

# Control of humanoid robots to realize haptic tasks in collaboration with a human operator

Paul Evrard

## ► To cite this version:

Paul Evrard. Control of humanoid robots to realize haptic tasks in collaboration with a human operator. Robotics [cs.RO]. Université Montpellier II - Sciences et Techniques du Languedoc, 2009. English. NNT: . tel-00807094

## HAL Id: tel-00807094 https://theses.hal.science/tel-00807094

Submitted on 3 Apr 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés. Nº d'ordre: xxxx

## THÈSE

présentée

## devant l'Université de Montpellier 2

pour obtenir

le grade de : DOCTEUR DE L'UNIVERSITÉ DE MONTPELLIER 2 Mention Systèmes Automatiques et Microélectroniques

par

Paul EVRARD

Laboratoire d'accueil : CNRS-AIST JRL, UMI3218/CRT École Doctorale : Information, Structures, Systèmes

Titre de la thèse :

## Contrôle d'humanoïdes pour réaliser des tâches haptiques en coopération avec un opérateur humain

Soutenance prévue le 07/12/2009 devant la commission d'examen

Étienne	BURDET	Rapporteurs
Alain	MICAELLI	
Aude	BILLARD	Examinateurs
Martin	BUSS	
André	CROSNIER	
Konstantinos	TZAFESTAS	
Kazuhito	YOKOI	
Abderrahmane	KHEDDAR	Directeur de Thèse

# Abstract

HAPTIC collaborative tasks are actions performed jointly by several partners, involving direct or indirect physical contact among them. A typical example of such tasks are collaborative manipulation tasks, where the partners apply forces on a same object to impose it a desired motion or bring it to a target location. Human beings learn naturally how to perform such tasks with other human partners, but implementing such behaviors on a robotic platform is challenging. When jointly manipulating an object, the partners no longer act independently, and must *negotiate a common plan* to perform the task.

To avoid conflicts among the partners' intentions, the *leader-follower* model defines a task leader, who imposes a task plan to the other partners, while the latter act as follower and follow at best the intentions of the leader. This model has often been used in physical Human-Robot Interaction (pHRI). Because robotic systems have limited cognitive capabilities in comparison to human beings, a follower role has generally been assigned to robotic systems to cooperate with human operators. Recently, thanks to the increasing computational power embedded into the robots, more and more initiative has been given to robotic assistants. In some recent works, robots were sometimes even given the possibility to lead human operators.

In the context of physical tasks, where the partners are in direct or indirect contact through an object and exchange mechanical energy, we believe that the haptic channel is a favored and fast way for the partners to exchange information about their intentions. Therefore, this thesis will focus on the kinesthetic aspects of collaborative tasks. The long-term aim of the project is to endow humanoid robots with the necessary haptic skills to perform collaborative tasks with a human operator *as a partner rather than as a helper*. The work presented here proposes solutions towards this direction.

In its first part, our contribution is to extend the leader-follower model to continuous, time-varying role distributions among the partners in the context of haptic dyadic collaborative tasks. This model describes the behavior of each partner of the dyad as a variable weighting between the two extreme leader and follower behaviors. Our goal is to abstract the concept of role distribution from the implementation of the underlying controllers, and to describe the behavior of dyads using two independent functions that will shape the behavior of each partner in term of leadership. We exemplify the use of our model in a virtual reality scenario where a human operator manipulates an object in cooperation with a virtual robotic system. We also explore possible strategies to exploit it. The problem we adress is to define how the weighting between both behaviors can be adjusted automatically on a robotic system, depending on various criteria such as constraints of the robot or knowledge from human-human haptic interaction. Simulations and experiments conducted on a humanoid robot are presented to illustrate the proposed solutions. The results show that the extended leader-follower model can be applied to realize collaborative tasks with a human operator while avoiding self-collision. The model also encompasses the *specialization* phenomenon recently highlighted in human-human collaborative haptic tasks.

We then propose to use a programming by demonstration method to teach col-

laborative skills to a robotic system. This method uses a probabilistic framework to encode the characteristics of the task and reproduce it autonomously. This framework is based on Gaussian Mixture Models and Gaussian Mixture Regression and has been successfully applied to various stand-alone tasks. We remind the main components of this framework and present its application to collaborative lifting tasks between a humanoid robot and a human operator. Our first contribution is the design of the experimental setup, based on a teleoperation system whith kinesthetic feedback which allows the human teacher to demonstrate the task while taking into account the constraints and sensor data of the robotic system. The main contribution, however, is the use of this methodology to attempt to *assess the validity of our extended leader-follower model*, by highlighting smooth switching behaviors on human partners during collaborative lifting tasks. The experimental data aquired during reproductions of the task is analyzed within this perspective.

The second part of this thesis focuses on the control of humanoid robots in the context of pHRI. We examine several paradigms of interaction: interaction between two remote human partners through a tele-presence system, direct interaction between an autonomous humanoid robot and a human operator, and collaborative transportation tasks between a human operator and a humanoid robot. Behind these different paradigms of interaction lies one common problem: the generation of whole-body motion and gait in response to external forces that arise from the haptic interaction with a human operator. This thesis does not aim at tackling the problem of gait generation at the mechanical and control level. We will rather use state-of-the-art algorithms which do not consider external disturbances, and show to what extent they can be used to generate complex and intuitive collaborative behaviors. Our contributions in this part are thus to integrate impedance control and gait generation within an existing control architecture in a generic and flexible way in order to (i) use the resulting controller in various contexts, (ii) demonstrate how the basic principles of impedance control can be implemented on a *complex* platform biped humanoid robot while exploiting all the capabilities of such platforms and (iii), highlight the limitations of the passivity-based approaches often used in pHRI, and thereby justify further research in the field of pHRI. The work presented in this part has been integrated within a complex demonstrator where the robot walks in a teleoperated manner and performs autonomously a collaborative transportation task with a human operator.

# Acknowledgements

HE works presented here have been conducted in the CNRS-AIST Joint Robotics Laboratory, in Tsukuba, Japan. I would like to thank my supervisor Abderrahmane Kheddar, for having offered me such a fantastic opportunity. I am grateful for all the help and support he provided me, and for the knowledge and know-how I gained through these three years.

I thank Etienne Burdet and Alain Micaelli as well as all members of the committee for accepting to review this thesis, for attending my defence, and for the helpful comments.

I would like to thank Kazuhito Yokoi and Eiichi Yoshida for welcoming me in their laboratory, and for their precious help with Japanese administration. For all the help in Japanese I thank Tokiko Uga, Jean-Rémy Chardonnet, Hitoshi Arisumi and Makoto Yamaguchi. My Ph.D. student life in Japan would not have been as confortable without their priceless assistance.

Special thanks to Aude Billard, Martin Buss, Sylvain Calinon, Satoshi Endo, Elena Gribovskaya, Sandra Hirche, Inga Krause, Angelika Peer, Thomas Schauss and Carolina Weber for their collaboration within the European projects Robot@CWE and ImmerSence.

For the time spent with our favorite robot, all the discussions, his friendship and his role in one of the most important events of my life, I deeply thank Olivier Stasse. I would also like to express my special thoughts towards François Keith, for being such a trustworthy friend, collaborator, and coding buddy.

Special thoughts also to Diane Larlus for helping me improve my typing ability and for dealing with a couple of LaTeX issues; special thoughts, and congratulations to Nicolas Mansard, who managed to convince me to practice some sports despite my busy schedule.

Many thanks to Audrey Imbert, Sebastien Lengagne, Claire Dune, François Maillard and Anthony David for their assistance when moving back to France. Many thanks also to all the people who made this stay in Japan so enriching.

Last, but not least, I would like to express my deepest love and gratitude to my wonderful wife, who supported me through these years. There can be no bad day as long as I know I am seeing her when going back home.

# Contents

At	ostrac	t		i
Co	onten	ts		v
Li	st of I	Figures		ix
Li	st of '	Tables		xvi
1	Нар 1.1	tic inter Introdu	raction: a survey	1 3
		1.1.1	Humanoid robots	3
		1.1.2	Haptic collaborative tasks	3
		1.1.3	Interaction control and the leader-follower model	5
		1.1.4	Human-human collaboration	5
	1.2	Impeda	ance control	7
		1.2.1	Force control and compliance control	7
		1.2.2	Impedance control	8
		1.2.3	Contact instability	14
		1.2.4	Conclusion	15
	1.3	Impeda	ance control and collaborative tasks	15
		1.3.1	Why impedance control?	16
		1.3.2	Applications in the previous works	17
		1.3.3	Limitations	19
		1.3.4	Conclusion	21
	1.4	Recent	advances: towards perfect followers?	22
		1.4.1	Human impedance models	22
		1.4.2	Active following	23
		1.4.3	Proactive behaviors	25
		1.4.4	Leader robots?	25
		1.4.5	Conclusion	26
	1.5	Conclu	ision	26
2	Phy	sical co	llaborative tasks: a homotopy-based interaction model	29
	2.1	A hom	otopy switching model for physical Human-Robot Interaction	31
		2.1.1	The leader-follower model	31
		2.1.2	Intuition	31
		2.1.3	Mathematical formulation	32
		2.1.4	Example: linear homotopy	33
		2.1.5	Genericity of the proposed model	34
		2.1.6	Challenges for the application to pHRI: reasons for a haptic lan-	
			guage	35

		2.1.7 Relation to previous works	36
		2.1.8 Behavioral architecture	37
	2.2	A concrete example	38
		2.2.1 Setup	39
		2.2.2 Results	40
		2.2.3 Extension to a Person-Object-Avatar scenario	41
	2.3	Defining the homotopy	44
		2.3.1 Enforcing constraints: self-collision avoidance	45
		2.3.2 parameterizing phenomena observed in human-human collabo-	
		ration	47
		2.3.3 Conclusion	52
	2.4	Conclusion	52
3	Lear	rning haptic patterns	55
	3.1	Programming by demonstration of collaborative tasks	57
		3.1.1 Introduction	57
		3.1.2 Programming by Demonstration	58
		3.1.3 Application to Human-Robot collaborative lifting tasks	59
	3.2	Demonstration setup and experiments	59
		3.2.1 Hardware setup and controller	60
		3.2.2 Experiments	62
		3.2.3 Conslusions	62
	3.3	Learning collaborative tasks	64
		3.3.1 Probabilistic encoding of the task	65
		3.3.2 Probabilistic reconstruction of the task	66
		3.3.3 Control scheme	68
	3.4	Task reproduction	70
		3.4.1 Reproduction setup	70
		3.4.2 Preliminary analysis	70
		3.4.3 Quantitative analysis	73
	3.5	Conclusion	79
4	Coll	laborative tasks via telepresence	81
	4.1	Teleoperation of humanoid robots: a brief review	83
	4.2	The telepresence setup	84
	4.3	Taxonomies of coupling	85
	4.4	Whole-body control and extension to wide-area motions	87
		4.4.1 Whole-body motion control	88
		4.4.2 Force-based control	89
		4.4.3 Wide-area motions	90
	4.5	Experiments	90
		4.5.1 Setup	90
		4.5.2 Control	91
		4.5.3 Results and Discussion	93
	4.6	Conclusion	96
5	Exp	eriments on virtual avatars and humanoid robots	99
	5.1	Introduction	101
	5.2	Collaborative transportation task	102
		5.2.1 Problem statement	102
		5.2.2 Stack of Tasks	103

		5.2.3 Force-based control	103
		5.2.4 Real-time Walking Pattern Generator	105
		5.2.5 Footprints planning: the Mobile Robot Helper approach 1	106
		5.2.6 Decoupling the gait and manipulation tasks	108
	5.3	Integration and implementation	111
		5.3.1 Software framework	11
		5.3.2 Integration	14
		5.3.3 Implementation issues	14
		5.3.4 Experimental results	116
		5.3.5 Integration to the final demonstrator of the Robot@CWE project 1	18
		5.3.6 Discussion and future works	21
	5.4	Conclusion	22
	0		
А	Bila	teral coupling of the end-effectors for a tele-cooperation system 1	129
	A.1	Network model	29
		A.1.1 Modelling the operator and environment	129
		A.1.2 Adding a second human in the loop	129
		A.1.3 Teleoperation interface and communication setup 1	130
	A.2	Stability analysis	131
		A.2.1 Model	131
		A.2.2 Stability criterion	131
		A.2.3 Numerical analysis	133
В	A st	udy of inter-trial adaptation in dyadic collaborative lifting tasks 1	135
	<b>B</b> .1	Problem statement	135
	B.2	Experimental setup	135
	B.3	Method	136
	<b>B.</b> 4	Adaptation law	136
	B.5	Optimal adaptive behavior	137
	B.6	Conclusion	138
~	-		
С	Deve	elopment of an interactive simulation framework with haptic	
	rend	lering 1	139
	C.1	Introduction	39
	C.2	Software architecture	139
		C.2.1 Presentation of Amelif	139
		C.2.2 Basic components	140
	C.3	Demonstrator	142
		C.3.1 Scenario	142
		C.3.2 The virtual HRP-2 avatar	142
		C.3.3 Dynamic simulation	142
		C.3.4 Focus on the haptic probe	143
		C.3.5 Task	145
		C.3.6 Results	145
P	A 171		
D	A Pl	HAN IOM <sup>®</sup> Device with 6DOF Force Feedback and Sensing	
	Capa		L47
	D.I		147
	D.2		148
	D.3	Conclusion	149

Bibliography

# List of Figures

1.1	Collaborative transportation task between a human and a humanoid robot.	4
1.2	Structure of a closed-loop force servo controller. C is a compen- sator, C is a feedforward gain. D is a function applied to force	
	measurements (typically low-pass filtering) and P is the plant	8
1.3	Robot end-effector (represented by a torus) moving along a bar.	0
	The torus is coupled to a mass which moves along the bar and is	
	subject to viscous friction.	9
1.4	Electrical equivalent of a manipulation task at steady state per-	
	formed by a motion controlled manipulator.	10
1.5	Electrical equivalent of a manipulation task at steady state per-	
	formed by an impedance controlled manipulator	10
1.6	Target mass-spring-damper system. $M$ is the target mass, $B$ the	
	target damping, <i>K</i> the target stiffness, $x_0$ is the virtual compliance	
	center and x is the position of the mass, which corresponds to the	
	position of the end-effector.	13
1.7	Position-based impedance controller. A is the target impedance, C	1.4
1.0	is a position controller, and P is the plant.	14
1.8	One doi point-to-point task.	10
1.9	trias to impose a desired position V, to the object. The partner	
	applying the force $E_i$ appears as a disturbance to the other partner	17
1 10	Target equivalent networks for different control schemes: on the	17
1.10	left both partners are position controlled and try to act as a position	
	source. The target network is not feasible. Middle: one of the	
	partners only is position controlled, and the other acts as a force	
	source. The current that flows through the inductance corresponds	
	to the motion of the object and is imposed by only one partner.	
	On the right: target network when both partners are impedance	
	controlled. The network is feasible, and symmetric: both partners	
	can act on the motion of the object	18
1.11	Passive impedance control	20
2.1	Illustration of one degree of freedom homotopy for each individ-	
	ual of the physical interaction task dyad (holding a table). Each	
	$\alpha_{i \in \{1,2\}}$ may evolve independently from the other and their time	
	function results on a dynamic sliding between 0 and 1 during task	
	execution.	34

2.2	Person-Object-Avatar physical collaborative task architecture and main modules. The solid and dashed lines represent the <i>a priori</i> plans of the partners. The red point represents the starting position of the object and the yellow one represents the target destination. Two data flows can be seen on the figure. The flow that goes through the light gray boxes represents data that is closely re-	
	lated to the communication with the human partner. The dark gray blocks are more related to the internal state and constraints of the	20
2.2	robol.	38 20
2.3 2.4	Result of the simulation of a collaborative task between two virtual humanoids when both act as leader. Left: desired trajectory for each avatar, and actual trajectory of the object. Right: vertical	39
2.5	Trajectory of the object when one of the partners is a leader and the other is a follower in the Avatar-Object-Avatar experiment, where	40
2.6	two virtual avatars lift an object and pass it over an obstacle Experimental setup: a human operator uses a PHANToM Desktop device to lift a virtual object in collaboration with a virtual robotic	41
	partner	41
2.7	Plan of the virtual avatar: desired motion of the object.	42
2.8	Typical joint torques at the joints of the avatar (chest joint and right arm joints). The torque references sent to the actuators of the robot	12
2.9	Trajectories when the human leads all the task (blue) and when he leads part of the task (green). The human trajectory goes further	45
	from the obstacle.	43
2.10	A human operator and HRP-2 jointly moving an object towards the left-hand side of the robot. During the motion, the right arm of the robot moves towards the short increasing the rick of salf callicion	16
2.11	Evolution of the homotopy parameter and collision distance along the task performance (bottom graph, solid line: homotopy param- eter, dashed line: collision distance). On the top graph, the trajec- tory of the robot is shown. The blue line corresponds to the plan of the robot, the yellow line corresponds to the actual trajectory	40
	followed by the robot.	47
2.12	1 degree-of-freedom collaborative positioning task	48
2.13	Profile of the homotopy variable for different values of $\beta$ . High values of $\beta$ give highly specialized profiles. The specialization is less pronounced for small values of $\beta$ . An unspecialized profile is	
	obtained for $\beta = 0$	49
2.14	Simulation model. $PD_i$ denotes the Proportional-Derivative con- troller used to implement the leader behavior. $Z_i^f$ denotes the fol-	
	lower impedance controller.	50
2.15	Force applied by each partner on the object to reach a common target position, for different specialization profiles (from non-	
2.16	specialized to fully specialized)	50
	target position: $\beta = 0$ , unspecialized case (up); $\beta = 1$ , fully specialized case (down).	51
		51

2.17	Motion of the object (up), forces applied by the partners (middle) and homotopy parameters of each partner (down).	51
3.1	The three variables considered in our study. The vertical force $f$ is the force exerted on the robot's wrist.	60
3.2	A human (teacher) teleoperates the humanoid robot HRP-2 through a force feedback haptic display to demonstrate how to	
	perform a collaborative task with a human partner (operator)	61
3.3	First row: force against time and position against forces. The	
	case (in green) are clearly separated. Second row: velocity profiles	
	plotted against the position and force variables. The trajectories	
	in red and the trajectories in green correspond respectively to the	
	demonstrations of the robot acting as a leader and as a follower.	
	The circles represent the beginning of the motions	63
3.4	The two data-sets recorded during the demonstration. The data-	
	set $D_{\mathscr{L}}$ corresponding to the demonstrations with a leader robot is depicted in red, and the data set $D_{\mathcal{L}}$ corresponding to the com-	
	plementary role distribution (follower robot and leader human) is	
	shown in green.	65
3.5	Gaussian Mixture Models encoding the task dynamics. The red	
	ellipsoids represent the components of the GMM $\mathscr{G}_{\mathscr{L}}$ encoding the	
	task dynamics with the "leader robot-follower human" role distri-	
	button. The green ellipsoids represent the components of the GMM $(I_{a})$ associated with the "follower robot loader human" demonstra	
	signassociated with the Tonower Tobot-Teader human demonstra-	67
3.6	Illustration of the GMR process. The red plane represents the im-	
	age of the input space through the GMR process. The orientation	
	of this plane gives information on how the reconstructed output $\nabla (\langle S O   S \rangle)$ is a second secon	60
27	$E(p(\xi^{\circ} \xi^{1}))$ will vary with the values of the inputs $\xi_{1}^{1}$ and $I\xi_{2}^{2}$	69
5.7	The surface represents the image of the input space through the	
	GMR process. The blue (respectively red) part shows the area	
	where the blue (respectively red) component has more effect on	
	the value of $E(p(\xi^{O} \xi^{I}))$ . In the blue (red) area, the surface is	
	almost the image plane obtained with only the blue (red) compo-	
	nent such as in figure 3.6. Between these areas, the colors are	
	The figure shows how the blue and red components are used to	
	compute $E_{\text{hlue}}(p(\xi^{O} \xi^{I}))$ and $E_{\text{red}}(p(\xi^{O} \xi^{I}))$ , which are located	
	on the cross-ruled planes. These values are weighted to obtain	
	$E(p(\xi^O \xi^I))$ , located on the surface S	69
3.8	Autonomous replication of a lifting task by HRP-2 jointly with a	
2.0	human partner.	71
3.9	Failed reproduction attempts. The object was successfully lifted, but the dynamics of the motion is not the demonstrated one: the	
	human partner applied very large forces on the object which we	
	interpret as an unsatisfied will to increase the pace.	71
3.10	Reproduction trials where the dynamics of the motion is close to	
	the demonstrated "human leader-robot follower" task dynamics.	72

3.11	Reproduction trial where the dynamics of the motion is close to the demonstrated "human follower-robot leader" task dynamics.	72
3.12	Reproduction trials where the dynamics of the motion switched from the "robot leader" to the "robot follower" dynamics.	73
3.13	Reproduction trials where the dynamics of the motion switched twice between both demonstrated dynamics.	74
3.14	Image of the force-position plane through the GMR process. The ellipsoids represent the components of the GMM, and the surface shows the velocity output $E(p(\dot{x} x, f))$ for each value of the position $x$ and force $f$ . Orange areas correspond to the higher velocities, and blue ones to the lower velocities.	75
3.15	Image of the force-position plane through the GMR process. The area in light gray is the area where the success criterion is satisfied, <i>i.e.</i> points $\xi$ such that $\log(L(\xi, \mathcal{G})) > l_{lim}$ .	76
3.16	Left: successful trials. The data-points have been projected on the position-velocity plane. Right: failed trials. By comparing these pictures, we notice that successful trials generally followed the shape suggested by the ellipsoids, and reached a final position that was closer to the demonstrated one, above the final position of	
3.17	the failed trials	76
3.18	two areas is smooth	77
3.19	the data-points are plotted over the leader and follower areas Velocity of the gripper of the robot and value of the homotopy parameter during one trial. The smooth transition between both task dynamics can be seen from the continuous transition of the homotopy parameter from 1 to 0	77
		70
4.1 4.2	Telepresence setup for distant collaborative tasks.         End-effector coupling.         Discussion in the setup of	85 85
4.5	sult in a wrong intention estimation by the distant human partner.	86
4.4	and attitude of both the distant human and the object are mapped onto the master device. Bottom: the same object is manipulated on both sides, and only the posture of the distant human is mapped	
15	onto the mobile haptic interface.	86
4.3 4.6	<i>E</i> - <i>P</i> bilateral coupling: local control on the slave site	87 88
4.0 1 7	Photos of the experimental setup. On the left the operator site with	00
4./	the human operator and the mobile haptic interface in Munich is depicted. On the right the teleoperator HRP2 and the human col-	
	laborator in Tsukuba are shown.	91
4.8	Cooperative telemanipulation task: a) approach, b) grasp, c) lift, d) put down	93

4.9	Left: teleoperated HRP-2 collaborating with a human on the	
	remote-site. Right: immersed human operator operating the HRP-	
	2 robot using the VISHARD arm	94
4.10	Force and position tracking during experiment. 1: approaching	
	phase, 2: moving phase, 3: releasing phase, shaded area: human	
	located at the remote site applies forces to the object	95
4.11	Position tracking of body and end-effectors with a dominant hu-	
	man operator at the master site. Data of the human operator at mas-	
	ter site is depicted in black, data of the teleoperator in gray. The	
	dashed lines represent the body position whereas the solid lines	
	represent the right hand/end-effector.	96
4.12	Position tracking of body and end-effectors with a dominant hu-	
	man collaborator at remote site. Data of the human operator at	
	master site is depicted in black, data of the teleoperator in gray.	
	The dashed lines represent the body position whereas the solid	
	lines represent the right hand/end-effector.	97
5.1	Four contexts of task realization in the physical common workspace 1	01
5.2	Cooperative table transportation between an HRP-2 humanoid	
	robot and a human operator	03
5.3	Virtual coupling of the end-effectors and the body of a humanoid	
	robot to synchronize the arms and the body of the robot during	
	a collaborative transportation task. $M_{base}$ , $B_{base}$ , $M_{arms}$ , $B_{arms}$ and	
	$K_{arms}$ are the virtual impedance parameters used to couple the arms	
	and the base, and to generate the motion of the base. $F_{ext}$ is the	
	force applied by the human operator on the arms. The model de-	
	picted on the right can be seen as desired dynamics for the motion	
	of the body and arms of the robot	.06
5.4	Top: velocity of the arms and base using the algorithm of Kosuge	
	with time delay. Middle: distance between the arms and base.	
	This distance must be kept slow enough to fit the workspace of the	
	robot. Bottom: virtual force applied on the base of the robot due	
	to the virtual coupling between the arms and base	.07
5.5	Velocity of the arms and base using the algorithm of Kosuge <i>with</i> -	
	out time delay. Compared to the case with time delay, the velocity	
	profiles are smooth	08
5.6	Top: lateral swinging motion of the waist during on-place stepping	
	of the robot. The amplitude of the motion is about 10 cm. Bottom:	
	reduced oscillations of the gripper (amplitude of about 1 mm) after	
	the compensation of the legs' motion	.09
5.7	Decoupled admittance control and gait: the admittance of the end-	
	effector of the robot is a damper fixed to the ground rather than	
	to the body of the robot, allowing to decouple force-based control	_
	from the motion of the body of the robot	09
5.8	Reference frames used for the synchronization of the feet and hands. I	10
5.9	Safe area for the landing foot	11

5.10	Graph of entities at a given time of the control (particular case of the teleoperation experiment presented in the previous chapter). Such a graph is useful to have an idea of the data flow in a given control architecture and can be generated using a simple command line	113
5.11	Simplified view of the architecture used for the collaborative trans- portation experiment. OpenHRP is used as a proxy between the robot and the Stack of Tasks control architecture. From the state of the robot, a relative position error between the end-effectors and the feet of the robot is computed, and resulting footprints are planned and sent to the pattern generator. From these footprints, the pattern generator computes a reference CoM trajectory and ref- erence trajectories for the feet. The Stack of task then computes a global control law to realize the CoM and feet trajectories as well as the admittance control of the end-effectors, which has the lowest	
5.12	Reference and real ZMP in a direct interaction experiment. On the	114
5.13	right: zoom between $t = 40$ s and $t = 50$ s	117
	forces rarely exceed 10 N	118
5.14	Experimental results: direct interaction between a user and the HRP-2 robot	119
5.15	Cooperative table transportation between an HRP-2 humanoid	112
5.16	robot and a human operator	120
	corresponds to the collaborative transportation phase	121
A.1 A.2	Two-port model of a teleoperation setup interacting with the hu- man operator on master site and the remote environment Two-port model of a teleoperation setup with bounded impedances	129
A.3 A.4 A.5 A.6 A.7	at the interaction ports	<ol> <li>130</li> <li>132</li> <li>132</li> <li>133</li> <li>134</li> <li>134</li> </ol>
B.1	Experimental setup: collaborative lifting task between an HRP-2 humanoid robot and a human subject.	136
C.1	Structure of the demonstrator	143

C.2	The different steps of the simulation: (a) Go to initial configuration	
	(b) Trigger the robot by patting its hand (c) Show it the object (d)	
	Move the object together	144
C.3	Interacting with the HRP-2 avatar using a haptic device	144
D.1	Mechanical parts of the new SensAble 6dof stylus with force feed-	
	back integration	149
D.2	Different views (left side and isometric) of the new PHANTOM	
	6dof handle with the 6dof Nano43 force sensor from ATI	150
D.3	A 6dof PHANTOM device with 6DOF sensing sensor from ATI	
	(Nano43) and the new handle.	150

**List of Tables** 

# Haptic interaction: a survey

## Contents

1.1	Introdu	uction	3
	1.1.1	Humanoid robots	3
	1.1.2	Haptic collaborative tasks	3
	1.1.3	Interaction control and the leader-follower model	5
	1.1.4	Human-human collaboration	5
1.2	Imped	ance control	7
	1.2.1	Force control and compliance control	7
	1.2.2	Impedance control	8
	1.2.3	Contact instability	14
	1.2.4	Conclusion	15
1.3	Imped	ance control and collaborative tasks	15
	1.3.1	Why impedance control?	16
	1.3.2	Applications in the previous works	17
	1.3.3	Limitations	19
	1.3.4	Conclusion	21
1.4	Recent	t advances: towards perfect followers?	22
	1.4.1	Human impedance models	22
	1.4.2	Active following	23
	1.4.3	Proactive behaviors	25
	1.4.4	Leader robots?	25
	1.4.5	Conclusion	26
1.5	Conclu	usion	26

**O**NE of the features of human beings is their ability to collaborate to perform tasks. Collaborative tasks range from searching an object, piloting a plane to handing over objects, dancing, assembly tasks or daily bulky or heavy object transportation, etc. In this work, we are interested by tasks which involve physical contacts between two partners, typically manipulation tasks where more than one agent act on an object of interest to the task. More specifically, our main target is to achieve a similar task on virtual or robotic avatars such as humanoids (virtual or robotic ones). In the long term, we aim at enhancing robotic or virtual avatars with physical interaction cognition so that they become more human-like partners in physical collaborative tasks. In this chapter, we introduce haptic collaborative tasks and present different problems that must be solved to give a robotic system the ability to perform such tasks jointly with a human partner. We introduce the leader-follower model, which has been used (often implicitly) in most research works in the field of physical Human-Robot-Interaction (pHRI). We also present some knowledge gathered about human-human behavior in physical collaborative tasks. This knowledge gives insight on how to partially answer one of the problems that characterizes collaborative tasks, namely *load sharing*. Since most works in pHRI are based on impedance control, we recall its fundamentals and highlight the reason for its use in pHRI. Finally, a summary of the previous works in the field is given, showing a trend to give more and more responsibility to the robotic partners for the realization of collaborative tasks, which culminates in assigning leader roles to the robot, which is the keystone of next chapter.

## **1.1 Introduction**

#### 1.1.1 Humanoid robots

Humanoid robots are robotic platforms whose overall shape is similar to the human body: it generally features a torso, two arms, and a head. Some platforms consist in a humanoid torso mounted on a mobile base, such as the ARMAR humanoid robot [4], and other are bipeds, such as Honda's ASIMO, or robots from the HRP series. In this thesis, we will implicitly refer to this second category when mentioning humanoid robots.

Biped humanoid robots are redundant, free-floating robots, which require appropriate motion control strategies to synchronize all the parts of their body to perform different tasks. As walking systems, humanoid robots have a wide workspace, but can fall (lose their static or dynamic balance). Static walking refers to gait where the robot is constantly in static equilibrium. This condition is not necessary to walk without falling: the ZMP criterion [114] has been widely used as a basis for dynamic walking.

#### **1.1.2 Haptic collaborative tasks**

The goal of this work is to study how a humanoid robot can be endowed with the ability to perform haptic collaborative tasks with a human operator. The redundancy that characterizes humanoid robots and their shape make them polyvalent tools that can potentially exert various physical works (pick up and manipulate objects, repair vehicles, assembly tasks) in various environments (homes, offices or hostile environments). Contrary to industrial robots, humanoid robots are expected to spread in everyday's environment such as homes and offices. Thus, humanoid robots must be able to operate in the same workspace as humans, and are likely to interact with them. One paradigm of interaction takes form through physical contact between the agents, and occurs in different contexts, such as involuntary contact, pushing aside, handshaking, dancing, transporting objects and so on.

Situations where the dynamics of a human and a robot are coupled by direct contact or contact through an object, are referred to as *physical Human-Robot Interaction*, or pHRI. We will focus on the case where the contact is intentional, and established by the human and robotic partners to perform a collaborative task. In such configurations, we believe that haptic cues play an important role, hence we will call these tasks *haptic collaborative tasks*.

More precisely, we will study dyadic *collaborative manipulation tasks*. A collaborative manipulation task is an instance of haptic collaboration whose goal is to modify the position and orientation of an object with time, by applying forces on the object. We look at the case where two partners participate to the task, and aim at the realization of such tasks by a human operator and a humanoid robot. An instance of collaborative manipulation task is shown on figure 1.1.

To achieve a task such as the one depicted on figure 1.1, a *mutual adaptation process* must take place between the partners, at different levels, which are detailed below.

#### **Dynamic interaction**

The dynamics of the human operator and the humanoid are coupled by contact with a common object. This must somehow be taken into account in their con-



Figure 1.1 – Collaborative transportation task between a human and a humanoid robot.

trol strategies: their motions must be coordinated to keep contact with the object and maintain interaction forces at a reasonable level while moving towards a goal configuration. When both systems are coupled, they also have to take into account their respective constraints. As we are considering bipeds, one very important constraint is keeping dynamic balance. The mechanical power flow between the partners must be appropriately absorbed or transformed into kinetic energy in order not to fall. One way to do this is to perform steps.

#### Load sharing

The goal of collaborating to perform a manipulation task is to share the task load. For simple tasks, this mainly consists of sharing the *physical* load, *i.e.* by sharing the inertial load and countering the effects of friction or any other dissipative effect. For more complex tasks, this can also imply sharing a *cognitive* load. When performing a collaborative manipulation task, the partners have to agree on a load sharing strategy so as to act in a complementary way.

We can imagine several ways to share the cognitive load during a task, by decomposing the task into subtasks. For example, when transporting a table upon which a vase is standing, one partner can ensure a desired trajectory of the table in the horizontal plane, while the other can adjust the altitude of the table and ensure it is kept horizontal. There is no guarantee, however, that this decomposition will be natural for a human operator. Hence, to devise cognitive load sharing strategies for a robotic partner, we must first understand how humans themselves decompose tasks, if they do so.

The problem of sharing the physical load is unfortunately not simpler. In order to move an object, only a 6-dimensional force/torque vector is necessary. When two partners move the same object, they form a redundant system, as an infinite set of pairs of force vectors can be applied to result in the same motion. The set of possibilities is only limited by the constraints of each partner, namely, their balance and their muscular force. Here also, the study of human-human collaboration can provide useful information.

#### Planning

When two humans perform a collaborative manipulation task, they both might have their own initial plan. However, the object will finally follow only one trajectory to move from its initial to its final configuration. This means that either one of the partner has to totally ignore her/his initial plan, or that the plans of both partners must be adapted to result in a final, common plan. Somehow, we can say that collaborative tasks must be *negotiated* by the partners.

#### 1.1.3 Interaction control and the leader-follower model

Previous works in pHRI are articulated around two concepts: *interaction control*, and the *leader-follower model*. Interaction control is an approach of robot control that addresses the problem of dynamic interaction between robots and their environment. Whereas servo control aims at tracking reference signals, interaction control places *interactive behavior* above command following in the design criteria [15]. A well-known instance of interaction is impedance control [44], which has been used as a basis for a great amount of research in pHRI [3,64,70,86,110].

Besides impedance control, a common model use in pHRI is the leaderfollower model. According to this model, one agent (the leader) has knowledge about the tasks and imposes his plan to his partners. Generally, the robot is assigned a follower role. While most works imposed a passive behavior to the robot, some researchers argued that passive following increased the necessary work to be performed by the human operator in order to move the object, since the robot is dissipating part of it [16, 70]. They proposed active following schemes for pointto-point tasks, where the robot predicts the human motion, based on the minimum jerk model [31]. In such schemes, the limit between the leader and the follower roles becomes fuzzier. When the robot fails to estimate the human intentions, the human partner can adapt to the resulting motion of the robot instead of sticking to his or her original intentions. Hence, active following does not mean the absence of mutual adaptation.

Even if they can actively follow human operators, robotic systems have generally not been programmed to have their own task plan and negotiate it with the human operator: the human partner is almost always leading the task. We believe that such a *binary, fixed role distribution does not reflect* the strategies adopted by two humans when they jointly perform physical tasks. We propose to *extend the leader-follower model to a continuous, time-varying distribution between the two roles*. This idea is formalized in the next chapter and is the keystone of the first part of this thesis.

#### 1.1.4 Human-human collaboration

To implement efficient collaborative behaviors on a robotic system, it can be interesting to understand how humans themselves perform collaborative tasks. The study in [75] shows that the minimum jerk model does not hold for collaborative point-to-point tasks. In this study, an object is moved from one place to another, in two conditions: by one partner alone, or by two partners. When the task was performed by two partners, the mass of the object was doubled. The comparison between these two conditions shows that the object reaches higher altitudes when manipulated by two partners. However, the velocity profiles obtained in the case where two partners manipulate the object was close to the average between the velocity profiles obtained with each partner alone.

In his thesis, Kyle Reed thoroughly studies the state-of-the-art in humanhuman physical interaction and in the related fields: human motion control in free space, human-robot physical interaction and tele-operation [90]. Kyle Reed then studies a one degree-of-freedom symmetrical task, where two partners turn a crank to reach a common target angle. He noticed, among others, a difference force (which produces no motion) in the steady state, when partners have reached the target and stop. He speculates that this contraction could help partners stabilize the system or feel each other. He also validated Fitt's law [30] for dyads, though he pointed out that Fitt's law did not apply when both partners were given conflicting target positions [92].

His major result is highlighting a phenomenon he named *specialization*. Experimental results showed that after several trials, the force patterns applied by the participants change so that each partner contributes to a subtask and hinders the other subtasks. During his experiments, most dyads adopted an acceleration/deceleration specialization: one partner will accelerate the crank at the beginning of the motion, while the other one will decelerate it during the second part of the movement. Other dyads, though eventually adopted a left/right specialization: one partner contributed to motions in one direction and vice-versa. The hypothetic reasons for specialization are the resolution of the redundancy and increase performance by focusing on one subtask only. He could yet not determine how the humans specialize (except that it establishes via the haptic channel), neither who specializes for which subtask, nor why. It is essential to underline that the task he studied was symmetrical, thus it did not suggest any specialization a priori.

Reed tried reproducing the performance of human-human dyads with humanrobot dyads. In his experiments, the robot was applying a "specialized" force profile that was obtained by averaging the profiles of a human subject. Though the setup did suggest a (complementary) specialization for the human, the latter kept an unspecialized force profile during all the trials. The mechanism of communication through which specialization emerges is thus not characterized, and yet not understood. Reed suggests that the human subjects did not specialize because the robot was specialized from the first trials. He cites the work of Corteville *et al.* [16], where a human adapted his motion according to the behavior of the robot. In these experiments, the robot was also adapting to the motion of the human. This adaptation of the robot to the human is maybe a necessary process for the human to also adapt his force profile.

Specialization could partly answer the problem of physical load sharing, as it can be seen as a way to resolve redundancy. The question is yet not totally solved, since the works just cited are purely *descriptive*. We do not know how to *generate* the described behavior, given the fact that what is described is a *joint behavior* of the robotic and human partners, and we can only control the robotic partner. However, from the works presented in this paragraph, we can extract the following information:

- ▷ the output behavior of a dyad as a whole can be (but is not always!) an average of the behaviors of the individuals [75];
- ▷ the physical load can be shared by specializing for subtasks [90];
- specialization is an emerging phenomenon, which means that any attempt to model it should take into account "degrees" of specialization (unspecialized, hardly specialized, totally specialized).

Specialization will be mentioned again in the next chapter, where it is related with *our continuous, time-varying, role distribution hypothesis*. The model proposed in the next chapter encompasses the three concluding remarks of this paragraph. The remainder of this chapter recalls the fundamentals about impedance control, and shows why it is a relevant control scheme for haptic collaboration. Then, a critical review of previous works in pHRI is presented and we evaluate to what extent the problem of physical collaboration is solved (or unsolved). We conclude this chapter by summarizing the main ideas defended in this thesis.

### **1.2 Impedance control**

When a human operator interacts with a robotic system, it is necessary to limit, to some extent, the interaction forces to prevent injuries of the human partner. Controlling forces is precisely the purpose of force control. However, among force controllers, two categories can be distinguished: *explicit force controllers*, and *implicit force controllers*. Controllers of the first category follow the design specifications of servo controllers, which focus on tracking a signal, treating the environment as a source of disturbances. Impedance control belongs to the second category, but restricting impedance control to a form of force control is restrictive. Impedance control actually falls under *interaction control*, which places interactive behavior above command following in the design specifications [15]. This section first recalls some facts about force control, and then goes into the details about impedance control, to highlight the aspects that make impedance control an appropriate control strategy for pHRI.

#### **1.2.1** Force control and compliance control

Motion control consists in sending commands to the robot so that a given point on the robot tracks a desired motion. Motion control is efficient for robotic tasks which do not involve interaction with the environment, such as spray-painting, or any task which only requires trajectory tracking in free space. When the motion of the effector is constrained, however, there exist directions along which the motion can no longer be controlled and is imposed by the environment. On these directions, the interaction force resulting from the environment constraint can be controlled.

Force control consists in controlling, directly or indirectly, the forces applied by the manipulator on the environment. Force control appeared in the early 1970s [49, 50, 100], and was formalized mainly in the second half of the decade [17, 74, 82, 94, 117]. Force control and position control are often jointly applied, in complementary subspaces [17, 74]. Force control is applied along the constrained direction of the manipulator, and position control is applied on the unconstrained ones.

Two different issues are actually considered in what is called "force control".

The first one, sometimes referred to as "explicit force control" consists of tracking a desired interaction force between a manipulator and its environment. This can be done in an open-loop fashion, based on the dynamic model of the manipulator and the environment, but more precision can be achieved by closing the loop around force measurements. Figure 1.2 shows a basic structure for closed-loop force control. Note that force controllers can be built around inner position or velocity control loops (see e.g. [58]).

The second one is concerned with positioning the manipulator while limiting to some extent the interaction forces. Stiffness, damping and impedance control fall under this category. This objective can be achieved by defining a relation between the position error and the contact force. Since the basic idea is to generate motions that comply to the environmental constraints, this strategy can be termed *compliance control* [44].

