



# Manipulation collaborative distante à retour d'effort : application au prototypage virtuel

Mathieu Gautier

► **To cite this version:**

Mathieu Gautier. Manipulation collaborative distante à retour d'effort : application au prototypage virtuel. Automatique / Robotique. Université Pierre et Marie Curie - Paris VI, 2009. Français. <NNT : 2009PA066643>. <tel-00813368>

**HAL Id: tel-00813368**

**<https://tel.archives-ouvertes.fr/tel-00813368>**

Submitted on 15 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.



**THESE DE DOCTORAT DE  
L'UNIVERSITE PIERRE ET MARIE CURIE**

spécialité

**Robotique**

Présentée par

**Mathieu Gautier**

Pour obtenir le grade de

**DOCTEUR de L'UNIVERSITÉ PIERRE ET MARIE CURIE**

Sujet de la thèse :

**Manipulation collaborative distante à retour d'effort :  
application au prototypage virtuel**

Soutenue le

devant le jury composé de :

M.	Bruno ARNALDI	Professeur à l'INSA de Rennes	Rapporteur
M.	Laurent GRISONI	Professeur à l'université de Lille 1	Rapporteur
Mme	Sabine COQUILLART	Directrice de recherche à l'INRIA	Examineur
M.	Vincent HAYWARD	Professeur à l'UPMC	Examineur
M.	Michel DE MATHELIN	Professeur à l'université de Strasbourg	Invité
M.	Claude ANDRIOT	Expert sénior au CEA-LIST	Encadrant
M.	Philippe BIDAUD	Professeur à l'UPMC	Directeur de thèse



## Résumé

LES processus de conception industriels font de plus en plus appel à l'utilisation de maquettes numériques. En effet, le recours aux environnements virtuels pour les phases de prototypage permet de s'abstraire de modèles réels. La manipulation d'objets complexes requiert alors l'utilisation d'interfaces à retour d'effort afin de rendre l'interaction plus intuitive. D'autre part, ces processus de conception et de développement s'inscrivent de plus en plus dans une démarche de travail collaboratif entre plusieurs sites distants.

Dans ce contexte, la distribution de la plateforme de simulation numérique avec retour d'effort entre plusieurs sites permettrait d'étendre l'utilisation de ces outils de simulations à un cadre collaboratif.

Pour cela nous proposons d'étudier une architecture distribuée en vue de manipuler simultanément des objets dans un environnement virtuel.

Afin de garantir la stabilité du système, nous détaillons ensuite deux méthodes dérivées du domaine de la téléopération. La première repose sur les couplages virtuels proportionnels dérivés dans le cas de retards faibles. La seconde est basée sur les variables d'onde quelque soit le retard.

Nous proposons, enfin, d'enrichir le retour d'effort en ajoutant de l'haptique événementielle. Cette méthode consiste à superposer une information haute fréquence au retour d'effort. Nous présentons une évaluation de cette démarche dans un environnement représentatif de la plateforme distribuée.

Ces différents éléments sont intégrés au sein d'une plateforme collaborative.

**Mots clés :** Réalité virtuel, Retour haptique, Environnements virtuels collaboratifs, Variables d'onde, Haptique événementielle

Cette thèse a été réalisée au sein du Laboratoire de Simulation Interactive (LSI) dépendant du Commissariat à l'Énergie Atomique - Laboratoire d'Intégration des Systèmes et des Technologies (CEA-LIST) :

CEA LIST, Laboratoire de simulation interactive  
18 route du Panorama  
BP6 FONTENAY AUX ROSES  
F-92265 France



# Haptic feedback dedicated to virtual collaborative prototyping environment

## Abstract

NOWADAYS, industrial applications are increasingly relying on digital mock-ups. Indeed, the use of virtual environments for prototyping tasks replaces the need of real mock-ups. The manipulation of virtual objects requires the use of force feedback interfaces to make the interaction more intuitive. At the same time, the industrial design processes depends more and more on collaborative works between remote locations.

In this context, the use of a distributed collaborative environment with force feedback between several locations would increase the use of such digital tools in a collaborative framework.

To do so, we propose a platform based on a peer to peer architecture to manipulate objects simultaneously in a virtual environment. Especially, we present the integration of a 6 degrees of freedom haptic feedback taking into account the delays introduced by the distance between remote locations.

To ensure system stability, we develop two methods inspired by the teleoperation field. The first one is based on proportional derivative controllers in the case of small delays. The second one is based on the wave variables transform. This last method is stable regardless of the delay.

Finally, we suggest to use event-based haptic to increase the user experience in a collaborative environment. This method consists in superimposing high frequency transient forces to the traditional haptic feedback. We present an evaluation of this approach.

**Keyword** : Virtual reality, Haptic feedback, Virtual collaborative environment, Wave variables, Event-based haptic



# Remerciements

I

Lieu, 18 septembre 2009.





# Table des matières

<b>Tables des matières</b>	<b>i</b>
<b>Liste des figures</b>	<b>v</b>
<b>Liste des tables</b>	<b>ix</b>
<b>Introduction</b>	<b>1</b>
<b>1 État de l'art</b>	<b>5</b>
1.1 Simulation physique interactive d'univers virtuel . . . . .	6
1.1.1 Algorithmes de détection de collision pour les objets rigides . . . . .	6
1.1.1.1 Algorithme discret . . . . .	8
1.1.1.2 Détection de collision continue . . . . .	9
1.1.2 Simulation physique . . . . .	10
1.1.2.1 Équations d'évolution d'objets rigides . . . . .	10
1.1.2.2 Modèle de contact pour des objets rigides parfaits . . . . .	11
1.1.2.3 Résolution du contact . . . . .	12
1.1.2.4 Schéma d'intégration . . . . .	13
1.1.3 Moteur physique . . . . .	14
1.2 Retour haptique . . . . .	15
1.2.1 Interfaces à retour d'effort . . . . .	15
1.2.1.1 Retour haptique . . . . .	15
1.2.1.2 Système à retour d'effort . . . . .	16
1.2.2 Stabilité du retour d'effort . . . . .	17
1.2.2.1 Théorie de la passivité . . . . .	17
1.2.2.2 Couplage virtuel . . . . .	22
1.2.2.3 Rendu haptique événementiel . . . . .	25
1.3 Interactions haptiques distantes . . . . .	26
1.3.1 Couplage virtuel . . . . .	27
1.3.2 Observateur et contrôleur de passivité . . . . .	28
1.3.3 Variables d'onde . . . . .	29
1.3.3.1 Influence de l'impédance . . . . .	32
1.3.3.2 Suivi en position . . . . .	33
1.3.3.3 Retard variable . . . . .	34
1.4 Collaboration et coopération distantes . . . . .	36
1.4.1 Architecture réseau des plateformes collaboratives . . . . .	36
1.4.1.1 Architecture client-serveur . . . . .	37
1.4.1.2 Architecture pair-à-pair . . . . .	38
1.4.1.3 Architectures hybrides . . . . .	42
1.4.2 Retour haptique sur une tâche collaborative . . . . .	42

1.4.3	Influence du retard sur la coopération . . . . .	43
	Conclusion . . . . .	45
<b>2</b>	<b>Plateforme de collaboration</b>	<b>49</b>
2.1	Simulation physique à 6 degrés de liberté . . . . .	50
2.1.1	Formulation . . . . .	50
2.1.2	Formulation quasi-statique . . . . .	50
2.2	Architecture réseau pair-à-pair . . . . .	52
2.2.1	Motivations . . . . .	52
2.2.2	Conception . . . . .	53
2.3	Architecture logicielle proposée . . . . .	54
2.3.1	Architecture des nœuds . . . . .	54
2.3.2	Les aides visuelles . . . . .	56
2.4	Divergence et cohérence . . . . .	56
2.4.1	Point de préhension et point de couplage . . . . .	57
2.4.1.1	Cas des translations et définition d'une mesure de la divergence . . . . .	57
2.4.1.2	Cas des rotations . . . . .	60
2.4.2	Point de couplage et manipulation multi-utilisateurs . . . . .	61
2.4.3	Gestion de la cohérence . . . . .	62
	Conclusion . . . . .	64
<b>3</b>	<b>Asservissement par couplage virtuel proportionnel dérivé</b>	<b>67</b>
3.1	Couplage Proportionnel-Dérivé et retour haptique distant . . . . .	68
3.1.1	Motivation . . . . .	68
3.1.2	Couplage en 6 degré de liberté . . . . .	69
3.2	Analyse en 1 degré de liberté . . . . .	70
3.2.1	Les différents cas de manipulation . . . . .	70
3.2.2	Formulation discrète . . . . .	73
3.2.3	Matrices d'immittance des différents cas de manipulations . . . . .	73
3.2.3.1	Manipulation d'un objet par deux personnes . . . . .	74
3.2.3.2	Manipulation d'un objet libre et manipulation avec contact distant . . . . .	75
3.2.3.3	Manipulation d'un objet avec contact local . . . . .	76
3.2.4	Étude de stabilité . . . . .	76
3.2.4.1	Influence du retard . . . . .	77
3.2.5	Efforts de traînée . . . . .	78
3.3	Compensation des efforts de traînée . . . . .	81
3.3.1	Boucle de compensation . . . . .	81
3.3.2	Effet de la boucle de compensation . . . . .	83
	Conclusion . . . . .	85
<b>4</b>	<b>Asservissement par variables d'onde</b>	<b>87</b>
4.1	Couplage par variables d'onde articulaire . . . . .	87
4.1.1	Motivation . . . . .	87
4.1.2	Formulation articulaire . . . . .	89
4.1.3	Contrôle de la dérive . . . . .	90
4.1.4	Filtrage des réflexions d'onde . . . . .	92
4.2	Expérimentation sur la plateforme collaborative . . . . .	93
4.2.1	Simulation . . . . .	93
4.2.2	Plateforme collaborative . . . . .	93

