapts the pose of the next footstep depending on the input velocities or disturbances, quintic polynomial interpolation using the current and the predicted footstep pose at each iteration was used to ensure smooth 3D trajectories of the swing foot.

The MPC itself was solved with *qpOASES* (Ferreau et al., 2014) in an average time of 1 ms on an *Intel^(R) Core i7*, 3.4 GHz and 7.8 GB RAM PC. However, because the inverse kinematics solver took $6 - 11\text{ ms}$ for each foot, the sampling time was set at $T = 0.040\text{ s}$. The other MPC's parameters were set as follows: step duration $T_{sp} = 0.640\text{ s}$, gains $\beta = 0.20$, $\gamma = 1.50$, $\kappa = 0.80$, $\alpha_\theta = 1.0^{-6}$, $\beta_\theta = 1.00$, $\gamma_\theta = 1.00$, $\kappa_\theta = 0.10$.

3.4.1 VELOCITY DRIVEN OMNIDIRECTIONAL WALKING

This experiment validates the ability of the proposed locomotion controller to generate on-line stable and reactive omnidirectional walking trajectories and to stabilize the robot around them. Thus, the robot performs a combination of translations (longitudinal and lateral) and rotations. The desired velocity of the CoM or rather of a frame attached to the CoM is defined relative to the robot by the vector $[v_x\ v_y\ \omega_z]^T$, representing respectively the robot's longitudinal and lateral motion in $[m/s]$ and the rotation motion in $[rad/s]$.

The sequence of desired motion performed by the robot during this experiment is summarized in Table 3.1.

Table 3.1: Desired velocities during the velocity-driven walking experiment

$v_x\ [m/s]$	$v_y\ [m/s]$	$\omega_z\ [rad/s]$	$time\ [s]$
+0.06	0.03	0.00	6
0.00	0.00	-0.10	8
+0.05	0.00	-0.05	6

At the beginning, the robot translates in both longitudinal and lateral directions, then performs a pure rotation and finally combines a translation and a rotation. As can be seen in Figure 3.9 depicting the trajectories of the CP and the CoM, the robot followed its prescribed velocities and reference trajectories while staying stable.

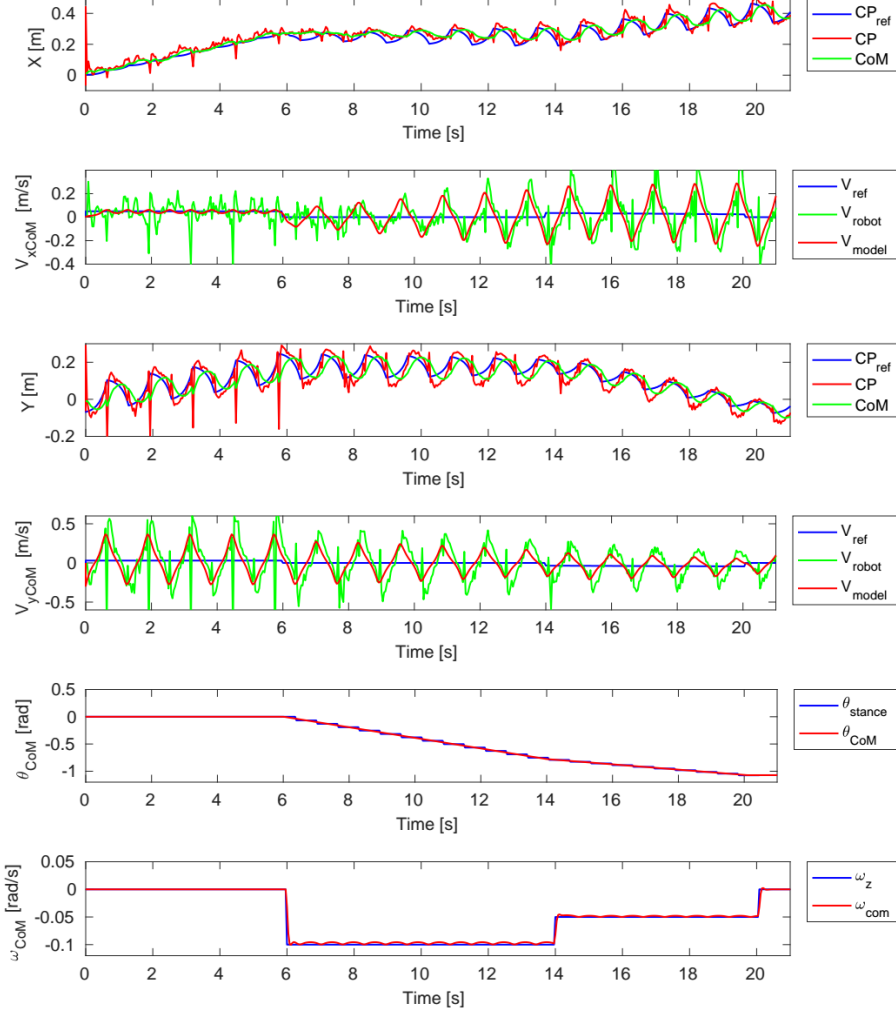


Figure 3.9: Time evolution of the CP and CoM positions and velocities during the tracking task. The value of θ_{CoM} is actually the value of the orientation of the robot base (root) frame. The first two figures at the top and the two figures in the middle represent the translational positions and velocities in the X and Y direction, respectively. The last two figures represent the angular positions and velocities of the base of the robot.

The ZMP, here the control variable, varies along its reference values generated by the algorithm. However, it stays within the support polygon as can be seen in Figure 3.10, which depicts the footstep poses, the ZMP (reference and actual) and their associated support polygons. If, for instance, the ZMP was kept constant at the 6th second when the CoM state changed abruptly, the CP would have evolved from that new state according to (3.3.7). Consequently, it would have required a much bigger step to maintain the balance of the robot. However, because the ZMP is free to move within the support polygon, the proposed controller reacted by computing a minimal action that steered the robot state back towards its desired value.

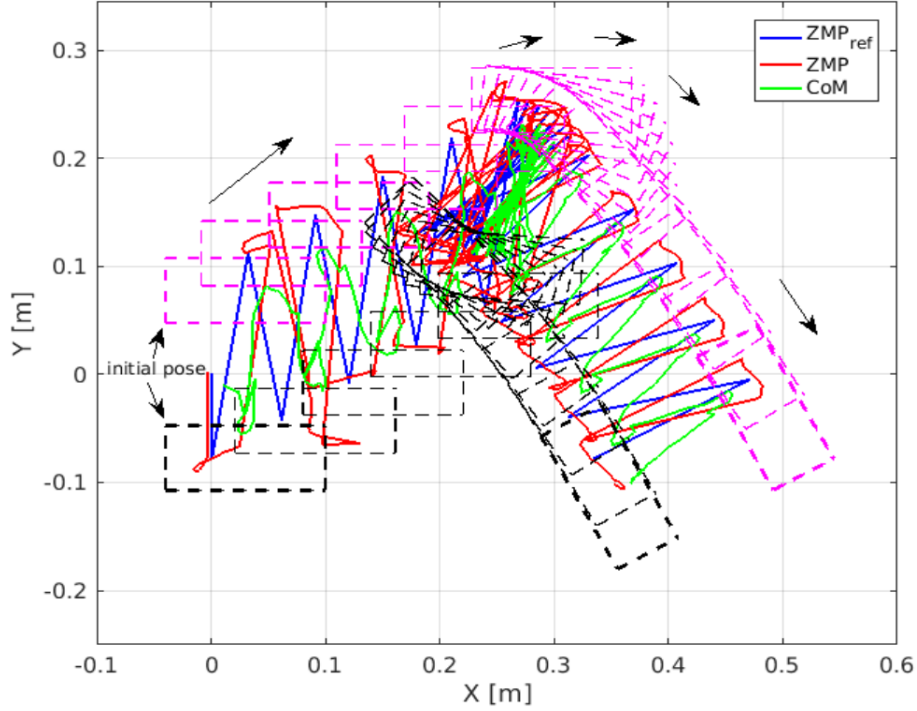


Figure 3.10: Footstep positions and orientations during the velocity tracking task. The magenta and black rectangles represent the support polygons of the left and right foot of the robot, respectively. The black arrows indicate the walking direction.

3.4.2 INTERACTION FORCE BASED REACTIVE WALKING

In this category two experiments were conducted: bimanual guidance and cooperative transportation tasks. They illustrate hypothetical collaboration tasks between the robot (follower) and a human with whom the robot interacts through forces. To move the robot in a given direction, the human who acts as leader can either push or pull the robot or even force it to rotate. The robot must generate stable walking motions that comply with the intention of the leader. The first uses the ZMP feedback and the second the arm forces/torques sensor information to detect the leader intention.

Figure 3.11 and Figure 3.12 show the results of the feedback based reactive walking. Throughout this experiment, the desired velocity of the robot was set to zero. At the beginning the velocity loop was open and then closed after 4s. The robot started by rocking around its initial position before being pulled continuously by its arms in the longitudinal direction and then pushed and pulled in the lateral direction. Furthermore, a torque around the vertical axis was exerted on the robot in both clockwise and anticlockwise direction before stopping the experiment, which lasted 60s.

The induced changes of the ZMP and the vertical moment with respect to their reference values are interpreted as perturbations to the desired state of the robot. To reduce these perturbations, the compensator shown in Figure

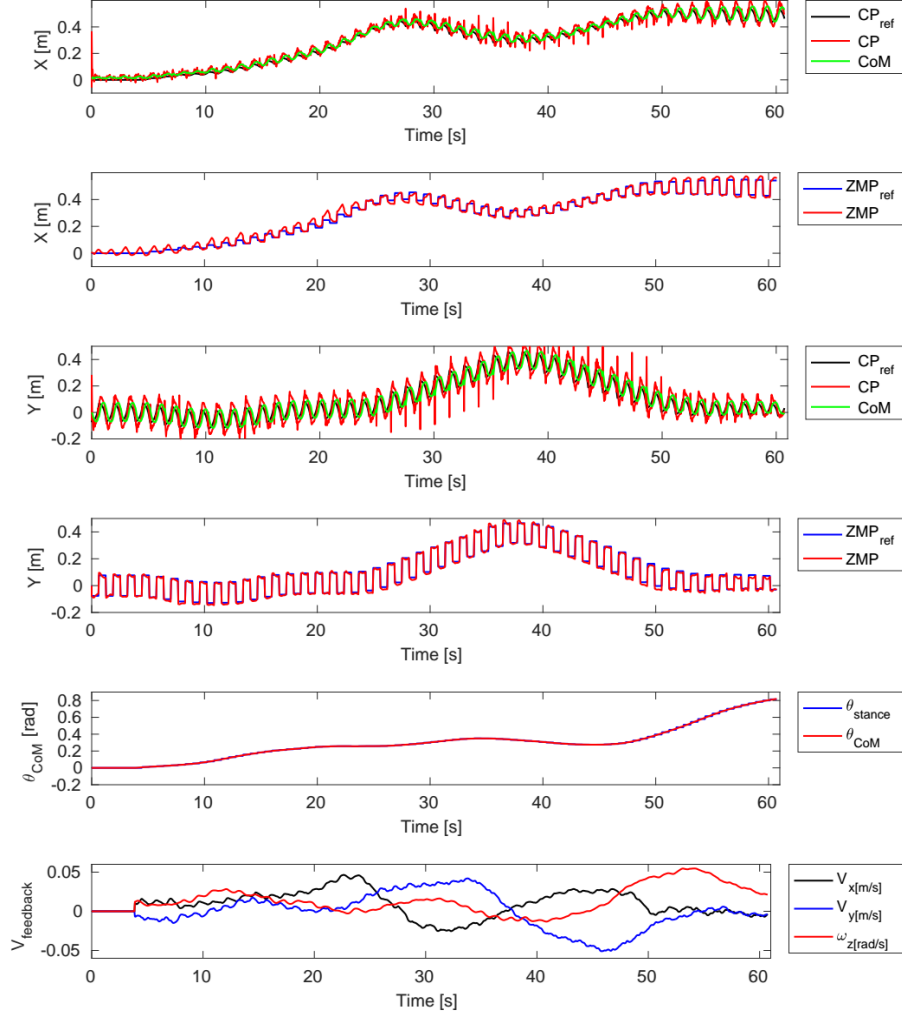


Figure 3.11: Trajectories of the CP, ZMP and CoM and feedback velocities generated by the controller in response to forces/torques exerted on the robot. The two figures at the top and in the middle represent the motion in the X and Y direction, respectively. The last two figures represent the orientation of the robot's base and the feedback velocities, respectively.

3.3, computed the feedback velocities shown in Figure 3.11. The short delay observed between the velocities and the robot's motion is due to a low-pass filter embedded in the compensator. In Figure 3.12, it can be seen that the generated trajectories keep the robot stable as its ZMP stays within the support polygon.

For the transportation task, the forces and torques applied on the robot, through the transported object, are measured by the robot's arms forces sensors and converted by an admittance law into velocities to be tracked by the robot. Some snapshots of this experiment are shown in Figure 3.13.

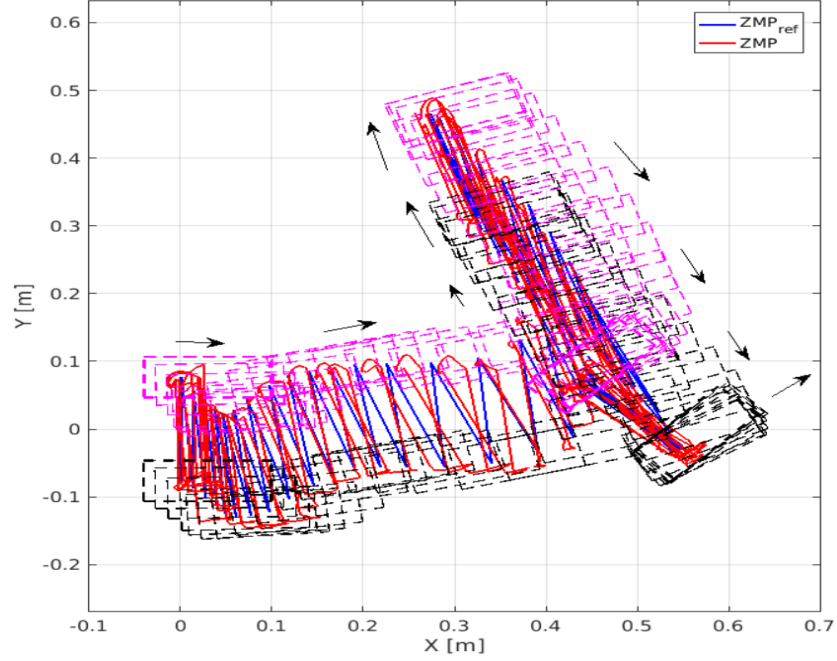


Figure 3.12: Footstep positions and orientations automatically generated during the interactive guidance task. The magenta and black rectangles represent the support polygons of the left and right foot of the robot, respectively. Their positions generated on-the-fly are represented by ZMP_{ref} . The black arrows indicate the walking direction.

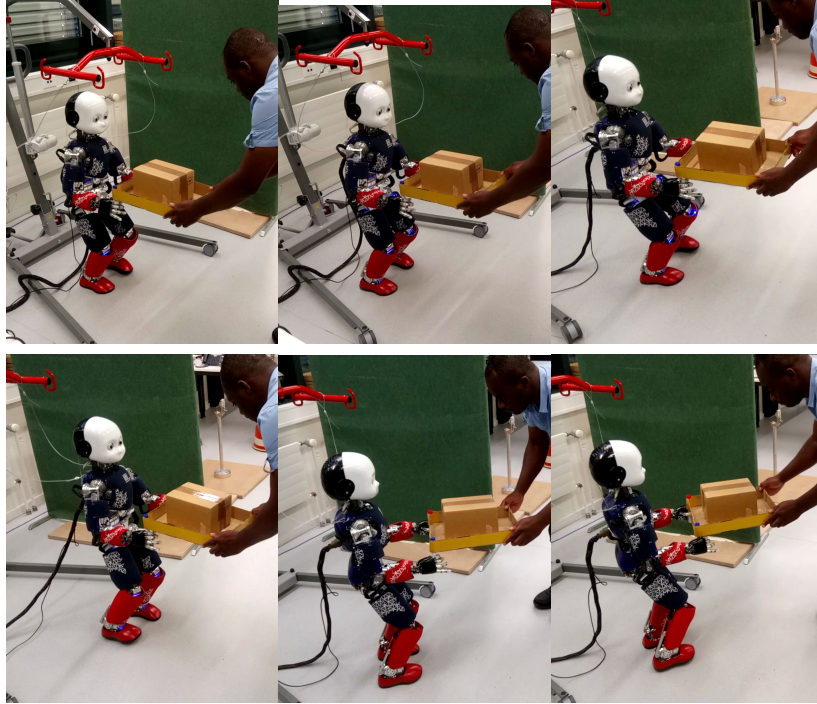


Figure 3.13: Illustration of reactive walking in a cooperative transportation task. The bimanual interaction forces sensed by the robot are converted into velocities that the robot tracks to comply to the human intentions.

3.5 Conclusion and Future Works

This Chapter presented a Capture-Point based walking controller able to generate reactively omnidirectional walking patterns for a biped robot and to stabilize the robot around them. By implementing the proposed controller on the humanoid robot iCub, its effectiveness was successfully demonstrated on two kinds of tasks where the classical walking approach based on footsteps planning could not apply. The first reactive walking experiment showed that the robot could track omnidirectional velocities and could even rotate around a spot when following a pure rotational velocity. The second experiment demonstrated how this ability to automatically generate stable omnidirectional walking motions could be further exploited in human-robot cooperative tasks. The robot, acting as a follower, successfully adapted its footsteps in order to comply with the intentions of the human, first in a guidance task and then in a cooperative transporting task.

As future works, the proposed algorithm, currently tested in position control mode, will be implemented with torque control in a whole body control framework. Thus, the linear and angular momentum of the robot could be explicitly regulated and the robot's motions made more compliant. This will reduce the high jerky motion observed during the experiments and could further improve the robustness of the walking.

Although the transition between phases could play a key role to produce dynamic bimanual coordination, it is often overlooked in the literature. We consider transitions between unconstrained and constrained phases. One of our objectives is to unify the control of the free and constrained motion phases.

BIMANUAL COORDINATED MOTION AND FORCES FOR COOPERATIVE COMPLIANT MANIPULATION

This Chapter presents our approach to achieving coordinated bimanual reach-to-grasp and cooperative compliant manipulation of an object by a humanoid robot. After briefly reviewing relevant works, it introduces our coordination approach in the context of whole-body control. It describes our formulation of the free-motion coordination based on dynamical systems and a shrinkable virtual object. Then, it presents our approach to cooperative manipulation with force generation based on quadratic programming. Finally, it presents validation results and discusses them before concluding¹.

4.1 Introduction

HUMANOID robots with their human-like structure, and their combined manipulation and locomotion capabilities are expected to work alongside and collaborate with humans. Endowing these robots with bimanual manipulation skills would allow them to accomplish tasks that are too complex or delicate for one hand. As stated in Section 1.3, we seek to realize bimanual manipulation tasks that consider the coordinated free and constrained motion phases with quasi-static transitions when performed by a humanoid robot. More specifically, we want a humanoid robot to perform coordinated reach-to-grasp tasks that smoothly transition to cooperative manipulation tasks of an object while being compliant.

Solving this problem is challenging as the humanoid robot, with its particular structure and dynamics, must be balanced and the two arms must be coordinated both in motion and force to ensure compliant interaction. As outlined in Section 2.4.2, previous works, in general, have only partially addressed the dual-arm coordination problem on such robots. The works that considered the balance in addition to the coordination problem assumed already grasped

¹Supplementary materials:

- codes: https://github.com/epfl-lasa/icub_whole_body_task_controller
- video: <https://youtu.be/odoaZ80h7IA>

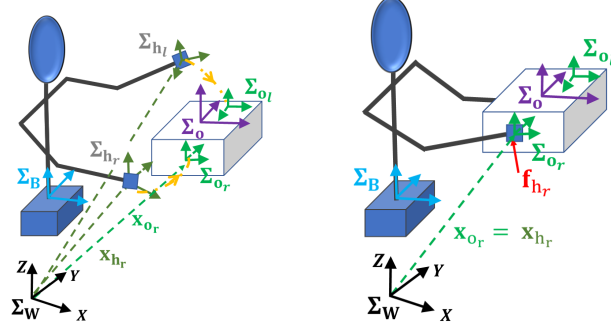


Figure 4.1: The two situations of the considered during bimanual manipulation tasks. (left): Reaching-to-grasp with hands in free-motion dynamics. (right): Cooperative manipulation with hands in constrained dynamics. Σ_W and Σ_o are the world and the object frame. Σ_{hl} and Σ_{hr} denote the robot's left and right hands frames, respectively, whereas Σ_{ol} and Σ_{or} denote their respective desired grasping configurations on the object side.

objects and focused on the cooperative manipulation task. The works that considered the reach-to-grasp and the cooperative manipulation phases including the transitions between them were not concerned with the balance problem.

Thus, this thesis proposes an algorithm integrated into a QP-based whole-body controller to achieve bimanual motion coordination and cooperative manipulation on a humanoid robot while enforcing the robot's balance. Inspired by (Salehian et al., 2016a), our approach uses dynamical systems and exploits a virtual object with its associated constraints to achieve robust coordination of the two robotic hands. However, our virtual object is shrinkable to enable soft transitions between non-contact and contact phases by ensuring smooth transitions between virtual constraints in free motion and real constraints when grabbing the object with unilateral forces. Assuming approximate knowledge of the object's mass and the friction coefficients, our approach uses QP to generate online interaction wrenches that achieve stable grasp and manipulation tasks while enforcing explicitly the contacts constraints. The stability and convergence of the motion coordination scheme are proven and the proposed algorithm is validated.

4.2 Problem definition

Consider the bimanual task illustrated in Figure 4.1, where a humanoid robot is required to reach and grasp a static or moving object for manipulation purposes. The dynamics of the humanoid robot, the object and their coupling are described by the equations (2.2.1), (2.2.2) and (2.2.3).

To control the humanoid robot with its multiple task objectives and constraints, we assume that we have a whole-body controller that executes tasks specified in terms of desired motion and forces. We further assume that the whole-body controller is based on quadratic programming (QP) (De Lasa et al., 2010; Mansard, 2012; Saab et al., 2013; Escande et al., 2014; Herzog et al.,

2016) that can enforce explicitly inequality constraints such as the unilateral constraints of the feet contact forces necessary for the robot balance. Hence, our QP-based whole-body controller that computes the joint torques $\mathbf{\Gamma}^*$ necessary to perform the desired manipulation task with desired hand motion $\ddot{\mathbf{x}}_h^*$, interaction forces \mathbf{f}_h^* and balance task $\dot{\mathbf{h}}^*$ is formulated as follows

$$\mathbf{\Gamma}^* = \underset{\mathbf{\Gamma}, \ddot{\mathbf{q}}, \mathbf{f}_f, \boldsymbol{\epsilon}}{\operatorname{argmin}} \mathcal{J}_{\text{wb}}(\mathbf{\Gamma}, \ddot{\mathbf{q}}, \mathbf{f}_f, \boldsymbol{\epsilon}) \quad (4.2.1)$$

$$\text{subject to } \mathbf{M}\ddot{\mathbf{q}} - \mathbf{S}_r^\top \mathbf{\Gamma} - \mathbf{J}_f^\top \mathbf{f}_f = -\mathbf{b} + \mathbf{J}_h^\top \mathbf{f}_h^* \quad (4.2.2)$$

$$\mathbf{G}_{c_f} \mathbf{f}_f + \boldsymbol{\epsilon}_h = \dot{\mathbf{h}}^* - \mathbf{G}_{c_h} \mathbf{f}_h^* - \mathbf{f}_{\text{gvt}} \quad (4.2.3)$$

$$\mathbf{J}_t \ddot{\mathbf{q}} + \boldsymbol{\epsilon}_t = \ddot{\mathbf{x}}_t^* - \dot{\mathbf{J}}_t \dot{\mathbf{q}} \quad (4.2.4)$$

$$\mathbf{C}_u \mathbf{u} \leq \mathbf{d}_u \quad (4.2.5)$$

where the cost function is defined as

$$\mathcal{J}_{\text{wb}}(\mathbf{\Gamma}, \ddot{\mathbf{q}}, \mathbf{f}_f, \boldsymbol{\epsilon}) = \|\mathbf{W}_\Gamma \mathbf{\Gamma}\|^2 + \|\mathbf{W}_{\ddot{\mathbf{q}}} \ddot{\mathbf{q}}\|^2 + \|\mathbf{W}_f \mathbf{f}_f\|^2 + \|\mathbf{W}_\epsilon \boldsymbol{\epsilon}\|^2 \quad (4.2.6)$$

where all decision variables are gathered in a vector $\mathbf{u} \triangleq [\mathbf{\Gamma}^\top, \ddot{\mathbf{q}}^\top, \mathbf{f}_f^\top, \boldsymbol{\epsilon}^\top]^\top$, where $\boldsymbol{\epsilon} = [\boldsymbol{\epsilon}_h^\top \quad \boldsymbol{\epsilon}_t^\top]^\top$ are slack variables used to soften the tasks constraints. \mathbf{W}_i are the weight matrices associated to each decision variable. The constraints (4.2.2) ensures the dynamic consistency of the obtained solution. The constraint (4.2.3) represents the balance task, where \mathbf{G}_{c_f} and \mathbf{G}_{c_h} are transformation matrices mapping the feet and hands wrenches to the center of mass (CoM) of the robot, respectively (see Appendix A.1). \mathbf{f}_{gvt} is the gravity wrench acting upon it. $\dot{\mathbf{h}}^*$ denotes the desired rate of change of the robot's centroidal momentum. The constraint (4.2.4) represents the overall motion task of all the robot's end-effectors, including the joints posture. $\mathbf{J}_t \in \mathbb{R}^{n_{\text{task}} \times (n^D)}$ and $\ddot{\mathbf{x}}_t^* \in \mathbb{R}^{n_{\text{task}}}$ represent respectively the stacked Jacobian matrix and its associated desired acceleration vector ($\ddot{\mathbf{x}}_h^* \subset \ddot{\mathbf{x}}_t^*$). Finally, constraint (4.2.5) gathers all inequality constraints, namely the torque and joint limits, the friction cone and center of pressure constraints.

Given the above whole-body controller, our main problem can now be stated as follow:

P4.1: How to generate $\dot{\mathbf{h}}^* = [\dot{\mathcal{P}}^{*\top} \quad \dot{\mathcal{L}}^{*\top}]^\top \in \mathbb{R}^6$ such that the robot remains balanced and stable throughout the task while being compliant?

P4.2: How to generate the overall motion task $\ddot{\mathbf{x}}_t^*$, more specifically, the motion of the hands $\ddot{\mathbf{x}}_h^* = [\ddot{\mathbf{x}}_{h_l}^\top \quad \ddot{\mathbf{x}}_{h_r}^\top]^\top \in \mathbb{R}^{12}$ such that the robot reaches the object to be grasped in a coordinated manner?

P4.3: Once the object grasped, how to determine the necessary hands wrenches $\mathbf{f}_h^* = [\mathbf{f}_{h_l}^{*\top} \quad \mathbf{f}_{h_r}^{*\top}]^\top \in \mathbb{R}^{12}$ to be applied on the object so as to maintain a stable grasp and achieve the desired manipulation task?

The problem **P4.1** concerns the balance task, which is defined here by the desired rate of the robot centroidal momentum $\dot{\mathbf{h}}^*$. In this thesis, $\dot{\mathbf{h}}^*$ is designed as a proportional derivative (PD) law that defines the robot's desired CoM behavior and floating base orientation.

$$\dot{\mathbf{h}}^* = -\mathbf{D}_C \mathbf{h} - \mathbf{K}_C \begin{bmatrix} (\mathbf{x}_C - \mathbf{x}_C^*) \\ \boldsymbol{\sigma}(\boldsymbol{\phi}_B, \boldsymbol{\phi}_B^*) \end{bmatrix} \quad (4.2.7)$$

where $\mathbf{D}_C \in \mathbb{R}^{6 \times 6}$ and $\mathbf{K}_C \in \mathbb{R}^{6 \times 6}$ denote the stiffness and damping matrices, respectively. $\mathbf{x}_C \in \mathbb{R}^3$ and $\mathbf{x}_C^* \in \mathbb{R}^3$ are the current and desired position of the CoM, respectively. $\boldsymbol{\sigma}(\boldsymbol{\phi}_B, \boldsymbol{\phi}_B^*) \in \mathbb{R}^3$ represents the relative orientation between the current orientation of the floating base $\boldsymbol{\phi}_B$ and its desired value $\boldsymbol{\phi}_B^*$.

This Chapter focuses on **P4.2** and **P4.3**, which define the bimanual coordination problem with the desired coordinated motion $\dot{\mathbf{x}}_h^*$ and force \mathbf{f}_h^* . Before addressing this problem, we make the following assumptions for the object

- A4.1:** the object is rigid with known shape and dimensions, and known inertia properties (mass and moments of inertia).
- A4.2:** the contacts between the object and end-effectors are unilateral and frictional. They are geometrically modeled by contact surfaces (convex hull of contact points).
- A4.3:** the grasping points are known and correspond to the centers of the contact surfaces, they have coordinate frames attached to them to describe their relative positions and orientations.
- A4.4:** The contact surfaces comply with a Coulomb friction model (Murray et al., 1994) and their associated friction coefficients are known (at least their minimum values).

Next, we present our approach to generate the bimanual coordinated motion and force with smooth transition between the manipulation phases.

4.3 Proposed Approach

To address the bimanual coordination problem, we propose to encode both free and constrained motions of the robot's hands with autonomous dynamical systems (DS). Thus, we propose to generate the desired hands motion \mathbf{v}_h^d and wrench \mathbf{f}_h^d respectively as follows

$$\mathbf{v}_h^d = (1 - \lambda_r) f_u(\mathbf{x}_h, \boldsymbol{\phi}_h) + \lambda_r \mathbf{G}_o^\top f_c(\mathbf{x}_o, \boldsymbol{\phi}_o) \quad (4.3.1)$$

$$\mathbf{f}_h^d = \begin{cases} \mathbf{f}_h(f_c(\mathbf{x}_o, \boldsymbol{\phi}_o), \mathbf{M}_o, \mathbf{b}_o, \mathbf{G}_o) & \text{if } \lambda_r = 1 \\ 0 & \text{if } \lambda_r = 0 \end{cases} \quad (4.3.2)$$

where $f_u(\mathbf{x}_h, \phi_h) \in \mathbb{R}^{12}$ is a DS generating the unconstrained reach-to-grasp motion of the robot's hands, where \mathbf{x}_h and ϕ_h represent the positions and orientations of the dual-arm end-effectors, respectively. $f_c(\mathbf{x}_o, \phi_o) \in \mathbb{R}^6$ is a DS that generates the desired object's motion and provides the constrained motion of the robot's hands through the mapping $\mathbf{G}_o^\top f_c(\mathbf{x}_o, \phi_o)$, where $\mathbf{G}_o \in \mathbb{R}^{6 \times 12}$ is the object's grasp matrix. \mathbf{x}_o and ϕ_o represent the position and orientation of the object. $\lambda_r \in [0, 1]$ is binary variable that becomes 1 when the object is grasped and 0 otherwise. \mathbf{M}_o and \mathbf{b}_o are the object's inertia matrix and Coriolis and gravity forces vector, respectively. A block diagram of the proposed bimanual control scheme is shown in Figure 4.2.

4.3.1 BIMANUAL COORDINATED REACH-TO-GRASP MOTION

To generate coordinated motion of the robot's hands, we use a *virtual object* as in (Salehian et al., 2016a). However, to allow bimanual grasping through unilateral contacts, we consider a *virtual object* initially bigger than the real object (see Figure 4.3). As the robot's hands approach the real object, the *virtual object* shrinks to match the size of the real object. This leads to the closing of the robot's hands aperture and thereby the grasping of the object. Overall, the coordinated motion of the robot's hands results from the combination of three coupled dynamical systems (DS): one for the motion of the hands to the scaled up *virtual object* (*synchronization*: $\Sigma_{h_i} \rightarrow \Sigma_{v_i}$), one for the motion of the scaled up *virtual object*'s frame to the real object's frame (*approach*: $\Sigma_a \rightarrow \Sigma_o \equiv \Sigma_{v_i^*} \rightarrow \Sigma_{o_i}$), and the last for the shrinking motion (*grasping*: $\Sigma_{v_i} \rightarrow \Sigma_{v_i^*}$).

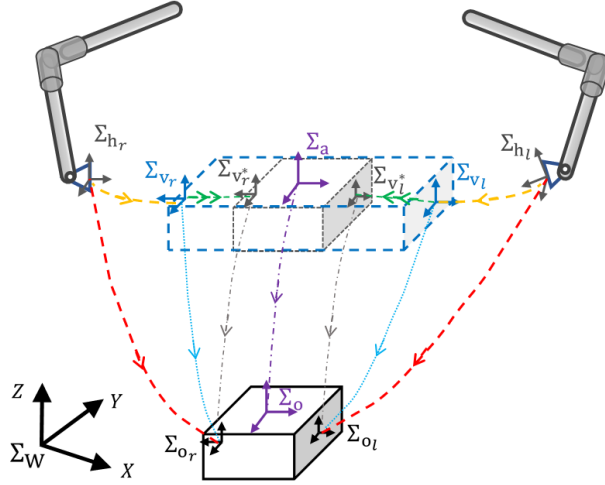


Figure 4.3: Geometrical variables of the reach-to-grasp task with configurations of the real object (black continuous lines) and *virtual object* (blue broken line in its scaled up size and black dash lines when shrunk to the real object's size). For index ($i \equiv \text{left, right}$) Σ_{h_i} and Σ_{o_i} are respectively the i^{th} hand's and object's grasp configuration frames. Σ_{v_i} and $\Sigma_{v_i^*}$ denote the i^{th} frame of the scaled up and real object's size *virtual object*, respectively. Σ_o is the frame of the real object, while Σ_a is the frame attached to the *virtual object* with its position as middle of Σ_{h_l} and Σ_{h_r} and its orientation that of Σ_o .

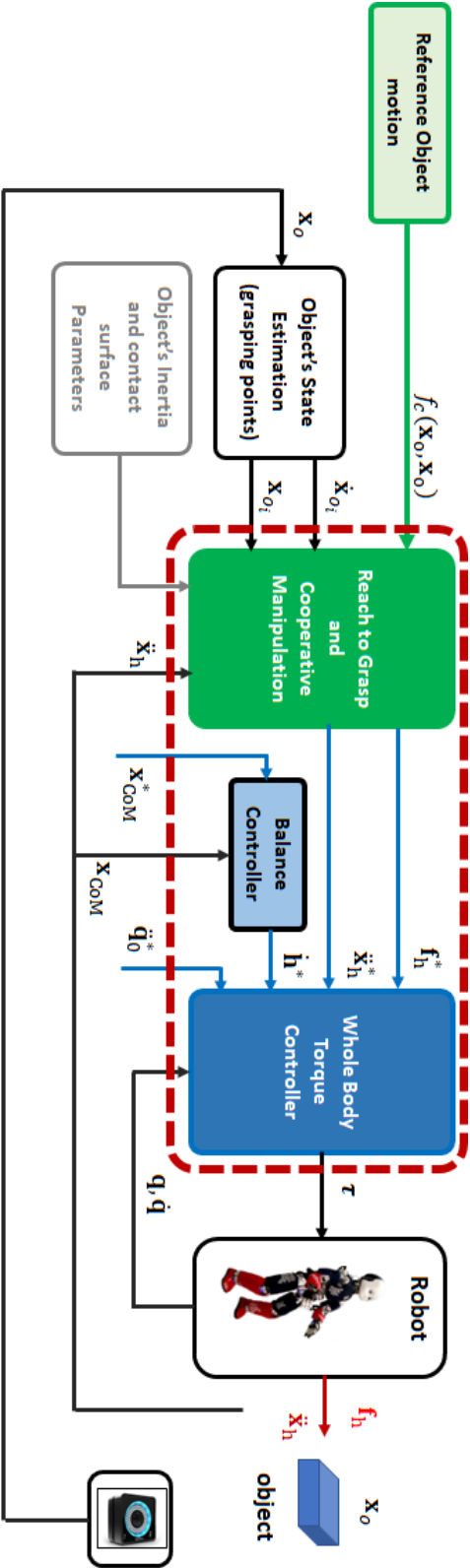


Figure 4.2: Block diagram of the proposed control architecture for bimanual grabbing and cooperative manipulation on a humanoid robot

Thus, using linear DS, we propose to generate the desired motion of the i^{th} robot's hand as follows

$$\mathbf{v}_{h_i}^d = f_{u_i}(\mathbf{x}_h, \phi_h) = \mathbf{C}_{h_i}(\boldsymbol{\xi}_{h_i})\boldsymbol{\xi}_{h_i} \quad (4.3.3)$$

where $\mathbf{v}_{h_i}^d = [\dot{\mathbf{x}}_{h_i}^\top \quad \boldsymbol{\omega}_{h_i}^\top]^\top \in \mathbb{R}^6$ is the desired velocity twist of the i^{th} hand's frame Σ_{h_i} . $\boldsymbol{\xi}_{h_i} \triangleq [\boldsymbol{\xi}_{h_i v_i}^\top \quad \boldsymbol{\xi}_{v_i^* o_i}^\top \quad \boldsymbol{\xi}_{v_i v_i^*}^\top]^\top \in \mathbb{R}^{18}$ is the overall state vector, concatenating the state vectors of the three DS associated with the *synchronization*, the *approach* and the *grasping* motion phases, respectively. $\mathbf{C}_{h_i} \in \mathbb{R}^{6 \times 18}$ maps the states in local frames to velocity twist in world frame. It is defined as

$$\mathbf{C}_{h_i} = [\mathbf{L}_{h_i v_i}^{-1} A_{h_i v_i} \quad \mathbf{L}_{v_i v_i^*}^{-1} A_{v_i^* o_i} \quad \mathbf{L}_{v_i^* o_i}^{-1} A_{v_i v_i^*}] \quad (4.3.4)$$

where $\mathbf{L}_{jk} \in \mathbb{R}^{6 \times 6}$ with $jk \equiv \{h_i v_i, v_i v_i^*, v_i^* o_i\}$ is a differential map such that $\dot{\boldsymbol{\xi}}_{jk} = \mathbf{L}_{jk}(\dot{\mathbf{x}}_j - \dot{\mathbf{x}}_k)$ (see Appendix C.2 for more details). The matrices $A_{jk} \in \mathbb{R}^{6 \times 6}$ correspond to the dynamic matrices of each of the DS governing the coordination such that

$$\dot{\boldsymbol{\xi}}_{jk} = A_{jk} \boldsymbol{\xi}_{jk} \quad (4.3.5)$$

The state vectors $\boldsymbol{\xi}_{jk} \in \mathbb{R}^6$ are defined as relative poses between frames (position of frame Σ_j relative to Σ_k and expressed in Σ_k , and axis/angle representation of the orientation of Σ_j relative to Σ_k). Using such representation allows to better handle both position and orientation simultaneously. Moreover, for all motion phases (*synchronization*, *approach* and *grasping*), it yields autonomous DS (with equilibrium at zero ($\boldsymbol{\xi}_{jk}^d = [\mathbf{0}_{1 \times 3} \quad \mathbf{0}_{1 \times 3}]^\top$)).

The *virtual object* is key to the coordination as it imposes constraints on the motion of the dual-arm hands. Its position is the average between the robot's hands, and its orientation is equal to the real object's when the latter is reachable. We define state-dependent couplings between the state vectors to ensure smooth transitions and coordination between the task phases. For instance, coupling the grasping and approach phases such that the aperture of the robot hands is closed only when the hands are near the object. Thus, we design $\boldsymbol{\xi}_{jk}$ as

$$\boldsymbol{\xi}_{jk} \triangleq \begin{bmatrix} {}^w \mathbf{R}_k^\top [{}^w \mathbf{x}_j - \gamma_{jk} {}^w \mathbf{x}_k - (1 - \gamma_{jk}) {}^w \mathbf{x}_j^d] \\ \theta \boldsymbol{\mu}({}^w \mathbf{R}_j, {}^w \mathbf{R}_k, {}^w \mathbf{R}_j^d, \gamma_{jk}) \end{bmatrix} \quad (4.3.6)$$

where ${}^w \mathbf{x}_j$ and ${}^w \mathbf{R}_j$ respectively ${}^w \mathbf{x}_k$ and ${}^w \mathbf{R}_k$ are the position and rotation matrix of the frame Σ_j respectively Σ_k with respect to the world Σ_w . ${}^w \mathbf{x}_j^d$ and ${}^w \mathbf{R}_j^d$ denote respectively desired standby position and rotation matrix of Σ_j relative to Σ_w . $\gamma_{jk} \in \mathbb{R}^1$ is a state-dependent coordination factor that smoothly varies between $[0, 1]$. $\theta \boldsymbol{\mu}$ is the axis/angle representation of the orientation. It varies from $\theta \boldsymbol{\mu} = \theta \boldsymbol{\mu}({}^d \mathbf{R}_j)$ when $\gamma_{jk} = 0$ towards $\theta \boldsymbol{\mu} = \theta \boldsymbol{\mu}({}^k \mathbf{R}_j)$ when $\gamma_{jk} = 1$.

For the *synchronization* phase, γ_{jk} depends on the reachability of the object

as in [Salehian et al. \(2016a\)](#). However, for the *approach* and *grasping* phases, γ_{jk} is defined as

$$\gamma_{jk} = 1 - \exp\left(-\frac{\beta}{\|\boldsymbol{\xi}_{jk-1}\| + \varepsilon}\right) \quad (4.3.7)$$

where $\|\boldsymbol{\xi}_{jk-1}\|$ denotes the error norm of the *synchronization* and *approach* phase, respectively. $\beta > 0$ is a scalar that tunes the variation of γ_{jk} .

The proposed DS-based scheme is stable and asymptotically converges to its equilibrium, that is: $\lim_{t \rightarrow \infty} \boldsymbol{\xi}_{h_i} = \mathbf{0} \ \forall i \in \{\text{left}, \text{right}\}$, if the following condition is satisfied (the proof is provided in appendix [C.3](#)).

$$\mathbf{A}_{h_i}^\top \mathbf{P}_{h_i} + \mathbf{P}_{h_i} \mathbf{A}_{h_i} = -\mathbf{Q}_{h_i} \quad \forall i \in \{\text{left}, \text{right}\} \quad (4.3.8)$$

where $\mathbf{P}_{h_i} \in \mathbb{R}^{18 \times 18}$ and $\mathbf{Q}_{h_i} \in \mathbb{R}^{18 \times 18}$ are two positive definite matrices, and $\mathbf{A}_{h_i} \triangleq \text{diag}\{A_{h_i v_i}, A_{v_i^* o_i}, A_{v_i v_i^*}\}$. When the DS reaches its equilibrium, the pose of the i^{th} hand reaches its desired configuration on the real object with a vanishing velocity (i.e. $\lim_{t \rightarrow \infty} \dot{\boldsymbol{\xi}}_{h_i} = \mathbf{0}$). This yields a smooth transition from the non-contact to contact phase.

4.3.2 QP-BASED BIMANUAL COOPERATIVE MANIPULATION

When the robot's hands have reached the object, how well the object is grasped depends on the induced internal wrenches. The contacts being unilateral with limited friction, insufficient or excessive wrenches might result in contact slippage or damage of the object and/or end-effectors. This is particularly important, when the applied wrenches have to be adjusted to satisfy varying task requirements.

Thus, given an object manipulation task specified by the desired effective wrench $\mathbf{f}_o^d = \mathbf{f}_o(f_c(\mathbf{x}_o, \phi_o), \mathbf{M}_o, \mathbf{b}_o, \mathbf{G}_o) \in \mathbb{R}^6$, the problem is to determine each robot's individual wrench contribution to produce \mathbf{f}_o^d with minimal internal wrenches and satisfy the contact constraints. For instance, if the desired task is encoded as $\mathbf{v}_o^d = f_c(\mathbf{x}_o, \phi_o)$, the object's effective wrench can be computed using a passive DS approach ([Kronander and Billard, 2016](#)) as ²

$$\mathbf{f}_o^d = -\mathbf{D}_o(\mathbf{x}_o, \phi_o)(\mathbf{v}_o - f_c(\mathbf{x}_o, \phi_o)) + \mathbf{b}_o \quad (4.3.9)$$

where $\mathbf{D}_o(\mathbf{x}_o, \phi_o) \in \mathbb{R}^{6 \times 6}$ denotes a state varying damping matrix.

Considering assumptions [A4.1-A4.4](#), we address this problem from an object-centered perspective ([Okamura et al., 2000](#)) and propose to use quadratic programming (QP) to generate minimum grasp and manipulation wrench, \mathbf{f}_h^d , that

² Alternatively, if the desired acceleration of the object $\dot{\mathbf{v}}_o^d$ is available, the inverse dynamics of the object can be used: $\mathbf{f}_o^d = \mathbf{M}_o \dot{\mathbf{v}}_o^d + \mathbf{b}_o$

accomplishes the desired task and enforces the contact constraints. Hence,

$$\mathbf{f}_h^d = \operatorname{argmin} \|\mathbf{G}_o \mathbf{f}_h - \mathbf{f}_o^d\|^2 + \|\mathbf{W}_h \mathbf{f}_h\|^2 \quad (4.3.10)$$

$$s.t. \quad \mathbf{C}_{f_h} \mathbf{f}_h \leq \mathbf{d}_{f_h} \quad (4.3.11)$$

$$\mathbf{Q}_{ho} \mathbf{f}_h = \mathbf{0} \quad (4.3.12)$$

where \mathbf{W}_h is a regularization matrix. (4.3.11) encapsulates the constraints on the contact wrenches (i.e. unilateral, friction cone, center of pressure), with \mathbf{C}_{f_h} and \mathbf{d}_{f_h} their associated constraints matrix and vector (see Appendix A.2). \mathbf{Q}_{ho} represents the constraint matrix associated with the complementary condition (4.3.12) between the force components normal to the contact, denoted $f_{h_{iz}}$, and the normal distance to the contact, denoted $z_{h_{io_i}}$ (i = left and right). For the i^{th} hand, this condition translates into

$$z_{h_{io_i}} f_{h_{iz}} = 0, \quad f_{h_{iz}} \geq 0, \quad z_{h_{io_i}} \geq 0 \quad (4.3.13)$$

Condition 4.3.13 ensures that the QP generates forces only when the contacts are established (if $f_{h_{iz}} > 0$ then $z_{h_{io_i}} = 0$, and if $z_{h_{io_i}} > 0$ then $f_{h_{iz}} = 0$).

QP (4.3.10)-(4.3.12) performs an optimal distribution of \mathbf{f}_o^d between the two robot's hands based on the coupling (2.2.3) with unilateral contact forces.

4.4 Validation

The validation of the proposed controller is carried out on a simulated humanoid robot iCub (Metta et al., 2008) in Gazebo environment. In our implementation, 29 out the 51 degrees-of-freedom of the iCub robot were controlled in torque, in particular three joints in the torso, seven in each hand and six in each leg. The dynamics of the robot was computed using *iDynTree* (Nori et al., 2015) and the *yarpWholeBodyInterface* module was used for the communication with the robot. All QPs (i.e. for the whole-body and the grasp wrenches) are solved using CVXGEN (Mattingley and Boyd, 2012). The overall task is solved in 2 to 5 ms on a 3.6 GHz i7 PC. The controller is run at 100Hz.

Three main scenarios were considered : 1) coordinated reaching and manipulation task, 2) coordinated reaching and grasping of a moving object, and 3) whole-body and object-level compliant interaction. A video of the corresponding simulations is provided as supplementary material.

4.4.1 BIMANUAL COORDINATED REACHING AND MANIPULATION TASK

In this scenario, we evaluate the motion coordination capability of our controller and its ability to generate wrenches that perform a given manipulation task with the object. The bimanual task consists of reaching an object of 0.5 kg, lifting it

to a height of 0.6 m and tracking in the YZ plane a circular trajectory define by two sinusoidal motion along Y and Z axis. Figure 4.4 provides some snapshots of the task execution.



Figure 4.4: Snapshots of the task consisting of coordinated reaching and grasping of an object (first raw) and manipulation (second and third raws).