The first approach belongs to servo control, and as such does not take into account the dynamics of interaction between the manipulator and the environment. This means that if the robot interacts with a human operator, or with another robot,



Figure 1.2 – Structure of a closed-loop force servo controller. C is a compensator, G is a feedforward gain, D is a function applied to force measurements (typically, low-pass filtering), and P is the plant.

these will be considered as sources of disturbances and the actions they will take will be rejected by the controller. As will be shown in the remainder of this section, the second approach is more appropriate for pHRI, especially for manipulation tasks, which are formulated in terms of desired motions rather than desired forces.

A complete review of the field of force control, providing implementation details, can be found in [69]. The next paragraph gives further details about impedance control and briefly presents two categories of implementation of impedance controllers.

#### **1.2.2 Impedance control**

Impedance control [44] can be seen as a generalization of stiffness and damping control. The main point of impedance control is that it defines a physically consistent framework for the control of robotic systems that interact with their environment. This framework has been extended in [15], where the difference between *servo control* and *interaction control* are clearly stated in terms of *objectives*, resulting in two different approaches to design controllers: in interaction control, the so-called *interactive behavior* has more importance than command following.

#### Intuition

Position and force control are appropriate when the environment of the manipulator is perfectly modeled and can be treated as *kinematic constraints* on the robot's motion. Consider a robot with a torus end-effector sliding along a bar without friction: along the bar, the motion of the robot is not constrained, but in the orthogonal directions, no motion is possible. However, forces can be applied along these directions. Assuming the position of the bar is exactly known, hybrid controllers [88] will allow to perform any desired motion along the bar and to apply any desired force in the orthogonal directions. The only limitation to the motions and forces come from the capabilities of the robot itself. Note, also, that no mechanical power flows between the robot and its environment during the task.

Let us consider the same environment, where this time *dynamic* constraints are imposed to the manipulator: the manipulator is coupled to a mass, which can only slide along the bar, subject to viscous friction (see figure 1.3). This task corresponds to a one degree-of-freedom manipulation task, except that the contact between the robot and the object to be manipulated (the mass) is bilateral.



Figure 1.3 – Robot end-effector (represented by a torus) moving along a bar. The torus is coupled to a mass which moves along the bar and is subject to viscous friction.

Because the end-effector of the robot is coupled to the mass, its motion is no longer unconstrained. The motion of the end-effector will be constrained by the dynamic equations that govern the motion of the mass the robot is coupled to. Because of this coupling, on the same direction (along the bar), the robot will experience both force and motion: in other terms, mechanical power will flow between the robot and its environment.

If the mass and the damping it is subject to are significant, the robot will be able to accelerate the object arbitrarily or attain arbitrary high velocities only at the expense of a significant energy. The highest the mass is, the more energy will be necessary for high frequency motions. When the damping increases, lower velocities can be attained for the same mechanical power provided by the robot. This means that if the robot acts as a position source (which would the target behavior of a position controlled system), then to avoid too high energy flows, the desired motion of the system should be adapted. If this motion is computed off-line by a planner, this will severely limit the applicability of the system (each trajectory will be compatible with only a range of masses and dampings), or will require ad-hoc trajectory adaptations for new environments. In environments with uncertainties or variability, this approach is impractical.

To introduce impedance control as a solution to this power flow problem, let us use electrical analogies. Instead of considering force and motion, we will consider voltage and current. Intuitively, motion control with a servo approach aims at making the manipulator behave as a motion source. At steady state, this motion can be countered by resistive elements in the environment like dampers, whose electrical equivalent are resistors. Hence, a position controlled manipulator performing a manipulation task at steady state can be modelled by the electrical network shown on fig 1.4.

For very resistive environments, the power flow between the manipulator and the environment is  $P = R i_d^2$  and will increase with the resistance of the environment (*i.e.* as the environment admittance decreases to zero) for arbitrarily high desired currents  $i_d$ . One way to control the power flow between the manipulator and the environment is to add a second resistor in parallel with the environment, as shown in figure 1.5.

In figure 1.5, the manipulator no longer appears as a current (or motion) source,



Figure 1.4 – *Electrical equivalent of a manipulation task at steady state performed by a motion controlled manipulator.* 



Figure 1.5 – *Electrical equivalent of a manipulation task at steady state performed by an impedance controlled manipulator.* 

but as a voltage source, where the voltage depends on the difference between the desired current and the current flowing through the environment. The voltage across the environment terminals and the power flow between the manipulator and the environment are given by:

$$u_{d} = R_{d} (i_{d} - i)$$

$$P = \frac{R_{d}^{2} (i_{d} - i)^{2}}{R}$$
(1.1)

where  $R_d$  is a controller gain. Now, if R increases, the network is transformed into the circuit of figure 1.4, except that this time, the resistance is  $R_d$  and is *set* by the designer. Hence, in case of very resistive environments, the power flow will be bound by  $R_d i_d^2$ . When the resistance of the environment becomes close to zero, then  $i = i_d$ ,  $u_d = 0$  and no power flows through the environment. This degenerated case corresponds to unconstrained motion, and the equivalent network is no longer feasible, since current flows are ideal elements. The important point is that the power flow between the manipulator and the environment will tend to zero as the environment resistance decreases, and be bounded by  $R_d i_d^2$  as the environment resistance increases. Of course, the power flow can still grow to infinity for a given value of  $R_d$  for arbitrarily large desired currents  $i_d$ , but the bounds on the power flow now *entirely depends on the manipulator control system* and do not depend on the environment. Moreover, the resistance parameter  $R_d$  can be adjusted to allow for higher desired velocities for a given maximum power flow, at the cost, of course, of potentially larger current errors  $i_0 = i_d - i$ .

This electrical analogy illustrates the purpose of impedance control. In impedance control, the target system is not a pure position source, but a position source with a target dynamic response to motion errors. The difference with pure motion control is that the primary goal is *not* to reject all disturbances and track a desired motion. The primary goal is the dynamics of the response to the motion errors. Command following comes after, and will be ensured when no disturbance occurs. One of the effects is that the mechanical power flow between the manipulator and the environment is somehow controlled, which is an important criterion to avoid physical damage of the manipulator or the manipulated object.

One can argue that motion control would in reality have exactly the same structure as impedance control. In our electrical network on figure 1.5, by using very high values for  $R_d$ , we tend to make our manipulator behave as a pure current source. Likewise, in control theory, position controllers are often implemented using Proportional-Derivative (PD) controllers, which corresponds to coupling the object to the desired position using a virtual spring-damper system, which is a common target dynamics in impedance control, as will be seen further in this section. Stated like this, there seems to be no difference in the structure of a PD position controller and an impedance controller. However, the *goal* of position control is to make the manipulator behave as a motion source, which is not the case in impedance control. On figure 1.5, position control will want very high values for  $R_d$ , while this is generally not the case with impedance control.

Note that the general case of impedance control considers any form of impedance. Our electrical example contained only resistors, which are the electrical equivalent of dampers. More general impedances could have been considered by including inductances (the equivalent of masses) and capacitors (the equivalent of mechanical springs). We now briefly present the history of impedance control, admittance control, and, more generally, interaction control.

#### Impedance control: a history

Impedance control was proposed by Neville Hogan [44] as an alternative to pure position or force control and hybrid control. As Hogan pointed out, neither position nor force control allows the control of the mechanical energy exchanged between a robotic system and its environment. Since the mechanical energy transfer at the interaction port depends on both the velocity and force variables, he proposed to impose a relation between them.

The approach proposed by Hogan consists of imposing a dynamic behavior to the system. Thus, instead of tracking either the interaction force or velocity of the robot's tool tip, a *dynamic* relation between the two variables is enforced. This allows the control of the mechanical power flow between the robot and its environment, which is defined by the product of force and velocity. Hogan also postulates that no controller can make the robot manipulator appear as anything but a physical system to the environment. As a result, the manipulator can not impose both a velocity and a force to the environment at the interaction port, but only one of them. Two categories of systems can be distinguished: impedances, which input flow and output effort, and admittances, which input effort and output flow. As the environment of a robotic manipulator arm is mainly composed of inertial loads, which are admittances, Hogan states that, to be compatible with its environment, the manipulator should be controlled to appear as an impedance, hence the term "impedance control".

Citing the case of robots jointly performing a task, Colgate suggests that the control of systems that dynamically interact with their environment requires a different approach from servo control [15]. In servo control, the environment of the control system is generally seen as a source of disturbances to be rejected. This approach is not suitable when the environment is composed of other robots or human operators. Compared to servo control, Colgate adds coupled stability and *interactive behavior* as design criteria for interaction control. Coupled stability is obtained if the system remains stable when coupled to any environment. Interactive behavior focuses on the dynamics of the overall system in interaction with its environment, whereas in servo control, the stress is put on tracking a desired trajectory for one single variable (in the Single Input, Single Output case). Colgate also derives a necessary and sufficient condition for linear systems to be stably coupled to passive environment: the controlled system must exhibit a positive real driving point impedance. This result is extended to active system under the condition that the system and the state-dependant control together are passive, and the active part is state-independent. Finally, a systematic approach is proposed to design interaction controllers to obtain both stable coupling and good performance (namely, command following and interactive behavior). The coupled stability constraint relies on the concept of "worst environment", which are the most destabilizing.

Whereas Hogan states that manipulator arms physically interacting with their environment should behave as impedances, a common control scheme for robots physically interacting with their environment consists in taking into account force feedback within the position control loop. This corresponds to admittance control. Admittance controllers are easily built upon low level position controllers by computing the position command depending on the force feedback. They are suitable on platforms with high inertia and friction. More specifically, dry friction tends to decrease the performance of force controllers. This problem can be overcome by using a high-gain position controller; the admittance controller then computes position commands from a force signal, which has to be measured by a sensor. Admittance controller can be implemented on devices with a large workspace and can display high stiffnesses, but generally have a reduced bandwidth compared to impedance controllers [112]. Note that when the robot is directly interacting with a human (such as haptic interfaces), if we consider the human as an impedance, then the robot can be admittance controlled without breaking causality.

It has been suggested [57] that "impedance control" would be an inappropriate term to refer to the control strategy proposed by Hogan. The main argument is that the input of the controller is a desired position, and impedance controllers simply define a dynamical relationship between the force exerted on the environment and the position error. Hence, impedance control would fall under position control, and since the generated motion is compliant, the term "compliant motion control" is proposed. This is merely a matter of vocabulary, nevertheless it is important to state that the approach adopted in impedance control and position control are totally different. Impedance control does not belong to servo control, whereas position control traditionally does. In position control, the environment is seen as a source of disturbances to be rejected, and command following has a high priority in the design criteria [15]. Impedance control is rather focused on the dynamic behavior of the overall system when the environment and manipulator dynamics are coupled (as in contact situations) [44, 47]. Command following has a lower priority with respect to interactive behavior [15].

To conclude about this vocabulary issue, let us say that in impedance control (or compliant motion control), finally, neither the force nor the position are controlled, since they both depend on the environment the manipulator interacts with. The position will only be controlled in the free space, while no force command is explicitly given to the controller. The only defined entity is the manipulator *impedance*, *i.e.* its force response to motion inputs. Thus, we feel that "impedance control" is as valid as a term as "compliant motion control", keeping in mind that this control strategy does not belong to servo control anyway.

#### **Impedance control implementations**

Roughly speaking, impedance control consists in defining a target dynamics and controlling the manipulator to track this desired dynamics. A usual target dynamic behavior is to make the end-effector behave as a linear mass-spring-damper system, such as in figure 1.6.



Figure 1.6 – Target mass-spring-damper system. M is the target mass, B the target damping, K the target stiffness,  $x_0$  is the virtual compliance center and x is the position of the mass, which corresponds to the position of the end-effector.

The target system dynamics is described by the following equation:

$$\mathbf{M}\,\ddot{\mathbf{x}} = \mathbf{f} + \mathbf{B}\,(\dot{\mathbf{x}}_0 - \dot{\mathbf{x}}) + \mathbf{K}\,(\mathbf{x}_0 - \mathbf{x}) \tag{1.2}$$

where **M**, **B**, **K** are the target mass, damping and stiffness of the system, x is the position of the end-effector of the manipulator,  $x_0$  is the equilibrium position, and **f** is the external force applied on the end-effector.

The initial implementations proposed by Hogan [45, 46] target torque controlled manipulators, *i.e.* manipulators that are controlled by specifying desired actuator torques. On such manipulators, impedance control can be implemented using resolved acceleration control. If force sensing is not available, but the dynamic model of the manipulator is known, arbitrary damping and stiffness coefficients can be chosen, but the manipulator inertia can not be changed: that is, the target mass M must be the actual apparent mass of the manipulator, obtained by projecting the inertia matrix in the operational space. If the dynamic model of the manipulator is not available, the target dynamics can be approximated, if the target inertia is the actual manipulator inertia, by specifying the following actuator torques:

$$\tau_{\mathbf{a}} = \mathbf{J}(\mathbf{q})^T \left( \mathbf{B} \left( \dot{\mathbf{x}}_{\mathbf{0}} - \dot{\mathbf{x}} \right) + \mathbf{K} \left( \mathbf{x}_{\mathbf{0}} - \mathbf{x} \right) \right)$$
(1.3)

where J(q) is the manipulator jacobian, and q is the configuration vector of the manipulator.

Many robotic platforms cannot be controlled by specifying desired torques, but by specifying desired positions or velocities. In this case, the impedance controller is built around the inner position or velocity control loop. This means that the output of the controller is no longer a force, but a motion. For this reason, the control scheme might be considered as an "admittance controller". However, since we consider a nominal position, which will be tracked whenever the manipulator is in free space, we can still consider motions as the input of the overall control system. Hence, the term "position based impedance control" can be found in the literature. We will use either of them in this thesis.

Position-based impedance controllers require force sensing devices. The sensed force is integrated using equation (1.2), to obtain a reference velocity  $\dot{\mathbf{x}}$  and/or a reference position  $\mathbf{x}$  to send to the inner position controller. Figure 1.7 shows the structure of a position-based impedance controller.



Figure 1.7 – Position-based impedance controller. A is the target impedance, C is a position controller, and P is the plant.

Position based impedance controllers offer the advantage of a certain simplicity of implementation, since the manipulator dynamics are masked behind the inner motion control loop. However, they loose the advantage of passive compliance, which is a significant on redundant manipulators. Indeed, the output of a positionbased impedance controller is a vector of joint positions to be sent to an inner high-gain joint position controller. This means that the posture of the robot will be fixed for a given sensed force, even if the manipulator is redundant. On the other hand, torque-based impedance controllers produce an output torque as a reference for each joint of the manipulator. It is then possible for an operator to modify the posture of a redundant robot by applying forces at arbitrary points of the robot; the posture of the robot will be modified according to the dynamics of the manipulator under the constraint that the end-point dynamic behavior imposed by the impedance controller is unchanged. Position-based impedance controllers also suffer from other drawbacks: the performance of such controllers is dependant on the quality of the inner position control loop [67]. Finally, time delays in the control law computation, and due to the transmission dynamics tend to have a greater impact on contact instability.

#### **1.2.3** Contact instability

Stability (having a bounded output in response to bounded inputs) is the most important design criteria for a controller. Among the robotic tasks, those that involve physical contact with the environment are prone to instability. In position and force control, these instabilities can arise because these approaches focus on the regulation of one variable and neglect the dynamics of interaction [46]. However, the dynamics of the robotic system before and after physical contact dramatically

change. Moreover, the disturbances can no longer be considered as independent from the controlled system since the robot is dynamically interacting with its environment [47]. Actually, there are well-known sources of contact instability that limit the efficiency of any force feedback controller, including impedance controllers: the most critical ones are maybe the non-collocation of sensors and actuators, and the transmission dynamics [14, 22]. Colgate highlighted a fundamental limitation to the performance of force feedback controllers, resulting from the non-collocation of actuators and sensors: the apparent end-point inertia of a system controlled with force feedback can not be reduced by more than half without risking instability when interacting with the environment. Other researchers studied the stability properties of different implementations of impedance control [67]. Position-based impedance control (or admittance control) appears to have different stability boundaries from torque-based control. While the stability of the latter is only affected by time delay, the former is also affected by the performance of the inner position control loop. These effects decrease as the position control gains increase. These results are consistent with [14, 22] since the non-collocation of sensors and actuators causes time delay.

#### 1.2.4 Conclusion

This section has recalled some fundamental aspects of impedance control. After having shown how it distinguishes from explicit force control schemes, a simple example has been introduced to show how impedance control can act as an energy regulator. The history of impedance control has been quickly presented, as well as the two classes of implementation of impedance controllers. Finally, some elements about the stability of impedance controllers have been briefly mentioned, to complete the introduction.

Impedance control has played an essential role in the history of pHRI. This is partly due to the fact that impedance control is suitable for manipulation tasks, which collaborative manipulation tasks are a subset of. But more importantly, the key feature of impedance control is that it regulates the mechanical power flow between the robot and the environment during physical interactions, which is safe if the environment the robot interacts with is another human. Finally, impedance control is a way to make a robot comply to external forces, which is one way to implement follower robots, as will be shown in the next section.

## **1.3** Impedance control and collaborative tasks

Impedance control has been proposed as way to regulate the mechanical energy exchanged between a robot and its environment. Considering human-robot collaborative tasks as an exchange of mechanical energy between the partners, impedance control seems to be an appropriate basis to design control schemes that allow a robot to perform physical tasks jointly with a human partner. We will show in this section why impedance control *is* relevant in the context of collaborative tasks.

However, we will also show that, the general problem of pHRI cannot be solved by impedance control on its own 1.3.3. Impedance control tells us to enforce a relationship between force and motion, but it does not say, in general, how they must be related. Impedance controllers have parameters that can vary, and can take a virtual position input. It is not sufficient just to define parameters and trajectories offline: even if impedance control will guarantee that the mechanical energy flow between the robot and its environment will be controlled in case of disagreement between the partners, it does not guarantee that the task can be accomplished. Impedance control provides a robust basis to address the low-level aspects of physical collaborative tasks, but leaves a blank page in the chapter on high-level control and communication aspects induced by such tasks between the partners.

#### **1.3.1** Why impedance control?

We place ourselves in the context of a physical collaborative task, where two partners jointly manipulate an object. We will consider a simple instance of a collaborative manipulation task, depicted on figure 1.8. Both partners are considered as force sources. The forces  $f_h$  (h for human) and  $f_r$  (r for robot) applied to the object result in an acceleration  $\ddot{x}$ . The motion of the object is described by the equation:

$$\ddot{x} = M^{-1} \left( f_h + f_r \right) \tag{1.4}$$

where *M* is the mass of the object.



Figure 1.8 - One dof point-to-point task.

The goal of the task is to bring the object from a location  $x_i$  to a location  $x_f$ . The collaborative motion problem can be stated as: given these locations, what forces should the partners exert on the object to reach the target location?

One solution in general to bring an object to a desired location is position control. To join  $x_i$  to  $x_f$ , a desired trajectory  $x_d(t)$  can be computed, for instance a trajectory minimizing jerk, as seems to be generally used by humans [31]. The position of the object is then "servoed" to track  $x_d(t)$ . While this is easily performed when one agent acts on the object, it is no longer a suitable approach when considering joint manipulation. If we extend this approach to the case where two agents act on the object, we have two desired trajectories,  $x_d^h(t)$  and  $x_d^r(t)$ , one for each agent. However, both agents cannot act as position sources, since in the end, the object can have only one position. This means that position control would be possible only in the very improbable case where both partners have exactly the same desired trajectory  $x_d(t) = x_d^h(t) = x_d^r(t)$ . If both partners attempt to behave as position sources, conflicts in the desired trajectories will result in significant interaction forces.

On the same degree of freedom, there can be only one pure position source. This means that only one agent can impose a desired trajectory, hence only one agent should be position controlled. But in this case, what force should the other partner apply? Any force applied that disturbs the desired motion of the position controlled partner will be rejected: the second partner is finally pretty useless and this scenario hardly corresponds to a collaboration. Moreover, the position controlled partner finally finds her/himself performing a manipulation task in presence of disturbances using position control, as shown on figure 1.9. We have seen in the previous section (1.2.2) that this approach could lead to potential problems.



Figure 1.9 – Servo-control approach to collaborative manipulation. One partner tries to impose a desired position  $X_d$  to the object. The partner applying the force  $F_d$  appears as a disturbance to the other partner.

We have seen that impedance control consists in defining a desired force response of a system to positioning errors, rather than making the system appear as a position source. Applied to collaborative manipulation, this would lead to defining a dynamic response for each partner to the disturbances that occur because of the differences in their target trajectories. If one partner exerts no force on the object, then the other partner is free to impose a desired trajectory to the object. Otherwise, even in case of conflicts, the interaction forces will be limited. Moreover, impedance control allows symmetry in the controllers of the partners. Impedance control applied to the example shown in figure 1.8 gives the following expressions for  $f_h$  and  $f_r$ :

$$\begin{aligned}
f_h &= B_h \left( \dot{x}_d^h - \dot{x} \right) + K_h \left( x_d^h - x \right) \\
f_r &= B_r \left( \dot{x}_d^r - \dot{x} \right) + K_r \left( x_d^r - x \right)
\end{aligned} \tag{1.5}$$

Again, a position controller could be implemented using the same structure for each partner. However, the design criteria for the choice of  $B_h$ ,  $K_h$ ,  $B_r$  and  $K_r$  would have been different. In position control, the gains would be set to enforce position tracking, which would lead to high values for all gains. In impedance control, the gains will be set to guarantee limited interaction forces in case of motion errors.

To further illustrate the difference in position control and impedance control design, figure 1.10 shows the equivalent target networks corresponding to position control for both partners, position control for one partner and force control for the other, and impedance control for both partners, in the context of the example task illustrated in figure 1.8.

Now that impedance control has been introduced as a settlement for collaborative manipulation tasks, we will review its implementations and applications in the last decades.

#### **1.3.2** Applications in the previous works

#### **Desired dynamics**

Impedance control has been used as a basis for a great amount of research aiming at designing robot helpers to physically assist human operators [3, 64, 70, 86, 110].

The idea in [64], as in [39, 40, 42], is to impose a suitable dynamics to the system composed by the manipulated object and the helper robot(s). In [64], the dynamics of the system is the one of a linear impedance and opposes a reaction


Figure 1.10 – Target equivalent networks for different control schemes: on the left, both partners are position controlled and try to act as a position source. The target network is not feasible. Middle: one of the partners only is position controlled, and the other acts as a force source. The current that flows through the inductance corresponds to the motion of the object and is imposed by only one partner. On the right: target network when both partners are impedance controlled. The network is feasible, and symmetric: both partners can act on the motion of the object.

force  $\mathbf{f}_{ext}$  to its motion:

$$\mathbf{f}_{\mathbf{ext}} = \mathbf{M} \, \ddot{\mathbf{x}} + \mathbf{B} \, \dot{\mathbf{x}} \tag{1.6}$$

where  $\mathbf{x}$  is a generalized coordinate vector of a point fixed on the manipulated object. The global system comprising the object, the robotic systems and the human operators is shown to be stable assuming a passive behavior for the human operators, and assuming that the system is controlled to have exactly the prescribed dynamics.

Complex tasks where a long object is manipulated raise an ambiguity regarding the interpretation of forces/torques at the grasping point. To generate pure lateral motion in coordination with the helper robot, high torques must be applied by the human operator to compensate for the momentum of the force s/he applied on the object, while a lateral drift will tend to occur when the operator wants to perform a pure rotational motion. This problem has been tackled in [3] by imposing a virtual nonholonomic constraint to the robot, resulting in caster-like motions of the system, a motion human operators are familiar with. Caster-like motions have largely been used for the manipulation of long objects [39, 40, 42], but alternative strategies have also been proposed [118], where the objects is imposed a pump-like motion. This problem has also been addressed using oral communication to switch between rotational and translational motions [120].

Several teams proposed variable impedance controllers to assign more complex dynamics to the system. In [110], the collaborative task is divided into different phases based on a difficulty index, with different sets of impedance parameters for the different phases. Duchaine *et al.* [20] proposed to use the time derivative of the force applied by the human operator to modify the damping parameter of their impedance controller, based on the idea that this signal conveys information on the human intentions. In order to produce more human-like motions when a robotic system assists a human operator by following him, the impedance of a human arm has been identified and embedded in an impedance controller [86]. In the context of this investigation, the dynamic behavior of the arm was modeled as a second order linear impedance (mass-spring-damper system) with variable parameters. It has been found that the overall dynamics was dominated by the damping term [48].

In the aforementioned works, the robot behaves passively. For simple tasks where a model can be used to predict the human motion, active following can also be implemented. The case of point-to-point tasks has been studied in [16, 70]. For more complex interactions, like dancing [65], the use of Hidden Markov Models

(HMM) has been proposed [107], as a way to predict the next dance step, based on the forces applied by the human dancer on the robot partner, and a knowledge of the different steps and possible transitions.

Some of these works will be analyzed in more detail in the next section, where we see how the performance of robotic followers has been improved by assigning them more active behaviors.

#### Stability

In practice, the desired dynamics defined by the impedance controllers differs from the real dynamics of the system. For instance, many industrial platforms embed a low-level position or velocity controller, and impedance controllers are implemented as an outer control loop which sends position or velocity commands to the inner control loop. Such setups require the tool tip or end effector of the robot to be mounted with a force sensor. The non-collocation of the force sensor with the actuators, plus the time delay due to the sampling rate and the actuator and sensor dynamics make it impossible to track exactly the dynamics prescribed by the impedance controller (one may argue that the sensor and actuator dynamics might be modelled and taken into account in the controller to track the desired impedance, but whatever the sophistication of the controller, it is *never* possible to compensate for the *time delay*). These factors, apart from inducing an impedance error, can destabilize the system. In the specific context of human-robot interaction, a study performed by Tsumugiwa et al. considered the impact of the sensor dynamics and the human reaction time on the stability of impedance control [111]. Without considering time delay, a common approach is to impose a passive behavior to the robot to guarantee the stability of the system [39, 42]. In [8], the authors propose a less conservative stability criterion to allow for easier human-robot interaction by taking into account the impedance characteristics of the target environment the robot is to interact with (namely, a human operator). The authors also propose a computational design method for interaction controllers based on the optimization of a performance measure under stability constraints. In some recent work, Lyapunov stability was considered for impedance controlled devices, to obtain critical values of the impedance parameters [21]. This work was later used as a basis to design a stable adaptive admittance controller [19].

# 1.3.3 Limitations

Impedance control as a model for the motion control of partners jointly performing a manipulation task allows a symmetric representation of the partners in dyadic tasks, and the extension to tasks with more than two participants. It allows the partners to accommodate to differences in their desired trajectories without resulting in too large interaction forces. Therefore, it partially solves the problem of dynamic interaction between the partners. The problem is only partially solved, because impedance control does not explicitly take into account the nature of the controlled system and assumes fixed-base manipulators (see e.g. the implementation suggested in [46]).

Our work is concerned with humanoid robots, which are highly redundant, underactuated systems. The fact that their base is not fixed, and the dynamic balance constraint, limit the range of achievable impedances by the robot. We will see in the second part of this thesis that *the application of impedance control without regard for the specific nature of humanoid robots* leads to unsatisfying results. The fact is, impedance control is merely a principle, and further work must be performed to implement it on underactuated platforms such as humanoid robots.

Restraining ourselves to the one degree-of-freedom case, impedance control leaves several problems unsolved. Many of the previous works in the field of pHRI considered robots deprived of any intelligence, and considered that only the human partner had knowledge about the task to perform. In terms of collaborative manipulation, this means that the robotic system has *a priori* no desired trajectory. Hence, the robot is often programmed to be fully passive. This is illustrated on figure 1.11 in terms of electrical equivalents.



Figure 1.11 – Passive impedance control.

As a result, the robot appears as a purely dissipative elements, which makes the environment more resistive then when the human operator performs the task on her/his own. This fact is pointed out in [16]. Still, passive impedance control is not useless: when manipulating a heavy object, a robot can be programmed to keep the object at a fixed altitude (in this case, it acts as a position source, so the human cannot control the altitude of the object). On the horizontal plane, the robot will behave passively, so that the human can move the object just as if it had a slightly larger inertia and was subject to some damping. The robot will be programmed to be as little dissipative as possible on the horizontal plane, under stability constraints. The human is at least relieved from the task of fighting against gravity. Note that using force feedback, it is also possible to control the robot to behave so that the system composed of the robot and the object is passive, with a reduced apparent inertia, following the arguments given in [46]. However, in practice, the apparent inertia can not be significantly decreased without causing instability [14], and additional damping will hinder high velocity motions from the human and requires her/him to constantly input mechanical power into the system to keep a constant velocity.

To relieve the operator from this additional workload, researchers followed two different paths. The first one consists in adjusting and varying the impedance parameters. On the other hand, to actively, or even better, *proactively* assist the human operator, the robotic partner must be endowed with higher level controllers that will input suitable commands into the low level impedance controller. The first approach is the safest, because the apparent system dynamics is still the one of a passive system, with varying parameters. The second approach goes further, and allows the robot to input energy into the system, by specifying desired motions to the robot. This approach, first applied in [70] to point-to-point tasks, necessitates either knowledge about the task, and/or a way to predict the desired motion of the human. As soon as the robot can have a desired motion, conflicts with the human intentions will result in task failure. This problem goes beyond what impedance control can address, and concerns higher level control.

Finally, when we consider robotic systems that can input mechanical power to the object to follow a desired path, the question arises of what amount of energy will be input by the robot, and what amount will be input by the human. Considering that their respective desired motions can be slightly different, the actual trajectory of the object will depend on the respective amount of mechanical power input by both partners into the system. This problem of load sharing arises on each degree-of-freedom of a task, and on each subtask that compose a complex task. This also goes beyond the impedance control principles, and, assuming low-level impedance controllers, can be formulated as: how can impedance be adjusted in each actor to reach a desired load sharing. This problem is formalized in the next chapter.

#### 1.3.4 Conclusion

In this section, we have related impedance control to collaborative tasks. The problem of dyadic collaborative manipulation can be formulated as having two partners with two different desired trajectories to impose to one object. The partners then have to apply appropriate forces to try to track their respective desired trajectories, while maintaining a reasonable energy flow through the object, given the potential discrepancies between their desired motions. Impedance control allows to solve this problem for small discrepancies, by limiting the forces applied by the partners in response to positioning errors.

In the early history of pHRI, however, robotic partners were not programmed to track a desired trajectory, but to only comply to the human partner's intentions. Impedance control was then used to assign a desired, intuitive (and often passive) dynamics to the robot, or to the whole {object + robot} system. As a result, a human operator is able to manipulate objects simply by applying forces on it. This can be used to transport bulky objects, where a robot will support the whole weight on the vertical axis, and comply to the forces applied by the human partner on the horizontal plane, according to the dynamics assigned by the impedance controller. We briefly presented the enhancements that have been brought to impedance controllers to decrease the workload of the human operator.

The stability of impedance controllers was also studied in the context of pHRI, because it is a key element to the human safety. A quick look at some works in the field shows that different approaches are possible, depending on the design objectives.

Finally, the limitations of impedance control as a stand-alone solution to implement collaborative behaviors have been highlighted. The first obvious limitation is that not any impedance can be displayed by any device. Free floating robots such as humanoids have a higher bound on the maximal impedance they can display without slipping or falling. More generally, impedance control is concerned with the lower level aspects of manipulation tasks, but does not tackle any decisional aspect, and no general rule exists to adjust the impedance parameters depending on the context of the task. These limitations have called for more research that has resulted in different enhancements. This is the topic of the next section, where we also show that the progress in pHRI comes with a more and more important role in the task. This will bring the robot from the role of passive follower to the role of proactive partner, and the whole point of the next chapter will be to see how to go even beyond, and assign a leader role to the robot.

# 1.4 Recent advances: towards perfect followers?

As mentioned in the beginning of this chapter, most implementations of controllers on robotic systems in order to jointly manipulate objects with a human operator consider the robot as a follower, and leave the responsibility of leading the task to the human operator. We have briefly introduced in the previous section (1.3.2) some advanced controllers where the parameters vary with time, or depending on the human intention. We present some of these schemes in more detail, to highlight the fact that the performance of robotic partners has generally been improved by considering active behaviors rather than purely passive ones.

## 1.4.1 Human impedance models

More work has been done regarding temporal variations of the impedance of the robot, with different concerns: either relieving the human operator from workload, or having human-like dynamics. Regarding the first concern, an early work has been performed by Tsumugiwa *et al.* regarding variable impedance depending on the task phase [110]. We hereby focus on the second concern, with the works of Ikeura *et al.* and Rahman *et al.* [48, 86, 87]. In their work, these authors first proposed a model for the distribution of the forces applied by the partners in a collaborative task:

$$\begin{cases} f_1 = \alpha f + f_{int} \\ f_2 = (1 - \alpha) f - f_{int} \\ f = M\ddot{x} \end{cases}$$
(1.7)

According to this model, the partners each apply part of the force f necessary to move the object, plus an internal force  $f_{int}$  such that the internal force components of each partner sum out to zero. As we will see in the next section, this model introduces the idea that *both partners can contribute to the task*, but this model is only descriptive, since it applies to the output variable of the partners (considering the partners as systems), *i.e.* the force they apply on the object. This model cannot apply directly to design a decentralized control strategy, because the same variable  $\alpha$  is used for both partners. However, it has been used to study the haptic dominance in human-human collaborative tasks [35].

In their work, the authors assume that only one partner is doing most of the job, while the other is following. In other words, they assume that either  $\alpha = 0$  or  $\alpha = 1$ . Then, they identify  $f_{int}$  as the output of the impedance characteristics of the follower arm. They model this impedance by a second order, linear impedance with time-varying parameters and identify the gains for a human subject. Then, they embed the identified impedance in a robotic system to replay the task, and their results show a trajectory that is close to the minimum-jerk trajectory.

One of the most noticed and reused results [16,90] is that the gains they identify are so that the damping term is dominating the dynamics of the follower arm. The model used in their work is not easy to interpret though. Because they used the starting position as a nominal position for their spring and damper, they identify a *negative* stiffness coefficient, which corresponds to a destabilizing force field. Of course, the amplitude of the stiffness can then only decrease along the task, else the corresponding behavior would be to move further and further away from the starting position, with increasing velocities. An interpretation given by the authors is that the arm was in an unsteady position because the subject did not know the direction of the motion in advance.

By slightly changing the model, though, a positive stiffness could be identified, with another, sensible interpretation. The impedance model proposed by the authors was:

$$M\ddot{x} + B\dot{x} + Kx = f \tag{1.8}$$

where f is the force applied on the human hand, x is the position of the hand, and M, B and K are impedance parameters to identify. Let us replace the last term so as to obtain the following model:

$$M\ddot{x} + B\dot{x} + K_0 (x - x_0) = f$$
(1.9)

where  $K_0$  is a stiffness, and  $x_0$  is the nominal position of the arm impedance and is this time chosen to be the *target destination*, instead of the starting position. Then, keeping the same values for K, B and M as the ones identified by the authors, we have the following relationship between the stiffness K identified by the authors and  $K_0$ :

$$K_0 = K \frac{x}{x - x_0} \tag{1.10}$$

Assuming that K is such that  $K x/(x-x_0)$  tends towards 0 when x tends towards  $x_0^{-1}$ , we then can obtain a *positive* stiffness  $K_0$  which is 0 at the beginning and the end of the motion, and is maximal in the middle of the motion. This can be interpreted as follows: at the beginning of the motion, the human subject does not know where the leader robot will go. Hence, the arm stiffness is adjusted to zero, so that the arm does not restore any energy to produce motion. In the middle of the motion, we can suppose that the human is *able to predict, to some extend, the target position*. Hence, he slightly helps the motion by increasing the arm stiffness. At the end of the motion, the desired location being only imprecisely estimated by the human, the arm stiffness is decreased again so as not to perturb the final positioning phase.

This interpretation is highly speculative, of course, since we don't know how the human would estimate the target location. It has the advantage, however, of producing a positive stiffness parameter that is easier to interpret. Moreover, the corresponding interpretation is consistent with active following schemes such as in [16, 70], that we will examine later.

#### 1.4.2 Active following

A more recent work by Duchaine *et al.* [20] considered the time derivative of the force applied by the human. Their idea is that derivating a signal is doing prediction. Hence, the time derivative of the force applied by the human partner on his robot helper gives a clue on the human's intentions. More to the point, they applied this idea to tune the damping parameter of an impedance controller, so as to increase it when the human operator wants to decelerate and decrease it when the human operator wants to accelerate. The resulting dynamics is given by:

$$\mathbf{f} = \mathbf{M}\ddot{\mathbf{x}} + \left(\mathbf{B} - \alpha\dot{\mathbf{f}}\right)\dot{\mathbf{x}} \tag{1.11}$$

<sup>&</sup>lt;sup>1</sup>This is the case for example if *K* is of the shape  $e^{\frac{x^{\alpha}}{x-x_0}}$ 

where **x** is the Cartesian position of the end-effector, **f** is the force applied by the human operator, **M** is a virtual mass parameter, **B** is a damping parameter, and  $\alpha$  is a weighting factor.

An interesting point is that for high values of  $\alpha$  and  $\hat{\mathbf{f}}$ , the overall damping of the system, *i.e.*  $\mathbf{B} - \alpha \dot{\mathbf{f}}$ , can become negative. In this case, the robot no longer acts passively. Since this will happen only based on the actions of the human operator, and since the robot has no desired task plan, the robot is still following the human, but instead of only dissipating energy, the robot can sometimes *actively* follow the human operator.

In this control law, active following is a consequence of the controller design, but was not specified as a goal for the controller. The key point is rather to use the time derivative of the force as a predictor of the intentions of the human operator.

In some works [16], however, it is explicitly argued that passive following generally merely helps the human, since it dissipates part of the energy introduced by the human into the system, so that only part of it is transformed into kinetic energy. To increase the performance of robotic assistants, some researchers proposed model-based active following schemes [16, 70]. The idea is the following: if the helper robot is able to predict the desired trajectory of the human operator, then it can actively track it rather than just passively react on the external forces.

A common model for point-to-point human motions is the minimum jerk model [31]. According to this model, the trajectory followed by the human hand for point-to-point tasks is a straight line, and the position along this line is given by:

$$x(t) = (x_f - x_i)(6\tau^5 - 15\tau^4 + 10\tau^3)$$
  

$$\tau = \frac{t}{T}$$
(1.12)

where  $x_i$  and  $x_f$  are the initial and final positions, *T* is the duration of the motion and *t* is the time.

In [70], the parameters  $x_f$  and T are estimated using a least squares method. The control law implemented on the robot was:

$$f = M\ddot{x} + B\dot{x} + K(x - \hat{x}) \tag{1.13}$$

where  $\hat{x}$  was the position obtained from the minimum jerk model with estimated parameters  $x_f$  and T. The stiffness parameter K was varying along the task: because little data is available at the beginning of the task, the starting value for Kwas set to zero, and increased progressively to its maximal value. At the end of the task, in order not to hinder precise positioning by the human operator (because of an imprecise evaluation of the end position), the stiffness was again decreased to zero. This algorithm is then consistent with the reinterpretation we gave of the human impedance identification in the previous paragraph (1.4.1). Finally, the authors also show that the unnecessary energy transfer can be reduced when the robot actively follows the human using their proposed coordination scheme.

The idea of identifying the parameters of the minimum jerk model was also used in [16]. However, in [16], the final position was fixed and only the duration of the motion was estimated. From the estimated duration of the motion, the desired velocity of the human was computed, and used to reduce the apparent damping of a pure damping controller. The main idea in this work is that this active assistance can be scaled by a weighting factor  $\alpha$ . For  $\alpha = 0$ , the robot was acting passively. The higher  $\alpha$  is, the less effort the human operator has to produce to perform his desired motion. The value  $\alpha = 1$  is impossible in practice since the robot needs some input from the human operator to estimate her/his desired trajectory.

Active following, especially in the case of [16], is a good illustration of the fact that even if the robot is following the human intentions, this does not mean that the human operator does not adapt to the robot. Assuming a given desired time for a point-to-point task in the context of [16], for different levels of assistance (for different values of the scaling parameter  $\alpha$ ), the human operator will have to apply different forces to produce the same desired motion.

Hence, we state again that *the leader/follower model does not exclude mutual adaptation*. We also see from the works presented in this paragraph that *the leader and follower concepts are decoupled from the concepts of active and passive behaviors*: this paragraph has presented three different active following schemes, which shows that a follower does not need to be passive.

# **1.4.3** Proactive behaviors

In the previous paragraph, we have presented works were the robot was *actively* following the human operator, based on an estimation of his intentions. These works did not consider ambiguities, and estimations errors are implicitly handled by the compliance of the controller, which limits the effect of discrepancies between the real and estimated intentions of the human operator.

If we consider that there can be ambiguities in the information provided by the human operator, so that the robot "hesitates" between two actions to take, what should the robot do?

In such a situation, one can either wait or act passively, or one can be *proactive* and take an initiative which will hopefully lead to a disambiguation of the partner's intention. A cognitive architecture has been proposed for proactive human-robot cooperation, based on Dynamic Bayesian Networks [97]. In their paper, Schrempf *et al.* propose a Bayesian approach to intention recognition based on system knowledge acquired by hard-coding or programming by demonstration, and the concept of proactive behavior. Proactive behavior is relevant when ambiguities arise regarding the human partner's intentions. When the intention of the partner is ambiguous and a (small) number of possible actions are possible from the robot, the robot chooses the one that optimally reduces ambiguity about the human intention and starts executing it, and possibly rolls back if it appears that the wrong action has been taken.