4.2.3	Influence du retard . . . . .	95
	Conclusion . . . . .	100
<b>5</b>	<b>Retour haptique événementiel</b>	<b>101</b>
5.1	Motivation . . . . .	101
5.2	Calcul du rendu haptique événementiel . . . . .	103
5.2.1	Surévaluation de l'impact . . . . .	103
5.3	Expérimentation . . . . .	105
5.3.1	Objectif . . . . .	105
5.3.2	Sujets . . . . .	106
5.3.3	Dispositif expérimental . . . . .	106
5.3.4	Procédure . . . . .	107
5.4	Résultats . . . . .	109
5.4.1	Résultat des efforts . . . . .	109
5.4.2	Résultat pour les erreurs de manipulation . . . . .	111
5.4.3	Résultats pour le temps de réalisation . . . . .	112
5.5	Discussion . . . . .	112
5.5.1	Effets du retour haptique événementiel . . . . .	113
	Conclusion . . . . .	114
	<b>Conclusion et perspectives</b>	<b>117</b>
<b>A</b>	<b>Annexe</b>	<b>121</b>
A.1	Notations . . . . .	121
A.1.1	Notation . . . . .	121
A.1.2	Cinématique des objets rigides . . . . .	121
A.2	Expression des matrices d'immittances pour le couplage proportionnel dérivé . . . . .	123
A.2.1	Expression sans la boucle de compensation . . . . .	123
A.2.1.1	Cas mono-utilisateur . . . . .	123
A.2.2	Ajout d'une boucle de compensation . . . . .	124
A.2.2.1	Cas de manipulation collaborative . . . . .	124
A.2.2.2	Cas mono-utilisateur . . . . .	125
	<b>Publications</b>	<b>127</b>
	<b>Bibliographie</b>	<b>129</b>



# Table des figures

1	Exemple de tâche de prototypage virtuel avec retour d'effort . . . . .	1
1.1	Exemple d'application permettant une interaction collaborative distante . . . . .	5
1.2	Algorithme VPS . . . . .	8
1.3	Algorithme LMD . . . . .	9
1.4	Exemples d'interface haptique . . . . .	17
1.5	Théorème de la passivité . . . . .	18
1.6	Architecture d'un environnement avec rendu haptique en admittance . . . . .	19
1.7	Architecture du rendu direct . . . . .	19
1.8	Système à un port avec contrôleur de passivité en parallèle . . . . .	21
1.9	Schéma bloc de rendu haptique avec un couplage virtuel en impédance. . . . .	22
1.10	Schéma bloc de rendu haptique avec un couplage virtuel. . . . .	23
1.11	Effet passe-muraille avec un rendu direct. Lorsque l'objet à atteint la position de droite, l'utilisateur à l'impression d'avoir traversé l'objet gris. . . . .	24
1.12	Schéma bloc d'asservissement en téléopération . . . . .	26
1.13	Représentation mécanique d'un asservissement position-position pour un cas de téléopération . . . . .	27
1.14	Extension de schéma OP/CP au cas d'un système à deux ports . . . . .	28
1.15	Quadripôle . . . . .	29
1.16	Transformation en variable d'ondes . . . . .	30
1.17	Transformation en variable d'onde avec filtre à horizon fini (issu de [TN05] . . . . .	33
1.18	Variable d'onde déformée . . . . .	35
1.19	Architecture client-serveur . . . . .	37
1.20	Architecture clients-serveur avec retour d'effort . . . . .	38
1.21	Architecture pair-à-pair . . . . .	39
1.22	Cas d'incohérence entre deux nœuds. . . . .	39
1.23	Architecture pair-à-pair avec retour d'effort . . . . .	39
1.24	Organisation des moteurs physiques suivant un anneau . . . . .	41
1.25	Décorateurs révélant le retard . . . . .	45
2.1	Simulation physique : cas d'une collaboration . . . . .	51
2.2	Cas d'une légère incohérence . . . . .	54
2.3	Architecture de la plateforme collaborative . . . . .	54
2.4	Transformation en architecture hybride . . . . .	55
2.5	Architecture logicielle . . . . .	55
2.6	Architecture logicielle : cas à deux nœuds . . . . .	56
2.7	aide visuelle : fantôme . . . . .	56
2.8	Exemple d'incohérence . . . . .	57
2.9	[Position des points de couplage pour une force . . . . .	58
2.10	Mesure de la divergence . . . . .	59

2.11	Position des points de couplage pour une rotation . . . . .	60
2.12	Exemple de deux couplages en $A, A'$ et $B, B'$ . . . . .	61
2.13	Exemple de comanipulation d'un objet . . . . .	62
2.14	Exemple de trajectoire de deux instances d'un objet répliqué . . . . .	63
2.15	Exemple de discrétisation de la trajectoire et de détection d'incohérence . . . . .	64
2.16	Implémentation de la détection d'incohérence . . . . .	64
3.1	Asservissements proportionnel-dérivés . . . . .	68
3.2	Manipulation d'un objet entre deux nœuds . . . . .	69
3.3	Manipulation à deux personnes d'un objet répliqué . . . . .	70
3.4	Cas de manipulation . . . . .	72
3.5	Schéma bloc de la manipulation par deux personnes d'un objet répliqué . . . . .	75
3.6	Manipulation par un seul manipulateur . . . . .	75
3.7	Cas d'un contact local . . . . .	76
3.8	Domaine de stabilité pour un retard minimal pour $k_c = k_h$ et $b_c = b_h$ . . . . .	77
3.9	Domaine de stabilité en fonction de retards croissants . . . . .	79
3.10	Efforts de traînée lors d'un mouvement libre . . . . .	80
3.11	Trajectoire de test, position de l'objet . . . . .	80
3.12	Schéma bloc d'une manipulation libre, sans l'interface haptique . . . . .	80
3.13	Schéma bloc avec compensation . . . . .	81
3.14	Effet de la compensation sur les marges de stabilité . . . . .	82
3.15	Effet de la compensation sur un mouvement libre . . . . .	82
3.16	Effet d'une compensation de 0.75 sur les translations . . . . .	83
3.17	Effet d'une compensation de 0.25 sur les rotations . . . . .	83
3.18	Effet de la compensation lors d'un contact distant . . . . .	84
3.19	Effet de la compensation lors du contact . . . . .	84
4.1	Divergence induit par la transformation en variables d'onde . . . . .	88
4.2	Exemple d'un asservissement articulaire pour deux robots redondants identiques . . . . .	88
4.3	Plateforme collaborative à 6 DDLS utilisant les variables d'onde . . . . .	89
4.4	Variables d'onde dans l'espace articulaire . . . . .	91
4.5	Variables d'onde avec suivi en position. . . . .	92
4.6	Effets du filtrage des variables d'onde . . . . .	94
4.7	Deux utilisateurs manipulent chacun une pince à souder . . . . .	95
4.8	Suivi en position selon l'axe $x$ pour le cas des pinces à souder. . . . .	95
4.9	Couples sur chacune des interfaces haptiques selon l'axe le plus actif . . . . .	96
4.10	Un utilisateur maintient son bouton en position pendant que le second essaie de le pousser . . . . .	96
4.11	Divergence et effort pour un retard de 100ms . . . . .	97
4.12	Divergence et effort pour un retard de 200ms . . . . .	97
4.13	Divergence et effort pour un retard de 400ms . . . . .	98
4.14	Divergence et effort pour un retard de 600ms . . . . .	98
4.15	Divergence et effort pour un retard de 50ms avec variables d'onde . . . . .	99
4.16	Divergence et effort pour un retard de 50ms avec un couplage ressort amortisseur et la compensation activée . . . . .	99
5.1	Surévaluation de l'impact . . . . .	104
5.2	Effets des méthodes de réalité virtuel et de téléopération . . . . .	105
5.3	Dispositif technique . . . . .	106
5.4	Labyrinthe utilisé pour l'évaluation . . . . .	107

---

5.5	Le parcours effectivement vu par le sujet . . . . .	108
5.6	Moyennes et boîte à moustaches pour les efforts maximum le long de l'axe $x$ . .	109
5.7	Moyennes et boîtes à moustache pour les efforts maximum le long de l'axe $y$ . .	110
5.8	Moyennes et boîtes à moustache pour le temps de réalisation . . . . .	110
5.9	Différences significatives pour le temps, les efforts et les erreurs . . . . .	110
5.10	Effet du retour haptique événementiel . . . . .	114
A.1	Schéma bloc détaillé . . . . .	124
A.2	Schéma bloc détaillé du cas collaboratif avec boucle de compensation (Figure 3.13)	124
A.3	Schéma bloc détaillé du cas collaboratif avec boucle de compensation (Figure 3.13)	125





# Liste des tableaux

3.1	Ordre de grandeur de retard en le CEA Fontenay aux Roses et des universités. . .	78
5.1	Effets induit par les méthodes couramment rencontrée dans les environnements virtuels . . . . .	102
5.2	ANOVA à trois facteurs des efforts . . . . .	109
5.3	Moyennes ( $m$ ) et écarts type ( $\sigma$ ) du nombre d'erreurs . . . . .	111
5.4	ANOVA à trois facteurs sur le temps de réalisation . . . . .	112



# Introduction

ENCORE peu répandues il y a quelques années, les techniques de la réalité virtuelle trouvent de nos jours des applications de plus en plus utilisées dans une très grande variété de domaines tels que le jeu vidéo, le domaine médical, la formation technique, l'architecture, le prototypage virtuel. Dans ce dernier cas, l'utilisation de maquettes numériques dans un projet industriel permet de réduire le temps alloué aux différentes étapes de ce projet (design, conception, prototypage, formation). Notamment, dans le cas du prototypage virtuel, qui sera plus particulièrement abordé dans le cadre de cette thèse, les dessins issus de la conception par ordinateur peuvent être très rapidement convertis en maquette numérique afin de valider les modalités de montage des différents éléments.



**Figure 1** – Exemple de tâche de prototypage virtuel avec retour d'effort

L'ajout de modalités (auditives et/ou haptiques<sup>1</sup>) lors de l'interaction avec ces environnements virtuels permet d'améliorer l'immersion dans ces univers en rendant les interactions plus intuitives. Dans le cas de la manipulation d'objets virtuels, le retour d'effort (ou retour haptique) est un moyen efficace de fournir les informations de contact entre les objets d'un environnement virtuel. En contraignant le mouvement des utilisateurs, le retour d'effort permet de fournir une meilleure compréhension des interactions entre objets virtuels.