The initial poses (with Euler angle) of the left and right hands are respectively $\mathbf{x}_{h_l}(t_o) = (0.187, 0.208, 0.639)$ and $\phi_{h_l}(t_o) = (-1.271, -0.233, 0.190)$, and $\mathbf{x}_{h_r}(t_o) = (0.187, -0.207, 0.639)$ and $\phi_{h_r}(t_o) = (-1.871, 0.232, 0.190)$. The corresponding grasp poses on the object are respectively $\mathbf{x}_{o_l}(t_o) = (0.221, 0.091, 0.245)$ and $\phi_{o_l}(t_o) = (-1.571, -0.00, 0.00)$ and $\mathbf{x}_{o_r}(t_o) = (0.221, -0.070, 0.245)$ and $\phi_{o_r}(t_o) = (-1.571, -0.00, 0.00)$. The two hands are required to move in a coordinated manner their positions and orientations so as to reach the object at the same time. Once the object reached, the grasping wrenches are applied and the object is lifted.

The reaching phase is evaluated through the poses errors between the robot's hands and their desired grasp locations on the object. The time evolution of these errors and the grasping wrenches are shown in Figures 4.5 and 4.6, respectively. It can be seen that the robot reached the object around $t = 20\text{ s}$ and lifted it up. The zero pose errors during the task mean that the hands have reached and remain at their desired grasping locations, which indicates stability of contacts. The computed forces, as shown in Figure 4.6, were saturated to keep them within the limits of the robots. At $t = 40\text{ s}$, when the object is required to follow the sinusoidal trajectories in the Y and Z directions, one can notice that the forces generated in the X direction have also sinusoidal profiles. This is because the controller adapts the applied forces in order to maintain the desired x motion at its desired values. Similar sinusoidal patterns are also followed by their associated moments.

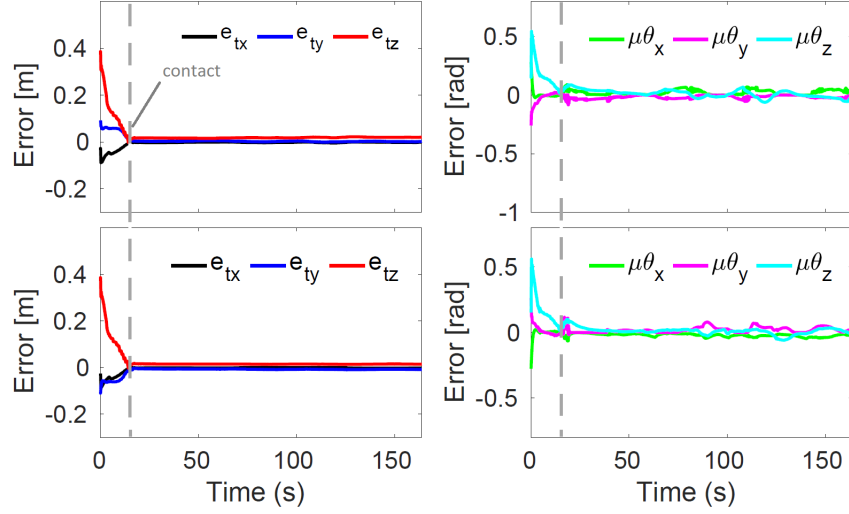


Figure 4.5: Position and orientation errors of the left and right hands with respect to their desired grasping locations on the object during the reach-to-grasp (before contacts) and manipulation (after contacts) task. The contacts establishment is indicated by the vertical line. The zero pose errors mean that the hands have reached and stay at their desired grasping points

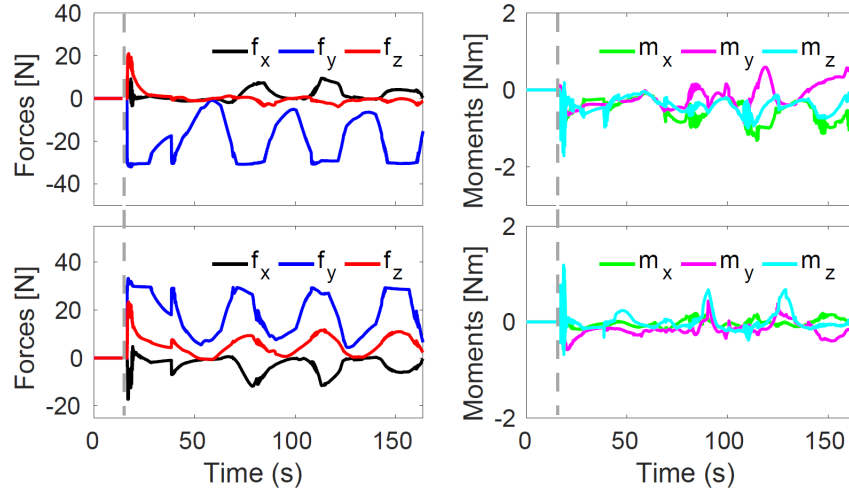


Figure 4.6: Applied forces and moments of the left and right hands on the object throughout the reach-to-grasp and manipulation task. They remain at zero until $t = 20$ s when the contacts with object are established

The *virtual object* that helps to coordinate the reaching phase and its 3D trajectory is shown in Figure 4.7, whereas Figure 4.8 shows the 3D components of the object's motion during the manipulation phase. It can be observed that the object follows fairly well its desired trajectories.

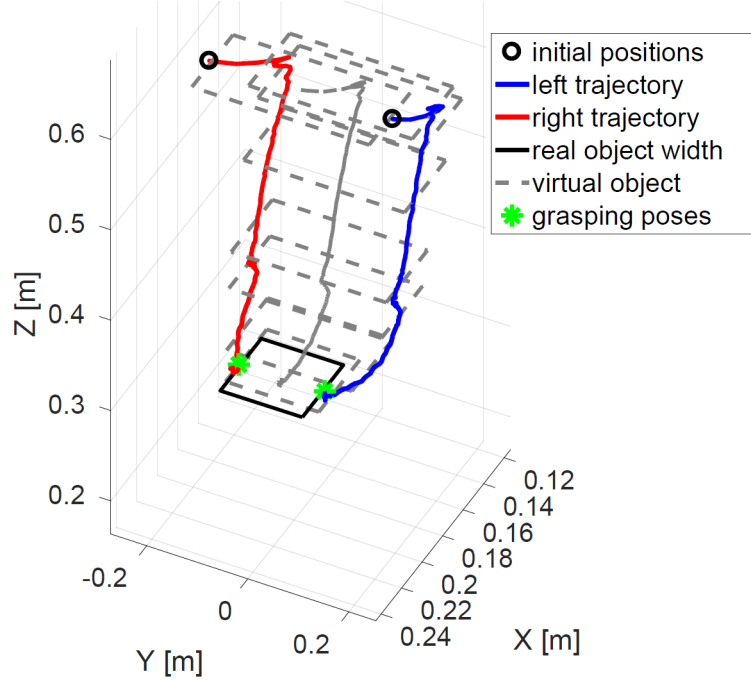


Figure 4.7: Illustration of the shrinkable virtual object as it is used to coordinate the robot's hands motion during the reaching phase

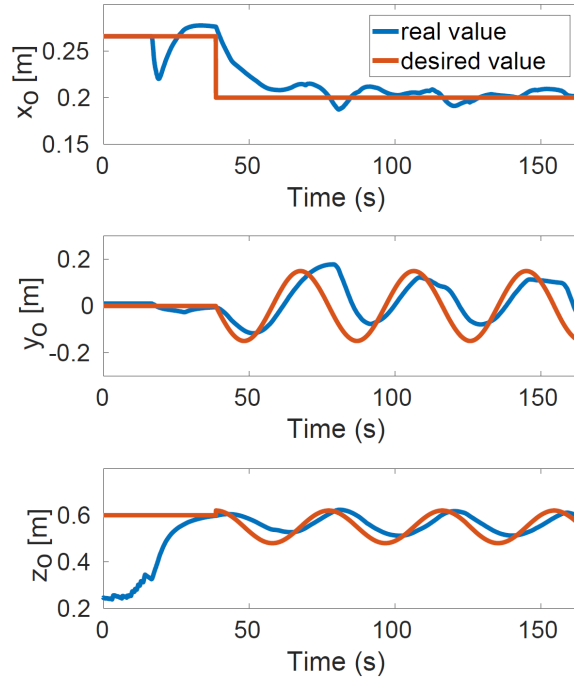


Figure 4.8: 3D trajectories realized by the object compared to the desired trajectories during the manipulation phase

The robustness of the wrenches generator was also tested by reaching and grasping other objects such as a cylinder and a sphere as illustrated in Figure 4.9.

In those case, the contact surface reduces into a line and a point, respectively.

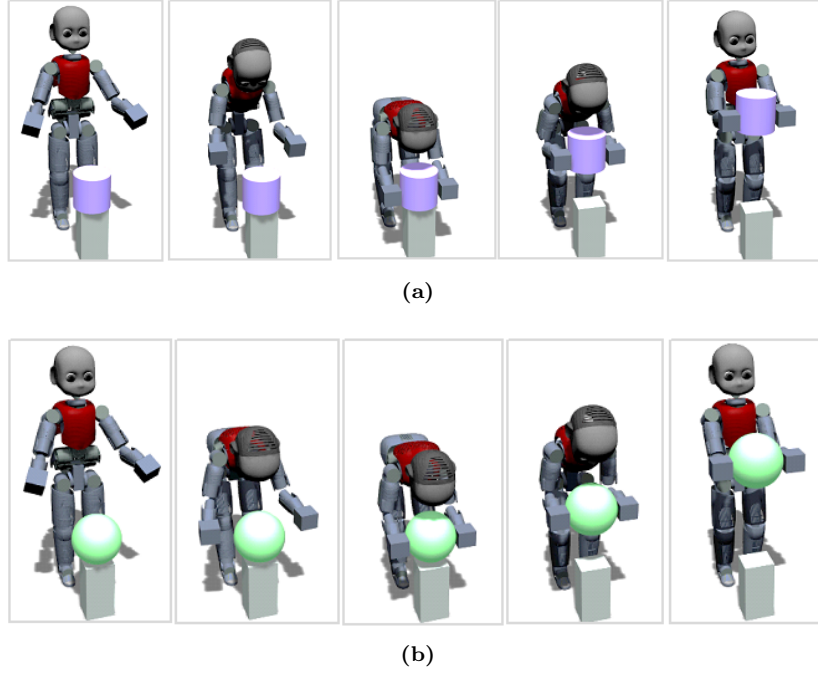


Figure 4.9: Simulations of bimanual reaching and grasping object with reduced contact surfaces: (a) cylinder (line contact) (b) sphere (point contact)

To further assess our controller, we run multiple simulations of the task. The ranges of initial poses (positions and orientations) are summarized in Table 4.1. Figure 4.10, provides a visualization of the 3D trajectories of 15 simulated experiments of coordinated reach-to-grasp task (starting from different initial poses). The average final reaching errors are shown in Figure 4.11.

Table 4.1: Ranges of initial poses of the robot’s hands and the object’s grasping points during the assessment of the reaching and grasping task. The poses are expressed with respect to the world frame, with the orientation using Euler angles.

(a)

	$x [m]$	$y [m]$	$z [m]$
left hand	[0.047, 0.210]	[0.148, 0.346]	[0.583, 0.951]
right hand	[0.004, 0.210]	[−0.335, −0.180]	[0.629, 0.952]
left grasp point	[0.157, 0.276]	[−0.035, 0.146]	[0.214, 0.349]
right grasp point	[0.178, 0.276]	[−0.189, −0.013]	[0.214, 0.349]

(b)

	$\psi_x [rad]$	$\theta_y [rad]$	$\varphi_z [rad]$
left hand	[−1.277, 0.131]	[−0.792, 0.543]	[0.089, 1.125]
right hand	[−2.820, −1.830]	[−0.861, 0.532]	[−1.419, −0.278]
left grasp point	[−1.570, −1.570]	[0.00, 0.00]	[−0.273, 0.523]
right grasp point	[−1.570, −1.570]	[0.00, 0.00]	[−0.273, 0.523]

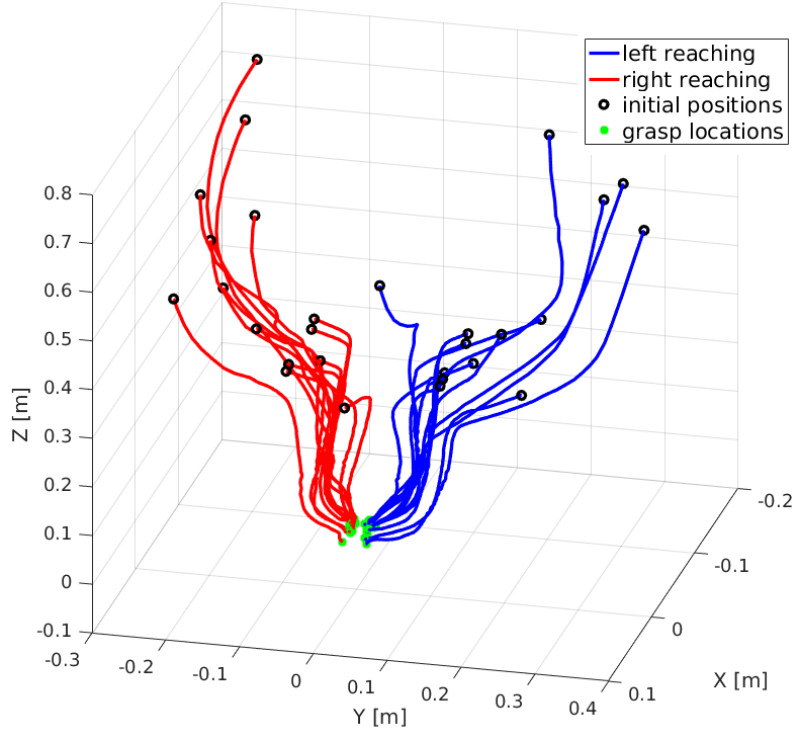


Figure 4.10: Example of 15 three-dimensional coordinated reach-to-grasp trajectories of the humanoid robot's hands starting from different initial poses.

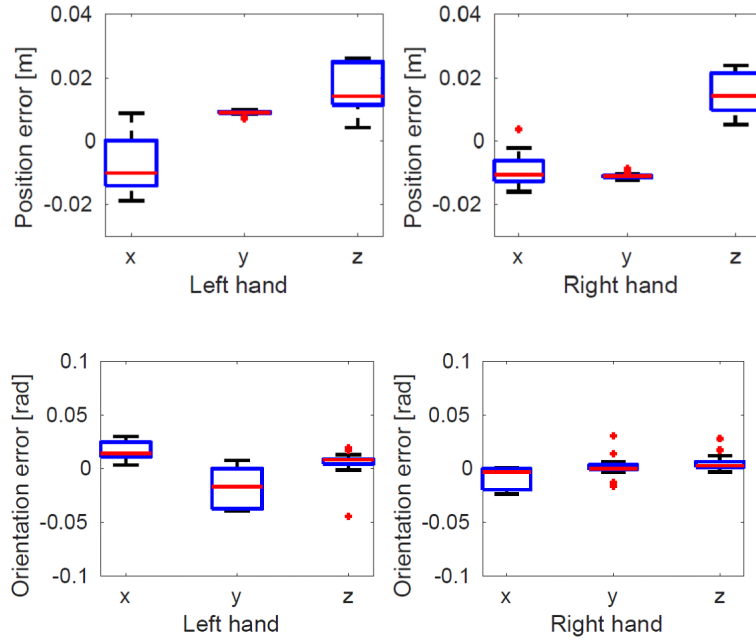


Figure 4.11: Boxplots of final reaching position errors along each 3D coordinate for the left and right hand of the humanoid robot when performing coordinated reach-to-grasp tasks from initial poses shown in Figure 4.10.

It can be noticed that the position errors in the y direction is relatively small compared to the x and z directions. This is because it corresponds to the squeezing direction, where the robot always ends in contact with the object due to the applied forces. For the x and z directions, the errors depend on their relative positions at the time the contact is assumed to have been established. In effect, a threshold of $0.015m$ was set to consider that the robot's end-effectors have reached their desired locations on the object.

4.4.2 COORDINATED REACHING AND GRASPING OF A MOVING OBJECT

We now evaluate the ability of our controller to generate coordinated motion for a moving target. The velocity of the object goes up to $0.2m/s$. When the object is outside of the hands reachable workspace, the robot's hands go to a resting pose. As soon as the object enters the reachable space of the hands, the latter start tracking their corresponding attractors on the object. As the robot uses its entire body to perform this task, the whole-body controller ensures that the robot remains stable throughout the task.

Snapshots of the robot during this task are given in Figure 4.12. Figures 4.13 shows poses errors and the grasping wrenches of the left hand during the task. Although not shown here, similar patterns are observable for the right hand. The controller generates no wrench until the contacts between the object and both hands are established (around $t = 112s$).

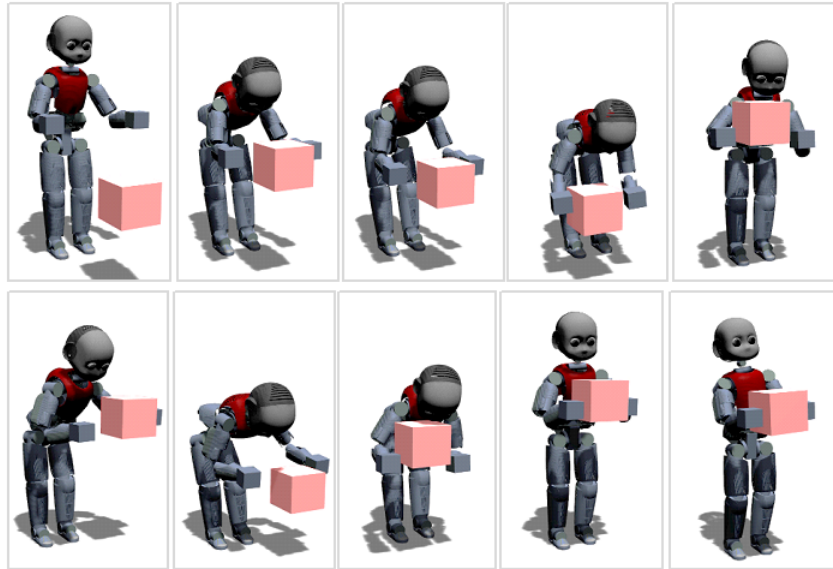


Figure 4.12: Illustration of a whole-body bimanual coordinated tracking and grasping of a moving object performed by a humanoid robot while keeping its balance.

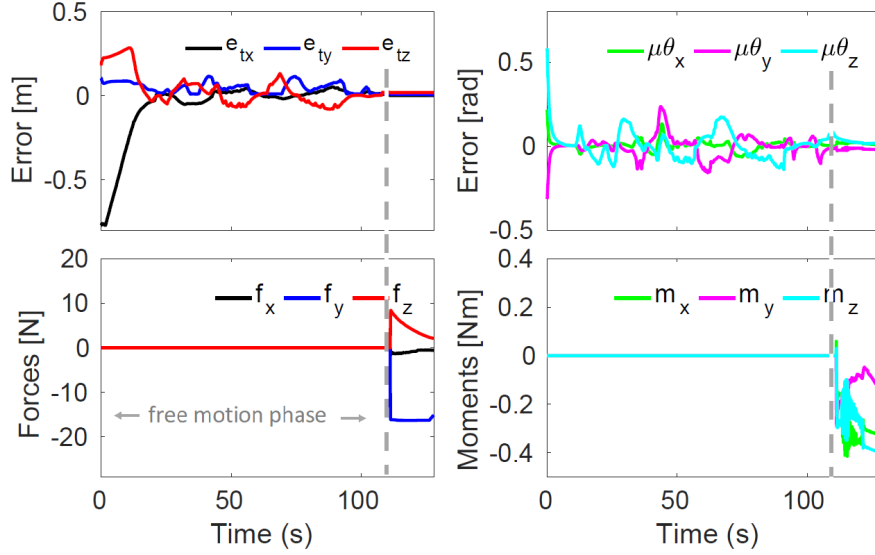


Figure 4.13: Poses errors and applied wrench for the left hand during the tracking and grasping of a moving object. The wrenches are zero throughout the tracking task (from $t = 0$ to $t = 112$). They appear only when the contacts are established between the hands on the object ($t = 112$)

4.4.3 WHOLE-BODY AND OBJECT-LEVEL COMPLIANT INTERACTION

We also tested some tasks that exploit the redundancy and compliance of the robot under the designed whole-body torque controller. For instance, Figure 4.14, shows a bimanual manipulation task where the robot is holding the object at a given position while balancing and squatting, whereas Figure 4.14(b) illustrates a case of interaction to external forces on the robot's body during a manipulation task. Here, the external force is applied on the hip of the robot, the latter complies while balancing and thanks to its redundancy, it maintains the object at the same location.

In Figure 4.15, we show an other interaction case where the external force is applied, this time, directly on the grasped object. The challenge for the robot is to keep its balance, maintain stable grasp of the object while being compliant. In this case, a desired attractor was defined for the object, \mathbf{f}_o^d . Thus, when the object is pushed away from its attractor, the object's effective wrench \mathbf{f}_o^d (see (4.3.9)) increases and the hands wrenches are adjusted accordingly to maintain the grasp.

The variations of wrenches during this task are shown in Figure 4.16. In the

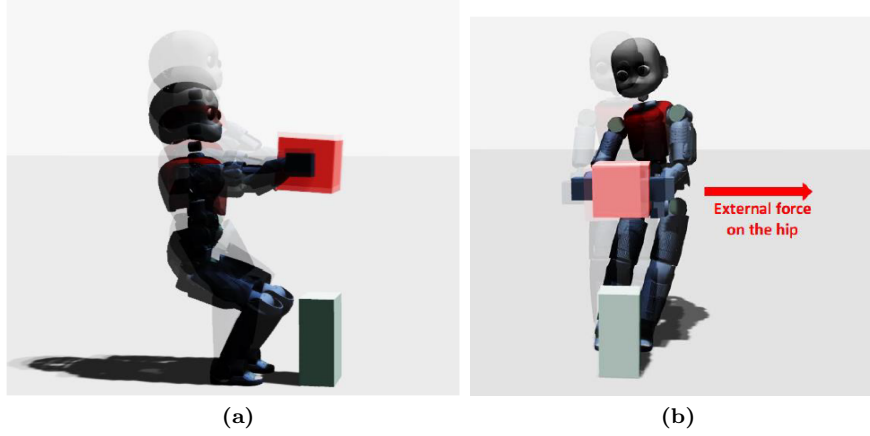


Figure 4.14: Simulations of bimanual manipulation scenarios that exploit the redundancy and the compliance of the humanoid robot. (a) maintaining desired object's location when squatting. (b) keeping balance and maintaining desired object's location despite external force acting on the robot's hip

first 20s, before the contacts are established, the wrenches are zero. Afterwards, they suddenly increase to grasp the object and lift it. Thereafter, they start decreasing as the object approach its attractor. Around $t = 50$ s an external force is applied on the object, one can see that the wrenches increase to maintain the grasp of the object. As mentioned before, the applied forces were saturated around 20N. When a stronger external force was applied around $t = 240$ s, the robot could not compensated for it (see green circle in Figure 4.16), the contacts with the object was broken and the wrenches returned to zero. However, the robot started automatically a reach-to-grasp phase and re-grasped the object 20s later.

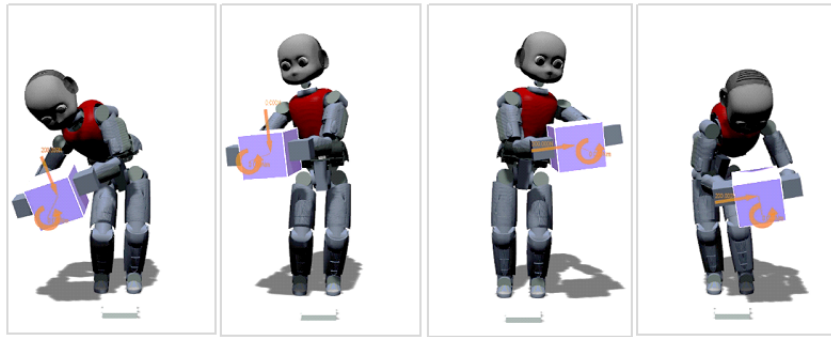


Figure 4.15: Bimanual grasp with compliance to external forces acting directly on the object

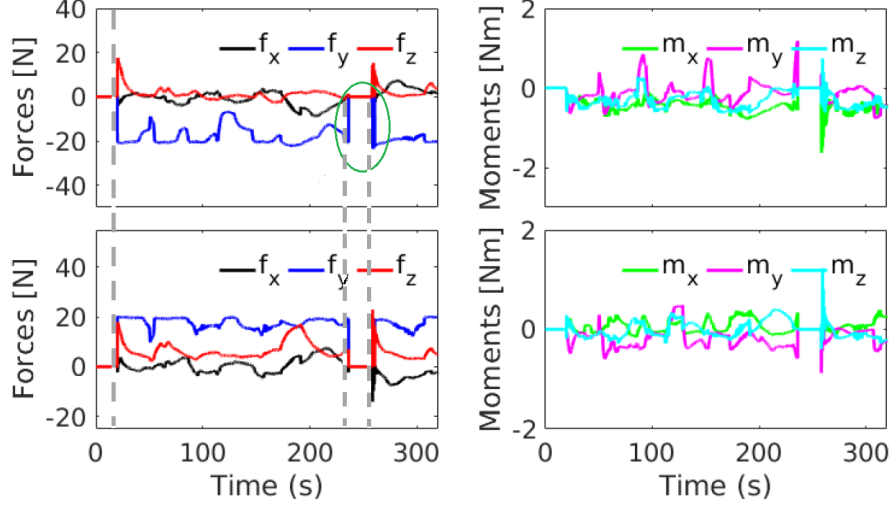


Figure 4.16: Variations of the grasping wrenches of the left and right hands during the interaction task with the object. The wrenches are updated on-line in response to external forces applied directly on the object. The abrupt drop of the wrenches to zero (the green circle region) around $t = 250s$ is due to contacts loss caused by a strong perturbation in the contact direction. It is then followed by a re-grasping of the object

4.4.4 REAL ROBOT BIMANUAL OBJECT REACHING AND GRABBING BASED ON WHOLE-BODY INVERSE KINEMATICS

Finally, to validate our bimanual coordination scheme on the real humanoid robot, we implemented our framework using whole-body inverse kinematics as our experimental platform had issues with its torque controller. Moreover, the joints of the hands are not controllable in torque. Thus, the motion was generated, as in simulation, using our proposed DS. However, the grasping forces were now applied through an impedance controller that exploits the relative distance between the two robot's hands. The balance task is ensured by the whole-body inverse kinematics-based controller.

Figure 4.17 provides snapshots illustrating a bimanual coordinated reaching, grabbing, and lifting up an object. The latter is equipped with markers that allow the estimation of its pose using an Optitrack vision system. The grasping points are defined with respect to the object and constitute the attractor for the dynamical system that generates the coordinated reaching motion. As can be observed, the humanoid robot successfully uses its whole body to bend and reach the object before grabbing and lifting it while remaining balanced. Although, we achieve the desired bimanual task, we lost, the whole-body compliance, as shown in the previous section. This is because of the stiff control in position, obtained from integrating the velocity generated by the inverse kinematics.

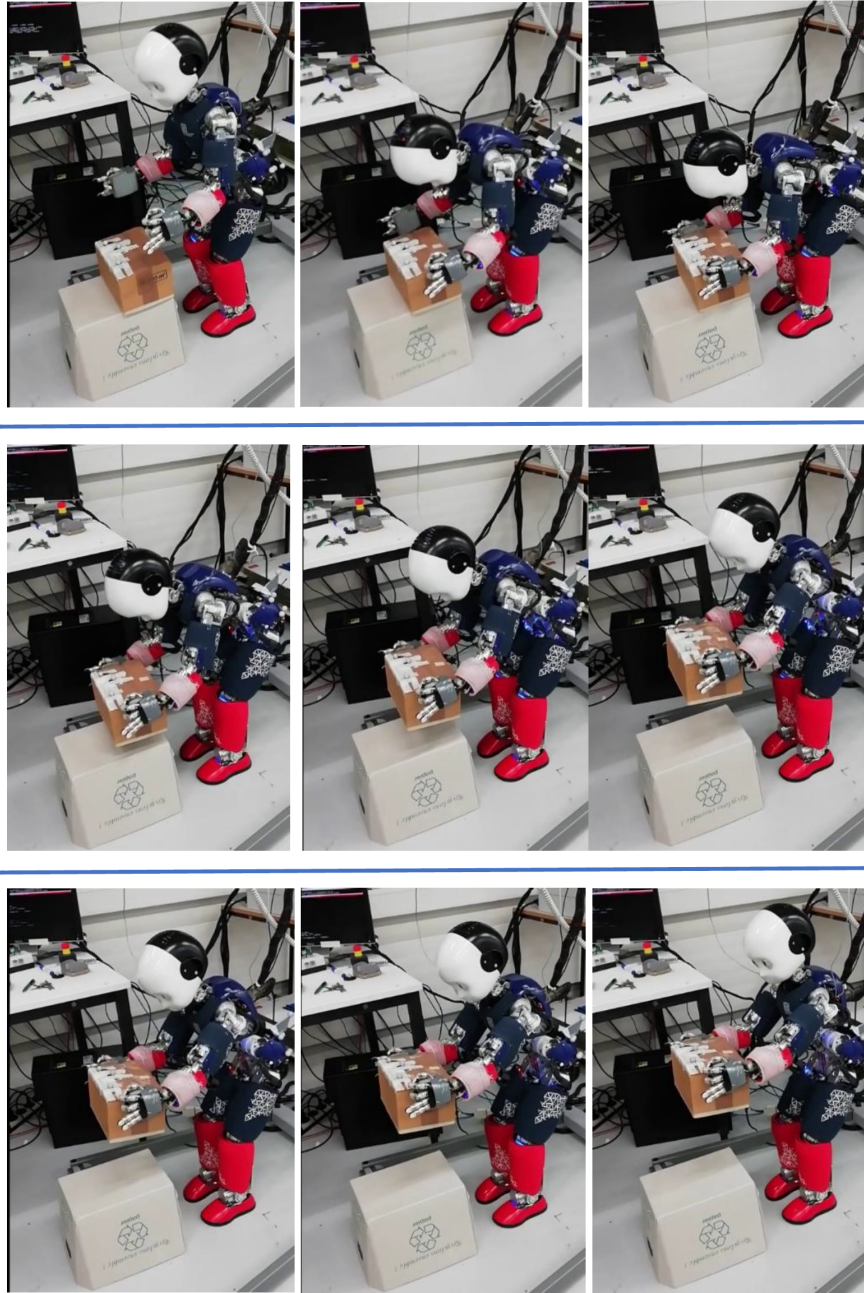


Figure 4.17: Snapshot of real humanoid robot iCub performing a bimanual coordinated reaching and grabbing of an object while balancing using whole-body inverse kinematic-based control

4.5 Conclusion

This Chapter proposed a combined dynamical systems QP-based algorithm to achieve robust reach-to-grasp and stable grasp and manipulation of an object by a humanoid robot. Integrated to a whole-body controller, the generation of coordinated motion and forces in non-contact and contact phases was theoretically demonstrated and validated in simulation. The results suggest that the proposed approach enables bimanual coordinated reaching and grasping of static as well as moving objects. As the proposed motion generation is based on time-invariant dynamical systems, it has inherited the robustness and the fast re-planning ability of the DS. Successful tracking and grabbing of moving objects while balancing were demonstrated. Moreover, with the online computation of constraint-consistent interaction wrenches using QP, the proposed approach showed its robustness to the change of contact area (surface, line, and point contacts). It also allows performing various manipulation tasks with direct interaction with objects while keeping the grasp and the balance of the robot.

However, the proposed approach has some limitations that need to be addressed. For instance, the desired motion of the DS was not well executed by the robot due to its slow dynamics. The reachable space of the bimanual system was not computed explicitly but was roughly approximated. One way to tackle this problem could be to learn it as in (Kim et al., 2014). Another problem that could cause some failures of the task is the collision of the hands with the object to grasp. This problem can be addressed by modulating the DS using, for instance, the approach proposed in (Huber et al., 2019b) to move around part of the object and reach the desired attractor. However, attention should be particularly paid to the fact that the attractor is located on one side of the “obstacle” which, in our case, is the object itself. Besides, the collision at the end-effectors, it was sometimes observed that the humanoid robot was moving in directions that could lead to joint limits or to unnatural configurations. This is mainly due to the fact that the proposed DS focuses only on the task space motion and it is agnostic of what happens at the joint level. Thus, designing a control scheme that could generate or induce some specific motion in the joint space could be a solution to this problem. Moreover, although validated in simulation and on a stiff-controlled robot, the effectiveness of the proposed framework with compliant bimanual manipulation would be demonstrated only once tested on a real humanoid robot with full torque control capabilities. In the next Chapter, we will implement our framework, this time with dynamic transitions, on a pair of torque-controlled robotic arms.

DUAL-ARM CONTROL FOR COORDINATED FAST GRABBING AND TOSSING OF AN OBJECT

This Chapter presents a novel unified coordination framework for bimanual dynamic manipulation that enables reaching, grabbing with impact and tossing of an object in one swipe. It models the dual-arm situation and presents our formulation using modulated dynamical systems to unify the manipulation phases and locally shape the generated motion to satisfy the impact and tossing tasks requirements. Then, it presents the QP-based strategy to generate the bimanual contact forces that stabilize the object. Finally, it presents experimental results on a pair of two KUKA LBR robots and shows quantitatively that our approach reduces the task duration and the energy expenditure when compared to the classical pick-and-place task.

Publication note: The material presented in this Chapter was adopted from:

- Bombile, M.B. and Billard, A., 2022. Dual-Arm Control for Coordinated Fast Grabbing and Tossing of an Object: Proposing a New Approach. IEEE Robotics & Automation Magazine.

Source codes:

- Dual arm controller
https://github.com/epfl-lasa/iam_dual_arm_control

Supplementary Video:

- Dual-arm control for coordinated fast grabbing and tossing of an object
<https://youtu.be/CeLoqXdPI0U>

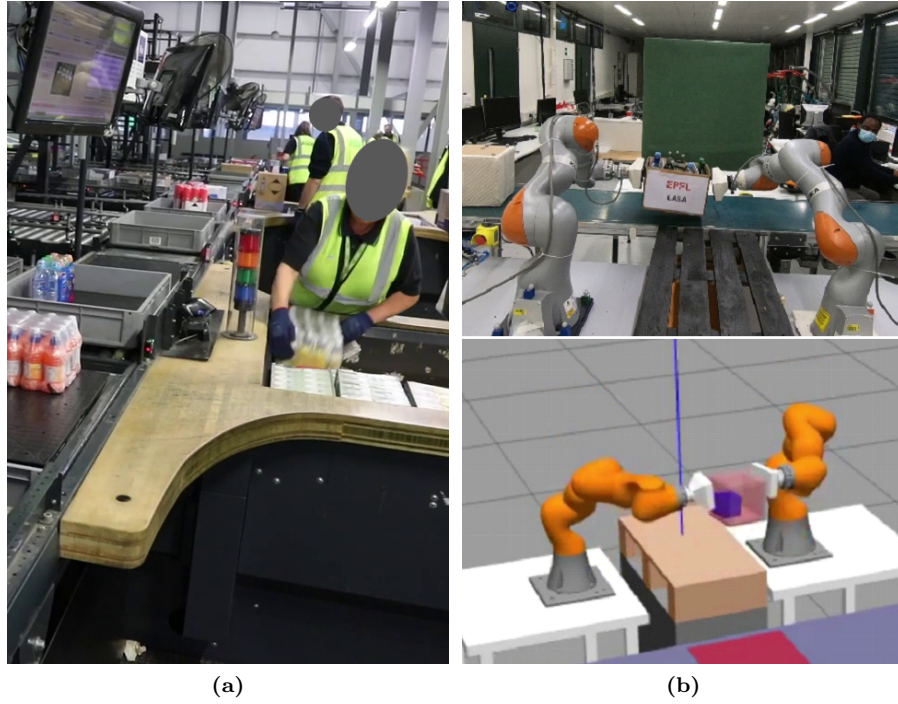


Figure 5.1: Illustration of a dual-arm manual and robotic pick-and-place operations. (a) human dual-arm grabbing and placing objects in a palletizing task within Vanderlande facility (photo courtesy of Vanderlande). (b) pair of two real and simulated real KUKA LBR IIWA robots grabbing an open box (top) and an object containing a small moving object inside (bottom).

5.1 Introduction

SWIFT robot manipulation of objects in unstructured and dynamic environments is crucial for the industry. This Chapter considers the problem of dynamic grabbing and releasing an object in one swipe with a dual-arm robotic system. The desired manipulation task is motivated by the need to pick and position faster objects in a depalletizing context, see Figure 5.1. Humans usually perform such repetitive and physically demanding work for lack of similarly fast, precise, and robust bimanual robot systems. As outlined in Section 2.4.3, the bimanual tasks envisioned here extend the complexity of the control problem as it requires, in addition to generating the desired impact and tossing motion, to enforce the coordination of the two robotic arms throughout the task. For instance, a poorly coordinated system, where one arm reaches the object before the other, would lead to uncontrolled impact.

Achieving desired grabbing with impact and tossing of objects is challenging as these tasks require that desired transitory states expressed in terms of desired position and velocity be simultaneously satisfied (during contact or at the release time, respectively). Moreover, the motion should be robust throughout the task from grabbing with impact to release be it by placing, handing over, or tossing

the object. Despite the ability of state-of-the-art approaches to generate safe impacts for the robot, works dealing with bimanual grabbing with impacts have only been limited to this task. They did not consider combining it with other bimanual manipulation tasks such as placing or throwing objects. Also, works that considered throwing were not confronted with the problem of coordination since they used special robotic devices, single-arm robots, or parallel robots.

In this Chapter, in addition to a dynamic grab, we also consider a dynamic release by throwing the objects with a dual-arm robotic system. Controlling robustly such coordinated tasks of multi-arm systems opens the door to a variety of applications. Besides depalletizing, this could include manipulations that are too complex or heavy for a single robot and require two or more robotic arms. Some applications could be fast picking up of open trays or cases, fast picking up of luggage from airports' conveyor belts, etc. Unlike planning-based methods which are less reactive and prone to spatial and temporal perturbations, we adopt a motion generation approach based on autonomous DS for their fast and time-independent re-planning abilities and their robustness to perturbations. More specifically, we use modulated DS to allow local shaping of the motion flow and thus achieve the desired objectives of impact and tossing tasks. We formulate the motion coordination problem using the *extended cooperative task space* (ECTS) representation (Park and Lee, 2015). To stabilize the grasp and achieve the desired bimanual manipulation tasks, as in Chapter 4, interaction wrenches consistent with contact constraints are generated online using QP.

5.2 Problem statement

To control the dual-arm system, we assume that each robot is equipped with torque controller that can compute the joint torques for tasks specified in terms of desired task-space motion and forces. We consider, for instance, the passivity-inspired impedance-based controller (Kronander and Billard, 2016) with the gravity wrench already compensated for¹. Hence, each robot's joint torques to achieve a desired task-space motion and wrench can be computed as

$$\mathbf{\Gamma}_h = \mathbf{\Gamma}_{h\dot{\mathbf{x}}} + \mathbf{\Gamma}_{h\omega} \text{ with } \begin{cases} \mathbf{\Gamma}_{h\dot{\mathbf{x}}} &= \mathbf{J}_{h\dot{\mathbf{x}}}^\top(\mathbf{q}) [\mathbf{D}_h(\dot{\mathbf{x}}_h^d)(\dot{\mathbf{x}}_h - \dot{\mathbf{x}}_h^d) + \mathbf{f}_h^d] \\ \mathbf{\Gamma}_{h\omega} &= \mathbf{J}_{h\omega}^\top(\mathbf{q}) [\mathbf{D}_h(\omega_h^d)(\omega_h - \omega_h^d) + \boldsymbol{\tau}_h^d] \end{cases} \quad (5.2.1)$$

where $\mathbf{\Gamma}_h = \mathbf{\Gamma}_{h\dot{\mathbf{x}}} + \mathbf{\Gamma}_{h\omega}$ is the vector of joint torques of the h^{th} robot, with ($h \equiv l, r$) referring to the left or right robot of the dual-arm system. $\mathbf{\Gamma}_{h\dot{\mathbf{x}}}$ and $\mathbf{\Gamma}_{h\omega}$ denote the torque component due to the task-space linear motion or force and angular motion or moments, respectively. Similarly, $\mathbf{J}_{h\dot{\mathbf{x}}}(\mathbf{q})$ and $\mathbf{J}_{h\omega}(\mathbf{q})$ are Jacobian matrices associated with the linear and angular velocities $\dot{\mathbf{x}}_h \in \mathbb{R}^3$ and $\omega_h \in \mathbb{R}^3$, respectively. $\dot{\mathbf{x}}_h^d \in \mathbb{R}^3$ and $\omega_h^d \in \mathbb{R}^3$ are desired linear and angular

¹For the dual KUKA arm platform at our disposal, we can use a built-in gravity compensation module. In the absence of such a module, one should estimate the inertia of the links and subtract this.

velocities, respectively. $\mathbf{f}_h^d \in \mathbb{R}^3$ and $\boldsymbol{\tau}_h^d \in \mathbb{R}^3$ denote the force and moment components of the desired task wrench, respectively. $\mathbf{D}_h(\dot{\mathbf{x}}_h^d) \in \mathbb{R}^{3 \times 3}$ is a state-dependent damping matrix defined as $\mathbf{D}_h(\dot{\mathbf{x}}_h^d) \triangleq \mathbf{Q}_h(\dot{\mathbf{x}}_h^d) \boldsymbol{\Lambda}_{h \times} \mathbf{Q}_{h_i}^\top(\dot{\mathbf{x}}_h^d)$, where $\mathbf{Q}_h(\dot{\mathbf{x}}_h^d)$ is an orthogonal matrix whose first eigenvector is aligned with $\dot{\mathbf{x}}_h^d$. $\boldsymbol{\Lambda}_{h \times}$ is a negative definite diagonal gain matrix (refer to (Kronander and Billard, 2016) for more details). $\mathbf{D}_h(\boldsymbol{\omega}_h^d)$ follows similarly definition to $\mathbf{D}_h(\dot{\mathbf{x}}_h^d)$ but for $\boldsymbol{\omega}_h^d$.

As in the previous Chapter 4, our problem is to find ways to generate the desired task-space motion $\{\dot{\mathbf{x}}_h^d, \boldsymbol{\omega}_h^d\}$ and wrench $\{\mathbf{f}_h^d, \boldsymbol{\tau}_h^d\}$ that will produce coordinated swift grabbing with impact and stable manipulation of the object by the dual-arm robot.

We also make the following assumptions regarding the object and contacts:

- A5.1:** the object is rigid with known shape and dimensions, and known inertia properties (mass and moments of inertia).
- A5.2:** the contacts between the object and end-effectors are unilateral and frictional. They are geometrically modeled by contact surfaces (convex hull of contact points).
- A5.3:** the grasping points are known and correspond to the centers of the contact surfaces, they have coordinate frames attached to them to describe their relative positions and orientations.
- A5.4:** The contact surfaces comply with a Coulomb friction model (Murray et al., 1994) and their associated friction coefficients are known (at least their minimum values).

Moreover, the scope of this Chapter being essentially motion and force generation, we leave aside the impact dynamics and assume that the associated states' jumps remain within the robots' limits. The interested reader is referred to (Rijnen et al., 2019) for the control of impact with states jump mitigation, and (Wang et al., 2020) or (Dehio and Kheddar, 2021; Dehio et al., 2022) for explicit enforcement of hardware limits during impact generation.

5.3 Proposed Approach

To induce the desired motion on the object when fast picking and when tossing requires that the two robots' arms adopt the required velocity just prior to impact (for fast picking) and prior to release (when tossing). To obtain this behavior, the robot arms must transit, at contact and release time, through desired states expressed in both position and velocity simultaneously. Unlike attractors, these transitory states are not equilibrium points and therefore the robotic system can only transit through such states. Thus, to realize the desired

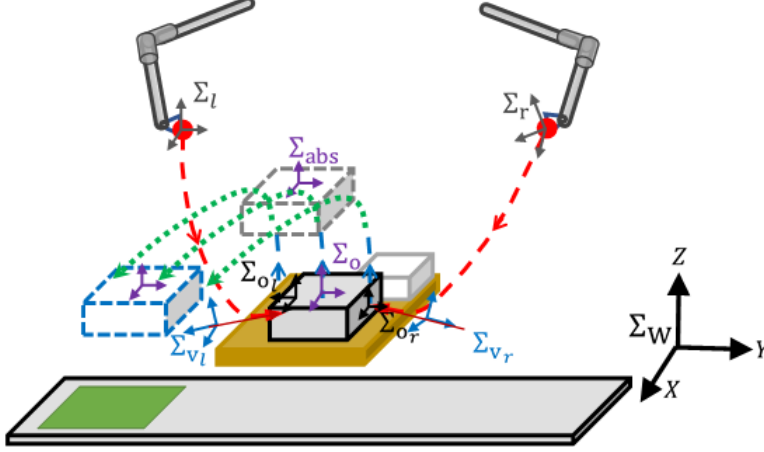


Figure 5.2: Schematics illustration of the considered dual-arm task, which can be seen as succession of two main phases: free motion phase when reaching (red dashed line), and constrained motion phase when in contact and executing the placing or tossing motion (blue and green dashed lines)). Σ_W and Σ_o are the world and the object frames. Σ_l and Σ_r denote respectively the frames of the left and right end-effectors, while Σ_{o_l} and Σ_{o_r} denote respectively their desired grasping configuration on the object side. Σ_i and Σ_{o_i} are respectively the i^{th} end-effector's and object's grasp configuration frames, while Σ_{v_i} denotes the i^{th} frame of an auxiliary attractor that shapes the trajectory for impact, with index ($i \equiv \text{left, right}$).

task in a robust way, we proposed an approach based on modulated dynamical systems, where state-dependent functions shape locally the generated motion of the robot - prior to contact or release of the object - such that the motion aligns first with the desired velocity while moving towards the desired contact or release position. Therefore, for a dual-arm system which requires coordination, to realize fast grabbing and afterward a tossing task, we formulate at the position level² the following modulated dynamical system (MDS).

$$\dot{\mathbf{x}} = M(\mathbf{x})f_n(\mathbf{x}) + f_g(\mathbf{x}) \quad (5.3.1)$$

where $\mathbf{x} = \begin{bmatrix} \mathbf{x}^L \\ \mathbf{x}^R \end{bmatrix} \in \mathbb{R}^6$ is the state vector of the DS with \mathbf{x}^L and \mathbf{x}^R representing respectively the position of the left and right robot of the dual-arm system. $f_n(\mathbf{x}) \in \mathbb{R}^6$ is the nominal DS that generates the coordinated motion towards transitory attractors located in the vicinity of the desired positions. $f_g(\mathbf{x})$ represents the equivalent grasping force in the motion space, whereas $M(\mathbf{x}) \in \mathbb{R}^{6 \times 6}$ is the state-dependent modulation matrix that shapes locally the motion generated by $f_n(\mathbf{x})$. It is defined as

$$M(\mathbf{x}) = E(\mathbf{x})\Lambda(\mathbf{x})E^\top(\mathbf{x}) \in \mathbb{R}^{6 \times 6} \quad (5.3.2)$$

where $E(\mathbf{x}) \in \mathbb{R}^{6 \times 6}$ and $\Lambda(\mathbf{x}) \in \mathbb{R}^{6 \times 6}$ are block-diagonal matrices respectively

²The control of orientation is described in appendix D.1

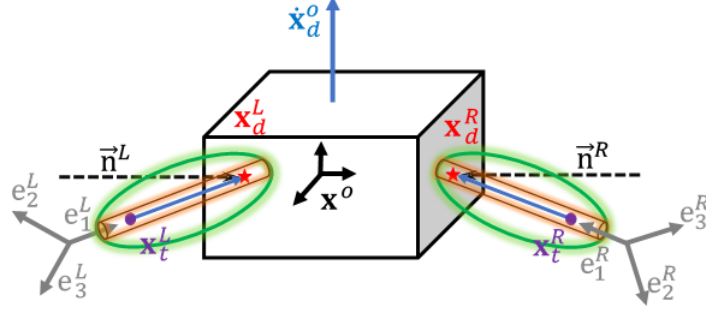


Figure 5.3: Geometric representation of orthonormal basis $E^L(x)$, $E^R(x)$ and illustration of modulation region (green ellipsoid) within which the dark-red cylindrical region represents the activation of the normal distance to the vector $\mathbf{x}_d^h - \mathbf{x}_t^h$

of state-dependent orthonormal basis and gains for the left and right robotic arm. They are respectively defined as

$$E(\mathbf{x}) = \text{diag}\{E^L(\mathbf{x}), E^R(\mathbf{x})\}$$

$$\Lambda(\mathbf{x}) = \text{diag}\{\Lambda^L(\mathbf{x}), \Lambda^R(\mathbf{x})\}$$

In $E(\mathbf{x})$, each basis $E^h(\mathbf{x}) = [\mathbf{e}_1^h \ \mathbf{e}_2^h \ \mathbf{e}_3^h] \in \mathbb{R}^{3 \times 3}$ with $h = \{L, R\}$ is designed such that its first vector \mathbf{e}_1^h is aligned with the intended impact direction at contact.³ That is

$$\mathbf{e}_1^h = \frac{\dot{\mathbf{x}}_d^h}{\|\dot{\mathbf{x}}_d^h\|} = \frac{\mathbf{x}_d^h - \mathbf{x}_t^h}{\|\mathbf{x}_d^h - \mathbf{x}_t^h\|}$$

where $[\mathbf{x}_d^h \ \dot{\mathbf{x}}_d^h]$ denotes the desired impact or tossing state of the h^{th} robot, and \mathbf{x}_t^h is the transitory attractor defined at a distance ρ of \mathbf{x}_d^h , such that it is given by $\mathbf{x}_t^h = \mathbf{x}_d^h + E^h(\mathbf{x})[-\rho \ 0 \ 0]^T$. Thus, as illustrated in Figure 5.3 for the case of grabbing with impact, each robot is driven first towards a transitory attractor \mathbf{x}_t^h before being moved with the appropriate orientation along \mathbf{e}_1^h towards \mathbf{x}_d^h , the real attractor. In $\Lambda(\mathbf{x})$, each sub-matrix $\Lambda^h(\mathbf{x}) \in \mathbb{R}^{3 \times 3}$ has entries $\bar{\lambda}_{ij}^h(\mathbf{x})$ defined as

$$\bar{\lambda}_{ij}^h(\mathbf{x}) = \begin{cases} \alpha(\mathbf{x})\lambda_{ij}^h(\mathbf{x}) + (1 - \alpha(\mathbf{x})) & \text{if } i = j \\ \alpha(\mathbf{x})\lambda_{ij}^h(\mathbf{x}) & \text{if } i \neq j \end{cases} \quad (5.3.3)$$

where $\lambda_{ij}^h(\mathbf{x}) \in \mathbb{R}^1$ represent state-dependent scalar terms defined in section 5.4.1, $\alpha(\mathbf{x}) \in [0, 1]$ activates the modulation when the robots are in the vicinity of their desired attractors.