#### **1.4.4** Leader robots?

Up to now, all the works we have presented considered follower robots. Even when the robot takes initiatives when the human intentions are unclear, the robot always tries to comply to the intentions of the human. Let us state again that this does not mean that the human operator does not have to adapt to the robotic partner. Typically, if the follower robot implements a passive behavior with a virtual nonholonomic constraint [3], the human operator will adapt to this constraint when performing collaborative tasks with the robot. More recently, the use of servo brakes has been proposed to implement obstacle avoidance for assistive robots. Using this technology, the robot can increasingly resist to the human operator when moving towards obstacles [41]. As a result, the human operator can adapt his trajectory to avoid the obstacle. Again, this highlights adaptation from the human operator to a follower robot. The same technology can also be used to track a desired trajectory. Of course, since the robot is passive (it only uses servo brakes), the human operator has to inject energy into the system, for example by pushing the robot. The robot can then use the servo brakes to track a reference trajectory. In such a configuration, the robot imposes a trajectory to the human, and the human operator will be responsible for the motion *along* the trajectory. Such a situation is encompassed by the model we present in the next chapter. We will also present other works where the robot is not acting as a pure follower.

# 1.4.5 Conclusion

In this section, we have studied different recent models that have been proposed either to describe or to implement collaborative behaviors on a robotic system. The first model we study (paragraph 1.4.1) decomposes the force applied by each partner into a component that generates motion of the object and an internal force. The proposed model is descriptive, and has been used in the case where only one partner (the leader) generates motion of the object, and the other partner (the follower) only applies disturbing forces. The force applied by the follower has been fit by a second order linear impedance model with time varying parameters. By reinterpreting this result, we highlight a behavior that is *consistent with active following algorithms* presented in this section. We conclude that active following is a realistic behavior for a human partner, which further validates the approaches presented in [16, 70] and recalled in this section. Finally, this section shows that robotic partners have been recently given more and more active roles and decision capabilities, sometimes even acting as a leader.

# 1.5 Conclusion

This thesis focuses on dyadic collaborative manipulation tasks, where two partners apply forces on a same object to impose a desired motion, or to bring the object into a desired configuration in space. This chapter has introduced characteristic problems of such tasks:

- dynamic interaction: how the partner can deal with each other's dynamics and constraints, and handle the differences in their intentions in a stable manner. This problem is partially solved by using impedance control approaches, which limit the energy flow between the partners given different desired trajectories for the partners. However, for some specific platforms, such as humanoid robots, which can lose balance, impedance control is not sufficient to have a safe physical interaction with a human operator. Moreover, even if the energy flow is regulated, impedance control will not guarantee that a task can be realized if the partners have different intentions.
- load sharing: how the partners can resolve redundancy when manipulating the object. In general, an infinite set of force vectors acting on an object can lead to the same motion, because internal forces on the object will not influence its motion. How can the partners find strategies to solve redundancy efficiently? The specialization phenomenon highlighted by Kyle Reed and presented in the first section of this chapter can be interpreted as an answer to this problem for a specific class of tasks. However, we do not have a gen-

eral framework to represent load sharing strategies among the partners. *The model presented in the next chapter is an attempt to fill this gap.* 

▷ *planning*: how can two partners *negociate* a common task plan, and to what extent is it possible to do so through the haptic channel?

To solve this problem, *mutual adaptation processes* must take place, which in turns requires to be able to understand the behavior and to some extent *predict the intentions* of the other partner. Mutual adaptation at the dynamic behavior level can be somehow implemented on a robotic system using impedance control. Impedance control can be seen as a way to accommodate to the human partner by limiting the force response to positioning errors. In other terms, impedance control allows a robot to have a compliant behavior, which is the reason why it has been used in most works in the field of pHRI. Its fundamentals and history have been presented in this chapter, to justify its importance in the context of haptic collaborative tasks.

Impedance control has been used in such a way that the robot implements a *follower behavior*, letting the human operator take the responsibility of imposing a task plan, *i.e. leading* the task. This is a way to solve at the same time the load sharing problem and the planning problem. The robot complies to the plan imposed by the human partner, generally supports the whole weight of the manipulated object, and the human simply adapts to the dynamic constraints imposed by the robotic system and its controller. While early robotic followers adopted a passive behavior, recent works proposed active following schemes, and a trend in the last years consisted in giving more and more responsibility to the robot to achieve a task. This trend has been highlighted by presenting some recent models in detail.

Finally, despite the trend in giving more and more initiative to the robotic partner, only a very few works considered assigning a leader behavior to robotic systems. In the next chapter, we devise an abstract framework to consider an equal attribution of the responsibility to perform the task to both the human and the robotic partners. This framework assumes that the robot can take initiatives of its own, and adopt active behaviors that do not solely depend on the human partner's intentions. We also formulate the hypothesis that the leader-follower model can be extended to a *continuous, time-varying role distribution* and propose a model built upon this hypothesis in order to encompass most behaviors that can be encountered in haptic collaborative tasks.

# Physical collaborative tasks: a 2 homotopy-based interaction model

Content	S		
2.1	A homotopy switching model for physical Human-Robot Interaction		31
	2.1.1	The leader-follower model	31
	2.1.2	Intuition	31
	2.1.3	Mathematical formulation	32
	2.1.4	Example: linear homotopy	33
	2.1.5	Genericity of the proposed model	34
	2.1.6	Challenges for the application to pHRI: reasons for a haptic lan-	
		guage	35
	2.1.7	Relation to previous works	36
	2.1.8	Behavioral architecture	37
2.2	A concrete example		38
	2.2.1	Setup	39
	2.2.2	Results	40
	2.2.3	Extension to a Person-Object-Avatar scenario	41
2.3	Defining the homotopy		44
	2.3.1	Enforcing constraints: self-collision avoidance	45
	2.3.2	parameterizing phenomena observed in human-human collabo- ration	47
	2.3.3	Conclusion	52
2.4	Conclusion		52

IN the previous chapter, we described existing approaches to physical Human-Robot Interaction. A recurrent assumption in these works is to appoint a unilateral distribution of the roles to the partners that will not change all along the task execution.

In this chapter, we go beyond the fixed leader-follower distribution archetype. Our main contribution is a model based on homotopy switching between intrinsically distinct controllers, to allow *continuous, time-varying role distribution* among the partners. The proposed model is designed to encompass most behaviors encountered in dyadic haptic collaborative tasks through an intermediate object. The basic idea is to switch continuously between two distinct extreme behaviors (leader and follower) for each individual. The physical collaborative interaction is then described only with two distinct homotopy time-functions that vary independently. We also come up with the idea that these functions can describe the signature of a collaborative task.

This chapter introduces the model through insights and a mathematical basis. We then use the model in three different applications: i) modelling and ii) simulating behaviors that were observed in human-human collaborative tasks, and iii) collision avoidance. We exemplify the use of this model in a collaborative task between a human and a virtual avatar using *a priori* homotopies. The collision avoidance application is also demonstrated on a real humanoid platform.

# 2.1 A homotopy switching model for physical Human-Robot Interaction

In this section, we examine a new model for dyadic tasks that encompasses most behavioral situations that are encountered in pHRI. This model is based on a timevarying weighting between a leader and a follower behavior. This model is based on the intuition that when two *human* partners perform a task, each one is likely to have his own task plan, and therefore, it is not possible to know *a priori* which partner will be the leader and which will be the follower.

# 2.1.1 The leader-follower model

Physical collaborative tasks raise the problem of motion coordination. When two partners jointly manipulate an object, they have potentially two different desired trajectories for the object, but in the end, the object will have only one trajectory. To avoid too large interaction forces between the partners through the object, both partners must be compliant when trying to impose a motion to the object. As we have shown in the previous chapter, this is a reason for implementing impedance controllers on robotic systems to cooperate with human partners.

This compliance, however, will only absorb reasonable differences in the partners' intentions. In an extreme case, if both partners intend to move to opposite directions, compliance will avoid a too sudden increase of the interaction forces, but the task will fail anyway as the object will not move. While we expect that two human partners solve such situations "naturally", we do not know exactly *how* humans negotiate a task when they have incompatible intentions. Therefore, handling conflicting situations in human-robot collaborative tasks can be more problematic.

Such situations will not occur, however, if one of the partners has *no* intention but to follow the other partner. The task is then conducted like a dance, where one partners leads the other [32]. This configuration defines the *leader-follower* model, where one partner has knowledge about the task and takes initiatives, while the other attempts to follow the leader as closely as possible.

This model has been extensively used in robotics to implement human-objectrobot collaborative tasks. It is certainly easier to affix a robot to behave in either one of these cases. This is made simply by programming the appropriate controller. However, it is nearly impossible to force the human partner to behave in an exclusively passive role. Therefore, in robotics, this problem has recurrently been solved by programming robots to be followers and assigning the human operator the leadership in the task.

## 2.1.2 Intuition

In human-human interaction, both partners have intelligence and knowledge about the task to perform. Especially in tasks were the roles of the participants are symmetric, it is thus impossible, *a priori*, to assign a leader role to one partner and a follower role to the other partner. If such roles can be defined, their attribution must be negotiated "on-line" by the partners.

In the previous chapter, we mentioned the problem of load sharing. Given a task, what will be the contributions of the partners, and how will they organize themselves to act complementarily? A solution can be to decompose the task into subtasks and subsets of constraints. Complementary roles can then be negotiated

between the partners to be the leader for some subtasks and act as a follower for the other ones. Each partner is therefore *leading and following at the same time*.

Another point to highlight is that the distribution of the leadership for the different subtasks does not need to be affixed. Depending on the context, on the amount of information available, each partner may choose to give up or claim for the leadership of a subtask. This can especially occur if one partner is close to violate one of his own constraints that are not directly related to the collaborative task. For example, a robot which follows a human being may come close to a joint limit or singularity which will prevent it from behaving as the human partner intends to. A biped robot can be about to lose dynamic balance. Before it happens, the robot may claim for leadership and handle the subtask in a way that will keep it far from its constraints (of course this requires that the task can be achieved in several ways, *i.e.* that the task offers some degree of redundancy).

This role switching should occur in a smooth way, so that i) the human partner has time to react and negotiate progressively the role sharing and ii) the motion of the robot is not abrupt and jerky. Moreover, a smooth transition between the leader and follower roles, and its timing, is necessary to translate *progressive ne-gotiation* and *hesitation*. When switching abruptly between these states, the only way to translate hesitation is to oscillate from one state to the other while trying to decide what to do. On the contrary, if the switching is smooth, the role redistribution and sharing is progressive. This allows each partner to have knowledge and understanding on what the collaborative partners' intents are.

Finally, by allowing each partner to adopt a behavior "between" the leader and follower ones, we introduce the idea of mutual adaptation: each partner can try to find a compromise between her/his own intentions, and what he perceives from the other partner's intentions.

The remainder of this section more formally presents a model that accounts for these intuitions.

# 2.1.3 Mathematical formulation

**Definition 2.1.1** A homotopy between two continuous function maps  $f : X \to Y$  and  $g : X \to Y$ , where X and Y are topological spaces, is defined to be a continuous function map  $h : X \times [0,1] \to Y$  such that:

$$h(x,0) = f(x)$$
 and  $h(x,1) = g(x)$  (2.1)

The function h describes a "continuous deformation" of f into g. If the homotopy parameter (the second parameter of h) is 0, we have the function f, and if it is 1, we have the function g.

We then propose to extend the leader-follower model as follows:

In dyadic, physical collaborative tasks, the behavior of each partner *i* can be characterized by:

- ▷ a homotopy  $h_i: X_i \times [0, 1] \to Y_i$  between a map  $f_i: X_i \to Y_i$  characterizing a *leader* behavior, and a map  $g_i: X_i \to Y_i$  characterizing a *follower* behavior;
- ▷ a continuous function of time,  $\alpha_i : \mathbb{R}^+ \to [0, 1]$ .

At time t, the behavior of each partner is thus characterized by the function

$$x \in X_i \to h_i(x, \alpha_i(t)) \tag{2.2}$$

The model does not define the spaces  $X_i$  and  $Y_i$ , nor the functions  $\alpha_i$ ,  $f_i$  and  $g_i$ . They are just supposed to be a characterization of a leader and a follower behaviors. An analytical definition of these elements is actually *not necessary* to translate the idea behind this model: each partner will be "between" the leader and follower modes, and that his "weighting" between these two modes will vary continuously with time.

We need, however, to make clear what a leader behavior and a follower behavior are. By saying a partner has a leader behavior, we understand that the partner has a task plan he tries to follow, and that this task plan is not computed to match the intentions of the other partner. This does not mean that the plan can not be adapted in response to actions taken by the other partner, however, these adaptations will be local, and the overall task plan will be decided according to the partner's own intentions.

On the contrary, a follower will never have any task plan, and will only act based on the intentions of the human, or simply comply to the external forces applied by the other partner. Note that a partner that resists more and more to the forces applied by the other partner to avoid an obstacle, as in [41], can not be considered as a leader, even if it somehow "refuses" to follow. Indeed, avoiding an obstacle is not imposing a task plan to the human partner. More generally, enforcing unilateral constraint does not correspond to our idea of a leader behavior, since a leader would try to impose *one* specific trajectory or task plan to the other partner.

We will see in the remainder of this chapter that the homotopy need not be defined explicitly at the lowest level of the control space (*i.e.* the torques on torque control robots, reference positions on position controlled robots and so on). It is also possible to define specific leader and follower behaviors and the homotopy at a higher control level. The homotopy then extends implicitly to the control space, assuming continuous control laws.

## 2.1.4 Example: linear homotopy

We now propose a typical application of this model using a linear homotopy between two controllers. Let  $(\mathcal{U}_i)_{i=1,2}$  be the controllers used by partners 1 and 2 to realize a physical collaborative task. We propose to define  $(\mathcal{U}_i)_{i=1,2}$  as the image of a real number  $\alpha_i$  by a linear homotopy between a controller  $\mathcal{F}_i$  that implements a follower behavior and a controller  $\mathcal{L}_i$  that implements a leader behavior, that is:

$$\mathscr{U}_i = \alpha_i \mathscr{L}_i + (1 - \alpha_i) \mathscr{F}_i \tag{2.3}$$

Both controllers  $\mathscr{L}_i$  and  $\mathscr{F}_i$  are seen as functions that produce an output command (a joint torque, joint position, or joint velocity vector) and are defined on a common space where the state of the collaborative task can be described. Typically,  $\mathscr{L}_i$  can be a position controller, and  $\mathscr{F}_i$  can be an impedance controller, though other schemes can be imagined, as we will see later. The controller  $\mathscr{U}_i$  will compute the outputs of the controllers  $\mathscr{L}_i$  and  $\mathscr{F}_i$ , and produce a weighted sum of these outputs.

This model allows the description of several behaviors that can be expected during physical tasks using only two time-varying independent mappings,  $\alpha_1$  and  $\alpha_2$ , that take values between 0 and 1, as shown on figure 2.1. These mappings are continuous functions of time and can have different values on each dimension

of the control space. Various dynamics can be given to the controllers  $\mathcal{L}_i$  in response to tracking errors, but the idea is to remain stiff enough to impose a desired trajectory.  $\mathcal{F}_i$ , on the other hand, can be set as a compliant controller to implement a passive follower role, or an active one, as proposed in [16]. Even if  $\mathcal{F}_i$  can implement active following, it should be as compliant as possible in case of misunderstanding of the leader's intentions.



Figure 2.1 – Illustration of one degree of freedom homotopy for each individual of the physical interaction task dyad (holding a table). Each  $\alpha_{i \in \{1,2\}}$  may evolve independently from the other and their time function results on a dynamic sliding between 0 and 1 during task execution.

Considering only one dimension, if  $\alpha_1 = \alpha_2 = 0$ , both partners will wait for each other since they both behave as pure followers, and nothing will happen. If  $\alpha_1 = 1$  and  $\alpha_2 = 0$ , then the task plan of the first partner is accepted by partner 2. If  $\alpha_1 = \alpha_2 = 1$ , then if both partner have the same task plan, they will follow it. Else, they will conflict and each of them will try to stick to his/her plan, which can result in unwanted behavior, such as high interaction forces, task constraint violation or inability to achieve the task. For example, think of two partners who hold a table and want to walk around an obstacle, but each partner wants to go on a different side, if each partner sticks to his/her plan and have comparable strength, they will just run into the obstacle or will have to stop, while significant forces will be applied on the table as the partners try to pull it to different sides.

## 2.1.5 Genericity of the proposed model

Our model is an abstract model and can be implemented in various ways, using very sophisticated controllers for  $\mathcal{L}_i$  and  $\mathcal{F}_i$ . A potential implementation for the leader controller would be the one proposed in [16], where the follower robot can be a more or less active follower. Any level of sophistication can be chosen for the implementation, including implementation based on stack of tasks [102] or operational space formulation [60].

An important point is that the homotopy does not need to be defined at the lowest control level. Actually, in most of the examples presented in this chapter, the homotopy will be defined at the joint position level: the output of the controller is a joint position vector that is sent to an inner position control loop. However, by defining a homotopy at the desired joint position level, we also *implicitly define a homotopy at the actuator torque level*, assuming that the inner position control loop computes reference torques, as long as the function that links the state of the robot and the desired joint position vector to the output torques is continuous.

We can also choose to define the homotopy at the Cartesian trajectory level,

and to use the output trajectory as a virtual position input in an impedance controller [44]. Let M, B and K be a virtual mass, damping and stiffness used to implement an impedance controller on a robotic partner. The controller takes a desired trajectory  $x^0$  as an input, and imposes the following dynamic behavior to the end-effector of the robot:

$$f = M\ddot{x} + B\dot{x} + Kv$$
  

$$v = x - x^{0}$$
(2.4)

where *f* is the force applied on the end-effector of the robot. This equation can then be used to compute a reference force or a reference position for a lower-level controller. We then can define the input trajectory  $x^0$  as the image of an homotopy between a leader and a follower behavior:

$$x^{0}(y,t) = \alpha(y,t)\mathcal{L}(y,t) + (1 - \alpha(y,t))\mathcal{F}(y,t)$$
(2.5)

where t is the time, y is a state vector and  $\alpha$  is the homotopy parameter. The functions  $\mathscr{L}$  and  $\mathscr{F}$  correspond to trajectories that are characteristic of leader and follower behaviors: to give an extreme example, the leader trajectory will be state independent, while the follower trajectory will be computed by estimating the desired trajectory of the human partner, like in [16,70].

The dynamic behavior (i.e. how the robot reacts to disturbances while tracking its own desired trajectories) are then defined by setting the gains of the low level impedance controller. These gains can be set according to the context of task execution, on stability criteria, or even be a function of the homotopy variable, so that the final controller is a homotopy between two impedance controllers. This is richer than interpolating the gains between follower gains and leader gains, as the leader-mode impedance controller is not constrained to have fixed gains. Hence, it is possible to have a "soft" leader, when the other partner is gently following, or a "stiff" follower <sup>1</sup>, when the follower wants to limit the motion speed, without necessarily having a specific desired trajectory (to have a flavor of what acting as a "stiff" follower is, try to let someone guide you by pulling your hand in a cluttered environment, while closing your eyes).

## 2.1.6 Challenges for the application to pHRI: reasons for a haptic language

Our idea is to use the proposed model to implement controllers on a robotic platform to perform physical collaborative tasks with a human partner. Our model allows the robot to behave as a pure leader, or as a pure follower, or to adopt any intermediate behavior. *By tuning the homotopy parameter, the behavior of the robot can be shaped* between the extreme leader and follower roles.

The main question is now: according to what criteria should this parameter be defined? In simple words, when should the robot behave as a leader, or as a follower? How to shape its behavior appropriately?

A possible solution is to assume that this model can describe the human behavior, and to try to *identify the homotopy parameter of the human partner*. The robot can then use a complementary behavior. However, the identification of this parameter, if the model is valid (a preliminary assessment is presented in the next chapter), is likely to be difficult. Indeed, as we mentioned in the previous chapter,

<sup>&</sup>lt;sup>1</sup>Here, the work 'stiff" does not refer to a mechanical stiffness, but rather to a very dissipative behavior.

a follower does not need to be passive, and can even be proactive. However, if the human partner is proactive, he can misunderstand the intention of the robot, which will generate interaction forces. But from the point of view of the robotic system, it will be impossible to know whether these forces arise from an *intention of the human partner to lead*, or from a *misunderstanding* of the intentions of the robotic partner.

In such an ambiguous situation, looking locally at the interaction forces will not allow to analyze the situation. For the robotic system to guess whether the human is trying to lead or "failing to follow", a form of communication must probably take place, where both partners will hesitate and wait and see what happens, and so on. For instance, in the situation we just described, we can imagine that the robot will try to act as a follower, and see that nothing happens, so it might try to lead again. But if meanwhile the human partner has decided that because the robot stopped, he should take the lead, conflicts will arise, which will lead to interaction forces, and so on. We see here the premises of a *haptic dialog* between the partners to decide who leads and who follows. This hypothetic dialog can be an exciting research topic, but it goes beyond the scope of the work presented in this thesis.

Finally, assuming stable leader and follower controllers, there is no guarantee that the controller obtained from the homotopy between these two controllers will also be stable [78].

# 2.1.7 Relation to previous works

In most previous works in pHRI, the leader's role is assigned to the human operator, and a follower's role to all other agents. Some researchers [86], however, suggested that the contribution to the motion of the manipulated object can be distributed between partners, but studied the case where a robot follower collaborates with a human leader. The model used in [86] has been detailed in the previous chapter. It states that each partner applies a part of the force necessary to produce the actual motion of the object, and an internal force. When summing the forces applied by each partner, the internal forces cancel out, leaving only the resultant force. Let us recall the equations:

$$\begin{cases} f_1 = \alpha f + f_{int} \\ f_2 = (1 - \alpha) f - f_{int} \\ f = M\ddot{x} \end{cases}$$
(2.6)

where  $f_i$  is the force applied by partner *i*,  $f_{int}$  is an internal force, *m* is the mass of the manipulated object, *x* is its position, and  $\alpha \in [0; 1]$  is a weighting factor. The signification of this weighting factor is close to the signification of the homotopy parameter used in our model: indeed, when  $\alpha = 1$ , partner 1 only is responsible for the motion of the object, and conversely when  $\alpha = 0$ , the motion of the object is only cause by partner 2. Hence, this parameter could be seen as a description of a continuous role distribution. However, looking more closely, we see that the parameter  $\alpha$  used in this model has not the same meaning as our homotopy parameter. The parameter used in eq. (2.6) describes the *actual* role distribution, in terms of which participant has the "higher responsibility" in producing the motion of the object. Note that the parameter is the same in both equations, which implies a complementarity in the role distribution. Hence, this model allows to know who was *de facto* the "leader" of the task, whereas our model translates an *intention* to act as a leader. In our model, *both partners can try to act as leader*, hence we have two independent parameters, one for each partner, and no relation is imposed *a priori* between those parameters. This is essential to use the model to implement behaviors on a robotic system, because we can not impose a role to the human partner. The actual value of the parameter  $\alpha$  in eq. (2.6) can only be known if both  $f_1$  and  $f_2$ are known. However, these forces are supposed to be the *output* of the controllers.

Note also that the parameter  $\alpha$  used in eq. (2.6) is not necessarily related to the homotopy parameters of each partner, assuming our model describes their behavior. Actually, who will be the leader *de facto* depends not only on the homotopy parameter of each partner, but also on their physical abilities: both partners can try to act as a leader, but in the end, the stronger partner will be leading the task in the sense of eq. (2.6). However, one can try to obtain a given dominance distribution among the partners by setting appropriate values for the homotopy parameters. In other terms, a robotic partner will have more chance to dominate a human partner if it is always trying to act as a leader.

The model used in [86] has been used in [35] to derive a measure of dominance in collaborative task. This measure is used to analyze human-human collaborative tasks where both partners share the same intention (the task consist in following a given track, and both partners have the same feedback about the task). One of the results presented in [35] is that the dominance can be separated into two components: one being intrinsic to the partners, and one coming from the interaction between the partners. The authors suggest that to obtain a natural behavior from a robotic partner, both components should be taken into account. In the current state of their work, only physical aspects of collaborative tasks have been studied, and in the context of one specific task. Hence, the results they obtain might not be applicable to other tasks. However, such studies about human-human interaction are interesting since they can provide strategies to define the homotopy parameter of a robotic system controlled using our behavior weighting model.

Other works have proposed to assign a leader role to robotic partners, and to switch during the task. Role switching and defining passive and active roles has been evoked in [116], and successfully applied to human-robot hand-shaking in [115]. Our contribution is to systematize this concept and propose a framework that allows us to abstract from the implementation of the underlying controllers, to focus only on the role distribution. Moreover, we consider *continuous* role distributions: we hypothesize that the behavior of an individual in a collaborative task can be *in between* the extreme leader and follower roles. Note that the work presented in [115] is interesting, since the robot will switch between the leader and follower roles based on the identified role of the human. This gives us hope that the role of the human partner can be estimated to some extent, which might be useful to adjust the homotopy parameter of the robotic partner.

# 2.1.8 Behavioral architecture

Our goal is to enhance robotic or virtual avatars with physical interaction cognition so that they become more human-like partners in physical collaborative tasks, see Figure 2.2. Our newly proposed approach to physical interaction provides a model on which a behavioral architecture could be built.

The task represented on figure 2.2 consists in bringing a wooden board from an initial position to a final position. The solid and dashed lines represent the respective plans of the human and the robotic partner. The planned trajectory of



Figure 2.2 – Person-Object-Avatar physical collaborative task architecture and main modules. The solid and dashed lines represent the a priori plans of the partners. The red point represents the starting position of the object and the yellow one represents the target destination. Two data flows can be seen on the figure. The flow that goes through the light gray boxes represents data that is closely related to the communication with the human partner. The dark gray blocks are more related to the internal state and constraints of the robot.

the robot can have been computed by a planner and presented to the human partner to agree on a rough plan before starting the task. However, the final plan will be negotiated during the task.

The extreme leader and follower behaviors take part of two different data flows. The leader controller will mainly rely on the internal state of the robot with regard to its constraints and its *a priori* plan. The task and robot state can come from the perception of the robot through the interpretation of measured signals. Basically, the leader behavior will especially involve the robot's own state or internal data (dark blocks on figure 2.2).

The follower behavior, on the other hand, will call for perceptual capabilities and estimation of the human partner's intentions. The information gathered from this interpretation of the human partner's actions will be used as an input by the follower controller. The leader behavior, in general, will involve communication with the human partner (light blocks on figure 2.2).

The leader and follower behaviors will be blended by the homotopy weighting module, which will rely on predefined task signatures, on the state of the robot, or on the perception of the human intentions, as will be shown in section 2.3. This module is not yet clearly defined, and will certainly benefit from studies of human-human interaction such as the ones conducted in [29, 35, 90].

Finally, the haptic patterns communication module is simply a hypothetical module which generates distinct haptic patterns (signals) on top of or blended with the control signal of the avatar: its main mission is to try to communicate avatar intention to the human operator through the haptic channel. We don't know yet whether such a module is relevant, but this module would implement haptic communication capabilities to face such situations as described in paragraph 2.1.6.

# 2.2 A concrete example

We now propose a concrete implementation of the model to exemplify its use. We will apply it to an Avatar-Object-Avatar scenario: two virtual humanoid robots

manipulate an object to move it over an obstacle. We will define the behavior of each robot using the homotopy controller.

# 2.2.1 Setup

The task to be performed is illustrated on figure 2.3: two virtual avatars have to lift and pass an object over an obstacle.



Figure 2.3 – Collaborative task between two virtual avatars.

For the leader behavior, we implement a Cartesian position controller. The follower behavior will be a damping controller (*i.e.* a static relation between the external force and the velocity is specified). Both controllers will rely on a pseudo-inverse of the Jacobian matrix that links joint displacements to Cartesian displacements of the robot's gripper, and will compute an output desired joint velocity. From the desired trajectory of the object, a desired trajectory  $\mathbf{p}_i^{\mathbf{d}}(t)$  is computed as an input to the gripper position controller for each robot  $i \in \{1,2\}$ . We define  $\mathbf{e}_i = \mathbf{p}_i^{\mathbf{d}} - \mathbf{p}_i$  as the tracking error, where  $\mathbf{p}_i$  is the actual position of the gripper of the virtual robot <sup>2</sup>. We denote  $\mathbf{f}_i$  the interaction force sensed at the wrist of robot *i*. The leader and follower controllers  $\mathcal{L}_i$  and  $\mathcal{F}_i$  are defined as follows:

$$\mathcal{L}_{i}: \dot{\mathbf{q}}_{i}^{\mathcal{L}} = \alpha_{i}\lambda_{p} \mathbf{J}_{i}^{*}\mathbf{e}_{i}$$
  
$$\mathcal{F}_{i}: \dot{\mathbf{q}}_{i}^{\mathcal{F}} = (1 - \alpha_{i})\lambda_{f} \mathbf{J}_{i}^{*}\mathbf{f}_{i}$$
(2.7)

where  $J_i^{\#}$  is the pseudo-inverse of the Jacobian,  $\lambda_p$ ,  $\lambda_f$  are given controller gains. The leader controller is a position controller: the leader behavior will consist in tracking a desired, pre-programmed trajectory. The follower controller is a damping controller, so that when the virtual avatar acts as a follower, the motion of its gripper complies with the forces applied by the human operator.

The controller  $\mathcal{U}_i$  is then defined by:

$$\mathscr{U}_{i}:\dot{\mathbf{q}}_{i}^{\mathbf{d}}=\dot{\mathbf{q}}_{i}^{\mathscr{L}}+\dot{\mathbf{q}}_{i}^{\mathscr{F}}$$
(2.8)

where  $\dot{q}_i^d$  is integrated and sent to an inner joint position controller.

 $<sup>^{2}</sup>$ Unless the contrary is stated, positions will refer to positions and orientations, and forces will refer to forces and torques.

Both virtual partners have different desired trajectories: though the starting and ending position of the object are the same for each avatar, the maximum altitude is different.

#### 2.2.2 Results

To illustrate the effect of the homotopy parameters  $\alpha_{i \in \{1,2\}}$ , a simulation has been performed for different values of each parameter. First, when  $\alpha_1 = \alpha_2 = 0$ , nothing happens, as expected. The object remains on the table, and each avatar "waits" for the other to perform the task.

When  $\alpha_1 = \alpha_2 = 1$ , each avatar tries to impose its desired trajectory. As the same gains have been used for each avatar's leader and follower controllers, the actual trajectory of the object is the average of both desired trajectories, as illustrated on figure 2.4. We can also see that significant interaction forces are generated (about 500 N of internal forces along the vertical axis).



Figure 2.4 – Result of the simulation of a collaborative task between two virtual humanoids when both act as leader. Left: desired trajectory for each avatar, and actual trajectory of the object. Right: vertical force applied by each avatar.

When  $\alpha_1 = \alpha_2 = 0.5$ , both avatars try at the same time to track their desired trajectory, but part of their action consists in complying to external forces. As all controllers have the same gains, the trajectory of the object is again between the desired trajectories of both partners. However, the internal forces along the vertical axis are only about 200 N. Note that in this case, the homotopy controller corresponds to a position-based first-order linear impedance controller, if we assume a stiff low-level position controller. If we define  $B = 2/\lambda_f$  and  $K = \lambda_p/\lambda_f$ , then assuming the Jacobian **J**<sub>i</sub> is full rank and the low-level position control makes the robot behave as a perfect position source, the motion of the gripper of partner *i* is



Figure 2.5 – *Trajectory of the object when one of the partners is a leader and the other is a follower in the Avatar-Object-Avatar experiment, where two virtual avatars lift an object and pass it over an obstacle.* 

described by the equation:

$$\mathbf{f}_{\mathbf{i}} = B\dot{\mathbf{p}}_{\mathbf{i}} + K\left(\mathbf{p}_{\mathbf{i}} - \mathbf{p}_{\mathbf{i}}^{\mathbf{d}}\right) \tag{2.9}$$

Finally, if  $\alpha_1 = 1$  and  $\alpha_2 = 0$ , the object follows the trajectory desired by partner 1, as shown on figure 2.5. We also notice a position overshoot, which means that the gains of the low-level position controller did not make the robot appear as a position source. Indeed, the position controller did not overcome the inertial effects at the end of the lifting phase.

# 2.2.3 Extension to a Person-Object-Avatar scenario



Figure 2.6 – *Experimental setup: a human operator uses a PHANToM Desktop device to lift a virtual object in collaboration with a virtual robotic partner.* 

We can extend the scenario by replacing one of the avatars by a human operator. The human partner operates through a PHANToM Desktop haptic device and has a 3-dimensional force feedback (see Figure 2.6). Figure 2.7 shows part of the



Figure 2.7 – Plan of the virtual avatar: desired motion of the object.

virtual scene and the trajectory of the object desired by robot to complete the task. This trajectory is unknown to the human partner and is not displayed during the task.

To assess the usability of the proposed model, we have conducted several experiments with one subject, where this time the homotopy parameter of the virtual avatar varies with time. The control law implemented on the avatar is the same as in the previous subsection. The task was either not divided, or divided into two or three parts, depending on the time-profile of the homotopy variable. We used the following time profiles:

- $\triangleright$  L: the virtual avatar tries to lead the whole task.
- $\triangleright$  *F*: the virtual avatar follows the human all along the task.
- ▷ *L*-*F*: the virtual avatar tries to lead (follows) during the first half of the motion, and follows (tries to lead) the human at the end of the motion.
- $\triangleright$  *L-F-L* (*F-L-F*): the avatar leads (follows) during the lifting and landing phases only, while the human operator leads (follows) while the object passes over the obstacle.

#### Stability

One important thing to consider when switching between two controllers in such a linear way is stability: for a bounded input, the controller must produce a bounded output. Though we did not tackle this issue theoretically, we plotted the output of the controller along the task to check its smoothness during the transitions between the leader's and follower's states. Figure 2.8 shows a typical joint torque output from the homotopy controller. The signal appeared to be smooth, with higher torques during the transitions.

#### Specialization

We could not highlight specialization from the force applied by the subject and his virtual partner on the object. This result was expected from [89], where even using a very simple setup, specialization of a human partner could not be obtained when collaborating with a robotic partner.

When the human user talked about his impressions on how the robot virtual avatar behaved, he explained that he did not trust the avatar, and thus applied a "safety" force, even when he felt like the avatar was leading the task, to be sure to



Figure 2.8 – Typical joint torques at the joints of the avatar (chest joint and right arm joints). The torque references sent to the actuators of the robot are smooth even when the value of the homotopy variable changes.

avoid the obstacle. This is a probable reason why we could not observe specialization. Moreover, we only looked for the same functional specialization as the one discovered by Reed, which consist in specializing in the acceleration and deceleration of the object. As our task is more complex, other specializations might have been elected by human partners performing the task.



#### **Favorite profile**

Figure 2.9 – *Trajectories when the human leads all the task (blue) and when he leads part of the task (green). The human trajectory goes further from the obstacle.* 

Questioned about his experience, the subject reported that he felt more comfortable when the avatar was following him during the lifting and landing phase (*F-L-F* time profile of the homotopy variable  $\alpha$ ), and especially when being close to the corners of the object, as he did not trust the virtual avatar. He also reported that the robot was suggesting him a more time-optimal trajectory, as it was moving much closer to the object than he would have done. Figure 2.9 highlights this fact by showing a typical trajectory of the object when the subject lead the task and when the robot was leading part of it. Note that when the robot is a follower all along the task, the trajectories of the object are very close to the ones reported in [75], while the trajectory desired by the avatar is more square-shaped as it closely follows the contour of the object (thus leading to more jerky motions). Hence, the reference path of the avatar is maybe unnatural to the human operator, which could explain why he did not follow the avatar passively.

#### Feeling of control over the task

Surprisingly, though the subject was presented several trials where the avatar was leading the entire task, the subject felt like it was never the case. This might be related to the stiffness of the leader controller  $\mathcal{L}$ . Maybe the controller did not offer enough disturbance rejection, thus giving the subject the feeling that he had some control over the task.

#### Unnatural changes in the robot behavior

Finally, it appeared that the subject was somehow disrupted at first when the robot changed his homotopy variable during the task. It took time for him to understand what was happening. A similar result has been reported in [16], in which subjects felt that a robot varying his level of assistance during the task, rather than between tasks, was unnatural. Our time profiles of the homotopy variable corresponded to artificial specialization, which was applied without taking into account contextual aspects or force signals, and which started abruptly after a set of trials in which the robot was a pure follower. This scenario seems not very probable in real-life situations, as Reed reported that specialization emerged quickly, but after several trials [91]. In future work, we plan to investigate on how to define the homotopy variable depending on the task, its context and the forces perceived by the virtual avatar so that its behavior is more user-friendly. This might require to use a more appropriate implementation of our abstract model (in other words, to change the  $\mathcal{L}$  and  $\mathcal{F}$  controllers).

# **2.3 Defining the homotopy**

In the previous sections, we have presented a new model to describe the behavior of two partners performing a physical collaborative task. The behavior of each partner is defined as a homotopy between a leader controller and a follower controller. The implementation of this model has been exemplified in the case of a linear homotopy with either a constant parameters or arbitrary time profiles.

In this section, we examine how the time profile of the homotopy parameter could be defined to enhance collaborative tasks. The first solution is to define the homotopy parameter by looking at the constraints of the system on which the controller is implemented. As an example, we propose the case of self-collision avoidance. A second strategy is to parameterize phenomena observed in humanhuman collaboration. We will investigate how our model could possibly reproduce the specialization phenomenon observed by Kyle Reed.

### 2.3.1 Enforcing constraints: self-collision avoidance

The homotopy controller switching model we propose allows a robotic system to smoothly switch between a leader and a follower behavior. The robot can exploit this switching to impose the user some specific trajectories that fit its own constraints.

Let us define the following scenario: a human and a humanoid robot are lifting and moving an object. The human partner initiates the motion and starts lifting the object. The robot then acts as a follower and complies to the intentions of the human. At some point, the trajectory suggested by the operator leads to auto-collision between the arm of the robot and its waist. Hence the robot takes leadership and the human operator then switches to a follower role.

This can be implemented by defining a homotopy parameter that depends on the collision distance between the arm and the trunk of the robot. Let  $\beta$  be a real number between 0 and 1. Consider the function  $\alpha$  defined as follows, built from a smoothed Heaviside function as proposed in [93]:

$$\begin{cases} y = \max\left(0, \min\left(1, \frac{d - d_{\min}}{d_{\max} - d_{\min}}\right)\right) \\ \alpha(y) = \frac{1}{2} + \beta \left(\frac{1}{2} \tanh\left(\frac{1}{y} - \frac{1}{1 - y}\right)\right) \end{cases}$$
(2.10)

where *d* is the minimum distance to self-collision, and  $d_{\min}$  and  $d_{\max}$  define the interval where the robot will switch from follower to leader. For distances  $d \le d_{\min}$ ,  $\alpha = 1$ , and for distances  $d \ge d_{\max}$ ,  $\alpha = 0$ . For  $d_{\min} < d < d_{\max}$ ,  $\alpha$  will smoothly decrease from 1 to 0.

This function can be used to define the homotopy between the leader and follower behaviors. When the robot is far from self-collision, it will behave as a follower. From distances below a threshold  $d_{\text{max}}$ , it will smoothly switch to a leader role, and for distances below  $d_{\text{min}}$ , it will act as a pure leader, following its own desired, collision-free trajectory. If the desired trajectory is reduced to a desired, safe position, then the obtained controller implements a compliance control with self collision avoidance.

This scenario raises planning issues, since at the beginning of the motion the humanoid robot is acting as a follower and hence it can be brought to any state. When it gets close to self-collision and switches to a leader role, the question arises of how to catch the original plan. If the state of the robot is far from its initial plan, a re-planning is necessary. As the transition between the follower and leader roles is smooth, the robot should compute what it would do if it was switching to a leader role from the current state, even when it is acting as a follower.

We implemented this collision avoidance scenario on an HRP-2 humanoid robot. The task consists in grasping an object and moving it in front of the robot. At first, the object is far on the right of the robot and its arm is far from its body. When the task is performed, the arm gets closer to its waist. In more complex tasks, it can be suitable that the robot starts leading the task in such situations, to suggest a plan that keeps it far from its own constraints. In our case, the robot will thus start the motion as a follower and switch to a leader role when the arm gets too close to the trunk, see figure 2.10.