Plus récent que la réalité virtuelle, les environnements virtuels avec retour d'effort commencent à être de plus en plus utilisés dans différents domaines, comme la formation au geste technique ou le prototypage virtuel. Néanmoins, que ce soit pour la formation ou pour la conception, il est courant que plusieurs intervenants puissent interagir dans un même environnement virtuel. La réalité virtuelle permet de s'affranchir de ces limites géographiques et donc autorise plusieurs intervenants à collaborer depuis des sites distants. Des plateformes grand public, tel *Second Life*<sup>2</sup>, permettent en effet à de nombreux utilisateurs répartis sur l'ensemble du globe d'interagir au sein d'un même environnement virtuel. Ce type d'environnement virtuel a été utilisé à plus petite échelle, en termes de participants, pour des tâches de formation entre un professeur et des apprentis ne se situant pas en un même lieu. En parallèle de ces évolutions, les besoins en termes de collaboration à distance dans le milieu industriel sont croissants. De plus en plus d'équipes de recherche et développement situées

---

1. haptique : relatif au sens du toucher

2. <http://www.secondlife.com>

dans différents pays sont amenées à collaborer. L'utilisation d'environnements virtuels dans de tels contextes est évidemment un gain de temps, financier et environnemental appréciable.

Cependant, les environnements virtuels reliant différents sites géographiques sont confrontés au problème du temps de propagation de l'information. Un tel retard peut être compris entre quelques millisecondes, si les sites sont très proches, jusqu'à 300 ms pour des sites très éloignés. Ce retard peut également être variable et donc être momentanément plus important. Le retard entre l'émission d'une information en un lieu et sa réception à endroit différent pose de multiples problèmes. Notamment, un retard de quelques millisecondes dans une boucle de retour d'effort peut la rendre instable. De plus, un retard d'information de plusieurs dizaines de millisecondes entre une action et la réaction associée peut rompre la perception de la causalité entre deux actions. Enfin, le retard a une influence différente sur le comportement d'un univers virtuel en fonction de l'architecture utilisée pour simuler cet univers. Certains choix d'architecture peuvent mener à l'existence de plusieurs simulations physiques qui doivent interagir entre elles. Ces simulations physiques doivent alors être couplées afin que les actions d'un utilisateur sur un objet virtuel soit perçues et ressenties par les autres participants quelque soit le lieu où ils se trouvent.

La présence et les effets de ce retard mène à considérer les trois problèmes suivant :

- Quelle architecture matérielle informatique et logicielle utiliser pour héberger les applications destinées au prototypage virtuel ?
- Comment garantir la stabilité du retour d'effort par l'intermédiaire d'interface à retour d'effort ?
- Quelle est l'influence de ces choix sur la qualité du retour d'effort pour ce type d'application ?

Pour répondre à ces trois points, nous proposerons, tout d'abord, d'étudier une architecture pair-à-pair afin de réaliser des tâches de collaboration en utilisant la modalité haptique. Pour garantir un retour d'effort stable, nous développerons et expérimenterons deux approches en se basant sur des méthodes issues du domaine de la téléopération à retour d'effort. Nous explorerons complémentarément à ce retour d'effort, un retour d'effort événementiel pour tenter d'améliorer l'interaction avec l'environnement virtuel. Tout au long de ces différents chapitres, nous évaluerons expérimentalement, de manière précise, le comportement du retour d'effort en présentant les performances en termes de qualité du retour d'effort et de suivi pour les différents choix retenus et approches développées.

L'ensemble de nos contributions ont été mis en œuvre dans une application logicielle distribuée associée à des interfaces à retour d'effort permettant de réaliser des tâches collaboratives que nous désignerons sous le terme de plateforme collaborative à retour d'effort.

Dans ces travaux, nous nous sommes limités au cas simplifié d'un retard fixe, bien que dans le cas d'une collaboration distante ce dernier ne soit pas nécessairement fixe. En effet, la variation du retard est pénalisante en termes de stabilité et en termes de perception.

Dans le [premier chapitre](#), nous présenterons une synthèse de l'état de l'art suivant quatre axes. Le premier axe permettra d'introduire les problématiques et méthodes spécifiques à la simulation physique interactive d'objet rigide. Dans un second temps, nous détaillerons les travaux permettant de réaliser un retour d'effort dans un environnement virtuel. Les méthodes issues de la téléopération traitant du retour d'effort à distance seront ensuite présentées. Enfin, nous aborderons, au travers de plateformes collaboratives haptiques existantes, les choix liés à l'architecture de ces plateformes et les problèmes rencontrés lors de réalisation de tâches collaboratives distantes.

Le [second chapitre](#) décrira la plateforme haptique collaborative élaborée dans le cadre de cette thèse. Nous avons choisi d'utiliser une architecture distribuée (pair-à-pair) pour cette plateforme. Les aspects logiciels de simulation et d'architecture seront ensuite exposés. La problématique de divergence d'états des différents pairs de la plateforme sera particulièrement abordée. Nous proposerons de tolérer cette divergence et décrirons une méthode simple, basée sur les détecteurs de collision afin de signaler l'apparition d'incohérences.

Les différents pairs ou nœuds de cette plateforme distribuée doivent être reliés entre eux d'une manière qui permettent de réaliser un retour d'effort stable. Les problèmes de stabilité qui découlent seront abordés dans les deux chapitres suivants.

Le [troisième chapitre](#) exposera une méthode d'asservissement stable permettant de minimiser la divergence entre les états des différents pairs. Elle est basée sur des couplages virtuels ressorts-amortisseurs. Cette méthode sera présentée dans le cas de retards constants faibles (entre 50 ms et 100 ms). Nous introduirons une méthode permettant de compenser les efforts de traînée introduits par cette méthode afin d'obtenir un mouvement libre, quasiment transparent. Une évaluation de cette méthode sera présentée dans un cas de manipulation à 6 degrés de liberté.

Le [quatrième chapitre](#) décrira une seconde méthode dans le cas de retards fixes quelconques. Nous avons adapté la transformation en variables d'onde à notre cas spécifique d'environnement virtuel partagé. Nous développerons une formulation articulaire qui sera ensuite utilisée sur la plateforme de collaboration dans des cas du prototypage virtuel.

Le [cinquième chapitre](#) abordera l'utilisation de retour haptique événementiel. Les méthodes proposées précédemment dégradent d'une manière ou d'une autre l'information fournie par le retour d'effort (en réduisant la transparence, en diminuant l'impédance transmise du système). Nous envisagerons d'ajouter à ce retour d'effort dégradé une information transitoire haute fréquence relative à l'impact. Une expérience sur un cas simple sera présentée afin d'évaluer le bénéfice de l'haptique événementielle ajoutée à un retour haptique classique dégradé.

Enfin, nous concluons sur les résultats principaux et sur la pertinence du retour d'effort dans le cas de tâches collaboratives distantes dédiées au prototypage virtuel.



# État de l'art

## Sommaire

<b>1.1 Simulation physique interactive d'univers virtuel</b>	<b>6</b>
1.1.1 Algorithmes de détection de collision pour les objets rigides	6
1.1.2 Simulation physique	10
1.1.3 Moteur physique	14
<b>1.2 Retour haptique</b>	<b>15</b>
1.2.1 Interfaces à retour d'effort	15
1.2.2 Stabilité du retour d'effort	17
<b>1.3 Interactions haptiques distantes</b>	<b>26</b>
1.3.1 Couplage virtuel	27
1.3.2 Observateur et contrôleur de passivité	28
1.3.3 Variables d'onde	29
<b>1.4 Collaboration et coopération distantes</b>	<b>36</b>
1.4.1 Architecture réseau des plateformes collaboratives	36
1.4.2 Retour haptique sur une tâche collaborative	42
1.4.3 Influence du retard sur la coopération	43
<b>Conclusion</b>	<b>45</b>

UNE manipulation collaborative distante avec retour d'effort est réalisée par l'intermédiaire d'un système basé sur plusieurs éléments. Ce système, que nous appellerons plateforme collaborative dans la suite du manuscrit est constitué d'un ou plusieurs moteurs physiques permettant de simuler l'univers virtuel. Le retour d'effort est ensuite réalisé par l'intermédiaire d'une interface haptique. Enfin, les interactions collaboratives distantes sont permises car les sites distants peuvent communiquer entre eux par l'intermédiaire d'un réseau informatique (voir Figure 1.1). Ce médium de communication introduit un retard lors de la transmission des informations, retard qui pose des problèmes de stabilité au niveau du retour d'effort.

À partir de ces différents éléments, quatre aspects principaux peuvent être mis en évidence : la simulation de l'univers virtuel via le moteur physique, l'ajout du retour d'effort stable à un environnement virtuel, les problèmes de stabilités introduit par le retard lors de

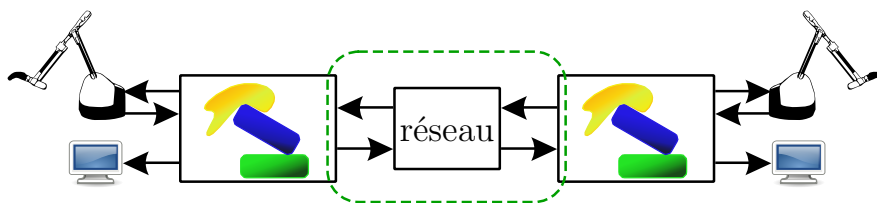


Figure 1.1 – Exemple d'application permettant une interaction collaborative distante



retour d'effort distant et enfin, la collaboration distante et l'influence de la topologie de la plateforme collaborative et du retard sur cette collaboration. Ces quatre points forment les quatre axes de la synthèse de l'état de l'art que nous présenterons dans ce chapitre. Cette synthèse se limitera aux éléments utilisés en particulier dans le cas d'univers virtuel avec retour d'effort. Elle ne sera donc pas forcément complète dans chacun des domaines abordés.