The modulation is active in a region defined by an ellipsoid along the vector $\mathbf{x}_d^h - \mathbf{x}_t^h$ as illustrated in Figure 5.3. To characterize the modulation, we define three activation parameters namely: δ_{radial} , δ_{normal} and $\delta_{tangent}$ which represent distances with their origin at \mathbf{x}_t^h in the basis $E^h(\mathbf{x})$. These distances are measured respectively in $3D$, in $2D$ normal to \mathbf{e}_1^h , and in $1D$ along \mathbf{e}_1^h . Accord-

³The direction of impact is not limited to be normal to the contact surface, but can also have other orientation.

ingly, we define associated activation functions $\phi_i(\mathbf{x}^h) \in [0, 1]$ with $i=\{radial, normal, tangent\}$ such that

$$\phi_i(\mathbf{x}^h) = \frac{1}{1 + e^{-a_i(\delta_i - \Gamma_i(\mathbf{x}^h))}} \quad (5.3.4)$$

where $\Gamma_i^h(\mathbf{x}^h)$ represents state-dependent distances of the h^{th} robot, whose expressions are as follow $\Gamma_i^h(\mathbf{x}^h) = [(\mathbf{x}^h - \mathbf{x}_t^h)^\top E^h(\mathbf{x}^h) \Sigma_i (E^h(\mathbf{x}^h))^\top (\mathbf{x}^h - \mathbf{x}_t^h)]^{\frac{1}{2}}$. $\Sigma_i \in \mathbb{R}^{3 \times 3}$ are diagonal matrices that select the considered directions of $E^h(\mathbf{x})$. The elements of Σ_i are mainly 0, but 1 at index(es) of the desired direction(s). Hence, $\alpha(\mathbf{x})$ in Eq. (5.3.3), is designed such that $\alpha(\mathbf{x}) = \frac{1}{2} \sum_{h=1}^2 \phi_{radial}(\mathbf{x}^h)$. The behavior of the DS and the activation functions are shown in Figure 5.4.

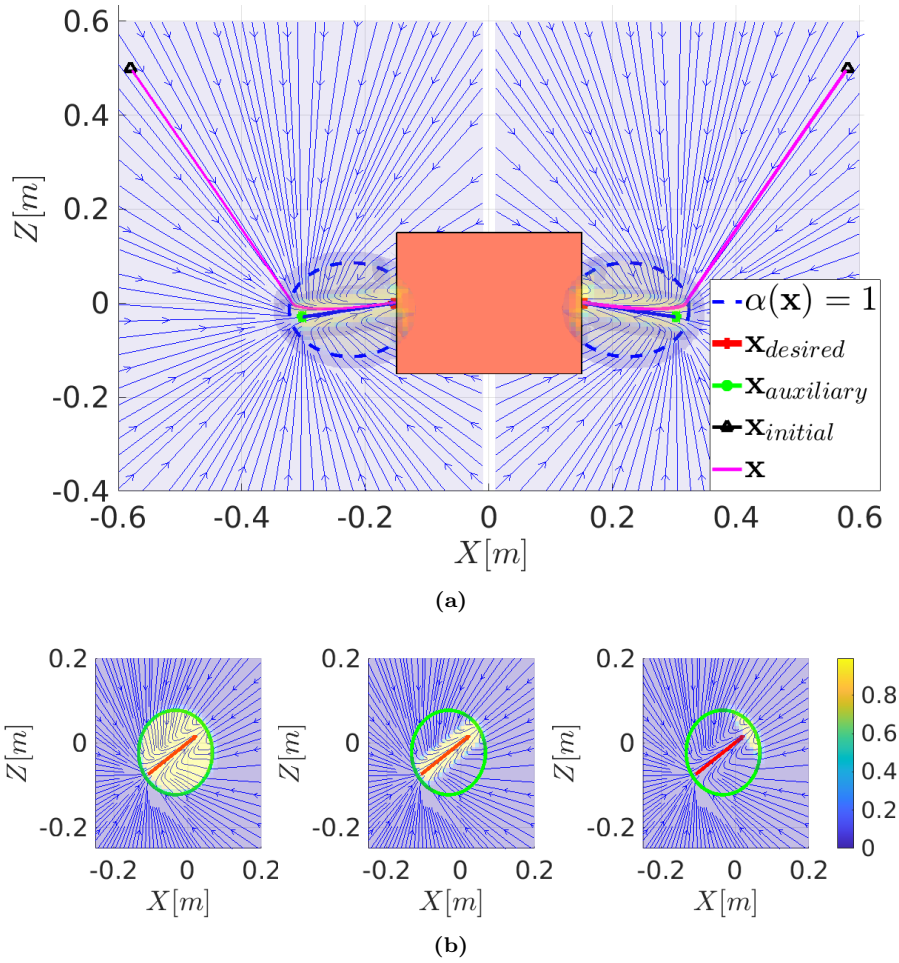


Figure 5.4: Illustration of motion flow generated by the DS outside and within the modulated region. (a) the motion of each robot is shaped within the modulation region (thick dotted blue line) such that it passes through the desired transitory state (here an impact state) with the desired position represented by the red dot and the direction of the desired velocity represented by the blue arrow. (b) representation of spatial activation of the modulation terms, respectively $\phi_{radial}(\mathbf{x})$, $\phi_{normal}(\mathbf{x})$, and $\phi_{tangent}(\mathbf{x})$, which become equal to 1 in the yellow region.

5.4 Modulation-based Coordinated Control

The motion coordination of the dual-arm system, in this thesis, exploits the cooperative task space representation (Park and Lee, 2015), which relates the states of each robot to the cooperative coordinates formed by the absolute and relative states of the dual-arm system. The coordination is achieved by controlling the two robots cooperative coordinates and mapping the resulting motion to each robot.

Thus, assuming that the nominal DS $f_n(\mathbf{x})$ is linear, the coordinated motion that it generates can be written as:

$$f_n(\mathbf{x}) = \dot{\mathbf{x}} = \underbrace{T_b^{-1} A T_b}_{A'} (\mathbf{x} - \mathbf{x}_*) \quad (5.4.1)$$

where $T_b = \begin{bmatrix} \frac{1}{2}\mathbf{I}_3 & \frac{1}{2}\mathbf{I}_3 \\ -\mathbf{I}_3 & \mathbf{I}_3 \end{bmatrix} \in \mathbb{R}^{6 \times 6}$ is a matrix that maps the two robot positions (\mathbf{x}^L and \mathbf{x}^R) to the absolute position $\mathbf{x}^{abs} \in \mathbb{R}^3$ and relative position $\mathbf{x}^{rel} \in \mathbb{R}^3$ of the dual-arm system, such that $\begin{bmatrix} \mathbf{x}^{abs} \\ \mathbf{x}^{rel} \end{bmatrix} = T_b \begin{bmatrix} \mathbf{x}^L \\ \mathbf{x}^R \end{bmatrix}$ and where \mathbf{I}_3 is a 3×3 unit matrix. In Eq. (5.4.1), $A \in \mathbb{R}^{6 \times 6}$ denotes the dynamics or gain matrix, which is negative definite ($A < 0$) to ensure stability and convergence to a given attractor \mathbf{x}_* .

The coordination is thus achieved by controlling the dynamics of \mathbf{x}^{abs} and \mathbf{x}^{rel} , which amounts to control respectively the two robots joint motion and their relative displacement and thereby their synchronization. The resulting motion of \mathbf{x}^{abs} and \mathbf{x}^{rel} is then mapped through T_b^{-1} to each robot's motion.

The modulation shapes the behavior of the nominal DS in the region where it is active. This shaping must preserve the DS stability and convergence properties of the nominal dynamical system $f_n(\mathbf{x})$ towards its equilibrium points \mathbf{x}_* .

5.4.1 STABILITY AND CONVERGENCE TO ATTRACTORS

Based on the previous work (Mirrazavi Salehian and Billard, 2018) on single robot, we can state the following proposition for dual-arm system:

PROPOSITION 1

For any given state $\{\mathbf{x} \in \mathbb{R}^3 \mid \alpha(\mathbf{x}) = 1, f_n^h(\mathbf{x}) \neq 0\}$ setting the state dependent coefficients of the modulation matrix $\Lambda^h(\mathbf{x})$ as

$$\lambda_{ij}^h(\mathbf{x}) = (\mathbf{e}_i^h)^\top f_{m_i}^h(\mathbf{x}) \frac{f_n^h(\mathbf{x})^\top \mathbf{e}_j^h}{f_n^h(\mathbf{x})^\top f_n^h(\mathbf{x})}, \quad (5.4.2)$$

the motion generated by Eq. (5.3.1) will be governed by the dynamical system $f_{m_i}^h(\mathbf{x})$. Moreover, if $f_{m_i}^h(\mathbf{x})$ is a stable linear or linear parameters varying (LPV) DS, for instance, of the form of Eq. (5.4.1), the state \mathbf{x} will asymptotically reach

its attractor \mathbf{x}_* while maintaining the coordination between robots of the dual-arm system. That is $\lim_{t \rightarrow \infty} \|\mathbf{x} - \mathbf{x}_*\| = 0$.

Proof: See Appendix D.3.

5.4.2 GENERATION OF IMPACT VELOCITY

To generate desired grabbing impact velocities with the dual-arm system, we introduce the following proposition:

PROPOSITION 2

For any given state $\{\mathbf{x} \in \mathbb{R}^3 | f_r^h(\mathbf{x}) \neq 0\}$ setting the attractor \mathbf{x}_* as follows

$$\mathbf{x}_* = \begin{cases} (1 - \gamma(\mathbf{x}))\mathbf{x}_t + \gamma(\mathbf{x})(\mathbf{x} - A'^{-1}\dot{\mathbf{x}}_d) & \text{if } e_{i=1}^h \\ \mathbf{x}_t & \text{if } e_{i \neq 1}^h \end{cases} \quad (5.4.3)$$

with $\gamma(\mathbf{x}) \triangleq \phi_{radial}(\mathbf{x})\phi_{normal}(\mathbf{x})$, then for any state, the modulated DS (5.3.1) with Eq. (5.4.1) will generate motion first towards \mathbf{x}_t and then, when $\gamma(\mathbf{x}) = 1$, it will generate desired velocity $\dot{\mathbf{x}}_d = \begin{bmatrix} \dot{\mathbf{x}}_d^L \\ \dot{\mathbf{x}}_d^R \end{bmatrix} \in \mathbb{R}^6$ along the vectors e_1^L and e_1^R while maintaining attraction towards them in their respective orthogonal directions $((e_2^L, e_3^L)$ and $(e_2^R, e_3^R))$.

Proof: See Appendix D.4.

5.4.3 MANIPULATION TASK

To generate desired motion of an object carried by a dual-arm system using the proposed DS in Eq. (5.3.1), we introduce the following proposition:

PROPOSITION 3

For an object grasped by a dual-arm system and whose position is given by \mathbf{x}^o and desired attractor is given \mathbf{x}_d^o , for any state $\{\mathbf{x} \in \mathbb{R}^3 | \kappa(\mathbf{x}) = 1, f_r^h(\mathbf{x}) \neq 0\}$ setting the attractor \mathbf{x}_* of the proposed modulated DS in Eq. (5.3.1) as

$$\mathbf{x}_* = \kappa(\mathbf{x})T_b^{-1} \begin{bmatrix} \mathbf{x}_d^o + (\mathbf{x}^{abs} - \mathbf{x}^o) \\ \mathbf{x}_o^R - \mathbf{x}_o^L \end{bmatrix} \quad (5.4.4)$$

where $(\mathbf{x}^{abs} - \mathbf{x}^o)$ denotes the offset between the end-effector absolute position and the object's origin, and where \mathbf{x}_o^L and \mathbf{x}_o^R denote respectively the positions of the grasping points of left and right robot, the proposed modulated DS will generate coordinated robots motion that will make the grasped object's position to asymptotically converge to \mathbf{x}_d^o .

The scalar function $\kappa(\mathbf{x}) \in [0, 1]$ indicates the contact and it is defined as $\kappa(\mathbf{x}) \triangleq \gamma(\mathbf{x})\phi_{tangent}(\mathbf{x})$.

Proof: See Appendix D.5.

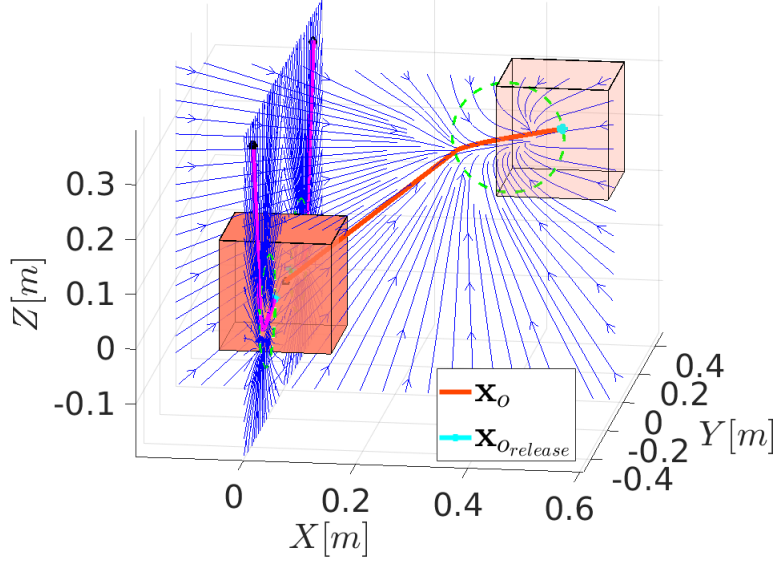


Figure 5.5: Illustration of object's motion flow generated by the DS once the object is grabbed and carried by the dual-arm system. The boundaries of the modulation region is represented by the thick green dashed line.

5.4.4 TOSSING TASK

The tossing task is a form of manipulation task where the grabbed object needs to be released at a desired position simultaneously with a desired velocity. Thus, to perform a tossing task of an object with a dual-arm system using the DS (5.3.1), we propose to generate the corresponding desired coordinated motion by setting the desired absolute and relative velocities ($\dot{\mathbf{x}}_d^{abs}$ and $\dot{\mathbf{x}}_d^{rel}$) as follows

$$\dot{\mathbf{x}}_d^{abs} = M^o(\mathbf{x}^o)f_o(\mathbf{x}^o) \text{ and } \dot{\mathbf{x}}_d^{rel} = -[\mathbf{x}_o^R - \mathbf{x}_o^L] \times \boldsymbol{\omega}^o \quad (5.4.5)$$

where similarly to Eq. (5.3.1), $f_o(\mathbf{x}^o) \in \mathbb{R}^3$ denotes the nominal DS of the object and $M^o(\mathbf{x}^o) = E^o(\mathbf{x}^o)\Lambda^o(\mathbf{x}^o)(E^o(\mathbf{x}^o))^\top \in \mathbb{R}^{3 \times 3}$ is the associated modulation matrix with $E^o(\mathbf{x}) \in \mathbb{R}^{3 \times 3}$ an orthonormal basis. To meet the tossing motion constraints, $E^o(\mathbf{x}^o)$ is defined with its origin at the desired release position \mathbf{x}_r^o and with its first vector \mathbf{e}_1^o aligned with the desired release velocity ($\mathbf{e}_1^o = \frac{\dot{\mathbf{x}}_r^o}{\|\dot{\mathbf{x}}_r^o\|}$). $\boldsymbol{\omega}^o$ is the angular velocity of the object. In pure translation ($\boldsymbol{\omega}^o = 0$), $\dot{\mathbf{x}}_d^{rel}$ is set to 0.

Now, the convergence to the release position with the desired release velocity is a corollary of *Proposition 1* and *Proposition 2* applied to the object state, and with

$$\mathbf{x}_*^o = \begin{cases} (1 - \gamma(\mathbf{x}^o))\mathbf{x}_t^o + \gamma(\mathbf{x}^o)(\mathbf{x}^o - A'^{-1}\dot{\mathbf{x}}_r^o) & \text{if } \mathbf{e}_{i=1}^o \\ \mathbf{x}_t^o & \text{if } \mathbf{e}_{i \neq 1}^o \end{cases} \quad (5.4.6)$$

where $\dot{\mathbf{x}}_r^o$ denotes the desired release velocity and $\mathbf{x}_t^o \in \mathbb{R}^3$ and $\gamma(\mathbf{x}^o)$ are defined as in (5.4.3), but this time for the object.

5.4.5 QP-BASED CONTACT FORCE GENERATION

When the robots end-effectors have reached the object, the grasp quality depends on the induced internal wrenches. The contacts being unilateral with limited friction, insufficient or excessive wrenches might result in contact slippage or damage of the object and/or end-effectors. Under the assumption that the object's mass and the friction coefficient (at least its minimum value) are known, we propose to generate the contact wrenches, \mathbf{f}_e^d , satisfying the desired task and the contact constraints through the following optimization

$$\begin{aligned} \mathbf{f}_e^d = \operatorname{argmin} \quad & \|\mathbf{G}_o \mathbf{f}_e - \mathbf{f}_o^d\|_{\mathbf{w}_o}^2 + \|\mathbf{f}_e - \mathbf{f}_e^*\|_{\mathbf{w}_e}^2 \\ \text{s.t.} \quad & \mathbf{C}_f \mathbf{f}_e \leq \mathbf{d}_f \\ & \mathbf{Q}_{eo} \mathbf{f}_e = \mathbf{0} \end{aligned} \quad (5.4.7)$$

where $\mathbf{f}_o^d \in \mathbb{R}^6$ is the effective wrench required to perform the desired manipulation task. \mathbf{w}_o and \mathbf{w}_e are weight matrices. The constraint $\mathbf{C}_f \mathbf{f}_e \leq \mathbf{d}_f$ encapsulates the friction cone and center of pressure constraints, whereas the constraints $\mathbf{Q}_{eo} \mathbf{f}_e = \mathbf{0}$ represents the complementary condition between the force components normal to the contact and the normal distance to the contact.

To find the object's effective wrench \mathbf{f}_o^d required in Eq. (5.4.7), if, for instance, the desired task is encoded as $\dot{\mathbf{x}}_o^d = \mathbf{f}(\mathbf{x}_o)$, one can use a passive DS approach (Kronander and Billard, 2016) to compute as ⁴ $\mathbf{f}_o^d = -\mathbf{D}_o(\mathbf{x}_o)(\dot{\mathbf{x}}_o - \mathbf{f}(\mathbf{x}_o)) + \mathbf{b}_o$, where $\mathbf{D}_o(\mathbf{x}_o) \in \mathbb{R}^{6 \times 6}$ denotes a state varying damping matrix and \mathbf{b}_o is the object's Coriolis and gravity forces.

The QP in Eq. (5.4.7) is based on an object-centered approach and performs nothing but a distribution of \mathbf{f}_o^d between the two robot's hands. It seeks a distribution that will yield minimum internal wrenches and satisfy the contact constraints.

5.5 Empirical Validation

The validation of the proposed algorithm is carried out on a dual-arm robotic system consisting of two KUKA LBR robots, an IIWA7 and an IIWA14 spaced by 1m in the lateral direction. Each robot end-effector is mounted with a 3D printed grabber offering a contact surface of $(0.15 \times 0.1) m^2$. The robots are connected to a 3.4 GHz i7 PC, that runs the dual-arm controller. The desired task space velocity generated by the proposed law in Eq. (5.3.1) is executed by a passivity inspired torque controller of the form (Kronander and Billard, 2016).

⁴Alternatively, if the desired acceleration of the object $\ddot{\mathbf{x}}_o^d$ is available, \mathbf{f}_o^d can also be obtained from the inverse dynamics of the object as $\mathbf{f}_o^d = \mathbf{M}_o \ddot{\mathbf{x}}_o^d + \mathbf{b}_o + \mathbf{f}_{env}$

We extensively tested our algorithm both in Gazebo simulation and on the real robots, but we report primarily on results obtained on the real platforms, unless explicitly stated otherwise. To evaluate the performance of the proposed algorithm, we compare it to a coordination algorithm that contacts the object with near-zero velocity. We assess the controller’s ability to properly coordinate the robots’ motion to ensure that the two robot impact the object simultaneously. We further quantify the performance of the process by computing task duration and energy expenditure. Finally, we also show that such a tossing procedure expands the workspace of the robots and quantify the extension. A video of the corresponding experiments is provided as supplementary material and the code made available (see Section 1.7).

5.5.1 FAST OBJECT GRABBING WITH IMPACT

We evaluate here the motion coordination capability of our control scheme and its ability to execute fast grabbing with impact while generating contact wrenches that stabilize the grasp and perform the desired tossing motion of the object. The robotic task consists of reaching and swiftly grabbing an object of dimension $(0.2, 0.2 \times 0.2) \text{ m}$, mass 0.7 kg , initially located at $(0.41, 0.0 \times 0.32) \text{ m}$ and tossing it at $(0.7, 0.0 \times 0.7) \text{ m}$ with of velocity of $(0.8, 0.0 \times 0.8) \text{ m}$. Figure 5.6 provides some snapshots of the task execution.

The velocities of the dual arm system throughout the task are shown in Figure 5.7, where for both the left and the right arm, the norm of the velocity appears on top and a close-up views of the velocities between the contact and the release of the object appear at the bottom. The DS generating these velocities has been “boosted” by converting the generated motion flow into a unitary vector field modulated in amplitude by reaching speed and tossing speed during the reaching phase and the tossing phase, respectively. In Figure 5.7, it can be observed that the end-effectors reach and impact the object around $t=1.05\text{s}$ with a speed of 0.5 m/s . The impact induces a drop of the real velocities despite the increase of the desired end-effector velocities. A closer look at the bottom sub-figure indicates that it is mainly the y components (normal to the contact) that is dropping at the contact, whereas the x and z components slightly dropped while continuing their pre-impact trends in the direction of the desired motion of the object.

During the tossing phase, which starts from contact to release, the DS generates for the object a velocity with the desired tossing speed which is distributed between the arms. To further illustrate the motion and forces coordination between the two end-effectors, the grasp position errors and grasping forces during the task are shown in Figure 5.8. Despite the small offset of the right arm in the x direction (tangent to the contact), it can be noticed that both end-effectors make simultaneous contact with the object, while the grasping forces are quasi-symmetric in the contact direction.

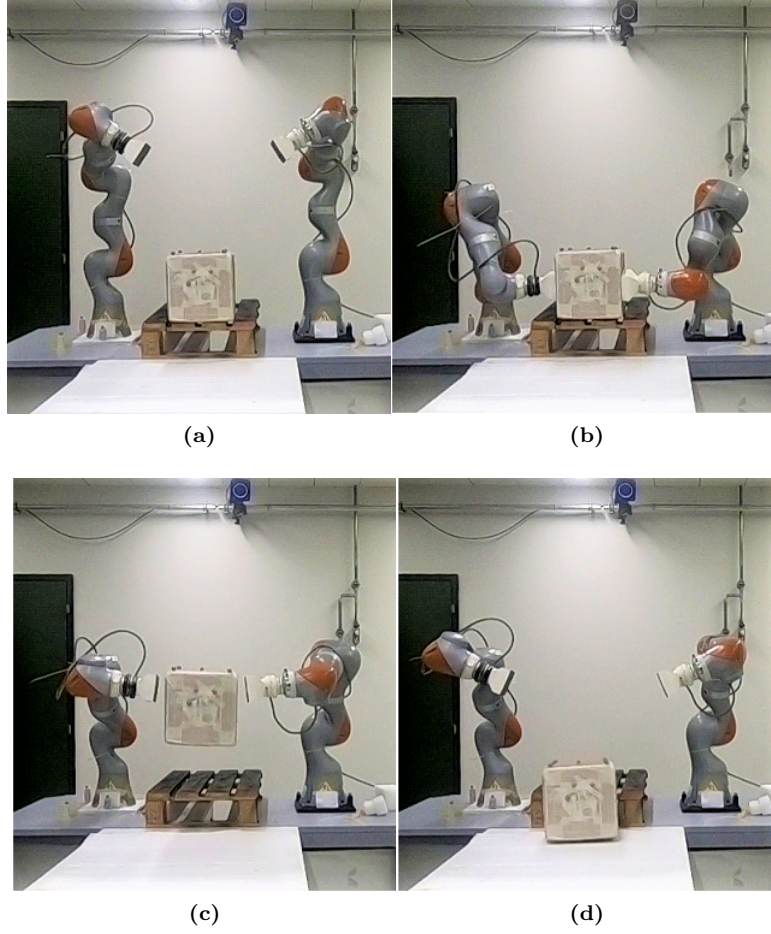
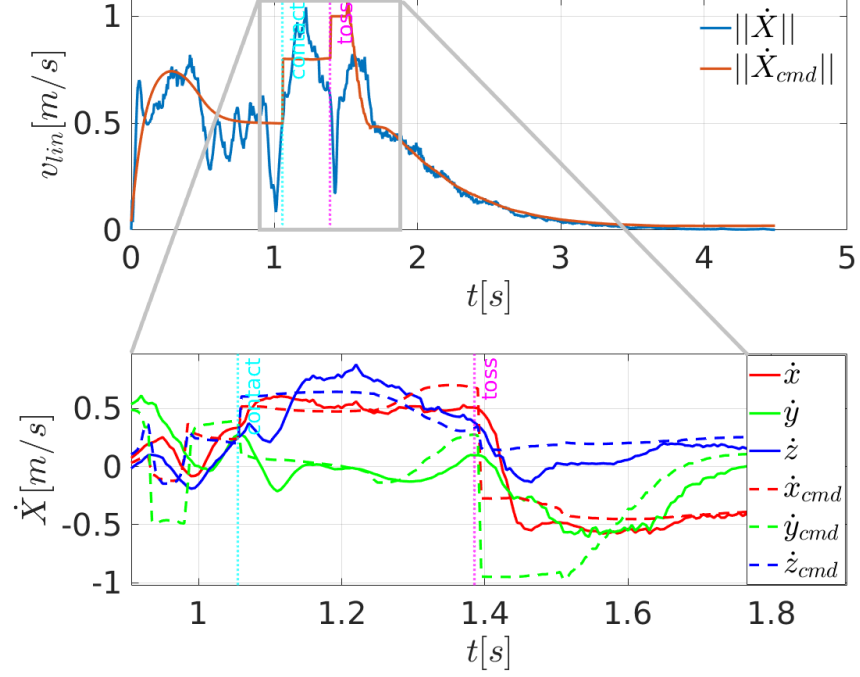


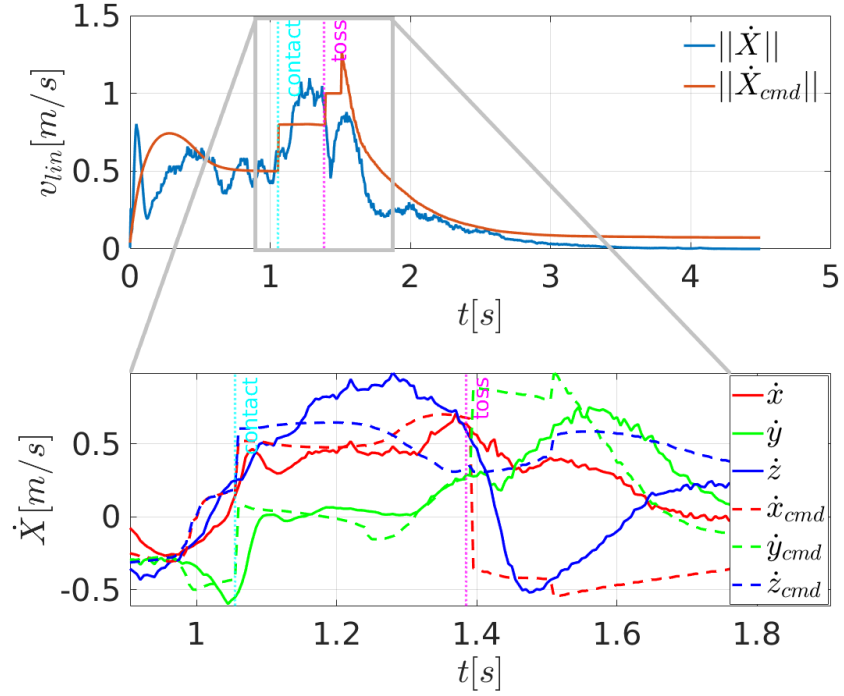
Figure 5.6: Snapshots of fast dual arm grabbing with impact and holding an object. From left to right, the snapshots show: the initial robots' configuration, the pose of the robots' arm at initial contact with the object, robot arms tossing the object with the desired tossing velocity and at the desired release location; and fourth, the landing of the object.

During the tossing phase, which starts from contact to release, the DS generates for the object a velocity with the desired tossing speed which is distributed between the arms. To further illustrate the motion and forces coordination between the two end-effectors, the grasp position errors and grasping forces during the task are shown in Figure 5.8. Despite the small offset of the right arm in the x direction (tangent to the contact), it can be noticed that both end-effectors make simultaneous contact with the object, while the grasping forces are quasi-symmetric in the contact direction.

The object's desired and real motion over time are shown in sub-figure 5.9a, whereas sub-figure 5.9b shows the 3D trajectories of the object and those of the dual-arm system for a task cycle.



(a) Left arm velocities



(b) Right arm velocities

Figure 5.7: Velocities of dual-arm end-effectors during a fast grabbing and tossing of an object at 0.8 m/s . (a): left arm (KUKA IIWA7) and (b): right arm (KUKA IIWA14). For both sub-figures (a) and (b): top row shows the norm of the velocities, and the bottom row shows a close-up view of the robots' linear velocities in the time period running from before contact and after releasing of the object.

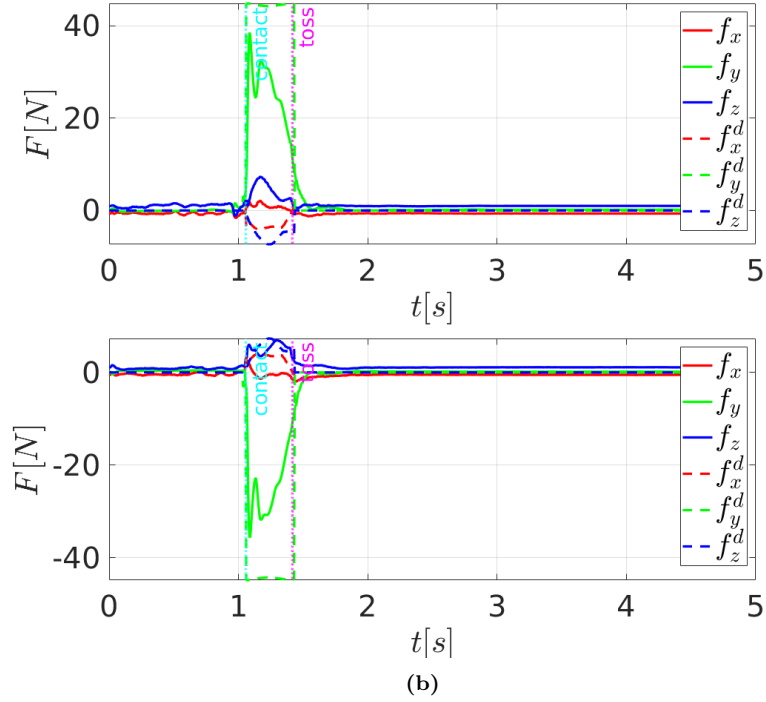
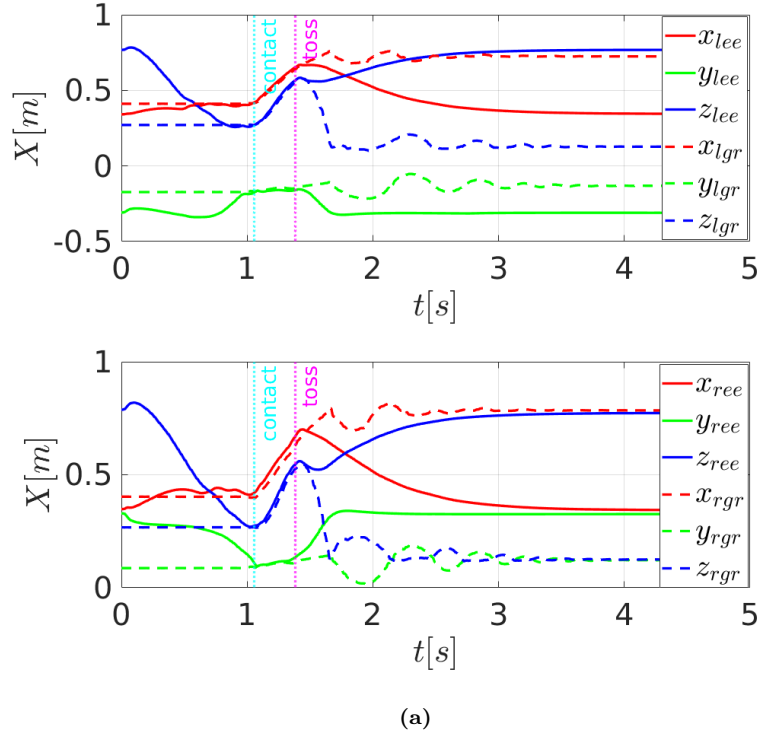
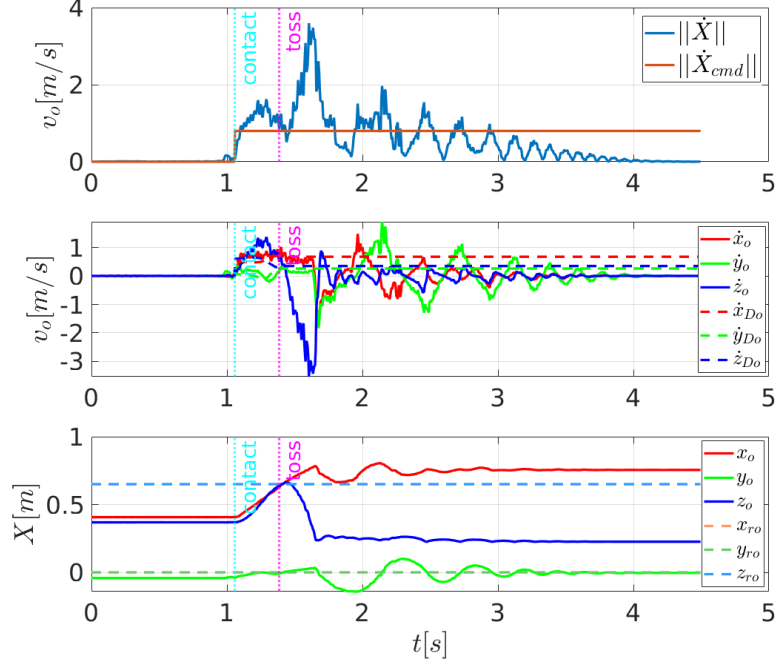
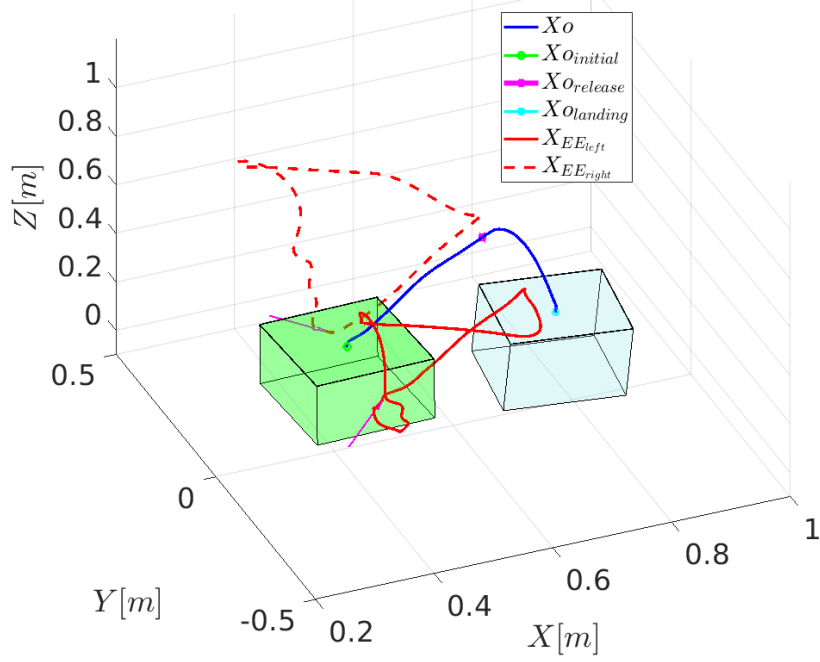


Figure 5.8: (a) Time evolution throughout the task of the position errors between the end-effectors and their respective grasping points on the object. The top and bottom sub-plots are respectively for the left and right end-effectors. (b) Time evolution throughout the task of grasping forces for the left and the right end-effectors respectively on the top and bottom sub-plots.



(a)



(b)

Figure 5.9: (a). Motion of the object during the fast grabbing and tossing task. top: norm of the linear velocity; middle: linear velocity; and bottom: position. The grabbing and lifting phases start at the contact indicated by a cyan vertical line, whereas the release instant is indicated by the magenta vertical line. (b). 3D trajectories of the overall dual arm and object system during the fast grabbing and tossing task. The red continuous and dotted lines are for the left and right end-effector, respectively, while the blue line is for the object. The object at its initial and final location is depicted by the light green and blue box, respectively. The magenta arrows indicate the direction of the impact velocity when the contact is established.

In sub-figure 5.9a, one can see from top to bottom, the time evolution of the velocity norm, the linear velocity components and the position of the object. It can be observed that the object carried by the two end-effectors follows its desired motion, although with tracking error, until it reaches the desired release location within a tolerance of 0.03m. Then, the object is suddenly released and its motion is now governed by the projectile dynamics. As the object is falling under the gravity field, its vertical velocity component (blue continuous lines on the middle plot) can be seen to linearly decrease while its position (on the bottom plot) decreases in a quadratic-like manner. How well, under the proposed scheme, the real position of the object denoted by (x_o, y_o, z_o) satisfies its desired position denoted by (x_{ro}, y_{ro}, z_{ro}) at the release time can be seen on the bottom plot of sub-figure 5.9a.

In sub-figure 5.9b, the 3D trajectory of the object is shown in solid blue from its initial position represented by the green box to its landing location represented by the light blue box. The end-effector trajectories represented in red described loops that start at the initial positions, go to the grasping points of the object, lift the latter towards the release position and go back to their initial positions. The magenta arrows represent the directions of the impact velocities of the end-effectors when grabbing the object; it can be noticed that they are aligned with the desired object motion.

A set of 50 trajectories of the tossed object with their landing positions along with the reachable space of the end-effectors is shown in Figure 5.10. Thus, with the range of achieved velocities, the “tossing reachable space” extends the dual-arm system reachable workspace by at least 20%. The joint velocity and torque limits during 30 experiments are provided in Appendix D.6

5.5.2 PICK-AND-PLACE VS. PROPOSED PICK-AND-TOSS

The goal of this experiment is to evaluate the benefits obtained, in terms of task duration (cycle time) and energy expenditure, in a depalletizing task when using the usual pick-and-place operation with near-zero contact and release velocities, and when using the proposed fast grabbing with impact and tossing. As previously, the task of the dual-arm system consists of grabbing an object from a pallet and moving it to a table located in front of the robots. The velocities of the robots resulting from the two approaches are shown respectively in Figure 5.11a (top) and (bottom), whereas the associated power and energy expenditure of both the left and right arm are shown in Figure 5.11b for the classical (top), and the proposed approach (bottom), respectively.

In the pick-and-place case, the robots contact the object with a velocity of 0.11 m/s mainly in the normal direction to the contact surfaces. The object is then moved with a velocity up to 1.0 m/s and later released with a velocity of

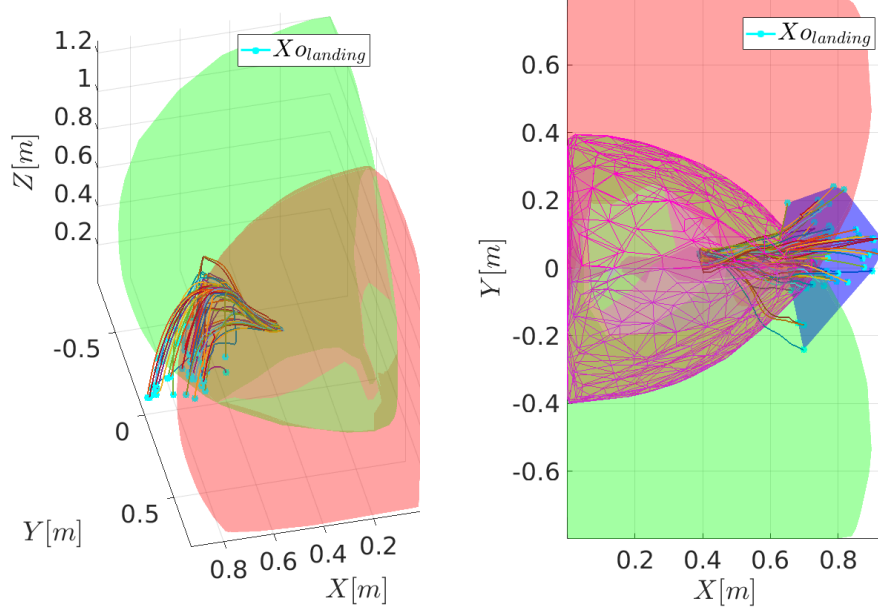


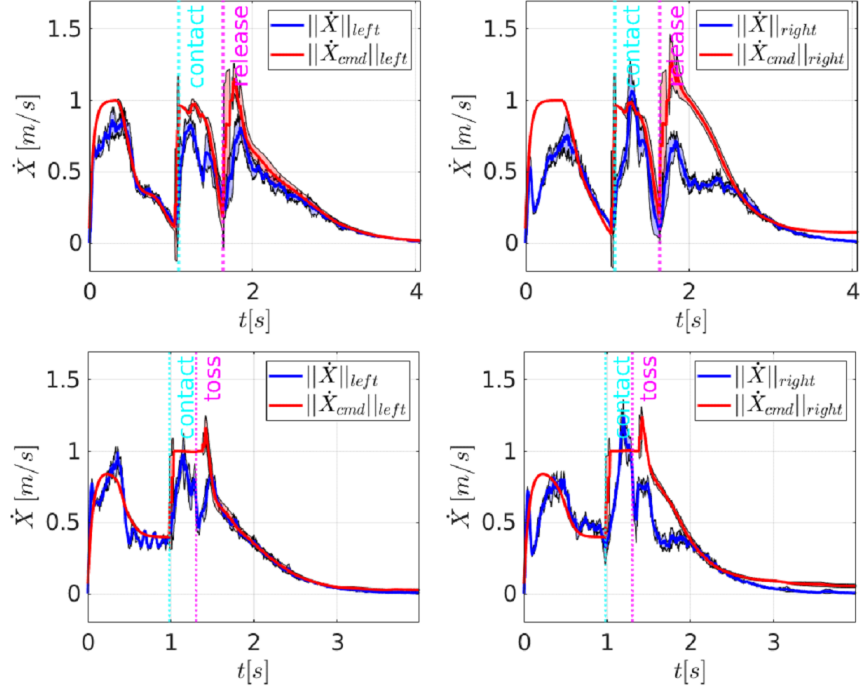
Figure 5.10: Extension of the dual-arm system's reachable space with the tossing reachable space. (left): set of 50 3D trajectories of tossed object with respect the dual-arm workspace (the green and red areas depict the left and right arms space of reachable positions). (right): top view of the tossing-based extension (blue area with the landing positions in cyan) of the joint reachable space depicted with the meshed region in magenta.

about 0.13 m/s at $t \approx 1.64$ m/s (see Figure 5.11a (top)). In the pick-and-toss case, like previously, the robots impact the object with an average speed of 0.40 m/s. The object is then moved throughout the task with a desired velocity of 1.0 m/s, but tossed instead of being gently placed (see Figure 5.11a (bottom)).

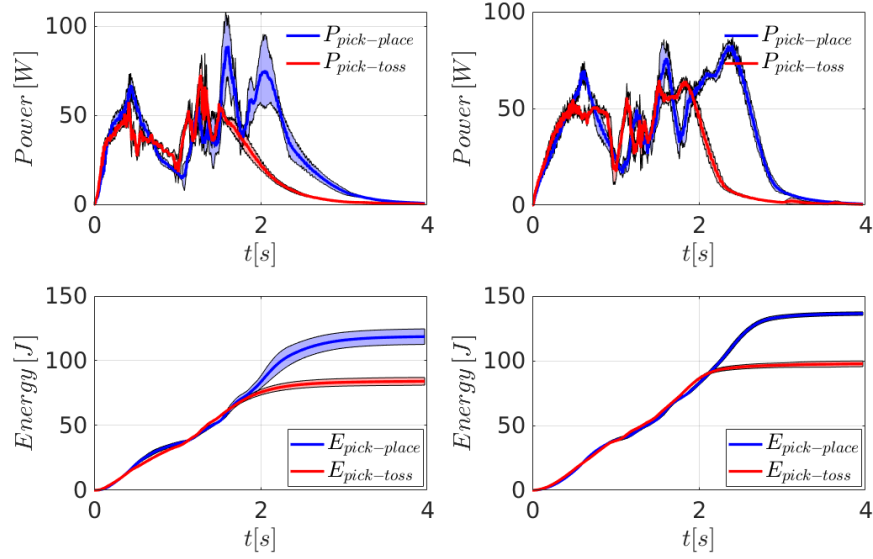
Thus, by reducing deceleration phase at pickup and release time, the task duration in the proposed approach for tossing velocity of 1.0 m/s is around 19% shorter than in the classical pick-and-place operation. Regarding the energy expenditure for the considered velocity, it can be observed in Figure 5.11b that the proposed approach consumed 24% less than the classical pick-and-place.

The difference of the observed variances between the two robots is mainly due to their different dynamic characteristics. The left arm corresponds to the KUKA LBR IIWA7 robot, which has smaller inertia properties and is less damped than the right arm (KUKA LBR IIWA14 robot), and therefore it can accelerate much faster for the same applied torque.