To obtain this behavior, we defined the homotopy variable of the robot using equation 2.10. As the HRP-2 robot is position controlled, we used controllers that are similar to the one defined in the previous section: the leader controller is a Cartesian position controller, and the follower controller is a Cartesian admittance



Figure 2.10 - A human operator and HRP-2 jointly moving an object towards the left-hand side of the robot. During the motion, the right arm of the robot moves towards the chest, increasing the risk of self-collision.

controller. The homotopy controller  $\mathscr{U}$  outputs a desired velocity  $\dot{\mathbf{x}}_{\mathbf{r}}$  for the robot's gripper which is sent to an inner controller performing inverse kinematics and PD joint control to track the desired trajectory of the gripper. We performed the inverse kinematics using the pseudo-inverse  $\mathbf{J}_{\mathbf{r}}^{\#}$  of the Jacobian. The desired joint velocity  $\dot{\mathbf{q}}_{\mathbf{d}}$  reference is computed by solving for the output reference velocities of the leader and follower controllers,  $\dot{\mathbf{x}}_{\mathbf{r}}^{\mathscr{G}}$ :

$$\begin{cases} \dot{\mathbf{x}}_{\mathbf{r}}^{\mathscr{L}} = \lambda_{l} \left( \mathbf{x}_{\mathbf{o}}^{\mathbf{r}} - \mathbf{x} \right) \\ \mathbf{M}_{\mathbf{r}} \ddot{\mathbf{x}}_{\mathbf{r}}^{\mathscr{F}} + \mathbf{B}_{\mathbf{r}} \dot{\mathbf{x}}_{\mathbf{r}}^{\mathscr{F}} = \mathbf{f}_{\mathbf{s}} \end{cases}$$
(2.11)

where **x** is the actual position of the end-effector of the robot, and the force  $\mathbf{f}_s$  applied on the gripper of the robot is computed from the measurements of a 6 axes force sensor placed at the robot's wrist. The Cartesian reference velocity  $\dot{\mathbf{x}}_r^{\mathcal{U}}$  for the end-effector and the desired joint velocity  $\dot{\mathbf{q}}_d$  are then computed as:

$$\begin{cases} \dot{\mathbf{x}}_{\mathbf{r}}^{\mathscr{U}} = \alpha_{r} \dot{\mathbf{x}}_{\mathbf{r}}^{\mathscr{L}} + (1 - \alpha_{r}) \dot{\mathbf{x}}_{\mathbf{r}}^{\mathscr{F}} \\ \dot{\mathbf{q}}_{\mathbf{d}} = \mathbf{J}_{\mathbf{r}}^{\mathscr{H}} \dot{\mathbf{x}}_{\mathbf{r}}^{\mathscr{U}} \end{cases}$$
(2.12)

The joint velocity reference  $q_d$  is numerically integrated and sent to the PD joint controller of the robot provided by the control software OpenHRP.

The desired trajectory for the leader controller is defined by a set of key points, which are interpolated using splines. The resulting function  $\mathbf{x}_{\mathbf{0}}^{\mathbf{r}}$  is a smooth function defined on [0,1]. The reference sent at time *t* to the leader controller is  $\mathbf{x}_{\mathbf{0}}^{\mathbf{r}}(\rho(Y_t))$ , with  $Y_t = \max(0, \min(1, \frac{t}{T}))$ , where  $\rho$  has a bell-shaped profile and  $\dot{\rho}$  reaches its maximum for  $t = \frac{T}{2}$ . Hence if the desired trajectory of the gripper is perfectly tracked, the maximum velocity is reached at the middle of the range of the motion.

The minimum distance between the arm and the trunk of the robot is computed by [23], and the homotopy parameter is computed at each time using equation 2.10. Figure 2.11 shows the motion of the object on the (y, z) plane, the minimum distance before self-collision and the homotopy parameter of the robot. At the beginning of the motion, the robot follows the human. When the human brings the object in front of the robot, the robot starts switching to a leader role and takes control of the task. Note that a similar effect could be obtained by implementing an impedance controller and changing the gains online, as in [41]. However, our approach will allow the use of different controller shapes for the leader and follower behaviors, and an *active* behavior in response to the risk of constraint violation.



Figure 2.11 – Evolution of the homotopy parameter and collision distance along the task performance (bottom graph, solid line: homotopy parameter, dashed line: collision distance). On the top graph, the trajectory of the robot is shown. The blue line corresponds to the plan of the robot, the yellow line corresponds to the actual trajectory followed by the robot.

Here, the planning issue mentioned earlier appears clearly. Around t = 15s, the homotopy variable of the robot is close enough to 1 for the robot to mostly contribute to the motion of the object. However, by the time it happens, the desired state of the robot is already the final position of the hand. Consequently, there is a jump in the velocity of the hand of the robot, which starts going down abruptly.

To tackle this issue, a more appropriate strategy would be suitable. A possibility would be that the robot replans a desired motion regularly even when it behaves as a follower, so that when it switches to a leader mode, the current state of the task is close enough to the plan of the robot. This is however only possible for simple tasks. Investigations on collaborative planning should be conducted to tackle this problem appropriately and to design suitable leader behaviors for complex tasks.

## 2.3.2 parameterizing phenomena observed in human-human collaboration

Another strategy to define appropriate homotopies between leader and follower behaviors is to look at phenomena observed in human-human collaborative tasks. An interesting phenomenon, from which we can easily derive a homotopy, is the specialization phenomenon highlighted by Reed [91]. Specialization has been shown to emerge across trials of a collaborative positioning task where two human partners rotate a two-handled crank to reach a desired angle in a minimum amount of time. During the early trials of the task, partners participated to both the acceleration and the deceleration of the crank. However, a different role distribution was generally established in later trials, with one partner accelerating the crank and the other partner decelerating the crank.

We defined specialized behavior as having a homotopy parameter that is different from  $\frac{1}{2}$ , and that can switch to values closer to 0 to values closer to 1 depending on some criteria. The more extreme the values of the homotopy parameter is, the

"more specialized" the behavior is. Hence, specialization is described by specific profiles of the homotopy mappings for each partner. In this scenario, we present profiles that correspond to the particular case of the specialization highlighted by Reed, and show that by defining appropriate  $\mathcal{L}_i$  and  $\mathcal{F}_i$  controllers, specialization also applies to point-to-point tasks.



Figure 2.12 – 1 degree-of-freedom collaborative positioning task.

The collaborative task we consider is a 1 degree-of-freedom point-to-point task where two partners, 1 and 2 move a punctual mass M by applying forces  $f_1$  and  $f_2$  on the object. The position of the object is denoted x,  $x_i$  is the starting position of the object and  $x_f$  is its target position. Figure 2.12 summarizes the notations.

The specialization phenomenon described by Reed is characterized by having one partner accelerating the object and one partner decelerating the object. While one partner accelerates the object, the other applies less force in the accelerating direction. Hence, we can interpret this as having one partner leading the acceleration phase and one partner leading the deceleration phase. Each partner switches from one role to the other during the task, when switching from the acceleration to the deceleration phase. To translate this evolution, we must define a profile for the homotopy mappings that will have different extreme values at the beginning and end of the task. To obtain a smooth behavior from the partners, the homotopy parameter should smoothly switch from one value to the other.

To implement this switching, let us define the following smooth heaviside function:

$$\begin{cases} y = \max\left(0, \min\left(1, \frac{x - x_i}{x_f - x_i}\right)\right) \\ \alpha(y) = \frac{1}{2} + \beta \left(\frac{1}{2} \tanh\left(\frac{1}{y} - \frac{1}{1 - y}\right)\right) \end{cases}$$
(2.13)

where *y* is the normalized position of the object, such that  $y(x \le x_i) = 0$  and  $y(x \ge x_f) = 1$ . This function is so that  $\alpha(0) = \frac{1}{2} + \frac{\beta}{2}$ ,  $\alpha(1) = \frac{1}{2} - \frac{\beta}{2}$ , and  $\alpha(\frac{1}{2}) = \frac{1}{2}$ . The plot of this function for different values of  $\beta$  is shown on figure 2.13.

We define the homotopy parameter for each partner as:

$$\begin{cases} \alpha_1(y) = \alpha(y) \\ \alpha_2(y) = 1 - \alpha(y) \end{cases}$$
(2.14)

then, if  $\beta = 0$ , both partners are between the leader and follower mode during the whole motion, and do not specialize. If  $\beta > 0$ , partner 1 starts the task as a leader and smoothly changes his/her around the middle of the motion to finish the task as a follower, while partner 2 adopts the opposite behavior. If  $\beta = 1$ , then the behaviors at the beginning and at the end of the task are pure leader and follower behaviors. To simulate the emergence of specialized behavior, we can simulate partners 1 and 2 using increasing values of  $\beta$  from 0 to 1 from trial to trial.

Now that we have defined a specialized behavior, we define the controllers  $\mathscr{L}_i$  and  $\mathscr{F}_i$  to obtain a behavior similar to that adopted by human partners, i.e.



Figure 2.13 – Profile of the homotopy variable for different values of  $\beta$ . High values of  $\beta$  give highly specialized profiles. The specialization is less pronounced for small values of  $\beta$ . An unspecialized profile is obtained for  $\beta = 0$ .

a functional specialization where one partner accelerates the object and the other partner decelerates it. That is, for  $i \in \{1,2\}$ :

$$\begin{cases} \mathscr{L}_{i}: f_{i}^{\mathscr{L}} = K_{i} \left( x_{o}^{i} - x \right) + B_{i}^{l} \left( \dot{x}_{o}^{i} - \dot{x} \right) \\ \mathscr{F}_{i}: f_{i}^{\mathscr{F}} = -M_{i} \ddot{x} - B_{i}^{f} \dot{x} \end{cases}$$
(2.15)

where  $x_o^i$  is the reference trajectory of the object for partner *i*, and  $K_i$ ,  $B_i^{l,f}$ ,  $M_i$  and  $K_i$  are positive control gains. The output  $f_i$  of the controller  $\mathscr{U}_i = \alpha_i \mathscr{L}_i + (1 - \alpha_i) \mathscr{F}_i$  is a force applied to the object by partner *i*. Hence the motion of the object obeys the following equation:

$$M\ddot{x} = f_1 + f_2 \tag{2.16}$$

Finally, the desired motion of the object for partner *i* is defined using the minimum jerk model:

$$\begin{cases} T_i = \min\left(1, \frac{t}{T_d^i}\right) \\ x_o^i(T_i) = x_i + (x_f - x_i)(10T_i^3 - 15T_i^4 + 6T_i^5) \end{cases}$$
(2.17)

Here, we assume same desired initial and final positions  $x_i$  and  $x_f$  for each partner, and only the duration of the motion  $T_d^i$  is different for partners 1 and 2. This means that both partners want the object to reach the same target position, but each partner may choose to move at a different pace. The system is simulated under MATLAB, with the profiles shown on figure 2.13 for the homotopy parameters  $\alpha_1$  and  $\alpha_2$ . Figure 2.14 shows an overview of the simulation model.

The resulting forces applied by the partners to achieve the task are plotted on figure 2.15. The non-specialized and fully specialized profiles are plotted separately on figure 2.16 to highlight the difference between these two extreme cases. The motion of the object in the fully specialized case as well as its velocity, the applied forces and the homotopy parameters are shown on figure 2.17.

In the case of the bottom plot of figure 2.16, the force profiles are similar to the specialized profiles shown in [91]: one partner mainly applies positive forces, with



Figure 2.14 – Simulation model.  $PD_i$  denotes the Proportional-Derivative controller used to implement the leader behavior.  $Z_i^f$  denotes the follower impedance controller.



Figure 2.15 – Force applied by each partner on the object to reach a common target position, for different specialization profiles (from non-specialized to fully specialized).

high forces during the acceleration phase and low forces during the deceleration phase. The other partner applies mainly negative forces, with higher forces during the deceleration phase.

This simulation was performed with different gains for each partner and different desired motion durations. However, the difference between the gains and desired durations was kept relatively small compared to their values (random values were generated with a normal distribution around an identical values for each partner, and a standard deviation arbitrarily chosen between 1% and 20% of the values). The following facts have been empirically noticed:

- specialized force profiles are generally obtained when the follower applies relatively low forces on the object;
- ▷ specialized force profiles appeared more often when the partner with the smaller motion duration started as a leader.

This last point can be intuitively understood. If the slower partner starts the motion, then at the middle of the range, when the faster partner starts leading the



Figure 2.16 – Force applied by each partner on the object to reach a common target position:  $\beta = 0$ , unspecialized case (up);  $\beta = 1$ , fully specialized case (down).



Figure 2.17 – Motion of the object (up), forces applied by the partners (middle) and homotopy parameters of each partner (down).

task, s/he is behind schedule regarding her/his plan. Hence s/he will accelerate the object to catch up her/his original plan. As a result, the partner who leads the end of the task will not only decelerate, but will also accelerate the object, which does not correspond to the functional specialization described in [91]. This gives us a hint on *how* the partners specialize, i.e. how they will decide who will adopt a *leader-to-follower* scheme and who will adopt the reverse *follower-to-leader* scheme: the partner who leads the beginning of the task might be the partner who has the faster desired motion. However, our simulations are not sufficient to draw solid conclusions on this issue.

## 2.3.3 Conclusion

In this section, we have examined two strategies to define the homotopy between the leader and follower behavior on a robotic system. The first one consists in looking at internal constraints of the robot, so that the robot will only act as a follower when it is far from its constraints. This concept has been exemplified with self-collision avoidance.

The second strategy is to parameterize phenomena observed in human-human behaviors. By parameterizing phenomena such as specialization, one can obtain *task signatures* in terms of role distribution which can then be used by a robotic system to adopt a natural, human-like behavior.

Finally, a third strategy that we did not mention would be to look at the behavior of the human and to adapt the homotopy parameter based on the human state or intentions. A typical example would consist in taking the lead when the human is inactive, and switch to a follower mode when the human takes an initiative.

# 2.4 Conclusion

The motivation of our work is to establish a physical haptic interaction model which does not appoint any agent composing a dyad to be in either a follower or leader role. Our approach consists in considering that in realizing a given task, each individual would behave in an extreme case as either an agent who imposes his intention suspecting from the collaborator to be a gentle follower or in a reverse way. We believe that in reality these extreme cases are rarely reached and made the hypothesis that each individual behaves according to a continuous weighted control between these two extreme cases. This weighting is realized by a homotopy (interpolation) switching between either two distinct controllers (one ensuring a follower behavior the other one the leader behavior), or between two sets of gains for a single controller (case of adjustable impedance).

We believe that this model is expressive enough to encompass many scenarios that can be imagined in dyadic collaborative tasks, such as conflicting situations, dead-locks, or hesitations. We also believe that such a model can call for haptic communication, because the partners will have to negotiate a role distribution along the task.

We demonstrated how our model can be applied to human/humanoid physical collaboration through exemplifying its concrete use in two scenarios: (i) homotopy is able to encompass role switching in specialization and (ii) homotopy role switching appears as an elegant way for the robot to avoid self-collisions by acting directly on the mapping to take the leadership of the task; this scenario is experienced with the HRP-2 humanoid robot moving collaboratively an object with a human partner: the robot takes leadership when getting close to violate the self-collision constraint. These two scenarios define two classes of strategies to define the homotopy, namely looking at the internal constraints of the robot, and parameterizing phenomena observed in human-human collaborative tasks. These parameterizations can then be seen as *task signatures* in terms of role distribution. A third class of strategies can be to adjust the homotopy parameter based on the interpretation of the human behavior and intention. This option has not been investigated in the context of this thesis, but will be examined in the near future.

Our model also raises the necessity of further investigations. First of all, stability conditions have to be investigated. Even if the leader and follower controllers are stable, there is no guarantee that their homotopy mapping will be a stable controller for any function  $\alpha$ . Second, assuming that this model properly describes the behavior of a cooperating human, which still has to be assessed, then, based on limited knowledge on the underlying controllers, how can the homotopy variable of a human operator be evaluated? This question is important if we want to implement human-robot collaboration schemes, since this knowledge is likely to be useful to adjust the homotopy variable of the robot partner.

The following chapter is a first step towards the validation of our model as a description of human behavior. Assuming the model is valid, the results of next chapter suggest that the weighting between the leader and follower behavior would actually be *continuous*, and would not correspond to abrupt changes between a leader and a follower behavior.
## Learning haptic patterns

## Contents

Programming by demonstration of collaborative tasks		
3.1.1	Introduction	57
3.1.2	Programming by Demonstration	58
3.1.3	Application to Human-Robot collaborative lifting tasks	59
Demo	nstration setup and experiments	59
3.2.1	Hardware setup and controller	60
3.2.2	Experiments	62
3.2.3	Conslusions	62
Learni	ing collaborative tasks	64
3.3.1	Probabilistic encoding of the task	65
3.3.2	Probabilistic reconstruction of the task	66
3.3.3	Control scheme	68
Task r	eproduction	70
3.4.1	Reproduction setup	70
3.4.2	Preliminary analysis	70
3.4.3	Quantitative analysis	73
Concl	usion	79
	Progra 3.1.1 3.1.2 3.1.3 Demo 3.2.1 3.2.2 3.2.3 Learn 3.3.1 3.3.2 3.3.3 Task r 3.4.1 3.4.2 3.4.3 Concl	Programming by demonstration of collaborative tasks3.1.1Introduction3.1.2Programming by Demonstration3.1.3Application to Human-Robot collaborative lifting tasksDemonstration setup and experiments3.2.1Hardware setup and controller3.2.2Experiments3.2.3ConslusionsLearning collaborative tasks3.3.1Probabilistic encoding of the task3.3.2Probabilistic reconstruction of the task3.3.3Control scheme3.4.1Reproduction setup3.4.2Preliminary analysis3.4.3Quantitative analysisConclusion

W<sub>E</sub> are interested in giving robotic systems the ability to cooperate with human operators as partners, and not just as (passive) followers. In this chapter, we adopt a *Programming by Demonstration* approach: we propose to *teach collaborative tasks* to a robot, rather than explicitly program them. By the mean of probabilistic tools used in automatic learning, we try to encapsulate the complexity of collaborative tasks by automatically extracting the important features that characterize them.

Hereby, we present a probabilistic framework and an experimental setup which allows us to demonstrate a collaborative lifting task to a robot. During the demonstrations, the robot is taught the two extreme leader and follower behaviors. These two behaviors can be clearly separated in the data-space, and their characteristics are encoded using Gaussian Mixture Models. The models of both behaviors are used jointly during the reproduction phase to retrieve the skill by performing a Gaussian Mixture Regression. As a consequence, the robot can choose between both behaviors when performing the task with a human partner. Using the proposed method, several successful lifting tasks could be performed. It appears that during the reproduction experiments, several subjects acted in such a way that the robot switched between the leader and follower behaviors during the task. This supports the hypothesis formulated in the previous chapter, namely, that partners can switch roles during a collaborative task and that this switching is continuous.

The work presented in this chapter has been realized in collaboration with Sylvain Calinon, Elena Gribovskaya and Aude Billard from the Learning Algorithms and Systems Laboratory, Ecole Polytechnique Federale de Lausanne. They have provided probabilistic framework for teaching collaborative tasks by demonstration and focused on the demonstration method and the learning phase. Our contributions have been to implement the teleoperation setup for the demonstrations, to integrate the controller developped at the LASA, to conduct the reproduction experiments and analyze the experimental data, and to *exploit the probabilistic framework to assess the extended leader-follower model proposed in this thesis.* For clarity, some technical details about the probabilistic framework are briefly introduced here. The interested reader is kindly invited to read [11] and other papers from the LASA, EPFL for a deeper presentation of the aspects related to automatic learning.

## 3.1.1 Introduction

To perform collaborative manipulation tasks, the partners concurrently apply forces on a common object of interest. During the task, they regulate the exchange of mechanical energy not only based on the task to realize and environmental constraints, but also through mutual understanding of each other's intentions. Since most tasks can be realized in different manners (*e.g.* different velocity profiles or trajectories), the partners have to mutually adapt to negotiate a common plan and perform the task in a good synergy.

We believe that this multilateral adaptation process relies on *haptic communication* among the partners. This is however merely an hypothesis and little is known about mutual adaptation processes in collaborative tasks. It is therefore challenging to replace a human partner with a robotic one. The problem is usually simplified by assigning a fixed follower role to the robotic partner, thus avoiding the negotiation of a common plan between the robot and the human partner.

We proposed in chapter 2 an extension of the usual leader-follower model that allows the robot to smoothly switch between the leader and follower behaviors, or to adopt a mixed behavior during the whole task [26]. The behavior of the robot is therefore characterized by a time-varying leadership ratio that balances both behaviors. We proposed different strategies [25] to define and adapt this ratio. The first one is based on knowledge about usual role distributions used by human partners for a given task. The second one is based on the constraints of the task or of the robotic partner.

In order to generalize these strategies to *any* task to adapt the leadership ratio will require investigations on each task to extract role distributions favored by human partners (if any). Moreover, a mapping between the task constraints and the leadership ratio must be devised. Finally, for the robot to act as a leader, a task model is also necessary. Implementing these strategies explicitly can therefore be a daunting task.

Bringing the programming of a machine (computer or robotic system) to a less technical level is the aim of the *Programming by Demonstration* paradigm. Instead of programming a complex motion based on an explicit model of a task, a *teacher* will for instance demonstrate a task to the robot by seizing its arms and moving them to perform the task, much like a parent will teach a child how to cut his meat. Hence, our goal is to avoid deriving explicit models and role switching strategies for the task, by using a Programming by Demonstration framework.

This chapter is built upon the work presented in [11], which investigates how to demonstrate a collaborative lifting task to a robotic system through a teleoperation setup. For the robot to extract the characteristics of the task and generalize to undemonstrated situations, a representation of the task through *Gaussian Mixture Models* (GMM) is used, which encode local correlations between the variables of the task. From a set of input data, outputs can then be reconstructed on-the-fly using a *Gaussian Mixture Regression* process.

We performed reproduction experiments on a real humanoid platform, to assess whether the model presented in [11] is able to *catch all the aspects and complexity of the haptic cue exchange during collaborative physical task*, including the dynamics of the motion, the synchronization and adaptation processes and the underlying haptic communication. We analyze the experimental result to evaluate the teaching method, the learning process and, more generally, the usability of the Programming by Demonstration and GMM-GMR framework to endow a robot with collaborative skills. Then we try to use obtained results to see whether our theory developed in the previous chapter is somehow viable, at least for what concerns role switching.

The work presented in [11] is hereby reviewed with the benefit of hindsight gained through the experiments, and in the perspective of the extended leaderfollower model presented in the previous chapter. What is presented in this chapter is therefore not the approach initially followed to transfer collaborative skills by demonstration to a robot. It is rather a synthesis that shows how the Programming by Demonstration methodology and probabilistic framework proposed in [11] can be used to allow smooth switching between the leader and follower roles without explicit models, how it relates with our extended leader-follower model and how it can be used to assess its validity.

## 3.1.2 Programming by Demonstration

Programming by demonstration appeared in the 1980's where it has been used in industrial robotics to avoid the explicit programming of robotic tasks. A complete review of the history and state of the art in the field, with many references, is given in [7]. The purpose of this paragraph is not to provide a complete survey about the topic, but to recall some basic notions in this paragraph, which are useful to understand the chapter.

In the Programming by Demonstration paradigm, a teacher will demonstrate a task to a robot, so that the robot can autonomously reproduce the task. A simple way to reproduce a skill is for example to extract keypoints of a trajectory, or take snapshots of the robot's state during the demonstration, and to replay the trajectories by interpolating between these keypoints. Such a strategy will yet not allow the robot to generalize the skill to adapt to new situations. Advances in machine learning allowed to address the generalization problem at two levels: how to extract the important features of a task from a set of demonstrations, and how to generalize to new situations.

#### Generalization

The problem of generalizing across the demonstrations consists in extracting the constraints of a task. Let us consider the example of picking a cube on a table and passing it over an obstacle. The first phase of the task, which consists in reaching the cube, can be accomplished in many different ways, namely by starting from any point in space, and by moving at different velocities towards the cube. However, at some point, the distance between the hand and the cube must become zero. This is a characteristic of the task that the robot must be able to extract by generalizing this property over all the demonstrations.

When teaching a task, a human operator will not be able to demonstrate all possible ways of realizing the task in all possible contexts. Hence, we expect the robot to be able to adapt to some extent to new situations. In the task described above, the objectives remain unchanged whatever the starting position of the cube, or the height of the obstacle. Therefore, we expect that if new obstacles are presented to the robot, the latter will not fail to do the task. To increase the chances that the robot can *actually* adapt to new situations, two aspects must be considered: first, demonstrations must be performed with enough variability in the variables over which we expect the robot to generalize (e.g. the relative position of the cube and the obstacle at the end of the task). Else, the robot might generalize over the trials and extract constraints on these variables, preventing any adaptation. Second, appropriate variables must be defined to describe the task. For example, considering the absolute altitude of the cube when passing over the obstacle will not allow to generalize to obstacles of a different size. Considering the distance to the obstacle is more relevant in this case.

#### **Demonstration modalities**

Different interfaces can be used to transmit a skill to a robot. A very natural cue is to rely on the visual perception of the robot, so that the teacher naturally performs the task, and the robot looks at him to extract the features of the task. This can be augmented by using motion capture to further get more precise information on the posture of the teacher. Other possibilities are to teleoperate the robot, or to directly seize and move its members to perform the tasks. Examples and references can be found in [7], and in [9, 10].

## 3.1.3 Application to Human-Robot collaborative lifting tasks

If two partners perform a lifting task collaboratively, they are likely to have different intentions regarding some task parameters, such as the velocity profile, or the target altitude of the object. Hence, during the motion, they will have to adjust towards a common plan. We make the hypothesis that this is made partly through haptic communication cues, *i.e.* that the partners will guess each other's intentions by interpreting the interaction forces they sense at the grasping points of the manipulated object. This hypothesis relies on the common sense fact that a disagreement between the partners will result in higher interaction forces.

We hereby consider a one-dimensional task where the position of the lifted object is described by its altitude. We also consider the vertical velocity of the object and the force applied by the robot along the vertical axis. We assume the mass of the object is known, and more importantly, that the task will be demonstrated in predetermined stereotypical manners, where *one partner is asked to always lead* and the *other is asked to act as a pure follower*. Finally, we also assume that the partner acting as a follower is blindfolded, so that visual cues do not interfere with the task.

Rahman *et. al* showed that the dynamics of a follower human arm during collaborative task was dominated by a variable damping term [86]. The force applied on the robot's wrist and the gripper velocity should therefore be taken into consideration in order to reproduce human-like dynamics during the task. In our case, we consider that the robot can also have its own desired trajectory to perform the task. A position-dependent term can be added to the dynamics to drive the arm towards a target position. In the context of this work, we will consider the position x of the robot's gripper and its velocity  $\dot{x}$ , as well as the force f sensed by the robot at the grasping point, as depicted on figure 3.1.

## **3.2** Demonstration setup and experiments

This section presents the experimental setup adopted to demonstrate a collaborative lifting task to the humanoid robot HRP-2. The task is demonstrated by adopting two extreme role distributions: one where the human partner is leading, and one



Figure 3.1 - The three variables considered in our study. The vertical force f is the force exerted on the robot's wrist.

where he is following. In both cases, the robot is teleoperated to adopt a complementary behavior. The analysis of the data recorded during the demonstrations shows that the data-sets corresponding to both role distributions are clearly separated along the force axis. We conclude that role swithcing during the reproductions can be considered.

## 3.2.1 Hardware setup and controller

#### Hardware setup

Collaborative lifting tasks have been demonstrated on the full sized humanoid robot HRP-2. As the joints of the HRP-2 robot are subject to significant friction, kinesthetic teaching where the teacher manipulates the robot by direct contact is not possible, as the unactuated motion of the HRP-2 robot is not compliant enough. Active compliance can be implemented, but forces need to be applied after the wrist force sensor, which is impractical when the robot is holding an object. As a result, a teleoperation setup with kinesthetic feedback was chosen to demonstrate the collaborative lifting tasks to the HRP-2 robot.

Figure 3.2 shows the experimental setup used during the teaching phase. During the demonstrations, a human operator (the *teacher*), teleoperates the robot using PHANToM device with 6 degrees-of-freedom force feedback. Hence, the teacher had a full feedback of the interaction wrench measured at the gripper of the robot. A second operator (the *operator*) assists the teleoperated robot to lift a beam, while keeping it horizontal. Only the right arm is used to perform the lifting task, while the robot is standing. The wrist of the robot is constrained to move only along the vertical direction during the whole task, while its orientation is constrained to remain constant.

#### **Overall control scheme**

PHANToM devices are impedance type devices, meaning that they are low friction, low inertia mechanisms, and accept force and torque references, whereas the HRP-2 robot accepts joint position commands. Thus, a natural coupling scheme is a bilateral 2-channel Velocity-Force coupling. The vertical velocity of the tip of the PHANToM device was measured and sent as a velocity reference to the robot. The vertical force was measured at the wrist of the robot, and sent as a reference to the PHANToM device.



Figure 3.2 - A human (teacher) teleoperates the humanoid robot HRP-2 through a force feedback haptic display to demonstrate how to perform a collaborative task with a human partner (operator).

## Stability

The reader who is familiar to teleoperation systems will notice that this scheme is particularly simple and will certainly lead to instability when the teleoperator will be in contact with a stiff environment, because of unavoidable time delays. Different solutions exist to avoid these instabilities: the transmission of wave variables instead of power variables, to ensure the passivity of the transmission system [79], and the addition of impedances at the ports of the transmission setup. Nevertheless, this is at the cost of transparency, and will reduce the bandwidth of the system. As we want to investigate human-human haptic communication, we want to have a bandwidth as large as possible.

To ensure the stability of the system in the context of our experiment, we rescaled the force fed back to the PHANToM device. This rescaling did not reduce the bandwidth of the system. The value scale factor applied to the robot sensor forces was set to 0.3, and was obtained empirically. Since only 30% of the sensed force was fed back to the operator, we feared that the task performance would be decreased. Thus, we also tried to damp the motion, so as not to impact the amplitude of the forces in the bandwidth of the system. However, the damping caused by usual stabilizing methods in teleoperation resulted in slower motions, and the slight forces caused by differences in the operator's intentions could hardly be felt by the teacher using the master device. The force rescaling method felt more natural to the human teacher and resulted in better demonstration performance than reducing the bandwidth of the system.

#### Haptic communication

The stability of the proposed setup can be affected by time delay, but the system allows the robot to perform fast motions, as a human would, because its apparent inertia to the teacher is actually the one of the PHANToM device. Furthermore, the operator will sense all and only the forces sensed at the wrist of the robot, and these forces will not be filtered by any admittance before being transmitted to the operator. This has two implications:

▷ little information will be lost between the teleoperator and the operator, because only the very high frequencies of the force signal will be filtered out; ▷ if the operator manages to perform the task, then the resulting performance *defines what we could definitely expect* from a robot performing such a task based only on sensor measurements.

This second point is very important, because our goal is to endow robotic platforms with the ability of performing physical collaborative tasks with a human partner, and we believe that the haptic signal has a great importance in the task performance. In our setup, the teacher using the PHANToM device has his eyes closed when following the other human partner. Since the forces transmitted to this operator are only the ones measured at the robot's wrist, if the operator can perform the task with a very good performance, it means that the *haptic signal measured at the robot's wrist conveys enough information* to perform the task in a stand-alone mode with the same performance as the human operator. This allows us to know where to put our expectations.

## **3.2.2** Experiments

The teaching process was conducted with one teacher (the robot's behavior was thus personalized to this teacher), and two humans as task-partners. The task consisted in lifting an object synchronously so as to keep the object horizontal. The subjects were instructed not to vary the final altitude of the object across trials, although no marker indicated a common target altitude to reach. The mutual adaptation between both partners thus lied in the velocity profile to reach the target altitude, and in the force profiles applied by each partner to realize the motion.

Two sets of scenarios were demonstrated to the robot. In the first set, the teacher closed the eyes and the user initiated and ended the motion. In the other set, the roles were exchanged, and the teacher lead the onset and end of the motion. Given the short duration of the motion (less than 10 seconds), this practically corresponds to caricatural fixed leader-follower distributions. Therefore, the robot was taught to realize the task with two different dynamics, corresponding to two extreme role distributions among the partners.

The position and velocity of the robot's gripper and the force measured by a force/torque sensor located at the robot's wrist were recorded at 200Hz. A challenge when dealing with the analysis of a force signal coming from a force/torque sensor is noise. The force signals had to be filtered prior to analysis, which somehow limited the frequencies in which we could observe potential haptic phenomena.

The results are illustrated in figure 3.3. We notice a certain consistency in the position-velocity plane. The force-velocity plane shows two separate patterns for the two different teaching behaviors (robot leader and robot follower and the counterpart task-partner follower and task-partner leader respectively). The forces in the case where the robot is leader are lower than in the follower case, because the partner follows the robot with a lag. Hence the robot has to "pull" the human partner and a negative force with a higher amplitude is measured. This is clearly visible on the forces' time profiles on figure 3.3.

## 3.2.3 Conslusions

We can draw several conclusions from the graphs shown in figure 3.3. By looking at the relationship between position and velocity (bottom-left corner), we see that



Figure 3.3 – First row: force against time and position against forces. The forces in the "leader robot" case (in red) and the "follower robot" case (in green) are clearly separated. Second row: velocity profiles plotted against the position and force variables. The trajectories in red and the trajectories in green correspond respectively to the demonstrations of the robot acting as a leader and as a follower. The circles represent the beginning of the motions.

the kinematics of the task in both role distributions are not very different. Therefore, an observer looking at the two partners when they perform the task is unlikely to tell which of the partners is leading and which is following. The only slight difference is that the velocity decreases at an earlier stage of the motion when the robot follows the human partner.

By looking at the bottom-right graph on figure 3.3, however, we see that the dynamics of the motion, *i.e.* the concurrent evolution of force and velocity, is different in both cases.

Finally, we notice that what separates both behaviors is the force measured at the wrist of the robot. This means that given an input force, it is possible to determine the role distribution among the partners: roughly speaking, forces above -7N correspond the the case where the robot acts as a follower, and forces below this value correspond to the case where the human acts as a follower<sup>1</sup>. This distinction is however impossible to do with the kinematical variables: for a given position or velocity, it is impossible to know the applied force, and thus to know the role distribution among the partners. Therefore, without regard for the algorithms and controllers used to reproduce the task, the measured force will have to be an *input* to the system.

Considering measured forces as an input will allow to determine, during reproductions, what task dynamics the robot should select: the one learned when it acted as a leader or the one when it aced as a follower. Thus, by asking the teacher and his partners to demonstrate two extreme role distributions, a criterion to select the role distribution could be extracted for the task. In this particular context, it is possible to consider switching between the two role distributions based on the behavior of the human partner, for the partner can decrease the applied force during the task so as to be in the "leader robot" domain, or increase it to move towards the "follower robot" domain.

## 3.3 Learning collaborative tasks

The previous section has introduced a setup and an experimental method to demonstrate a collaborative lifting task to a robotic system. The analysis of the data recorded during the demonstrations shows *two distinct patterns* in the forceposition-velocity space corresponding to the two demonstrated role distributions.

In this section, we present the probabilistic framework which we used to encode these patterns, so as to retrieve the skill in autonomous lifting tasks performed with human operators. This framework was developped at the LASA, EPFL and has been applied to collaborative tasks in the context of a joint work in the European project Robot@CWE.

Understanding these tools is important to analyze and understand the experimental results, and to draw conclusions about the usability of this framework to transfer collaborative skills to a robot. Some aspects must also be presented to fit the experimental results with the extended leader-follower model presented in chapter 2.

<sup>&</sup>lt;sup>1</sup>We will see later in the chapter that the limit between both behaviors is not that sharp.

## **3.3.1** Probabilistic encoding of the task

The data consist in a set of points  $D = {\xi_k}, k = 1..N_d$  recorded during the demonstrations of the task ( $N_d$  is the number of recorded data points). Each point  $\xi$  is defined as  $\xi = [x f \dot{x}]^T$ , where  $x, \dot{x}$  and f are the position and velocity of the robot's gripper and the vertical force measured at the robot's wrist at a given time of given demonstration.

We considered the data-sets corresponding to the two role distributions separately:  $D_{\mathscr{L}}$  is the data-set corresponding to the "leader robot-follower human" distribution, and is depicted in red on figure 3.4. Conversely,  $D_{\mathscr{F}}$  is the data-set corresponding to the "follower robot-leader human" distribution, and is depicted in green on figure 3.4.



Figure 3.4 – The two data-sets recorded during the demonstration. The data-set  $D_{\mathscr{L}}$  corresponding to the demonstrations with a leader robot is depicted in red, and the data-set  $D_{\mathscr{F}}$  corresponding to the complementary role distribution (follower robot and leader human) is shown in green.

The probabilistic framework used to encode the characteristics of the demonstrated task is based on Gaussian Mixture Models (GMM). A GMM is a combination of Gaussian probability density functions, used to estimate an unobservable density function from a set of data samples. GMMs are often used as classifiers, but in our context, they are used as an efficient and compact way to encode local correlations and variations across different variables in a data-set. A GMM is defined by a set of *K* components corresponding to *K* Normal distributions  $\mathcal{N}(\mu_i, \Sigma_i), i = 1..n$ . The joint density of a random vector  $\xi$  given a GMM  $\mathscr{G}$  is defined by:

$$p(\xi = \hat{\xi}|\mathscr{G}) = \sum_{i=1}^{K} h_i \mathscr{N}(\hat{\xi}; \mu_i, \Sigma_i)$$
(3.1)

$$\mathcal{N}(\hat{\xi};\mu_i,\Sigma_i) = \frac{1}{\sqrt{(2\pi)^N |\Sigma_i|}} \exp\left(-\frac{1}{2}\left((\hat{\xi}-\mu_i)^T \Sigma_i^{-1}(\hat{\xi}-\mu_i)\right)\right) \quad (3.2)$$

where  $h_i$  is the a priori probability of the component *i* of the GMM,  $\mu_i$  is the mean vector of the *i*th Gaussian component, and  $\Sigma_i$  is its covariance matrix; the integer *N* is the dimension of the data-space.

The data-sets  $D_{\mathscr{L}}$  and  $D_{\mathscr{F}}$  have been encoded in two different GMMs,  $\mathscr{G}_{\mathscr{L}}$ and  $\mathscr{G}_{\mathscr{F}}$ . The idea is to see each point of a data-set  $D_{j \in \{\mathscr{L}, \mathscr{F}\}}$  as the realization of a random vector  $\xi$  whose probability density function is given by  $p(\xi | \mathscr{G}_j)$ . The parameters of this probability density function are estimated using the Expectation Maximization algorithm, which maximizes the likelihood of the model:

$$\mathscr{G}_{j} = \arg \max_{\mathscr{G}_{j}} L(\mathscr{G}_{j}|D_{j}) = \prod_{\xi \in D_{j}} p(\xi|\mathscr{G}_{j})$$
(3.3)

The two GMMs  $\mathscr{G}_{\mathscr{L}}$  and  $\mathscr{G}_{\mathscr{F}}$  are then mixed to build a single GMM with two subsets of components associated with both demonstrated role distributions.

Note that this way of encoding the data is much more compact than storing the whole data-set. The components of a GMM give local information about the characteristics of the task: local correlations between variables, and constraints (areas with low variability). Moreover, from a limited set of samples, we construct a model that is defined on the *whole* data space. This is very useful to allow the reproduction of the task in unlearned situations, *i.e* when some of the variables take values that were never demonstrated.

A GMM can be graphically represented by plotting each of the gaussian components of the model. Each component is represented by a set of points that share the same value of the probability density function associated with the component. Therefore, each component is represented by an ellipse in 2D, and by an ellipsoid in 3D. The coordinates of the center of the ellipse/ellipsoid are given by the mean vector of the component. The orientation of the ellipse or ellipsoid depends on the covariance matrix of the component. The GMM encoding the demonstrated task is depicted on figure 3.5. The subsets of components corresponding to the demonstrated role distributions are represented in two different colors.

## 3.3.2 Probabilistic reconstruction of the task

In the previous paragraph, we explained how GMMs can be used to encode the characteristics of the task. GMMs are an approximation of the density function of the data recorded during demonstrations of the task. Therefore, we implicitely define the task as the realization of a random vector with a given probability density function that we estimate using our GMM.

This interpretation is useful to consider the reproduction of the task. When reproducing the task, some of the variables (the inputs) are known, while the outputs are to be computed so as to produce the desired behavior. If we consider a task encoded through a GMM, it is possible, given the values of the input variables, to compute *the conditional probability density function* of the random vector composed of the output variables. An approach to reproduce the task is then to compute the conditional expected value of these output variables and use this value as an output to produce the desired behavior. In simpler terms, we compute *the expected value of the outputs, given the inputs, and given a probability density function*.

Let us state this formally. Let  $\mathscr{G}$  be a GMM defining the probability density function of a random vector  $\xi$  and encoding a given task. Let us decompose the vector  $\xi$  in a vector of input variables  $\xi^I$  whose value is given when reproducing the task, and a vector of output variables  $\xi^O$  whose value is unknown during the reproduction. To reproduce the task, we compute a command vector  $\hat{\xi}^O$  as:

$$\hat{\xi}^{O} = E\left(p(\xi^{O}|\xi^{I},\mathscr{G})\right) \tag{3.4}$$

where  $E\left(p(\xi^{O}|\xi^{I},\mathscr{G})\right)$  is the conditional expected value of the random vector  $\xi^{O}$ 



Figure 3.5 – Gaussian Mixture Models encoding the task dynamics. The red ellipsoids represent the components of the GMM  $\mathscr{G}_{\mathscr{L}}$  encoding the task dynamics with the "leader robot-follower human" role distribution. The green ellipsoids represent the components of the GMM  $\mathscr{G}_{\mathscr{F}}$  associated with the "follower robot-leader human" demonstrations.

given the value of the input random vector  $\xi^I$  and the GMM  $\mathscr{G}$ . This process is called *Gaussian Mixture Regression* (GMR)<sup>2</sup>.