Ce chapitre présentera tout d'abord (Section 1.1) les problématiques liées à la simulation interactive d'univers physiques dans des environnements virtuels et les différentes méthodes de simulations présentes dans la littérature.

La seconde partie de ce chapitre (Section 1.2) s'intéressera particulièrement au retour d'effort associé aux environnements virtuels. Une première partie décrira brièvement les interfaces haptiques. Dans une deuxième partie, les méthodes permettant de stabiliser un système de réalité virtuel avec retour d'effort seront présentées.

La troisième partie (Section 1.3) traitera plus particulièrement du problème des retards dans les systèmes à retour d'efforts en présentant les travaux issus du domaine de la téléopération. Certains résultats de ces travaux ont déjà été appliqués au cas des environnements virtuels collaboratifs avec retour d'effort.

Enfin, la dernière partie (Section 1.4) abordera les notions de collaboration et de coopération distantes et les solutions techniques permettant de les réaliser. Cette partie se focalisera sur les cas collaboratifs avec retour d'effort. Une attention particulière sera portée sur les principales plateformes proposées dans la littérature, en particulier, celles qui permettent d'effectuer des tâches de collaboration avec un retour d'effort. Les effets du retard vis-à-vis de la manipulation haptique sont soulignés.

## 1.1 Simulation physique interactive d'univers virtuel

Les simulations physiques utilisées dans le cadre du prototypage virtuel permettent de simuler l'évolution d'objets soumis à des contraintes bilatérales ou unilatérales. Pour simuler un tel univers virtuel, la simulation physique doit résoudre un système d'équation sous contraintes constitué d'une loi d'évolution et d'un modèle de contact. L'utilisation d'un modèle de contact nécessite des informations concernant les contacts entre les objets de l'univers virtuel. Généralement, dans le cadre de la simulation d'objets rigides que nous considérons ici, ces informations sont fournies à la simulation physique par un détecteur de collision. Ainsi, un univers virtuel composé d'objets rigides est le plus souvent simulé grâce à deux éléments : un détecteur de collision et une simulation physique regroupés au sein d'un moteur physique.

Nous présenterons dans un premier temps les algorithmes de détection de collision pour les objets rigides utilisés par les détecteurs de collision. Dans un deuxième temps les lois physiques gouvernant l'évolution des objets rigides seront détaillées. Enfin les schémas d'intégration permettant de calculer l'évolution de l'univers virtuel seront évoqués.

### 1.1.1 Algorithmes de détection de collision pour les objets rigides

La détection de collision a donné lieu à de nombreux travaux, car il existe de nombreuses approches possibles adressant une grande variété de problèmes rencontrés en réalité virtuelle [LM08]. Le terme détection de collision regroupe d'une part le fait de fournir une information de configuration de contact mais aussi celui de fournir des informations de proximité ou d'interpénétration. Chaque algorithme de détection de collision ne peut pas forcément fournir toutes ces informations.

Il existe trois axes permettant de classifier les algorithmes de détection de collision. Un algorithme de détection de collision est d'abord caractérisé par la représentation géométrique des objets virtuels qu'il est capable de traiter. Ensuite, les algorithmes de détection de collision n'ont pas tous la même granularité, certains permettent de détecter deux objets en collision, alors que d'autres permettent d'identifier le lieu précis des contacts à la surface des objets. Enfin, ces algorithmes peuvent être *continus* ou *discrets*.

Premièrement, un algorithme de détection de collision est caractérisé par la représentation géométrique des objets virtuels qu'il peut traiter. Ces objets pouvant être notamment représentés sous la forme de nuage de points, soupe de polygones ou sous forme de primitives. Dans le cadre du prototypage virtuel, la représentation généralement utilisée est celle sous forme de soupe de polygones, en particulier, de triangles.

Le deuxième axe de classification des algorithmes de détection de collision est relatif à l'échelle à laquelle ces algorithmes opèrent. En effet, la phase de détection de collision peut être très coûteuse en temps de calcul. Cette phase est généralement divisée en plusieurs étapes qui ont des granularités différentes, caractéristiques des algorithmes de détection de collision :

- La première phase (*broad phase collision detection*) consiste à détecter des paires d'objets potentiellement en contact. Ce test par paire a une complexité en  $O(n^2)$  où  $n$  est le nombre d'objets de la scène [CLMP95]. Afin de diminuer cette complexité, trois approches peuvent être utilisées :
  - l'utilisation de volumes englobants, notamment des sphères [Hub95] ou des boîtes englobantes orientées (OOBB) [GLM96].
  - une partition de l'espace, comme une subdivision binaire (BSPs) [NAT90] binaire à  $k$ -dimensions (kD-tree)
  - la cohérence spatio-temporelle [CLMP95]

Les paires d'objets fournis à l'issue de cette étape peuvent être dans trois configurations : proche mais pas en contact, en contact exact, interpénétrés.

- Une phase intermédiaire (*mid phase collision detection*) utilisant les mêmes algorithmes permet de déterminer quelles portions des objets de chaque paire est potentiellement en contact ou interpénétrés. Cette phase intermédiaire peut être fusionnée avec la première étape en une seule phase si le nombre d'objets est faible.
- La dernière étape, (*narrow phase collision detection*) permet de calculer les informations de contact en fonction des représentations géométriques des objets identifiés comme étant potentiellement en contact lors des phases précédentes.

Les algorithmes détaillés par la suite sont utilisés dans cette dernière phase. Cette dernière étape revêt une importance particulière dans nos travaux, car les environnements virtuels dédiés au prototypage comportent généralement très peu d'objets mobiles. Les deux premières phases ne sont donc pas toujours présentes dans ces simulations physiques. Le choix des algorithmes détermine le choix de la méthode de résolution des contacts et donc de l'évolution du système. Ce choix a donc un impact majeur sur le comportement du système.

Parmi ces algorithmes, deux familles peuvent être distinguées. Les algorithmes discrets donnent une information de contact à un pas de temps donné. Les algorithmes de détection continue indiquent l'instant précis du contact. Le choix de l'algorithme influe ainsi sur le choix de la gestion des contacts au sein du moteur physique.

Les algorithmes de détection de collision décrits par la suite sont utilisés dans la der-

nière étape de détection de collision (*narrow phase collision detection*). Ils utilisent en outre une représentation géométrique à base de soupe de polygones. Les méthodes spécifiques aux objets concaves, aux objets décrits analytiquement et aux représentations sous forme de nuage de points ne seront donc pas abordées. Une présentation plus complète est disponible dans [LM08].

### 1.1.1.1 Algorithme discret

Les algorithmes de détection de collision discrets décrivent à un instant donné l'état des contacts entre les objets virtuels. Pour cela, les trajectoires des objets sont discrétisés dans le temps. Ces algorithmes sont les plus utilisés car ils sont particulièrement efficaces sur des objets ayant une géométrie complexe.

#### 1.1.1.1.a Voxmap-Point-Shell : Voxelization

L'algorithme de détection de collision Voxmap-Point-Shell (VPS) proposé par McNeely *et al.* [MPT99] fournit une information sur les collisions entre un objet représenté sous forme d'un nuage de points orienté (*point shell*) et un second objet représenté par une représentation volumétrique (*voxmap*) en voxels (voir Figure 1.2a). L'algorithme consiste ensuite à déterminer si un point du *point shell* se trouve à l'intérieur d'un voxel de la *voxmap*. VPS est couramment utilisé pour détecter les collisions entre des objets ayant une représentation géométrique sous forme de soupe de polygones. Dans ce cas, VPS utilise des représentations approchées de la géométrie des objets. Son avantage est la grande rapidité des requêtes permettant d'indiquer la présence d'un point dans un voxel. Cet algorithme nécessite préalablement un calcul hors-ligne permettant de construire la *voxmap* et la *point shell*.

Lorsque la scène comporte plus de deux objets, chaque objet doit posséder les deux représentations sous forme de *voxmap* et sous forme de *point shell*.

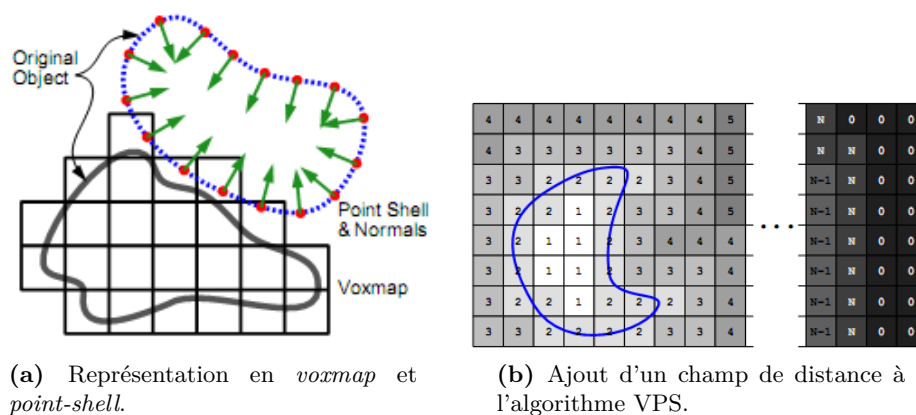


Figure 1.2 – Algorithme VPS issu de [MPT06]

VPS permet ainsi de détecter à posteriori des interpénétrations et de déterminer la direction et la profondeur de celle-ci si elle est inférieure à un voxel.

Pour offrir des informations de proximités et donc permettre à la simulation physique d'éviter les interpénétrations, un champ de distances peut être ajouté en discrétisant également l'espace environnant de la *voxmap* (voir Figure 1.2b). À ces voxels sont associés un état indiquant si le *voxel* se situe dans l'objet, à sa surface, ou à une distance de  $n$  voxels de la surface [MPT06].