For velocities ranging from 0.5 m/s to 1.0 m/s, the comparison of the average task duration and the overall energy expenditure of the two approaches is summarized by the bar plots shown in Figure 5.12a and Figure 5.12b, respectively.



(a)



(b)

Figure 5.11: (a) Time evolution of mean and standard deviation of measured and commanded velocity norms of the left and right arm for five pick-and-placing (top), and five pick-and-tossing (bottom) experiments at 1.0 m/s. (b) Time evolution of estimated energy consumption of the left and right arm for five pick-and-placing (blue), and five pick-and-tossing (red) experiments at 1.0 m/s.

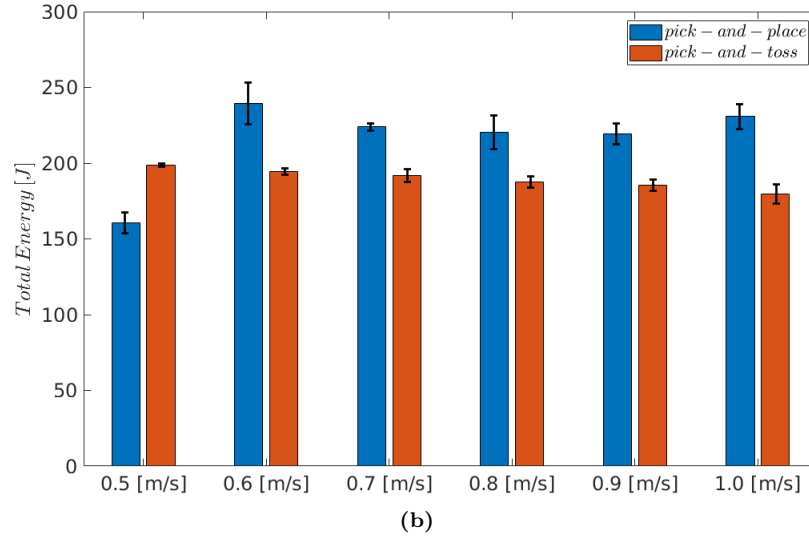
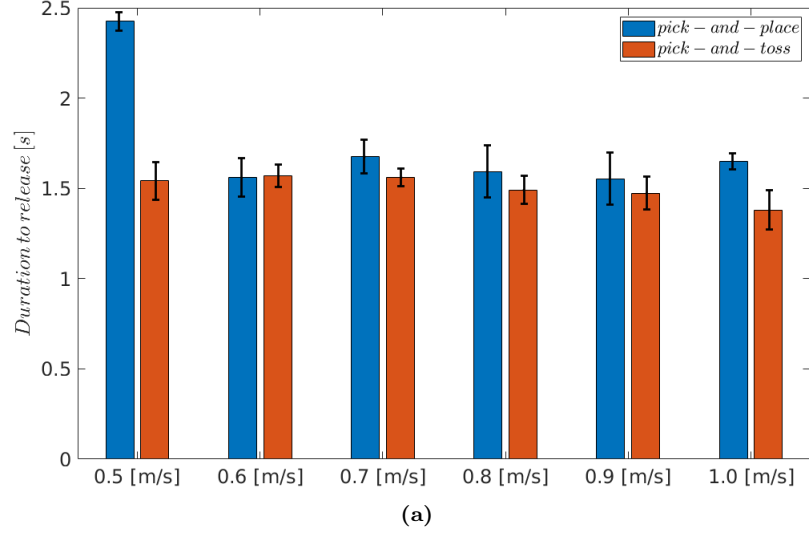


Figure 5.12: (a) Comparison for velocities ranging between 0.5 and 1.0 m/s of the task duration when the pick-up and placing happened at near-zero velocity (blue) and when the dual arm system leverage impact at the pickup and toss the object (red). (b) Comparison of overall energy expenditure of the dual arm system for tossing velocities ranging between 0.5 and 1.0m/s when grabbing the object with near-zero velocity (blue) and when performing fast object grabbing in a swipe (red).

5.6 Conclusion

This chapter proposed, for a dual-arm robotic system, a modulated DS-based unified motion generation algorithm that allows swift grabbing and release of an object. The desired states at contact and release are achieved through a local shaping of the robots movement. The proposed motion generation algorithm, whose stability and convergence towards the desired states was theoretically proved, has been experimentally validated in simulation and on real robots. The results confirmed that the proposed approach, besides motion coordination, enables to generate, for the dual-arm system, desired impact and tossing motions. The obtained results also suggest that grabbing with impact and tossing, especially when the impact direction anticipates the upcoming motion of the object, lead to shorter and more energy efficient pick and place tasks. These results are consistent with those obtained in (Raptopoulos et al., 2020), where a pick-and-toss approach with a Delta robot was compared to a classical pick-and-place approach in a waste sorting facility.

However, the application realm of such a dynamic pick and release extends well beyond the experiments conducted in this thesis. Bimanual grabbing of an object from the sides is more suitable than picking from the top with suction cups mechanisms in many situations, such as picking up filled trays with open lid, lifting cases with too fragile cover for supporting the case’s weight, or when placing boxes in shelves with limited vertical space. Also, it is worth noting that the proposed pick-and-toss scheme should be used for objects that may not risk breaking under the impact or when such damage is not important. This is important as the robot by itself cannot forecast the effects of impact on tossed objects. Tossing of boxes filled with open bottles, as shown the supplementary material, may be applicable when recycling or dispatching used bottles.

The system we described here can be integrated easily in current industrial settings using existing industrial manipulators. Our synchronized control law generates a reference trajectory that can be embedded in reference trajectories for an industrial controller. Additionally, nothing in our control system prevents a user to equip the robots with suction cups, instead of the flat grippers, used in our experiments. This would allow the robots to lift heavier weights. Moreover, the synchronization of the two robots could be easily disabled (by modifying Eq. 4 and setting A' to be block diagonal, and \mathbf{x}_* to specify one attractor for each arm). This would allow each robot to perform individual tasks, when appropriate; and to switch back to coordinated control, when the object that needs to be grabbed requires a bimanual grasp. Such an approach would then increase the flexibility of current industrial manipulators.

Nevertheless, the presented approach in its implementation has limitations. The Passive DS controller (Kronander and Billard, 2016) used to compute the joint torque from the generated motion has shown limited tracking abilities

particularly during the tossing. Moreover, we used first order linear DS, which works well for fixed or slowly varying attractors, but have no compensation ability in tracking and therefore differ this burden to the low-level controller (here the Passive DS). Future work could use second order DS able to address the compensation problem in tracking as in ([Mirrazavi Salehian et al., 2016](#)). In addition, instead of the Passive DS, one could use an inverse dynamic controller that not only compensates for gravity, but also for the inertial torques produced by strong accelerations during tossing. Finally, the impact and the object dynamics could also be included to guarantee the hardware safety and improve the task accuracy.

BIMANUAL DYNAMIC GRABBING AND TOSSING OF OBJECTS ONTO A MOVING TARGET

This Chapter presents a dynamic manipulation approach where a dual-arm robotic system picks up an object and throws it precisely on a moving target carried by a conveyor belt. It describes our formulation to determine the throwing parameters necessary to accomplish the precise positioning task. Then, it introduces a modeling method for the dual-arm tossable workspace and presents a dual-arm motion generation strategy that is robust and adaptive. Finally, the Chapter presents the experimental results validating the proposed approach and compares it to a traditional pick and place strategy before concluding.

Publication note: The material presented in this Chapter was adopted from:

- Bombile, M. and Billard, A., Bimanual Dynamic Grabbing and Tossing of Objects Onto a Moving Target.
Under review at the Journal of Robotics and Autonomous Systems.

Source codes:

- Dual arm controller
https://github.com/epfl-lasa/iam_dual_arm_control

Supplementary Video:

- Bimanual dynamic grabbing and tossing of objects onto a moving target
<https://youtu.be/pRSHH1866ug>

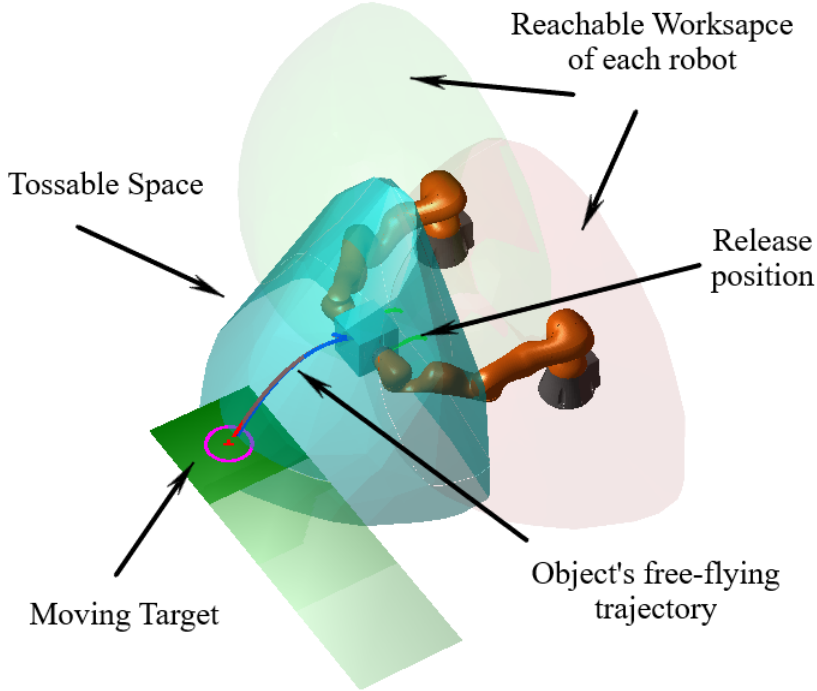


Figure 6.1: Example of dual-arm tossing task: an object is tossed at a target (in green) partly located in the convex hull of the throwable workspace (tossable space) (in cyan) of the dual-arm robot. The usual reachable workspace of each robot is shown in light green and light red (left and right arms, respectively).

6.1 Introduction

THROWING or tossing is a method for quickly moving objects to desired locations beyond the physical workspace. One can throw an object, if not too fragile, to someone located at a certain distance instead of walking. Throwing packages on a moving conveyor belt is fairly common in the logistics industry and storehouse handling, particularly in manual depalletizing or sorting facilities. Given the physical demands of such manual tasks and considering the aging population in the West (Nations et al., 2019) this thesis proposes a robotic solution for throwing packages to a target moving along a conveyor belt as a fast and reliable alternative for package handling.

For throwing to be useful in industry, it must be accurate. The thrown object must land on the target within a given tolerance. This implies the knowledge of the throwing parameters, particularly the release position and velocity, which must be appropriate for the thrown object to reach its target. Once an object is thrown, the thrower can no longer apply corrective action if the object does not follow the desired trajectory.

As outlined in Section 2.4.4, tossing an object onto a moving target using a robotic system is challenging. It requires a solution to the following main sub-

problems: 1) Finding a feasible intercept position for the thrown object to meet the moving target. 2) Finding feasible throwing parameters (release position, release speed, and direction). 3) Generating the motion of the robot to pick up the object and successfully reach the desired release state on time, such that the thrown object intercepts the target at the desired location.

We saw that the throwing parameters can be determined using physics-based models or data-based models. In general, physics-based models allow better generalization in that they can be applied in different conditions and with different objects. However, their precision depends on the simplifying assumptions they are based on, especially with complex phenomena. Data-based models are able to capture these complex phenomena and can be more precise. However, their ability to generalize to other objects and conditions is limited. Hence, in this thesis, to leverage the strengths of both models, we use a hybrid modeling approach where a throwing map is learned from a parameterized physics-based model of the free-flying dynamics.

Unlike works such as (Zeng et al., 2020) that consider only the release speed as the main throwing parameter, our approach considers the full release state (position and velocity) and ensures their kinematic feasibility. The optimality of the release velocity is encoded in a learned model of the throwing maps used in a bi-level optimization to compute the release states of both the object and robots.

Moreover, while previous robotic interception works whether with a single arm or dual arm system usually consider a fully controllable interceptor (the robot), this thesis considers an interceptor that is only partially controllable (the thrown object is only controllable up to the release).

Despite reports on successful dual-arm coordinated interception tasks including grabbing and catching flying objects, a post-grabbing or post-catching task such as placing or tossing task has not been considered. Although we addressed such a problem in the previous Chapter and demonstrated dual-arm fast grabbing and tossing of objects, the focus was however on the motion generation using a dynamical system. We did not consider finding feasible throwing parameters for the throwing task. These parameters were directly provided to the algorithm. In this Chapter, we determine the robot’s feasible throwing states for the desired moving target.

Furthermore, we proposed to estimate the throw-able space based on the kinematic feasibility of release states associated with landing points spanning the robot workspace and beyond. Unlike (Asgari and Nikoobin, 2021), we go beyond the estimation of the throwable points, we derive a probabilistic model of their distribution and use it to predict landing (intercept) positions that yield a high probability of success.

In summary, in this Chapter we propose a framework that enables a dual-arm robotic system to quickly grab and accurately toss an object to a target moving on a conveyor belt. We extend the work presented in Chapter 5 by including

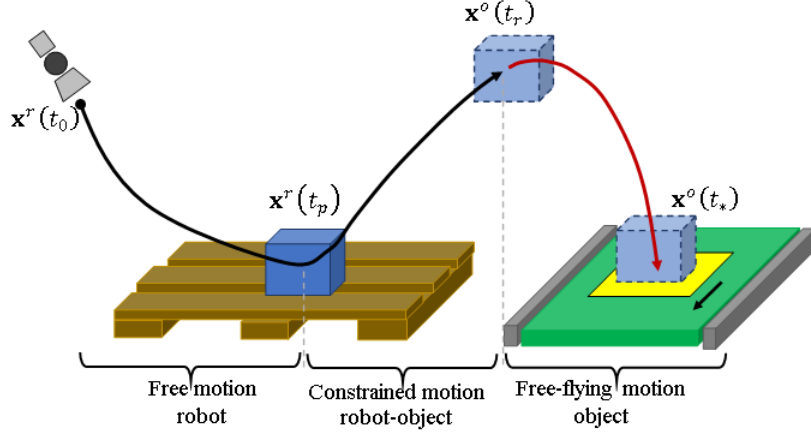


Figure 6.2: Illustration of dual-arm robotic depalletizing task with fast grabbing and tossing of an object. The overall motion can be split into three main phases; the system motion is determined by the free robot motion, the constrained robot–object motion, and the object free-flying motion before interception of the target at landing.

the free-flying object dynamics in the determination of the tossing velocity and position. In addition, we propose a mixed learning–optimization approach to compute an optimal kinematically feasible release state (position and velocity) for the dual-arm system. We use this framework to learn the tossing workspace (tossable space) of the dual-arm robot; it represents the set of all positions reachable by a given object if tossed by the dual-arm robot. Finally, we present an experimental validation and evaluation demonstrating that an actual dual-arm robotic system quickly grabbing an object and tossing it to intercept a specified target on a moving conveyor belt is advantageous in terms of task completion time and energy expenditure compared with more traditional pick and place systems.

6.2 Problem Definition and Proposed Approach

Consider a dual-arm robotic system tasked to quickly grab an object from a pallet and toss it onto a moving target on a conveyor belt, as shown in Figure 6.2. The system motion throughout the task is characterized by three main phases: a free-motion phase where only the robot moves (from initial time (t_0) to pick up time (t_p)), a constrained robot–object motion phase (from (t_p) to the release time (t_r)), and a free-flying motion phase for the object (from (t_r) to the landing time (t_l)). We assume that the task is executed under the following assumptions:

A6.1: the dual-arm robot establishes rigid contacts with the grasped object until its release.

A6.2: the object is rigid with known shape and dimensions, and known inertia properties (mass and moments of inertia).

A6.3: during its free-flying motion phase, the object does not rotate or tumble and is subjected only to gravity and aerodynamic drag forces.

A6.4: the post-landing impact of the object is negligible

6.2.1 ROBOT-OBJECT MOTION CONSTRAINTS

The dynamic model governing the motion of a dual-arm robot interacting with its environment has been described in Section 2.2.1. Now, according to assumption **A6.1**, when the object is firmly grasped (during the constrained robot-object motion phase), the kinematic constraints imposed by the rigid grasp (described earlier in Eq. (2.2.6)) can be rewritten as

$$\underbrace{(\mathbf{G}_o^+)^\top \mathbf{J}_e(\mathbf{q})}_{\mathbf{J}_o(\mathbf{q})} \dot{\mathbf{q}} = \begin{bmatrix} \dot{\mathbf{x}}^o \\ \boldsymbol{\omega}^o \end{bmatrix} \quad \text{and} \quad \underbrace{(\mathbf{I} - \mathbf{G}_o^+ \mathbf{G}_o)^\top \mathbf{J}_e(\mathbf{q})}_{\mathbf{J}_c(\mathbf{q})} = \mathbf{0} \quad (6.2.1)$$

where $\dot{\mathbf{x}}^o$ and $\boldsymbol{\omega}^o$ are the linear and angular velocity of the object, respectively. $\mathbf{J}_o(\mathbf{q}) \in \mathbb{R}^{6 \times n^D}$ and $\mathbf{J}_c(\mathbf{q}) \in \mathbb{R}^{12 \times n^D}$ denote the object's motion and contact constraint Jacobian, respectively. We recall that $\mathbf{G}_o \in \mathbb{R}^{6 \times 12}$ denotes the bimanual grasp matrix (Caccavale and Uchiyama, 2016) and \mathbf{G}_o^+ is its generalized inverse.

Therefore, during the constrained motion, all states of the dual-arm are restricted within the kinematic feasible set, \mathcal{F} , defined by

$$\mathcal{F} = \left\{ \mathbf{s} = \begin{bmatrix} \mathbf{x}(\mathbf{q}) \\ \dot{\mathbf{x}}(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} \left| \begin{array}{l} \mathbf{q}_{\min} \leq \mathbf{q} \leq \mathbf{q}_{\max} \\ \mathbf{J}_o(\mathbf{q}) \dot{\mathbf{q}} = [\dot{\mathbf{x}}^o{}^\top \boldsymbol{\omega}^o{}^\top]^\top \\ \mathbf{J}_c(\mathbf{q}) \dot{\mathbf{q}} = \mathbf{0} \\ |\dot{\mathbf{q}}| \leq \dot{\mathbf{q}}_{\max} \end{array} \right. \right\} \quad (6.2.2)$$

where $\mathbf{s} = (\mathbf{x}(\mathbf{q}), \dot{\mathbf{x}}(\mathbf{q}))$ represents the task-space state of the dual-arm robot; $\mathbf{x}(\mathbf{q})$ and $\dot{\mathbf{x}}(\mathbf{q}, \dot{\mathbf{q}})$ are the pose and twist velocity vectors, respectively, obtained through forward kinematics. \mathbf{q}_{\min} , \mathbf{q}_{\max} and $\dot{\mathbf{q}}_{\max}$ are the joint position and velocity limits, respectively.

Following assumption **A6.2**, the dynamic model of an object with mass m_o and inertia \mathcal{I}_o , described earlier in Section 2.2.2, can be expressed as

$$\begin{bmatrix} m_o \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathcal{I}_o \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}}^o \\ \dot{\boldsymbol{\omega}}^o \end{bmatrix} + \begin{bmatrix} -m_o \mathbf{g} \\ \boldsymbol{\omega}^o \times \mathcal{I}_o \boldsymbol{\omega}^o \end{bmatrix} = \begin{bmatrix} \mathbf{f}_o \\ \boldsymbol{\tau}_o \end{bmatrix} \quad (6.2.3)$$

where $\ddot{\mathbf{x}}^o \in \mathbb{R}^3$ and $\dot{\boldsymbol{\omega}}^o \in \mathbb{R}^3$ are the object's linear and angular acceleration, respectively. $\mathbf{g} \in \mathbb{R}^3$ represents the gravity vector; $\mathbf{f}_o \in \mathbb{R}^3$ and $\boldsymbol{\tau}_o \in \mathbb{R}^3$ denote the effective force and torque, respectively, acting at the object's frame Σ_o .

During the free-flying motion phase, considering assumption **A6.3** and further assuming that the center of mass (CoM) of the object corresponds to its geometric center, the object's dynamics reduces to the first row of Eq. (6.2.3). Thus, its motion will be governed by

$$\ddot{\mathbf{x}}^o = \mathbf{g} + \frac{\mathbf{f}_D}{m_o} \quad (6.2.4)$$

where $\mathbf{f}_D = \mathbf{f}_o$ and represents the aerodynamic drag force (the lift force is neglected) expressed as

$$\mathbf{f}_D = -\frac{\rho_{air} c_D A_o}{2} \dot{\mathbf{x}}^o \|\dot{\mathbf{x}}^o\| \quad (6.2.5)$$

where ρ_{air} is the air density, c_D denotes the drag coefficient and A_o represents the cross sectional area of the object in the motion direction. Eq. (6.2.4) can be simplified and written as a function of the object's states (\mathbf{x}^o and $\dot{\mathbf{x}}^o$) as

$$\frac{d}{dt} \begin{pmatrix} \mathbf{x}^o \\ \dot{\mathbf{x}}^o \end{pmatrix} = \begin{pmatrix} \dot{\mathbf{x}}^o \\ -\eta \dot{\mathbf{x}}^o \|\dot{\mathbf{x}}^o\| + \mathbf{g} \end{pmatrix} \quad (6.2.6)$$

where $\eta \triangleq \frac{\rho_{air} c_D A_o}{2m_o}$. With these assumptions, Eq. (6.2.6) allows prediction of the object trajectory from a given initial state that denoted as $\mathbf{s}_0^o \triangleq (\mathbf{x}_0^o, \dot{\mathbf{x}}_0^o)$ to its landing state denoted as $\mathbf{s}_l^o \triangleq (\mathbf{x}_l^o, \dot{\mathbf{x}}_l^o)$.

6.2.2 PROBLEM STATEMENT

Consider the robot-object motion constraints in Eqs. (6.2.1) and (6.2.2) and the object's dynamics in Eqs. (6.2.3) and (6.2.6). As in the previous Chapters, assume also that the robot is equipped with a low-level controller that can generate torque commands based on manipulation task expressed in terms of desired end-effector motion ¹ $\dot{\mathbf{x}}^d$ and forces ² \mathbf{f}_e^d . To accomplish the dual-arm task consisting of grabbing and tossing an object onto a moving target, the following three problems must be solve:

P6.1: how to determine a reachable intercept location (denoted by $\mathbf{x}_I \in \mathbb{R}^3$) between the tossed object and the moving target;

P6.2: how to determine the object's release position and velocity $\mathbf{s}_r^{o*} \triangleq (\mathbf{x}_r^{o*}, \dot{\mathbf{x}}_r^{o*})$ and the associated release state of the dual-arm system $\mathbf{s}_r^* \triangleq (\mathbf{x}_r^*, \dot{\mathbf{x}}_r^*)$ that will result in the tossed object landing at the intercept location \mathbf{x}_I ;

P6.3: how to generate the desired robot motion $\dot{\mathbf{x}}^d$ and force \mathbf{f}_e^d to grab the object and successfully reach the desired release state \mathbf{s}_r^* on time such that the thrown object intercepts the target at \mathbf{x}_I .

Addressing (**P6.1**) requires a model of the dual-arm robot tossable workspace to predict the likelihood of reaching any potential intercept location.

¹The motion task can also be expressed in terms of $\ddot{\mathbf{x}}^d$

²On the object side, \mathbf{f}_e^d is distributed between effective (\mathbf{f}_o , τ_o) and contact wrench

Solving (P6.2) requires finding a robot state that satisfies both kinematic feasibility constraints and grasping constraints on the object. Thus, according to (6.2.2), a valid release state for the robot should satisfy $\mathbf{s}_r^* \in \mathcal{F}$.

Solving (P6.3) is nothing but developing a robust coordinated motion and force control strategy for the dual-arm system to reach, grab, and toss the object to ensure successful interception of the moving target by the object in the presence of spatial and temporal perturbations. Such perturbations could consist of a live modification of the target motion (speeding up or slowing down) or a displacement of the target location on the conveyor belt.

6.2.3 OVERVIEW OF PROPOSED APPROACH

Our proposed approach to address the above problem and achieve dual-arm grabbing and precise tossing of an object on a moving target consists of three parts:

- S6.1:** devise a kinematics based optimization algorithm that computes constraints-satisfying throwing states while minimizing the throwing speed (see Section 6.3);
- S6.2:** learn a model of the tossable workspace, \mathcal{S}_T , from a distribution of reachable points determined using the algorithm in S6.1 (see Section 6.4);
- S6.3:** generate the desired motion $\dot{\mathbf{x}}_d$ and force \mathbf{f}_e^d using our previously developed dynamical system based dual-arm controller for fast grabbing and tossing of objects complemented with an adaptation factor $a(\mathbf{x})$ to compensate for changes of the target's motion, and prediction and tracking inaccuracies of the robot's motion (see Section 6.5)

The proposed approach is summarized in the control architecture illustrated in Figure 6.3. Target position measurements are used to estimate target motion and predict target trajectory (determined by the conveyor belt). The predicted trajectory of the target and the learned tossable workspace of the dual-arm robot and its predicted motion are used to determine an optimal interception point updated over time. The computed interception position is the desired landing position of the thrown object, and is used in conjunction with the learned projectile throwing map to determine a feasible release state of the object. The obtained feasible release state is sent as a reference for the DS-based controller. The motion generated by the DS-based controller is adapted to compensate for state prediction and control error during execution of the desired motion.

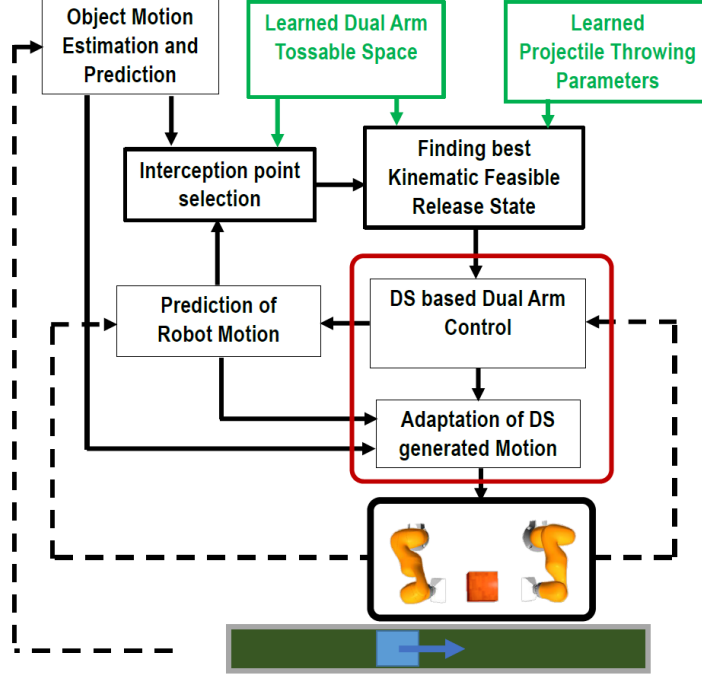


Figure 6.3: Block diagram showing processes and information flow in proposed approach for dual-arm tossing of an object onto a moving conveyor belt. The blocks "Learned Dual-Arm Tossable Space" and "Learned Projectile Throwing Parameters" are trained offline. The black continuous and dashed lines represent information flow updated at 200 Hz and 10 Hz, respectively. .

6.3 Estimation of Feasible Throwing States

In this section, we propose an approach for computing kinematically feasible throwing release states (position and velocity) for a dual-arm robotic system from desired landing positions. Finding the optimal feasible release state also generally implies finding the best objective function. However, our primary goal in this section is to find a solution that satisfies feasibility constraints rather than finding the best solution. Thus, obtaining a feasible release state is considered as a success.

6.3.1 LEARNING AN INVERSE THROWING MAP

Unlike the forward dynamics in Eq. (6.2.6), we are now concerned with the inverse throwing problem, which consists of finding an initial throwing state $(\mathbf{x}_0^o, \dot{\mathbf{x}}_0^o)$ that yields a desired landing position³ \mathbf{x}_l^o . As stated before, this problem is not trivial and admits multiple solutions.

³We leave the landing velocity $\dot{\mathbf{x}}_l^o$ as a free variable as we are more concerned with the object's landing accuracy on the target than the landing speed.

To address this challenge, we propose a two-step approach:

Step 1: We determine a function $\dot{\mathbf{x}}^o = \mathbf{v}_r(\bar{\mathbf{x}}^o) \in \mathbb{R}^3$ that computes the release velocity from the relative position $\bar{\mathbf{x}}^o \in \mathbb{R}^3$ between the desired landing position and the release position of the object ($\bar{\mathbf{x}}^o = \mathbf{x}_l^o - \mathbf{x}_0^o$). We resolve the inherent redundancy problem by adopting a throwing strategy that seeks the minimal throwing velocity. Such a strategy is also beneficial for the robot as it requires less kinetic energy for throwing. Other strategies such as minimum landing velocity (vertical or horizontal components) can be adopted (see Appendix E.2).

However, except for linear projectile motion, $\dot{\mathbf{x}}^o = \mathbf{v}_r(\bar{\mathbf{x}})$ does not have a closed-form solution mainly due to the non-linear aerodynamic forces. Thus, we propose a closed-form expression of $\mathbf{v}_r(\bar{\mathbf{x}}^o)$ learned from data. To that end, we use Gaussian mixture regression (GMR) (Sung, 2004) for its ability to handle multi-dimensional input and output data. We define throwing situations for a given object as $\mathcal{C}_v = \{\bar{\mathbf{x}}^o, \dot{\mathbf{x}}^o\}$ and model a dataset of N such throwing situations ($\{\mathcal{C}_v^i\}_{i=1\dots N}$) using a GMM. The model is assumed to have K_v Gaussian functions, can be represented by its parameters as $\Omega_{v_r} \equiv \{\pi^k, \boldsymbol{\mu}^k, \boldsymbol{\Sigma}^k\}_{k=1\dots K_v}$, where π^k , $\boldsymbol{\mu}^k$ and $\boldsymbol{\Sigma}^k$ are the prior, the mean and the covariance of the k^{th} Gaussian distribution, respectively.

At any query time, given a desired relative release position $\bar{\mathbf{x}}^{o*}$, the desired release velocity $\dot{\mathbf{x}}^{o*}$ is obtained by computing the expectation over the conditional distribution $p(\dot{\mathbf{x}}^{o*} | \bar{\mathbf{x}}^{o*}, \Omega_{v_r})$. The resulting function can be written as

$$\dot{\mathbf{x}}^o \approx \sum_{k=1}^K h^k(\bar{\mathbf{x}}^o) \tilde{\boldsymbol{\mu}}_{\dot{\mathbf{x}}^o | \bar{\mathbf{x}}^o}^k(\bar{\mathbf{x}}^o) \quad (6.3.1)$$

with $\tilde{\boldsymbol{\mu}}_{\dot{\mathbf{x}}^o | \bar{\mathbf{x}}^o}^k = \boldsymbol{\mu}_{\dot{\mathbf{x}}^o}^k + \boldsymbol{\Sigma}_{\dot{\mathbf{x}}^o \bar{\mathbf{x}}^o}^k (\boldsymbol{\Sigma}_{\bar{\mathbf{x}}^o \bar{\mathbf{x}}^o}^k)^{-1} (\bar{\mathbf{x}}^o - \boldsymbol{\mu}_{\bar{\mathbf{x}}^o}^k)$, where $\boldsymbol{\mu}_{\bar{\mathbf{x}}^o}^k$ and $\boldsymbol{\mu}_{\dot{\mathbf{x}}^o}^k$ are element vectors of the mean $\boldsymbol{\mu}^k$ of the k^{th} Gaussian function associated with input data $\bar{\mathbf{x}}^o$ and output data $\dot{\mathbf{x}}^o$, respectively. Similarly, $\boldsymbol{\Sigma}_{\bar{\mathbf{x}}^o \bar{\mathbf{x}}^o}^k$ and $\boldsymbol{\Sigma}_{\dot{\mathbf{x}}^o \bar{\mathbf{x}}^o}^k$ are element matrices extracted from the covariance matrix $\boldsymbol{\Sigma}^k$. In Eq. (6.3.1), $h^k(\bar{\mathbf{x}}^o)$ weights the relative importance of the k^{th} Gaussian function in the regression of $\mathbf{v}_r(\bar{\mathbf{x}}^o)$.

Step 2: Using the obtained expression of $\mathbf{v}_r(\bar{\mathbf{x}}^o)$, we seek through an optimization process a value of the release position \mathbf{x}_0^o that is reachable by the robot and yields a minimum feasible $\dot{\mathbf{x}}^o$. To that end, we must derive the Jacobian between variations of \mathbf{v}_r and $\bar{\mathbf{x}}^o$. With the smoothness of Gaussian functions, this Jacobian can be derived in closed form. As $\bar{\mathbf{x}}^o$ is a function of the robot's forward kinematics, such that

$$\bar{\mathbf{x}}^o = \mathbf{x}_l^o - \mathbf{x}^o(\mathbf{q}) \quad (6.3.2)$$

the Jacobian between variations of \mathbf{v}_r and \mathbf{q} can be obtained as

$$d\mathbf{v}_r = \mathbf{J}_{\mathbf{v}_r}(\mathbf{q})d\mathbf{q} = \underbrace{\frac{\partial \mathbf{v}_r}{\partial \bar{\mathbf{x}}^o}}_{\mathbf{J}_{\mathbf{v}_r}(\bar{\mathbf{x}}^o)} \underbrace{\frac{\partial \bar{\mathbf{x}}^o}{\partial \mathbf{x}^o}}_{-\mathbf{I}_{3 \times 3}} \underbrace{\frac{\partial \mathbf{x}^o}{\partial \mathbf{q}}}_{\mathbf{J}_{\text{ov}}(\mathbf{q})} d\mathbf{q} \quad (6.3.3)$$

where the expression of $\mathbf{J}_{\mathbf{v}_r}(\bar{\mathbf{x}}^o)$ is presented in Appendix (E.1). The $\mathbf{J}_{\text{ov}}(\mathbf{q}) \in \mathbb{R}^{3 \times n^D}$ term is simply the translational block component of the direction of the object's motion Jacobian $\mathbf{J}_o(\mathbf{q})$ in (6.2.1).

DATA GENERATION AND MODEL TRAINING

The inverse throwing map in Eq. (6.3.1) encodes release velocity solutions when the release position is given and a minimum release speed strategy is adopted. These are parameterized by η (defined in Eq.(6.2.6)).

Thus, to train the model of the inverse throwing map and thereby validate our approach, we need examples of throwing situations (set of release positions and release velocities). To that end, we proceed as follows.

First, we artificially generate 10^5 3D relative release position corresponding to 20 values of the non-linear drag force coefficient η within a spherical sector defined by a throwing reach⁴ of $[0 - 2.5]m$, and throwing directions within a cone angle of $[\pm \frac{5\pi}{12}] rad$ around the x axis as shown in Figure 6.5-(left).

Second, to generate the release velocity $\dot{\bar{\mathbf{x}}}$ corresponding to the generated set of $\bar{\mathbf{x}}$, we solve, for each point a two-point boundary value problem (TPBVP) Laetsch and Keller (1970) defined by the object's free-flying dynamics (6.2.6) with the origin $(0, 0, 0)$ as initial position and $\bar{\mathbf{x}}$ the final position. We solve this problem using a shooting algorithm.

We assume that the projectile motion lies in a plane to reduce the dimensionality of the problem from 3D to 2D. We use a Cartesian-to-cylindrical coordinate transformation to extract the equivalent planar coordinates lying in the vertical plane containing the origin and the landing position. This yields a planar release velocity parameterized by the release angle α_0 and release speed v_0 . To determine the release angle α_0^* and speed v_0^* , we initialize our shooting algorithm with the analytical solution of the linear ballistic motion that yields the minimal release speed.

$$\begin{aligned} \alpha_0 &= \arctan \left(\frac{z}{r} + \sqrt{\left(\frac{z}{r}\right)^2 + 1} \right) \\ v_0 &= \sqrt{\frac{g \cdot r^2 \cdot (1 + \tan^2(\alpha_0))}{2 \cdot (r \cdot \tan(\alpha_0) - z)}} \end{aligned} \quad (6.3.4)$$

Our shooting algorithm has two stages. In the first stage, using v_0 and starting at α_0 , we search for the angle α_0^* that yields the maximal reach with the velocity v_0 . Once α_0^* is obtained, the second stage searches for the release speed that

⁴This restriction on the spatial span of data considers the robot limits; for instance, a maximum reach of 2.5 m requires a minimum release speed of 5 m/s, which is beyond the robot capability.

leads to the landing position (r_l, z_l) , with the speed iteratively updated as

$$v_0(i+1) = v_0(i) + \kappa(r_l^d - r_l) \quad (6.3.5)$$

where κ denotes the update rate of the algorithm. The algorithm stops when the predicted release position reach the desired landing position within a small tolerance distance. Figure 6.4 illustrates the obtained planar throwing parameters (Figure 6.4(a)), with samples of three different values of η (Figure 6.4-(b)).

Once the release speed v_0^* and angle α_0^* were obtained, we reconstructed the 3D equivalent $\dot{\mathbf{x}}^o$, which together with $\bar{\mathbf{x}}^o$ constitute the dataset required for the training. The resulting dataset with position and velocity data points is shown in Figure 6.5.

With the obtained dataset, we trained our GMM model with two-thirds of the data using the expectation-maximization (EM) algorithm initialized with k-means. Our model uses the Bayesian information criterion (BIC) to choose the number of Gaussian functions, 25 in this case. With the obtained GMM, we can predict with GMR the throwing release velocity based on the desired relative position $\bar{\mathbf{x}}^o$.

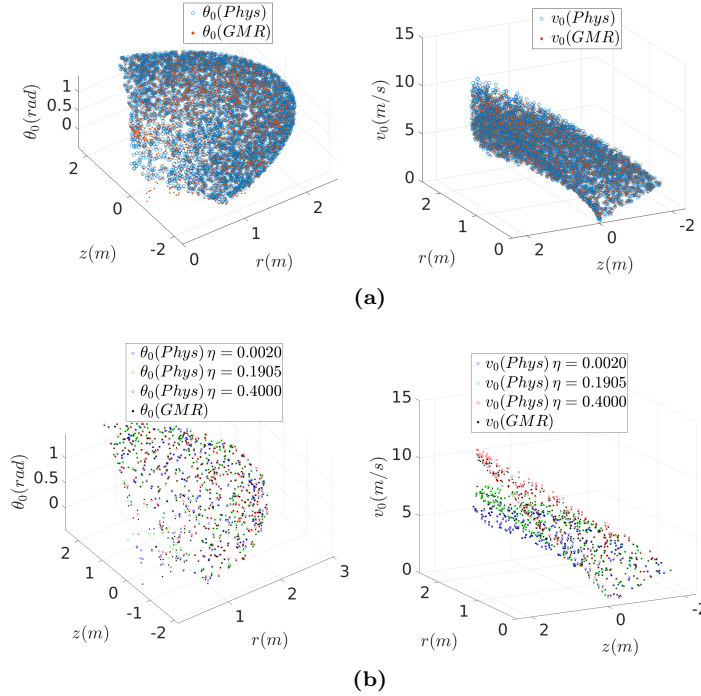


Figure 6.4: Illustration of optimal planar throwing parameters: (a) 5000 samples of throwing parameters; (b) throwing parameters corresponding to three different values of non-linear drag force η

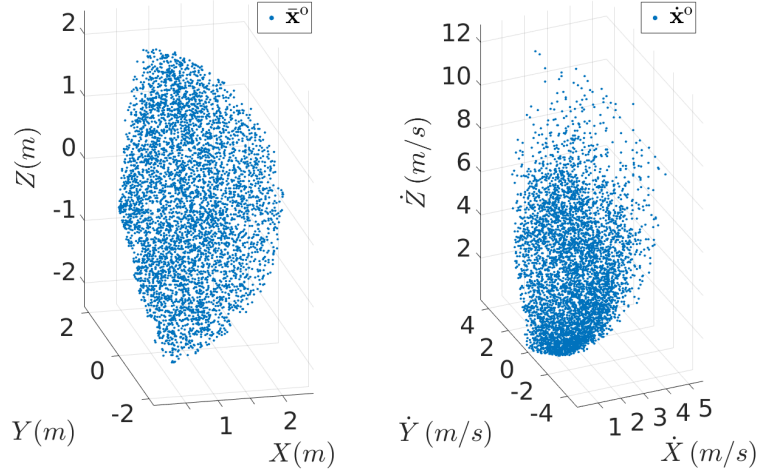


Figure 6.5: Example of dataset used to learn the inverse throwing map. The left figure shows the desired landing position relative to the frame origin; the right figure shows the corresponding 3D release velocities obtained when using a minimal throwing release speed strategy.

We validated our GMR model using the remaining one-third of the data. We obtained an RMSE of 0.13 m/s for the throwing velocities. Figure 6.6 shows error histograms between velocities generated by the learned GMR model and ground truth velocities (physics) in the x , y , and z dimensions.

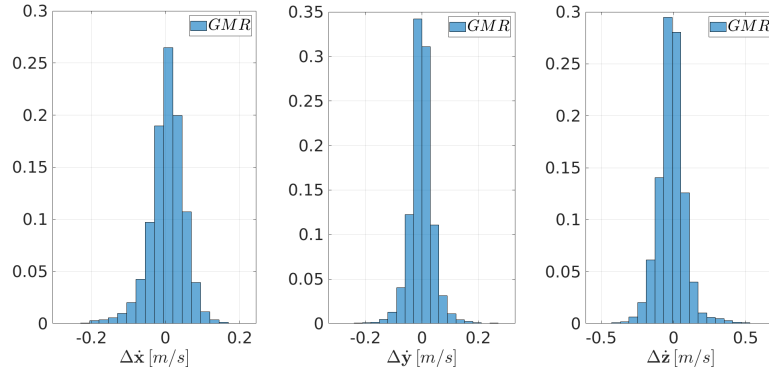


Figure 6.6: Validation errors between velocities generated by the learned GMR model and ground truth velocities (physics): velocity errors in x (left), in y (middle), and in z (right)

The landing position errors corresponding to the testing data set are illustrated in Figure 6.7 along with the ground truth landing errors (physics-based model). It is observed that the physics-based parameters are accurately predicted by the GMR; 97% of the predicted landing positions fall within 0.1m (4% of maximum reach of 2.5m) of the target and 80% fall within 0.05m.

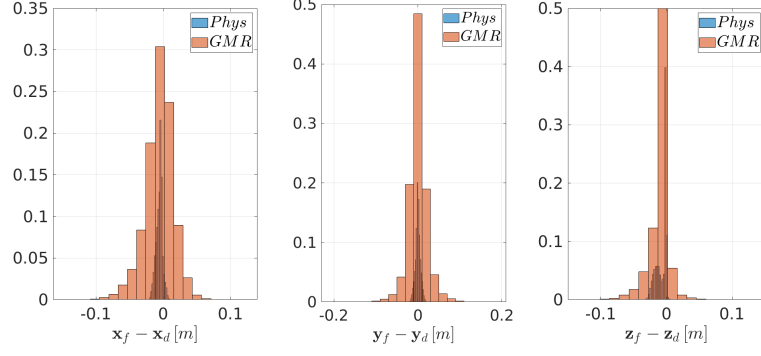
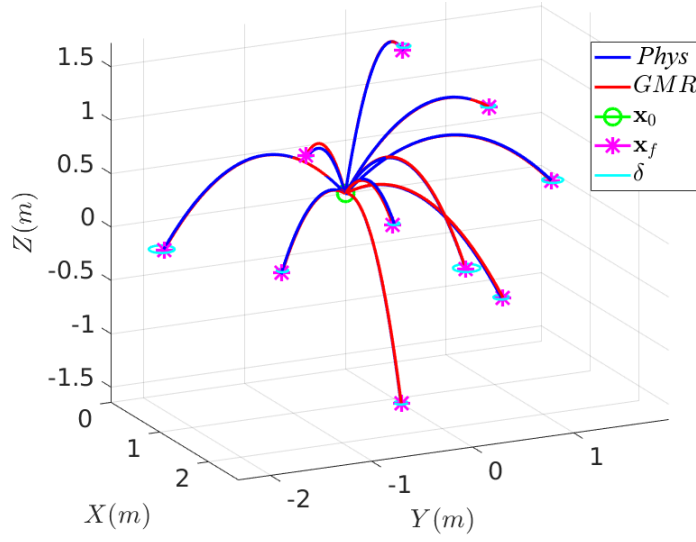
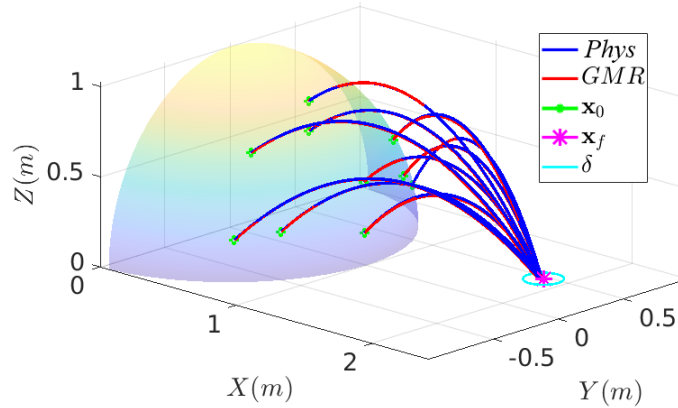


Figure 6.7: Validation errors for desired landing positions obtained using learned throwing parameters and ground truth parameters: errors in x (left), in y (middle), and in z (right)



(a)



(b)

Figure 6.8: Example of ten trajectories and landing positions for throwing tasks using learned throwing parameters and those obtained using physics: left: different desired landing positions from the same release position at the origin; right: different release positions for the same desired landing position

6.3.2 OPTIMAL FEASIBLE RELEASE STATE: CONCEPT

To compute the optimal feasible release state, our main idea is to define a throwing task-related cost function denoted as $l(\mathbf{q}, \dot{\mathbf{q}})$, and then compute its optimizer in terms of joint position and velocity $(\mathbf{q}^*, \dot{\mathbf{q}}^*)$ that satisfy the task constraints. The task-space release state is obtained through forward kinematics as $\mathbf{s}_r^* = (\mathbf{x}_r(\mathbf{q}^*), \dot{\mathbf{x}}_r(\mathbf{q}^*))$ with

$$\mathbf{q}^*, \dot{\mathbf{q}}^* = \underset{\mathbf{q}, \dot{\mathbf{q}}}{\operatorname{argmin}} l(\mathbf{q}, \dot{\mathbf{q}}) \quad (6.3.6)$$

$$s.t. \quad \mathbf{J}_c(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0} \quad (6.3.7)$$

$$\mathbf{q}_{\min} \leq \mathbf{q} \leq \mathbf{q}_{\max} \quad (6.3.8)$$

$$|\dot{\mathbf{q}}| \leq \dot{\mathbf{q}}_{\max} \quad (6.3.9)$$

where Eq. (6.3.7), (6.3.8) and (6.3.9) enforce the feasibility constraints, for the grasp, joint position and joint velocity, respectively.

To define the overall objective function $l(\mathbf{q}, \dot{\mathbf{q}})$, we use the previously learned throwing map $\mathbf{v}_r(\mathbf{x}_l^d, \mathbf{x}_r(\mathbf{q})) \in \mathbb{R}^3$ that computes the throwing velocity from the relative position between the release position $\mathbf{x}_r(\mathbf{q})$ and the desired landing position \mathbf{x}_l^d . We aim at achieving throwing tasks with minimal release speed as possible.

Based on the feasible set definition (6.2.2), we define the cost function as

$$\begin{aligned} l(\mathbf{q}, \dot{\mathbf{q}}) = & \|\mathbf{v}_r(\mathbf{q})\|_{w_p}^2 + \|\mathbf{J}_o(\mathbf{q})\dot{\mathbf{q}} - \mathbf{v}_r(\mathbf{q})\|_{w_v}^2 \\ & + \|\boldsymbol{\mu}\theta(\mathbf{R}_O(\mathbf{q}_k), \mathbf{R}_O^d)\|_{w_o}^2 \end{aligned} \quad (6.3.10)$$

Thus, minimizing the first sub-objective in Eq. (6.3.10) amounts to seeking joint configurations that yield minimal release velocity; minimizing the second sub-objective produces (for configuration \mathbf{q}) joint velocities $\dot{\mathbf{q}}$, whose associated task-space velocity approaches the desired throwing velocity ($\mathbf{J}_{vo}(\mathbf{q})\dot{\mathbf{q}} = \dot{\mathbf{x}}_r(\mathbf{q}) \rightarrow \mathbf{v}_r(\mathbf{q})$). The third sub-objective helps resolve the redundancy by specifying desired end-effector orientation during the throwing task.