The conditional expected value of the output vector is computed from the conditional distribution of the vector. If we split the random vector  $\xi$  into the output vector  $\xi^{O}$  and the input vector  $\xi^{I}$ , then for each component *i* of the GMM, the mean vector  $\mu_{i}$  and covariance matrix  $\Sigma_{i}$  have the following structure:

$$\mu_{i} = \begin{bmatrix} \mu_{i}^{I} \\ \mu_{i}^{O} \end{bmatrix}, \Sigma_{i} = \begin{bmatrix} \Sigma_{i}^{I} \Sigma_{i}^{IO} \\ \Sigma_{i}^{OI} \Sigma_{i}^{O} \end{bmatrix}$$
(3.5)

The conditional probability density function of the output vector given the input vector and the model is then also a mixture of gaussians [10,96]:

$$p(\xi^{O}|\xi^{I},\mathscr{G}) = \sum_{i=1}^{K} h_{i}(\xi^{I}) \mathscr{N}(\xi^{O};\hat{\mu}_{i},\hat{\Sigma}_{i})$$
(3.6)

$$h_i(\xi^I) = \frac{\mathscr{N}(\xi^I; \mu^I_i, \Sigma^I_i)}{\sum_{i=1}^K \mathscr{N}(\xi^I; \mu^I_i, \Sigma^I_i)}$$
(3.7)

$$\hat{\mu}_{i} = \mu_{i}^{O} + \Sigma_{i}^{OI} (\Sigma_{i}^{I})^{-1} (\xi^{I} - \mu_{i}^{I})$$
(3.8)

$$\hat{\Sigma}_{i} = \Sigma_{i}^{O} - \Sigma_{i}^{OI} \left(\Sigma_{i}^{I}\right)^{-1} \Sigma_{i}^{IO}$$
(3.9)

The conditional expectancy of the output vector given a value of the input vector and the model is thus obtained, from the linearity of the expectancy, by:

$$\hat{\xi}^{O} = E\left(p(\xi^{O}|\xi^{I},\mathscr{G})\right) = \sum_{i=1}^{K} h_{i}(\xi^{I}) \left(\mu_{i}^{O} + \Sigma_{i}^{OI}(\Sigma_{i}^{I})^{-1}(\xi^{I} - \mu_{i}^{I})\right)$$
(3.10)

The figure 3.6 illustrates the GMR process with a 2-dimensional input vector  $\xi^I = [\xi_1^I \xi_2^I]^T$  and a 1-dimensional output  $\xi^O$ . The GMM has only one components and gives the joint distribution  $p(\xi^I, \xi^O)$ . The red plane *S* represents the subspace of points whose altitude is given by the conditional expectancy of the output variable, given the inputs. The different conditional distributions are represented on the picture by ellipses (when only one input variable is given) or by a segment (when both input values are given).

When the GMM has more than one component, the output of the GMR  $E(p(\xi^O|\xi^I))$  can be obtained by computing  $E(p_i(\xi^O|\xi^I))$  for each component *i* of the GMM taken separately, and then weighting the results with the function  $h_i$  given in equation (3.7). This is illustrated on figure 3.7

## 3.3.3 Control scheme

As mentionned earlier in the chapter (see paragraph 3.2.3), the leader and follower data-sets are separated along the force axis. Therefore, if we consider building a controller to reproduce the tasks from these demonstrations, it is natural to consider force as an input to the system. It is also natural to consider the velocity of the gripper as an output of the controller rather than position, so that for a given force at a given position, a reference velocity should be computed.

<sup>&</sup>lt;sup>2</sup>Note that this approach is not specific to random vectors with Normal distributions; it is a possible approach whatever the distribution whose conditional distribution exists. An example with Bernoulli Mixture Models is cited in [10].



Figure 3.6 – Illustration of the GMR process. The red plane represents the image of the input space through the GMR process. The orientation of this plane gives information on how the reconstructed output  $E(p(\xi^{O}|\xi^{I}))$  will vary with the values of the inputs  $\xi_{1}^{I}$  and  $I\xi_{2}^{I}$ .



Figure 3.7 – Illustration of the GMR process with two gaussian components. The surface represents the image of the input space through the GMR process. The blue (respectively red) part shows the area where the blue (respectively red) component has more effect on the value of  $E(p(\xi^{O}|\xi^{I}))$ . In the blue (red) area, the surface is almost the image plane obtained with only the blue (red) component such as in figure 3.6. Between these areas, the colors are mixed, since both components will have influence on the output. The figure shows how the blue and red components are used to compute  $E_{blue}(p(\xi^{O}|\xi^{I}))$  and  $E_{red}(p(\xi^{O}|\xi^{I}))$ , which are located on the cross-ruled planes. These values are weighted to obtain  $E(p(\xi^{O}|\xi^{I}))$ , located on the surface S.

One approach to compute this reference velocity is to perform a GMR and compute the reference velocity as  $\dot{x}^* = \hat{\xi}^O = E(p(\dot{x}|x, f))$ . The reference velocity is then sent to some low-level Cartesian velocity controller.

A more sophisticated solution, proposed in [11] is to build a controller that will at the same time follow the trend of the motion given by  $E(p(\dot{x}|x, f))$ , while trying to stay in the learned domain, *i.e.* keep the gripper's position x close to the demonstrated values for a given gripper's velocity  $\dot{x}$  and a given sensed force f. The resulting controller is then:

$$\begin{aligned} \ddot{x}^* &= \kappa^{\gamma} (\dot{x} - \dot{x}) + \kappa^{\mathscr{P}} (\dot{x} - x) \\ \dot{\hat{x}} &= E(p(\dot{x}|x, f)) \\ \dot{x} &= E(p(x|f, \dot{x})) \end{aligned}$$
(3.11)

where  $\kappa^{\gamma}$  and  $\kappa^{\mathscr{P}}$  are positive gains. The reference acceleration  $\ddot{x}^*$  is then integrated to produce the reference velocity  $\dot{x}^*$ . More details about this controller can be found in [11]. During the reproduction experiments described in the next section, we used a very small position gain compared to the velocity gain (namely  $\kappa^{\mathscr{P}} \ll \kappa^{\gamma}$ ). Equation (3.11) thus practically reduced to:

$$\ddot{x}^* \simeq \kappa^{\mathscr{V}}(\hat{\dot{x}} - \dot{x}) \tag{3.12}$$

Note that for a constant reference velocity  $\hat{x}$ , the actual velocity of the robot's gripper  $\dot{x}$  will exponentially converge to  $\hat{x}$ . If the gain  $\kappa^{\psi}$  is large, the gripper's velocity will track the reference velocity with a small error, for the derivative of the reference velocity is not taken into account. We will from now on assume that equation (3.12) is a perfect equality, and that the reference velocity  $\hat{x} = E(p(\dot{x}|x, f))$  is perfectly tracked. The next section presents and analyzes the results of the reproduction experiments.

## **3.4** Task reproduction

In this section, we evaluate the usability of the proposed method and its robustness. We discuss experimental results obtained during reproduction attempts of the task. We also discuss how these results support the continuous role distribution model proposed in the previous chapter.

## **3.4.1 Reproduction setup**

The setup used for the reproduction of the demonstrated task was the same as for the teaching phase, except that the robot was acting autonomously, instead of being teleoperated. A reference velocity was computed at each control iteration using Eq (3.11). Each experiment lasted less than 5 seconds, time after which the robot came back to the initial position.

Different subjects were asked to lift the object together with the robot. The subjects were not given any specific instructions apart that they needed to lift the object naturally together with the robotic partner. Figure 3.8 shows an autonomous replication of a beam lifting task by HRP-2 jointly with a human task-partner.

## 3.4.2 Preliminary analysis

Several reproductions of the task were performed, involving 13 subjects (including the teacher). Only once, the object was not lifted and stayed at the starting point. A preliminary study was conducted by analyzing 71 trials among the whole set of reproductions, involving 7 of the 13 subjects (the teacher was not part of these 7 subjects). The velocity of the robot's gripper was plotted against the sensed force at the robot's wrist for each experiment. The resulting curve was compared to the GMM encoding the task. Among the 71 trials, 38 were considered successful: the subjects applied forces within the demonstrated range and the lifting motion was smooth.



Figure 3.8 – Autonomous replication of a lifting task by HRP-2 jointly with a human partner.

Most of the 33 failed attempts corresponded to the cases where the object could be lifted, but where the subject applied large forces on the object, such as shown on figure 3.9.



Figure 3.9 – Failed reproduction attempts. The object was successfully lifted, but the dynamics of the motion is not the demonstrated one: the human partner applied very large forces on the object, which we interpret as an unsatisfied will to increase the pace.

From now on, we will study and comment only the 38 trials that were considered successful. We were interested in observing how the subject and the robot would behave if we did not impose any role during the reproduction. Since we use a GMM composed from both leader and follower models, we expected that the robot would be able to act either as a leader or as a follower, depending on the preference of the human partner. In this preliminary analysis, the behavior of the robot is determined according to the forces applied by the partner. If the partner applies positive forces, the follower part of the GMM will have a stronger influence on the motion of the robot than the leader part, and conversely, negative forces will result in a stronger influence of the leader part.



Figure 3.10 – *Reproduction trials where the dynamics of the motion is close to the demonstrated "human leader-robot follower" task dynamics.* 

Among the 38 successful trials, the dynamics of the motion was most of the time close to the demonstrated "human leader-robot follower" task dynamics (23 trials), as shown on figure 3.10. The opposite case, illustrated on fig. 3.11 appeared only once.



Figure 3.11 – *Reproduction trial where the dynamics of the motion is close to the demonstrated "human follower-robot leader" task dynamics.* 

During the reproductions of the task, we were willing to notice the switching between the follower and leader behaviors in some experiments. Two extreme, complementary role distributions have been demonstrated to the robot, resulting in two different task dynamics, encoded in two GMMs. During the reproductions of the task, those two GMMs were used jointly, as a single GMM. By varying the applied force, the partner can place the motion of the gripper under the influence of either the "leader" components of the GMM, or of the "follower" components. This was possible because both role distributions are encoded in two GMMs that are clearly separated along the force axis. The consequence is that the robot can switch between the two demonstrated task dynamics according to the forces applied by the human.

The figures 3.12 illustrates 3 cases among 14 observed trials where such a switching between these two dynamics occured: the task dynamics switched from the demonstrated "leader robot-follower human" to the demonstrated "follower robot-leader human" dynamics. Figure 3.13 shows a case where the dynamics switched twice. These switchings suggest that role switching as defined in the extended leader-follower model presented in chapter 2 can occur during collaborative tasks, and can occur naturally even if one of the partner is a robotic system, if this system is given the ability to switch between both behaviors. It also suggests that the presented probabilistic framework can be used in a programming-by-demonstration setup to avoid explicitly devising the leader and follower behaviors and switching strategies.



Figure 3.12 – Reproduction trials where the dynamics of the motion switched from the "robot leader" to the "robot follower" dynamics.

## **3.4.3** Quantitative analysis

The previous paragraph presented a qualitative analysis of the experimental results, where the reproduction data was superimposed to the GMM to conclude about the dynamics of the task and the role distribution. Besides, unsuccessful trials were sorted out by looking at the range of applied forces: trials where forces above 5N were applied were considered as unsuccessful, for the magnitude of the demonstrated forces was below this value.



Figure 3.13 – *Reproduction trials where the dynamics of the motion switched twice between both demonstrated dynamics.* 

In this paragraph, we attempt to quantitatively evaluate the reproduction results, to conclude more precisely about the usability of the Programming by Demonstration framework presented in this chapter for collaborative tasks. The results are also fitted with our extended leader-follower model and we investigate to what extend the experimental results allow us to conclude about role switching.

This paragraph analyses the experimental results of 70 trials performed by 6 subjects: one of the subjects of the preliminary study was involved in only 1 trial. This subject was excluded from the quantitative study, so that the 6 subjects performed about 10 trials.

#### Image of the force-position plane through GMR

The learning phase provided us with a GMM for the lifting task. This GMM is used to perform a GMR and compute a reference velocity for the gripper of the robot, from its position and the force sensed at the wrist. To understand the behavior of the robot, it is useful to have an idea about what the output velocity will be, depending on the force and position variables. This can be done, as shown in the previous section, by drawing the surface  $\{[a \in \mathcal{R}; b \in \mathcal{R}; E(p(\dot{x}|x=a, f=b))]^T\}$ , as shown on figure 3.14.

Assuming that equation (3.12) holds, and that the reference velocity is perfectly tracked by the gripper, the data-points  $\{\xi = [x f \dot{x}]^T\}$  will be located on this surface, that we will refer to as the "image of the force-position plane through GMR". Note that the point  $\xi(t)$  representing the coordinates of the robot's gripper in the data-space at time *t* will not move freely on this surface: while the motion along the force axis depends on the force applied by the partner, the motion of the point along the position axis will depend on the velocity of the robot's gripper, *i.e.* on the altitude of point  $\xi(t)$ .



Figure 3.14 – Image of the force-position plane through the GMR process. The ellipsoids represent the components of the GMM, and the surface shows the velocity output  $E(p(\dot{x}|x, f))$  for each value of the position x and force f. Orange areas correspond to the higher velocities, and blue ones to the lower velocities.

#### Successful trials

In the preliminary analysis, the success rate was determined according to the force range. The intuition is that if subjects pulled too hard, they probably felt like the robot did not adapt properly. In this paragraph, we adopt another method and try to quantitatively establish a success rate. This can be done by comparing the task dynamics to the demonstrated one, which will take the sensed force into account. It is also possible to consider only the kinematics of the task: the task will be considered successful if the kinematics of the motion is close to the demonstrated one, whatever forces the subject had to apply. Here, we choose to consider the applied forces.

We need to evaluate, for each trial, how close the dynamics of the motion is to the demonstrated one. We propose to evaluate the log-likelihood of each data-point of the trial given the GMM, and to compare it to the maximum log-likelihood of the image of the force-position plane through the GMR process. We first compute a threshold  $l_{\text{lim}}$  as:

$$l_{\text{lim}} = l_{\text{min}} + 0.75 \ (l_{\text{max}} - l_{\text{min}})$$

$$l_{\text{min}} = \min_{\xi \in \mathscr{V}} \log(L(\xi, \mathscr{G}))$$

$$l_{\text{max}} = \max_{\xi \in \mathscr{V}} \log(L(\xi, \mathscr{G}))$$
(3.13)

where  $L(\xi, \mathscr{G})$  is the likelihood of the data-point  $\xi$  given the GMM  $\mathscr{G}$ .  $l_{\lim}$  is

thus at 75% the the interval  $[l_{\min}; l_{\max}]$ , where  $l_{\min}$  and  $l_{\max}$  are the log-likelihood extrema observed in the volume  $\mathscr{V}$ . The volume considered for this analysis was  $x \in [-0.1; 0.4]$  (in meter) and  $f \in [-20; 5]$  (in Newton) and  $\dot{x} \in [-0.1; 0.2]$  (in meter per second).

For a given trial, the log-likelihood of each point is computed and compared to  $l_{lim}$ . If all points have a log-likelihood over this value, the trial is considered successful. By applying this criterion to the 70 trials, 38 were considered successful. The success ratio is thus similar to the one obtained in the preliminary study. Figure 3.15 shows the points on the image of the force-position plane that satisfy the criterion. Figure 3.16 shows the successful and failed trials.



Figure 3.15 – Image of the force-position plane through the GMR process. The area in light gray is the area where the success criterion is satisfied, i.e. points  $\xi$  such that  $\log(L(\xi, \mathscr{G})) > l_{lim}$ .



Figure 3.16 – Left: successful trials. The data-points have been projected on the positionvelocity plane. Right: failed trials. By comparing these pictures, we notice that successful trials generally followed the shape suggested by the ellipsoids, and reached a final position that was closer to the demonstrated one, above the final position of the failed trials.



Figure 3.17 - In red: area where the robot will behave mostly according to the "leader robot-follower human" task dynamics. In green: area where the robot will behave mostly according to the "follower robot-leader human" task dynamics. The transition between these two areas is smooth.



Figure 3.18 – Trial where the task dynamics switched between both role distributions. The switching is more apparent on the bottom figure, where the data-points are plotted over the leader and follower areas.



Figure 3.19 – Velocity of the gripper of the robot and value of the homotopy parameter during one trial. The smooth transition between both task dynamics can be seen from the continuous transition of the homotopy parameter from 1 to 0.

#### Role distribution and behavior switching

In the preliminary analysis, the role distribution was estimated by looking at the applied forces and by projecting the reproduction data and the GMM in the force-velocity plane. The robot was considered follower when the curves were over the green ellipses (the components of the GMM that correspond to the "follower robot-leader human" demonstrations), and considered as a leader when the curves were over the red ellipses. The intuition behind this criterion is that the influence of the components of the GMM are weighted according to the relative values of the probability density functions of the components of the GMM. Since for a given component, this value increases when going towards the mean vector of the component (towards the center of the ellipse), this component will have more influence in the neighbourhood of the ellipse.

Let us state this idea more formally and analyze the experimental results quantitatively with regard to role distribution among the partners. For this purpose, let us first define the set  $I_{\mathscr{L}}$  of components of the GMM  $\mathscr{G}_{\mathscr{L}}$ , which encodes the "leader robot-follower human" task dynamics. Likewise,  $I_{\mathscr{F}}$  is the set of components of the GMM  $\mathscr{G}_{\mathscr{F}}$ . We also define  $I = I_{\mathscr{L}} \cup I_{\mathscr{F}}$ . The *i*-th component of the GMM  $\mathscr{G}$  will be denoted  $\mathscr{G}_i$ . Let us define the function  $\alpha$  as:

$$\alpha(\xi) = \sum_{i \in I_{\mathscr{L}}} \frac{p(\Xi = \xi | \mathscr{G}_i)}{p(\Xi = \xi | \mathscr{G})}, \quad \xi = \begin{bmatrix} x \\ f \\ \dot{x} \end{bmatrix}$$
(3.14)

Then for any data-point  $\xi$ , we have  $\alpha(\xi) \in ]0; 1[$ . If  $\alpha \to 1$ , the output velocity  $E(\dot{x}|x, f)$  computed through GMR will almost only depend on the components of GMM  $\mathscr{G}_{\mathscr{L}}$ , *i.e.* according to a "leader robot-follower human" task dynamics. The opposite case occurs when  $\alpha \to 0$ . Therefore, the function  $\alpha$  characterizes a homotopy between a leader controller and a follower controller. The leader and follower areas are depicted on figure 3.17.

The figure 3.17 shows that for a given position, the behavior of the robot (leader or follower) will depend on the force applied by the human partner. Thus, we

have *implicitely designed a behavior switching strategy*, based on the force applied by the partner. Though qualitatively, this strategy is straightforward (the robot becomes a follower if the human pulls the object, which means the human is trying to go up faster than the robot), the exact value that separates both behaviors is not easily determined and is likely to depend on the mass of the object. Therefore, transforming this intuitive idea into a mapping from force to a homotopy parameter is not an easy task. The Programming by Demonstration method used here allows us to implicitely determine both the qualitative and quantitative aspects of the role switching strategy. Note, also, that in our case the swithcing strategy can be seen as:

- ▷ Detecting the role of the human through the applied force;
- ▷ Adapt a complementary role.

The figure 3.18 shows a case where the role distribution switched from "leader robot-follower human" to "follower robot-leader human". The figure 3.19 shows the velocity of the robot's gripper and the evolution of the function  $\alpha$  along the task for this trial.

## 3.5 Conclusion

In this chapter, we have applied the Programming by Demonstration paradigm to collaborative lifting tasks. We restricted ourselves to a one degree-of-freedom description of such tasks. Based on this description, variables that describe the task were recorded during demonstrations of the task and the relevant features of the tasks were encoded in a Gaussian Mixture Model. This GMM was used in the reproduction phase to reconstruct the data using a Gaussian Mixture Regression process. The task could be successfully demonstrated using a teleoperation setup with kinesthetic feedback, on the humanoid robot HRP-2. The task was then successfully reproduced with several operators in an autonomous way.

From the learning point of view, the work performed here is a new application of Programming by Demonstration. Much work can still be done to extend the proposed framework to encompass different objects and investigate the variations between different human partners. We hereby assumed that even if the task had been demonstrated in collaboration with only two different partners, the robot would be able to perform the task with any operator, since human operators would be able to adapt to the dynamics of the motion of the robot.

In the perspective of the extended leader-follower model presented in this thesis, this chapter presents a method based on the Programming by Demonstration paradigm and a probabilistic framework to implicitely design role switching strategies for a given collaborative task. The method consists in demonstrating how the task is performed with two extreme role distributions and to encode these dynamics separately in two GMMs. If the two resulting models are well separated along one dimension, then a homotopy is implicitely defined by considering the relative influence of the leader and follower parts of the GMM during the reproduction of the task through GMR.

Experiments conducted with the humanoid robot HRP-2 assess the usability of the proposed method. It also allowed us to highlight switchings between both demonstrated role distributions during the reproductions of the collaborative lifting task. This result supports the model presented in the previous chapter, according to which partners in a collaborative task smoothly switch from a leader to a follower role.

This chapter concludes the first part of this thesis. In this part, we studied existing models for collaborative tasks, and proposed a theoretical framework to describe dyadic collaborative tasks. The main idea is to extend the classical leaderfollower approach to continuous, time-varying role distributions among the partners, thus giving an equal responsibility to both partners to perform the task. Different ways of adjusting the role distribution were examined. Finally, a study of haptic communication between human partners during collaborative lifting tasks using a Programming by Demonstration framework allowed us to demonstrate collaborative tasks to a robot so that it could successfully reproduce them, and to support our continuous switching model. The next part of the thesis is more focused on implementation aspects of collaborative tasks on robotic systems.

# Collaborative tasks via telepresence

## Contents

4.1	Teleoperation of humanoid robots: a brief review 83			
4.2	The telepresence setup			
4.3	Taxonomies of coupling85			
4.4	Whole	-body control and extension to wide-area motions 87		
	4.4.1	Whole-body motion control    88		
	4.4.2	Force-based control		
	4.4.3	Wide-area motions		
4.5	Experie	ments		
	4.5.1	Setup		
	4.5.2	Control		
	4.5.3	Results and Discussion		
4.6	Conclu	sion $\ldots$ $\ldots$ $\ldots$ $\ldots$ $9\epsilon$		

IN this chapter, we study the realization of collaborative tasks between two distant human operators. The work reported here has been conducted jointly with Martin Buss, Sandra Hirche, Inga Krause, Angelika Peer, Thomas Schauß, Carolina Weber (in alphabetical order) from the Institute of Automatic Control, at the Technische Universität München.

In this chapter, a scenario is proposed whereby a human operator takes control of a humanoid robot using a mobile telepresence setup to perform some task with another operator at a distance. The originality of this scenario lies in the second human operator being on-site. The human partners must then perform a physical collaborative task under reduced transparency, limited feedback and time delay. Another challenge is the whole-body control of the humanoid robot, and the coupling between two platforms with different dynamics: one is a haptic interface mounted on an omnidirectional mobile base, and moves continuously; the other is a humanoid robot, whose locomotion is realized by stepping.

After a more detailed presentation of the application and a review of the relevant literature, we investigate how such an experimental setup can be established, and discuss about the possible coupling schemes between the telepresence setup (*i.e.* the master device, to adopt a conventional terminology) and the distant humanoid robot (*i.e.* the teleoperator, or slave). We then extend the setup to wide-area manipulation, relying on the mobility of the telepresence setup and the gait capabilities of the teleoperated humanoid robot. The main components used for the whole-body control of the humanoid platform are described. Some of these components will be described in more details in the next chapter, where we present an experimental setup that uses the same components to perform collaborative tasks in an autonomous way with a human operator.

## 4.1 Teleoperation of humanoid robots: a brief review

Teleoperation allows a human operator to perform tasks at a distant location, or in hostile environments. It has required a substantial amount of work for several decades to solve the problems due to time delay, limited feedback, or to increase user-friendliness.

The study of bilateral teleoperation has focused on stability and transparency [62, 79]. Stability is probably the most important issue since it puts the safety of the operator at stake. One of the main reasons for instability in force reflecting teleoperation setups is the transmission time delay between the master and slave devices. Unfortunately, it is physically impossible to avoid or compensate for this time delay. Depending on the scale of transmission time delay, teleoperation systems will have to be more or less damped, *i.e.* reduce the bandwidth of the system. It has been shown [2, 79] that the transmission of power variables (forces and velocities) between the local and distant site is non-passive and thus can destabilize the system. A solution to this problem is to transmit wave variables rather than power variables [79].

Virtual Reality-based methods can be employed to enhance teleoperation setups [61]. Visually displaying the interaction forces rather than reflecting them at the master device's actuators avoids the stability issues inherent to bilateral teleoperation with force feedback. Likewise, supervisory control is a modality where the human will issue high-level commands and the low-level motions of the teleoperator will be driven only by local controllers. Obviously, such modalities impair the telepresence feeling, since the master and slave devices are no longer bilaterally coupled. Predictive display has also been proposed to avoid stability issues caused by time-delayed transmission in force reflecting teleoperation setups [5]. To avoid the so-called "Move and Wait" strategy adopted by operators in face of significant time delay, a model of the robot is super-imposed on the video feedback of the teleoperator. The motion of the virtual slave is computed locally using a model of the robot and the environment, or only with a model of the robot [51]. This approach can be applied to the display of any other artefact to improve the task performance, such as virtual beams, that help the operator estimate the distance between the teleoperator and the environment [109].

The teleoperation of humanoid platforms is a very appealing concept, since the teleoperator can really be seen as a projection of the body of the operator. Nevertheless, several challenges arise from the specificity of humanoid robots. These devices are often highly redundant, and always comprise many degrees of freedom, and obviously several end-effectors. As a consequence, the cognitive load for the operator increases compared to manipulator arms. Bilaterally coupling the whole platform can be done using master devices that can simultaneously control several aspects of the remote task in a user-friendly way, such as [106], or the "marionette" system presented in [108].

When teleoperating humanoid robots, all the degrees of freedom are generally not coupled to the master device, for several reasons. First, it would represent a high cognitive load for the operator, unless a very human friendly system is used which does not require the human to *think* about operating the robot, *i.e.* her/his motions would be naturally mapped on the slave device. But more importantly, humanoid platforms, as walking systems, must keep their balance. This specific constraint can be difficult to enforce for the human operator, especially with limited, time-delayed feedback. This is even more difficult if the master and slave devices are not of the same type (*e.g.* joysticks on the master side, and a humanoid robot on the slave side). A solution is to delegate part of the control to the teleoperator. This is the shared control paradigm, and has proven to be effective in the case of complex tasks or manipulators, such as dexterous tasks performed by hand manipulators [34], or peg-in-hole tasks with full force-torque reflection [37]. Shared control has been applied to the control of humanoid robots for the generation of whole-body motions and balance control, where the master devices are joysticks with different control modes corresponding to different effectors and tasks [77].

A setup that uses both shared control phases and supervisory control phases has been proposed in [119]. A humanoid robot is teleoperated to perform outdoor tasks. The scenario comprises a walking phase, a phase where the robot steps into a lift truck and an operating phase. The remote control device includes two masterarms for the control of the robot's arms, master-foot devices to teleoperate the legs without force feedback, and a device to control the head. Video feedback is displayed on a screen. Switches and voice commands are used to trigger the execution of pre-programmed tasks. Shared control has been used to generate whole body motions while the operator controls specific sub-tasks. It has been used specifically for collision avoidance: the operator commanded the positions of the arms, and the elbow was locally controlled to avoid self-collision. Walking and sitting into the truck was achieved using a preprogrammed walking pattern and motion. Vision was used to compute a collision-free sitting motion, with the assistance of the operator, to take a screenshot of landmarks installed in the backhoe. In this case, the operator and the teleoperator are somehow collaborating to perform a task (a sitting motion), but they act using two different modalities.

Collaborative teleoperation has been relatively little studied. Glassmire *et al.* presented an experiment involving a collaborative task between a human and the humanoid robot Robonaut [33]. In this setup, Robonaut is grounded and teleoperated by another human operator. The goal of the experiment was to evaluate the importance of force feedback in teleoperation setups. The study reveals that force feedback reduces peak forces but increases the time integral of the sensed force increases. The completion time was similar in both cases.

## 4.2 The telepresence setup

Our goal is to allow a human operator to participate to a collaborative task that takes place on a distant site, by the mean of a telepresence setup. Thanks to this setup, the operator has both haptic and visual feedbacks with a subjective point of view, and can take part in the distant task as if s/he was on-site. On the distant site, a physical task is realized by a human operator in collaboration with the teleoperated humanoid robot.

We consider long-distance teleoperation, and thus a transmission time delay between the local and distant sites. We make the following assumptions about the setup:

- ▷ The master device is an holonomous mobile haptic interface [113].
- ▷ The operator is immersed and has a subjective visual feedback.
- ▷ The teleoperator is the HRP-2 [63] humanoid robot.
- ▷ A tracking system on the master site can provide the attitude of the operator.



Figure 4.1 – Telepresence setup for distant collaborative tasks.

Fig. 4.1 summarizes these assumptions in a schematic view. On this figure, the tracking system appears as a camera, but it can be actually any device capable of estimating the attitude of the master operator.

## 4.3 Taxonomies of coupling

Assuming the setup described in the previous section, several coupling schemes between the master platform and the humanoid teleoperator can be proposed. This section discusses different couplings.

The simplest coupling consists in bilaterally coupling the end-effectors of both robotic systems, so that the interaction on the distant site is reflected on the master site. This way, the operator can feel as if it was directly interacting with the remote environment. This is illustrated on fig. 4.2.



Figure 4.2 – End-effector coupling.

The motions of both platforms are controlled independently; the platforms are just seen as tools to allow wide-range motions. While this setup is satisfactory on the master site, discrepancies are likely to occur between the postures of the operator and of the teleoperator. In some cases, this can be very disturbing for the distant human partner, since s/he might not interpret the intentions of the master operator properly. Figure 4.3 illustrates this potential issue. It appears clearly on this figure that the posture and attitude of the operator (*i.e.* her/his relative position with respect to the manipulated object) are important pieces of information to convey to the distant site.



Figure 4.3 – Discrepancies between the operator and teleoperator's posture result in a wrong intention estimation by the distant human partner.

To ensure the consistency between the postures on the local and distant sites, we can track each operator's posture and map it onto the master and slave devices. This illustrated on top of fig. 4.4. Since on the master side, we intend to reflect the interaction between the teleoperator and the manipulated object, the posture and the attitude of the object should be mapped onto the master device. Another solution is to manipulate the same object on both sides, and to reflect the posture of the distant human only onto the master device, as depicted on the bottom part of fig. 4.4. It is however impractical to manipulate the same object on both sides. Moreover, these bilateral posture couplings require a tracking system on *both* sides.



Figure 4.4 - Top: the teleoperator mimics the operator's posture; the posture and attitude of both the distant human and the object are mapped onto the master device. Bottom: the same object is manipulated on both sides, and only the posture of the distant human is mapped onto the mobile haptic interface.

In our particular case, we assume that the human on the local site is immersed, and has a subjective feedback of the remote environment. As a result, reflecting the posture and attitude of the distant human and object is not necessary, since the master operator can see them thanks to the immersive visual feedback. It is more important to accurately reflect the interaction between the end-effector of the teleoperator and the remote environment (mainly the manipulated object). Hence, we propose to enhance the end-effector coupling scheme depicted on fig. 4.2 with a *posture reflecting coupling*: the end-effectors of both platforms are bilaterally coupled, and the humanoid teleoperator uses its redundancy to mimic at best the posture and attitude of the human on master site. The posture and attitude of the operator is obtained *via* the tracking system on the local site. This coupling scheme is described on fig. 4.5.



Figure 4.5 – End-effector coupling with posture reflection.

This coupling scheme presents several advantages. First, it reduces the risk of misunderstanding of the operator's intentions. Second, the end-effectors can be coupled using classical techniques of bilateral teleoperation, which eases the stability and transparency analysis of the haptic interaction between the distant partners. Finally, this scheme can be adapted to several levels of shared control. When walking, the gait of the operator on master site and of the teleoperator can be either totally decoupled, or the operator can control the gait of the teleoperator using the tracking system, for a higher sense of presence.

Although the posture reflection scheme offers great possibilities, this chapter will discuss the implementation of the simpler end-effector coupling scheme, which is the basis upon which we can build a posture reflection scheme in the future. The remainder of this chapter presents a deeper analysis of the end-effector coupling scheme.

## 4.4 Whole-body control and extension to wide-area motions

When coupling two complex robotic systems to implement a teleoperation setup, several coupling schemes can be selected. In our case, we address the bilateral coupling of a holonomic mobile platform on which two haptic arms are mounted on the local site, and a full-sized, biped humanoid robot on the distant site. The robot is holding an object that it manipulates together with an onsite human collaborator. Our goal is to immerse the local operator into the distant environment, so that s/he feels the body of the humanoid robot as her/his own body, and so that s/he feels like s/he is performing the task at the remote location. For this purpose, reproducing the interaction between the humanoid robot and the manipulated object is very important. If this interaction is reproduced with enough fidelity on the master site, the operator will feel like s/he is actually grasping the object.

Although three- and four-channel architectures have been studied in the past,

two-channel architectures are commonly used and very intuitive. When transmitting power variables, four two-channel couplings are possible:

- Position-Position (P-P): the position of each device is sent as a command to the other device;
- Position-Force (P-F): the position of the master device is sent as a reference to the slave device, and the forces measured at the slave site are reflected at the master site;
- Force-Position (F-P): the forces applied by the operator on the master are sent as a reference to the slave device, while the position of the master device tracks the position of the slave device;
- ▷ Force-Force (F-F): the forces applied by the slave and master devices on their environment are coupled.

In this chapter, an F-P architecture is chosen to couple the end-effectors. The reasons for this choice are reported in a paper lead by our colleagues from the Institute of Automatic Control at the Technische Universität München [84]. Part of the work presented in [84] is recalled in the first appendix of this thesis, for the sake of completeness. For a good understanding of the remainder of this chapter, fig. 4.6 shows the main blocks of the F-P control architecture on slave site.



Figure 4.6 – F-P bilateral coupling: local control on the slave site.

This section presents the extension of the end-effector coupling to allow widerange teleoperation. We hereby focus on the control of the slave device: a humanoid robot will be used as a projection of the operator's body into the remote environment. The control of the humanoid platform for wide-area motions involves gait and dynamic balance issues, and the redundancy of the platform calls for appropriate schemes for whole-body control. Details about the control of the master device can be found in [113] and are briefly summarized in [27]. Parts of the components presented here are described in more details in the next chapter.

## 4.4.1 Whole-body motion control

A redundant humanoid robot is used as a slave platform. To take advantage of the redundancy, a task-based control approach is adopted [95, 99]. This approach

allows the decomposition of a complex behavior into a set of *n* tasks described by triplets  $(\mathbf{e_i}, \mathbf{J_i}, \dot{\mathbf{e_i}}^*)$ ,  $i \in \{1..n\}$ , where  $\mathbf{e_i}$  is typically an error between a signal and its desired value  $\mathbf{e_i} = \mathbf{s_i} - \mathbf{s_i^*}$ ,  $\mathbf{J_i} = \frac{\partial \mathbf{e_i}}{\partial \mathbf{q}}$  maps the displacements from the joint space to the task space, and  $\dot{\mathbf{e_i}}^*$  defines the reference behavior of the error function. Typically, imposing  $\dot{\mathbf{e_i}}^* = -\lambda_i \mathbf{e_i}$ , with  $\lambda_i$  a positive gain, will result in an exponential decay of the error with time.

The evolution of each task *i* is then given by

$$\dot{\mathbf{\xi}}_{\mathbf{i}} = \mathbf{J}_{\mathbf{i}} \dot{\mathbf{q}} \tag{4.1}$$

where  $\dot{\mathbf{q}}$  is the joint velocity vector used as a control input on the robot. To obtain the desired behavior  $\dot{\mathbf{e}}_{i}$ , the general solution is given by

$$\dot{\mathbf{q}} = \mathbf{J}_{\mathbf{i}}^{\#} \dot{\mathbf{e}}_{\mathbf{i}} + \mathbf{P}\mathbf{z} \tag{4.2}$$

where  $J_i^{\#}$  is a Pseudoinverse of  $J_i$  [6] and z is any arbitrary vector projected in the null space of the Jacobian  $J_i$ . Thanks to the projection, any value can be chosen, without effect on  $\dot{e}_i$ . It can thus be used to realize a second task. Following this reasoning, a set of *n* tasks can be hierarchically organized such that the first task has the highest priority, and the other tasks will be realized as possible with the guarantee of not hindering the tasks with higher priorities. The control law to ensure the realization of the set of *n* tasks with decreasing priorities can be computed using the following recursive equations [99]:

$$\dot{\mathbf{q}}_{i} = \dot{\mathbf{q}}_{i-1} + (\mathbf{J}_{i}\mathbf{P}_{i-1}^{\mathbf{A}})^{\#}(\dot{\mathbf{e}}_{i} - \mathbf{J}_{i}\dot{\mathbf{q}}_{i-1}), \ i = 1..n$$
 (4.3)

where  $\mathbf{P}_{i}^{A}$  is the projector onto the null-space of the augmented Jacobian  $\mathbf{J}_{i}^{A} = (\mathbf{J}_{1}, \dots, \mathbf{J}_{i}), \mathbf{J}_{i}\mathbf{P}_{i-1}^{A}$  is the limited Jacobian of the task *i* and  $\dot{\mathbf{q}}_{0} = 0$ . The robot joint velocity realizing all the tasks in the stack is  $\dot{\mathbf{q}} = \dot{\mathbf{q}}_{n}$ .

The stack of tasks (SoT) is a structure that orders the tasks that are currently active. Only the tasks in the stack are taken into account in the control law. The task at the bottom level has priority over all the others, and the priority decreases as the stack level increases. Any new task added in the stack does not disturb the tasks already in the stack. A complete description of the Stack of Tasks is given in [71].

### 4.4.2 Force-based control

Using the foregoing control scheme, it is straightforward to realize displacement tasks without contact. However, for force-based control and tasks involving contact, the inverse kinematics-based control does not directly apply. Moreover, the end-effector coupling scheme presented in the previous section requires the end-effector to behave as a desired admittance (or impedance). A solution to implement impedance control (or more generally, force-based control) on torque control robots is to use the *dynamic inverse* as proposed in [60,81].

For position controlled robots, position-based impedance control (the so-called admittance control) must be implemented to display a desired impedance. Admittance control can easily fit the task formalism: the space in which the control is designed is the operational space (6D position) of the contact point denoted by **r**. Admittance control consists in defining a desired dynamic behavior for this point, such as the one of a virtual mass-damping system:
where **M** and **B** are arbitrary mass and damping matrices of the equivalent virtual system, and **f** are forces and torques exerted on the equivalent virtual point. For the *F*-*P* coupling scheme setup, **f** is defined as the sum of the real forces measured at the contact point and the reference forces sent by the master operator.

The equation (4.4) is then integrated to compute a desired velocity  $\dot{\mathbf{r}}$ , which is then used as the desired behavior for a task i:  $\dot{\mathbf{e}}_{i} = \dot{\mathbf{r}}$ . We will see in the next chapter how the Stack of Tasks framework presented in the previous paragraph can be modified to improve the performance of this admittance control scheme.

#### 4.4.3 Wide-area motions

The admittance control tasks allow us to implement a bilateral F-P coupling. To allow wide-area motions, the humanoid robot will also have to walk. To address dynamic gait generation, we used a walking pattern generator based on the preview control and on the Zero-Momentum Point (ZMP) condition, proposed by Kajita [52]. This algorithm uses a preview window to compute the control law from a simplified Linearized Inverted Pendulum Model (LIPM), to compute the current control command from the future state of the system. The core part of this algorithm is to solve a quadratic programming (QP) problem, where the cost function is to minimize the jerk of the LIPM and to follow an ideal ZMP trajectory. From this ZMP trajectory defined by the foot prints given as input, the solution of the QP gives a Center of Mass (CoM) trajectory which is dynamically stable. Finally, the reference trajectories are integrated in the Stack of Tasks.

An appealing solution to apply this algorithm would be to recompute the CoM and foot trajectories for the whole preview window at each control cycle. However, the computation cost to solve the QP problem with the multibody model is too high for the control loop on the teleoperator <sup>1</sup>. Hence the pattern generator is used as a planner which generates a stable motion within a fixed time. Consequently, it is impossible to input any footprint reference within this time window.

The inputs of the pattern generator are desired footprints. From these footprints, CoM and feet trajectories that lead to a stable motion are computed. The problem is now to define the desired footprints. Assuming only the end-effectors of the master and slave devices are coupled, the role of the slave humanoid gait is simply to make possible arbitrarily large motions of the end-effectors. A simple way to achieve this is to have the feet "follow" the hands. Hence, the footprints will be selected to track a desired relative position of the feet with respect to the hands of the robot.

In practice, we used the pattern generator software described in [104]. To account for the preview window, four footprints had to be provided to the algorithm. A new footprint is then added each time one step has been executed by the robot. The outputs of the pattern generator are reference trajectories for the feet and for the CoM, that can be taken into account as top-priority tasks in the Stack of Tasks.

#### 4.5 Experiments

#### 4.5.1 Setup

We implemented the components described in this chapter to perform telepresence collaborative tasks between Munich, Germany, and Tsukuba, Japan.

<sup>&</sup>lt;sup>1</sup>The control loop runs at 200 Hz on the humanoid robot HRP-2.

On the remote side in Tsukuba, Japan, a HRP-2 humanoid robot with 30 actuated degrees of freedom (DOF) is used to mirror the actions of the master operator. It is capable of bipedal locomotion, and is equipped with force sensors at the wrists and at the ankles. Visual feedback is provided by stereovision from two cameras located on the head of the robot. PD joint position controllers run at 1 kHz. Desired positions are provided at 200 Hz to the PD controllers.