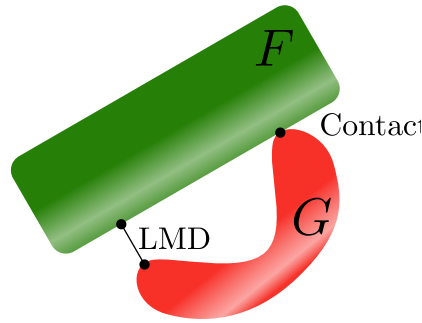


Figure 1.3 – Algorithme LMD. Il y a un contact et une distance minimum locale

Il est à noter que cet algorithme a été étendu au cas d'objets déformables, ainsi qu'à des objets très complexes en utilisant des techniques de niveaux de détails (*LOD - Level Of Details*) [BJ08].

#### 1.1.1.1.b Distance Locale Minimum (*Local Minimal Distance*)

L'algorithme LMD (*Local Minimal Distance*) proposé par Johnson *et al.* fournit des distances locales minimum entre des couples de points situés à la surface de deux objets et non pas seulement une distance globale entre ces deux objets [JC01, JWC05].

En reprenant les notations de [JC01], soit deux surfaces paramétrées  $\mathbf{f}(u, v)$  et  $\mathbf{g}(s, t)$ , la distance entre ces deux surfaces est définie par  $D(u, v, s, t) = \|\mathbf{f}(u, v) - \mathbf{g}(s, t)\|^2$ . Les distances minimales sont données par les solutions du système d'équations suivant :

$$\begin{cases} (\mathbf{f} - \mathbf{g}) \cdot \frac{\partial \mathbf{f}}{\partial u} = 0 \\ (\mathbf{f} - \mathbf{g}) \cdot \frac{\partial \mathbf{f}}{\partial v} = 0 \\ (\mathbf{f} - \mathbf{g}) \cdot \frac{\partial \mathbf{g}}{\partial s} = 0 \\ (\mathbf{f} - \mathbf{g}) \cdot \frac{\partial \mathbf{g}}{\partial t} = 0 \end{cases} \quad (1.1)$$

Les normales aux surfaces  $\mathbf{f}$  et  $\mathbf{g}$  aux points  $(u, v)$  et  $(s, t)$  sont donc colinéaires pour une distance minimale. Elles sont portées par le bipoint défini par  $(u, v)$  et  $(s, t)$ . Lors d'un contact, cette distance est nulle. Si les objets sont représentés par une soupe de polygones, le calcul de ces normales est obtenu en utilisant une hiérarchie de volumes englobant associés à des cônes permettant de décrire le lieu des normales. Si la soupe de polygones forme une variété différentielle, les cônes de normales peuvent être affinés. L'algorithme est particulièrement adapté à ce cas précis.

Merlhiot a étendu cet algorithme en se basant sur une analyse géométrique de la surface [Mer07]. L'introduction d'une formulation polaire des cônes permet de traiter plus efficacement les polygones, les arrêtes ou les points isolés dans le cas d'une soupe de polygones. Il introduit également la notion de quasi-LMD : les cônes de normales utilisés par le système 1.1 sont légèrement élargis par un angle de tolérance ce qui rend la détection de collision plus robuste.

#### 1.1.1.2 Détection de collision continue

Les deux algorithmes présentés précédemment fournissent à un instant donné des informations de contacts, de proximités ou d'interpénétrations. Les algorithmes de détection de

collision continue permettent, en prenant le temps comme paramètre, de déterminer l'instant du contact. Il faut pour cela considérer à la fois la géométrie des objets et leurs trajectoires. Le problème ainsi formulé est résolu analytiquement [Can86]. Redon *et al.* [RKC00] ont amélioré cette méthode en utilisant une hiérarchie de volumes englobants associée à l'arithmétique d'intervalle. La trajectoire d'un objet est donnée par le moteur physique sous la forme d'un ensemble de positions fourni à chaque pas de temps. Cette trajectoire est ensuite approximée entre deux états du système par un vissage. Cette approximation permet de fournir une formulation analytique simple de la trajectoire en prenant en compte les rotations et les translations. Dans certains cas il peut être avantageux de remplacer le vissage par une série de vissage [RKL04] afin d'établir une meilleure approximation de la trajectoire.

Les algorithmes de détection de collision continue permettent ainsi d'éviter que les objets puissent se traverser. En effet, les méthodes de détection de collision discrètes fournissent des informations de proximités, de contacts ou d'interpénétration à un instant donné. Entre deux états, ces informations ne sont a priori pas déterminées. Si une collision est susceptible d'apparaître entre ces deux états, l'algorithme de détection de collision peut ne pas la détecter. Dans le cas d'objets rapides ou d'objets fins, cette limitation peut donner l'impression qu'un objet a traversé un autre objet. En détectant l'instant de la collision, les méthodes continues prennent en compte ces cas de figure. Dans le cadre des méthodes discrètes, des mécanismes supplémentaires, appelés *backtracking*, peuvent traiter ces cas [MW88] en vérifiant le long de la trajectoire si il n'y a pas de tels situations.

### 1.1.2 Simulation physique

Les informations de contact issues de la détection de collision présentée précédemment sont utilisées dans la simulation physique afin de calculer l'état de l'environnement virtuel au pas de temps suivant. Dans le cadre d'environnements virtuels avec un rendu visuel et haptique, l'état du système doit être mis à jour à une fréquence élevée (1 kHz idéalement, voir Section 1.2). La simulation physique doit ainsi fournir un état de l'univers virtuel dans le pas de temps imparti.

La simulation physique itère à chaque pas de temps les 3 étapes suivantes :

1. résolution du mouvement à l'aide des équations d'évolution,
2. résolution du contact à partir des données issues de la détection de collision,
3. incrément d'un pas de temps

Les sections suivantes présentent les modèles physiques utilisés puis les problématiques liées à la résolution des équations d'évolutions construites à partir de ces modèles. Les méthodes les plus anciennes proviennent du domaine de l'informatique graphique, les méthodes plus récentes ont généralement été proposées dans le cadre de l'utilisation d'environnements virtuels avec rendu haptique.

#### 1.1.2.1 Équations d'évolution d'objets rigides

Dans le cas du prototypage virtuel avec des objets rigides, les objets ont des dimensions comprises entre quelques millimètres et quelques mètres. Les équations d'évolution alors utilisées pour décrire le comportement de ces objets sont généralement celles de la mécanique Newtonienne. Si  $\mathbf{q}$  est la position généralisée d'un objet, l'évolution de chaque objet peut être écrite à partir l'équation d'Euler-Newton :

$$\mathbf{M}(\mathbf{q}, \dot{\mathbf{q}}, t)\ddot{\mathbf{q}} + \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}, t)\dot{\mathbf{q}} + \mathbf{B}\dot{\mathbf{q}} = \sum_i \mathbf{f}_i(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (1.2)$$

$\mathbf{M}$  est la matrice d'inertie généralisée,  $\mathbf{N}$  est une matrice exprimant les forces de Coriolis,  $\mathbf{B}$  est une matrice de viscosité.  $\sum \mathbf{f}_i$  est l'ensemble des forces externes s'appliquant à l'objet considéré. Les vitesses des objets simulés étant suffisamment faible (quelques dixièmes de  $\text{ms}^{-1}$  ou quelques dixièmes de  $\text{rads}^{-1}$ ), les effets de Coriolis sont alors généralement ignorés dans la plupart des modélisations à cause de leur faible influence sur l'évolution du système. Les effets dus à l'inertie ne sont pas toujours intéressants dans une tâche de prototypage virtuelle. Dans ce cas, une formulation quasi-statique peut aussi être retenue :

$$B\dot{q} = \sum_i \mathbf{f}_i(q, \dot{q}, t) \quad (1.3)$$

### 1.1.2.2 Modèle de contact pour des objets rigides parfaits

Le mouvement des objets est régi par une équation d'évolution, il est en même temps contraint par les contacts entre objets. Les deux modèles classiquement utilisés sont la loi de Signorini pour la non-interpénétration au contact et la loi de Coulomb pour les frottements entre les objets rigides.

**Non-interpénétration** Dans un univers virtuel, les objets rigides parfait peuvent se déplacer librement mais ne doivent pas s'interpénétrer. La loi de Signorini permet d'établir ce lien par les contraintes suivantes [Sig33] :

$$\begin{cases} 0 \leq \mathbf{d} \\ 0 \leq \mathbf{f}_n \\ \mathbf{d} \cdot \mathbf{f}_n = 0 \end{cases} \quad (1.4)$$

A chaque point d'un objet une distance  $\mathbf{d}$  le séparant d'un autre objet peut être associée. Elle est positive si les objets ne s'interpénètrent pas, négative dans le cas contraire. A ce point, une composante normale  $\mathbf{f}_n$  d'une force de contact  $\mathbf{f}_c$  peut être exprimée.  $\mathbf{f}_n$  est portée par la normale  $\mathbf{n}$  au contact. La loi de Signorini exprime, d'une part, le fait que la composante normale  $\mathbf{f}_n$  de la force de contact est positive ou nulle (la contrainte est *unilatérale*) et, d'autre part, que la distance de contact  $d$  est positive ou nulle (condition de non-interpénétration). L'orthogonalité de  $\mathbf{f}_n$  et  $d$  impose, au contact, une distance nulle et une force normale de contact positive ou nulle. En l'absence de contact, cette force est nulle et la distance est positive.

**Frottement** Le frottement entre des objets rigides est un frottement sec. La force de frottement peut être modélisée par la loi de Coulomb :

$$\mathbf{f}_c = \mathbf{f}_t + \mathbf{f}_n \quad (1.5)$$

$$f_t \geq f_n \tan(\phi) \quad (1.6)$$

où  $\mathbf{f}_c$  et  $\mathbf{f}_n = f_n \mathbf{n}$  sont définies précédemment,  $\mathbf{f}_t$  est la force tangentielle ou force de frottement et  $f_t$  sa norme et  $\tan(\phi)$  est le coefficient de frottement. Lorsqu'une force tangentielle externe  $\mathbf{f}_e$  est appliqué à l'objet, deux cas peuvent être distingués. Premièrement, l'adhérence ( $f_e = f_t < f_n \tan(\phi)$ ) où la composante tangentielle est trop faible pour mettre un objet en mouvement. Deuxièmement, le glissement ( $\mathbf{f}_e > f_t = f_n \tan(\phi)$ ) où la force tangentielle est suffisante pour déplacer l'objet.