6.3.3 OPTIMAL FEASIBLE RELEASE STATE: SOLUTION

Optimization of Eq. (6.3.6) with the cost function in Eq. (6.3.10) is nonlinear in terms of joint configuration and does not have a closed-form solution. Thus, we propose to solve it iteratively using sequential quadratic programming (Boggs and Tolle, 1995) with the joint acceleration as the decision variable. Such an approach allows simultaneous updating of both the position and velocity subjected

to their respective constraints as follows

$$\begin{aligned}\mathbf{q}^* &\leftarrow \mathbf{q}_{k+1} = \mathbf{q}_k + \delta t \dot{\mathbf{q}}_k + (\delta t)^2 \ddot{\mathbf{q}}^* \\ \dot{\mathbf{q}}^* &\leftarrow \dot{\mathbf{q}}_{k+1} = \dot{\mathbf{q}}_k + \delta t \ddot{\mathbf{q}}^* \\ &\text{until convergence}\end{aligned}\tag{6.3.11}$$

where δt is the time step and represents the step length of the algorithm; $\ddot{\mathbf{q}}^*$ is the optimal acceleration at the k^{th} iteration and provides the stepping direction.

To compute the optimal acceleration $\ddot{\mathbf{q}}^*$, we reformulate the problem as a bilevel optimization problem (Colson et al., 2007); at the top level, we compute the feasible release velocity $\dot{\mathbf{x}}_r^f$ for the configuration \mathbf{q}_k as

$$\dot{\mathbf{x}}_r^f = \mathbf{J}_o(\mathbf{q}_k) \dot{\mathbf{q}}_k^f \quad \text{with} \tag{6.3.12}$$

$$\dot{\mathbf{q}}_k^f = \underset{\dot{\mathbf{q}}}{\operatorname{argmin}} \quad \|\mathbf{J}_o(\mathbf{q}_k) \dot{\mathbf{q}} - \mathbf{v}_r(\mathbf{q}_k)\|^2 \tag{6.3.13}$$

$$s.t. \quad |\dot{\mathbf{q}}| \leq \dot{\mathbf{q}}_{\max}$$

The velocity $\dot{\mathbf{x}}_r^f$ represents the closest throwing velocity to $\mathbf{v}_r(\mathbf{q}_k)$ that the robot can achieve from configuration \mathbf{q}_k , with $\dot{\mathbf{q}}_k^f$ as its corresponding joint-space value.

At the bottom level of the optimization, we solve for the acceleration $\ddot{\mathbf{q}}_k^*$ that minimizes: the difference between $\dot{\mathbf{x}}_r^f$ and $\mathbf{v}_r(\mathbf{q}_k)$, the module of $\mathbf{v}_r(\mathbf{q}_k)$ and the difference between the current and desired end-effector orientation ($\mathbf{R}_O(\mathbf{q})$ and \mathbf{R}_O^d). Thus, following Eq. (6.3.6), the bottom level problem is reformulated as

$$\ddot{\mathbf{q}}_k^* = \underset{\ddot{\mathbf{q}}}{\operatorname{argmin}} \quad \sum_i l'_i(\mathbf{q}_k, \dot{\mathbf{q}}_k, \ddot{\mathbf{q}}_k) \quad \text{with} \quad i = \{p, O\} \tag{6.3.14}$$

$$s.t. \quad \mathbf{J}_c(\mathbf{q}_k) \ddot{\mathbf{q}}_k + \mathbf{J}_c(\mathbf{q}_k) \dot{\mathbf{q}}_k = \mathbf{0} \tag{6.3.15}$$

$$\mathbf{q}_{\min} \leq \mathbf{q}_k + \delta t \dot{\mathbf{q}}_k + (\delta t)^2 \ddot{\mathbf{q}}_k \leq \mathbf{q}_{\max} \tag{6.3.16}$$

$$\dot{\mathbf{q}}_{\min} \leq \dot{\mathbf{q}}_k + \delta t \ddot{\mathbf{q}}_k \leq \dot{\mathbf{q}}_{\max} \tag{6.3.17}$$

$$\mathbf{q}_{\min} \leq \mathbf{q}_k + \delta t \dot{\mathbf{q}}_k + (\delta t)^2 \ddot{\mathbf{q}}_k \pm \Delta \mathbf{q} \leq \mathbf{q}_{\max} \tag{6.3.18}$$

where Eqs. (6.3.15)-(6.3.17) enforce the feasibility constraints. The constraint (6.3.18) ensures that the joint has motion range $\Delta \mathbf{q}$ to accelerate from zero to $\dot{\mathbf{q}}_k^f$. $\Delta \mathbf{q}$ for each joint is expressed as

$$\Delta q_j = \frac{(\dot{q}_j^f)^2}{2\ddot{q}_{j,\max}} \tag{6.3.19}$$

The term $l'_i(\mathbf{q}_k, \dot{\mathbf{q}}_k, \ddot{\mathbf{q}}_k)$ represents the redefined sub-objectives at the acceleration level. Exploiting the kinematics of the robots, we propose to formulate $l'_i(\mathbf{q}_k, \dot{\mathbf{q}}_k, \ddot{\mathbf{q}}_k)$ as in an acceleration-based inverse kinematics such that

$$l'_i(\mathbf{q}_k, \dot{\mathbf{q}}_k, \ddot{\mathbf{q}}_k) \triangleq \|\mathbf{J}_i(\mathbf{q}_k)\ddot{\mathbf{q}}_k - (\ddot{\mathbf{x}}_i^d - \dot{\mathbf{J}}_i(\mathbf{q}_k)\dot{\mathbf{q}}_k)\|_{\mathbf{w}_i}^2 \quad (6.3.20)$$

with $i = \{p, O\}$ where p and O relate to position and orientation, respectively. $\mathbf{J}_p(\mathbf{q})$ and $\mathbf{J}_O(\mathbf{q}) = \mathbf{J}_\omega(\mathbf{q})$ are the Jacobian matrices of the linear and angular velocity of the end-effectors, respectively. $\ddot{\mathbf{x}}_i^d$ represents the desired task accelerations, designed by defining error functions associated with each sub-objective of Eq. (6.3.10) forcing the dynamics of these error functions to exponentially decrease toward zero to achieve minimization. For instance, to design $\ddot{\mathbf{x}}_O^d$, we defined an orientation error function between the current and desired end-effector ($\mathbf{R}_O(\mathbf{q}_k)$ and \mathbf{R}_O^d) using axis-angle representation of the relative orientation.

We designed the dynamics of this error such that it exponentially converges toward zero using a stable proportional-derivative (PD) control law. We designed $\ddot{\mathbf{x}}_p^d$ as a convex combination of two task components associated with the throwing velocity as

$$\ddot{\mathbf{x}}_p^d = \gamma \ddot{\mathbf{x}}_{\text{ivr}}^d + (1 - \gamma) \ddot{\mathbf{x}}_{\text{mvr}}^d \quad (6.3.21)$$

where $\ddot{\mathbf{x}}_{\text{ivr}}^d \in \mathbb{R}^3$ and $\ddot{\mathbf{x}}_{\text{mvr}}^d \in \mathbb{R}^3$ represent task accelerations that provide directions toward feasibility of release velocity and minimum throwing velocity, respectively; γ is a scalar weighting their relative importance. We propose to define $\ddot{\mathbf{x}}_{\text{ivr}}^d$ and $\ddot{\mathbf{x}}_{\text{mvr}}^d$ as

$$\begin{aligned} \ddot{\mathbf{x}}_{\text{ivr}}^d &= -K_p(\dot{\mathbf{x}}_r^f - \mathbf{v}_r(\mathbf{q})) - K_d\mathbf{J}_p(\mathbf{q})\dot{\mathbf{q}} \\ \ddot{\mathbf{x}}_{\text{mvr}}^d &= -K_d(\mathbf{J}_p(\mathbf{q})\dot{\mathbf{q}} + K_p\mathbf{J}_r^{-1}(\mathbf{x})\mathbf{v}_r(\mathbf{q})) \end{aligned}$$

where $K_p \in \mathbb{R}^{3 \times 3}$ and $K_d \in \mathbb{R}^{3 \times 3}$ are positive definite gain matrices, and $\mathbf{J}_r(\mathbf{x}) = \frac{\partial \mathbf{v}_r(\mathbf{x})}{\partial \mathbf{x}} \in \mathbb{R}^{3 \times 3}$ is the Jacobian of the throwing velocity with respect to the release position.

We mainly consider three stopping conditions for the proposed iterative algorithm. The algorithm stops as soon as a feasible trajectory is found, with a feasible state whose predicted landing position is within a predefined tolerance of the desired landing position. The algorithm stops when the objective function reaches a plateau for number of iterations. The algorithm also stops when the maximum number of iterations is reached. The task-space release state $\mathbf{s}_r^* = (\mathbf{x}_r(\mathbf{q}^*), \dot{\mathbf{x}}_r(\mathbf{q}^*))$ is obtained through forward kinematics of the joint-space state. Algorithm 1 summarizes the proposed approach.

6.3.4 GENERATION OF BEST FEASIBLE RELEASE STATES

Once the inverse throwing map is validated and its corresponding Jacobian is obtained following Eq. (6.3.3), we can solve the optimization in Eq. (6.3.15) for the best feasible release configurations of the dual-arm robot for tossing

Algorithm 1: computing optimal feasible throwing state

```

1 Input:  $\mathbf{x}_f^d$ ,  $\mathbf{R}_O^d$ ,  $\mathbf{G}_O$ ,  $\mathbf{q}_{min}$ ,  $\mathbf{q}_{max}$ ,  $\dot{\mathbf{q}}_{max}$ ,  $\mathbf{v}_r(\mathbf{x}_f^d, \mathbf{x}_r(\mathbf{q}))$ 
2 Output : desired release state  $\mathbf{q}^*, \dot{\mathbf{q}}^*$  and  $\mathbf{s}_r^* = (\mathbf{x}_r(\mathbf{q}^*), \dot{\mathbf{x}}_r(\mathbf{q}^*, \dot{\mathbf{q}}^f))$ 
3 Initialization:  $\mathbf{q}_k \leftarrow \mathbf{q}_0$ ,  $\dot{\mathbf{q}}_k \leftarrow \mathbf{0}$ ,  $\ddot{\mathbf{q}}_k \leftarrow \mathbf{0}$ ,  $iter = 0$ ;
4 |
5 if ( $\mathbf{x}_f^d$  is reachable)
6 |   while (true) do
7 |     1: update model:  $\mathbf{x}_r(\mathbf{q}_k)$ ,  $\mathbf{J}_i(\mathbf{q}_k)$ ,  $\dot{\mathbf{J}}_i(\mathbf{q}_k)$  with  $i = \{p, O\}$ 
8 |     2: compute  $\bar{\mathbf{x}} = \mathbf{x}_f^d - \mathbf{x}_r(\mathbf{q}_k)$ ,  $\mathbf{v}_r(\bar{\mathbf{x}})$ ,  $\mathbf{J}_r(\bar{\mathbf{x}})$ 
9 |     3: compute feasible velocity:  $\dot{\mathbf{q}}_k^f$  and  $\dot{\mathbf{x}}_r^f(\mathbf{q}_k)$  according to (6.3.12)
10 |    4: compute task acceleration:  $\ddot{\mathbf{x}}_i^d$  with  $i = \{p, O\}$  (6.3.21)
11 |    5: compute  $\ddot{\mathbf{q}}_k^*$ : solve optimization (6.3.14)
12 |    6: update state:  $\mathbf{q}_k, \dot{\mathbf{q}}_k \leftarrow$  (6.3.11)
13 |    7: compute cost function:  $\sum l'_i(\mathbf{q}_k, \dot{\mathbf{q}}_k)$ 
14 |
15 |    if ( $(|\Delta l'_i(\mathbf{q}_k, \dot{\mathbf{q}}_k)| \approx 0, \forall k = \{k \dots k + n_{plateau}\})$ 
16 |      or ( $iter \geq iter_{max}$ ))
17 |         $\mathbf{q}^* \leftarrow \mathbf{q}_k$ ,  $\dot{\mathbf{q}}^* \leftarrow \dot{\mathbf{q}}_k^f$ 
18 |         $\mathbf{s}_r^* = (\mathbf{x}_r(\mathbf{q}^*), \dot{\mathbf{x}}_r(\mathbf{q}^*, \dot{\mathbf{q}}^f))$ 
19 |        break;
20 |      end if
21 |       $iter = iter + 1$ 
22 end if

```

the object. Figure 6.9 shows ten release configurations and the corresponding free-flying trajectories of the object to the desired landing position in task-space.

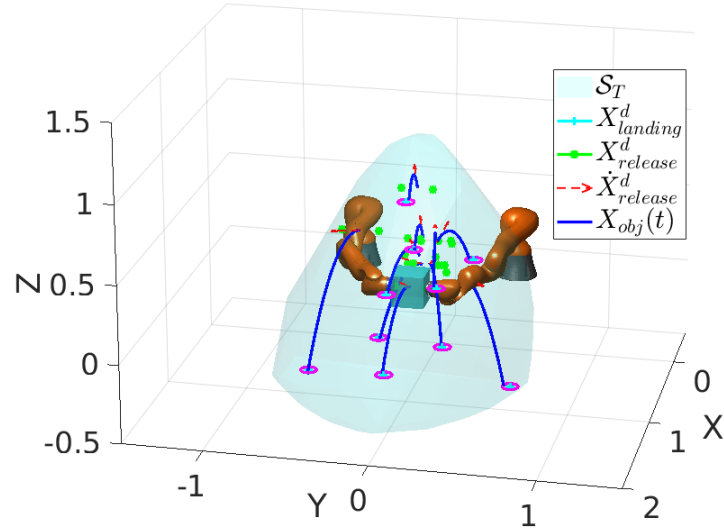


Figure 6.9: Illustration of ten computed feasible release configurations in 3D (green dot) and corresponding object trajectories (blue) to the desired landing position (cyan points). The magenta circle represents a tolerance radius of 0.05 m around the landing position. The red arrows indicate the directions of the 3D release velocities.

The corresponding joint-space release configurations for position and velocity are shown in Figure 6.10. It is observed that the computed configurations are within the robot joint limits, and thus are kinematically feasible, with some

joints reaching their maximum velocities.

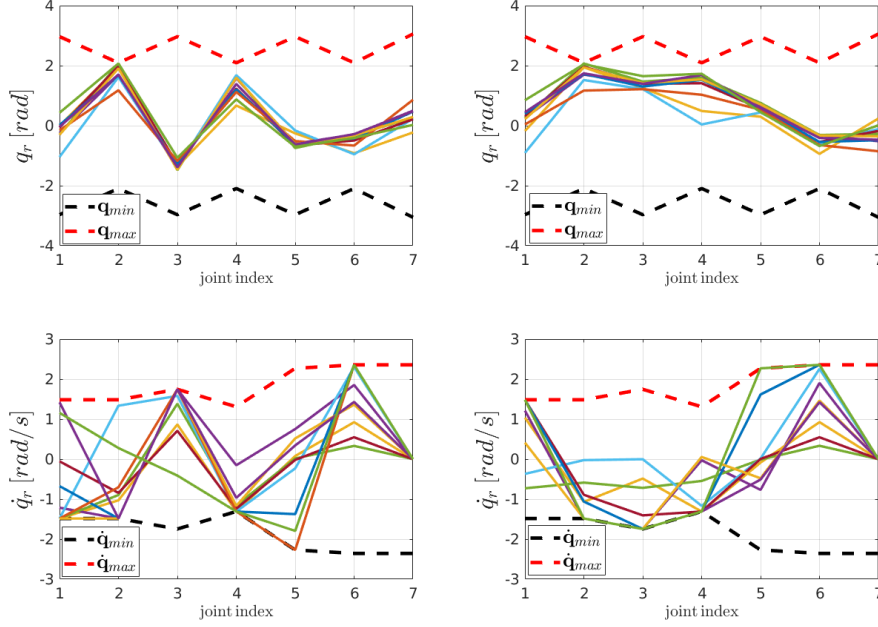


Figure 6.10: Illustration of joint-space configurations corresponding to ten computed feasible release states: top: joint positions for left and right robotic arms; bottom: joint velocities for left and right arms. The joint limits for position and velocity are shown in red and black dashed lines, respectively.

6.4 Learning Tossable Space

In this section, we address modeling of the tossable workspace of the dual-arm robotic system. The tossable workspace is the set of all positions reachable by an object if thrown by the robot. We propose to approximate it by modeling the distribution of the reachable positions. Such modeling provides a probability map of the reachability of possible landing positions. It is useful for selecting the best interception positions, chosen as points that represent a high probability of success.

We sampled the forward half space⁵ of the robot by randomly generating 10^5 uniformly distributed 3D positions within a radius of $0.2 - 1.75\text{ m}$ and within a cone of $(\pm \frac{\pi}{3}\text{ rad})$ around the x axis, as shown in Figure 6.11. We determined the throwing-reachability of the generated points using the optimization in Eq. (6.3.15), which checks for each point whether or not the robot admits a feasible release state. We modeled the probability distribution of the obtained feasible points using a GMM. As with the previously learned projectile dynamics, the

⁵without loss of generality, we only consider the space beyond the forward half plane of the robot's workspace

number of Gaussian functions, here 13 was determined using the BIC, and the model was trained with the expectation–maximization algorithm initialized with k-means. The likelihood contours are shown in Figure 6.12.

Thus, a given target position \mathbf{x}^t is considered in the robot tossable space if its likelihood (or probability density function) expressed as

$$\mathcal{P}(\mathbf{x}^t | \mathcal{M}_{toss}) = \sum_{k=1}^{K_t} \pi_k \mathcal{N}(\mathbf{x}^t | \mu_k, \Sigma_k) \quad (6.4.1)$$

exceeds a threshold δ_{toss} . We chose this likelihood threshold such that it yields 99% prediction accuracy on the training set of feasible configurations (we obtained a $\delta_{toss} = 0.15$).

Once the tossable space is learned, it is used to determine the interception point. Valid interception points are restricted to the region defined by the intersection of the robot tossable space and the target path, mainly determined by the conveyor belt. We described the target through a set of points and predicted their future positions along the conveyor belt. We chose the best interception location as the position along the target trajectory with the set of points yielding the highest tossable likelihood.

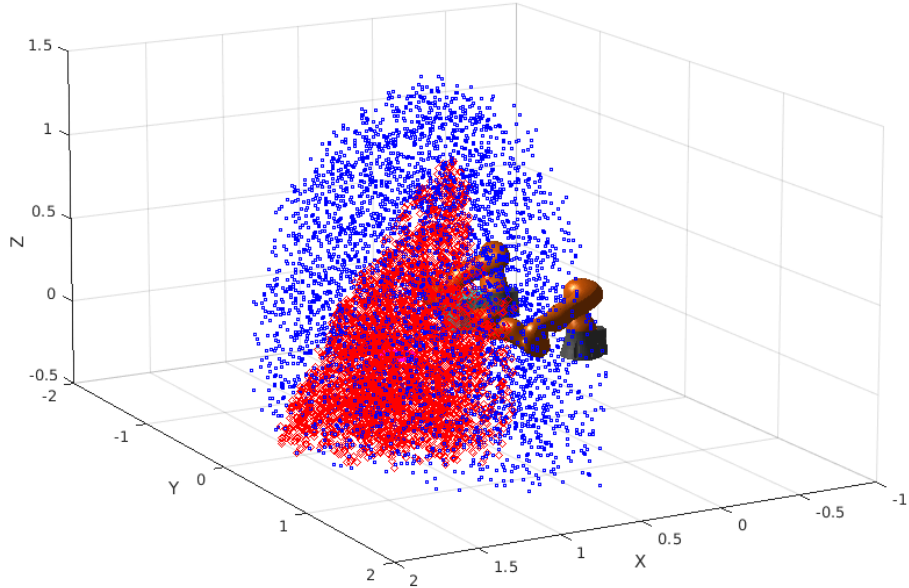


Figure 6.11: Example of 4000 samples from the generated data set to learn the tossable space of the dual-arm robot. The red points are those that admit feasible solutions, whereas the blue points do not according to our algorithm

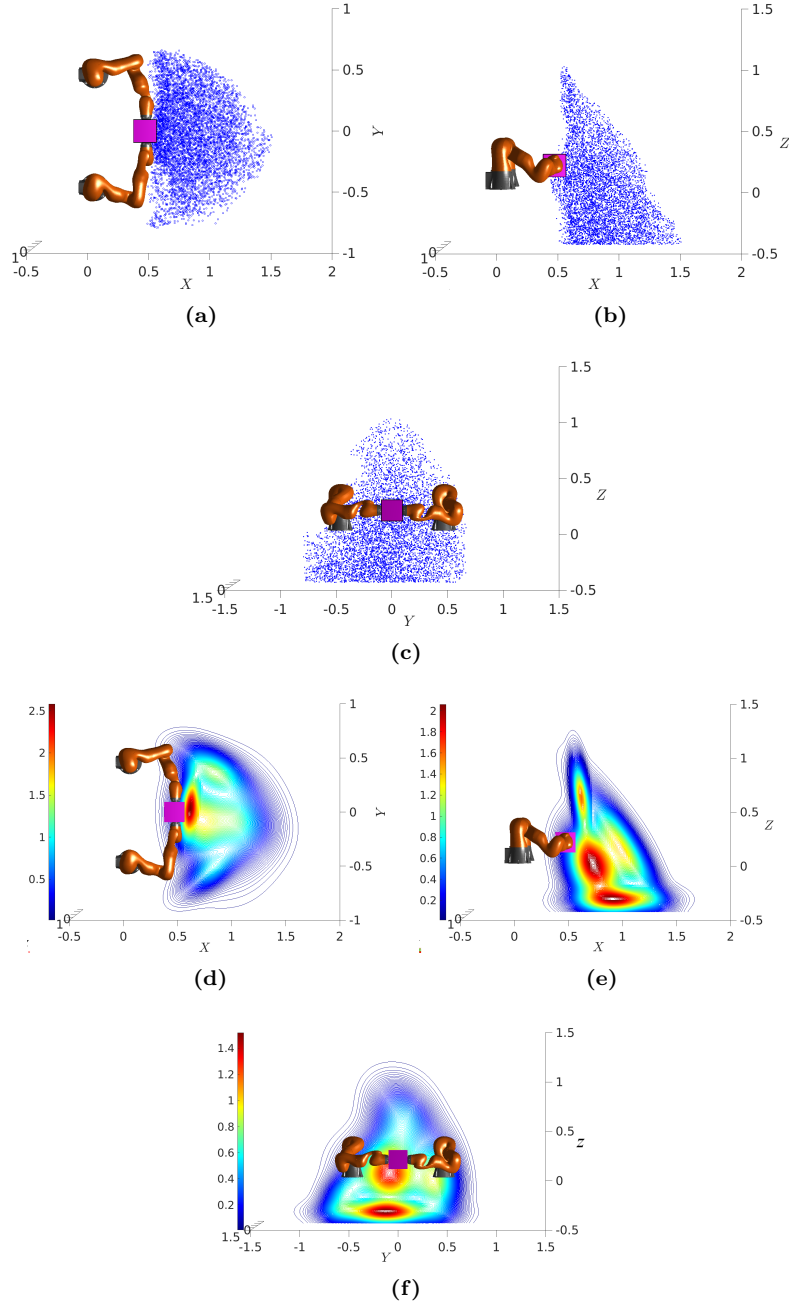


Figure 6.12: Representation of 2D projections of 3D tossable positions (top) and corresponding likelihood contours obtained by modeling the distribution of tossable points using a GMM with 13 Gaussian functions: (a)–(d): projection on XY plane; (b)–(e): projection on YZ plane; (c)–(f): projection on YZ plane. The likelihood of finding tossable states is lowest in blue regions and highest in red regions. The spread of these regions is not uniform across the workspace, and not symmetric with respect to the two robots (in our configuration, the second and sixth joints are not symmetric between the robots), as they are dependent on the highly nonlinear joint configurations for position and velocity.

6.5 Dual-arm Throwing Task Control

Once the interception point is selected and a feasible release configuration is found, the next step is to generate the motion of the robot to execute the throwing task. We controlled the coordinated motion of the dual-arm robot using our Modulated DS-based controller proposed in the previous Chapter, which allows us to leverage the kinetic energy of the robot through quick grabbing and tossing of the object. In this way, we ensure the dynamic feasibility of the task, which requires that the robot, from its current state with its inertia and that of the grasped object, can accelerate quickly enough to reach the desired release state while remaining within its hardware limits. Thus, we avoid conditions in which the robot holds the object and waits before tossing to intercept the target.

6.5.1 DUAL-ARM COORDINATED CONTROL

The desired motion of the dual-arm robot in the control framework presented in Chapter 5 can be written as

$$\dot{\mathbf{x}} = M(\mathbf{x})f_n(\mathbf{x}) + f_g(\mathbf{x}) \quad (6.5.1)$$

where $\mathbf{x} = \begin{bmatrix} \mathbf{x}^L \\ \mathbf{x}^R \end{bmatrix} \in \mathbb{R}^6$ is the state vector of the DS, with \mathbf{x}^L and \mathbf{x}^R representing the position of the left and right robot arms, respectively, in the dual-arm system. $f_n(\mathbf{x}) \in \mathbb{R}^6$ is the nominal DS, and $M(\mathbf{x}) \in \mathbb{R}^{6 \times 6}$ is the state-dependent modulation matrix that locally shapes the motion generated by $f_n(\mathbf{x})$. The $f_g(\mathbf{x})$ term represents the equivalent grasping force projected in the motion space.

In that framework, we proposed a tossing task that consisted of releasing a grabbed object at a desired position with a desired velocity by generating appropriate absolute and relative velocities ($\dot{\mathbf{x}}_d^{abs}$ and $\dot{\mathbf{x}}_d^{rel}$) for the dual-arm system (see Section 5.3 for more details).

However, regardless of the control strategy, the problem of when to release the target for successful interception must be addressed. To that end, we propose to determine a state of the target that should trigger the robot's movement (reach, pick and toss). We will refer to such a state as the target's "state-to-go".

6.5.2 ESTIMATION OF THE TARGET'S STATE-TO-GO

We assume that the target and thrown object intercept at a position \mathbf{x}_I^* and time t_* . Thus,

$$\mathbf{x}^t(t_*) = \mathbf{x}^o(t_*) = \mathbf{x}_I^* \quad (6.5.2)$$

where \mathbf{x}^t denotes the target's position, and \mathbf{x}^o denotes the object position. From the robot-object perspective t_* is expressed as the sum of: 1) the duration of

the robot motion from its starting time up to the release time (t_r) of the object, and 2) the free-flying time of the object. Hence

$$t_* = t_r + \Delta t_{ff} \quad (6.5.3)$$

where Δt_{ff} is the object's free flying duration from the release time t_r to the landing. We can also express t_* as function of the robot's path to the release position and the robot's average speed along that path as

$$t_* = \frac{L(\mathbf{x}(t_r), \mathbf{x}(t_0))}{\bar{v}} + \Delta t_{ff} \quad (6.5.4)$$

where \mathbf{x}^r is the robot position⁶, and $L(\mathbf{x}^r(t_r), \mathbf{x}^r(t_0))$ denotes the robot path length from its position at time t_0 ($\mathbf{x}^r(t_0)$) to its position at the release time t_r ($\mathbf{x}^r(t_r)$), expressed as

$$L(\mathbf{x}(t_r), \mathbf{x}(t_0)) = \int_{t_0}^{t_r} \|\dot{\mathbf{x}}(t)\| dt \quad (6.5.5)$$

where $\|\dot{\mathbf{x}}^r(t)\|$ denotes the L^2 -norm of the robot's velocity. In Eq. (6.5.4), \bar{v}^r represents the average speed of the robot along its path ($\bar{v} = \frac{1}{t_r - t_0} \int_{t_0}^{t_r} \|\dot{\mathbf{x}}(t)\| dt$).

Given that the target moves along a rectilinear path (the conveyor belt), for the interception to ideally happen at \mathbf{x}_I^* with the robot's motion starting at the time t_0 , the target's *state-to-go* or the state corresponding to t_0 can be determined as

$$\mathbf{x}^t(t_0) = \mathbf{x}_I^* - \frac{\dot{\mathbf{x}}^t}{\|\dot{\mathbf{x}}^t\|} L(\mathbf{x}_I^*, \mathbf{x}^t(t_0)) \quad (6.5.6)$$

where $\frac{\dot{\mathbf{x}}^t}{\|\dot{\mathbf{x}}^t\|}$ accounts for the target's direction of motion. $L(\mathbf{x}_I^*, \mathbf{x}^t(t_0))$ denotes the path length (distance) covered by the target during the interval from t_0 to t_* , it is expressed as a function of the robot's motion as

$$L(\mathbf{x}_I^*, \mathbf{x}^t(t_0)) = \bar{v}^t \left[\frac{L(\mathbf{x}(t_r), \mathbf{x}(t_0))}{\bar{v}} + \Delta t_{ff} \right] \quad (6.5.7)$$

where $\bar{v}^t = \frac{1}{t_r - t_0} \int_{t_0}^{t_r} \|\dot{\mathbf{x}}^t(t)\| dt$ represents the average speed of the target along the path from $\mathbf{x}^t(t_0)$ to $\mathbf{x}^t(t_*) = \mathbf{x}_I^*$.

Eqs. (6.5.6) and (6.5.7) indicate that given the intercept point \mathbf{x}_I^* and the feasible release position $\mathbf{x}_r^r = \mathbf{x}^r(t_r)$, the target's state-to-go can be easily estimated with few assumptions. Indeed, what is left to be estimated is: 1) the average speed of the target \bar{v}^t , which can be approximated from velocity mea-

⁶for the dual-arm robot, this position refers to the absolute position of the two end-effectors.

surements over a time window; 2) the average speed of the robot \bar{v}^r , which can be approximated from the forward integration of the velocity norm of the stable DS that drives the robot from its current position $\mathbf{x}^r(t)$ until the release position \mathbf{x}_r^* is reached; 3) the robot's path length $L(\mathbf{x}^r(t_r), \mathbf{x}^r(t_0))$, which can also be approximated, as in 2), from the DS; 4) the object's free-flying duration Δt_{ff} , which can be estimated from forward integration of the projectile dynamics given the desired intercept location and the computed release state.

6.5.3 MOTION ADAPTATION

The target's state-to-go estimated in the previous section was based on the average speed of the target and the DS motion. It does not account for possible perturbation of the target motion (slowing down or speeding up). Moreover, errors in motion tracking and prediction (through forward integration) will create divergence between the computed and actual states of the robot. Consequently, the interception will not happen as planned.

To ensure successful completion of the task, we propose a twofold adaptation strategy with, first, a velocity modulation strategy and second, an attractor adaptation strategy.

VELOCITY MODULATION

The DS can be accelerated or decelerated at will by multiplying the function by a positive scalar. This does not affect the stability properties at the attractor. Thus, we can adapt the robot's DS-based velocity at run time to adapt to changes in the velocity of the moving target as follows:

$$\dot{\mathbf{x}}_d = \beta(\mathbf{x})M(\mathbf{x})f_n(\mathbf{x}) + f_g(\mathbf{x}) \quad (6.5.8)$$

where $\beta(\mathbf{x})$ is the adaptation factor, which is simply a state-depend scaling factor computed as

$$\beta(\mathbf{x}) = \frac{\bar{v}^t}{\bar{v}^r} \left| \frac{L(\mathbf{x}^r(t_r), \mathbf{x}^r(t))}{L(\mathbf{x}_I^*, \mathbf{x}^t(t)) - \bar{v}^t \cdot \Delta t_{ff}} \right| \quad (6.5.9)$$

with $\beta(\mathbf{x}) \|\dot{\mathbf{x}}_d\| \leq \|\dot{\mathbf{x}}\|_{max}$, such that the adapted robot velocity does not exceed the maximum allowable velocity. $\beta(\mathbf{x}) \geq 0$ to preserve the stability of the DS.

Although the velocity modulation strategy can slow down or speed up the velocity of the robot, it cannot, however, reverse the robot's motion direction. Such reversal may be useful, for instance to force the robot to retract to an initial position. The robots needs to quickly accelerate to toss at the desired throwing speed. It may not be able to do so, if the path is too short, as joint limits may be reached. we propose additionally an adaptation strategy that adapts the attractor of the nominal DS $f_n(\mathbf{x})$ when the target's velocity changes its direction. Hence, we define the attractor as

$$\mathbf{x}_* = \alpha(\mathbf{x}^t, \dot{\mathbf{x}}^t)\mathbf{x}_d + (1 - \alpha(\mathbf{x}^t, \dot{\mathbf{x}}^t))\mathbf{x}_{stb} \quad (6.5.10)$$

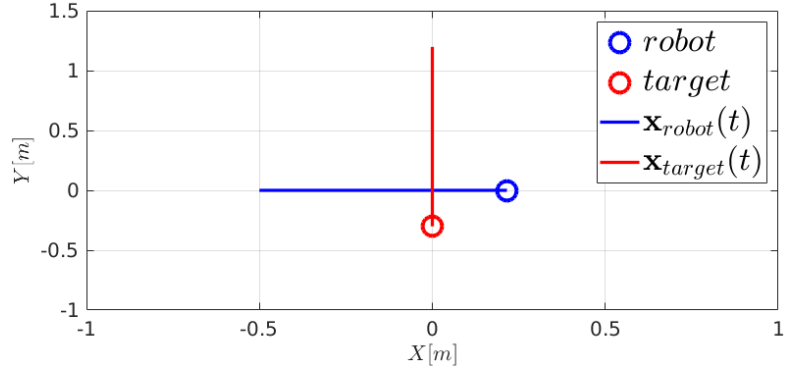
where \mathbf{x}_d is the desired attractor of the nominal DS, and \mathbf{x}_{stb} denotes a standby attractor to which the robot should retract to. $\alpha(\mathbf{x}^t, \dot{\mathbf{x}}^t) \in [0, 1]$ is a target's state-dependent scalar function that goes to 1 or 0 depending on whether the target moves in the direction of the interception or not. We defined $\alpha(\mathbf{x}^t, \dot{\mathbf{x}}^t)$ as

$$\alpha(\mathbf{x}^t, \dot{\mathbf{x}}^t) = \frac{1}{1 + e^{-a((\mathbf{x}_d^* - \mathbf{x}^t)^\top \dot{\mathbf{x}}^t)}} \quad (6.5.11)$$

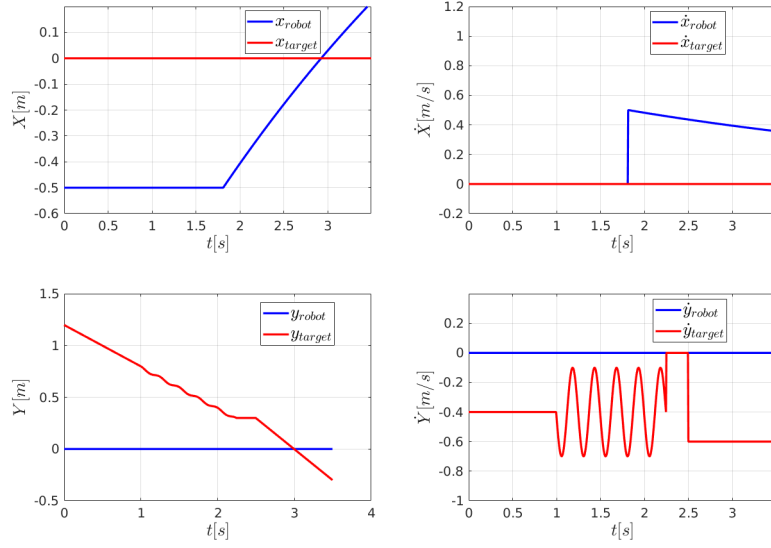
where $a > 0$ represents the steepness factor of the function defined by $\alpha(\mathbf{x}^t, \dot{\mathbf{x}}^t)$.

Example: To illustrate how the proposed adaptation strategy works, we simulate a simplified interception problem, where a robot moving along the \mathbf{x} -axis must intercept a target moving along the \mathbf{y} -axis. To successfully intercept the target, the robot follows the motion of the target. The initial velocity of the target is -0.4 m/s; after 1 s, the target velocity undergoes a sine wave perturbation with an amplitude of 0.3 m/s for 1.25 s and then the velocity goes at 0 m/s for 0.25 s before suddenly increasing to -0.6 m/s, as observed in the bottom-right of Figure 6.13-(b). Without adapting accordingly the velocity of the robot, interception fails, as can be seen in Figure 6.13-(a). The time evolution of the positions and velocities of the target and robot are shown in Figure 6.13-(b); Figure 6.14 shows the adaptation factor $\beta(\mathbf{x})$, which remains equal to 1.

When the robot velocity is adapted to compensate for changes in the target velocity, the robot successfully intercepts the target and they begin moving together. This is observed in Figure 6.15; the 2D motion is shown in Figure 6.15-(a) and the time-evolution of the positions and velocities are shown in Figure 6.15-(b). The adaptation factor, which now changes, is shown in Figure 6.16. Note that in both cases (without and with adaptation), the robot velocity remains initially at 0, and start changing only when the target reaches the estimated state-to-go.



(a)



(b)

Figure 6.13: Simulated simplified example of failed interception task when the target motion undergoes changes or perturbations and the robot motion is not adapted accordingly. (a): 2D positions of the target (red circle) and the robot (blue circle) with their trajectories represented in red and blue, respectively. (b): time-evolution of positions and velocities of the target and object in \mathbf{x} coordinates (top) and \mathbf{y} coordinates (bottom).

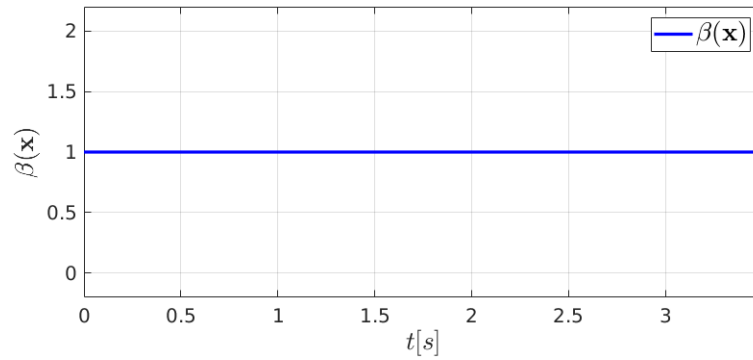
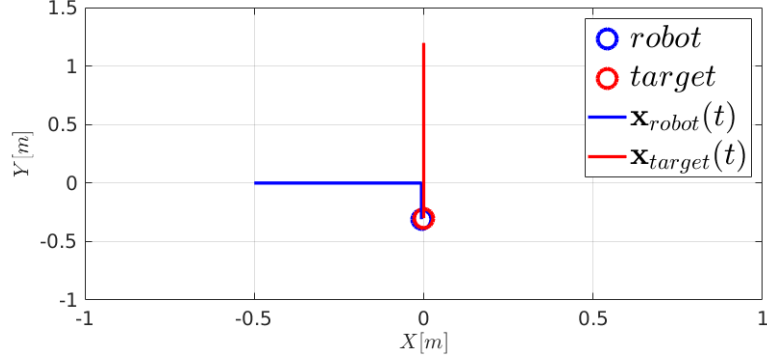
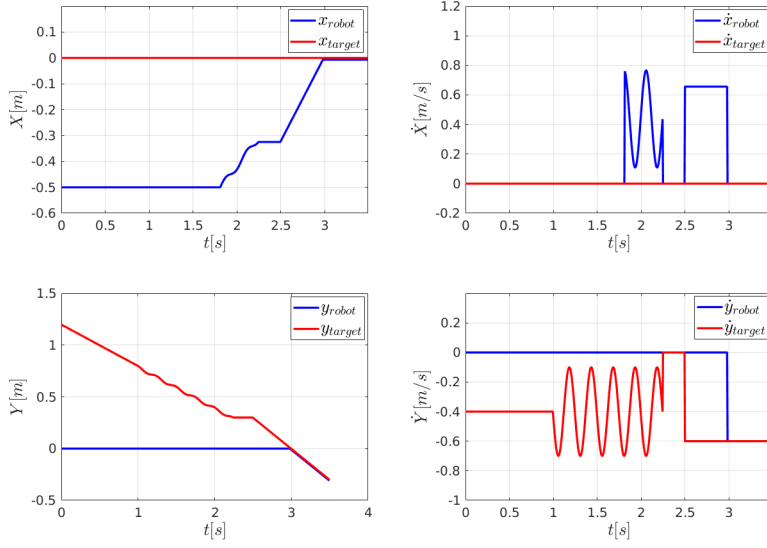


Figure 6.14: Time evolution of the adaptation factor $\beta(\mathbf{x})$, which remains equals to 1 when the adaptation is not active



(a)



(b)

Figure 6.15: Simulated simplified example of successful interception task when robot motion is adapted according to changes or perturbations in target motion. (a): 2D positions of target (red circle) and robot (blue circle) with their trajectories represented in red and blue, respectively. (b): time-evolution of positions and velocities of the target and object in \mathbf{x} coordinates (top) and \mathbf{y} coordinates (bottom).

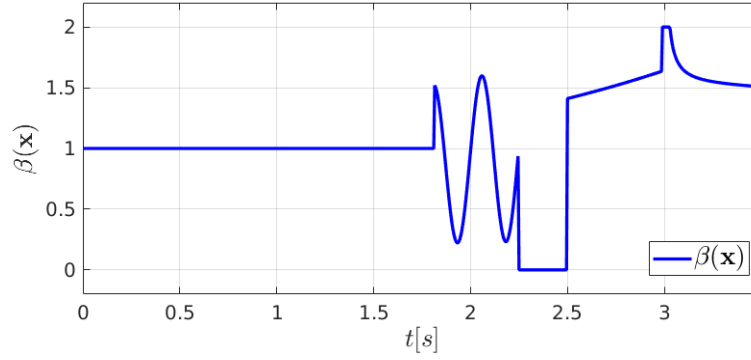


Figure 6.16: Time evolution of the adaptation factor $\beta(\mathbf{x})$, which start varying when the target reached the estimated state-to-go to compensate for changes in the target motion.

6.6 Empirical Validation

To validate the proposed approach, we used the same robotic setup as in our previous Chapter, a dual-arm system consisting of a pair of KUKA LBR IIWA7 and IIWA14 robots. To move the target, we used a conveyor belt with velocity ranging from 0.05 m/s to 1.5 m/s. The object and target position were measured using an *Optitrack* motion capture system; velocity information was estimated using a Savitzky-Golay (Savitzky and Golay, 1964) smoother and a Kalman filter. We used a 1.9-kg cubic box with dimensions of 0.26 m on each side. The target speed ranged from 0.10 m/s to 0.450 m/s, determined experimentally to avoid collision between the moving object after landing and a robot while retracting after releasing the object. Speeds of up to 0.65 m/s were reached with a smaller box (see video). We used a flat tray with dimensions of 0.40 m \times 0.30 m as the target. Figure 6.17 shows the robotic setup used for our experimental validation. In the implementation, the DS and the feasible release state are updated every 5 ms and 100 ms, respectively. We used qpOASES (Ferreau et al., 2014) to solve the optimizations (6.3.12) and (6.3.14). The whole feasibility algorithm is solved between 3 to 25 ms on a *Intel^(R) Core i7*, 3.4 GHz and 7.8 GB RAM PC.

The validation process was both simulated and conducted using an actual robot. Three main tasks were considered: (i) tossing an object to a target moving at a constant velocity; (ii) tossing an object to a target moving with a changing velocity; (iii) comparison of placing an object and tossing an object onto a moving target. The experiments can be visualized in a video provided as supplementary material. The implementation codes of the proposed approach are also made available (see Section 1.7).

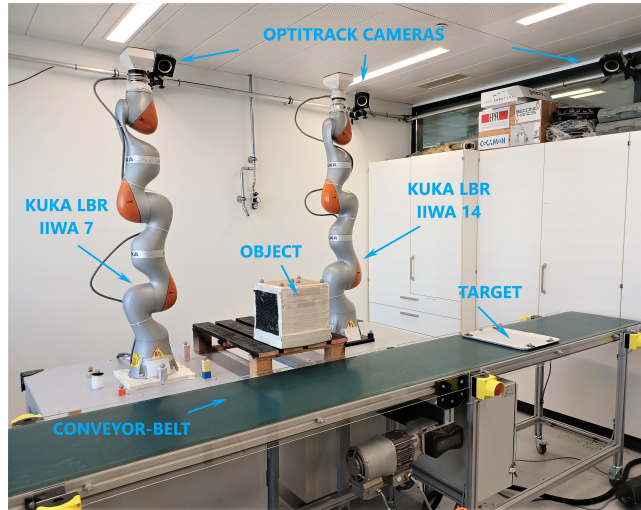
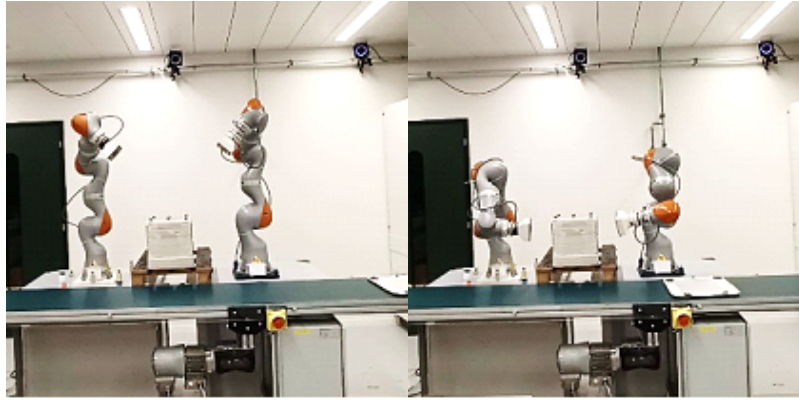
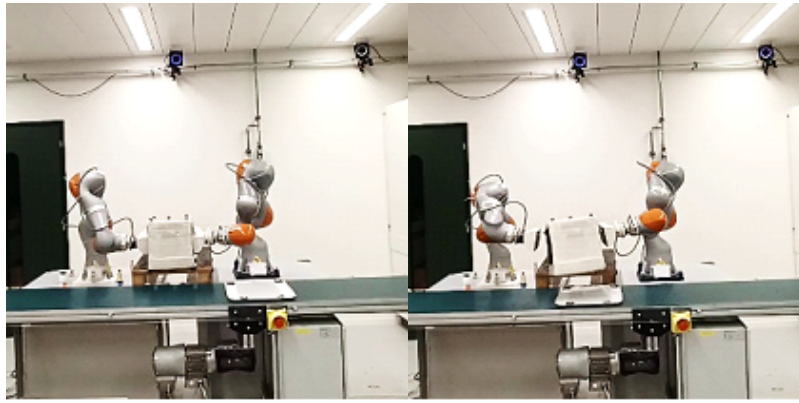


Figure 6.17: Experimental setup used for validating the proposed control scheme. The image shows the two KUKA robots, the conveyor belt, some Optitrack cameras, the target, and the object on a pallet.



(a)

(b)



(c)

(d)



(e)

(f)

Figure 6.18: Snapshots illustrating a dual-arm grabbing and tossing task of an object (white cubic box) on a moving target (white tray on the conveyor belt). At first, the robots wait for the target to reach the estimated state-to-go before starting moving towards the object to grab it and toss it on the target being moved by the conveyor belt. From left to right, the snapshots show the instant when the dual-arm system is waiting, when it is approaching the object, when the object is grabbed, when it is tossed, when it lands and intercepts the moving target, and when they both move together

6.6.1 TOSsing OBJECT TO TARGET MOVING AT CONSTANT VELOCITIES

The goal of this task is to evaluate the accuracy and repeatability of dual-arm based positioning through tossing of an object to a target that moves at different constant velocities. This task simulates a hypothetical depalletizing task on an already loaded conveyor belt, where the robot must place an object in available free space moving on the conveyor belt.

Initially, the robot remains stationary as the target approaches until the target reaches the state-to-go as determined in Eq. (6.5.6) in Section 6.5.2. Once reached, the dual-arm system moves, grabs the object, and throws it to the desired intercept position as shown in Figure 6.18; task sequences such as initiation of target movement and robot movement, throwing of the object, interception, and unified movement of the object and target are observed in the snapshots.