Reference positions are provided by an admittance controller, which takes the sum of the environmental forces read at the force sensor on the wrist, and the reference force sent from the master site. Since the PD joint position control loop runs 5 times faster than the admittance controller, and since the PD controller is stiff, the assumption that the teleoperator actually behaves like the imposed admittance is reasonable.

The human-system interface, located in Munich, Germany, consists of two VISHARD7 [83] mounted on a mobile base. The two 7-DOF arms have a humansized workspace and high force output capabilities. The haptic interfaces can perform movements in a workspace similar to that of the human operator at a fast speed. The interfaces were especially designed to be mounted on an omnidirectional, nonholonomic mobile base [113]. The mobile base can perform arbitrarily large movements at a slower speed. The combination is capable of displaying fast movements and high forces in an arbitrarily large workspace. The operator has visual feedback from the remote environment through a Head Mounted Display.

For control, forces and torques from both arms are sent from master to slave, while end-effector positions and orientation of both arms are sent from slave to master. At a transmission rate of 50 Hz using a protocol based on UDP, the packet loss was negligible (< 1%) and the round-trip time between Germany and Japan was approximately T = 280 ms, and almost constant.

The setup used for the experiments is depicted in Fig 4.7.



Figure 4.7 – Photos of the experimental setup. On the left the operator site with the human operator and the mobile haptic interface in Munich is depicted. On the right the teleoperator HRP2 and the human collaborator in Tsukuba are shown.

#### 4.5.2 Control

#### **Bilateral coupling of the end-effectors**

As mentioned earlier in this chapter, an F-P architecture has been implemented. The forces applied by the human operator in Munich are sent to the distant site in Japan, while the positions of the gripper of the teleoperator are sent as a reference for the position of the master platform's grippers. Forces on both sides contribute to the motion of the platform after having been filtered by an admittance. The admittance parameters on master site are  $m_m = 1 \text{ kg}$ ,  $b_m = 200 \text{ Ns/m}$  and  $k_m = 600 \text{ N/m}$  for the translational part, and  $m_m = 0.02 \text{ kg}m^2$ ,  $b_m = 2 \text{ Ns/rad}$  and  $k_m = 20 \text{ Nm/rad}$ .

On the slave side, the admittance controller performance could be improved by setting the virtual mass matrix of the admittance as the real inertia matrix of the robot projected into the operational space of the contact point. The damping was selected to stabilize the coupled system, according to the theoretical study presented in the first section:  $b_s = 200$  Ns/m for the translational part, and  $b_s = 1$ Ns/rad for the rotational part. The details about the admittance controller of the arms of the robot are presented in the next chapter.

#### Whole-body motion control

On the slave side, the Stack of Tasks was running with the following tasks:

- ▷ 1) e<sub>walk</sub> tracks the desired trajectories of the legs provided by the pattern generator. Only the position of the free flying foot is controlled, and the support foot is considered as fully constrained by the contact with the ground. Compared to controlling the absolute positions of both feet, this relaxes 6 DOF of the robot, thus allowing a wider range of motions.
- $\triangleright$  2)  $\mathbf{e_{com}}$  regulates the position of the CoM on the trajectory given by the pattern generator.
- ▷ 3+4)  $\mathbf{e_{zr}}$  and  $\mathbf{e_{zl}}$  regulate the altitude of both right and left hands of the robot to a desired fixed height, in order to simplify the task of the master operator when the robot is walking. The positions of the end-effectors  $\mathbf{r_r}$  and  $\mathbf{r_l}$  are controlled by taking into account only the *z* component of the vector. These tasks are removed when the robot is not walking.
- ▷ 5+6)  $\mathbf{e_{fr}}$  and  $\mathbf{e_{fl}}$  are the admittance control tasks for both right and left hands of the robot. They are defined as in (4.4). The reference force **f** is the sum of real measured forces and reference forces sent by the master. Each of these tasks is a 6 degrees-of-freedom task, but due to 3+4), only 5 degreesof-freedom are expressed during the control.
- ▷ 7) e<sub>head</sub> controls the orientation of the head according to the reference position given by the master operator.
- $\triangleright$  8)  $\mathbf{e_{grip}}$  is finally added to control specifically the gripper aperture to the reference transmitted from the master.

#### Gait

The challenge when controlling a walking humanoid platform to perform a physical collaboration task lies in the fact that interaction forces induced by the task create sudden perturbations on the ZMP of the robot. In order to keep dynamic balance, the footprints would have to be modified within the preview window. The



Figure 4.8 – Cooperative telemanipulation task: a) approach, b) grasp, c) lift, d) put down

technical difficulty to cope with this, is hence to be able to compute a new reference for the CoM and the feet trajectories with modified inputs in the preview window, and/or with a new initial state.

In this work, we have experimentally evaluated whether the humanoid platform could keep its balance while performing a collaborative task. It appears that the commercial stabilizer provided with the control software of the HRP-2 robot and the admittance control implemented for the end-effectors limit to some extent the interaction forces between the teleoperator and the onsite operator. Thus, perturbations on the ZMP of the robot are small enough for the stability criterion to be satisfied without changing the footprints in the preview window.

#### Control of the master platform

The haptic interfaces are controlled using admittance controllers. From the position of the end-effectors, the base is repositioned so as to always maximize the minimum manipulability of the two end-effectors. More details are given in [27, 83, 113].

#### 4.5.3 Results and Discussion

#### **Preliminary experiments**

A set of preliminary experiments has been performed where only fixed-based manipulation tasks were performed. These experiments allowed us to assess the quality of the bilateral coupling of the end-effectors, and already highlighted a number of problems related to a reduced bandwidth in the haptic communication channel between the human partners.

In these experiments, an object is moved from one side of a table to another. On the remote site, only the arm of the humanoid robot is moving, and the chest is allowed to rotate around the vertical axis. The setup is illustrated on figs. 4.8 and 4.9.

The system remained stable during all the phases of the task, including during interaction with the remote human operator or with the stiff surface of the table. Fig. 4.10 shows the data recorded during a typical trial of the task. Three phases can be distinguished: approaching and grasping (1), moving (2), and landing (3). During *free space motion* (phase 1) the position tracking is very good. The impedance implemented on the slave site contains an important damping component, hence forces are necessary to move the arm, even for unconstrained motion. In the contact phase, the positions deviate from each other, while the forces are well tracked. Actually, for the robot not to move, the remote and slave forces have to be exactly opposite, to result in a zero reference velocity output from the admittance controller. Hence, steady state phases will be characterized by good force tracking. Since the force tracking is good, but positions deviate, the impedance of



Figure 4.9 – Left: teleoperated HRP-2 collaborating with a human on the remote-site. Right: immersed human operator operating the HRP-2 robot using the VISHARD arm.

the remote environment is not well reflected on the master site: the environment is perceived softer as it really is.

This simple experiment needed a significant training period from the operator at master site. First, the narrow angle cameras used for stereovision resulted in a very limited visual feedback. As a result, the operator could not see the arm of the teleoperator and could only see the end-effector. However, although the arm of the teleoperator has six degrees of freedom, the wrists are not spherical, *i.e.* the last three axes do not intersect in a single point. As a result, large elbow motions occurred when rotating the gripper around the vertical axis, leading either to singular positions or self-collisions. To limit this problem, more damping was set on the slave platform around the vertical axis, to discourage the operator from performing this motion. This strategy was effective and the task could be performed without reaching dangerous postures.

On the communication aspects, the intentions of the remote operator could be easily conveyed to the master operator as far as forward-backward motions are considered. Indeed, it is very easy for the remote human operator to apply significant forces in these directions.

In the other directions, however, the perception of each operator's intentions was made difficult by the low bandwidth of the system and the low transparency of the system. It often occurred that both operators kept on moving in the same direction, waiting for each other to stop, or that the master operator did not understand that the remote operator intended to stop the motion. Since forces are only felt from the soft position coupling on the master site, the remotely located operator had to apply significant forces to keep the object from moving so that the master operator could feel some resistive forces.

#### Wide-area motions

Experiments in which both, the human operator and the teleoperator performed some steps have been successfully performed. In these experiments, forward and backward motions have been performed, in interaction with a human operator on slave site. The robot was able to walk and keep balance during the experiments, and the tracking of the end-effector positions was very good. Overall, the master operator did not face much difficulty to take control over the remote humanoid robot.

Fig. 4.11 depicts the results for an experiment in which the human operator



Figure 4.10 – Force and position tracking during experiment. 1: approaching phase, 2: moving phase, 3: releasing phase, shaded area: human located at the remote site applies forces to the object

in Munich applies most of the force necessary to initiate the movement and the human collaborator in Tsukuba loosely holds the end-effectors of the teleoperator.

A very good position tracking between the respective end-effectors is achieved (solid lines), also while walking. The slight deviations that can be seen are caused by time-delay in the communication channel and a desired compliant behavior of the haptic interfaces on master side. However, a large deviation of the body positions between the human operator and the teleoperator are observed (dashed lines). E.g. after 43 s the teleoperator starts moving backwards from the operator's point of view although the human operator is still moving forwards. Also, some low frequency oscillations of the teleoperator position that are caused by the current implementation of the walking pattern generator are evident. The human operator observes this as disturbing. Thus, a high degree of immersion and good task performance while walking were not achieved using this coupling scheme.



Figure 4.11 – Position tracking of body and end-effectors with a dominant human operator at the master site. Data of the human operator at master site is depicted in black, data of the teleoperator in gray. The dashed lines represent the body position whereas the solid lines represent the right hand/end-effector.

Fig. 4.12 depicts the results for the reverse experiment, i.e. the human operator in Munich loosely holds the end-effectors and is guided by the human collaborator in Tsukuba. Position tracking of the end-effectors is equally good. Again, a deviation of the body positions is apparent. However, as the human collaborator on site notices oscillations caused by the walking pattern generator more quickly and reacts immediately, no low frequency oscillations of the teleoperator's body position can be observed.

Summarizing the experimental results, it can be concluded that the goal of performing a teleoperation experiment, in which walking and haptic interaction occur simultaneously, has been achieved. Further improvements to this concept are however necessary to achieve a high degree of immersion and a good task performance. In both experiments, due to the current implementation of the walking pattern generator, the walking speed is relatively slow. The implementation of a faster walking pattern generator is in progress.

#### 4.6 Conclusion

We have presented a telepresence setup where a teleoperation system is sandwiched between two humans. The purpose of the setup is to allow two human operators located at different locations to perform collaborative tasks. One of the



Figure 4.12 – Position tracking of body and end-effectors with a dominant human collaborator at remote site. Data of the human operator at master site is depicted in black, data of the teleoperator in gray. The dashed lines represent the body position whereas the solid lines represent the right hand/end-effector.

human is directly located on the place where the task takes place, while the other assists her/him using an immersive teleoperation system.

We considered a setup where the master device is a mobile haptic interface and where the operator is immersed. On the slave site, we considered a biped humanoid robot, since this platform allows a natural projection of the body of the operator into the remote location.

These experiments were the first step towards a more complex setup were wider range motions were allowed, by exploiting the mobility of the mobile platform on the master site and the biped walking capabilities of the humanoid robot on master site. This involved a significant effort for the whole body control of the humanoid robot on the remote side, to include in a unified framework admittance control, for the bilateral coupling of the grippers, head tracking and gait. The control framework has been introduced, and the components will be more thoroughly presented in the next chapter.

A simple coupling scheme where the end-effectors are bilaterally coupled and where the motions of the mobile base and humanoid robot's body are controlled independently has been implemented. Using the presented control scheme, successful experiments could be performed where a human operator teaches the operator on master site how to operate the robot to walk. The operator on master site could then reproduce the walking motion.

In the next chapter, we present in more detail the control framework used in this experiment. Indeed, all components have been developed within an integrated framework, which allowed a great reusability, and many components used in this teleoperation experiment could be reused "as is" to performed autonomous collaborative transportation tasks.

# 5

## Experiments on virtual avatars and humanoid robots

Conten	ts		
5.1	Introduction		101
5.2	Collaborative transportation task		102
	5.2.1	Problem statement	102
5.3	5.2.2	Stack of Tasks	103
	5.2.3	Force-based control	103
	5.2.4	Real-time Walking Pattern Generator	105
	5.2.5	Footprints planning: the Mobile Robot Helper approach	106
	5.2.6	Decoupling the gait and manipulation tasks	108
	Integration and implementation		111
	5.3.1	Software framework	111
	5.3.2	Integration	114
	5.3.3	Implementation issues	114
	5.3.4	Experimental results	116
	5.3.5	Integration to the final demonstrator of the Robot@CWE project	118
	5.3.6	Discussion and future works	121
5.4	.4 Conclusion		

HIS chapter describes a framework for the control of a humanoid robot to perform collaborative tasks jointly with a human operator. The main components used to build our control architecture are first presented. We introduce an algorithm to realize Cartesian admittance control, while being consistent with the inertial properties of the multi-body at the joint level. We also study how locomotion and manipulation can be integrated together for the robot to perform collaborative transportation tasks as a pure follower.

We then give details about the integration and implementation of these components. We present different experiments where human operators interact directly or through a manipulated object with a humanoid robot HRP-2. We also present a demonstrator including a collaborative transportation task. This demonstrator has been the subject of a user study, which allowed us to assess the validity of our control architecture. Finally, the limitations of our control architecture are discussed and potential improvements are proposed.



Figure 5.1 – Four contexts of task realization in the physical common workspace

#### 5.1 Introduction

The work presented in this chapter has been realized in the context of the European project Robot@CWE, in collaboration with my colleagues Olivier Stasse, Nicolas Mansard, Pierre Gergondet and Nicolas Perrin. The Robot@CWE project aims at introducing advanced robotics platforms, such as humanoid robots embedding information systems, into collaborative working environments. As multi-purpose robotic systems they would be intended to manipulate product, carry tools, inspect product lines and so on. To separate the different contexts of collaborative work in nowadays companies' IT-infrastructure, we have introduced a taxonomy of *contexts* to realize collaborative tasks. This taxonomy is depicted in Fig.5.1. More precisely it consists in four different classes of situations:

- ▷ (1) An *autonomous* context realization when the robot is directly interacting with a human to perform a task, and particularly during physical interaction.
- $\triangleright$  (2) A *local* context realization when the robot is using the surrounding network and computer capabilities to expand its functional space.
- ▷ (3) A *semi-local* context realization when the robot is interacting with a collaborative working application targeted for an application or for a structure such as a company.
- ▷ (4) A *global* context realization when the robot is interacting with services external to its *semi-local* structure for instance Google Images services, manufacturer product specification, etc.

In [101], an example of semi-local context where a robot is asked to perform an inspection task has been presented. For collaborative tasks in a *global* context, the previous chapter presented how collaborative tasks can be performed by a robot in collaboration with a second human located in the workspace of the teleoperator [27, 84]. In this chapter, we focus on the *autonomous* context with the performance of autonomous collaborative tasks in collaboration with a human.

We describe the control architecture we implemented on a humanoid robot HRP-2 [63] to perform various kind of tasks, including direct interaction with a human, bimanual manipulation, and transportation tasks. We tackle some control aspects, as well as software architecture aspects and briefly present the software framework we used to implement our controllers. We show how, thanks to this framework, the work presented here could be integrated seamlessly within the final demonstrator for the Robot@CWE project.

#### 5.2 Collaborative transportation task

In this section, we describe how a collaborative transportation task can be implemented on a biped humanoid robot. We present the problems to solve and the main concepts that we used to achieve our goal.

#### 5.2.1 Problem statement

The scenario of the task is the following: an HRP-2 robot is holding one side of a board, and a human operator is holding the other side. Ideally, most of the weight of the table should be supported by the humanoid robot. In our setup, we will consider a follower robot. Therefore, the robot will have to react to the human intentions and will not perform actions of its own initiative. We consider the haptic channel as the only information flow between the robot and the human operator. Fig. 5.2 shows the experimental setup.

Physical joint actions with a human partner are among the most challenging tasks a humanoid robot is expected to achieve with robustness and safety. The scenario presented on fig. 5.2 raises the following difficulties:

- Since a human operator is in the loop, the trajectory of the table, and thus of the robot itself, can not be pre-programmed. Therefore, the robot needs to reactively plan footprints to adjust its trajectory, and to quickly generate a dynamically stable gait.
- When walking while manipulating an object jointly with a human operator, the robot will be subject to external forces which will threaten its balance. If these forces are significant, they have to be taken into account in the gait generation itself [80, 105]. In general, however, if both agents collaborate in a constructive way, and if the robotic partner has a compliant behavior, the interaction forces will be limited to some extent. Therefore we will not take these forces into account at the gait generation level in the context of our work.
- Since we consider the haptic channel as the only information channel through which the human operator can communication his intentions, the forces measured at the wrists of the humanoid robot must be mapped into a whole body motion, and footprints generation.



Figure 5.2 – *Cooperative table transportation between an HRP-2 humanoid robot and a human operator.* 

▷ Finally, given the discrete nature of the bipedal gait, the motion of the body of the robot should be computed to absorb as much as possible the fluctuations induced by the gait, for the comfort of the human operator.

The remainder of this section describes the main components that have been used to address these problems. Then, we show how they have been integrated together to achieve our goal.

#### 5.2.2 Stack of Tasks

To generate whole-body motion on the highly redundant HRP-2 humanoid robot, we make use of the Stack of Tasks briefly introduced in the previous chapter. From a task function **e**, its Jacobian  $\mathbf{J} = \frac{\partial \mathbf{e}}{\partial \mathbf{q}}$  and a desired behavior  $\dot{\mathbf{e}}^*$ , a joint velocity vector is computed, using the generic inverse kinematics solution:

$$\dot{\mathbf{q}} = \mathbf{J}^{\#} \dot{\mathbf{e}} + \mathbf{P} \mathbf{z} \tag{5.1}$$

where  $\mathbf{z}$  is an arbitrary vector projected into the null space of the Jacobian  $\mathbf{J}$ , and  $\mathbf{J}^{\#}$  can be *any* Pseudoinverse of the task Jacobian. This equation can be used recursively, using the vector  $\mathbf{z}$  to try to realize at best other tasks (see the brief introduction in the previous chapter (4.4.1), as well as *e.g.* [71]).

#### 5.2.3 Force-based control

For position controlled robots, admittance can be implemented to obtain an active compliant behavior of the robot. As stated in the previous chapter, admittance control consists in defining a desired dynamic behavior for a given point on the robot, such as the one of a virtual mass-damping system:

$$\mathbf{M}\ddot{\mathbf{r}} + \mathbf{B}\dot{\mathbf{r}} = \mathbf{f} \tag{5.2}$$

where **M** and **B** are arbitrary mass and damping matrices of the equivalent virtual system, and **f** are forces and torques exerted on the equivalent virtual point. The equation (5.2) is then integrated to compute a desired velocity  $\dot{\mathbf{r}}$ , which is then used as the desired behavior for a task i:  $\dot{\mathbf{e}}_i = \dot{\mathbf{r}}$ . Assuming that the robot has no dynamics and is a pure motion source, admittance control will result in the desired compliant behavior of the robot. In practice, this is not the case, and the gains will be limited by the control rate, and by the intrinsic dynamic properties of the controlled system.

In (5.1), the matrix  $J^{\#}$  can be chosen as *any* Pseudoinverse. Using the usual Moore-Penrose Pseudoinverse, the solution we obtain, assuming z = 0, is the one that minimizes the norm of  $\dot{q}$ . If the solution given by this Pseudoinverse is used as a reference joint velocity vector for the controlled robotic system, then the joints will participate to the compliant motion *regardless of the inertia they are subject to*. In other terms, we might impose a very different dynamics than the natural dynamics of each joint in the multi-body. The effect of force-feedback control is to reduce the apparent inertia of the system [46]. Thus, by "asking the contribution" of each joint independently of the inertia they are subject to, we are potentially trying to reduce significantly the apparent inertia of each joint. This can destabilize the system, because the low-level PD joint position controller of the robot might not be able to track the desired joint dynamics. Moreover, we impose very fast motion of joints subject to high inertias, which is not optimal in terms of energy consumption, and will tend to excite oscillatory behaviors if the PD position controller can not perfectly track the imposed dynamics.

To solve this problem, we propose to apply a weighted Pseudoinverse [18], defined when J is full-row-rank by:

$$\mathbf{J}^{\#\mathbf{W}} = \mathbf{W}^{-1} \mathbf{J}^{\mathbf{T}} \left( \mathbf{J} \mathbf{W}^{-1} \mathbf{J}^{\mathbf{T}} \right)^{-1}$$
(5.3)

with W an arbitrary invertible matrix that represents the weights on the joints.

Taking the derivative of (5.1) with z = 0 yields:

$$\ddot{\mathbf{q}} - \dot{\mathbf{J}}^{\#\mathbf{W}} \dot{\mathbf{e}}^* = \mathbf{W}^{-1} \mathbf{J}^{\mathbf{T}} \left( \mathbf{J} \mathbf{W}^{-1} \mathbf{J}^{\mathbf{T}} \right)^{-1} \mathbf{M}^{-1} \left( \mathbf{f} - \mathbf{B} \dot{\mathbf{r}} \right)$$
(5.4)

Selecting the weight **W** as the inertia matrix **A** of the robot and the virtual mass **M** as  $\Lambda = (\mathbf{J}\mathbf{A}^{-1}\mathbf{J}^{T})^{-1}$ , the apparent inertia of the end-effector of the robot, we obtain:

$$\mathbf{A}\ddot{\mathbf{q}} + \mathbf{B}_{\mathbf{q}}\dot{\mathbf{q}} = \mathbf{J}^{\mathrm{T}}\mathbf{f}$$
(5.5)

where  $\mathbf{B}_q = \mathbf{J}^T \mathbf{B} \mathbf{J} - \mathbf{A} \dot{\mathbf{J}}^{\#W} \mathbf{J}$  is the friction factor of the whole body structure. It can reasonably be assumed, if enough friction has been set for the desired dynamic behavior, that the friction term is equivalent to the projection of the desired friction into the joint space. This last equation corresponds to a simplified version of the dynamic equation of the whole-body compliant robot with gravity compensation, with forces **f** acting on **r** and a friction  $\mathbf{B}_q$  that may be tuned by selecting **B** to stabilize the system.

Selecting the parameters as proposed, namely, setting the desired mass as the real apparent mass at the contact point, and weighting the Pseudoinverse by the inertia matrix of the robot have two effects. The first one comes from the Pseudoinverse: by weighting it, we avoid to solicit joints which are subject to high inertias. The second effect is that by further not imposing a different inertial behavior to the

robot from the natural one, the dynamics of the system can be approximated by the dynamics of a multibody with gravity compensation and tunable joint friction.

At first, we can think that keeping the real apparent mass of the contact point will reduce the performance of the system. However, on the HRP-2 robot, the main benefit of force feedback control is actually to help the operator fight against friction. The joint friction of the HRP-2 robot makes it very difficult to exploit passive compliance and to move the body of the robot when it is unactuated. Hence, this control scheme effectively eases human-robot interaction even if the prescribed inertia is not reduced.

#### 5.2.4 Real-time Walking Pattern Generator

Considering a humanoid robot with a state vector  $\mathbf{x} = [\mathbf{r}, \mathbf{q}]^T$ , where  $\mathbf{r}$  are the free floating parameters and  $\mathbf{q}$  the joint position vector. To perform motions while keeping balance, a trajectory  $\mathbf{q}(t)$  has to be generated such that the Zero Momentum Point of the robot lies within the support polygon defined by the contact points between the robot and its environment.

One approach to generate such movements consists in first defining a set of contact (typically, footprints), then computing a desired ZMP trajectory that is compatible with this contact (which lies within the support polygon at all time), and then compute a motion  $\mathbf{q}(t)$  such that the ZMP of the system tracks the desired ZMP trajectory.

To solve this problem, a simplified Linear Inverted Pendulum Model is often used [53]. From this model, a relationship can be established between the ZMP of the system and the CoM of the pendulum. This relationship can be written under the form of a dynamical system where the jerk of the CoM is used as an input. We can thus derive a controller that compute appropriate values of the jerk of the CoM to obtain the desired ZMP trajectory. However, the structure of the system is such that, to have a good tracking of the desired ZMP trajectory, the position of the CoM should move in advance. In other terms, the current output of the ZMP controller (the jerk of the CoM) must be computed from the future desired ZMP trajectory. As a result, the ZMP trajectory must be specified for a preview window of a given length. In addition, the algorithm proposed in [52] takes into account, to some extent, the whole multi-body model by computing the ZMP error between the LIPM and the multi-body model, and a resulting compensation in advance.

As a result, new reference footprints can be used as a reference to generate walking motions with a delay corresponding to the length of the preview window. This considerably reduces the reactivity of the system. The cost of the dynamic filter is too high to change in real-time the feet positions inside the preview window, when the CoM reference trajectory is generated every 5 ms. In Nishiwaki et al. [80] a method is proposed to change on-line, and inside a step-cycle, the ZMP and CoM trajectories while including a dynamic filter. This impressive method has been used for vision based reactive planning [12] and reactive interaction with heavy objects [105]. The method relies mostly onto two main concepts: increasing the time period to generate the reference CoM trajectory with the dynamic filter (from 5 ms to 20 ms) and increasing the frequency of the ZMP controller (from 200 Hz to 1 kHz).

Another technique, proposed by Harada, and Morisawa [38, 76], considers that the ZMP trajectory is a third order polynomial. This allows the derivation of an analytical solution for the CoM trajectory and the modification of the steps in preview



Figure 5.3 – Virtual coupling of the end-effectors and the body of a humanoid robot to synchronize the arms and the body of the robot during a collaborative transportation task.  $M_{base}$ ,  $B_{base}$ ,  $M_{arms}$ ,  $B_{arms}$  and  $K_{arms}$  are the virtual impedance parameters used to couple the arms and the base, and to generate the motion of the base.  $F_{ext}$  is the force applied by the human operator on the arms. The model depicted on the right can be seen as desired dynamics for the motion of the body and arms of the robot.

window. One particularity of this algorithm is that changing the ZMP reference in the preview window can induce fluctuations of the ZMP reference when connecting the new reference to the old one. This is due to the polynomial nature of the ZMP reference in this algorithm. To compensate for these fluctuations, it has been proposed in [76] to increase the single support time of the step.

#### 5.2.5 Footprints planning: the Mobile Robot Helper approach

The problem of footprints planning can be tackled in different ways. In [120], the authors implemented the algorithm proposed by Kosuge for his Mobile Robot Helper in [66] to synchronize the motion of the arms with the motion of the body of the robot. The idea of this algorithm is to decompose the robot into two parts: the arms and a mobile base, and to virtually couple the motion of these elements. This is illustrated on fig. 5.3.

Intuitively, one can see that a problem arises with this algorithm. The motion of the base specified by the desired dynamics shown on fig. 5.3 is continuous, but a biped humanoid robot will typically perform discrete steps, which will result in a body motion that is different from the prescribed dynamics. Unless this prescribed dynamics is taken into account *during* the gait generation phase, it will not be correctly tracked by the robot.

In our case, we will avoid, as possible, to modify the gait generation of a robot, which we will consider as a black box. Hence, the only way to approximate the dynamics specified by fig. 5.3 is to plan appropriate footprints. The reactive walking pattern generator presented in the previous paragraph allows us to modify the next step to be performed by the robot during each double support phase. In our setup,



Figure 5.4 – Top: velocity of the arms and base using the algorithm of Kosuge with time delay. Middle: distance between the arms and base. This distance must be kept slow enough to fit the workspace of the robot. Bottom: virtual force applied on the base of the robot due to the virtual coupling between the arms and base.

these phases are separated by a duration of approximately one second. Therefore, we can *roughly* model the nature of the motion of the humanoid body by adding a time delay of one second in the system depicted on fig. 5.3: that is, the force due to the virtual coupling between the arms and the base will be transmitted with a one second time delay to the base.

If we denote  $F_{arm/base}$  the force applied by the spring damper system  $K_{arms}$ ,  $B_{arms}$  on the base of the robot, then the motion on the base would then be given by:

$$V_{base}(t) = B_{base}(t-T)^{-1} F_{arm/base}$$
(5.6)

where T = 1s is the time delay induced by the discrete nature of the stepping and the computation constraints of the pattern generator.

To evaluate the impact of this time delay on the gait of the robot, we simulated this system using Simulink, with gains that we knew by experience to be appropriate for the admittance control of the arms of the robot <sup>1</sup>. Fig. 5.4 shows the simulation results for  $M_{arms} = 5$  kg,  $B_{arms} = 100$  Ns/m,  $K_{arms} = 25$  N/m,  $M_{base} = 5$  kg, and  $B_{base} = 50$  Ns/m. The figure 5.5 shows the result with the same gains without time delay.

We notice that the time delay introduces a highly oscillatory behavior, which might disturb the human operator. To avoid these oscillations, we rather use a different approach where the base and the arms are not *bilaterally* coupled, but only unilaterally. This is explained in the next paragraph.

<sup>&</sup>lt;sup>1</sup>If we consider a fixed mobile base, the virtual coupling between the arms and the base reduces to impedance/admittance control of the arms.



Figure 5.5 – Velocity of the arms and base using the algorithm of Kosuge without time delay. Compared to the case with time delay, the velocity profiles are smooth.

#### 5.2.6 Decoupling the gait and manipulation tasks

Regardless of the synchronization of the base of the robot with the arms to perform wide-range motions, a problem that arises when the robot walks using the LIPM and ZMP criterion is that the body of the robot swings laterally to track the CoM reference computed by the walking pattern generator. If this swinging motion is not compensated by the arms of the robot, then the human operator will experience oscillatory forces at the grasping points of the table. On top of being uncomfortable for the human operator, this will make precise positioning tasks difficult, because the grippers of the robot will oscillate.

This swinging motion can be compensated for by considering the appropriate Jacobian matrix for the admittance control task. The gait of a biped robot is a succession of double support phases and single support phases. During each of these phases, if we assume no slippage between the floor and the support foot, the support foot can be considered as a fixed base for the robot. This means that the absolute velocity of this body of the robot is considered to be zero. Thus, by considering the support foot as the root of the robot, and by considering Jacobian matrices with respect to this root, the motions of the legs when the robot is walking can be compensated for. Fig. 5.6 shows the swinging motion of the waist and the motion of the grippers, to show the amplitude of the swinging motion of the waist during the walk and the efficiency of the swinging motion of the grippers is 10 times less than the one of the waist. The small oscillations of the gripper could be explained by the damping of the Pseudoinverse used to compute the control law. This implementation issue is discussed in the next section.

Therefore, this allows us to implement the admittance control of the arms of the robot in such a way that in absence of external forces applied on the grippers of the robot, the end-effectors will have a zero absolute velocity, despite of the swinging of the body of the robot due to the gait. In other terms, this allows us to decouple the gait of the robot from the manipulation task, and to display a desired admittance on the grippers of the robot regardless of the body motion. This is illustrated on fig. 5.7.



Figure 5.6 – Top: lateral swinging motion of the waist during on-place stepping of the robot. The amplitude of the motion is about 10 cm. Bottom: reduced oscillations of the gripper (amplitude of about 1 mm) after the compensation of the legs' motion.



Figure 5.7 – Decoupled admittance control and gait: the admittance of the end-effector of the robot is a damper fixed to the ground rather than to the body of the robot, allowing to decouple force-based control from the motion of the body of the robot.

Now that the gait of the robot is decoupled from the admittance control task, the question is: how do we plan footprints to synchronize the locomotion with the gait? Indeed, the decoupling between the manipulation task and gait is possible only as long as we stay in the workspace of the robot. But to allow wider-range motions, the robot will have to walk.

Let us first define a reference frame related to the hands of the robot. The center of this frame is the middle of the segment joining the robot's end-effectors. One of the axes of this frame is supported by this segment, the other axis is vertical, and the third axis is set so as to obtain a right-hand frame. We note  ${}^{lf}H_{ref}$  and  ${}^{rf}H_{ref}$  the transformation matrices (homogeneous matrices) from the left and right foot to this reference frame, respectively, as depicted on fig. 5.8.



Figure 5.8 – Reference frames used for the synchronization of the feet and hands.

Synchronizing the hands and feet is defined as keeping these transformations as close as possible to two reference transformations  ${}^{lf}H^0_{ref}$  and  ${}^{rf}H^0_{ref}$ . This can be seen as tracking a reference posture, but instead of defining a full configuration in the joint space of the robot, we define a relationship between the feet and hand positions, giving more freedom to the robot.

During each double support phase, the relative position error between the next fly foot and the reference frame will be computed, and the landing position of the next fly foot will be selected so as to correct this error. The transformation describing the fly foot destination is:

$$H = ({}^{ff}H^0_{ref})^{-1} {}^{ff}H_{ref}$$
(5.7)

where ff means fly foot. This transformation is then projected into the appropriate coordinate system to send a new footprint reference to the walking pattern generator.

The planned footprint is then clipped to a safe area, which computation is described in [85]. This safe area is depicted on fig. 5.9.



Figure 5.9 – Safe area for the landing foot.

#### 5.3 Integration and implementation

This section presents how the concepts presented in the previous sections are integrated together to realize collaborative transportation tasks as well as direct physical interaction with a humanoid robot. We describe some issues we faced when implementing such a task on a real system and present the experimental results. The limitations of the presented scheme are explained and related to the concepts presented in the first part of this thesis.

#### 5.3.1 Software framework

All the components described in the previous section have been integrated within or interfaced with a flexible software framework centred on the Stack of Tasks.

#### Entities, graph of entities and signals

The control framework used to implement our algorithms has been thoroughly described in [73]. This framework is centred on computational elements called *entities*. Each entity has input and output signals and provides algorithms to update the values of the output signals based on the input signals. Each class of entity can be instantiated several times. Each instance is associated with a unique string identifier, and all entities are registered into a pool of entities.

The entities are connected together through the input and output signals to form a graph of entities. The data flow through the graph is based on the signals, which are associated to timestamps. This timestamps allow a signal to trigger its recomputation only once when its timestamp is not up-to-date. Output signals can be connected to zero, one or several input signals. If an output signal is not connected to any input signal, its value will never be updated.

The control framework is mainly built upon these concepts of entities and signals, which can be compared to the mechanism offered by Simulink. To build a controller, entities which perform elementary operations will be connected together to perform more complex actions. The framework can be extended via a plugin mechanism. The framework allows to load online dynamic libraries which contain entities, which can then be instantiated and used within the controller. We thus benefit a highly dynamic control framework which allows online modification of the structure and parameters of the controllers.

#### Scripting

The control framework comes with a shell and a script language, which allows to instantiate entities, to display information about them, and to connect them through their input and output signals. The script language and shell prompt also allow to call specific methods implemented by the entities, using a very simple syntax. For instance typing

```
pool.help
returns:
Pool:
    - list
    - listFeature
    - listTask
    - writegraph FileName
```

The first method list all the entities created in the current instance of the factory. The second list only the features, while the third provides the name of the task entities. The last one finally generates dot graph which can be displayed as presented in Fig.5.10.

#### **Stack of Tasks**

The control framework is built upon the concept of Stack of Tasks. The Stack of Task is a special entity that computes the control law using the formalism presented in the previous chapter. Different tasks can be pushed into and popped from the stack, and the tasks can be reorganized so as to change their respective priorities. The tasks pushed into the stack are again specific entities which implement the necessary computations to implement the control law. The tasks use feature to compute the task error. Features objects provide the vectors  $\mathbf{s_i}$ ,  $\mathbf{s_i^*}$  and the matrix  $\mathbf{L_{s_i}MJ_q}$  for a task  $\mathbf{e_i}$ . The role of the features here is strictly limited to:

- $\triangleright$  receive the desired values  $s_i^*$ , the current value  $s_i$  of the feature according to the robot state, and the robot articular jacobian in the proper reference frame i.e.  $MJ_q$ .
- $\triangleright$  compute the feature Jacobian  $L_{s_i}$ .

#### **External components**

The framework presented in [73] provides generic elements to compute control laws based on the Stack of Tasks. However, it does not come as a stand-alone solution to control a humanoid robot. Hence, specific entities have been programmed to be used as interfaces between the Stack of Tasks software framework and other libraries, such as libraries to compute the dynamical model of the robot, or the walking pattern generator, described in [104].



Figure 5.10 - Graph of entities at a given time of the control (particular case of the teleoperation experiment presented in the previous chapter). Such a graph is useful to have an idea of the data flow in a given control architecture and can be generated using a simple command line.



Figure 5.11 – Simplified view of the architecture used for the collaborative transportation experiment. OpenHRP is used as a proxy between the robot and the Stack of Tasks control architecture. From the state of the robot, a relative position error between the end-effectors and the feet of the robot is computed, and resulting footprints are planned and sent to the pattern generator. From these footprints, the pattern generator computes a reference CoM trajectory and reference trajectories for the feet. The Stack of task then computes a global control law to realize the CoM and feet trajectories as well as the admittance control of the end-effectors, which has the lowest priority.

#### 5.3.2 Integration

The Stack of Tasks framework has been used together with the OpenHRP [55] architecture to implement the proposed algorithms. OpenHRP is used as a proxy to get the state of the robot (joint position vector, sensor values) and send reference joint position vectors to the actuators. The global architecture is depicted in a simplified form on fig. 5.11. This figure also illustrates how the tele-cooperation setup presented in the previous chapter is seamlessly integrated within the framework.

Using the three tasks that appear in fig. 5.11, the robot can walk stably, while its arm will comply to external forces applied on the grippers. The highest priority task is the tracking of the trajectory for the feet. Then comes the tracking of the CoM trajectory. These two tasks are essential to keep the balance of the robot, thus they must be realized with the highest priority.

#### 5.3.3 Implementation issues

We faced several issues when implementing the algorithms on the real humanoid platform HRP-2. Some of these issues are inherent to the control architecture we used, and others are inherent to the mechanical structure of the robot itself. We detail these issues in this paragraph.

#### Arm manipulability

First of all, the gains of the admittance controller could not be selected to reach a very compliant behavior of the robot because of the high joint friction and the inertia of the robot. When performing bimanual manipulation tasks, the admittance has to be even lower than when moving the arms freely. Since we do not take into account the external forces in the gait generation, the less compliant the arms of the robot are, the higher the risk to lose dynamical balance will be.

The wrists of the HRP-2 robot are not spherical, which means that the endeffectors of the robot can not reach any desired orientation without moving the elbow. This results in very large motions of the elbow for only small orientation displacements of the end-effector, with two consequences:

- ▷ the elbow can easily get aligned with the shoulder, resulting in a kinematic singularity;
- ▷ the elbow can easily collide the trunk of the robot.

This second problem could probably be solved using collision avoidance algorithms. However, such algorithms do not easily fit the current implementation of the Stack of Tasks, which brings to a second class of issues.

#### Unilateral tasks

Unilateral tasks are tasks that are defined by inequalities. Typical unilateral tasks are contact tasks, collision avoidance task, singularity avoidance and joint limit avoidance tasks. The implementation of the Stack of Tasks we used for our experiments did not handle unilateral tasks. Attempts to treat unilateral tasks as bilateral ones can result in getting stuck on the constraints. This problem is evoked in [102].

Another possibility is to freeze the motion of the robot when colliding, but then we need a criterion to unfreeze the motion. A solution is to compute the control law all the times and check whether the robot is self-colliding or not, and to send only those commands that are safe for the robot.

Joint limits avoidance is another typical unilateral task. The HRP-2 robot can only bend a few degrees backwards using its chest joint. However, when pulling the end-effectors up, the robot tries to contribute to this motion using the chest joint, and hence reaches the chest joint limit. This problem can only be limited to some extent using the weighted Pseudoinverse method presented in section 5.2.3. Here, freezing the whole robot can result in strong limitations in the possible motions of the robot, since the joint limit can be reached for any upwards motion of the end-effectors.

One solution is then to only clamp the values of chest joint into its articular limits. However, if this is done *after* the computation of the control law, then the computed joint motion will lead to an incorrect realization of the tasks, since from the point of view of the Stack of Tasks, the chest joint is supposed to be used. This means that if the chest joint is clamped, this information must be taken into account in the computation of the control law, *i.e.* to handle unilateral tasks [72], which is not the case in our current implementation of the Stack of Tasks. The solution we adopted is thus to freeze the chest joints by adding a bilateral task which imposes a zero velocity to the chest joints. In practice, this results in a slight loss of performance because of the reduced workspace, but it greatly improved the repeatability of our experiments.

An elegant mathematical framework to address this problem has been recently proposed by Kanoun [56]. However, there is currently no solver that can compute the control law fast enough to fit the 5 ms of the HRP-2 control loop.

#### Singularities and damping

The Stack of Tasks relies on the computation of a Pseudoinverse of the Jacobian matrix associated to each task. This matrix can sometimes lose rank and become singular. In such situations, the reference joint velocities can be very high. To avoid this problem, the Pseudoinverses of the Jacobian matrices are damped. As a result, the tasks are not correctly realized when reaching singularities.

This had dramatic consequences in experiments where the robot was supposed to handle a table. In these experiments, a task was used to constrain the altitude and attitude of the end-effectors of the robot, so that they could only comply to the external forces by moving in the horizontal plane or by rotating along the vertical axis. However, due to the limited manipulability of the arms of the HRP-2 robot, the arms got very close to singularities when pulling on the table to make the robot step forward. As a result, the altitude-servoing task was not correctly realized when the arms were stretched. This resulted in instable behaviors and very high internal forces on the table.

The solution we adopted is to replace the servoing task by an impedance control task. We made use of the admittance control task and added a virtual spring along the vertical axis to maintain the table around a desired altitude. This has significantly improved the stability of the system. This issue was a good illustration of the relevance of impedance control for manipulation task compared to servo control.