### 1.1.2.3 Résolution du contact

Les méthodes de résolution du contact permettent au moteur de physique d'intégrer les lois de contacts présentée ci-dessus aux équations d'évolution de l'univers virtuel.

#### 1.1.2.3.a Approche par pénalités

Historiquement, la première approche utilisée dans les environnements virtuels avec retour d'effort est la méthode par pénalités, proposée par Moore *et al.* [MW88]. Elle se base sur une régularisation de la contrainte de non-interpénétration (loi de Signorini *pénalisée*)<sup>1</sup>.

La loi de non-interpénétration (Equation 1.4) devient :

$$\begin{cases} f_n = -kd - bd & \text{si } d < 0 \\ f_n = 0 & \text{si } d \geq 0 \end{cases} \quad (1.7)$$

où  $d$  est la distance d'interpénétration au point de contact portée par la normale  $\mathbf{n}$  au contact,  $k$  et  $b$  sont des paramètres de pénalisations. Cette formulation est celle d'un effort généré par un ressort-amorti de raideur  $k$  et de viscosité  $b$ . Des formulations non linéaires mais numériquement plus complexes à résoudre peuvent être aussi utilisées.

Des contacts sur des objets rigides peuvent être simulés avec cette approche en utilisant des coefficients  $k$  et  $b$  importants. Cependant, de grandes valeurs de  $k$  et  $b$  ont tendance à introduire des instabilités lors de la résolution numérique (voir Section 1.2.2.1.a).

La loi ainsi reformulée ne garantit plus l'absence d'interpénétration entre les objets rigides. Afin de limiter les interpénétrations entre les objets, la distance d'interpénétration peut être majorée par une distance  $\delta$  [MPT99]. Le contact n'est ainsi plus réalisé à la surface de l'objet mais à la surface de l'enveloppe définie par la surface  $\delta$ . Cette distance peut se remarquer visuellement par un espace entre les objets virtuels lors du contact.

En ne respectant pas strictement les contraintes de non-interpénétrations, la résolution d'une situation de contact en un lieu peut générer d'autres contacts en d'autre lieu des objets. Dans le cas d'un objet fortement contraint, par exemple une insertion cylindre-cylindre, ces nouvelles interpénétrations peuvent à leur tour générer d'autres interpénétrations et ainsi de suite sans converger vers un état stable. Des instabilités peuvent ainsi apparaître [RKLMO4].

L'approche par pénalité ne permet donc pas de simuler tous les cas de figure du prototype virtuel de manière stable.

#### 1.1.2.3.b Approche par contraintes

L'approche par contraintes est directement basée sur la loi de Signorini et permet contrairement aux approches par pénalité de garantir la non-interpénétration.

Dans le cas sans frottement, en se basant sur la formulation de la loi de Signorini (Equation 1.4) et sur les équations d'évolution (Equation 1.2 ou Equation 1.3), le système d'équations gouvernant l'évolution d'un objet peut être construit :

$$\begin{cases} \mathbf{q}_{obj} = \mathbf{A}\mathbf{f}_{n_{obj}} + \mathbf{b} \\ 0 \leq \mathbf{q}_{obj} \\ 0 \leq \mathbf{f}_{n_{obj}} \\ \mathbf{f}_{n_{obj}}^T \mathbf{q}_{obj} = 0 \end{cases} \quad (1.8)$$

1. Elle peut se rapprocher d'une modèle de contact de compliant suivant une loi de Kevin-Voigt

$\mathbf{q}_{obj} \in \mathbb{R}^n$  est la paramétrisation du mouvement de l'objet à chaque point de contact (espace des contacts) obtenu pendant la première phase de calcul de la simulation physique.  $\mathbf{f}_{n_{obj}} \in \mathbb{R}^n$  est l'ensemble des forces normales aux points de contact.  $\mathbf{A} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{b} \in \mathbb{R}^n$  sont données par l'Equation 1.2. Cette formulation est celle d'un problème de complémentarité linéaire (*LCP : Linear complementarity problem*). Si  $\mathbf{A}$  est défini positive, le système admet des solutions et n'importe quel algorithme permettant de résoudre numériquement un problème quadratique peut être adopté. Parmi ceux-ci, les algorithmes couramment utilisés sont les algorithmes de Gauss-Seidel ou de Dantzig qui sont des algorithmes itératifs ou l'algorithme de Lemke [Bar93].

L'introduction du modèle de frottement basé sur les lois de Coulomb introduit des non-linéarités dans la formulation du LCP. Le système ainsi construit est un NLCP (*Non Linear Complementarity Problem*), numériquement plus lent à résoudre. Le système peut être ramené à un *LCP* en formulant la loi de Coulomb avec des cônes de frottement discret [APS99].

Il est également à noter que cette formulation sous forme d'un LCP est équivalente à la formulation en programmation quadratique suivant  $\min Q(\mathbf{f}_{n_{obj}}, \mathbf{q}_{obj}) = \min(\mathbf{f}_{n_{obj}}^T \mathbf{q}_{obj})$  avec comme contrainte  $\mathbf{q}_{obj} \geq 0$  et  $\mathbf{f}_{n_{obj}} \geq 0$ . Une autre formulation équivalente a été proposée par Redon *et al.* [RKC02] en utilisant le principe de moindre contrainte de Gauss. Cette méthode permet une résolution dans l'espace cartésien et non pas dans l'espace des contacts. Cette méthode semble être plus rapide lorsque le nombre de points de contact est important. Une information sur les degrés de liberté d'un objet peut être immédiatement déduite.

L'approche par contraintes à l'avantage d'être stable et de fournir des informations de contact exact et donc de respecter les non interpénétrations entre objets rigides. Cependant, la résolution du LCP est plus coûteuse en temps de calcul que la résolution d'un système utilisant une approche par pénalité, notamment, lorsque le nombre de points de contact est important. Il y a donc un compromis lors du choix de la méthode de résolution des contacts entre d'un côté une résolution suffisamment rapide et d'autre part une résolution exacte et plus stable.

#### 1.1.2.3.c Approche impulsienne

Cette approche a été proposée par Mirtich et Canny [MC95], elle permet d'intégrer différentes contraintes (comme l'impact, le glissement) d'une manière unifiée, sous forme d'impulsions. Dans le cas de l'impact, lorsque deux objets rigides entrent en collision, ils sont soumis à une impulsion de valeur opposée les repoussant, permettant d'éviter une interpénétration. Le modèle de collision est basé sur un modèle d'impact Newtonien. Cette méthode introduit un certain nombre d'approximation afin de traiter comme impulsion des phénomènes qui ne sont pas impulsions, comme l'adhérence. Dans ce dernier cas, les objets physiques n'auront pas forcément un comportement réaliste. Mirtich et Canny [MC95] ont donc proposé d'utiliser des approches hybrides basées sur les approches par contraintes et impulsienne.

#### 1.1.2.4 Schéma d'intégration

Les lois d'évolution présentées précédemment sont souvent décrites sous la forme d'équations différentielles, notamment temporelle du type  $F_d(t, X(t), \dot{X}(t), \dots, X^{(n)}(t))$  où  $X(t)$  est l'état du système à l'instant  $t$ . Les méthodes d'intégration couramment utilisées étant discrètes, les solutions calculées sont par nature une approximation des solutions théoriques. Le choix du pas d'intégration est généralement soumis à un compromis entre le temps de calcul



nécessaire à sa résolution numérique et la précision des solutions. Deux familles d'intégrateur numérique peuvent être distinguées : les schémas d'intégration implicites et les schémas d'intégration explicites. Nous détaillons par la suite les méthodes les plus courantes rencontrées pour la simulation d'objets rigides. Une synthèse beaucoup plus complète est disponible dans [DR84].

Les schémas explicites utilisent l'état du système calculé au pas de temps précédent  $X(t - \Delta T)$ . Elles sont de la forme  $X(t) = f(X(t - \Delta T))$  où  $\Delta T$  est le pas de temps d'intégration. Dans le cas des équations différentielles du premier ordre en fonction du temps, de la forme  $\dot{X}(t) = F(t, X(t))$ , les méthodes explicites les plus courantes sont :

- Schéma d'Euler explicite :  $X(t) = X(t - \Delta T) + \Delta T F(t - \Delta T, X(t - \Delta T))$
- Méthodes de Runge-Kunta, habituellement d'ordre 2 ou 4

Les schémas implicites utilisent à la fois l'état du système calculé au pas de temps précédent et l'état du système du pas de temps courant. Ce dernier n'étant pas encore déterminé, il est défini implicitement par  $X(t) = g(X(t - \Delta t), X(t))$ .  $g$  est généralement obtenu en reformulant la fonction  $f$ . Le schéma implicite le plus courant est :

- Schéma d'Euler implicite :  $X(t) = X(t - \Delta T) + \Delta T F(t, X(t))$

Les méthodes explicites sont généralement plus rapides que les méthodes implicites mais tendent à être moins stables. Elles nécessitent généralement un pas de temps plus petit que les méthodes implicites. Les schémas implicites ont, en outre, tendance à être plus stables lorsque les lois d'évolutions ne sont pas continues, comme les lois de contacts.

Le choix d'un schéma d'intégration est donc soumis à un compromis entre une stabilité nécessaire de l'intégration numérique, le pas de temps retenu le temps alloué à l'intégration et de la méthode de résolution des contacts abordés dans la partie suivante.

### 1.1.3 Moteur physique

Il existe donc de nombreuses approches pour traiter les différents points d'une simulation numérique. Le compromis est à trouver entre le temps de calcul, la stabilité et la précision de la simulation physique. De plus, certaines approches sont plus naturelles à associer. Notamment, les méthodes de résolutions de contacts par contraintes peuvent utiliser facilement les informations de contacts fournies par une méthode de collision continue. La méthode par pénalité peut s'associer facilement avec un algorithme de détection de collision du type VPS. Ces deux méthodes étant relativement rapides, le temps de calcul requis est faible. Un univers virtuel d'autant plus complexe peut être simulé.