Corresponding plots of positions and velocities of the object and target are shown in Figure 6.19 (top) and (bottom), respectively. In Figure 6.19-(top), the

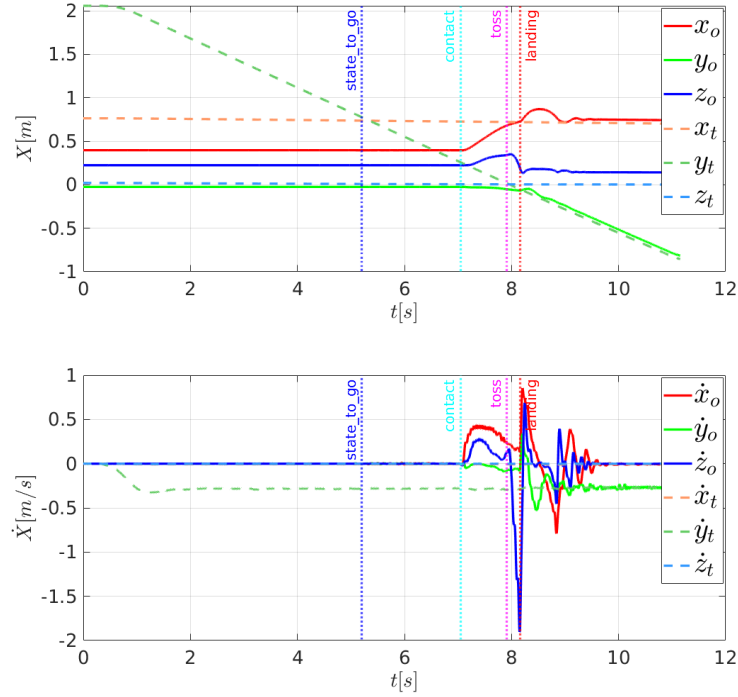


Figure 6.19: Position and velocity plots of object and target throughout the task. The target moves in the y-direction at a speed of -0.3 m/s (green dashed line in bottom plot); its starting position [0.75 m, 2.0 m, 0.25 m] decreases linearly in the y-direction (green dashed line in top plot); all other coordinates remain constant. The object starts moving only after contact with the robots.

initial position offset and correspondence between the object and target after landing are shown; the y-coordinates (representing the direction of conveyor belt movement) decrease continuously (negative speed). The constant offset of

the z-coordinate results from measurement of the object position at the center, producing a height offset with the target on the conveyor belt. Small oscillations result from bouncing of the object on the target caused by landing impact before stabilizing.

A 3D illustration is shown in Figure 6.20, including the trajectories described by the robotic system during the task, the object, and the target, indicated in red, blue, and green, respectively. The end-effector trajectories indicate complete cycles from standby positions and back after grabbing and tossing the object.

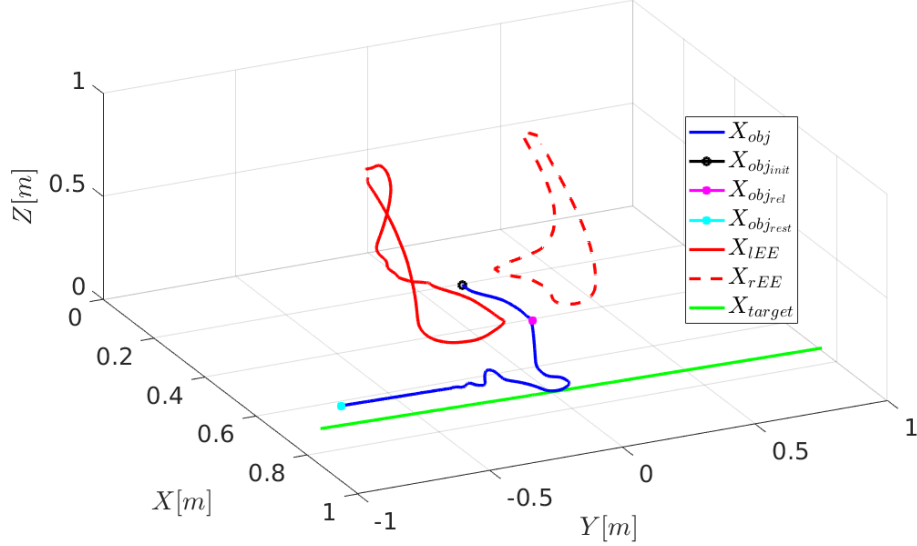


Figure 6.20: 3D trajectories of system in grabbing and tossing an object onto a moving target. The trajectories described by the end-effectors are shown in red (solid for left, dashed for right); the target trajectory is represented in green, the object trajectory in blue. The initial position of the object is shown in black, the release position in magenta. The rest position when the conveyor belt stops is shown in cyan.

The accuracy of the proposed scheme in intercepting moving targets is shown in Figure 6.21, which reports intercept position errors across the XY plane for ten experiments for each target speed. The norms of the errors per speed are shown in Figure 6.21-(a); the x and y components of the error contributing to the norm are shown in Figure 6.21-(b). The mean of the intercept error norm for each speed was less than 0.06 m, representing one-fifth of the target width (0.30 m). The variance in black indicates that few cases exceeded 0.06 m, remaining within a tolerance of 0.10 m. The 3D distribution of intercept positions defined by the target and object at release and landing is shown in Figure 6.22. Intercept locations are indicated with respect to feasibility and tossable workspace described in Sections 6.3 and 6.4.

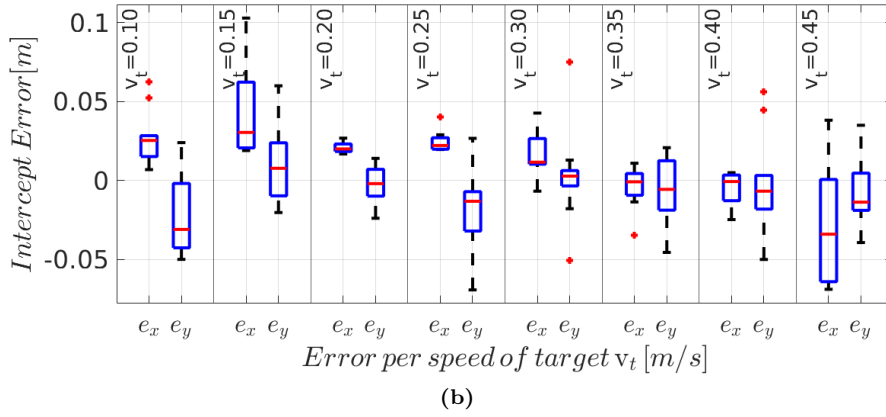
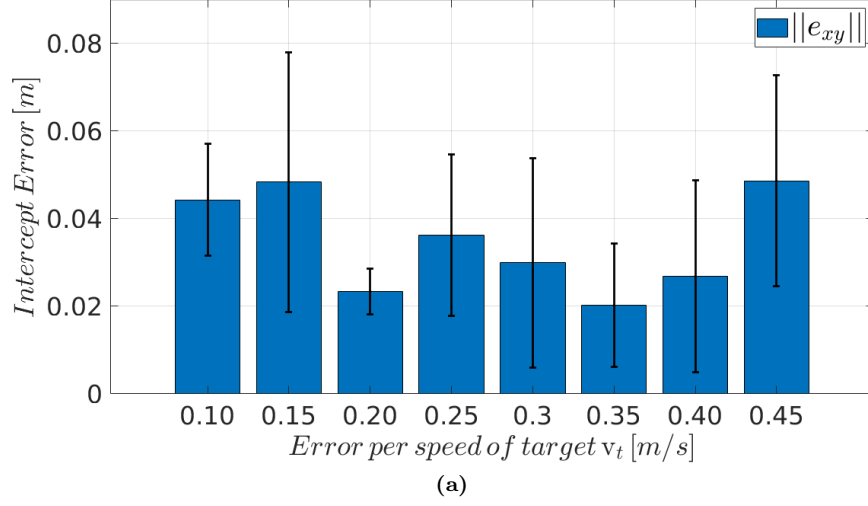


Figure 6.21: Intercept position error per speed across the XY plane (the z coordinate is not considered as the object height offset from its center is constant): (a) norm of intercept error per target speed; (b) intercept error per coordinate and per target speed. The intercept error for x-coordinates is denoted as e_x ; the error for y-coordinates is denoted as e_y

6.6.2 TOSING OBJECT ON TARGET MOVING WITH CHANGING VELOCITIES

The goal of this experiment is to assess the robustness and adaptivity of the proposed algorithm to changes in the target velocity. In other words, we evaluate how the proposed control strategy adapts the motion of the dual-arm system carrying the object for successful interception in presence of changes in the target motion.

To that end, while changing the velocity of the target, we started by testing the algorithm without adaptation ($\beta(\mathbf{x}) = 1$ in Eq. (6.5.8)) and then activated the adaptation scheme.

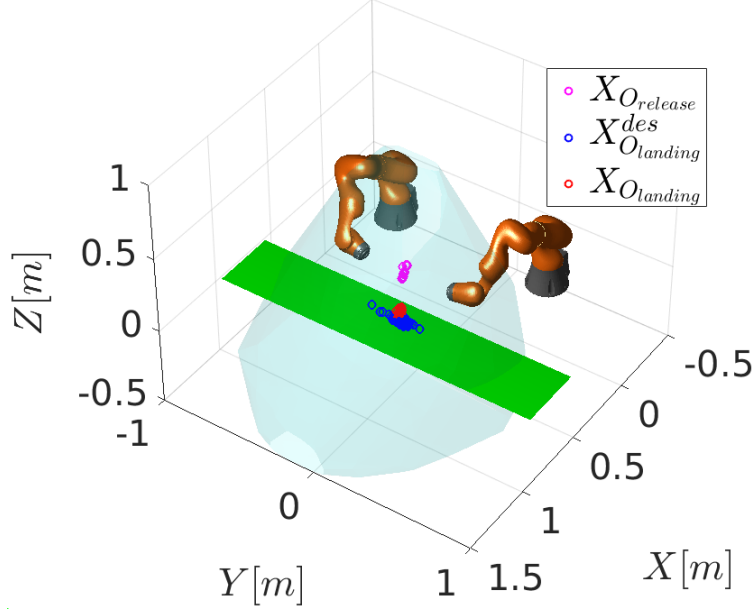


Figure 6.22: Relative 3D distribution of intercept positions defined by the target (blue), the object release position (magenta), and the object landing location on the conveyor belt (red) for the dual-arm system and its kinematically feasible tossing workspace (light blue)

The changing target speed was designed with a constant nominal component v_{nom}^t as in the previous experiment, and with a changing perturbation component v_{pert}^t , defined as $v_{pert}^t = a_{pert} \cdot \sin((\omega_p + 0.2 \cdot \text{rand}(\omega_p))t)$, where a_{pert} denotes the maximum amplitude of the perturbation; ω_p denotes its angular frequency, and t is the running time of the algorithm. We have conducted 20 experiments for each case; we set the following target speed parameters: $v_{nom}^t = 0.30$ m/s, $a_{pert} = 0.15$ m/s and $\omega_p = 2\pi$.

CASE WITHOUT ADAPTATION

Controlling robot motion based only on the state-to-go generally leads to failed interception of the target as soon as perturbation affects the system. Such a strategy amounts to open-loop control of interception, which only works if the conditions that predicted the state-to-go remain the same after the dual-arm system has initiated motion.

An example is illustrated in Figure 6.23. The X and Y position and velocity plots for the object and target are presented in Figure 6.23-(a). The norm of the linear velocities of the two robots triggered when the target reached the estimated state-to-go is shown in Figure 6.23-(b). The tossed object fails to intercept the target as the dual-arm motion generation ignores the changes in target speed.

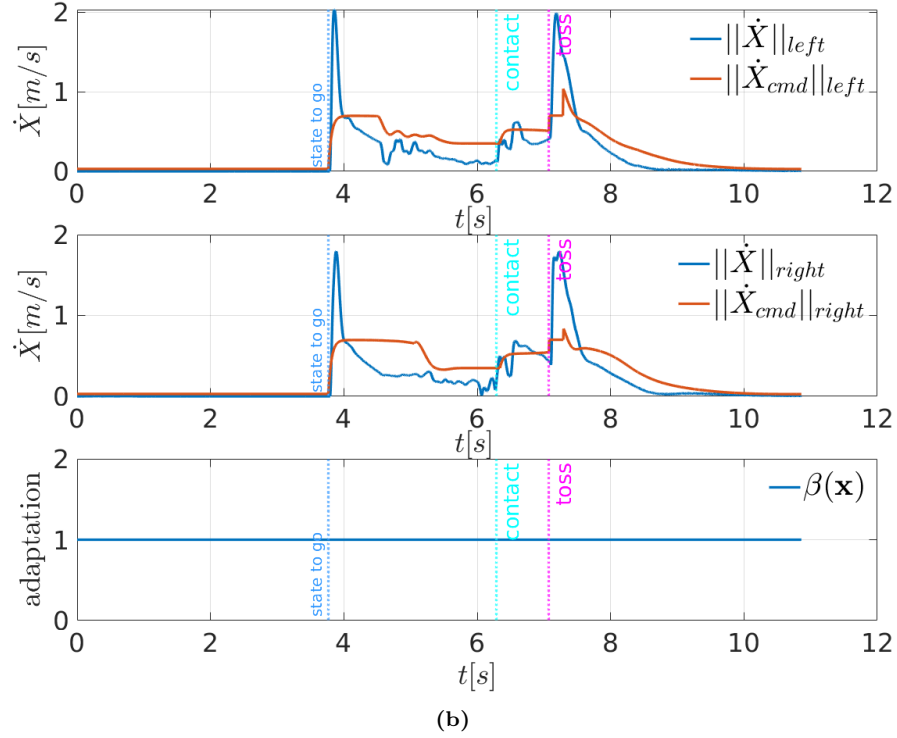
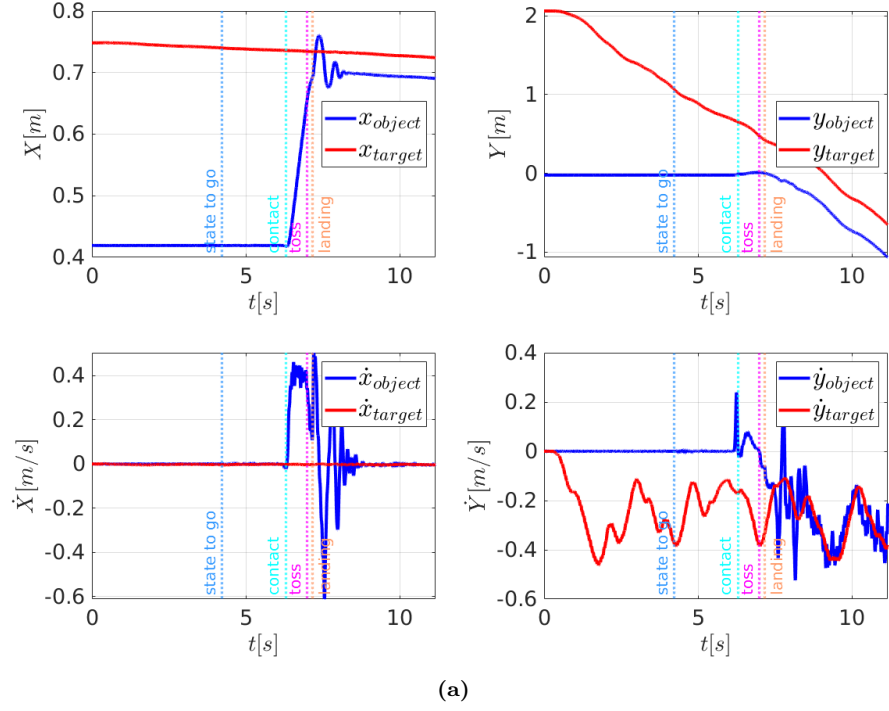


Figure 6.23: Position and velocity plots of dual-arm system grabbing and tossing an object onto a moving target with motion perturbation and **without** adaptation of robot motion: (a) X and Y evolution over time for position (top) and velocity (bottom) of the target (red) and object (blue); (b) norm of linear velocities of left robot (top), right robot (middle), and adaptation factor $\beta(\mathbf{x})$ (bottom). The robot motion remains unaffected by changes in target speed.

In this case, the adaptation scheme modulates the motion of the robot, slowing it down or speeding it up based on a continuously updated prediction of interception using current target and robot states. We evaluated adaptation with two types of perturbation. The first perturbation of target motion was similar to that in the case without adaptation; the second perturbation was caused by manually stopping, pulling back, or pushing forward the target as it moved on the conveyor belt.

Velocity-based perturbation

The position and velocity plots of the object and target are shown in Figure 6.24-(a). The object successfully intercepts the target despite changes in its speed. Velocity plots of the left and right robot end-effectors and the adaptation factor $\beta(\mathbf{x})$ are shown in Figure 6.24-(b), at the top, middle, and bottom, respectively.

Compared to the case without perturbation, the effect of $\beta(\mathbf{x})$ on the dual-arm velocities is clearly observed. The intercept position errors between the object and target with and without adaptation are shown in Figure 6.25; error histograms in the x-direction are shown in the left plot, and error histograms in the y-direction are shown in the right plot for 20 experiments for each case.

In both cases, the mean of the error is about 0. However, considering the variance, the error in the interception direction (Y) ranges from ~ 0.45 m to 0.20 m without adaptation, and between ~ 0.10 m and 0.10 m with adaptation; 13 out of 20 Y-position errors are within $[-0.05, 0.05]$ m. However, in the main tossing direction (X), the error with adaptation is slightly greater than the error without adaptation; 10 out of 20 experiments had an absolute error between 0.05 m and 0.10 m (two had errors of 0.11 m). This is mainly due to the change in momentum of the object before its release as the robot velocities are modulated. One solution to mitigate this effect is to stop modulation (set $\beta(\mathbf{x}) = 1$) as soon as the object is near the release position.

Without adaptation, the object successfully intercepted the target with a position error within the tolerance of 0.10 m in five out of 20 experiments. However, as the system was in open loop, the observed interceptions had a stochastic nature, stemming from the randomness introduced in the speed perturbation, and may have yielded target positions near the desired intercept location at the landing time of the tossed object.

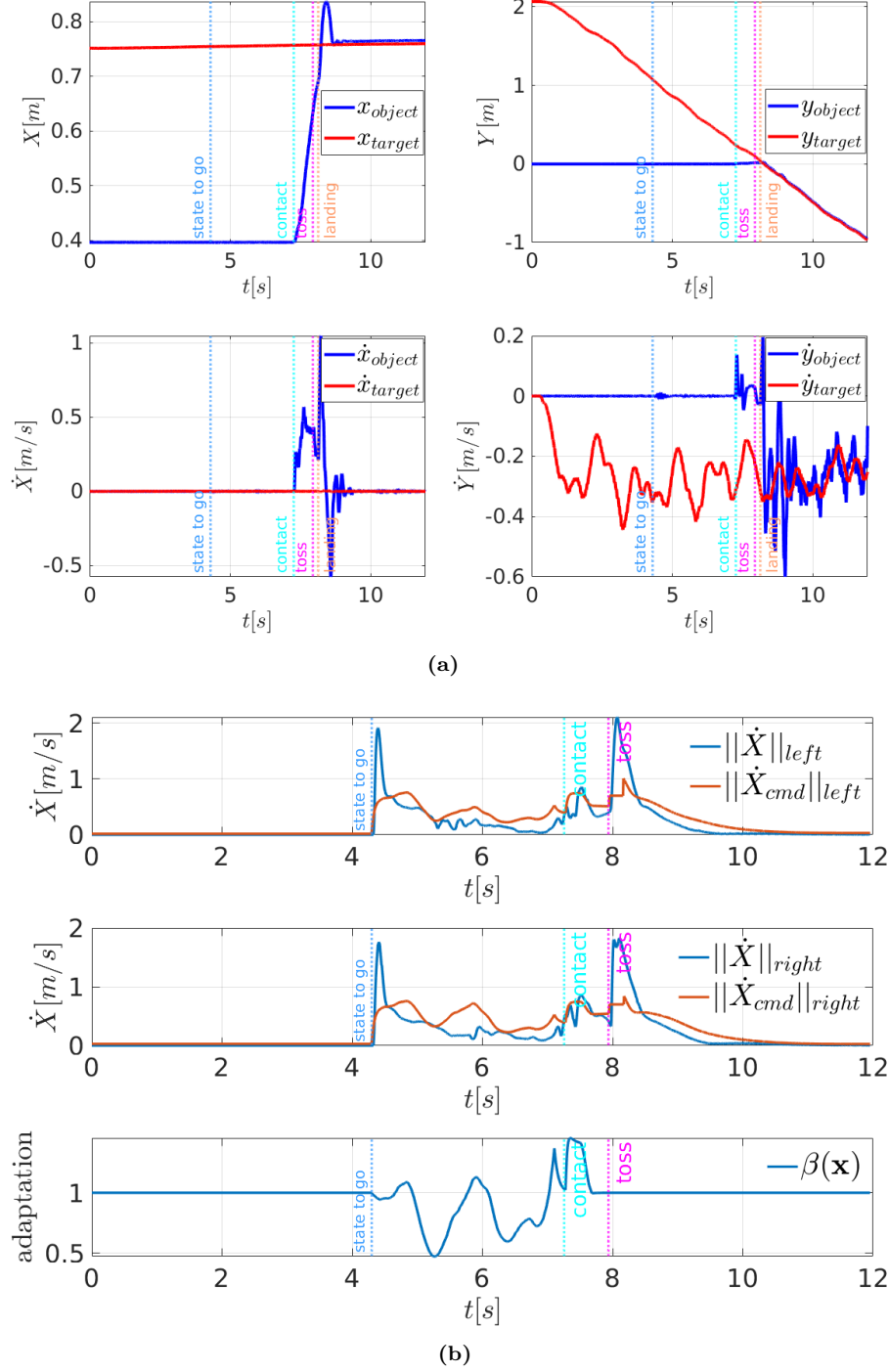


Figure 6.24: Position and velocity plots of dual-arm system grabbing and tossing an object onto a moving target with motion perturbation and **adaptation** of robot motion: (a) X and Y evolution over time for position (top) and velocity (bottom) of the target (red) and object (blue); (b) norm of linear velocities of left robot (top), right robot (middle), and adaptation factor $\beta(\mathbf{x})$ (bottom). (bottom). The robot motion was modulated based on estimated changes in target speed to ensure successful interception.

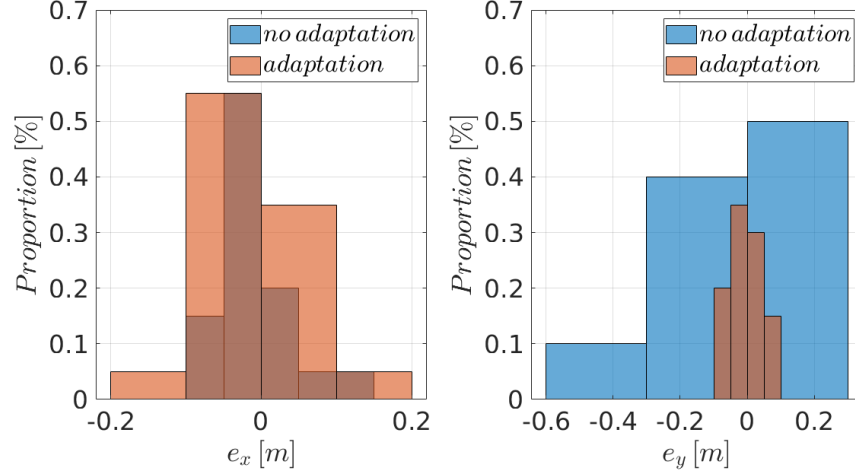


Figure 6.25: Distribution of intercept position errors in the x-direction (left) and y-direction (right) in tossing an object onto a moving target with motion perturbation without adaptation (blue) and with adaptation (orange). The distributions were derived from 20 experiments with and without adaptation. The x-direction represents the main tossing direction; the y-direction represents the interception direction in which the target moves.

Human interaction-based perturbation

In this case, the target speed perturbation was manually induced through interaction with the target. Figure 6.26 shows position and velocity plots for the target–object–robot system with perturbation and with adaptation to compensate for it. The y-components of the positions and velocities of the target and object are shown at the top-left and bottom-left, respectively. The linear velocity norm of the left end-effector is shown on the top-right (the right end-effector is not shown, but follows a similar pattern); the adaptation factor $\beta(\mathbf{x})$ and the y-velocity of the target that drives it are shown at the bottom-right.

Unlike the previous perturbation cases, the target velocity changes in sign from negative to positive and vice versa according to the perturbation. As $\beta(\mathbf{x}) \geq 0$, the modulation cannot reverse the motion direction of the robot (a negative $\beta(\mathbf{x})$ will make the system unstable). Thus, retraction of the robot is achieved by smoothly changing the attractors of the dual-arm system between the grabbing points on the box and the predefined standby position of the end-effectors.

Human interaction with the target, and which induced speed perturbation shown previously in Figure 6.26-(bottom) is illustrated in snapshots of Figure 6.27. The target is pulled back three times, as shown in Figures 6.27 (b)-(c), (d)-(e) and (f)-(g), respectively at $t = 5.95s$, $t = 8.08s$, and $t = 9.98s$, inducing thereby the three velocity picks shown on Figure 6.26-(bottom).

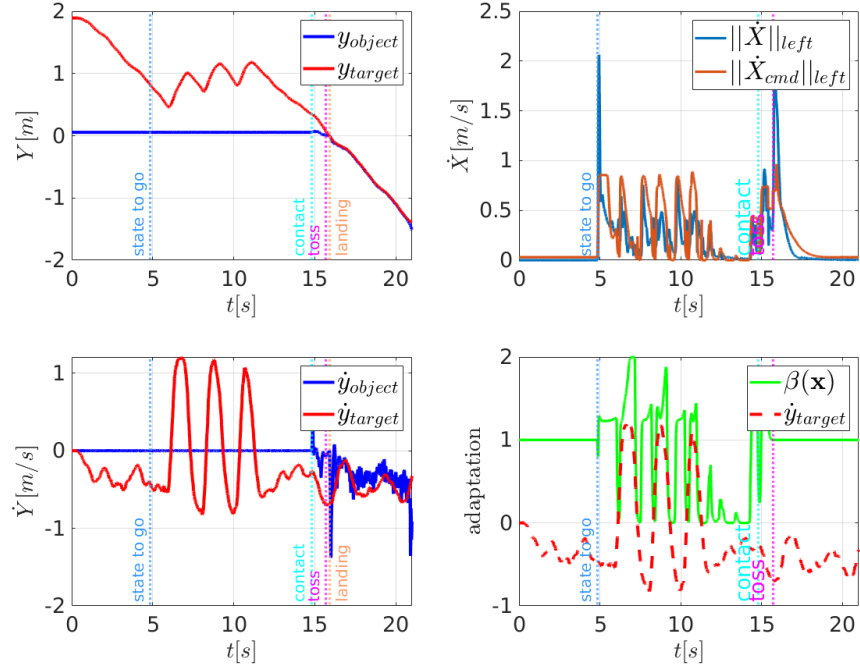


Figure 6.26: Position and velocity plots of dual-arm system with motion adaption while grabbing and tossing an object onto a moving target **with manual perturbation**: left: y-evolution of position (top) and velocity (bottom) of target (red) and object (blue), respectively, over time; (right): norm of linear velocity of left robot (top) and adaptation factor $\beta(\mathbf{x})$ and y-velocity of target (bottom). The robot motion was modulated based on estimated changes in target speed to ensure successful interception.

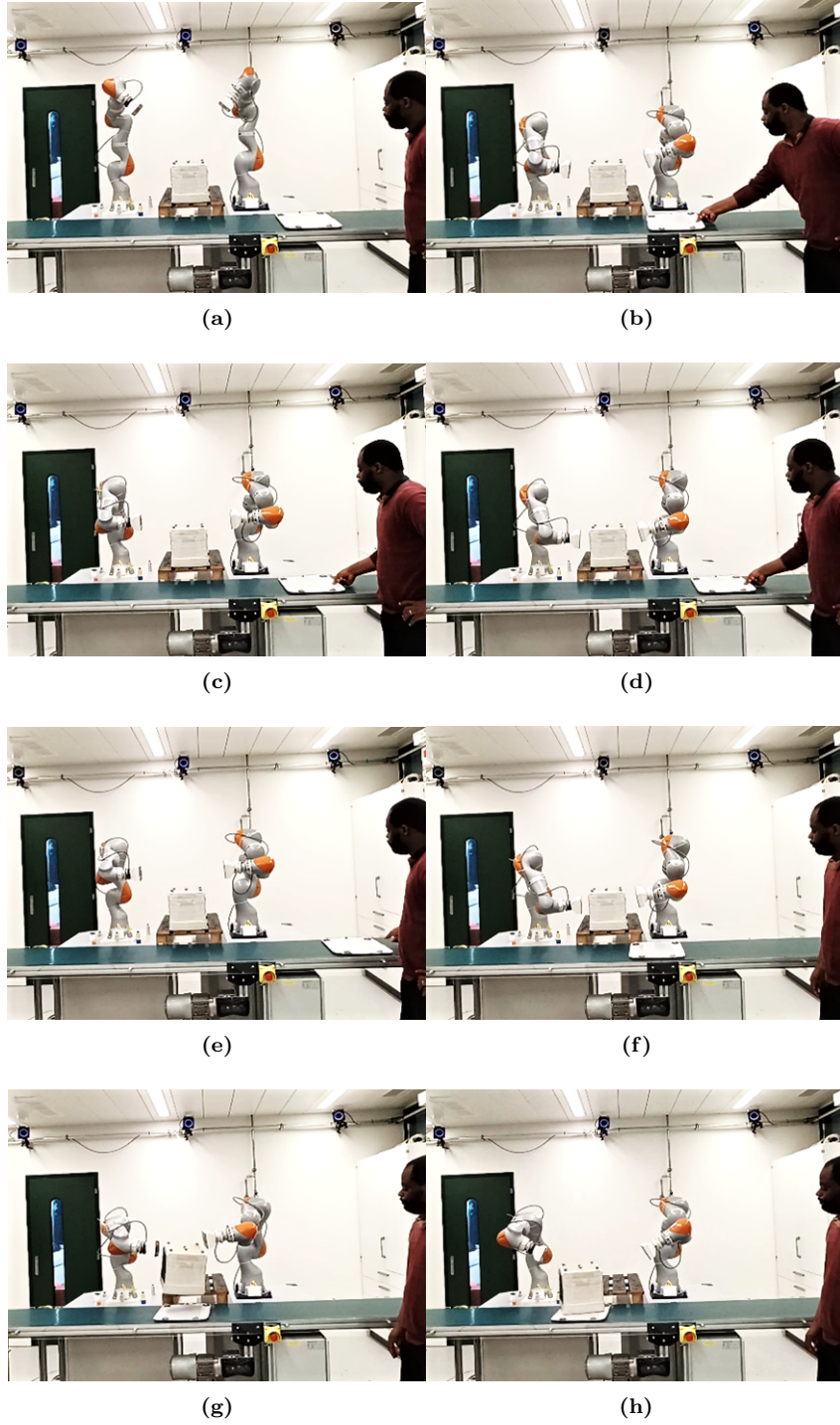


Figure 6.27: Snapshots illustrating adaptation of dual-arm system to manual perturbations of target motion in grabbing and tossing of an object onto a moving target: (a): dual-arm in standby, waiting for target to reach estimated state-to-go; (b)–(c), (d)–(e), and (f)–(g): perturbation introduced by manually pulling back moving target, causing retraction of robots; (h)–(i): grabbing and tossing of object as target moves; (j): motion of object and target after successful interception.

6.6.3 COMPARISON OF PLACING AND TOSSING OBJECT ONTO MOVING TARGET

The goal of this experiment was to compare the kinetic and energy efficiencies of the widely used picking and placing operation with those of the proposed picking and tossing of objects onto a moving target (on conveyor belt). Unlike previous Chapter 5, where such a comparison was conducted from the standby position to the release of the object, in this study, the comparison includes the entire cycle (from the standby position and back after executing the motion), at target speeds ranging from 0.1 m/s to 0.450 m/s in increments of 0.05 m/s. We conducted ten experiments at each target speed and estimated the cycle time and energy consumption of the dual-arm robotic system. Figure 6.28 shows the comparison results of the cycle time, whereas Figure 6.29 shows the energy consumption comparison.

The results indicate that the proposed picking and tossing produces a shorter cycle time and consumes less energy than the picking and placing operation, consistent with Chapter 5. Similar consistent results were observed across target speeds for cycle time and energy expenditure. These experiments were conducted with no perturbation of target speed; thus, the main control variable was the estimated state-to-go, with adaptation having little effect. Slight variations in results were caused by noise, state estimation, and control errors.

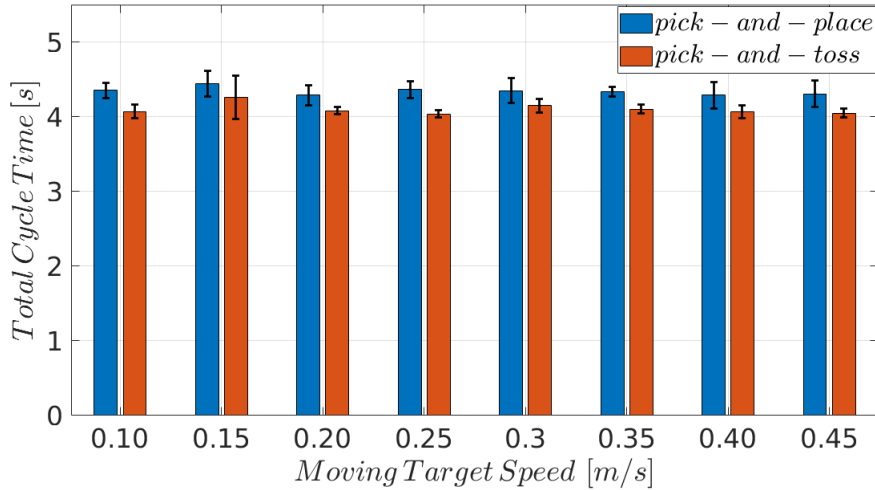


Figure 6.28: Comparison of total energy consumption for picking and placing and picking and tossing of an object onto a moving target at different speeds

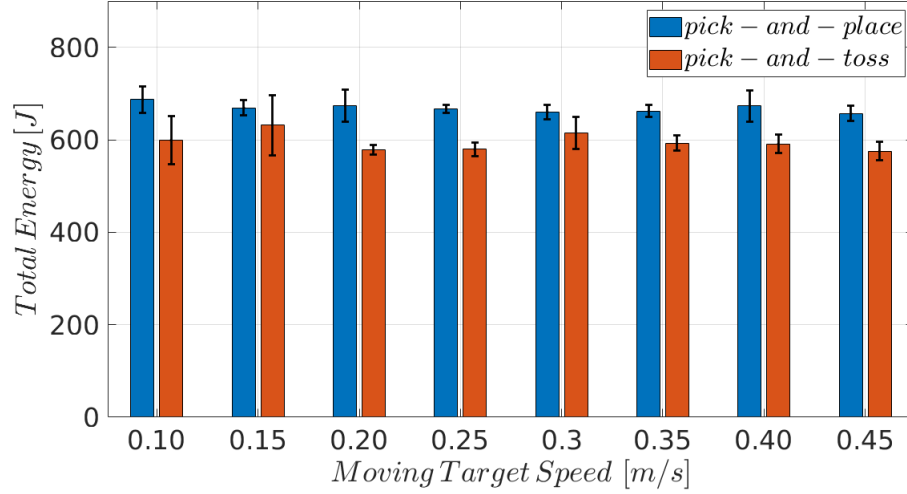


Figure 6.29: Comparison of total energy consumption between operations of picking and placing and picking and tossing of an object on moving target at different speed

The main results of the comparison of picking and placing and picking and tossing are summarized in the histogram shown in Figure 6.30. The proposed picking and tossing is approximately 5.5% faster and consumes approximately 11% less energy than the picking and placing operation for the same positioning task.

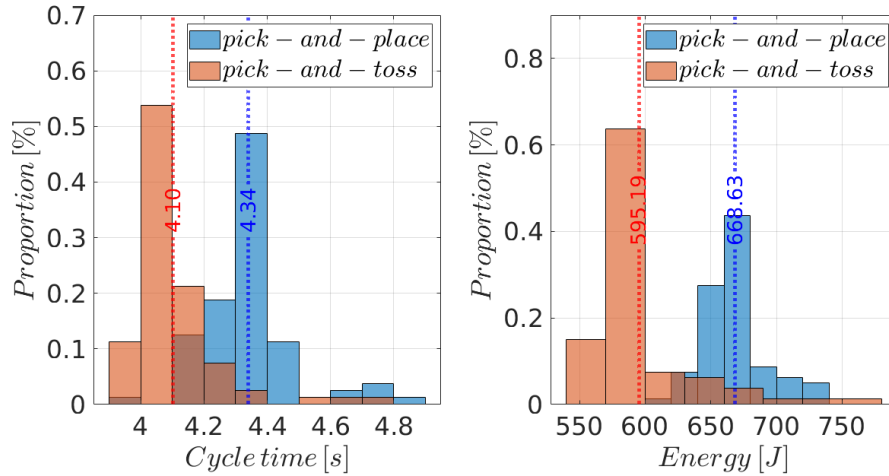


Figure 6.30: Histograms summarizing cycle time and energy expenditure comparisons in picking and placing and picking and tossing of an object onto a moving target

6.7 Discussion and Conclusion

In this Chapter, we have presented a control strategy enabling a dual-arm robotic system to pick up and toss an object to moving target locations on a conveyor belt. Positioning objects on moving conveyor belts is common in industry, and can represent depalletizing of parcels in a sorting facility.

To achieve precise dynamic positioning, we addressed this problem from an interception perspective; a bimanually tossed object governed by projectile dynamics was intended to land at a desired position along the path of a moving target. Using GMR, we learned an inverse throwing map from the nonlinear projectile dynamics to determine the tossing parameters (release position and velocity) necessary to reach the desired landing position, which serve as reference inputs for the robotic system (Section 6.3.1). To ensure feasibility of the release state, we embedded the learned throwing map into a bi-level kinematics-based optimization framework. We translated and solved such a problem at the acceleration level allowing to enforce concurrently velocity and position feasibility constraints with off-shelves solvers (Section 6.3.3).

Building upon the obtained release state feasibility algorithm, we proposed a method to model the tossable workspace of a dual-arm robot, representing the set of all positions reachable by an object tossed by the robot. We generated 10^5 desired positions within and outside the dual-arm system workspace and determined for each whether our algorithm could find a corresponding feasible kinematic release state. We derived a closed-form model of the tossable workspace by learning the distribution of all feasible landing positions using a GMM. The obtained model allows us to predict the the probability of reaching potential intercept positions of the object with the target before initiating the robot motion (Section 6.5). For robust tossing, we used our dynamical system-based control framework previously proposed in Chapter 5; and we complemented it with an adaptation strategy to modulate the generated motion of the dual-arm system and enable grabbing and tossing of an object onto a moving target with motion perturbations.

To demonstrate the validity of the proposed approach, in addition to simulations, we conducted experiments using a pair of actual KUKA robots. We evaluated the accuracy and repeatability of interception at different target speeds, with and without motion perturbations. To highlight the benefits of our approach with respect to classical positioning tasks based on pick-and-place operations, we implemented a picking and placing strategy for an object on a moving target and compared it in terms of cycle time and energy consumption with the proposed picking and tossing strategy.

We found that in the absence of target's motion perturbation and for the selected release configurations shown in Figure 6.22, the object landed within a radius of 0.05 m from the center of the target in 67 out of 80 tossing cases, that is 77%; the remaining tosses were within a radius of 0.10 m.

With target motion perturbation and no adaptation of the robot motion, the ratio of tosses with interception error norms less than 0.10 m decreased to 20%, mainly due to increased errors in the intercept direction (direction of target velocity); 75% of tosses were between 0.10 and 0.45 m from the target; in the tossing direction (object’s release velocity direction), 95% of tosses were within 0.10 m of the target center. With the proposed adaptation scheme, although errors in the tossing direction may slightly increase, we demonstrated that the intercept error norms decreased significantly; 90% of tosses were within a 0.10 m radius of target.

When compared to the traditional pick and place strategy, we found that the proposed pick and toss approach leads to a shorter cycle time (5.5% in our experiments). This is consistent with previous results in [Raptopoulos et al. \(2020\)](#); [Hassan et al. \(2022\)](#); [Bombile and Billard \(2022\)](#). Moreover, as in [Bombile and Billard \(2022\)](#), we also found that the proposed approach consumes less energy (11%).

Although the effectiveness of the proposed method was demonstrated using actual robots, some limitations in approach and implementation were evident. While the determined release states were kinematically feasible, their dynamical feasibility was not guaranteed. Dynamical feasibility depends on the initial robot configuration and the inertia of the object. Moreover, geometric properties of the object and dual-arm grabbers may limit the feasible states due to collisions at the release time if the robot cannot retract fast enough.

Furthermore, in determining the release configurations, we considered the free-flying dynamics of the object after release, but did not consider the impact dynamics of the object at landing. Thus, we observed that for some feasible release states determined by our algorithm, the tossed object bounced on the conveyor belt and fell from it instead of remaining on it. To prevent this, we constrained the search space of feasible release configurations to a small set of task-space release positions near the conveyor belt. Thus, the associated release velocities were reduced and the landing impacts too. A more general solution in determining the release state should consider the desired post-landing impact state of the object and not simply the landing state. Moreover, uncertainty in the release time should also be considered. Furthermore, to guarantee optimal and safe behavior using the proposed approach and promote its use in industry, future research should consider the full dynamics of the robots and the dynamics of their interactions with the environment.

GENERAL CONCLUSION

In this Chapter, we summarize the main contributions of this thesis. We also discuss the limitations of the work presented in this thesis and we provide possible improvement directions.

7.1 Main contributions

The main contribution of this thesis lies essentially in providing a generic and unified modulated DS-based motion generation framework for manipulation tasks with desired transitory states. The proposed framework is generic because it can be formulated in task space as well as in joint space with first or second-order dynamics. Its unified nature stems from the framework's ability to handle both free and constrained motions and to robustly transition between them in cooperative settings.

7.1.1 REACTIVE OMNIDIRECTIONAL WALKING MOTION GENERATOR

In Chapter 3, we have presented a motion generation algorithm for a more complex robotic system: a dual-arm humanoid robot. We focussed on dynamic balance and locomotion and proposed a capture-point-based walking controller able to generate on the fly omnidirectional walking patterns for a biped robot and to stabilize the robot around them. We combined the center of mass (CoM) and the capture-point dynamics into an MPC framework to generate automatically foot-steps positions and orientations and the associated motion of the CoM based on desired velocities of the robot. We validated experimentally the proposed control scheme on a real humanoid robot and we demonstrated its effectiveness in two types of tasks where a walking approach based on a footstep planner could not apply. In the first task, we required the robot to track changing translational and rotational velocities. The experiments showed that the humanoid robot under the proposed controller could track the commanded omnidirectional velocities and could even perform a rotation on a spot, particularly when following a pure rotational velocity. In the second type of experiment, we showed how the ability to automatically generate omnidirectional walking motions could serve in

human-robot cooperative tasks. Thus, with the humanoid robot in the role of follower, we demonstrated the reactive generation and adaptation of the robot’s footsteps in order to comply with the intentions of the human, first in a guidance task and then in a cooperative transportation task.

7.1.2 COOPERATIVE COMPLIANT MANIPULATION ON HUMANOID ROBOT

In Chapter 4, we proposed an approach to achieve stable bimanual reach-to-grasp and compliant manipulation of an object on a humanoid robot. We relied on dynamical systems and exploits a concept of shrinkable virtual objects to achieve motion coordination by imposing virtual constraints to the robot’s hands. Moreover, the shrinkage ensured smooth transition from virtual constraints in free-motion to real constraints when the object was grasped. To stabilize the grasp and achieve desired manipulation tasks, contacts-consistent optimal wrenches were computed online using QP. We validate the proposed solution on the humanoid robot iCub by simulating various bimanual manipulation tasks such as coordinated reaching and grabbing of static as well as a moving objects. Coordinated grabbing of objects with point, line, and surface contacts, and bimanual grasp stability of the object subjected to strong external forces were realized to illustrate the robustness of the proposed controller.

7.1.3 COORDINATED FAST GRABBING AND TOSSING OF OBJECTS

In Chapter 5, we presented the unified motion generation framework in coordinated control tasks of a dual-arm robotic system. The motion generated allowed the dual-arm robot to quickly grab with impact and toss an object in one swipe. The modulation allowed to shape locally the motion of the two robots in order to reach their desired states at the contact and release of the object while preserving the coordination. We theoretically proved the stability of the proposed framework as well as its convergence towards the desired states. Using a QP, we have generated and combined with the motion framework constraints-consistent forces necessary to stabilize the grasp and to achieve the dynamic cooperative manipulation task. We then validated experimentally the overall control strategy in pick and toss tasks. The obtained results confirmed that the proposed unified framework enables indeed a dual-arm system to generate in a coordinated manner desired impacts and tossing motion when grabbing and releasing an object, respectively. The results have shown that grabbing with impact and tossing, especially when the impact direction anticipates the upcoming motion of the object, leads to shorter and more energy-efficient pick and place tasks.

7.1.4 GRABBING AND TOSSING OF OBJECTS ON MOVING TARGET

In Chapter 6, we presented a control strategy that extends the dynamic capabilities of the framework presented in Chapter 5 by enabling a dual-arm robotic system to toss objects onto a moving target carried by a conveyor belt.

We proposed an algorithm to determine feasible tossing parameters (release position and release velocity) necessary to achieve the precise dynamic positioning task of the object. Thus, using GMM/GMR, we learned from the non-linear projectile dynamics of a thrown object an inverse throwing map and combined it with a kinematics-based optimization framework. The developed algorithm can generate, for the dual-arm system, kinematically feasible release states associated with a given tossing task.

Exploiting the obtained release states feasibility algorithm, we proposed a modeling approach for the tossable workspace of the dual-arm robot. Based on GMM, our approach yields a closed-form model allowing us to predict in real-time the reachability of a given tossing target position before initiating any action on the robot side. We relied on the DS-based framework developed in Chapter 5 to execute in a robust way the object tossing task on a moving target. To execute in a robust way the object tossing task, we relied on the DS-based framework developed in Chapter 5, for which we proposed a complementary adaptation strategy that enables tossing on the moving target in the presence of motion perturbations.

Finally, we have validated experimentally the proposed dynamic manipulation framework. We evaluated the accuracy and repeatability of the interception tasks between the bimanually tossed object and the target moving at different speeds. We also assessed the efficacy of the proposed adaptation strategy in the presence of strong and sudden perturbations of the target motion. The obtained results have confirmed the effectiveness of the proposed dynamic and cooperative manipulation strategy to enable a dual-arm system to swiftly grab and toss an object onto a moving target. Furthermore, we compared the efficiency in terms of cycle time and energy expenditure of the proposed tossing framework with respect to a classical framework where the object is gently placed on the moving target. As in Chapter 5, the comparison results have shown that the proposed tossing-based positioning framework is not only faster but consumes less energy than a placing-based framework.

7.2 Limitations and Future Works

Our thesis’s key contributions have limitations and shortcomings that were essentially discussed in their respective chapters. In this section, we further elaborate on the main limitations and provides research prospects that could improve the presented work.

In Chapter 5 and Chapter 6, we generated desired impact velocities when grabbing objects without however taking into account the impact dynamics (both at the task level and at the control level). Throughout our experiments, we limited the desired impact speeds and assumed that the induced impact forces, as well as the jumps of velocities and torques, remained within the safe limits for the robot and for the object. However, this assumption is limited since the impact forces and the velocity jumps do not only depend on the pre-impact velocities but also on the configurations of the robots and the properties of the impacting materials (e.g. rigidity, coefficient of restitution, etc.). To address the aforementioned problem, future works should include impact dynamics in determining impact velocity limits. Moreover, at the control level, a potential solution would be to combine the proposed DS with controllers such as those proposed in (Wang et al., 2020) or (Dehio and Kheddar, 2021; Dehio et al., 2022), designed to keep force jumps and subsequent torque jumps within the limits of the robot. In addition, the continuity of the torques sent to the robots’ motors can be ensured, for example, by projecting the command into an impact-invariant space as proposed in (Yang and Posa, 2021) or by using control strategies inspired by the reference spreading approach (Rijnen et al., 2019). Furthermore, while the notion of feasibility is fairly clear for the robot, the acceptable limits of impact forces and energy for the object are still to be determined and used in the design of the grabbing and tossing strategy. Instead of a minimal release velocity strategy as adopted for tossing in this thesis, considerations of the object’s limits might leads to adoption of a tossing strategy that seeks, for instance, to minimize the landing speed or its components.