#### 5.3.4 Experimental results

Two kinds of task were performed with the HRP-2 robot. The first interaction paradigm that was tested is direct interaction between a human operator and the robot, where a user moves the arms of the robot to trigger locomotion. This is illustrated on fig. 5.14. As can be seen, several users could interact with the robot.

#### **Good ZMP tracking**

Fig. 5.12 depicts the ZMP reference along the x axis computed by the pattern generator described in section 5.2.4, and obtained during an experiment where the motion of the robot was constrained on the sagittal plane (the robot stepped only forward and backward). The real ZMP is also represented. One can see the deformation induced by the change of the swinging foot position when this one starts its flying phase. The ZMP real is deviating of 2 cm from the reference trajectory which is quite similar to the result obtained in [80]. We conclude that the compliance of the arms of the robot is able to absorb most of the perturbation coming from the interaction with the user, preventing the fall of the robot. It is also important to note that the experiments were ran using the commercial stabilizer provided with OpenHRP [54]. This also explains the good quality of the ZMP tracking.



Figure 5.12 – *Reference and real ZMP in a direct interaction experiment. On the right: zoom between* t = 40s *and* t = 50s.



Figure 5.13 – Force exerted on the right wrist (component along the axis of the forearm). The impacts due to the steps of the humanoid robot result in additional noise in the signal. Overall, during direct physical interactions between the robot and human operators, the applied forces rarely exceed 10 N.

#### Low interaction forces

This is confirmed by looking at the interaction force measured by the wrist sensors. The component along the right arm is plotted on fig. 5.13. We first notice that the signal is very noisy. This is generally the case with force measurements, and this is made worse by the impacts between the feet and the ground, which generate structural vibration of the robot. However, the noise is filtered by the admittance controller, and has no noticeable impact on the motion of the arms. We also notice that forces do not exceed about 10 N in the plotted time interval. This value was actually an upper bound for the whole experiment from which the data has been recorded. This means that forward and backward motions, which are essentially generated by applying forces along the arms of the robot, can be generated by forces under 10 N. This also explains the relatively good quality of the ZMP tracking shown in fig. 5.12, considering that the interaction forces are neglected.

In a second phase, the robot was holding a board, and a human operator was applying forces on the other side of the board to move the board along with the robot. This is illustrated on fig. 5.15. More details about this experiment are given in the next paragraph.

#### 5.3.5 Integration to the final demonstrator of the Robot@CWE project

The "Human-Robot interaction through an Object" paradigm was integrated in the final demonstrator of the European project Robot@CWE. The scenario of the demonstrator also includes a teleoperation phase, which integrates the work presented in the previous chapter. The complete scenario of the demonstrator is the



Figure 5.14 – Experimental results: direct interaction between a user and the HRP-2 robot

following: a human supervisor requires the robot to go to a target location where an object of interest is located, using a tablet PC. The robot takes a picture of the object and sends it to the supervisor. The trajectory to the target location is computed using a planner. This phase of the demonstrator has been described in [101].

Once the picture has been taken, the robot waits for another human operator to teleoperate it, in order to walk to the object and grasp it. The robot then lifts the object autonomously, and the collaborative transportation task starts. The robot and the human jointly transport the object to a specified destination, the robotic partner acting as a *pure follower*. When reaching the target location, a new teleoperation phase starts, where the teleoperation system is sandwiched between two human operators to perform an assembly task.

A preliminary version of this demonstrator, including the autonomous walking phase, the two teleoperation phases and the transportation phase has been the subject of a user study conducted by researchers from the University of Salzburg. In the context of this study, 12 users with no experience in robotics have participated to the scenario of the demonstrator, taking the role of the human partner during the collaborative manipulation task. Among these 12 subjects, 11 managed to perform the task, and 3 of them from the first trial. This highlights the repeatability of the experiment and the robustness of the algorithms. We can also conclude that the proposed architecture for collaborative transportation task results in a sufficiently natural behavior of the robot, to allow inexperienced users to rapidly manage to perform the task successfully.



Figure 5.15 – *Cooperative table transportation between an HRP-2 humanoid robot and a human operator.* 



Figure 5.16 – Motion of the robot during one trial of the final demonstration scenario for the Robot@CWE (top view). The dashed gray line shows the rough trajectory of the robot. The arrows correspond to snapshots of the position and orientation of the robot every 3s. The red part corresponds to the beginning of the scenario, when the robot walks autonomously, following a computed plan. The purple part corresponds to the first teleoperation phase, where the robot is teleoperated to walk towards the object to transport. The blue part corresponds to the collaborative transportation phase.

#### 5.3.6 Discussion and future works

One of the main limitations encountered during our experiments, regardless of the implementation details mentioned in paragraph 5.3.3, is the low velocity of the locomotion of the humanoid robot when walking in reaction to the human forces. This slow motion can be appreciated by looking at figure 5.16, where the arrows are snapshots of the robot's position and orientation taken at constant time intervals (every 3 seconds). One can see that during the teleoperated phase and the collaborative transportation phase (purple and blue arrows), the locomotion of the robot is much slower than during the fully autonomous phase (red arrows).

The low velocity of the locomotion can be explained in different ways.

- One of the reasons for the slow locomotion comes from the reactivity of the walking pattern generator itself, which we use as a black box. Due to numerical issues in the implementation we used, small changes in the orientation of the footprints resulted in longer single support phases. This phenomenon is specific to the algorithm we used to generate gait [76, 103]. Note that this only occurs in the teleoperated and collaborative transportation phases, because in these phases, the footsteps are constantly replanned within the preview window used by the preview control in the walking pattern generator. In the autonomous phase, the footsteps are pre-programmed for the whole trajectory, and the steps in preview window are thus never modified.
- A second reason is that the position of the swinging foot is planned at the very beginning of the single support phases. Soon after the swinging foot leaves the ground, the landing position can no longer be modified, which limits the reactivity of the system <sup>2</sup>. This is not a limitation of the algorithm itself, and we could also try to allow the modification of the landing position of the swinging foot for a longer time after the double support phase.

 $<sup>^{2}</sup>$ This had been taken into account by adding a time delay in the model described in paragraph 5.2.5

However, modifying the landing position at an early stage of the single support phase is more conservative and has greater chances to result in a stable motion.

To overcome this issues, it is of course necessary to improve the gait generation itself, to be able to perform fully reactive gait. Another approach, which could allow to improve the task performance without working on the gait generation problem, is to focus on the aspects related to collaborative tasks.

The collaborative behavior we have presented in this chapter assigns a fully passive follower behavior to the robot. Since the locomotion and the manipulation of the robot are only unilaterally coupled (the manipulation task influences the locomotion, but the locomotion does not directly influences the manipulation task), what the human operator perceives is the dynamic behavior imposed by the admittance control of the end-effectors of the robot. The locomotion is only used as a tool to enlarge the workspace of the robot. Hence, even if the robot supports the weight of the manipulated object, it dissipates the energy introduced by the human operator, thus making the task actually *more difficult* (at least regarding the motions in the horizontal plane). Finally, one more difficulty comes from the instable nature of humanoid robots. Because external forces are not taken into account at the gait generation level, the human operator must avoid to apply too large forces on the end-effectors of the robot, thus limiting the velocity of the system.

This problem is the main reason why active following schemes such as in [16, 70], which we presented in the first chapter of this thesis, were proposed. However, the difficulty with active following is that it requires a model for the prediction of the human intentions. In [16, 70], simple point-to-point tasks were considered, which can be well described by the minimum jerk model [31]. For more complex tasks, such as the one envisaged in this chapter, where the robot has no clue on the task to realize, active following schemes will be probably very difficult to implement.

Therefore, to give a more active role to the robot, it would be useful to give a *a priori* task plan to the robot. This task plan will then need to be adapted and negotiated with the human operator. In such a situation, the model we presented in the second chapter of this thesis would be applicable.

#### 5.4 Conclusion

In this chapter, we have presented basic components used to build a control architecture to allow an HRP-2 humanoid robot to perform whole body motion and locomotion in response to forces applied by a human operator. The first section also introduced a way to integrate dynamically consistent admittance control within the Stack of Tasks framework, presented in this chapter and in the previous chapter. This admittance control makes use of the redundancy of the robot to display a Cartesian admittance while following a whole body dynamics which is consistent with the inertial properties of the multi-body system. Note that this controller does not need the robot to be force control and simply requires redundancy of the controlled robot. In the case where the robot is not redundant, this algorithm reduces to the case of classical Cartesian admittance control built around a joint position controller.

The second section has presented how the different components were integrated together to achieve our goal, namely, to have a real humanoid robot physically interact with a human operator. The software framework within which our components have been implemented allow a seamless integration of the work presented in this chapter and the previous one. This integration resulted in a complex demonstrator for the European project Robot@CWE. The usability of this demonstrator as well as the robustness of our algorithms have been assessed by a user study conducted within the project. During this study, users who were not experienced with robots were asked to perform a collaborative transportation task with the HRP-2 robot. Though at this time, the final results of the study are not known, only one of the twelve subjects could not perform the task, and three of them could perform the task from the first trial.

Finally, we have discussed the implementation issues and experimental results, in the context of direct interaction between the human operator and interaction through a jointly manipulated object. The results show that the interaction forces are low during the tasks, and that the perturbations on the dynamic balance of the robot are limited by the compliance control of the arms of the robot as well as the commercial stabilizer provided with the low-level control architecture of the robot.

The main possibility of improvement of our system is to accelerate the motion of the robot. We discussed the potential ways to achieve this. It can be done by addressing perturbation forces at the gait generation level, but the task performance could also be improved by enhancing the collaborative skills of the robot. Indeed, the behavior of the robot in our system corresponds to the one of a passive follower. This means that part of the energy provided by the user is dissipated instead of being transformed into kinematic energy. However, in order not to destabilize the robot, the amount of power input from the human operator is limited. As a result, the velocity of the robot is limited. Investigations must be conducted to evaluate how the collaborative behavior of the robot could be enhanced by the model we proposed in the second chapter of this thesis. A better understanding on how the human partner conveys his intentions would also allow to implement active following behaviors, which could lead to faster motions.

### Conclusion

PHYSICAL interaction between robotic systems and human beings is a challenging and interdisciplinary topic. In this thesis, we focused on one modality of physical interaction between one robotic and one human agents, namely, haptic collaborative tasks. The typical tasks we considered are collaborative manipulation tasks where the two partners apply forces on a same object to bring it to a target destination or to impose a desired motion. A general problem for such task is that both partners must agree on a common plan, even when their intentions differ, since only one trajectory can be imposed to the object. The partners have to adapt to the discrepancies between their respective intentions, and to each other's constraints.

The first part of this thesis is centred on this role distribution issue. The main contribution is to extend the classical leader-follower model to a continuous, timevarying role distribution between the leader and follower behaviors. We propose to model this idea using a homotopy between two controllers: one implementing the leader behavior, and the other implementing the follower behavior. The behavior of each partner of a dyad during collaborative tasks can then be described by two independent functions that characterize the role distribution among the partners and shape their behavior. This model raises the question of how to define the role distribution along the task. If we want to implement such a model on a robotic system, then we must be able to set the homotopy parameter that will define the behavior of the robot during the task. We propose three different strategies. The first one consists in looking at the internal constraints of the robot, and to adopt a leader behavior when these constraints are close to be violated. This allows the robot to impose a behavior which is consistent with its constraints to the human partner. The second strategy consists in parameterizing observed phenomena in human-human interaction. Finally, the third strategy is to look at the behavior of the human partner and to adapt the behavior of the robotic partner accordingly. For this purpose, the identification of the actual role of the human would be useful, although this is likely to be difficult. In this thesis, examples have been given to illustrate the first two strategies, which show that our model is rich enough to encompass the specialization phenomenon observed in human-human collaborative positioning tasks, and also allows the implementation of collaborative behaviors with self-collision avoidance.

Our extended leader-follower model allows the implementation of many behaviors on a robotic platform, but it can be interesting to assess its relevance to describe actual human behaviors. Another contribution of this thesis is to propose the use of *programming by demonstration* to teach collaborative skills for physical tasks to a robot. Using a programming by demonstration framework, where extreme leader and follower roles were demonstrated to a robot, collaborative lifting tasks could be successfully reproduced by the robot in an autonomous way. During the reproductions, we observed that for several subjects, the robot smoothly switched from a leader to a follower behavior. This switching suggests a complementary behavior of the human partner, *in accordance with our extended leader*-
*follower model*. This interesting result is a preliminary assessment of the relevance of this model to describe actual human behaviors.

The second part of the thesis focuses on the implementation of collaborative behaviors on a real humanoid platform. We focus on two different kinds of physical interaction. In the first setup, a telepresence system is sandwiched between two human operators located at two different locations. The telepresence system allows a human operator to project himself to a remote location to perform a physical task with an other human partner located on the teleoperator site. The second setup is centred on physical interaction between a humanoid robot and a human operator. We proposed a control architecture to perform physical interaction and collaborative transportation tasks involving whole-body motion control and locomotion. The experiments presented in this part of the thesis reveal the importance of the haptic channel to perform physical collaborative tasks: the performance of collaborative tasks performed through teleoperation with time delay is severely decreased by the lack of transparency of the system. The experiments also highlighted the point in developing advanced collaborative behaviors, such as active following, or leader/follower switching using our extended model, to allow for faster task execution with humanoid robots.

The model leader/follower switching model proposed in this thesis raises several questions which could not be answered yet. Short term objectives would include a stability study of the homotopy-based controller proposed for our extended leader-follower model. Even by selecting stable controllers for the extreme leader and follower behaviors, there is *a priori* no guarantee that any intermediate behavior obtained from a homotopy between these controllers will be stable. Furthermore, only few strategies to shape the behavior of the robot by tuning the homotopy parameter have been explored. Strategies based on the human behavior could be investigated to evaluate the concept. Finally, deeper investigations would also be needed to further assess that our homotopy switching model can describe human behavior in general haptic collaborative tasks. In the context of this thesis, the proposed extension of the leader-follower model has been investigated for simple tasks. We could apply this model to more complex setups, such as those described in the second part of the thesis, namely, cooperative telepresence and collaborative transportation tasks including biped locomotion. In the context of cooperative tasks via telepresence, role distributions could be enforced by the telepresence setup among the partners to resolve conflicts and overcome communication problems. Regarding collaborative transportation tasks, key applications would be to give a more active role to the robotic partner.

In the middle term, we could also investigate deeper how the role of the robot can be adjusted from the human behavior, going beyond simply illustrating the concept. This possibility has not been explored within this work, and it could be addressed in several different ways. One can try to identify the role of the human partner (*i.e.* his homotopy parameter) and to adopt a complementary behavior. This could be done using heuristics, or based on system identification methods. But we don't know actually what *should be* the appropriate behavior of the robot, given the knowledge of the homotopy parameter of the human partner. Moreover, the identification of the human role distribution is likely to be a very difficult task. Assuming we have thoroughly explored the different strategies to define the homotopy between the leader and follower controller for a robotic system, the final question is: *how can these strategies be used together*? How to know whether the homotopy parameter should be decided based upon the internal constraints of the

robot, or whether task signatures should have more importance than adaptations based on the partner's behavior?

Finally, we suggested several times in this thesis the existence of a haptic form of communication for collaborative tasks. We believe that because each partner participating to a physical collaborative task has his own task plan, a dialog must take place among them to converge to a common plan. This dialog must *bring the partners from a challenging stance to a collaborative stance*. How such a communication process can be established and what would be its primitives is still an open issue. Such a communication could explain how two partners can negotiate a role distribution or a common task plan. The process through which the *a priori* plan of each partner is deformed through haptic communication or identification of the other partner's intentions can be a very interesting research topic.

In the long term, physical interactions other than collaborative tasks could be addressed. Along the whole thesis, we studied the case of collaborative manipulation tasks. These are only a subset of possible physical collaborative tasks. Collaborative assembly tasks, for example, will require the coordination of very fine motions from the partners, along with force control. Such tasks will also probably require other cues than the haptic cue, and involve low-level control based on visuo-haptic feedback, as well as visuo-haptic communication among the partners. Collaborative physical tasks are also a subset of all possible physical interactions, which include dancing, involuntary contact, hugging, shoulder tapping, and many others. Each of these interactions come with low-level control aspects, but also involve social and emotional aspects in the sense of touch. When a robotic system interacts with another human, its haptic behavior must be adapted to social constraints: the inclination to physical contacts with human beings in households and offices should not be the same. Concerning emotional aspects, we can wonder how a robotic system can convey and trigger emotions from the human beings it interacts with.

# Bilateral coupling of the end-effectors for a tele-cooperation A system

The theoretical work presented in this appendix was conducted by members of the LSR at the Technische Universität München. The study is presented here for the sake of completeness. The original version of this work has been presented in more details in [84].

## A.1 Network model

#### A.1.1 Modelling the operator and environment

Two-channel bilateral teleoperation setups are generally modelled by two-port networks [36]. By considering generalized flows instead of velocities and generalized efforts instead of forces, it is possible to reason about teleoperation setups using electrical analogies.

When in use, teleoperation systems can be represented by a loaded two-port. The human operator on master site is modelled by a linear impedance  $Z_h$  and an effort source [43, 98]. The remote environment is modelled by an impedance  $Z_e$ . Fig. A.1 shows the two-port model of the teleoperation setup connected to the local and remote environments.



Figure A.1 – Two-port model of a teleoperation setup interacting with the human operator on master site and the remote environment.

#### A.1.2 Adding a second human in the loop

Our work focuses on collaborative tasks performed between two human located at two different sites, by mean of a teleoperation setup. In other terms, the teleoperation system is "sandwiched" between two humans. Thus, a human arm impedance and a force source must be incorporated in our model on the distant side in the network model. Since the environment and the human impedances are connected in series, they can be summed, resulting in an equivalent impedance  $Z_{e-rh}$ .

As the stability of a bilateral teleoperation system is conditioned on the environment impedances, considering any impedance for the local and remote environments may result in very conservative stability conditions. In reality, the impedance on both sites is bounded. This is taken into account by adding parallel impedances  $Z_h^{max}$  and  $Z_{e-rh}^{max}$  on each side of the two-port network (fig. A.2).



Figure A.2 – Two-port model of a teleoperation setup with bounded impedances at the interaction ports.

The impedance at the interaction ports are then defined by:

$$Z_h^* = \frac{Z_h Z_h^{max}}{Z_h + Z_h^{max}} \tag{A.1a}$$

$$Z_{e-rh}^{*} = \frac{Z_{e-rh} Z_{e-rh}^{max}}{Z_{e-rh} + Z_{e-rh}^{max}}$$
(A.1b)

#### A.1.3 Teleoperation interface and communication setup

Two-channel bilateral-couplings are classified according to the information exchanged between the master and slave site. Considering forces and positions, we can distinguish four architectures:

- ▷ P-P: the position of the master device is sent as a reference for a local position controller on the slave device. The position of the slave device is sent as a reference command for the master device.
- ▷ P-F: the position of the master device is sent as a reference to the slave device. The interaction force on the slave site is reflected on the master site.
- $\triangleright$  *F-P*: the force applied by the human operator on master site is sent as a reference to the slave device, and the position of the slave device is sent as a reference to the master device.
- $\triangleright$  *F-F*: the force applied by the human operator on master site is sent as a reference to the slave device, and the interaction force on slave site is reflected on the master device.

It has been shown [62] that transparency can not be achieved with P-P and F-F schemes. Thus, we will focus on the F-P and P-F schemes.

To implement F-P and P-F schemes, position and force controllers are needed, on the master or slave device, depending on the scheme. In presence of time delay,

tracking exactly the transmitted reference signals will result in instability. Intuitively, if the master device sends position references, and brings the slave device to contact with a rigid environment, the reaction force (and hence the information that contact occurred) will arrive with delay to the operator. Meanwhile, s/he can try to go further, which will dramatically increase the reaction force, since the slave device tries to strictly follow the position reference sent by the master. These significant reaction forces can either damage the slave device or the environment, or provoke a violent motion on the master side when the reaction force is transmitted to the operator.

Consequently, compliance must be added on both sides rather than strictly following the transmitted reference signals, at the cost of transparency. The effect of compliance on each side of the teleoperation system will makes environment look softer than they really are. In free space, the operator will feel damping forces that will hinder her/his motion.

## A.2 Stability analysis

So far, we have presented general concepts of bilateral coupling. We now apply these concepts to a concrete model of a teleoperation system, and present a method for the stability analysis of the system.

#### A.2.1 Model

We assume a linear teleoperation system, with position controlled master and slave devices. We also assume that the local controllers on the master and slave devices have significantly faster dynamics than the global coupling. Thus, we assume that the reference forces and velocities sent to the low-level controllers of the platforms are perfectly tracked. Lets us know determine the expressions of the hybrid matrix of the system for the *F*-*P* and *P*-*F* architectures.

In the case of the *F-P* scheme, reference forces are sent to the slave platform. The usual way to implement force control on positioned control mechanisms is admittance control. The slave platform must also be compliant to environment forces for stability reasons, as stated above. Hence, the reference force and environment force (measured by a sensor) are summed, and filtered by an admittance,  $Y_s^{fp}$ . The output of the admittance is a reference velocity that we integrate to obtain a reference position  $x_s^{fp}$ . We assume this position is perfectly tracked by the low-level controller of the slave device. The slave position is sent to the master site as a reference. A position error is computed by filtering the sensed force (which is sent to the slave site) by an admittance  $Y_m^{fp}$ . The position sent to the low-level controller is the sum of the reference position from the slave site and the position error. This is summarized on fig. A.3.

In the P-F scheme, the local controllers on each side also include admittances, but the exchanged signals are not the same: reference velocities are sent to the slave device, while the distant forces are fed back to the master device, see fig. A.4.

#### A.2.2 Stability criterion

Assuming all the components of the system are linear, the network model can be characterized by a hybrid matrix. Different matrices can be chosen, depending on which variables are considered independent, and which variables are seen as the



Figure A.3 – FP control architecture.



Figure A.4 – *PF control architecture*.

'input' of the system. However, the term 'hybrid' matrix refers to the case where both input variables are not homogeneous, *i.e.* one is a generalized effort and one is a generalized flow.

The stability of linear two-port can be studied using Llewelyn's absolute stability criteria [68]. An absolute linear two-port network is stable for any passive termination. Necessary and sufficient conditions on the hybrid matrix H for absolute stability are as follows [1,68,84]:

$$Re(h_{11}) > 0$$

$$Re(h_{22}) > 0$$

$$2 Re(h_{11})Re(h_{22}) - Re(h_{12} h_{21}) - |h_{12} h_{21}| > 0$$
(A.2)

where  $h_{ij}$  are the elements of the immitance matrix characterizing the two-port. For our analysis, we will select the matrix *H* defined by

$$\begin{bmatrix} F_m \\ -V_s \end{bmatrix} = H \begin{bmatrix} V_m \\ F_s \end{bmatrix}$$
(A.3)

for the P-F architecture, and the matrix G defined by

$$\begin{bmatrix} V_m \\ F_s \end{bmatrix} = G \begin{bmatrix} F_m \\ -V_s \end{bmatrix}$$
(A.4)

for the *F*-*P* architecture, where  $F_m$ ,  $V_m$ ,  $F_s$  and  $V_s$  are the generalized efforts and flows at the interaction ports of the network. This choice reflects the causality of each architecture, the input variables being the transmitted ones.

The derivation of matrices H and G is straightforward using network analysis and admittance association rules together with Kirchhoff's circuit laws. To ease the derivations, it is useful to represent the complete architecture as a network system, assuming that both master and slave platforms can be controlled to appear as pure admittances, as in fig. A.5. Since the form of matrix H has been given in detail in [84], we give here the complete form of matrix G.



Figure A.5 – Network representation of the F-P architecture.

Using the notations of fig. A.5, we have the following expressions for the elements of G:

$$g_{11} = \frac{1}{Z_h^{max}} + \frac{e^{-T_{sm}s} e^{-T_{ms}s}}{Z_{e-rh}^{max} + \frac{1}{Y_s^{fp}}} + Y_m^{fp}$$

$$g_{12} = -\frac{e^{-T_{sm}s} Z_{e-rh}^{max} Y_s^{fp}}{1 + Z_{e-rh}^{max} Y_s^{fp}}$$

$$g_{21} = \frac{e^{-T_{ms}s} Z_{e-rh}^{max} Y_s^{fp}}{1 + Z_{e-rh}^{max} Y_s^{fp}}$$

$$g_{22} = \frac{Z_{e-rh}^{max}}{1 + Z_{e-rh}^{max} Y_s^{fp}}$$
(A.5)

The expression of the admittances  $Y_{\{s,m\}}^{\{fp,pf\}}$  is given by the general expression:

$$Y = \frac{s}{m\,s^2 + b\,s + k} \tag{A.6}$$

and the impedance bounds  $Z_h^{max}$  and  $Z_{e-rh}^{max}$  have the following general form:

$$Z = \frac{b \, s + k}{s} \tag{A.7}$$

#### A.2.3 Numerical analysis

To select the admittance gains on the local and distant site, the Llewelyn conditions (A.2) are tested for different sets of parameters. To reduce the size of the parameter space, the parameters  $m_m$ ,  $k_m$ , and  $k_s$  are fixed *a priori*. The master mass parameter is selected to keep the system stable. On the master site, the stiffness parameter is set to  $k_m = 600$  N/m, and on the remote side, it is set to  $k_s = 0$  N/m. The time delay is assumed to be constant, and the parameters are set to  $T_{ms} = T_{sm} = 0.15$  s. The other parameters are gridded and the absolute stability is tested for each set. The results are shown on fig. A.6 for the *P*-*F* architecture, considering interaction with a stiff environment. The stable region is inside the envelope. The stable area increases with the slave inertia and damping.



Figure A.6 – *Stability envelope for the* P-F *architecture.*  $m_m = 5 kg$ .

The F-P architecture detailed in the previous paragraph leads to a wider range of stable parameters, as shown on fig. A.6. Starting from a minimal value of the slave damping, the coupling is always absolutely stable. Hence, this second scheme was implemented for the experimental evaluation.



Figure A.7 – Stability area for the F-P architecture.  $m_s = 5 kg$ .

# A study of inter-trial adaptation in dyadic collaborative lifting tasks

The work presented in this appendix has been performed in collaboration with Satoshi Endo and Alan Wing, from the University of Birmingham, in the context of the European project ImmerSence(www.immersence.info). It focuses on behavioral changes of the partners in response to errors in the previous trials of a collaborative lifting task.

# **B.1** Problem statement

During a dyadic collaborative lifting task where the attitude of the lifted object is to be maintained, e.g. the object must be kept horizontal, both partners must synchronize their motion so that both ends of the object are lifted at the same time, and with the same velocity. Synchronization errors will result in changes in the orientation of the object, which violates the constraint of the task.

We are interested in knowing how two human partners can use the error knowledge of one performance to adapt their behavior for the next trial. Ideally, if the lifting task is repeated again and again, the partners will adapt their behavior across trials to converge towards a synchronous motion. More specifically, we want to investigate how the *haptic channel* can be used by the partner to adapt their behavior across trials.

A common problem when studying human-human collaborative tasks is the variability of the human behavior. Due to this variability, it is difficult to reproduce twice the same condition to evaluate particular hypotheses about the behavior of one human partner. A possible solution is then to replace one of the human partner by a robot, which allows the exact reproduction of specified behaviors.

# **B.2** Experimental setup

We propose to study the adaptive behavior of human beings in collaborative lifting tasks by asking human subjects to perform such tasks in cooperation with a humanoid robot HRP-2. Fig. B.1 shows the experimental setup. Only the six degrees-of-freedom of the right arm were used in this experiment. The robot arm is controlled using the Stack of Tasks, presented in chapter 4 and 5. A high priority task was used to constrain the attitude of the robot's gripper. A low priority task was used to produce a vertical lifting motion (we ensured no motion was produced in the horizontal plane, to obtain a one degree-of-freedom motion).

Motion trackers (Ascension miniBirds) were installed on the manipulated object to track its motion. The data was acquired on a computer which was placed near the human partner. This computer displayed the maximal angle of the object in each trial to give an idea of the global performance to the subject. A value of zero meant that the subject performed the task in perfect synchronization with



Figure B.1 – *Experimental setup: collaborative lifting task between an HRP-2 humanoid robot and a human subject.* 

the robot. The computer located near the partner was controlled remotely using a CORBA client-server architecture so that the experimenter could control the overall experiment and check that validity of the data after each trial.

# **B.3** Method

Ten right-handed participants (age  $26.5 \pm 3.47$  years) sat at a table and grasped the handle at one side of the object. At the opposite side, the robot grasped another handle. The object had a symmetric mass distribution and a mass of about 1 kg. Because the actuators of the robot produce a lot of noise, white noise was presented to the participants using headphones so that they could not rely on the sound of the joint actuators to synchronize their motion with the robot. To avoid latencies at the beginning of the motion, a starting auditory signal was given after three fixed paced beeps. As the beeps started, the partners were asked to close their eyes, and could only open them again at the end of the motion to check their final altitude with respect to the robot, and the performance measure displayed by the computer nearby.

# **B.4** Adaptation law

In a first step, we want to assess an analytical form for an adaptation law from the human partner. In a first series of experiments, we played minimum jerk trajectories on the robot with different peak velocities. The motion was played in an open-loop manner and the robot did not adapt across trials.

An analysis of the data revealed a correlation between the peak velocity differences between the partners in one trial and the peak velocity difference for the human partner between this trial and the next one. Hence, the following adaptation law is proposed:

$$Z_{n}^{A} = Z_{n-1}^{A} + \alpha^{A} \left( Z_{n-1}^{B} - Z_{n-1}^{A} \right)$$
  

$$Z_{n}^{B} = Z_{n-1}^{B} + \alpha^{B} \left( Z_{n-1}^{A} - Z_{n-1}^{B} \right)$$
(B.1)

where  $Z_n^A$  and  $Z_n^B$  are the respective peak velocities of partners A and B during trial n, and  $\alpha^A$  and  $\alpha^B$  are positive adaptation gains. Given appropriate values for  $\alpha^A$  and  $\alpha^B$ , the peak velocities of partners A and B will converge to a common value.

## **B.5** Optimal adaptive behavior

In a second series of experiments, the adaptation law (B.1) was implemented on the robot, with different sets of adaptation gains  $\alpha_R$ . The gain of the robot was either high ( $\alpha^R = 0.8$ ), intermediate ( $\alpha^R = 0.5$ ), low ( $\alpha^R = 0.2$ ), zero, or randomly changed between 0 and 1 every trial, with a resolution of 0.1. Two different behaviors can be expected. Either the human partner will adapt to the gain adopted by the robot, or he will keep a consistent value for all trials.

This series of experiments was conducted with 10 partners who performed 15 trials for each condition (zero, low, intermediate, high and random adaptation gains). The results show that the adaptation gain  $\alpha^H$  of the human varied with the gain of the robot,  $\alpha^R$ , in such a way that  $\alpha^H + \alpha^R \approx 1$  ( $r \approx 0.892$ ).

Let us show that this correspond to an optimal behavior. The system (B.1) can be rewritten as:

$$\mathbf{z}_n = \mathbf{A} \, \mathbf{z}_{n-1} \tag{B.2}$$

with

$$\mathbf{A} = \begin{bmatrix} 1 - \alpha^A & \alpha^A \\ \alpha^B & 1 - \alpha^B \end{bmatrix}$$
(B.3)

and

$$\mathbf{z}_n = \begin{bmatrix} Z_n^A \\ Z_n^B \end{bmatrix}$$
(B.4)

The equation B.2 defines a geometric sequence which general term is:

$$\mathbf{z}_n = \mathbf{A}^n \, \mathbf{z}_0 \tag{B.5}$$

The convergence of such a sequence is a classical result. Let us study it for our case. A is diagonalizable, with eigenvalues  $\lambda_1 = 1$  and  $\lambda_2 = 1 - (\alpha^A + \alpha^B)$ . Hence, equation B.5 can be rewritten:

$$\mathbf{z}_n = \mathbf{P} \, \mathbf{D}^n \, \mathbf{P}^{-1} \, \mathbf{z}_0 \tag{B.6}$$

where  $\mathbf{P}$  is a 2x2 matrix of eigenvectors, and

$$\mathbf{D} = \begin{bmatrix} 1 & 0\\ 0 & 1 - (\alpha^A + \alpha^B) \end{bmatrix}$$
(B.7)

From equations B.6 and B.7, it appears that the convergence of  $(\mathbf{z})_n$  depends only on  $(1 - (\alpha^A + \alpha^B))^n$ . If  $\alpha^A + \alpha^B < 2$ , then  $\mathbf{z}_n$  will converge to  $\mathbf{z}_f = \mathbf{P} \mathbf{D}_f \mathbf{P}^{-1} \mathbf{z}_0$ , where

$$\mathbf{D}_f = \begin{bmatrix} 1 & 0\\ 0 & 0 \end{bmatrix} \tag{B.8}$$

- ▷ If  $\alpha^A + \alpha^B = 1$ , then  $\mathbf{z}_f$  is reached from the first iteration (n = 1). Hence the convergence is the fastest.
- ▷ If  $1 < \alpha^A + \alpha^B < 2$ , the term  $1 (\alpha^A + \alpha^B)$  will be negative, and  $(1 (\alpha^A + \alpha^B))^n$  will be alternatively positive and negative, depending on whether *n* is even or odd; but still converges to 0. Hence  $(\mathbf{z})_n$  asymptotically converges to  $\mathbf{z}_f$ , but the system will exhibit an oscillatory behaviour.
- ▷ If  $0 < \alpha^A + \alpha^B < 1$ , the term  $(1 (\alpha^A + \alpha^B))^n$  will converge to 0 and hence  $(\mathbf{z})_n$  asymptotically converges to  $\mathbf{z}_f$ , and the convergence is monotonic.
- ▷ For the limit cases where  $\alpha^A + \alpha^B = 0$  and  $\alpha^A + \alpha^B = 2$ , the system is marginally stable.  $(\mathbf{z})_n$  is bounded but does not converge, and indefinitely oscillates.
- ▷ If  $\alpha^A + \alpha^B > 2$ , the system is unstable:  $(\mathbf{z})_n$  diverges. As the gains are positive, we don't consider any other case.

Hence, the fact  $\alpha_H + \alpha_R \approx 1$  shows that the human behavior will change his adaptation gain so that the robot and him will quickly converge to a same velocity, the fastest convergence being obtained for  $\alpha_H + \alpha_R = 1$ .

# **B.6** Conclusion

In this appendix, we have studied an adaptation behavior of human beings in dyadic collaborative lifting tasks where the attitude of the lifted object is supposed to remain constant. The synchronization of the motions of both partners is very important for the task to be successfully realized. We presented a setup were one of the human partners is replaced by a robotic one. We proposed an adaptation law to describe the evolution of the peak velocities of the motion across trials. Experimental data suggests that this adaptation law is valid and that the adaptation gain of the human partner is adapted to the one of the robot to reach optimal convergence of the system.

# Development of an interactive simulation framework with haptic rendering

# C.1 Introduction

This appendix focuses on an integrative software architecture for dynamic simulation centred on haptic interaction with virtual avatars. This framework is devised to integrate developments in digital actors control with a focus on haptic tasks and communication. The framework integrates and extends the work presented in [59], where the implementation of constraint-based simulation including haptic interaction with virtual avatars through a haptic device is described.

The framework presented in this appendix intends to include not only taskdriven simulations, but also cognitive aspects linked to haptic interaction such as haptic communication and advanced interaction with digital actors that can be either virtual or real (robots). It is devised to allow fast prototyping of virtual reality experiments, such as the ones described in the chapter 2 of this thesis.

In this appendix, we mainly focus on the software design aspects of the framework rather than on the constraint-based dynamics or collision detection algorithms. A previous version of this framework is described in [24].

# C.2 Software architecture

#### C.2.1 Presentation of Amelif

In software engineering, a framework is a set of interfaces that describe how software components interact. A framework defines or imposes constraints on the control flow of an application, and lets the user specialize, define or override specific operations that are part of the control flow. A framework can be provided together with a library (a set of useful generic components, data structures, and tools).

Amelif is both a library and a framework. It provides useful pieces of code that can be reused to perform usual operations. The use of these components rather than handcrafted ones will increase homogeneity in the developments of the laboratory and lower the maintenance cost. Amelif also provides an application to run Virtual Reality (VR) simulations with a specified control flow, and interfaces for algorithms related to VR. Developing within Amelif should result in code that is fully integrated and that can interact with all the other components of the framework.

Amelif is developed in C++, and most components are portable. Some packages, due to some of their dependencies, might not work under all platforms though. This is the case of the OpenHaptics wrapper, which requires either Windows, or a SuSe or Red Hat distribution, with a 2.4 Linux kernel.

#### C.2.2 Basic components

This section briefly introduces the different components and libraries that have been developed within Amelif. Some of them are described in more details in [24].

#### **Core library**

Amelif is built upon a core library which regroups several generic components: small vectors and matrices, an implementation of the spatial algebra proposed in [28], generic XML parsers, generic data structures and so on. The core library also provides an applicative framework to quickly design multi-threaded applications based on any GUI library of choice. Finally, it also provides an extensible command line system which allows to register any object to a console so as to interact with it through a command prompt. An adapter class is provided so that this mechanism is not intrusive and existing code need not be modified to be interfaced with it.

#### **Basic application**

A basic GLUT application is provided as an additional package to avoid the redevelopment of repetitive GUI-related code. This application provides only basic functionalities, namely, a GL rendering window, and a command prompt interfaced with the console mechanism offered by the core library. A more sophisticated application existed in earlier versions of the framework and will be ported to the current version of the framework.

#### **Scene renderer**

A simple OpenGL rendering library has also been developed as an optional component of Amelif. This library includes wrapper classes around OpenGL, as well as scene graph classes which make use of template meta programming to avoid code repetition and ease the extension of the library.

#### State

The virtual scene is represented by a set of classes which store the state of each element of the scene. XML parsers are also provided to parse XML files where virtual scenes can be described.

#### **Engine and applicative framework**

The core library of Amelif provides an engine and an applicative framework. The applicative framework together with the engine define a specific control flow which allows to instantiate programs and attach them to threads. Programs are actually instances of an abstract interface which implement initialization and cleanup procedures, as well as the content of a simulation loop.

#### **Collision detection**

The collision detection module provides interfaces to trigger the detection of collisions between sets of bodies of the scene. A user has the possibility to gather different sets of bodies and to detect collisions among these sets independently. This can be useful when the virtual scene includes elements that are totally separated. Among the groups, it is possible to associate flags to pairs of bodies. These flags will determine which data is required from the collision detection low-level library about the specified pair of bodies (proximity distance, interpenetration...). Finally, it is possible to exclude pairs from a set of bodies, or to add pairs and detect collisions among the specified pairs only.

#### **Device management**

The devices library manages external devices such as haptic devices. It provides a high-level interface to get information about various devices that can be used within Amelif. The haptic device library possibly uses the collision detection and the dynamic simulator to handle interactions between the haptic probe and the virtual environment. This dependency can be reversed if the haptic device is provided with its own collision detection library and the ability to compute the feedback force; in this case, the dynamic simulator would rely on the haptic device to take into account the interaction force between the haptic tool-tip and the environment for the computation of direct dynamics.

#### **Dynamic simulator**

The dynamic simulator provides interfaces to drive a dynamic simulation and to run usual algorithms related to dynamics (direct dynamics, computation of the operational space inertia matrix...). The dynamics simulation is responsible for computing accelerations from the forces applied on bodies and for resolving algorithms to handle contacts, impacts, deformations, etc. It is also responsible for integrating the computed accelerations and updating the state of the scene. This module depends on the collision detection for bodies and articulated bodies not to pass through each other.

The collision and dynamics modules can communicate with other modules in two ways: they can either be used as usual components that are queried on demand; or they can be used as publishers that will send events to their listeners, according to the well-known Observer (or Publisher/Subscriber) design pattern. This way of communicating with other modules has a strong advantage upon the usual queryon-demand communication as it allows a clear and automatic separation between algorithms and code that handles output data types when polymorphism can not easily be used (a typical case in communication among modules). Client code can also be made aware of new output information automatically, whereas with queryon-demand, the user needs to read the documentation of the queried module to keep aware of new outputs. This mechanism is used by the collision detection module to send information about detected collisions to all the subscribers modules, which can either handle the collision as soon as it is received or store the information for future use. Likewise, the dynamic simulator publishes interactions, which are collisions augmented with a contact force information. A publisher module has the responsibility for defining the interface of its subscribers. Thus, the publisher module has control over the data that a client module can receive and can or must

handle. A publisher module is aware of the existence of its clients and thus can have control over them while being independent of the specific type of its client.

# C.3 Demonstrator

To illustrate the advantages of our software architecture and the capabilities of our simulator, we developed a small demonstrator in which a user interacts at different levels with a virtual avatar. We also introduce some modules that are currently under development and being integrated to Amelif: the Haptic Perception module and the Visual Perception module. The goal of this section is to emphasize the possibilities offered by our framework for prototyping algorithms centred on interaction with virtual avatars.

#### C.3.1 Scenario

In this demonstration, a virtual HRP-2 humanoid robot stands in front of a table upon which lies an object. The robot first goes to a predefined initial posture, and then stands idle. The user touches the robot with a haptic probe; the robot is triggered by this action and stares at the body that was just touched. The robot then sees the haptic probe near the touched body (if the haptic probe is in its field of view). It thus starts to stare at the probe. The user moves the probe toward the object on the table and just waits aside. After a given amount of time during which the probe is close to the object, the robot starts focusing on the object and grasps it. Once the object is grasped, the robot's arm is compliant to forces applied on the object, which allows the user to move the object to any configuration together with the robot.