Dans le cadre de l'utilisation d'un environnement virtuel avec un rendu haptique, les deux contraintes les plus importantes sont le respect du temps de rafraîchissement de l'interface haptique et la garantie de la stabilité du système. Ainsi, pour des univers virtuels relativement petits, une approche par contrainte associée à un schéma d'intégration numérique implicite sera préférée. Si cet univers doit pouvoir simuler des tâches d'insertion ou respecter strictement des contraintes de non pénétrations, le détecteur de collision de type LMD ou détecteur de collision continue seront choisis.

Afin de réduire le temps de calcul pris par le moteur physique, des approches multi-fréquentielles s'avèrent intéressantes. Otaduy et Lin [OL06] ont adopté une approche où la détection de collision n'est pas réalisée à chaque pas de temps. La résolution de contact, qui est réalisée à chaque pas de temps, se base sur un modèle linéarisé des contacts construit à partir des informations fournies par le détecteur de collision. La simulation physique utilise dans ce cas un modèle linéarisé pendant un certain nombre de pas temps jusqu'à ce que le

moteur de collision réactualise ce modèle. Otaduy et Lin ont proposé aussi un détecteur de collision basé sur les niveaux de détails (*LODs : Level of Details*) [OL03]. Cette approche utilisant plusieurs fréquences spatiales permet de diminuer le temps requis par la détection de collision.

## 1.2 Retour haptique

La quasi-totalité des univers virtuels fournit un retour visuel. Moins souvent, un retour sonore est également proposé afin d'enrichir l'interaction entre l'environnement virtuel et les utilisateurs. De la même manière, associer un dispositif haptique<sup>2</sup> à un environnement virtuel permet de solliciter une modalité supplémentaire de l'être humain, le sens du toucher.

L'adjectif haptique recouvre deux aspects du sens du toucher. La perception *tactile* qui désigne la perception à la surface de la peau, et la perception *kinesthésique* qui désigne la perception des efforts et des positions. La première est permise par la fusion des informations fournies par les récepteurs (thermorécepteurs, mécanorécepteurs et nocicepteurs) présent dans le derme, la seconde par la fusion des informations provenant des récepteurs répartis dans les muscles, les tendons et la peau [LL88].

De la même manière qu'il existe deux aspects au retour haptique, les interfaces suivent une classification en deux familles, interfaces tactiles d'une part et interface kinesthésiques ou à retour d'effort d'autre part. Cette classification n'est pas stricte car certaines interfaces proposent un rendu tactile et kinesthésique en employant plusieurs types d'actionneurs dédiés.

Les interfaces kinesthésiques sont aussi des interfaces d'entrée d'information. À la différence des autres modalités, le toucher permet à la fois de ressentir et d'agir sur l'environnement. Ce fonctionnement bilatéral introduit un couplage entre les utilisateurs et l'environnement virtuel qui pose des problèmes spécifiques de stabilité quand à l'utilisation d'interfaces kinesthésiques.

La suite de ce chapitre présente les interfaces à retour d'effort ainsi que les méthodes permettant de lier de telles interfaces à un univers virtuel. Les problèmes de stabilité sont particulièrement soulignés.

### 1.2.1 Interfaces à retour d'effort

#### 1.2.1.1 Retour haptique

De la même manière que pour le rendu visuel, le rendu haptique nécessite une certaine fréquence de rafraîchissement. Afin de fournir une bonne perception haptique, la fréquence de rafraîchissement doit être d'environ 1 kHz [SB97]. Le rendu haptique est donc plus exigeant que le rendu visuel qui lui nécessite un rafraîchissement de 50 Hz. Le moteur physique simulant l'univers virtuel doit donc être suffisamment rapide pour respecter cette contrainte.

Une interface à retour d'effort permet d'associer un effort  $F$  à une vitesse  $\dot{x}$ . À partir de cette définition, nous pouvons distinguer deux familles possibles d'interfaces à retour d'effort dépendant du mode de commande :

- Type impédance ( $F = Z\dot{x}$  avec  $Z$  l'impédance de l'interface) : l'interface haptique mesure les positions et vitesses de l'utilisateur et applique des efforts à celui-ci. Un mouvement libre peut être facilement rendu en appliquant une force nulle. Ces interfaces sont les plus répandues.

---

2. Le terme haptique vient du grec *haptomai* signifiant "je touche"

- Type admittance ( $\dot{x} = YF$  avec  $Y$  l'admittance de l'interface) : l'interface haptique mesure les efforts de l'utilisateur et contraint la position et la vitesse de l'interface. Un mouvement libre est plus difficile à restituer.

Choisir un mode de commande pour une interface permet d'arrêter un certains nombres de choix en matière de conception de l'interface (capteur de positions ou d'efforts, etc.).

Les interactions entre objets rigides recouvrent un large champ d'impédance, entre un mouvement totalement libre ( $Z = 0$ ), où il n'y a pas de résistance au déplacement et un mur fixe ( $Z = \infty$ ), où quelque soit la force appliquée contre ce mur, la vitesse de déplacement est nulle. Cette gamme d'impédance ne peut pas être restituée intégralement via une interface haptique à cause des contraintes de stabilité. Les interfaces haptiques peuvent donc être caractérisés par l'impédance qu'elles sont capables de restituer. Cette caractéristique est appelée *Z-width*. Le *Z-width* peut donc être pris comme mesure de qualité d'une interface haptique [CB94], de nombreux travaux portent également sur l'augmentation de cette grandeur.

### 1.2.1.2 Système à retour d'effort

Les interfaces à retour d'effort se présentent en général sous la forme de mécanismes robotisés séries ou parallèles. Ils sont constitués d'un ensemble d'actionneurs (souvent des moteurs pas à pas) et de capteurs de position et/ou d'effort. Ces interfaces peuvent être encore différenciées en fonction des architectures retenues, série ou parallèle. Nous décrirons brièvement dans les deux paragraphes suivants, les interfaces à retour d'effort couramment utilisées en réalité virtuelle.

#### 1.2.1.2.a Interfaces séries

Les interfaces séries se présentent sous la forme d'actionneurs montés en série sur un robot ressemblant généralement à un bras. Cette famille est la plus représentée. La gamme PHANTOM® de Sensable ou Virtuouse6D d'Haption rentrent dans cette catégorie. Tous les degrés de liberté ne sont pas forcément actifs. Ainsi, sur l'interface PHANTOM OMNI® les degrés de libertés en rotation ne sont pas actifs, le rendu haptique ne peut se faire qu'en translation.

Il existe aussi des interfaces se présentant sous la forme d'exosquelette. Ces interfaces se basent sur un mécanisme venant envelopper l'utilisateur afin d'appliquer des efforts sur les différentes parties du corps humain (bras, jambes, mains, etc.). Ce type d'interface haptique, initialement développé pour la rééducation fonctionnelle est utilisé dans les environnements virtuels pour pouvoir effectuer des préhensions. En effet, à la différence des interfaces précédemment présentées qui utilisent souvent une poignée, ce type d'interfaces peut agir sur les doigts indépendamment. Le CyberGrasp™ d'Immersion ou le Rutgers Master II [BBPB02] sont des exemples de tels exosquelettes.

#### 1.2.1.2.b Interfaces parallèles

Parmi les interfaces parallèles, il existe de nombreuses interfaces de bureau basée sur trois mécanismes en parallèle (Figure 1.4c) comme l'HapticMaster [Iwa94], l'Omega ou le Delta de Force Dimension, le Falcon de Novint ou le Virtuouse™ 6D Desktop d'Haption .

Il existe également des interfaces à câbles utilisant plusieurs câbles reliant chacun un même objet manipulé par un utilisateur (manette, poignée, ...) à un moteur. Le SPIDAR et SPIDAR-8 du Tokyo Institute of Technology sont deux exemples de ces interfaces à



Figure 1.4 – Exemples d'interface haptique

câbles [IS94].

Des descriptions plus complètes des interfaces haptiques existantes sont disponibles dans [HACH<sup>+</sup>04] et [BKHAK04] spécifiquement pour les interfaces tactiles.

## 1.2.2 Stabilité du retour d'effort

L'asservissement de systèmes robotisés présenté dans la partie précédente à un univers virtuel pose un problème de stabilité qui peut être étudié principalement de deux manières : en utilisant la théorie de la passivité ou en découplant l'interface haptique de l'environnement virtuel à l'aide d'un couplage virtuel.

Dans cette section nous explicitons les principaux résultats en matière de stabilité pour les univers virtuels à retour d'effort. À la suite de ces parties, les méthodes permettant d'étendre l'impédance transmise ( $Z$ -width) ou perçue sont présentées. Nous introduisons également, dans cette section, les notations que nous utiliserons dans la suite du manuscrit.

### 1.2.2.1 Théorie de la passivité

La théorie de la passivité [Pop73, DV75] est un outil puissant permettant de démontrer la stabilité d'un système. La passivité s'énonce de manière simple : un système passif est stable. Il suffit alors de démontrer qu'un système ne génère pas d'énergie pour prouver sa stabilité. Cependant un système rendu passif sera moins performant (temps de réponse, suivi, ...) qu'un système stable mais non passif.

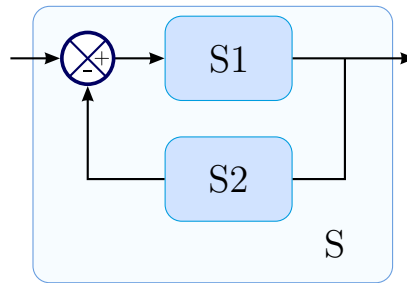
Dans d'un système échangeant des positions et des vitesses avec un élément extérieur, soit  $\mathbf{f}$  et  $\dot{\mathbf{x}}$  les efforts et les vitesses à l'interface du système considéré,  $E_s$  est l'énergie emmagasinée dans le système,  $P_{\text{diss}}$  est la puissance dissipée par le système. Le système est considéré

comme passif si l'énergie extraite de ce système au cours du temps ne dépasse pas l'énergie initialement stockée dans le système :

$$\forall t \geq 0, \int_0^t \mathbf{f}(\tau) \dot{\mathbf{x}}(\tau) d\tau = E_s(t) - E_s(0) + \int_0^t P_{\text{diss}}(\tau) d\tau \geq -E_s(0) \quad (1.9)$$

Un système à un port est passif si l'énergie extraite de ce système ne dépasse pas l'énergie initiale du système.