Regarding the parameters of the proposed dynamical systems, we limited ourselves to the demonstrations of the developed concepts and their functionality rather than to the optimality of the generated trajectories. Therefore, we used quasi-linear dynamic systems which do not generate optimal motions either in terms of energy or execution time. To address this problem, one solution would be to use Linear Parameters Varying (LPV) DS as in (Mirrazavi Salehian et al., 2017b) and embed the optimality of the trajectories in their parameters. These DS parameters could then be learned from the trajectories generated off-line by optimal controllers, for example, a minimum time controller as in (Hassan et al., 2022) to minimize the cycle time, or a minimum energy control to minimize the energy consumed during execution.

Although the proposed DS can be formulated both in task space and in joint

space, we only implemented our dual-arm framework in task space (where the coordination mattered the most). Hence the DS is agnostic of what happens at the joint level and cannot take advantage of the robots configurations to achieve a more efficient tossing motion. This was evident when analyzing the joint space velocities (see Appendix D.6). We noticed that among the two main joints contributing the most to the tossing motion (joint 2 and joint 4 of the KUKA LBR IIWA robot whose rotation is more aligned with the tossing direction) one of the joints, namely the joint 2, was under-utilized while the other was close to its saturation at tossing time. One possible solution could be to use of joint configuration-dependent modulation functions while still keeping the motion in task space for coordination.

Indeed, the motion modulation framework, at the core of the method presented in this thesis, offers a very powerful framework to embed soft and hard constraints into dynamical systems. While the handling of hard constraints such as non-penetrability of obstacles has been demonstrated with task space modulation functions in (Khansari-Zadeh and Billard, 2012) or (Huber et al., 2019a), nothing prevents to use joint space configuration-dependent functions in conjunction with the robot’s kinematics to modulate the motion of the robotic system in order to satisfy, for instance, the robot joint limits or avoid self-collisions. Thus, such a solution will result in constraint-aware DS, capable of generating motions that are consistent with the constraints of a given robotic system.

Regarding the method presented in Chapter 6, the validity of the generated feasible release states is subject to the satisfaction of mutual and self-collision constraints which were not explicitly included in the proposed algorithm. In general, we did not generate self-colliding configurations mainly because of the grasp constraints of the object and its chosen orientation during the tossing task. However, for operational safety in industrial settings, mutual and self-collision constraints must be included in our framework. In that regard, a potential solution to extend our presented framework could be inspired by the approach proposed in (Mirrazavi Salehian et al., 2018b). Moreover, since the decision variable in our framework is the joint acceleration, the inclusion of dynamic constraints should be straightforward to generate release configurations that are dynamically feasible for given initial configurations. Such an extension with the robot’s dynamics and the self-collision constraints will certainly come at the expense of increased computational cost. However, given the benefits provided by the proposed method, efforts to improve the approach and foster its adoption in the industry are worth being pursued.

The framework proposed in this thesis uses two robots simultaneously in a single station. For such a solution to be adopted in the industry, it needs to be flexible enough. While dual-arm systems can handle large or heavy objects or can perform tasks too complex for a single-arm system, they have, however, a major drawback when mounted on fixed bases. They have a reduced joint workspace, which limits their sphere of operations. Thus, the solution to

expand their joint workspace is to increase the degrees-of-freedom of their common base, or better endowing the dual-arm system with mobility. This is for example the case of the robot ARMAR-6 ([Asfour et al., 2018](#)) which combines its bimanual capabilities with its mobility and has demonstrated its ability to cover a large workspace in collaborative tasks with humans and also in cooperative manipulation tasks ([Gao et al., 2018](#)). Thus, future work to expand our proposed dynamic framework could explore such a direction while addressing the potential balance problem that it might create.

Regarding the reactive walking algorithm, given the current state of the art in humanoid balance and locomotion, for instance, with Boston’s dynamics robots exhibiting highly agile balance and locomotion skills, one could question the relevance of such a walking controller. However, exploiting the analogy between dual-arm manipulation and the balance and locomotion of a biped robot, the proposed framework, thanks to its reactivity could find applications in dynamic grasp adaptation or in-hand manipulation with contacts relocation. This stems from the observation that the force-level grasp adaptation is like the balance task and the contacts relocation like the stepping (used to recover from disturbance or to walk). Indeed, to compensate for strong grasp instability or to deliberately change the grasping points or the orientation of a grasped object, the proposed algorithm can be used to generate stable sequences of contact positions for the end-effectors. From that perspective, the concept of the capture-point and related capture-region ([Koolen et al., 2012](#)) (a region where a legged robot should step to come to a complete stop) can be expanded and used to identify a location where contact should be relocated to recover the grasp stability of the manipulated object. Research interested in dual-arm in-hand manipulation could explore such a direction.

Appendices

APPENDICES FOR CHAPTER 2

A.1 Computation of CoM wrench maps

The balance task of a n -DoF humanoid robot assumed to interact with its environment only through the hands and feet can be written as

$$\dot{\mathbf{h}} = \mathbf{A}_{\mathbf{q}}\ddot{\mathbf{q}} + \dot{\mathbf{A}}_{\mathbf{q}}\dot{\mathbf{q}} = \mathbf{G}_{\text{cf}}\mathbf{f}_{\text{f}} + \mathbf{G}_{\text{ch}}\mathbf{f}_{\text{h}} + \mathbf{f}_{\text{gvt}} \quad (\text{A.1.1})$$

where $\mathbf{A}_{\mathbf{q}} \in \mathbb{R}^{6 \times n}$ is the centroidal moment matrix (Orin and Goswami, 2008). $\mathbf{G}_{\text{cf}} \in \mathbb{R}^{6 \times 6}$ and $\mathbf{G}_{\text{ch}} \in \mathbb{R}^{6 \times 6}$ represent wrench transformation matrices that map respectively the feet and hands contact wrenches to the frame attached to the CoM. $\mathbf{f}_{\text{gvt}} \in \mathbb{R}^6$ is the gravity forces acting on the CoM frame.

The expressions of $\mathbf{A}_{\mathbf{q}}$, $\dot{\mathbf{A}}_{\mathbf{q}}$, \mathbf{G}_{cf} and \mathbf{G}_{ch} can be obtained through the following transformation

$${}^{\text{C}}\mathbf{X}_{\text{B}}^{-\top}(\mathbf{M}_{\text{u}}\ddot{\mathbf{q}} + \mathbf{b}_{\text{u}}) = {}^{\text{C}}\mathbf{X}_{\text{B}}^{-\top}(\mathbf{J}_{\text{fu}}^{\top}\mathbf{f}_{\text{f}} + \mathbf{J}_{\text{hu}}^{\top}\mathbf{f}_{\text{h}}) \quad (\text{A.1.2})$$

where the subscript $_{\text{u}}$ indicates the six upper rows of the robot's dynamics that relate to the floating base¹. ${}^{\text{C}}\mathbf{X}_{\text{B}} \in \mathbb{R}^{6 \times 6}$ is velocity twist transformation matrix from the robot's base frame to the frame attached to the center of mass (CoM). ${}^{\text{C}}\mathbf{X}_{\text{B}}$ is given by (Nava et al., 2016)

$${}^{\text{C}}\mathbf{X}_{\text{B}} \triangleq \begin{bmatrix} \mathbf{I} & -[\mathbf{x}_{\text{C}} - \mathbf{x}_{\text{B}}]_{\times} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (\text{A.1.3})$$

where \mathbf{x}_{C} and \mathbf{x}_{B} denote the positions of the CoM and floating base, respectively. The notation $[\cdot]_{\times}$ denotes a skew symmetric matrix. Hence,

$${}^{\text{C}}\mathbf{X}_{\text{B}}^{-\top}(\mathbf{M}_{\text{u}}\ddot{\mathbf{q}} + \mathbf{b}_{\text{u}}) = \mathbf{A}_{\mathbf{q}}\ddot{\mathbf{q}} + \dot{\mathbf{A}}_{\mathbf{q}}\dot{\mathbf{q}} - \mathbf{f}_{\text{gvt}} \quad (\text{A.1.4})$$

$${}^{\text{C}}\mathbf{X}_{\text{B}}^{-\top}\mathbf{J}_{\text{fu}}^{\top}\mathbf{f}_{\text{f}} \triangleq \mathbf{G}_{\text{cf}}\mathbf{f}_{\text{f}} \quad (\text{A.1.5})$$

$${}^{\text{C}}\mathbf{X}_{\text{B}}^{-\top}\mathbf{J}_{\text{hu}}^{\top}\mathbf{f}_{\text{h}} \triangleq \mathbf{G}_{\text{ch}}\mathbf{f}_{\text{h}} \quad (\text{A.1.6})$$

where $\mathbf{G}_{\text{cf}} = {}^{\text{C}}\mathbf{X}_{\text{B}}^{-\top}\mathbf{J}_{\text{fu}}^{\top}$ and $\mathbf{G}_{\text{ch}} = {}^{\text{C}}\mathbf{X}_{\text{B}}^{-\top}\mathbf{J}_{\text{hu}}^{\top}$.

¹The dynamics can be rewritten as $\begin{bmatrix} \mathbf{M}_{\text{u}} \\ \mathbf{M}_{\text{l}} \end{bmatrix} \ddot{\mathbf{q}} + \begin{bmatrix} \mathbf{b}_{\text{u}} \\ \mathbf{b}_{\text{l}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{f} \end{bmatrix} + \begin{bmatrix} \mathbf{J}_{\text{fp}}^{\top} \\ \mathbf{J}_{\text{fl}}^{\top} \end{bmatrix} \mathbf{f}_{\text{f}} + \begin{bmatrix} \mathbf{J}_{\text{hp}}^{\top} \\ \mathbf{J}_{\text{hl}}^{\top} \end{bmatrix} \mathbf{f}_{\text{h}}$

A.2 Lumped Contact constraints matrix and vector

Consider the contact situation represented in Figure A.1, where a robot's end-effector applies unilateral forces on a object.

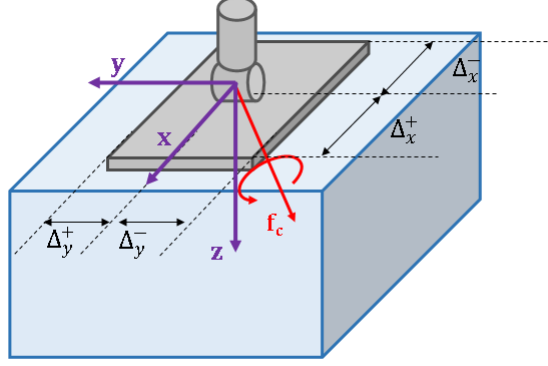


Figure A.1: Geometrical modeling of contact and associated interaction wrench.

Let $\mathbf{f}_c \triangleq [f_c^x \ f_c^y \ f_c^z \ \tau_c^x \ \tau_c^y \ \tau_c^z]^\top \in \mathbb{R}^6$ denotes the contact wrench, with $[f_c^x \ f_c^y \ f_c^z]$ and $[\tau_c^x \ \tau_c^y \ \tau_c^z]$ representing the force and moment components of \mathbf{f}_c , respectively. If the z direction is assumed to be normal to the contact surface, the unilateral force constraint implies that

$$f_c^z \geq 0 \quad (\text{A.2.1})$$

Moreover, using Coulomb friction model, a non slipping contact implies that all tangential components of \mathbf{f}_c remain within the friction cone. That is

$$\mu_f f_c^z \geq \sqrt{(f_c^x)^2 + (f_c^y)^2} \quad (\text{A.2.2})$$

$$\gamma_f f_c^z \geq |\tau_c^z| \quad (\text{A.2.3})$$

where μ_f and γ_f are respectively the linear and rotational friction coefficients.

Furthermore, the tipping over is avoided by ensuring that the center of pressure or zero moment point of each robot's end-effector in unilateral contact stays within the convex hull formed by the contact points or contact surface. Thus, this can be written as

$$\begin{aligned} \Delta_y^+ f_c^z &\geq \tau_c^x \geq -\Delta_y^- f_c^z \\ \Delta_x^+ f_c^z &\geq \tau_c^y \geq -\Delta_x^- f_c^z \end{aligned}$$

where Δ_x^+ and Δ_x^- , and Δ_y^+ and Δ_y^- as shown in Figure A.1 specify, respectively, the length and width dimensions of the contact surface.

Due to the friction cone, the constraint (A.2.2) is non-linear. However, it can be linearized by approximating the friction cone by a convex polyhedron (Wieber, 2002). In our case, we use a friction pyramid which results in the following constraints

$$\frac{\mu_f}{\sqrt{2}} f_c^z \geq |f_c^x| \quad \text{and} \quad \frac{\mu_f}{\sqrt{2}} f_c^z \geq |f_c^y| \quad (\text{A.2.4})$$

In the local frame, the above constraints can be lumped and expressed as

$$\mathbf{C}_c \mathbf{f}_c = \mathbf{d}_c \quad (\text{A.2.5})$$

where

$$\mathbf{C}_c = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & -\frac{\mu_f}{\sqrt{2}} & 0 & 0 & 0 \\ 1 & 0 & -\frac{\mu_f}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & -1 & -\frac{\mu_f}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 1 & -\frac{\mu_f}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & -\gamma_f & 0 & 0 & -1 \\ 0 & 0 & -\gamma_f & 0 & 0 & 1 \\ 0 & 0 & -\Delta_y^- & -1 & 0 & 0 \\ 0 & 0 & -\Delta_y^+ & 1 & 0 & 0 \\ 0 & 0 & -\Delta_x^- & 0 & -1 & 0 \\ 0 & 0 & -\Delta_x^+ & 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{d}_c = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (\text{A.2.6})$$

Hence, in the world frame, these constraints can be obtained by transforming the wrench from local frame to global frame using a wrench map.

APPENDICES FOR CHAPTER 3

B.1 Hessian matrix and gradient vector of the QP

The $Q_{h,ij}$ ($h = x, y$) and $p_{k,i}$ elements, respectively, of the Hessian matrix and gradient vector are given by

$$\begin{aligned}
 Q_k^{p_h} &= \beta U_c^\top E^\top E U_c + \gamma U_\xi^\top U_\xi + \kappa \Gamma^\top \Gamma \\
 Q_k^{p_h \Delta^r} &= (Q_k^{\Delta^r p_h})^\top = -\gamma U_\xi^\top (\underline{\Xi}_{r_m k} \mathbf{V}_f \underline{\Xi}_{r_m k} \mathbf{R}_h + \underline{\Xi}_{\xi_N k} \mathbf{R}_{h(3:)}) \\
 Q_k^{p_h \xi_N} &= (Q_k^{\xi_N p_h})^\top = -\gamma U_\xi^\top \underline{\Xi}_{\xi_N k} R_{h, s_{i+2}}^w \\
 Q_{k,h}^{\Delta^r} &= \gamma (\underline{\Xi}_{r_m k} \mathbf{V}_f \mathbf{R}_h + \underline{\Xi}_{\xi_N k} \mathbf{R}_{h(3:)})^\top (\underline{\Xi}_{r_m k} \mathbf{V}_f \mathbf{R}_h + \underline{\Xi}_{\xi_N k} \mathbf{R}_{h(3:)}) \\
 Q_{k,h}^{\Delta^r \xi_N} &= (Q_k^{\xi_N \Delta^r})^\top = \gamma (\underline{\Xi}_{r_m k} \mathbf{V}_f \mathbf{R}_h + \underline{\Xi}_{\xi_N k} \mathbf{R}_{h(3:)})^\top \underline{\Xi}_{\xi_N k} R_{h, s_{i+2}}^w \\
 Q_{k,h}^{\xi_N} &= \gamma (\underline{\Xi}_{\xi_N k} + R_{h, s_{i+2}}^w)^\top (\underline{\Xi}_{\xi_N k} + R_{h, s_{i+2}}^w) \\
 \begin{cases} Q_k^{\Delta^r} & \triangleq Q_{k,x}^{\Delta^r} + Q_{k,y}^{\Delta^r} \\ Q_k^{\Delta^r \xi_N} & \triangleq Q_{k,x}^{\Delta^r \xi_N} + Q_{k,y}^{\Delta^r \xi_N} \\ Q_k^{\xi_N} & \triangleq Q_{k,x}^{\xi_N} + Q_{k,y}^{\xi_N} \end{cases} \\
 Q_\theta &= \begin{bmatrix} \alpha_\theta U_\theta^\top U_\theta + \gamma_\theta U_\theta^\top U_\theta + \kappa_\theta I & -\gamma_\theta U_\theta^\top H_{k+1}^f \\ -\gamma_\theta (H_{k+1}^f)^\top U_\theta & \gamma_\theta (H_{k+1}^f)^\top H_{k+1}^f \end{bmatrix}, \tag{B.1.1}
 \end{aligned}$$

and the vector $\mathbf{p}_k \triangleq \begin{bmatrix} \mathbf{p}_{k_{xy}}^\top & \mathbf{p}_{k\theta}^\top \end{bmatrix}^\top$, with

$$\begin{aligned}
 p_k^{p_h} &= \kappa U_c^\top E^\top (E S_c \mathbf{x}_h(k) - \dot{\mathbf{c}}_k^{ref}) \\
 &\quad + \gamma U_\xi^\top (S_\xi \mathbf{x}_h(k) - \underline{\Xi}_{r_m k} (V_c + \mathbf{V}_f \mathbf{1}_3) r_{i,k}^w) \\
 &\quad + \gamma U_\xi^\top \underline{\Xi}_{\xi_N k} r_{i,k}^w - \kappa \Gamma^\top \mathbf{e}_1 p_{h,k-1} \\
 p_k^{\Delta^r} &= -\gamma (\underline{\Xi}_{r_m k} \mathbf{V}_f \mathbf{R}_h + \underline{\Xi}_{\xi_N k} \mathbf{R}_{h(3:)})^\top \\
 &\quad (S_\xi \mathbf{x}_h(k) - (\underline{\Xi}_{r_m k} (V_c + \mathbf{V}_f \mathbf{1}_3) + \underline{\Xi}_{\xi_N k}) r_{i,k}^w) \\
 p_k^{\xi_N} &= -\gamma (\underline{\Xi}_{\xi_N k} R_{h, s_{i+2}}^w)^\top (\underline{\Xi}_{r_m k} V_c r_{i,k}^w - S_\xi \mathbf{x}_h(k)) \\
 &\quad (S_\xi \mathbf{x}_h(k) - (\underline{\Xi}_{r_m k} (V_c + \mathbf{V}_f \mathbf{1}_3) + \underline{\Xi}_{\xi_N k}) r_{i,k}^w)
 \end{aligned}$$

$$\mathbf{p}_{\mathbf{k}\theta} = \begin{bmatrix} \alpha_{\theta} U_{\theta}^{\top} (S_{\theta} \boldsymbol{\theta}(k) - \hat{\boldsymbol{\theta}}_k^{ref}) + \gamma_{\theta} U_{\theta}^{\top} (S_{\theta} \boldsymbol{\theta}(k) - H_{k+1}^c \boldsymbol{\theta}_{f,i}^w) \\ -\gamma_{\theta} (H_{k+1}^f)^{\top} (S_{\theta} \boldsymbol{\theta}(k) - H_{k+1}^c \boldsymbol{\theta}_{f,i}^w) \end{bmatrix} \quad (\text{B.1.2})$$

APPENDICES FOR CHAPTER 4

C.1 Inequality constraints for the whole-body controller

C.2 Definition of Matrix \mathbf{L}_{jk}

The matrix $\mathbf{L}_{jk} \in \mathbb{R}^{6 \times 6}$ is here given by

$$\mathbf{L}_{ij} = \begin{bmatrix} {}^j\mathbf{R}_w & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{L}_{\theta\boldsymbol{\mu}_{jk}} {}^i\mathbf{R}_w \end{bmatrix} \quad (\text{C.2.1})$$

where ${}^j\mathbf{R}_w \in \mathbb{R}^{3 \times 3}$ and ${}^k\mathbf{R}_w \in \mathbb{R}^{3 \times 3}$ are rotation matrices of the world frame Σ_w with respect to Σ_j and Σ_k , respectively. $\mathbf{L}_{\theta\boldsymbol{\mu}_{jk}}$ is given by [Malis et al. \(1999\)](#)

$$\mathbf{L}_{\theta\boldsymbol{\mu}_{jk}} = \mathbf{I}_{3 \times 3} - \frac{\theta}{2} [\boldsymbol{\mu}_{jk}]_{\times} + \left(1 - \frac{\text{sinc } \theta}{\text{sinc}^2 \frac{\theta}{2}} \right) [\boldsymbol{\mu}_{jk}]_{\times}^2$$

where $\theta \text{sinc } \theta = \sin \theta$ and $[\boldsymbol{\mu}_{jk}]_{\times} \in \mathbb{R}^{3 \times 3}$ denotes a skew-symmetric matrix associated with $\boldsymbol{\mu}_{jk}$.

C.3 Stability and Convergence of the reach-to-grasp

Consider the following Lyapunov candidate function

$$V_{h_i} = \boldsymbol{\xi}_{h_i}^{\top} \mathbf{P}_{h_i} \boldsymbol{\xi}_{h_i} \succ 0 \quad (\text{C.3.1})$$

It is strictly positive except at $\boldsymbol{\xi}_{h_i} = \mathbf{0}$. \dot{V}_{h_i} will be given by

$$\dot{V}_{h_i} = \boldsymbol{\xi}_{h_i}^{\top} \mathbf{P}_{h_i} \dot{\boldsymbol{\xi}}_{h_i} + \dot{\boldsymbol{\xi}}_{h_i}^{\top} \mathbf{P}_{h_i} \boldsymbol{\xi}_{h_i} \quad (\text{C.3.2})$$

Using recursively $\dot{\boldsymbol{\xi}}_{jk} = \mathbf{L}_{jk}(\dot{\mathbf{x}}_j - \dot{\mathbf{x}}_k)$, it can be shown that

$$\dot{\boldsymbol{\xi}}_{h_i} = [\mathbf{L}_{h_i v_i}^{-1} \mathbf{L}_{v_i v_i^*}^{-1} \mathbf{L}_{v_i^* o_i}^{-1}]^+ (\dot{\mathbf{x}}_{h_i} - \dot{\mathbf{x}}_{o_i}) \quad (\text{C.3.3})$$

where $[\cdot]^+$ denotes a pseudo-inverse. Substituting (C.3.3) in (C.3.2) gives

$$\begin{aligned} \dot{V} &= \boldsymbol{\xi}_{h_i}^\top \mathbf{P}_{h_i} [\mathbf{L}_{h_i v_i}^{-1} \mathbf{L}_{v_i v_i^*}^{-1} \mathbf{L}_{v_i^* o_i}^{-1}]^+ (\dot{\mathbf{x}}_{h_i} - \dot{\mathbf{x}}_{o_i}) \\ &\quad + (\dot{\mathbf{x}}_{h_i}^\top - \dot{\mathbf{x}}_{o_i}^\top) [\mathbf{L}_{h_i v_i}^{-1} \mathbf{L}_{v_i v_i^*}^{-1} \mathbf{L}_{v_i^* o_i}^{-1}]^{+\top} \mathbf{P}_{h_i} \boldsymbol{\xi}_{h_i} \end{aligned}$$

Using the proposed law (4.3.3) with $\mathbf{C}_{h_i} = [\mathbf{L}_{h_i v_i}^{-1} \mathbf{L}_{v_i v_i^*}^{-1} \mathbf{L}_{v_i^* o_i}^{-1}] \mathbf{A}_{h_i}$ yields

$$\begin{aligned} \dot{V} &= \boldsymbol{\xi}_{h_i}^\top \mathbf{P}_{h_i} [\mathbf{L}_{h_i v_i}^{-1} \mathbf{L}_{v_i v_i^*}^{-1} \mathbf{L}_{v_i^* o_i}^{-1}]^+ [\mathbf{L}_{h_i v_i}^{-1} \mathbf{L}_{v_i v_i^*}^{-1} \mathbf{L}_{v_i^* o_i}^{-1}] \mathbf{A}_{h_i} \boldsymbol{\xi}_{h_i} \\ &\quad + \boldsymbol{\xi}_{h_i}^\top \mathbf{A}_{h_i}^\top [\mathbf{L}_{h_i v_i}^{-1} \mathbf{L}_{v_i v_i^*}^{-1} \mathbf{L}_{v_i^* o_i}^{-1}]^\top [\mathbf{L}_{h_i v_i}^{-1} \mathbf{L}_{v_i v_i^*}^{-1} \mathbf{L}_{v_i^* o_i}^{-1}]^{+\top} \mathbf{P}_{h_i} \boldsymbol{\xi}_{h_i} \\ &= \boldsymbol{\xi}_{h_i}^\top \mathbf{P}_{h_i} \mathbf{A}_{h_i} \boldsymbol{\xi}_{h_i} + \boldsymbol{\xi}_{h_i}^\top \mathbf{A}_{h_i}^\top \mathbf{P}_{h_i} \boldsymbol{\xi}_{h_i} \\ &= \boldsymbol{\xi}_{h_i}^\top \underbrace{(\mathbf{P}_{h_i} \mathbf{A}_{h_i} + \mathbf{A}_{h_i}^\top \mathbf{P}_{h_i})}_{-\mathbf{Q}_{h_i}} \boldsymbol{\xi}_{h_i} = -\boldsymbol{\xi}_{h_i}^\top \mathbf{Q}_{h_i} \boldsymbol{\xi}_{h_i} \prec 0 \end{aligned}$$

APPENDICES FOR CHAPTER 5

D.1 Orientation control

To control the orientation task, which consists of driving the current orientation of the h^{th} end-effector represented by the rotation matrix $\mathbf{R}_c^h \in \mathbb{R}^{3 \times 3}$ towards its desired value $\mathbf{R}_d^h \in \mathbb{R}^{3 \times 3}$, we define a state vector $\xi_\theta^h \in \mathbb{R}^3$ using the axis/angle representation of the relative orientation, ${}^d\mathbf{R}_c^h \triangleq (\mathbf{R}_d^h)^\top \mathbf{R}_c^h$. Hence, $\xi_\theta^h \triangleq \theta \boldsymbol{\mu}({}^d\mathbf{R}_c^h)$, where $\boldsymbol{\mu} \in \mathbb{R}^3$ and $\theta \in \mathbb{R}$ represent respectively the axis and the angle associated with the rotation matrix ${}^d\mathbf{R}_c^h$.

With ξ_θ^h defined as above, its desired value is located at the origin, that is $\xi_{\theta d}^h = 0$. Thus, similarly to the position task, if we assume a linear or linear parameters varying (LPV) DS for the orientation, we can write

$$\dot{\xi}_\theta^h = A_\theta(\xi_\theta^h - \xi_{\theta d}^h) = A_\theta \xi_\theta^h$$

where $A_\theta \in \mathbb{R}^{3 \times 3}$ is the dynamic matrix chosen to be negative definite ($A_\theta < 0$) to ensure asymptotic converges of ξ_θ^h towards its attractor $\mathbf{0}$ ($\lim_{t \rightarrow \infty} \xi_\theta^h = 0$). Such convergence indicates the matching \mathbf{R}_c^h with \mathbf{R}_d^h . The angular velocity associated with the orientation DS is obtained as follows

$$\omega^h = \mathbf{L}_{\xi^h}^{-1} \dot{\xi}_\theta^h = \mathbf{L}_{\xi^h}^{-1} A_\theta \xi_\theta^h$$

where $\mathbf{L}_{\xi^h} \triangleq \mathbf{L}_{\theta \boldsymbol{\mu}}^h (\mathbf{R}_c^h)^\top$ with $\mathbf{L}_{\theta \boldsymbol{\mu}}^h \in \mathbb{R}^{3 \times 3}$ a matrix mapping the angular velocity to the time derivative of orientation state vector ξ^h and given by [Malis et al. \(1999\)](#)

$$\mathbf{L}_{\theta \boldsymbol{\mu}}^h = \mathbf{I}_{3 \times 3} - \frac{\theta}{2} [\boldsymbol{\mu}^h]_\times + \left(1 - \frac{\text{sinc } \theta}{\text{sinc}^2 \frac{\theta}{2}} \right) [\boldsymbol{\mu}^h]_\times^2$$

where $\theta \text{sinc } \theta = \sin \theta$ and $[\boldsymbol{\mu}^h]_\times \in \mathbb{R}^{3 \times 3}$ denotes a skew-symmetric matrix associated with $\boldsymbol{\mu}^h$.

To coordinate the position and the orientation task, the latter was coupled to the position task using a state-depend coefficient $\eta(\mathbf{x})$ function of the error on the absolute position: $\eta(\mathbf{x}) = 1 - \exp\left(-\frac{\sigma}{\|\mathbf{x}^{abs} - \mathbf{x}_d^{abs}\| + \varepsilon}\right)$, where $\sigma > 0$ is a scalar that tunes how fast $\eta(\mathbf{x})$ varies within $[0, 1]$.

The orientation state vector ξ_θ^h is now computed as $\xi_\theta^h \triangleq \theta \mu(^*\mathbf{R}_c^h)$, with $^*\mathbf{R}_c^h \triangleq (\mathbf{R}_*^h(\eta))^\top \mathbf{R}_c^h$. Here $\mathbf{R}_*^h(\eta)$ denotes the rotation matrix computed from the spherical interpolation between a resting orientation \mathbf{R}_r^h and the desired orientation \mathbf{R}_d^h as function of $\eta(\mathbf{x})$. When $\eta(\mathbf{x}) \rightarrow 0$, $\mathbf{R}_*^h(\eta) \rightarrow \mathbf{R}_r^h$ and when $\eta(\mathbf{x}) \rightarrow 1$, $\mathbf{R}_*^h(\eta) \rightarrow \mathbf{R}_d^h$.

D.2 Determination of impact direction

To determine the impact direction, \vec{e}_1^h , that anticipates the desired motion of the object once grasped, we use the normal vector to the h^{th} contact denoted \vec{n}^h , and the desired object's effective force \mathbf{f}_o^d needed to realize the desired task to build an orthonormal basis $U^h = [\vec{u}_1^h \ \vec{u}_2^h \ \vec{u}_3^h]$ as

$$\vec{u}_1^h = \vec{n}^h; \quad \vec{u}_2^h = \vec{f}_o^d \times \vec{n}^h; \quad \text{and} \quad \vec{u}_3^h = \vec{u}_1^h \times \vec{u}_2^h$$

where $\vec{f}_o^d = \frac{\mathbf{f}_o^d}{\|\mathbf{f}_o^d\|}$ denotes a unitary vector in the direction of the desired object effective force. Then, we compute the angle formed by the impact direction with the normal to the surface as

$$\gamma^h = \arctan \left(\frac{\mathbf{f}_o^d \cdot \vec{u}_3^h}{\mathbf{f}_o^d \cdot \vec{u}_1^h} \right)$$

For stable contact, we have $0 \leq \tan(\gamma^h) \leq \tan(\gamma_{max}^h)$, with $\tan(\gamma_{max}^h) = \mu$, the coefficient of friction. Now, using the obtained angle γ^h and the basis U^h , the impact direction \vec{e}_1^h is computed as:

$$\vec{e}_1^h = U^h [\cos(\gamma^h) \ 0 \ \sin(\gamma^h)]^\top$$

D.3 Proof of Proposition 1

Substituting (5.4.2) in (5.3.2) and then in (5.3.1) and multiplying by $\text{diag}\{(\mathbf{e}_i^L)^\top, (\mathbf{e}_i^R)^\top\}$ gives

$$\begin{aligned} (\mathbf{e}_i^L)^\top \dot{\mathbf{x}}^L &= (\mathbf{e}_i^L)^\top E^L(\mathbf{x}) \Lambda^L(\mathbf{x}) (E^L(\mathbf{x}))^\top f_r^L(\mathbf{x}) \\ (\mathbf{e}_i^R)^\top \dot{\mathbf{x}}^R &= (\mathbf{e}_i^R)^\top E^R(\mathbf{x}) \Lambda^R(\mathbf{x}) (E^R(\mathbf{x}))^\top f_r^R(\mathbf{x}) \end{aligned}$$

Hence, for each $h = \{L, R\}$ component, we have¹

$$\begin{aligned} (\mathbf{e}_i^h)^\top \dot{\mathbf{x}}^h &= \sum_{j=1}^3 \lambda_{ij}^h(\mathbf{x}) (\mathbf{e}_j^h)^\top f_n^h(\mathbf{x}) \\ &= (\mathbf{e}_i^h)^\top f_{m_i}^h(\mathbf{x}) \underbrace{\sum_{j=1}^3 \frac{f_n^h(\mathbf{x})^\top \mathbf{e}_j^h}{f_n^h(\mathbf{x})^\top f_n^h(\mathbf{x})} (\mathbf{e}_j^h)^\top f_n^h(\mathbf{x})}_I \\ &= (\mathbf{e}_i^h)^\top f_{m_i}^h(\mathbf{x}) \end{aligned} \quad (\text{D.3.1})$$

Finally, substituting (5.4.1) in (D.3.1) yields

$$(\mathbf{e}_i^h)^\top \dot{\mathbf{x}}^h = (\mathbf{e}_i^h)^\top A'_h(\mathbf{x} - \mathbf{x}_*) \quad (\text{D.3.2})$$

which gives in terms of the left and right components

$$\begin{aligned} (\mathbf{e}_i^L)^\top \dot{\mathbf{x}}^L &= (\mathbf{e}_i^L)^\top [A'_{LL}(\mathbf{x}^L - \mathbf{x}_*^L) + A'_{LR}(\mathbf{x}^R - \mathbf{x}_*^R)] \\ (\mathbf{e}_i^R)^\top \dot{\mathbf{x}}^R &= (\mathbf{e}_i^R)^\top [A'_{RL}(\mathbf{x}^L - \mathbf{x}_*^L) + A'_{RR}(\mathbf{x}^R - \mathbf{x}_*^R)] \end{aligned} \quad (\text{D.3.3})$$

Clearly, Eq. (D.3.3) shows the two robots interaction, which is necessary to preserve the coordination. At the same time, all robots converge towards their attractors since at the equilibrium ($\dot{\mathbf{x}}^L = 0$ and $\dot{\mathbf{x}}^R = 0$), we have

$$0 = \begin{bmatrix} A'_{LL} & A'_{LR} \\ A'_{RL} & A'_{RR} \end{bmatrix} \begin{pmatrix} \mathbf{x}^L - \mathbf{x}_*^L \\ \mathbf{x}^R - \mathbf{x}_*^R \end{pmatrix} = A'(\mathbf{x} - \mathbf{x}_*)$$

This implies that $\mathbf{x} - \mathbf{x}_* = 0$ given that A' is full rank ■

D.4 Proof of proposition 2

Proving the first motion towards \mathbf{x}_t when $\gamma(\mathbf{x}) = 0$ is straightforward given (5.4.1) and similar expression for $f_{m_i}^h(\mathbf{x})$, since $\mathbf{x}_* = \mathbf{x}_t$.

However, when $\gamma(\mathbf{x}) = 1$ the attractor becomes $\mathbf{x}_* = \mathbf{x} - A'^{-1} \dot{\mathbf{x}}_d$ and when substituted in (D.3.2), the later becomes

¹where we use the following simplification

$$\sum_{j=1}^3 \frac{f_n^h(\mathbf{x})^\top \mathbf{e}_j^h}{f_n^h(\mathbf{x})^\top f_n^h(\mathbf{x})} (\mathbf{e}_i^h)^\top f_n^h(\mathbf{x}) = \frac{f_n^h(\mathbf{x})^\top \sum_{j=1}^3 \mathbf{e}_j^h (\mathbf{e}_j^h)^\top f_n^h(\mathbf{x})}{f_n^h(\mathbf{x})^\top f_n^h(\mathbf{x})} = I$$

$$(\mathbf{e}_1^h)^\top \dot{\mathbf{x}}^h = (\mathbf{e}_1^L)^\top \left[A'_h(A')^{-1} \begin{pmatrix} \dot{\mathbf{x}}_d^L \\ \dot{\mathbf{x}}_d^R \end{pmatrix} \right]$$

Given that $A'(A')^{-1} = \mathbf{I} \in \mathbb{R}^{6 \times 6}$ then

$$A'_L(A')^{-1} = [\mathbf{I} \quad \mathbf{0}] \quad \text{and} \quad A'_R(A')^{-1} = [\mathbf{0} \quad \mathbf{I}]$$

Therefore, the generated velocities along \mathbf{e}_1^L and \mathbf{e}_1^R for the two robots become

$$(\mathbf{e}_1^L)^\top \dot{\mathbf{x}}^L = (\mathbf{e}_1^L)^\top \dot{\mathbf{x}}_d^L \quad \text{and} \quad (\mathbf{e}_1^R)^\top \dot{\mathbf{x}}^R = (\mathbf{e}_1^R)^\top \dot{\mathbf{x}}_d^R$$

whereas for \mathbf{e}_i^h with $i = 2$ and 3 , the dynamics will remain $(\mathbf{e}_i^h)^\top \dot{\mathbf{x}}^h = (\mathbf{e}_i^h)^\top A'_h(\mathbf{x} - \mathbf{x}_t)$. ■

D.5 Proof of Proposition 3

Following the definition of T_b in section 5.4, when the DS, as shown in *Proposition 1*, asymptotically converges to its attractor, we have $\mathbf{x} = \mathbf{x}_*$. When using \mathbf{x}_* as defined in (5.4.4) we have

$$T_b^{-1} \begin{bmatrix} \mathbf{x}_d^{abs} \\ \mathbf{x}_d^{rel} \end{bmatrix} = T_b^{-1} \begin{bmatrix} \mathbf{x}^{abs} + (\mathbf{x}_d^o - \mathbf{x}^o) \\ \mathbf{x}_o^R - \mathbf{x}_o^L \end{bmatrix}$$

Rewriting and simplifying the previous equation yields

$$\begin{aligned} (\mathbf{x}_d^{abs} - \mathbf{x}_d^o) - (\mathbf{x}^{abs} - \mathbf{x}^o) &= 0 \\ \mathbf{x}_d^{rel} - (\mathbf{x}_o^R - \mathbf{x}_o^L) &= 0 \end{aligned}$$

The above expression is zero as it represents the difference between the offset $(\mathbf{x}^{abs} - \mathbf{x}^o)$ respectively the relative end-effectors position $(\mathbf{x}_o^R - \mathbf{x}_o^L) = \mathbf{x}^{rel}$ and their values after convergence. For stable grasp these quantities remain constant throughout the task and therefore their difference is 0. ■

D.6 Joint limits

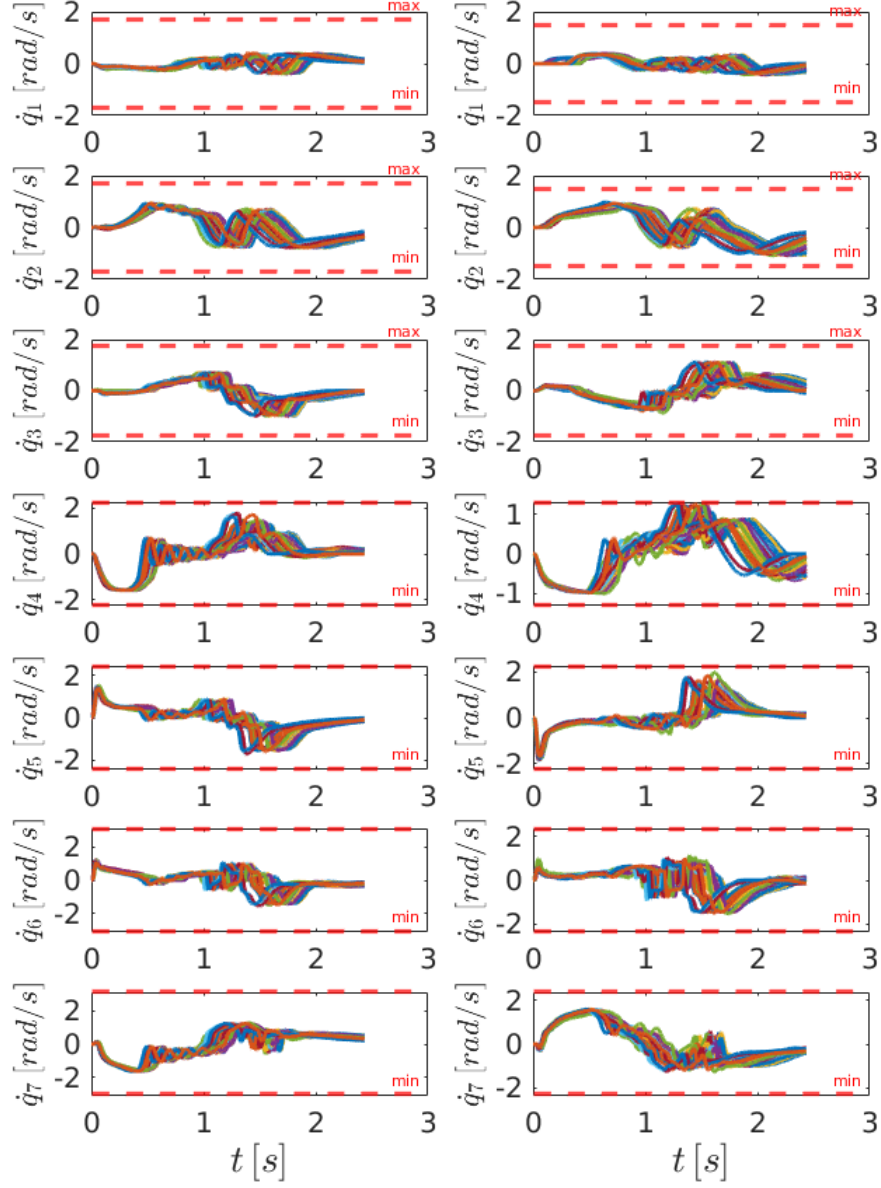


Figure D.1: Example of joint velocity limits for 30 experiments. The left subplots represent the limits of the left arm while the right subplots represent the limits of the right arm. The red horizontal dashed line represents the upper and lower limits.

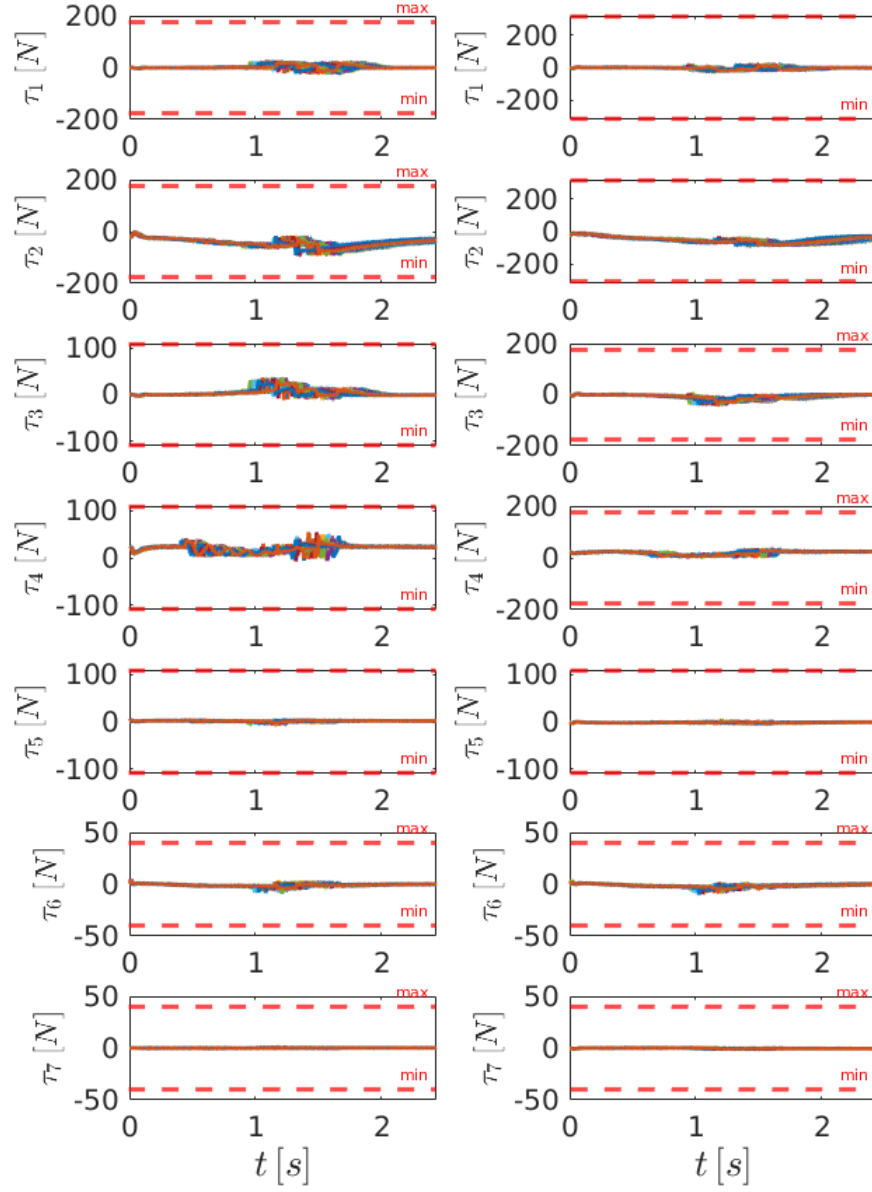


Figure D.2: Example of joint torque limits for 30 experiments. The left subplots represent the limits of the left arm while the right subplots represent the limits of the right arm. The red horizontal dashed line represents the upper and lower limits.