Let us examine how such a behavior can be implemented under Amelif and how, starting from this implementation, virtual prototyping for Virtual Realityrelated algorithms centred on human-avatar interaction can easily be realized.

#### C.3.2 The virtual HRP-2 avatar

The highest level component of the demonstrator is the HRP-2 avatar. This component will implement a finite state machine that will drive the avatar into different states. In each state, a different set of behaviors are activated. From the definition of the scenario, five states can be drawn: the "going to initial position" state, the "idle" state, the "focus on cursor" state, the "grasping phase" state and the "task" state. As the scenario is very simple, these states are just sequential. Once we leave a state, we can never come back to this state. This make the implementation of the Finite State Machine very easy.

The "going to initial position" state is straightforward and just relies on controllers that will servo the joints to follow specified trajectories defined off-line. The "idle" state is just a state where the robot keeps its current position and can just rely on a simple PD controller.

#### C.3.3 Dynamic simulation

The very first thing to do is to simulate the virtual environment. Amelif provides default implementations for the visual and haptic rendering of a scene, low-level haptic interaction with free-bodies and constraint-based dynamic simulation with



Figure C.1 – Structure of the demonstrator

frictions. All we have to do is to load the modules, parse XML files in which the environment is described using the parser provided with the framework, and run the components in the main program.

#### C.3.4 Focus on the haptic probe

The "idle" state is left when the robot is touched. Therefore, we need to let the HRP-2 avatar know when something touches it. Typically, this can be implemented with the Skin component, which is part of the Haptic Perception module. This module is currently under development and proposes interfaces and components related to haptic perceptions (various force sensors, skins...). The Haptic Perception module is implemented as a subscriber of the dynamic simulator. Indeed, the simulator outputs collisions and their associated interaction forces and bodies. When a virtual avatar is registered in the Haptic Perception module as having a skin, the Haptic Perception module will trigger it each time this avatar is involved in a collision, and will send it the associated interaction force. This communication between the Haptic Perception module and the avatar is realized through a Skin component, which allows users to model various phenomena related to the perception of forces by the skin.

The "focus on cursor" state can be implemented in a similar way. Instead of the Haptic Perception module, we use a Visual Perception module. This module allows an avatar to be triggered on different conditions. The simplest way of triggering the avatar is to associate a Field of View (FoV) component to the avatar. A FoV component raises messages when objects of the scene that have been registered as visible objects in the Visual Perception Module enter and leave the field of view. We have the possibility to easily add uncertainty about the position of the objects in the field of view by adding an occluding component between the FoV and the avatar.



Figure C.2 – *The different steps of the simulation: (a) Go to initial configuration (b) Trigger the robot by patting its hand (c) Show it the object (d) Move the object together* 



Figure C.3 – Interacting with the HRP-2 avatar using a haptic device.

In our demonstration, we did not add any uncertainty and considered that the avatar was able to perfectly determine the position of objects in his field of view. Using a very simple controller, we servo the joints of the robot's neck so as to stare at the cursor when the cursor is inside the field of view. When both the haptic probe and the object on the table are in the field of view, the avatar looks at the distance between them and if it stays under a given threshold during a given amount of time, the "focus on cursor" state is left and the "grasping phase" state is entered.

In our demonstration, the grasping phase is implemented using pre-defined joint trajectory. However, using the jacobian computation algorithm provided in the scene library, an inverse kinematics algorithm could be used to automatically reach the object at any location.

#### C.3.5 Task

In the "task" state, the virtual HRP-2 was position-controlled using a PD joint controller. To make the robot compliant to the interaction force with the haptic device, the desired joint positions  $q_d$  are defined by the following equation:

$$M\ddot{\mathbf{q}}_d + B\dot{\mathbf{q}}_d = \mathbf{J}_e^T \mathbf{f}_e \tag{C.1}$$

where M and B are diagonal matrices of positive coefficients corresponding to a virtual inertia and a virtual damping,  $\mathbf{f}_e$  is the interaction force with the haptic device, and  $\mathbf{J}_e$  is the Jacobian relating joint velocities to Cartesian velocities and angular velocities for the kinematic chain that starts from the root body of the avatar (here, the waist of the robot) and ends at the body touched by the haptic device. To get the haptic device force, we either need to subscribe to the dynamic simulator and listen for interactions between the haptic device and the environment or to directly query the haptic device manager. For our demonstration, we chose the first solution.

#### C.3.6 Results

Figure C.1 shows the final architecture of the demonstrator, including components that are part of the framework and components that are specific to the demonstrated scenario. The relationship between the different components and modules is represented by arrows. The only components that we need to write to implement this scenario are the controllers used by the avatar as well as the higher-level behavior encapsulated in the Avatar component. We ran the demonstration on a computer equipped with a PHANToM<sup>©</sup> Premium 1.5 High Force<sup>TM</sup>. The robot indeed reacted according to the scenario. Figure C.2 shows screenshots of the different states of the scenario. The translucid, light blue pyramid represents the field of view of the robot. The haptic probe is represented by the orange bar. The setup of the demonstration is shown on figure C.3. Although most parts of the demonstrator use naive and simple implementations, it lays the basis for more complicated simulations where sensor noise for the haptic and visual perception can easily be integrated. A user who focuses on the haptic interaction during the task phase and does not want to start from scratch can use such a demonstrator and focus on the task behavior without caring about the other components, while other users can at the same time work on aspects related to vision. The integration of their own components to the demonstrator is easily realized if the users comply to the proposed interfaces. The overall amount of code for this simple, skeletic demonstrator is rather small and the development cost for the demonstrator was no more

than a couple of person-hours (not including the haptic and visual perception modules which are provided to the user). The demonstrator is robust to changes in the modules on which it depends; the implementation of the dynamics simulator and collision detection modules used by this simulator would have no influence on the demonstrator's code. A user can replace one of these modules by any other implementation by changing only two lines of the demonstrator's code. Any haptic device can be integrated to Amelif; replacing the haptic device used in the demonstrator by another integrated haptic device would then require to change only two lines of code.

# A PHANTOM<sup>®</sup> Device with 6DOF D Force Feedback and Sensing Capabilities

This appendix describes a technical solution that has been devised to embed the SensAble PHANTOM<sup>®</sup> 6dof force feedback device with a light-weight 6dof force sensor from ATI: the Nano43. The design has been made in a way such that original performances are kept in terms of force feedback and sensing.

# **D.1** Introduction

At CNRS, in the frame of the ImmerSence project<sup>1</sup>, we aim at achieving a demonstrator with various modelling of the interaction between two persons through an intermediary object (person-object-person, acronymed POP in the remaining). Examples where such an interaction applies, are collaborative transportation and assembly tasks, virtual prototyping, etc. Models will be built upon knowledge acquired from collaborative robotics, cognitive science as well as the use of haptic and multimodal patterns of communication. These models, integrated together will allow us to provide avatars with a set of realistic behaviors during the realization of POP tasks. The POP tasks demonstrated can be carried out with different configurations: either two persons manipulating an object in a shared virtual or mixed (i.e. augmented reality) environments or a person performing a collaborative task with a virtual avatar.

Preliminary investigations on this scenario [59] using desktop haptic devices, such as the PHANTOM Omni<sup>®</sup>, revealed limitations in performing interactive POP tasks because of the lack of torque feedback. Moreover force and torque sensing will allow a better haptic patterns identification and interpretation. These supposed *haptic patterns* can ideally be used as bricks of an hypothetical *haptic language* which, in turn, can be used at a higher level control of virtual avatars interacting with users through direct touch (pure communicative aspects) or through a virtual object manipulation (functional tasks aspects). Therefore, it appeared important to conceive a 6dof<sup>2</sup> haptic sensing/display device.

This appendix presents a technical solution which allowed to mount a 6dof force sensors from ATI<sup>3</sup> (the Nano43) on a SensAble PHANTOM 6dof force feedback device (High Force 1.5). The obtained system is an integrated 6dof force sensing/display device that is aimed for our research and development in haptic interaction in POP application scenarios.

In addition to the previous concern, there are other reasons and benefits gained by a force display embedded with force sensor:

<sup>&</sup>lt;sup>1</sup>http://www.immersence.info/

<sup>&</sup>lt;sup>2</sup>dof, states for Degrees Of Freedom.

<sup>&</sup>lt;sup>3</sup>http://www.ati-ia.com/

- it is possible to have a direct measure of the resultant wrench acting on the device without noise and computations/approximations that are consequent to collecting these data from an identification algorithm which uses the motors' torques (in direct drive case) or current measures;
- force sensor can be used directly in admittance-type feedback (i.e. when forces are obtained from constrained-based methods, it is more robust and stable to know actual external forces applied by the user on the manipulated object (proxy), so that they are directly integrated in the computation of the dynamics);
- characterization of haptic texture rendering as presented in [13] where a force sensor from ATI (the Nano17 model) has been mounted on a 3dof force feedback from SensAble.

The remainder of the appendix will describe the prototype we devised and that is made by SensAble Technologies to provide a solution that satisfies our requirements.

# **D.2** Technical solution

At first we envisaged using the same mounting solution that has been proposed in [13]. However given our specific requirements we favored a solution where the force sensor is mounted between the handle and the remaining mechanical linkage of the force feedback display. The reason for this is to be as close as possible to the interaction point between the user and the force display. Nevertheless, because of the presence of an actuator within the handle which ensures torque feedback along the handle's axis; the armature of the handle's actuator (i.e. external cover) is attached to the handle where as its shaft is statically linked to the hammerhead and subsequently, to the remaining mechanical linkage of the PHANTOM 6dof. Therefore, it was not possible to use a small and plain force sensor such as the Nano17 ATI force sensor.

In addition, secondary but important constraints which guided the design, is to have a reasonable price solution, which means:

- ▷ using commercially available products,
- ▷ keep performances similar to the original products, and
- ▷ limit the possible modifications/adaptations.

Therefore, the adopted technical solution allowing to customize the PHAN-TOM 6dof with a 6dof sensing capability close to the hand is as follows:

- make use of an available PHANTOM Premium 1.5 High Force/6DOF device, although the solution would also apply to other PHANTOM 6dof devices;
- make use of an ATI Nano43 with DAQ F/T transducer: this force sensor model has a through hole which allows to bypass the difficulty of changing the handle and the attachment mechanism;
- ▷ A modified handle to accommodate the ATI force sensor.



Figure D.1 – Mechanical parts of the new SensAble 6dof stylus with force feedback integration.

Note that the solution is in fact using two commercially available products; the problem remains only in the mechanical design of the handle (named also stylus) in order to accommodate the ATI sensor without extensively changing the original mechanical design. The sensor is mounted closer to the pivoting gimbals and thereby closer to the original end-point of the force feedback device (hammerhead). This provides more accurate measurement of the forces and torques exerted by the user. The position of the control switch button was moved behind the sensor so that the user can easily reach it during operation.

The resulting new handle is illustrated under different views in the figure D.2. The overall new device is illustrated in the figure D.3.

In this version the additional weight (the force sensor weighs only 40g) induced by the force sensor is negligible and is mechanically compensated by the counterweight balance used originally by the display. The PHANTOM 6DOF device with the new customizable handle was tested and adjusted to take up to 50g of weight in the handle without losing performance. This was done for the 7DOF extensions for the new handle (scissors/pinch) which weight approximately 50g and are already commercially available. The apparent inertia can be compensated in the software by using a simple closed-loop gravity compensation; this will be used in future work that makes usage of this solution in our applications.

### **D.3** Conclusion

This appendix presented a technical solution for the design of A PHANTOM device with 6dof force feedback display and sensing capabilities. The design allowed keeping the performances of both the PHANTOM device and the force sensor. It will be used in further investigations for research on desktop human/virtual avatars interaction.



Figure D.2 – *Different views (left side and isometric) of the new PHANTOM 6dof handle with the 6dof Nano43 force sensor from ATI.* 



Figure D.3 – A 6dof PHANTOM device with 6DOF sensing sensor from ATI (Nano43) and the new handle.

# Bibliography

- [1] Richard J. Adams. *Stable Haptic Interaction with Virtual Environments*. PhD thesis, University of Washington, 1999. (Cité page 132.)
- [2] Robert J. Anderson and Mark W. Spong. Bilateral Control of Teleoperators with Time Delay. *IEEE Trans. on Automatic Control*, 34:494–501, 1989. (Cité page 83.)
- [3] Hirohiko Arai, Tomohito Takubo, Yasuo Hayashibara, and Kazuo Tanie. Human-robot cooperative manipulation using a virtual nonholonomic constraint. In *IEEE Int. Conf. on Robotics and Automation*, pages 4064–4070, San Francisco, CA, April 2000. (Cité pages 5, 17, 18 et 25.)
- [4] Tamim Asfour, Karsten Berns, and Rüdiger Dillmann. The humanoid robot armar: Design and control. In *in IEEE/APS Intl Conference on Humanoid Robots*, pages 7–8, 2000. (Cité page 3.)
- [5] Antal K. Bejczy, Won S. Kim, and Steven C. Venema. The Phantom Robot: Predictive Displays for Teleoperation with Time Delay. In *IEEE Int. Conf.* on Robotics and Automation, 1990. (Cité page 83.)
- [6] A. Ben-Israel and T. Greville. *Generalized Inverses: Theory and Applications*. CMS Books in Mathematics. Springer, 2nd edition, 2003. (Cité page 89.)
- [7] Aude Billard, Sylvain Calinon, Rüdiger Dillmann, and Stefan Schaal. Robot Programming by Demonstration. In *Handbook of Robotics*, volume chapter 59. MIT Press, 2008. (Cité pages 58 et 59.)
- [8] Stephen P. Buerger and Neville Hogan. Complementary Stability and Loop Shaping for Improved Human-Robot Interaction. *IEEE Transactions on Robotics*, 23:232–244, 2007. (Cité page 19.)
- [9] S. Calinon and A. Billard. Active teaching in robot programming by demonstration. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 702–707, August 2007. (Cité page 59.)
- [10] S. Calinon, F. Guenter, and A. Billard. On learning, representing and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man* and Cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation, 37(2):286–298, 2007. (Cité pages 59 et 68.)
- [11] Sylvain Calinon, Paul Evrard, Elena Gribovskaya, Aude Billard, and Abderrahmane Kheddar. Learning collaborative manipulation tasks by demonstration using a haptic interface. In *IEEE Int. Conf. on Advanced Robotics*, Munich, Germany, June 2009. (Cité pages 56, 57, 58, 69 et 70.)
- [12] Joel Chestnutt. Navigation Planning for Legged Robots. PhD thesis, The Robotics Institute, Carnegie Mellon University, 2007. (Cité page 105.)

- [13] S. Choi and H. Z. Tan. Perceived instability of virtual haptic texture: Ii effect of collision detection algorithm. *PRESENCE*, 14(4):463–481, August 2005. (Cité page 148.)
- [14] Ed Colgate. An analysis of contact instability in terms of passive physical equivalents. In *IEEE Int. Conf. on Robotics and Automation*, 1989. (Cité pages 15 et 20.)
- [15] James Edward Colgate and Neville Hogan. Robust control of dynamically interacting systems. *International Journal of Control*, 48(1):65–88, 1988.
   (Cité pages 5, 7, 8, 12 et 13.)
- [16] B. Corteville, E. Aertbelien, Herman Bruyninckx, Joris De Schutter, and Hendrik Van Brussel. Human-inspired robot assistant for fast point-to-point movements. In *IEEE Int. Conf. on Robotics and Automation*, pages 3639– 3644. IEEE, 2007. (Cité pages 5, 6, 18, 20, 22, 23, 24, 25, 26, 34, 35, 44 et 122.)
- [17] J.J. Craig and M.H. Raibert. A systematic method of hybrid position/force control of a manipulator. In *IEEE Computer Society's Third International Computer Software and Applications Conference*, 1979. (Cité page 7.)
- [18] K. Doty, C. Melchiorri, and C. Bonivento. A Theory of Generalized Inverses Applied to Robotics. *Int. J. Robotics Research*, 12(1):1–19, 1993. (Cité page 104.)
- [19] Vincent Duchaine and Clement Gosselin. Safe, Stable and Intuitive Control for Physical Human-Robot Interaction. In *IEEE Int. Conf. on Robotics and Automation*, 2009. (Cité page 19.)
- [20] Vincent Duchaine and Clement M. Gosselin. General model of humanrobot cooperation using a novel velocity based variable impedance control. In *Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 446–451, Washington, DC, USA, 2007. IEEE Computer Society. (Cité pages 18 et 23.)
- [21] Vincent Duchaine and Clement M. Gosselin. Investigation of Human-Robot Interaction Stability Using Lyapunov Theory. In *IEEE Int. Conf. on Robotics and Automation*, 2008. (Cité page 19.)
- [22] Steven D. Eppinger and Warren P. Seering. Understanding Bandwidth Limitations in Robot Force Control. In *IEEE Int. Conf. on Robotics and Automation*, 1987. (Cité page 15.)
- [23] Adrien Escande, Sylvain Miossec, and Abderrahmane Kheddar. Continuous gradient proximity distance for humanoids free-collision optimizedpostures. In *IEEE RAS/RSJ Conference on Humanoids Robots*, Pittsburg, USA, 2007. (Cité page 46.)
- [24] P. Evrard, F. Keith, J-R. Chardonnet, and A. Kheddar. Framework for haptic interaction with virtual avatars. In 17th IEEE Int. Symposium on Robot and Human Interactive Communication (Ro-Man 2008), Munich, Germany, August 2008. (Cité pages 139 et 140.)

- [25] Paul Evrard and Abderrahmane Kheddar. Homotopy-based controller for human-robot interaction. In 18th IEEE Int. Symposium on Robot and Human Interactive Communication (Ro-Man 2009), Toyama, Japan, 27 september – 2 october 2009. (Cité page 57.)
- [26] Paul Evrard and Abderrahmane Kheddar. Homotopy switching model for dyad haptic interaction in physical collaborative tasks. In *Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Salt Lake City, USA, March 2009. (Cité page 57.)
- [27] Paul Evrard, Olivier Stasse, Nicolas Mansard, Abderrahmane Kheddar, Thomas Schauss, Carolina Weber, Angelika Peer, and Martin Buss. Intercontinental, Multimodal, Wide-area Tele-Cooperation using a Humanoid Robot. In *IEEE/RSJ Int. Conf. on Robots and Intelligent Systems*, 2009. (Cité pages 88, 93 et 102.)
- [28] Roy Featherstone and David Orin. Springer Handbook of Robotics, chapter Dynamics. Springer, 2008. (Cité page 140.)
- [29] Daniela Feth, Raphaela Groten, Angelika Peer, and Martin Buss. Controltheoretic model of haptic human-human interaction in a pursuit tracking task. In 18th IEEE Int. Symposium on Robot and Human Interactive Communication (Ro-Man 2009), 2009. (Cité page 38.)
- [30] Paul M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381–391, 1954. (Cité page 5.)
- [31] T. Flash and N. Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *The Journal of Neuroscience*, 5(7):1688–1703, July 1985. (Cité pages 5, 16, 24 et 122.)
- [32] S. Gentry and E. Feron. Musicality experiments in lead and follow dance. In *IEEE International Conference on Systems, Man, and Cybernetics*, volume 1, pages 984–988, 2004. (Cité page 31.)
- [33] John Glassmire, Marcia O'Malley, William Bluethmann, and Robert Ambrose. Cooperative manipulation between humans and teleoperated agents. *haptics*, 00:114–120, 2004. (Cité page 84.)
- [34] Weston B. Griffin, William R. Provancher, and Mark R. Cutkosky. Feedback Strategies for Telemanipulation with Shared Control of Object Handling Forces. *Presence: Teleoper. Virtual Environ.*, 14(6):720–731, 2005. (Cité page 84.)
- [35] Raphaela Groten, Daniela Feth, Angelika Peer, D. A. Kenny, and Martin Buss. Experimental analysis of dominance in haptic collaboration. In 18th IEEE Int. Symposium on Robot and Human Interactive Communication (Ro-Man 2009), 2009. (Cité pages 22, 37 et 38.)
- [36] Blake Hannaford. A Design Framework for Teleoperators with Kinesthetic Feedback. *IEEE Trans. on Robotics and Automation*, 5:426–434, 1989. (Cité page 129.)

- [37] Blake Hannaford, Laurie Wood, Douglas A. McAffee, and Haya Zak. Performance Evaluation of a Six-Axis Generalized Force-Reflecting Teleoperator. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):620–633, 1991. (Cité page 84.)
- [38] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa. An analytical method for real-time gait planning for humanoid robots. *Internation Journal of Humanoid Robotics*, 3(1):1–19, 2006. (Cité page 105.)
- [39] Yasuhisa Hirata and Kazuhiro Kosuge. Distributed robot helpers handling a single object in cooperation with a human. In *IEEE Int. Conf. on Robotics* and Automation, pages 458–463, 2000. (Cité pages 17, 18 et 19.)
- [40] Yasuhisa Hirata, Youhei Kume, Zhi-Dong Wang, and Kazuhiro Kosuge. Decentralized control of multiple mobile manipulators based on virtual 3D caster motion of handling an object in cooperation with a human. In *IEEE Int. Conf. on Robotics and Automation*, pages 938–943, Taipei, Taiwan, September 2003. (Cité pages 17 et 18.)
- [41] Yasuhisa Hirata, Yosuke Ojima, and Kazuhiro Kosuge. Variable Motion Characteristics Control of an Object by Multiple Passive Mobile Robots in Cooperation with a Human. In *IEEE Int. Conf. on Robotics and Automation*, 2008. (Cité pages 25, 33 et 47.)
- [42] Yasuhisa Hirata, Takeo Takagi, Kazuhiro Kosuge, Hajime Asama, Hayato Kaetsu, and Kuniaki Kawabata. Manipulation of a large object by multiple DR helpers in cooperation with a human. In *IEEE/RSJ Int. Conf. on Robots and Intelligent Systems*, pages 126–131, Maui, Hawaii, October–November 2001. (Cité pages 17, 18 et 19.)
- [43] Sandra Hirche, Manuel Ferre, Jordi Barrio, Claudio Melchiorri, and Martin Buss. *Advances in Telerobotics*, chapter Bilateral Control Architectures for Telerobotics, pages 163–176. Springer, 2007. (Cité page 129.)
- [44] Neville Hogan. Impedance Control: An Approach to Manipulation, Part 1 Theory. *Journal of Dynamic Systems, Measurement, and Control*, 107:1–7, March 1985. (Cité pages 5, 7, 8, 11, 13 et 35.)
- [45] Neville Hogan. Impedance Control: An Approach to Manipulation: Part ii Implementation. Journal of Dynamic Systems, Measurement, and Control, 108:8–16, 1985. (Cité page 13.)
- [46] Neville Hogan. Stable execution of contact tasks using impedance control. In *IEEE Int. Conf. on Robotics and Automation*, 1987. (Cité pages 13, 14, 19, 20 et 104.)
- [47] Neville Hogan. On the Stability of Manipulator Performing Contact Tasks. *IEEE Journal of Robotics and Automation*, 4:667–686, 1988. (Cité pages 13 et 15.)
- [48] R. Ikeura, H. Monden, and H. Inooka. Cooperative motion control of a robot and a human. In 3nd IEEE Int. Symposium on Robot and Human Interactive Communication (Ro-Man 1994), pages 112–117, July 1994. (Cité pages 18 et 22.)

- [49] H. Inoue. Computer Controlled Bilateral Manipulator, 1971. (Cité page 7.)
- [50] H. Inoue. Force Feedback in Precise Assembly Tasks, 1974. (Cité page 7.)
- [51] Tomotaka Itoh, Koji Yudate, Shinsuke Ito, and Takashi Matsui. New Predictive Display Method of Motion and Force Information for Network Teleoperation Without Using Virtual Environment Model. In *IEEE/RSJ Int. Conf.* on Robots and Intelligent Systems, 2003. (Cité page 83.)
- [52] 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. In *IEEE Int. Conf. on Robotics And Automation*, pages 1620–1626, 2003. (Cité pages 90 et 105.)
- [53] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. The 3d linear inverted pendulum mode : A simple modeling of a biped walking pattern generation. In *IEEE/RSJ Int. Conf. on Robots and Intelligent Systems*, pages 239–246, 2001. (Cité page 105.)
- [54] S. Kajita, K. Yokoi, M. Saigo, and K. Tanie. Balancing a humanoid robot using backdrive concerned torque control and direct angular momentum feedback. In *IEEE Int. Conf. on Robotics and Automation*, pages 3376–3382, 2001. (Cité page 116.)
- [55] F. Kanehiro, K. Fujiwara, S. Kajita, K. Yokoi, K. Kaneko, H. Hirukawa, Y. Nakamura, and K. Yamane. Open architecture humanoid robotics platform. In *International Conference on Robotics and Automation*, pages 24– 30, May 2002. (Cité page 114.)
- [56] O. Kanoun, F. Lamiraux, F. Kanehiro, E. Yoshida, J.-P. Laumond, and P.-B. Wieber. Prioritizing linear equality and inequality systems: Application to local motion planning for redundant robots. In *IEEE Int. Conf. on Robotics* and Automation, pages 2939–2944, 2009. (Cité page 116.)
- [57] H. Kazerooni. Robust Compliant Control for Manipulators, Part i: The Fundamental Concepts of Compliant Motion. *IEEE Journal of Robotics* and Automation, RA-2:83–92, 1986. (Cité page 12.)
- [58] H. Kazerooni. Human/robot interaction via the transfer of power and information signals; part 1: Dynamics and control analysis. In *IEEE Int. Conf.* on Robotics and Automation, Scottsdale, Arizona, May 1989. (Cité page 7.)
- [59] François Keith, Paul Evrard, Jean-Rémy Chardonnet, Sylvain Miossec, and Abderrahmane Kheddar. Haptic interaction with virtual avatars. In Manuel Ferre, editor, *EuroHaptics*, volume 5024 of *Lecture Notes in Computer Science*, pages 630–639. Springer, 2008. (Cité pages 139 et 147.)
- [60] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal of Robotics* and Automation, 3(1):43–53, 1987. (Cité pages 34 et 89.)
- [61] Abderrahmane Kheddar, Ee-Sian Neo, Riichiro Tadakuma, and Kazuhito Yokoi. Advances in Telerobotics, chapter Enhanced Teleoperation Through Virtual Reality Techniques, pages 139–159. Springer, 2007. (Cité page 83.)

- [62] Jonghyun Kim, Pyung Hun Chang, and Hyung-Soon Park. Transparent Teleoperation Using Two-channel Control Architectures. In *IEEE/RSJ Int. Conf. on Robots and Intelligent Systems*, 2005. (Cité pages 83 et 130.)
- [63] K.Kaneko, F.Kanehiro, S.Kajita, H.Hirukawa, T.Kawasaki, M.Hirata, K.Akachi, and T.Isozumi. Humanoid robot hrp-2. In *IEEE/RSJ Int. Conf.* on Robots and Intelligent Systems, volume 2, pages 1083–1090, 2004. (Cité pages 84 et 102.)
- [64] K. Kosuge, H. Yoshida, and T. Fukuda. Dynamic control for robot-human collaboration. In 2nd IEEE Int. Symposium on Robot and Human Interactive Communication (Ro-Man 1993), pages 389–401, Tokyo, Japan, 1993. (Cité pages 5 et 17.)
- [65] Kazuhiro Kosuge, Tomohiro Hayashi, Yasuhisa Hirata, and Ryosuke Tobiyama. Dance partner robot –Ms DanceR–. In *IEEE/RSJ Int. Conf. on Robots and Intelligent Systems*, pages 3459–3464, Las Vegas, Nevada, October 2003. (Cité page 18.)
- [66] Kazuhiro Kosuge, Manabu Sato, and Norihide Kazamura. Mobile robot helper. In *IEEE Int. Conf. on Robotics and Automation*, pages 583–588, San Francisco, CA, USA, April 2000. (Cité page 106.)
- [67] Dale A. Lawrence. Impedance Control Stability Properties in Common Implementations. In *IEEE Int. Conf. on Robotics and Automation*, 1988. (Cité pages 14 et 15.)
- [68] F.B. Llewellyn. Some fundamental properties of transmission systems. Proceedings of IRE, pages 271–283, 1952. (Cité page 132.)
- [69] Joris De Schutter Luigi Villani. Springer Handbook of Robotics, chapter Force Control, pages 161–185. Springer, 2008. (Cité page 8.)
- [70] Yusuke Maeda, Takayuki Hara, and Tamio Arai. Human-robot cooperative manipulation with motion estimation. In *IEEE/RSJ Int. Conf. on Robots and Intelligent Systems*, pages 2240–2245, Maui, Hawaii, October–November 2001. (Cité pages 5, 17, 18, 20, 23, 24, 26, 35 et 122.)
- [71] N. Mansard and F. Chaumette. Task Sequencing for Sensor-Based Control. *IEEE Trans. on Robotics*, 23(1):60–72, February 2007. (Cité pages 89 et 103.)
- [72] Nicolas Mansard, Oussama Khatib, and Abderrahmane Kheddar. A unified approach to integrate unilateral constraints in the stack of tasks. *IEEE Trans.* on Robotics, 25:670–685, 2009. (Cité page 115.)
- [73] Nicolas Mansard, Olivier Stasse, Paul Evrard, and Abderrahmane Kheddar. A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: the stack of task. In *IEEE Int. Conf. on Advanced Robotics*, Munich, Germany, June 22-26 2009. (Cité pages 111 et 112.)
- [74] Matthew T. Mason. Compliance and force control for computer controlled manipulators. Master's thesis, Massachusetts Institute of Technology, 1979. (Cité page 7.)

- [75] Sylvain Miossec and Abderrahmane Kheddar. Human hand motion in cooperative tasks: moving object case study. In *IEEE Int. Conf. on Robotics and Biomimetics*, Bangkok, Thailand, December 14-17 2008. (Cité pages 5, 6 et 44.)
- [76] M. Morisawa, K. Harada, S. Kajita, S. Nakaoka, K. Fujiwara, F. Kanehiro, K. Kaneko, and H. Hirukawa. Experimentation of humanoid walking allowing immediate modification of foot place based on analytical solution. In *IEEE Int. Conf. on Robotics and Automation*, pages 3989–3994, 2007. (Cité pages 105, 106 et 121.)
- [77] Ee-Sian Neo, Kazuhiko Yokoi, Shuuji Kajita, and Kazuo Tanie. Whole-Body Motion Generation Integrating Operator's Intention and Robot's Autonomy in Controlling Humanoid Robots. *IEEE Trans. on Robotics*, 23:763–775, 2007. (Cité page 84.)
- [78] H. Niemann, J. Stoustrup, and R. B. Abrahamsen. Switching between multivariable controllers. *Optimal control applications and methods*, 25:51–66, 2004. (Cité page 36.)
- [79] Günter Niemeyer and Jean-Jacques E. Slotine. Stable Adaptive Teleoperation. *IEEE Journal of Oceanic Engineering*, 16:152–162, 1991. (Cité pages 61 et 83.)
- [80] K. Nishiwaki and S. Kagami. Online walking control system for humanoids with short cycle pattern generation. *International Journal of Robotics Research*, 28(6):729–742, 2009. (Cité pages 102, 105 et 116.)
- [81] J. Park and O. Khatib. Contact Consistent Control Framework for Humanoid Robots. In *IEEE Int. Conf. on Robotics and Automation*, 2006. (Cité page 89.)
- [82] R. Paul and B. Shimano. Compliance and control. In Proceedings of the 1976 Joint Automatic Control Conference, 1976. (Cité page 7.)
- [83] A. Peer and M. Buss. A New Admittance-Type Haptic Interface for Bimanual Manipulations. *IEEE/ASME Trans. on Mechatronics*, 13:416–428, 2008. (Cité pages 91 et 93.)
- [84] Angelika Peer, Sandra Hirche, Carolina Weber, Inga Krause, Martin Buss, Sylvain Miossec, Paul Evrard, Olivier Stasse, Ee-Sian Neo, Abderrahmane Kheddar, and Kazuhito Yokoi. Intercontinental multimodal tele-cooperation using a humanoid robot. In *IEEE/RSJ Int. Conf. on Robots and Intelligent Systems*, pages 405–411, Nice, France, 2008. (Cité pages 88, 102, 129, 132 et 133.)
- [85] Nicolas Perrin, Olivier Stasse, Florent Lamiraux, Paul Evrard, and Abderrahmane Kheddar. On the problem of online footsteps correction for humanoid robots. In *Proceedings of the RSJ conference*, 2009. (Cité page 110.)
- [86] Md.Mozasser Rahman, Ryojun Ikeura, and Kazuki Mizutani. Investigation of the impedance characteristic of human arm for development of robots to cooperate with humans. *JSME Int. Journal Series C, Mechanical Systems, Machine Elements and Manufacturing*, 45(2):510–518, 2002. (Cité pages 5, 17, 18, 22, 36, 37 et 59.)

- [87] M.M. Rahman, R. Ikeura, and K. Mizutani. Control characteristics of two humans in cooperative task. In *IEEE International Conference on Systems*, *Man, and Cybernetics*, volume 2, pages 1301–1306, 2000. (Cité page 22.)
- [88] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 103:126– 133, June 1981. (Cité page 8.)
- [89] Kyle Reed, James Patton, and Michael Peshkin. Replicating human-human physical interaction. In *IEEE Int. Conf. on Robotics and Automation*, 2007. (Cité page 42.)
- [90] Kyle B. Reed. Understanding the Haptic Interactions of Working Together. PhD thesis, Northwestern University, June 2007. (Cité pages 5, 6, 22 et 38.)
- [91] Kyle B. Reed and Michael A. Peshkin. Physical collaboration of humanhuman and human-robot teams. *IEEE Transactions on Haptics*, 1(2):108– 120, August-December 2008. (Cité pages 44, 47, 49 et 51.)
- [92] Kylle Reed, Michael Peshkin, J. Edward Colgate, and James Patton. Initial studies in human-robot-human interaction: Fitts' law for two people. In *IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 2333–2338, 2004. (Cité page 6.)
- [93] A. Remazeilles, N. Mansard, and F. Chaumette. Qualitative visual servoing: application to the visibility constraint. In *IEEE/RSJ Int. Conf. on Robots* and Intelligent Systems, pages 4297–4303, Beijing, China, October 2006. (Cité page 45.)
- [94] J. Kenneth Salisbury. Active stiffness control of a manipulator in cartesian coordinates. In *IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, 1980. (Cité page 7.)
- [95] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control: the Task Function Approach*. Clarendon Press, Oxford, United Kingdom, 1991. (Cité page 88.)
- [96] Gilbert Saporta. Probabilités, Analyse des données et Statistiques. Editions Technip, 1990. (Cité page 68.)
- [97] Oliver C. Schrempf, Uwe D. Hanebeck, Andreas J. Schmid, and Heinz Wörn. A Novel Approach To Proactive Human-Robot Cooperation. In 14th IEEE Int. Symposium on Robot and Human Interactive Communication (Ro-Man 2005), 2005. (Cité page 25.)
- [98] T.B. Sheridan and Staff. Mit Research in Telerobotics. In *Proc. of the Workshop on Space Telerobotics*, 1987. (Cité page 129.)
- [99] B. Siciliano and J-J. Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. In *IEEE Int. Conf. on Advanced Robotics*, pages 1211–1216, Pisa, Italy, 1991. (Cité pages 88 et 89.)
- [100] D. Silver. The Little Robot System, 1973. (Cité page 7.)

- [101] O. Stasse, F. Lamiraux, R. Ruland, A. Kheddar, K. Yokoi, and W. Prinz. Integration of humanoid robots in collaborative working environment: A case study on motion generation. *Intelligent Service Robotics*, page accepted, 2009. (Cité pages 102 et 119.)
- [102] Olivier Stasse, Adrien Escande, Nicolas Mansard, Sylvain Miossec, Paul Evrard, and Abderrahmane Kheddar. Real-time (self)-collision avoidance task on a HRP-2 humanoid robot. In *IEEE Int. Conf. on Robotics and Automation*, pages 3200–3205, Pasadena, California, May 2008. (Cité pages 34 et 115.)
- [103] Olivier Stasse, Paul Evrard, Nicolas Perrin, Nicolas Mansard, and Abderrahmane Kheddar. Fast foot prints replanning and generation during walking in physical human-humanoid interaction. In *IEEE-RAS International Conference on Humanoid Robots*, 2009. (Cité page 121.)
- [104] Olivier Stasse, Bjorn Verrelst, Pierre-Brice Wieber, Bram Vanderborghtand, Paul Evrard, Abderrahmane Kheddar, and Kazuhito Yokoi. Modular architecture for humanoid walking pattern prototyping and experiments. Advanced Robotics, Special Issue on Middleware for Robotics –Software and Hardware Module in Robotics System, 22(6):589–611, 2008. (Cité pages 90 et 112.)
- [105] M. Stilman, K. Nishiwaki, and S. Kagami. Learning object models for whole body manipulation. In *IEEE-RAS International Conference on Humanoid Robots*, pages 820–826, 2007. (Cité pages 102 et 105.)
- [106] Susumu Tachi, Kiyoshi Komiyama, Kazuya Sawada, Takashi Nishiyama, Toshiyuki Itoko, Masami Kobayashi, and Kozo Inoue. Development of telexistence cockpit for humanoid robot control. In *Proceedings of the 32nd ISR (International Symposium on Robotics)*, 2001. (Cité page 83.)
- [107] Takahiro Takeda, Kazuhiro Kosuge, and Yasuhisa Hirata. HMM-based dance step estimation for dance partner robot –MS DanceR–. In *IEEE/RSJ Int. Conf. on Robots and Intelligent Systems*, pages 1602–1607, 2005. (Cité page 19.)
- [108] Tomohito Takubo, Kazutoshi Nishii, Kenji Inoue, Yasushi Mae, and Tatsuo Arai. Marionette System for Operating and Displaying Robot Whole-Body Motion -Development of Similar Humanoid-Type Device-. In *IEEE/RSJ Int. Conf. on Robots and Intelligent Systems*, 2004. (Cité page 83.)
- [109] Y. Tsumaki and M. Uchiyama. Predictive Display of Virtual Beam for Space Teleoperation. In *IEEE/RSJ Int. Conf. on Robots and Intelligent Systems*, 1996. (Cité page 83.)
- [110] Toru Tsumugiwa, Ryuichi Yokogawa, and Kei Hara. Variable impedance control with regard to working process for man-machine cooperation-work system. In *IEEE/RSJ Int. Conf. on Robots and Intelligent Systems*, pages 1564–1569, Maui, Hawaii, October–November 2001. (Cité pages 5, 17, 18 et 22.)
- [111] Toru Tsumugiwa, Ryuichi Yokogawa, and Kazunobu Yoshida. Stability analysis for impedance control of robot for human-robot cooperative task

system. In *IEEE/RSJ Int. Conf. on Robots and Intelligent Systems*, pages 3883–3888, Sendai, Japan, September–October 2004. (Cité page 19.)

- [112] Marc Ueberle and Martin Buss. Control of Kinesthetic Haptic Interfaces. In IEEE/RSJ Int. Conf. on Robots and Intelligent Systems, 2004, Workshop on Touch and Haptics, 2004. (Cité page 12.)
- [113] U. Unterhinninghofen, T. Schauss, and M. Buss. Control of a mobile haptic interface. In *IEEE Int. Conf. on Robotics and Automation*, volume pp. 2085-2090, 2008. (Cité pages 84, 88, 91 et 93.)
- [114] M. Vukobratovic and B. Borovac. Zero moment point-thirty-five years of its life. *International Journal of Humanoid Robotics*, 1(1):157–173, 2004. (Cité page 3.)
- [115] Zheng Wang, Angelika Peer, and Martin Buss. An hmm approach to realistic haptic human-robot interaction. In *Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 374–379, 2009. (Cité page 37.)
- [116] Zhi-Dong Wang, E. Nakano, and T. Matsukawa. A new approach to multiple robots' behavior design for cooperative object manipulation. *Distributed Autonomous Robotic Systems*, 2:350–361, March 1996. (Cité page 37.)
- [117] D. E. Whitney. Force feedback control of manipulator fine motions. In Proceedings of the 1976 Joint Automatic Control Conference, 1976. (Cité page 7.)
- [118] S. Yigit, C. Burghart, and H. Wörn. Cooperative carrying using pump-like constraints. In *IEEE/RSJ Int. Conf. on Robots and Intelligent Systems*, pages 3877–3882, Sendai, Japan, September–October 2004. (Cité page 18.)
- [119] Kazuhito Yokoi, Katsumi Nakashima, Masami Kobayashi, Humisato Mihune, Hitoshi Hasunuma, Yoshitaka Yanagihara, Takao Ueno, Takuya Gokyuu, and Ken Endou. A Tele-operated Humanoid Operator. *The International Journal of Robotics Research*, 25:593–602, 2006. (Cité page 84.)
- [120] Kazuhiko Yokoyama, Hiroyuki Handa, Takakatsu Isozumi, Yutaro Fukase, Kenji Kaneko, Fumio Kanehiro, Yoshihiro Kawai, Fumiaki Tomita, and Hirohisa Hirukawa. Cooperative Works by a Human and a Humanoid Robot. In *IEEE Int. Conf. on Robotics and Automation*, 2003. (Cité pages 18 et 106.)