APPENDICES FOR CHAPTER 6

E.1 Jacobian of Inverse throwing map

To account for objects with different mass and aerodynamic properties, the inverse throwing map introduced in Section 6.3.1 is parameterized by η and can be written as

$$\mathbf{v}_r \approx \sum_{k=1}^K h^k(\bar{\mathbf{x}}^o; \eta) \tilde{\boldsymbol{\mu}}_{\bar{\mathbf{x}}^o | (\bar{\mathbf{x}}^o; \eta)}^k(\bar{\mathbf{x}}^o; \eta) \quad (\text{E.1.1})$$

where $\tilde{\boldsymbol{\mu}}_{\bar{\mathbf{x}}^o | (\bar{\mathbf{x}}^o; \eta)}^k = \boldsymbol{\mu}_{\bar{\mathbf{x}}^o}^k + \Sigma_{\bar{\mathbf{x}}^o \bar{\mathbf{x}}^o}^k \Phi_{\bar{\mathbf{x}}^o \bar{\mathbf{x}}^o}^k (\bar{\mathbf{x}}^o - \boldsymbol{\mu}_{\bar{\mathbf{x}}^o}^k) + \xi_{\bar{\mathbf{x}}^o \eta}^k$ and where $\xi_{\bar{\mathbf{x}}^o \eta}^k$ is defined as

$$\begin{aligned} \xi_{\bar{\mathbf{x}}^o \eta}^k &= \Sigma_{\bar{\mathbf{x}}^o \eta}^k \Phi_{\eta \bar{\mathbf{x}}^o}^k (\bar{\mathbf{x}}^o - \boldsymbol{\mu}_{\bar{\mathbf{x}}^o}^k) \\ &+ [\Sigma_{\bar{\mathbf{x}}^o \bar{\mathbf{x}}^o}^k \Phi_{\bar{\mathbf{x}}^o \eta}^k + \Sigma_{\bar{\mathbf{x}}^o \eta}^k \Phi_{\eta \eta}^k] (\eta - \boldsymbol{\mu}_{\eta}^k) \end{aligned}$$

The Jacobian of \mathbf{v}_r with respect to $\bar{\mathbf{x}}^o$ and parametrized by η can be written as

$$\mathbf{J}_{\mathbf{v}_r}(\bar{\mathbf{x}}^o) = \frac{\partial \mathbf{v}_r}{\partial \bar{\mathbf{x}}^o} = \sum_{k=1}^K \left[(c_{\mathbf{v} \bar{\mathbf{x}}}^k + S_{\mathbf{v} \bar{\mathbf{x}}}^k \bar{\mathbf{x}}^o) \frac{\partial h^k(\bar{\mathbf{x}}^o)}{\partial \bar{\mathbf{x}}^o} + h^k(\bar{\mathbf{x}}^o) S_{\mathbf{v} \bar{\mathbf{x}}}^k \right] \quad (\text{E.1.2})$$

where the matrices $S_{\mathbf{v} \bar{\mathbf{x}}}^k$ and vectors $c_{\mathbf{v} \bar{\mathbf{x}}}^k$ are given by

$$\begin{aligned} S_{\mathbf{v} \bar{\mathbf{x}}}^k &= \Sigma_{\bar{\mathbf{x}}^o \bar{\mathbf{x}}^o}^k \Phi_{\bar{\mathbf{x}}^o \bar{\mathbf{x}}^o}^k + \Xi_{\mathbf{v} \eta}^k \\ c_{\mathbf{v} \bar{\mathbf{x}}}^k &= \boldsymbol{\mu}_{\bar{\mathbf{x}}^o}^k - S_{\mathbf{v} \bar{\mathbf{x}}}^k \boldsymbol{\mu}_{\bar{\mathbf{x}}^o}^k + \zeta_{\mathbf{v} \eta}^k \end{aligned}$$

with

$$\begin{aligned} \Xi_{\mathbf{v} \eta}^k &= \Sigma_{\bar{\mathbf{x}}^o \eta}^k \Phi_{\eta \bar{\mathbf{x}}^o}^k \\ \zeta_{\mathbf{v} \eta}^k &= S_{\mathbf{v} \eta}^k (\eta - \boldsymbol{\mu}_{\eta}^k) \end{aligned}$$

where $S_{\mathbf{v} \eta}^k = \Sigma_{\bar{\mathbf{x}}^o \bar{\mathbf{x}}^o}^k \Phi_{\bar{\mathbf{x}}^o \eta}^k + \Sigma_{\bar{\mathbf{x}}^o \eta}^k \Phi_{\eta \eta}^k$ with the Φ_{ii}^k defined as

$$(\Sigma_{\bar{\mathbf{x}}}^k)^{-1} = \begin{bmatrix} \Sigma_{\bar{\mathbf{x}}^o \bar{\mathbf{x}}^o}^k & \Sigma_{\bar{\mathbf{x}}^o \eta}^k \\ \Sigma_{\eta \bar{\mathbf{x}}^o}^k & \Sigma_{\eta \eta}^k \end{bmatrix}^{-1} \triangleq \begin{bmatrix} \Phi_{\bar{\mathbf{x}}^o \bar{\mathbf{x}}^o}^k & \Phi_{\bar{\mathbf{x}}^o \eta}^k \\ \Phi_{\eta \bar{\mathbf{x}}^o}^k & \Phi_{\eta \eta}^k \end{bmatrix} \quad (\text{E.1.3})$$

The expressions of $\pi^k(\bar{\mathbf{x}})$ and $\frac{\partial \pi^k(\bar{\mathbf{x}})}{\partial \bar{\mathbf{x}}}$ in the Jacobian are computed as follows

$$h^k(\bar{\mathbf{x}}^o) = \frac{\alpha^k p(\bar{\mathbf{x}}^o | \boldsymbol{\mu}_{\bar{\mathbf{x}}}^k, \Sigma_{\bar{\mathbf{x}}}^k)}{\sum_{k=1}^K \alpha^k p(\bar{\mathbf{x}}^o | \boldsymbol{\mu}_{\bar{\mathbf{x}}}^k, \Sigma_{\bar{\mathbf{x}}}^k)} \quad (\text{E.1.4})$$

and

$$\frac{\partial h^k(\bar{\mathbf{x}}^o)}{\partial \bar{\mathbf{x}}^o} = \frac{1}{D^2} \left(\frac{\partial N}{\partial \bar{\mathbf{x}}^o} \cdot D - N \cdot \frac{\partial D}{\partial \bar{\mathbf{x}}^o} \right) \quad (\text{E.1.5})$$

where the terms $N(\bar{\mathbf{x}})$, $D(\bar{\mathbf{x}})$, $\frac{\partial N^k}{\partial \bar{\mathbf{x}}}$, and $\frac{\partial D^k}{\partial \bar{\mathbf{x}}}$ are respectively given by

$$N(\bar{\mathbf{x}}^o) = \frac{\alpha^k}{(2\pi)^{\frac{N}{2}} |\Sigma^k|^{\frac{1}{2}}} e^{-\frac{1}{2} (\chi_\mu^k)^\top (\Sigma^k)^{-1} (\chi_\mu^k)} \quad (\text{E.1.6})$$

$$D(\bar{\mathbf{x}}^o) = \sum_{k=1}^K \frac{\alpha^k}{(2\pi)^{\frac{N}{2}} |\Sigma^k|^{\frac{1}{2}}} e^{-\frac{1}{2} (\chi_\mu^k)^\top (\Sigma^k)^{-1} (\chi_\mu^k)} \quad (\text{E.1.7})$$

$$\frac{\partial N^k}{\partial \bar{\mathbf{x}}^o} = -\frac{\alpha^k}{(2\pi)^{\frac{N}{2}} |\Sigma^k|^{\frac{1}{2}}} e^{-\frac{1}{2} (\chi_\mu^k)^\top (\Sigma^k)^{-1} (\chi_\mu^k)} (\chi_\mu^k)^\top (\Sigma^k)^{-1} S_{\bar{\mathbf{x}}} \quad (\text{E.1.8})$$

$$\frac{\partial D^k}{\partial \bar{\mathbf{x}}^o} = -\frac{1}{(2\pi)^{\frac{N}{2}}} \sum_{k=1}^K \frac{\alpha^k}{|\Sigma^k|^{\frac{1}{2}}} e^{-\frac{1}{2} (\chi_\mu^k)^\top (\Sigma^k)^{-1} (\chi_\mu^k)} (\chi_\mu^k)^\top (\Sigma^k)^{-1} S_{\bar{\mathbf{x}}} \quad (\text{E.1.9})$$

where \mathbf{x}_μ^k is defined as $\chi_\mu^k = \begin{bmatrix} \bar{\mathbf{x}}^o - \boldsymbol{\mu}_{\bar{\mathbf{x}}^o}^k \\ \eta - \boldsymbol{\mu}_\eta^k \end{bmatrix}$ and $S_{\bar{\mathbf{x}}} = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 1} \\ 0_{1 \times 3} & 0_{1 \times 1} \end{bmatrix}$

E.2 Throwing objectives

In the learning the inverse throwing map, to resolve the inherent redundancy problem by adopting a throwing strategy that seeks the minimal throwing velocity. Here we compare it with other strategies such as minimum landing vertical velocity and minimal landing horizontal velocity.

The comparison results are shown in Table E.1, where the top two images concerned the minimum horizontal landing speed and the bottom two images concerned the minimum vertical landing speed. Although these two approaches may minimize the landing impact of the object, they require however larger release speeds.

Table E.1: Illustration of alternative throwing objectives to address the redundancy problem in a planar throwing task with given release and landing positions. (top): minimal landing horizontal speed. (bottom): minimum landing vertical speed

Objective	v_0	v_l
$\min(\dot{r}_l)$		
$\min(\dot{z}_l)$		

REFERENCES

- Eric W Aboaf, Christopher G Atkeson, and David J Reinkensmeyer. Task-level robot learning. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 1309–1310. IEEE, 1988.
- Eric W Aboaf, Steven Mark Drucker, and Christopher G Atkeson. Task-level robot learning: Juggling a tennis ball more accurately. In *Proceedings, 1989 International Conference on Robotics and Automation*, pages 1290–1295. IEEE, 1989.
- Leopoldo Acosta, JJ Rodrigo, Juan A Mendez, G Nicolás Marichal, and Marta Sigut. Ping-pong player prototype. *IEEE robotics & automation magazine*, 10(4):44–52, 2003.
- Farhad Aghili. A unified approach for inverse and direct dynamics of constrained multibody systems based on linear projection operator: applications to control and simulation. *IEEE Transactions on Robotics*, 21(5):834–849, 2005.
- Adel Akbarimajd and Majid Nili Ahmadabadi. Manipulation by juggling of planar polygonal objects using two 3-dof manipulators. In *2007 IEEE/ASME international conference on advanced intelligent mechatronics*, pages 1–6. IEEE, 2007.
- Iretiayo Akinola, Jingxi Xu, Shuran Song, and Peter K Allen. Dynamic grasping with reachability and motion awareness. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9422–9429. IEEE, 2021.
- Peter K Allen, Aleksandar Timcenko, Billibon Yoshimi, and Paul Michelman. Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Transactions on Robotics and Automation*, 9(2):152–165, 1993.
- Walid Amanhoud, Mahdi Khoramshahi, and Aude Billard. A dynamical system approach to motion and force generation in contact tasks. In *Proceedings of Robotics: Science and Systems*, Freiburg/Breisgau, Germany, June 2019. doi: 10.15607/RSS.2019.XV.021.
- Russell L Anderson. *A robot ping-pong player: experiment in real-time intelligent control*. MIT press, 1988.
- Tamim Asfour, Lukas Kaul, Mirko Wächter, Simon Ottenhaus, Pascal Weiner,

- Samuel Rader, Raphael Grimm, You Zhou, Markus Grotz, Fabian Paus, et al. Armar-6: A collaborative humanoid robot for industrial environments. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 447–454. IEEE, 2018.
- Mohsen Asgari and Amin Nikoobin. A variational approach to determination of maximum throw-able workspace of robotic manipulators in optimal ball pitching motion. *Transactions of the Institute of Measurement and Control*, 43(10):2378–2391, 2021.
- Wilhelm August, Steffen Waeldele, Bjoern Hein, Heinz Woern, and G Wyeth. Accurate object throwing by an industrial robot manipulator. In *Proceedings of the Australasian Conference on Robotics and Automation 2010, ACRA 10, Brisbane, Australia*, pages 74–81, 2010.
- Georg Bätz, Arhan Yaqub, Haiyan Wu, Kolja Kühnlenz, Dirk Wollherr, and Martin Buss. Dynamic manipulation: Nonprehensile ball catching. In *18th Mediterranean Conference on Control and Automation, MED’10*, pages 365–370. IEEE, 2010.
- Antonio Bicchi and Vijay Kumar. Robotic grasping and contact: A review. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 1, pages 348–353. IEEE, 2000.
- Bigcommerce. Ecommerce 101: The history and future of online shopping, May 2022. URL <https://www.bigcommerce.com/articles/ecommerce/>.
- Paul T Boggs and Jon W Tolle. Sequential quadratic programming. *Acta numerica*, 4:1–51, 1995.
- Michael Bombile. Visual servoing based positioning and object tracking on humanoid robot. In *New Trends in Networking, Computing, E-learning, Systems Sciences, and Engineering*, pages 19–27. Springer, 2015a.
- Michael Bombile. Visual servo control on a humanoid robot. Master’s thesis, University of Cape Town, 2015b.
- Michael Bombile and Aude Billard. Capture-point based balance and reactive omnidirectional walking controller. In *IEEE RAS International Conference on Humanoid Robots*, number EPFL-CONF-231920, 2017a.
- Michael Bombile and Aude Billard. Capture-point based balance and reactive omnidirectional walking controller. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 17–24. IEEE, 2017b.
- Michael Bosongo Bombile and Aude Billard. Dual-arm control for coordinated fast grabbing and tossing of an object: Proposing a new approach. *IEEE Robotics & Automation Magazine*, 2022.

- RC Bonitz and Tien C Hsia. Internal force-based impedance control for cooperating manipulators. *IEEE Transactions on Robotics and Automation*, 12(1): 78–89, 1996a.
- Robert G Bonitz and Tien C Hsia. Robust dual-arm manipulation of rigid objects via palm grasping-theory and experiments. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 4, pages 3047–3054. IEEE, 1996b.
- Karim Bouyarmane, Joris Vaillant, Kévin Chappellet, and Abderrahmane Kheddar. Multi-robot and task-space force control with quadratic programming. 2017.
- Martin Buhler, Daniel E Koditschek, and Peter J Kindlmann. A family of robot control strategies for intermittent dynamical environments. *IEEE Control Systems Magazine*, 10(2):16–22, 1990.
- Robert R Burridge, Alfred A Rizzi, and Daniel E Koditschek. Toward a dynamical pick and place. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 2, pages 292–297. IEEE, 1995.
- Fabrizio Caccavale and Masaru Uchiyama. Cooperative manipulators. In *Springer handbook of robotics*, pages 701–718. Springer, 2008.
- Fabrizio Caccavale and Masaru Uchiyama. Cooperative manipulation. In *Springer Handbook of Robotics*, pages 989–1006. Springer, 2016.
- Fabrizio Caccavale, Pasquale Chiacchio, Alessandro Marino, and Luigi Villani. Six-dof impedance control of dual-arm cooperative manipulators. *IEEE/ASME Transactions On Mechatronics*, 13(5):576–586, 2008.
- Stephanie Chevalier. Global retail e-commerce market size 2014-2023. <https://www.statista.com/statistics/728530/industrial-robot-market-size-worldwide/>, Feb 2022. URL <https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/>.
- Pasquale Chiacchio, Stefano Chiaverini, Lorenzo Sciavicco, and Bruno Siciliano. Global task space manipulability ellipsoids for multiple-arm systems. *IEEE Transactions on Robotics and Automation*, 7(5):678–685, 1991.
- Benoît Colson, Patrice Marcotte, and Gilles Savard. An overview of bilevel optimization. *Annals of operations research*, 153(1):235–256, 2007.
- John J Craig. *Introduction to robotics: mechanics and control*. Pearson Education, 2005.
- Elizabeth A Croft, Robert G Fenton, and Beno Benhabib. Time-optimal interception of objects moving along predictable paths. In *Proceedings. IEEE*

- International Symposium on Assembly and Task Planning*, pages 419–425. IEEE, 1995.
- Elizabeth A Croft, Robert G Fenton, and Beno Benhabib. Optimal rendezvous-point selection for robotic interception of moving objects. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 28(2):192–204, 1998.
- Hongkai Dai, Andrés Valenzuela, and Russ Tedrake. Whole-body motion planning with centroidal dynamics and full kinematics. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 295–302. IEEE, 2014.
- Martin De Lasa, Igor Mordatch, and Aaron Hertzmann. Feature-based locomotion controllers. In *ACM Transactions on Graphics (TOG)*, volume 29, page 131. ACM, 2010.
- N. Dehio, J. Smith, D. L. Wigand, G. Xin, H. Lin, J. J. Steil, and M. Mistry. Modeling and control of multi-arm and multi-leg robots: Compensating for object dynamics during grasping. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 294–301, May 2018. doi: 10.1109/ICRA.2018.8462872.
- Niels Dehio and Abderrahmane Kheddar. Robot-Safe Impacts with Soft Contacts Based on Learned Deformations. In *ICRA*, Xi’an, China, May 2021. URL <https://hal.archives-ouvertes.fr/hal-02973947>.
- Niels Dehio, Yuquan Wang, and Abderrahmane Kheddar. Dual-arm box grabbing with impact-aware mpc utilizing soft deformable end-effector pads. *IEEE Robotics and Automation Letters*, 7(2):5647–5654, 2022.
- H. Diedam, D. Dimitrov, P.-B. Wieber, K. Mombaur, and M. Diehl. Online walking gait generation with adaptive foot positioning through Linear Model Predictive control. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1121–1126, September 2008. doi: 10.1109/IROS.2008.4651055.
- Ke Dong, Karime Pereida, Florian Shkurti, and Angela P Schoellig. Catch the ball: Accurate high-speed motions for mobile manipulators via inverse dynamics learning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6718–6725. IEEE, 2020.
- Johannes Engelsberger and Christian Ott. Walking stabilization for humanoid robots based on control of the capture point. *AT-AUTOMATISIERUNGSTECHNIK*, 60(11):692–703, 2012a.
- Johannes Engelsberger and Christian Ott. Integration of vertical com motion and angular momentum in an extended capture point tracking controller for bipedal walking. In *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, pages 183–189. IEEE, 2012b.
- Johannes Engelsberger, Christian Ott, Máximo A Roa, Alin Albu-Schäffer, and

- Gerhard Hirzinger. Bipedal walking control based on capture point dynamics. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4420–4427. IEEE, 2011.
- Sebastian Erhart, Dominik Sieber, and Sandra Hirche. An impedance-based control architecture for multi-robot cooperative dual-arm mobile manipulation. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 315–322, 2013.
- Adrien Escande, Nicolas Mansard, and Pierre-Brice Wieber. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 33(7):1006–1028, 2014.
- Hans Jacob S Feder and J-JE Slotine. Real-time path planning using harmonic potentials in dynamic environments. In *Proceedings of International Conference on Robotics and Automation*, volume 1, pages 874–881. IEEE, 1997.
- Siyuan Feng, Eric Whitman, X Xinjilefu, and Christopher G Atkeson. Optimization-based full body control for the darpa robotics challenge. *Journal of field robotics*, 32(2):293–312, 2015.
- H.J. Ferreau, C. Kirches, A. Potschka, H.G. Bock, and M. Diehl. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363, 2014.
- Nadia Figueroa and Aude Billard. Locally active globally stable dynamical systems: Theory, learning, and experiments. *The International Journal of Robotics Research*, 41(3):312–347, 2022. doi: 10.1177/02783649211030952. URL <https://doi.org/10.1177/02783649211030952>.
- Heinz Frank. Design and simulation of a numerical controlled throwing device. In *2008 Second Asia International Conference on Modelling & Simulation (AMS)*, pages 777–782. IEEE, 2008a.
- Heinz Frank. Determination of launching parameters for throwing objects in logistic processes with direct hits. In *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, pages 58–61. IEEE, 2008b.
- Heinz Frank, Norbert Wellerdick-Wojtasik, Bianca Hagebeuker, Gregor Novak, and Stefan Mahlknecht. Throwing objects—a bio-inspired approach for the transportation of parts. In *2006 IEEE International Conference on Robotics and Biomimetics*, pages 91–96. IEEE, 2006.
- Heinz Frank, Anton Mittnacht, and Johann Scheiermann. Throwing of cylinder-shaped objects. In *2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 59–64. IEEE, 2009.
- André Gallant. *Optimisation de trajectoire pour l’augmentation des capacités des manipulateurs robotiques*. PhD thesis, Université Laval, 2020.

- J. Gao, Y. Zhou, and T. Asfour. Projected force-admittance control for compliant bimanual tasks. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 1–9, Nov 2018. doi: 10.1109/HUMANOIDS.2018.8624916.
- Juan Miguel Garcia-Haro, Bernd Henze, George Mesesan, Santiago Martinez, and Christian Ott. Integration of dual-arm manipulation in a passivity based whole-body controller for torque-controlled humanoid robots. In *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pages 644–650. IEEE, 2019.
- Michael Gienger, Marc Toussaint, and Christian Goerick. Task maps in humanoid robot manipulation. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2758–2764. IEEE, 2008.
- Robert J Griffin and Alexander Leonessa. Model predictive control for dynamic footstep adjustment using the divergent component of motion. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1763–1768. IEEE, 2016.
- Ghina Hassan, Marc Gouttefarde, Ahmed Chemori, Pierre-Elie Hervé, Maher El Rafei, Clovis Francis, and Damien Sallé. Time-optimal pick-and-throw s-curve trajectories for fast parallel robots. *IEEE/ASME Transactions on Mechatronics*, 2022.
- A Herdt, N Perrin, and Pierre-Brice Wieber. Walking without thinking about it. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 190–195, October 2010a. doi: 10.1109/IROS.2010.5654429.
- Andrei Herdt, Holger Diedam, Pierre-Brice Wieber, Dimitar Dimitrov, Katja Mombaur, and Moritz Diehl. Online walking motion generation with automatic footstep placement. *Advanced Robotics*, 24(5-6):719–737, 2010b.
- Alexander Herzog, Nicholas Rotella, Sean Mason, Felix Grimmering, Stefan Schaal, and Ludovic Righetti. Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid. *Autonomous Robots*, 40(3):473–491, 2016.
- At L Hof. The ‘extrapolated center of mass’ concept suggests a simple control of balance in walking. *Human movement science*, 27(1):112–125, 2008.
- Enrico Mingo Hoffman, Brice Clément, Chengxu Zhou, Nikos G Tsagarakis, Jean-Baptiste Mouret, and Serena Ivaldi. Whole-body compliant control of icub: first results with opensot. In *IEEE/RAS ICRA Workshop on Dynamic Legged Locomotion in Realistic Terrains*, 2018.
- N. Hogan. Impedance control: An approach to manipulation. In *1984 American Control Conference*, pages 304–313, June 1984. doi: 10.23919/ACC.1984.4788393.
- Steven W Holland, Lothar Rossol, and Mitchell R Ward. Consight-i: a vision-

- controlled robot system for transferring parts from belt conveyors. In *Computer Vision and Sensor-Based Robots*, pages 81–100. Springer, 1979.
- Barbara Hove and Jean-Jacques E Slotine. Experiments in robotic catching. In *1991 American Control Conference*, pages 380–386. IEEE, 1991.
- Ping Hsu. Coordinated control of multiple manipulator systems. *IEEE Transactions on Robotics and Automation*, 9(4):400–410, 1993.
- Ping Hsu and Steven Su. Coordinated control of multiple manipulator systems-experimental results. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pages 2199–2204. IEEE, 1992.
- L. Huber, A. Billard, and J.-J. Slotine. Avoidance of convex and concave obstacles with convergence ensured through contraction. *IEEE Robotics and Automation Letters*, 2019a.
- Lukas Huber, Aude Billard, and Jean-Jacques Slotine. Avoidance of convex and concave obstacles with convergence ensured through contraction. *IEEE Robotics and Automation Letters*, 4(2):1462–1469, 2019b.
- Shigenori Ichinose, Shunsuke Katsumata, Shigeki Nakaura, and Mitsuji Sampei. Throwing motion control experiment utilizing 2-link arm passive joint. In *2008 SICE Annual Conference*, pages 3256–3261. IEEE, 2008.
- Auke Jan Ijspeert, Jun Nakanishi, Tomohiro Shibata, and Stefan Schaal. Non-linear dynamical systems for imitation with humanoid robots. In *Proceedings of the IEEE/RAS International Conference on Humanoids Robots (Humanoids2001)*, number CONF, pages 219–226, 2001.
- Yan-Bin Jia, Matthew Gardner, and Xiaoqian Mu. Batting an in-flight object to the target. *The International Journal of Robotics Research*, 38(4):451–485, 2019.
- S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. *2003 IEEE International Conference on Robotics and Automation*, pages 1620–1626, 2003. doi: 10.1109/ROBOT.2003.1241826.
- Shuuji Kajita and Bernard Espiau. Legged robot, 2008.
- Shuuji Kajita, Fumio Kanehiro, Kenji Kando, Kazuhito Yokoi, and Hirohisa Hirukawa. The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation. pages 239–246, 2001.
- Shuuji Kajita, Mitsuharu Morisawa, Kensuke Harada, Kenji Kaneko, Fumio Kanehiro, Kiyoshi Fujiwara, and Hirohisa Hirukawa. Biped Walking Pattern Generator allowing Auxiliary ZMP Control. *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2:2993–2999, October 2006. doi: 10.1109/IROS.2006.282233.

- Oussama Kanoun, Florent Lamiraux, and Pierre-Brice Wieber. Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task. *IEEE Transactions on Robotics*, 27(4):785–792, 2011.
- Lydia E Kavraki and Steven M LaValle. Motion planning. In *Springer Handbook of Robotics*, pages 139–162. Springer, 2016.
- JA Scott Kelso. *Dynamic patterns: The self-organization of brain and behavior*. MIT press, 1995.
- S Mohammad Khansari-Zadeh and Aude Billard. Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics*, 27(5):943–957, 2011.
- Seyed Mohammad Khansari-Zadeh and Aude Billard. A dynamical system approach to realtime obstacle avoidance. *Autonomous Robots*, 32(4):433–454, 2012.
- Oussama Khatib. Inertial properties in robotic manipulation: An object-level framework. *The international journal of robotics research*, 14(1):19–36, 1995.
- Oussama Khatib, Kazu Yokoi, Kyong I. Chang, Diego C. Ruspini, Robert Holmberg, and Arancha Casal. Coordination and decentralized cooperation of multiple mobile manipulators. *J. Field Robotics*, 13:755–764, 1996.
- Oussama Khatib, Kazu Yokoi, Oliver Brock, K Chang, and Arancha Casal. Robots in human environments: Basic autonomous capabilities. *The International Journal of Robotics Research*, 18(7):684–696, 1999.
- Oussama Khatib, Luis Sentis, Jaeheung Park, and James Warren. Whole-body dynamic behavior and control of human-like robots. *International Journal of Humanoid Robotics*, 1(01):29–43, 2004.
- Mahdi Khoramshahi and Aude Billard. A dynamical system approach to task-adaptation in physical human–robot interaction. *Autonomous Robots*, 43(4): 927–946, 2019.
- Joo H Kim, Yujiang Xiang, Rajankumar Bhatt, Jingzhou Yang, Jasbir S Arora, and Karim Abdel-Malek. Throwing motion generation of a biped human model. In *2008 2nd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 587–592. IEEE, 2008.
- Seungsu Kim and Stéphane Doncieux. Learning highly diverse robot throwing movements through quality diversity search. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1177–1178, 2017.
- Seungsu Kim, Ashwini Shukla, and Aude Billard. Catching objects in flight. *IEEE Transactions on Robotics*, 30(5):1049–1065, 2014.
- Jens Kober, Erhan Oztop, and Jan Peters. Reinforcement learning to adjust

- robot movements to new situations. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.
- Atsushi Konno, Tomoya Myojin, Takaaki Matsumoto, Teppei Tsujita, and Masaru Uchiyama. An impact dynamics model and sequential optimization to generate impact motions for a humanoid robot. *The International Journal of Robotics Research*, 30(13):1596–1608, 2011.
- Twan Koolen, Tomas De Boer, John Rebula, Ambarish Goswami, and Jerry Pratt. Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models. *The International Journal of Robotics Research*, 31(9):1094–1113, 2012.
- Manuel Krause, Johannes Engelsberger, Pierre-Brice Wieber, and Christian Ott. Stabilization of the capture point dynamics for bipedal walking based on model predictive control. In *Robot Control*, volume 10, pages 165–171, 2012.
- Klas Kronander and Aude Billard. Passive interaction control with dynamical systems. *IEEE Robotics and Automation Letters*, 1(1):106–113, 2016.
- Klas Kronander, Mohammad Khansari, and Aude Billard. Incremental motion learning with locally modulated dynamical systems. *Robotics and Autonomous Systems*, 70:52–62, 2015.
- Theodore Laetsch and JB Keller. The number of solutions of a nonlinear two point boundary value problem. *Indiana University Mathematics Journal*, 20(1):1–13, 1970.
- CH Lai and TI James Tsay. Self-learning for a humanoid robotic ping-pong player. *Advanced Robotics*, 25(9-10):1183–1208, 2011.
- Puttichai Lertkultanon and Quang-Cuong Pham. A certified-complete bimanual manipulation planner. *IEEE Transactions on Automation Science and Engineering*, 15(3):1355–1368, 2018.
- Yanmei Li and Imin Kao. A review of modeling of soft-contact fingers and stiffness control for dextrous manipulation in robotics. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 3, pages 3055–3060. IEEE, 2001.
- Hsiu-Chin Lin, Joshua Smith, Keyhan Kouhkiloui Babarahmati, Niels Dehio, and Michael Mistry. A projected inverse dynamics approach for multi-arm cartesian impedance control. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–5, 2018.
- Kevin M Lynch and Craig K Black. Recurrence, controllability, and stabilization of juggling. *IEEE Transactions on Robotics and Automation*, 17(2):113–124, 2001.
- Kevin M Lynch and Matthew T Mason. Dynamic nonprehensile manipula-

- tion: Controllability, planning, and experiments. *The International Journal of Robotics Research*, 18(1):64–92, 1999.
- E. Malis, F. Chaumette, and S. Boudet. 2 1/2 d visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2):238–250, April 1999. ISSN 1042-296X. doi: 10.1109/70.760345.
- Nicolas Mansard. A dedicated solver for fast operational-space inverse dynamics. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4943–4949. IEEE, 2012.
- Matthew T Mason and Kevin M Lynch. Dynamic manipulation. In *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’93)*, volume 1, pages 152–159. IEEE, 1993.
- Matthew T Mason and J Kenneth Salisbury Jr. Robot hands and the mechanics of manipulation. 1985.
- Jacob Mattingley and Stephen Boyd. Cvxgen: A code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, 2012.
- Mehran Mehrandezh, MN Sela, Robert G Fenton, and Beno Benhabib. Proportional navigation guidance in robot trajectory planning for intercepting moving objects. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, volume 1, pages 145–150. IEEE, 1999.
- Giorgio Metta, Giulio Sandini, David Vernon, Lorenzo Natale, and Francesco Nori. The icub humanoid robot: an open platform for research in embodied cognition. In *Proceedings of the 8th workshop on performance metrics for intelligent systems*, pages 50–56. ACM, 2008.
- Marc D Mikesell and Raymond J Cipra. Development of a real time intelligent robotic tracking system. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 12860, pages 213–222. American Society of Mechanical Engineers, 1994.
- S. S. Mirrazavi Salehian and A. Billard. Stable transitions from free-space to contact: A dynamical system based approach. In *Proceedings of the Workshop on Towards Robots that Exhibit Manipulation Intelligence*, 2018.
- S. S. Mirrazavi Salehian, M. Khoramshahi, and A. Billard. A dynamical system approach for catching softly a flying object: Theory and experiment. in *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 462-471, April 2016., 2016.
- S. S. Mirrazavi Salehian, N. Figueroa, and A. Billard. Dynamical system-based motion planning for multi-arm systems: Reaching for moving objects. In *Proceedings of International Joint Conference on Artificial Intelligence 2017*, Melbourne, Australia, 2017a.

- S. S. Mirrazavi Salehian, Hsiu-Chin Lin, N. Figueroa, Joshua Smith, Michael Mistry, and A. Billard. Transitioning with confidence during contact/non-contact scenarios. In *Proceedings of the Human-Robot Cooperation and Collaboration in Manipulation: Advancements and Challenges workshop*, 2018a.
- Seyed Sina Mirrazavi Salehian, Nadia Barbara Figueroa Fernandez, and Aude Billard. Dynamical system-based motion planning for multi-arm systems: reaching for moving objects. Technical report, 2017b.
- Seyed Sina Mirrazavi Salehian, Nadia Figueroa, and Aude Billard. A unified framework for coordinated multi-arm motion planning. *The International Journal of Robotics Research*, 37(10):1205–1232, 2018b.
- XJ Mo and LZ Liu. A robot system with vision touch and slide sensors for the grip onto a moving conveyor belt. In *Proc. 15th Int. Symp. Industrial Robots*, pages 129–136, 1985.
- Wataru Mori, Jun Ueda, and Tsukasa Ogasawara. A 1-dof dynamic pitching robot that independently controls velocity, angular velocity and direction of a ball. *Advanced robotics*, 24(5-6):921–942, 2010.
- Mitsuharu Morisawa, Kensuke Harada, Shuuji Kajita, Shinichiro Nakaoka, Kiyoshi Fujiwara, Fumio Kanehiro, Kenji Kaneko, and Hirohisa Hirukawa. Experimentation of Humanoid Walking Allowing Immediate Modification of Foot Place Based on Analytical Solution. (April):10–14, 2007.
- Katharina Mülling, Jens Kober, Oliver Kroemer, and Jan Peters. Learning to select and generalize striking movements in robot table tennis. *The International Journal of Robotics Research*, 32(3):263–279, 2013.
- Richard M Murray, S Shankar Sastry, and Li Zexiang. A mathematical introduction to robotic manipulation, 1994.
- Eiji Nakano, Shotaro Ozaki, Tatsuzo Ishida, and Ichiro Kato. Cooperational control of the anthropomorphous manipulator’melarm’. In *Proc. of 4th International Symposium on Industrial Robots*, volume 251, page 260, 1974.
- U Nations et al. World population ageing 2019: highlights. *New York*, 2019.
- Gabriele Nava, Francesco Romano, Francesco Nori, and Daniele Pucci. Stability analysis and design of momentum-based controllers for humanoid robots. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 680–687. IEEE, 2016.
- Maximilien Naveau, Manuel Kudruss, Olivier Stasse, Christian Kirches, Katja Mombaur, and Philippe Souères. A reactive walking pattern generator based on nonlinear model predictive control. *IEEE Robotics and Automation Letters*, 2(1):10–17, 2017.
- Francesco Nori, Silvio Traversaro, Jorhabib Eljaik, Francesco Romano, Andrea

- Del Prete, and Daniele Pucci. icub whole-body control through force regulation on rigid non-coplanar contacts. *Frontiers in Robotics and AI*, 2:6, 2015.
- Masafumi Okada, Alexander Pekarovskiy, and Martin Buss. Robust trajectory design for object throwing based on sensitivity for model uncertainties. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3089–3094. IEEE, 2015.
- Allison M Okamura, Niels Smaby, and Mark R Cutkosky. An overview of dexterous manipulation. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 1, pages 255–262. IEEE, 2000.
- David E Orin and Ambarish Goswami. Centroidal momentum matrix of a humanoid robot: Structure and properties. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 653–659. IEEE, 2008.
- David E Orin, Ambarish Goswami, and Sung-Hee Lee. Centroidal dynamics of a humanoid robot. *Autonomous Robots*, 35(2-3):161–176, 2013.
- H Andy Park and CS George Lee. Extended cooperative task space for manipulation tasks of humanoid robots. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6088–6093. IEEE, 2015.
- Tae Hyoung Park and Beom Hee Lee. An approach to robot motion analysis and planning for conveyor tracking. *IEEE transactions on systems, man, and cybernetics*, 22(2):378–384, 1992.
- Katherine L Poggensee, Albert H Li, Daniel Sotsaichik, Bike Zhang, Prasanth Kotaru, Mark Mueller, and Koushil Sreenath. Ball juggling on the bipedal robot cassie. In *2020 European Control Conference (ECC)*, pages 875–880. IEEE, 2020.
- Jerry Pratt, John Carff, Sergey Drakunov, and Ambarish Goswami. Capture Point: A Step toward Humanoid Push Recovery. *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 200–207, December 2006. doi: 10.1109/ICHR.2006.321385.
- Jerry Pratt, Twan Koolen, Tomas De Boer, John Rebula, Sebastien Cotton, John Carff, Matthew Johnson, and Peter Neuhaus. Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower body humanoid. *The International Journal of Robotics Research*, page 0278364912452762, 2012.
- Marc H Raibert, John J Craig, et al. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 103(2): 126–133, 1981.
- Daniel Rakita, Bilge Mutlu, Michael Gleicher, and Laura M Hiatt. Shared control-based bimanual robot manipulation. *Science Robotics*, 4(30):

eaaw0955, 2019.

Fredy Raptopoulos, Maria Koskinopoulou, and Michail Maniadakis. Robotic pick-and-toss facilitates urban waste sorting. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 1149–1154. IEEE, 2020.

Philipp Reist and Raffaello D’Andrea. Design and analysis of a blind juggling robot. *IEEE Transactions on Robotics*, 28(6):1228–1243, 2012.

Mark Rijnen, Alessandro Saccon, and Henk Nijmeijer. Reference spreading: Tracking performance for impact trajectories of a 1dof setup. *IEEE Transactions on Control Systems Technology*, 28(3):1124–1131, 2019.

E Rimón and DE Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, 1992.

Fabio Ruggiero, Vincenzo Lippiello, and Bruno Siciliano. Nonprehensile dynamic manipulation: A survey. *IEEE Robotics and Automation Letters*, 3(3):1711–1718, 2018.

Layale Saab, Oscar E Ramos, François Keith, Nicolas Mansard, Philippe Soueres, and Jean-Yves Fourquet. Dynamic whole-body motion generation under rigid contacts and other unilateral constraints. *IEEE Transactions on Robotics*, 29(2):346–362, 2013.

Seyed Sina Mirrazavi Salehian and Aude Billard. A dynamical system based approach for controlling robotic manipulators during non-contact/contact transitions. *IEEE Robotics and Automation Letters*, 2018.

Seyed Sina Mirrazavi Salehian, Nadia Figueroa, and Aude Billard. Coordinated multi-arm motion planning: Reaching for moving objects in the face of uncertainty. In *Proceedings of Robotics: Science and Systems*, Ann Arbor, Michigan, June 2016a. doi: 10.15607/RSS.2016.XII.019.

Seyed Sina Mirrazavi Salehian, Mahdi Khoramshahi, and Aude Billard. A dynamical system approach for softly catching a flying object: Theory and experiment. *IEEE Transactions on Robotics*, 32(2):462–471, 2016b.

Joseph Salini, Sébastien Barthélemy, and Philippe Bidaud. Lqp controller design for generic whole body motion. In *Mobile Robotics: Solutions and Challenges*, pages 1081–1090. World Scientific, 2010.

Aykut C. Satıcı, Fabio Ruggiero, Vincenzo Lippiello, and Bruno Siciliano. A coordinate-free framework for robotic pizza tossing and catching. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3932–3939, 2016. doi: 10.1109/ICRA.2016.7487582.

Aykut C Satıcı, Fabio Ruggiero, Vincenzo Lippiello, and Bruno Siciliano. A

- coordinate-free framework for robotic pizza tossing and catching. In *Robot Dynamic Manipulation*, pages 207–227. Springer, 2022.
- Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.
- Stefan Schaal and Christopher G Atkeson. Open loop stable control strategies for robot juggling. In *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pages 913–918. IEEE, 1993.
- Stefan Schaal, Shinya Kotosaka, and Dagmar Sternad. Nonlinear dynamical systems as movement primitives. In *IEEE international conference on humanoid robotics*, pages 1–11, 2000.
- Markus M Schill and Martin Buss. Robust ballistic catching: A hybrid system stabilization problem. *IEEE Transactions on Robotics*, 34(6):1502–1517, 2018.
- Stanley A Schneider and Robert H Cannon. Object impedance control for cooperative manipulation: Theory and experimental results. *IEEE Transactions on Robotics and Automation*, 8(3):383–394, 1992.
- Allen I Selverston. Are central pattern generators understandable? *Behavioral and Brain Sciences*, 3(4):535–540, 1980.
- Taku Senoo, Akio Namiki, and Masatoshi Ishikawa. Ball control in high-speed batting motion using hybrid trajectory generator. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 1762–1767. IEEE, 2006.
- Taku Senoo, Akio Namiki, and Masatoshi Ishikawa. High-speed throwing motion based on kinetic chain approach. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3206–3211. IEEE, 2008.
- Luis Sentis and Oussama Khatib. A whole-body control framework for humanoids operating in human environments. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2641–2648. IEEE, 2006.
- Diana Serra, Aykut C Satıcı, Fabio Ruggiero, Vincenzo Lippiello, and Bruno Siciliano. An optimal trajectory planner for a robotic batting task: the table tennis example. In *International Conference on Informatics in Control, Automation and Robotics*, volume 3, pages 90–101. SCITEPRESS, 2016.
- Diana Serra, Fabio Ruggiero, Vincenzo Lippiello, and Bruno Siciliano. A nonlinear least squares approach for nonprehensile dual-hand robotic ball juggling. *IFAC-PapersOnLine*, 50(1):11485–11490, 2017.
- Milad Shafiee-Ashtiani, Aghil Yousefi-Koma, Masoud Shariat-Panahi, and Majid Khadiv. Push recovery of a humanoid robot based on model predictive

- control and capture point. In *Robotics and Mechatronics (ICROM), 2016 4th International Conference on*, pages 433–438. IEEE, 2016.
- Milad Shafiee-Ashtiani, Aghil Yousefi-Koma, and Masoud Shariat-Panahi. Robust bipedal locomotion control based on model predictive control and divergent component of motion. In *Robotics and Automation (ICRA), 2017 International Conference on*, pages 3505–3510. IEEE, 2017.
- Rajeev Sharma, J-Y Herve, and Peter Cucka. Dynamic robot manipulation using visual tracking. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pages 1844–1845. IEEE Computer Society, 1992.
- Avishai Sintov and Amir Shapiro. A stochastic dynamic motion planning algorithm for object-throwing. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2475–2480. IEEE, 2015.
- Christian Smith, Yiannis Karayiannidis, Lazaros Nalpantidis, Xavi Gratal, Peng Qi, Dimos V Dimarogonas, and Danica Kragic. Dual arm manipulation a survey. *Robotics and Autonomous systems*, 60(10):1340–1353, 2012.
- Nicolas Sommer, Klas Kronander, and Aude Billard. Learning externally modulated dynamical systems. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3956–3963. IEEE, 2017.
- Benjamin J Stephens and Christopher G Atkeson. Push recovery by stepping for humanoid robots with force controlled joints. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 52–59. IEEE, 2010.
- Walter B Sturek, Harry A Dwyer, Lyle D Kayser, Charles J Nietubicz, Robert P Reklis, and Klaus O Opalka. Computations of magnus effects for a yawed, spinning body of revolution. *AIAA Journal*, 16(7):687–692, 1978.
- Hsi Guang Sung. *Gaussian mixture regression and classification*. PhD thesis, Rice University, 2004.
- Masaru Uchiyama and Pierre Dauchez. Symmetric kinematic formulation and non-master/slave coordinated control of two-arm robots. *Advanced Robotics*, 7(4):361–383, 1992.
- Junichi Urata, Koichi Nshiwaki, Yuto Nakanishi, Kei Okada, Satoshi Kagami, and Masayuki Inaba. Online decision of foot placement using singular lq preview regulation. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 13–18. IEEE, 2011.
- Nikolaus Vahrenkamp, Dmitry Berenson, Tamim Asfour, James Kuffner, and Rüdiger Dillmann. Humanoid motion planning for dual-arm manipulation and re-grasping tasks. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2464–2470. IEEE, 2009.

- Nikolaus Vahrenkamp, Martin Do, Tamim Asfour, and Rüdiger Dillmann. Integrated grasp and motion planning. In *2010 IEEE International Conference on Robotics and Automation*, pages 2883–2888. IEEE, 2010.
- Miomir Vukobratović and Juri Stepanenko. On the stability of anthropomorphic systems. *Mathematical biosciences*, 15(1-2):1–37, 1972.
- Yuquan Wang and Abderrahmane Kheddar. Impact-friendly robust control design with task-space quadratic optimization. In *Robotics: Science and Systems (RSS)*, 2019.
- Yuquan Wang, Niels Dehio, Arnaud Tanguy, and Abderrahmane Kheddar. Impact-Aware Task-Space Quadratic-Programming Control. working paper or preprint, November 2020. URL <https://hal.archives-ouvertes.fr/hal-02741682>.
- Pierre-Brice Wieber. On the stability of walking systems. In *Proceedings of the international workshop on humanoid and human friendly robotics*, 2002.
- Pierre-brice Wieber. Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations. *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 137–142, December 2006. doi: 10.1109/ICHR.2006.321375.
- David Williams and Oussama Khatib. The virtual linkage: A model for internal forces in multi-grasp manipulation. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 1025–1030. IEEE, 1993.
- Thomas Wimböck, Christian Ott, Alin Albu-Schäffer, and Gerd Hirzinger. Comparison of object-level grasp controllers for dynamic dexterous manipulation. *The International Journal of Robotics Research*, 31(1):3–23, 2012.
- Ning Xi, Tzyh-Jong Tarn, and Antal K Bejczy. Intelligent planning and control for multirobot coordination: An event-based approach. *IEEE transactions on robotics and automation*, 12(3):439–452, 1996.
- William Yang and Michael Posa. Impact invariant control with applications to bipedal locomotion. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5151–5158. IEEE, 2021.
- Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Tossingbot: Learning to throw arbitrary objects with residual physics. *IEEE Transactions on Robotics*, 36(4):1307–1319, 2020.
- Yajia Zhang, Jingru Luo, and Kris Hauser. Sampling-based motion planning with dynamic intermediate state objectives: Application to throwing. In *2012 IEEE International Conference on Robotics and Automation*, pages 2551–2556. IEEE, 2012.

CURRICULUM VITAE

Michael Bosongo Bombile

No. 9, Chemin de Rionza, 1020 Renens, Switzerland
Email: michael.bombile@epfl.ch; micbombile@gmail.com;
Tel : +41 (0)21 693 56 53, Cell: +41(0)78 649 12 32
Web: michaelbombile.github.io

RESEARCH INTERESTS

Dynamical Systems
Dynamic and Dexterous Multi-robot Manipulation
Cooperative control in Human-Robot and Robot-Robot interactions
Whole-body Control of Humanoid robot in multi-contact scenario
Control of Hybrid Dynamical Systems
Dynamic Visual Servoing
Learning from demonstration

EDUCATION

- 2018-2022** **PhD Student in Robotics**, Control, and Intelligent Systems, at the Swiss Federal Institute of Technology (EPFL), Switzerland
Thesis title: Towards Dynamic Cooperative Manipulation: A Bi-Manual Coordinated Motion and Force Perspective
Thesis advisor: Prof. Aude Billard
- 2012-2015** **MSc by Research in Electrical Engineering** (obtained with **Distinction**), at University of Cape Town (UCT), South Africa
Thesis Title: Visual Servo Control of a Humanoid Robot
- 2005-2010** **Diplôme d'Ingénieur Civil Electromecanicien** (equivalent to **BSc + MSc** in Electromechanical Engineering), at University of Lubumbashi (UNILU) (obtained with **Distinction**)
Thesis Title: Modelling and Control of a three-phase Electrical Arc Furnace by means of Artificial Neural Networks

RESEARCH EXPERIENCE

- 2019 - Present** **Research Assistant**, LASA, EPFL, Lausanne, Switzerland
- Supervision of master projects
 - Teaching Assistant in Robotics Practicals
 - Teaching Assistant in Machine Learning Programming
 -
- 2017 - 2019** **Scientific Assistant**, LASA, EPFL
Developed whole-Body control algorithms for a humanoid robot to perform a bimanual grasping of fixed and moving objects while keeping the robot balanced
- Feb 2017-Jul 2017** **Research Intern**, LASA, EPFL

Developed a reactive balance and walking for humanoid robots

Apr 2016 -Jul 2016 **Candidate to PhD Studies** in Electrical Engineering (Space studies), UCT
worked on a proposal to tackle the capturing and stabilisation of a non-cooperative and tumbling space target using a space robot for on-orbit servicing or removal purposes.

2012 – 2015 **As Master's Student** in Electrical Engineering, UCT
Developed a general Visual Servoing Scheme for Humanoid Robots, which is applicable to tasks such as Positioning, object Tracking, and Grasping while Walking.
Successfully implemented the developed scheme on the Humanoid robot NAO.

2012-2013 **Tutor**, in Electrical Engineering, UCT, Cape Town, South Africa

- Control Engineering I & II (Continuous and Digital Control)
- Model Predictive Control (MPC) / Process Control and Instrumentation

INDUSTRIAL EXPERIENCE

Oct-Dec 2010 Intern Maintenance Engineer, Usines de Lubumbashi, Gecamines (Ltd.)

AFFILIATION

2017-present Member of the Engineering Council of South Africa (ECSA)

PUBLICATIONS

Journal **M. Bombile** and A. Billard, "Dual-Arm Control for Coordinated Fast Grabbing and Tossing of an Object: Proposing a New Approach," in *IEEE Robotics & Automation Magazine*, vol. 29, no. 3, pp. 127-138, Sept. 2022, doi: 10.1109/MRA.2022.3177355.

M. Bombile and A. Billard. "Bi-manual dynamic grabbing and tossing of objects on a moving target." *Robotics and Autonomous Systems* (under review) (2022).

Conference H. Khurana, **M. Bombile** and A. Billard. " Learning to Hit: A statistical Dynamical System based approach." *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp, 9415-942, doi: 10.1109/IROS51168.2021.9635976

Bombile, M and Billard A. "Capture-Point based Reactive Balance and Omnidirectional Walking Controller." *In Proceedings of the 17th IEEE-RAS International Conference on Humanoid Robots, Birmingham, UK, November 15-17, 2017*, pages 17-24 .

Book Chapters **Bombile, Michael**. "Visual Servoing Based Positioning and Object Tracking on Humanoid Robot." *New Trends in Networking, Computing, E-learning, Systems Sciences, and Engineering*. Springer International Publishing, 2015. 19-27.

Bohra, P., and **M. Bombile**. "Towards a Passive Gait: Modeling the Fully Actuated Humanoid NAO." *New Trends in Networking, Computing, E-learning, Systems Sciences, and Engineering*. Springer International Publishing, 2015. 505-512.

Thesis

Bombile, Michael. "Visual Servo Control on a Humanoid Robot". MSc Thesis, University of Cape Town, 2015. Available here: <https://open.uct.ac.za/handle/11427/18197>

LANGUAGES

French – Mother tongue
English – Fluent
Swahili – Spoken

TECHNICAL SKILLS

Robots:	KUKA LBR IIWA, KUKA LWR, UR5, Katana, Humanoid iCub, Humanoid Coman, Humanoid NAO, Ridgeback mobile robot
Programming:	C/C++, Matlab, Python, ROS
Computing and Simulation:	MATLAB/Simulink, Maple, Mathematica, Gazebo
CAD Softwares:	Solidworks, CATIA, AutoCAD, Eagle CAD
PLC Programming:	Relay Ladder Logic, Instruction List, Structured text
Office Softwares	Latex, Microsoft Office

HOBBIES AND INTERESTS

Soccer, hiking, Meditation,
Designing, building electronic circuits and devices