Dans le cas de systèmes à plusieurs entrées et sorties, un des résultats immédiats de ce théorème, est qu'un ensemble de systèmes passifs est passif et donc stable, quelque soit l'agencement des blocs (parallèle ou série) [DV75]. Si les blocs  $S1$  et  $S2$  de la Figure 1.5 sont passifs alors le système  $S$  est passif. Ce résultat s'étend immédiatement à un système à  $n$  blocs. Ainsi l'analyse d'un système peut se simplifier en le découpant en sous-systèmes. Prouver la passivité de ces sous-systèmes suffit à prouver la passivité du système global. Cette théorie s'applique à des systèmes linéaires ou non-linéaires ainsi qu'à des systèmes discrets ou continus.



**Figure 1.5** – Théorème de la passivité appliqué à un système possédant une entrée et une sortie. Si les systèmes  $S1$  et  $S2$  sont passifs alors le système  $S$  est passif.

Dans le cas d'un environnement virtuel lié à une interface haptique, les lois physiques simulées sont généralement conservatives et donc passives. Cependant, les approximations utilisées ont tendance à générer de l'énergie. Celle-ci sont dues à, notamment, la discrétisation du problème traité, les convertisseurs numériques-analogiques au sein des interfaces haptiques, l'intégration numérique. Le système n'est donc plus garanti d'être passif. Les parties suivantes présentent l'application de la théorie de la passivité à un système à retour d'effort. La première partie décrit l'analyse d'un système à retour d'effort. La seconde partie présente le principe du contrôleur de passivité permettant de garantir la passivité du système.

### 1.2.2.1.a Rendu direct

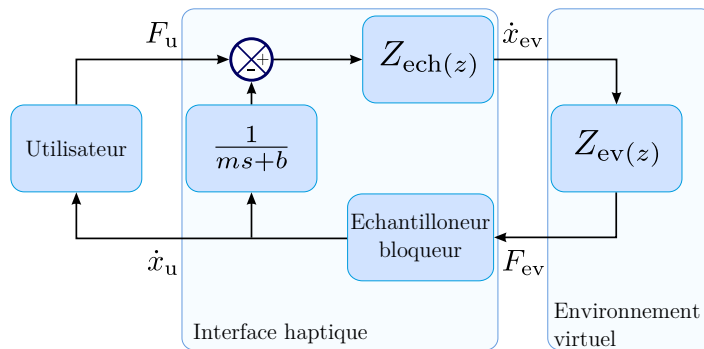
La méthode la plus directe pour lier une interface haptique à un univers virtuel est d'appliquer les informations issues de la simulation physique (effort ou position de l'objet virtuel manipulé) directement à l'interface haptique. Les informations d'état de l'interface haptique étant fournies directement au moteur physique, elles permettent de mettre à jour l'état de l'objet virtuel. L'architecture d'un environnement virtuel avec retour d'effort peut être séparée en trois éléments interconnectés, l'utilisateur, l'interface haptique et l'environnement virtuel (voir 1.6).

Les premières études sur l'analyse de la passivité de ce système ont été réalisées pour des systèmes à un degré de liberté. Bien que simplifiée, cette analyse établit des résultats valables pour des systèmes plus complexes. Dans le cas à un degré de liberté, le modèle retenu par Colgate *et al.* [CGSS93] pour établir un critère d'analyse de la passivité est constitué



**Figure 1.6** – Architecture d'un environnement avec rendu haptique en admittance

d'un mur virtuel sous la forme d'une contrainte unilatérale et d'une interface haptique modélisée par une masse  $m$  amortie par une viscosité  $b$ . Le mur virtuel est modélisé par une fonction de transfert discrète  $Z_{ev}(z)$ . La conversion du domaine discret au domaine continu est effectuée par des convertisseurs numérique-analogique et analogique-numérique, modélisés respectivement par un échantillonneur  $Z_{ech}(z)$ , de période d'échantillonnage  $\Delta T$  et par un échantillonneur bloqueur d'ordre zéro :



**Figure 1.7** – Schéma bloc de l'architecture à un degré de liberté (Colgate *et al.* [CGSS93])

$F_u, F_{ev}, \dot{x}_u, \dot{x}_{ev}$  étant respectivement les efforts de l'utilisateur et de l'environnement sur l'interface et les vitesses échangées entre l'utilisateur et l'interface et entre l'environnement virtuel et l'interface. Au contact, le mur virtuel est exprimé par un ressort-amortisseur (méthode par pénalité Section 1.1.2.3.a) de raideur  $k_m$  et de viscosité  $b_m$ . Un ressort-amortisseur est normalement passif, puisque qu'il est constitué d'un ressort qui est conservatif et d'un amortisseur qui est dissipatif. Cependant dans le domaine discret, ces éléments peuvent devenir actifs. Le ressort n'évolue plus de manière continue, il relâche de l'énergie à chaque pas de temps par à-coup (Equation 1.10). Si l'amortisseur ne dissipe pas cette énergie, le mur n'est plus passif. Le mur et l'énergie du ressort peuvent s'exprimer :

$$Z_{ev}(z) = k_m + b_m \frac{z-1}{\Delta T z} \quad (1.10)$$

$$E_{ressort} = \frac{1}{2} k_m (v \Delta T)^2, v = x_{k+1} - x_k$$

Dans ce cas, Colgate et Schenkel [CS94] établissent le critère de passivité suivant :

$$b > \frac{k_m \Delta T}{2} + |b_m| \quad (1.11)$$

Afin de rendre passif un tel système,  $k_m$ ,  $\Delta T$  ou  $|b_m|$  peuvent être diminués ou  $b$  augmenté.  $\Delta T$  est généralement fixé par le temps nécessaire à la simulation physique. Diminuer  $k_m$  ou  $b_m$  tend à pénaliser de manière plus importante la loi de contact et donc à diminuer l'impédance transmise du mur virtuel. La viscosité physique  $b$  de l'interface haptique joue donc un rôle très important. Augmenter  $b$  permet d'augmenter l'impédance que peut transmettre l'interface

haptique. Cependant augmenter  $b$  revient à diminuer la transparence de l'interface haptique, cette viscosité physique sera ressentie par l'utilisateur. De plus, rendre la viscosité virtuelle  $b_m$  nulle permet d'avoir une raideur virtuelle  $k_m$  maximum.

De manière similaire à ces travaux, Abbott et Okamura [AO05] ont étudié les effets de quantification dus à la résolution  $\delta$  des capteurs de position. Ces capteurs fournissent une information échantillonnée. Ainsi, de façon équivalente au problème de la discrétisation temporelle, la discrétisation spatiale entraîne une création d'énergie par le ressort virtuel. Le modèle d'interface haptique que les auteurs ont retenu est plus complexe que celui présenté ci-dessus, car il inclut les efforts de frottement  $f_f$  intrinsèques à l'interface haptique. Ces forces s'opposant au mouvement (voir le paragraphe 1.1.2.2 sur la loi de Coulomb) sont dissipatives.

Dans le cas où l'amortissement  $b_m$  du mur virtuel est nul, le critère suivant est obtenu :

$$k_m \leq \min\left(\frac{2b}{\Delta T}, \frac{2f_f}{\delta}\right) \quad (1.12)$$

Diolati *et al.* [DNBS06] ont encore étendu ce principe en introduisant des grandeurs adimensionnelles afin de comparer facilement les différents critères permettant de rendre ce système passif. Cette approche permet aussi de prendre en compte les retards additionnels dus aux temps de calculs des simulations. Le cas des non-linéarités de l'environnement virtuel ont été intégré par Miller *et al.* [MCF00] en introduisant une mesure  $\alpha$  de la non-passivité du système. Une viscosité physique  $b$  de l'interface haptique suffisamment grande peut alors dissiper l'énergie générée par les non-linéarités de l'environnement virtuel. Le retard entre le moteur physique et l'interface haptique peut aussi être intégré dans cette formulation [GSH<sup>+</sup>07]. Plus le retard  $T_r$  croît plus les gains du mur virtuel doivent être faibles afin de conserver un système passif :

$$k_m < \frac{b_m + b}{\frac{\Delta T}{2} + T_r} \quad (1.13)$$

Le rendu haptique direct a été largement utilisé pour de nombreuses applications. La configuration de l'objet étant directement fournie par la simulation physique, la mise en œuvre du rendu haptique est très simple, même pour des objets rigides à six de degrés de libertés. Ainsi, Gregory *et al.* [GME<sup>+</sup>00] ont utilisé du rendu haptique direct avec une interface haptique à six degrés de liberté et une méthode de résolution de contact par pénalités. Johnson et Willemsem [JW03] ont également eu recours au rendu direct avec l'algorithme LMD (Section 1.1.1.1.b).

### 1.2.2.1.b Observateur et contrôleur de passivité

Une autre manière d'assurer la stabilité d'un environnement virtuel avec rendu haptique, en utilisant le théorème de la passivité, est d'ajouter un contrôleur qui garantisse en permanence la passivité du système. Contrairement au rendu haptique direct où l'énergie produite par un élément du système doit être absorbée par d'autres éléments du système, telle que la viscosité de l'interface haptique, le contrôleur de passivité a pour unique tâche de dissiper l'énergie émise par le système. Le contrôleur modifie l'énergie du système de telle manière que le système reste passif. Cette méthode est également appelée *energy shaping*. L'énergie échangée par le système doit donc être mesurée. A cette fin, un observateur de passivité est associé au contrôleur de passivité. Gillespie et Cutkosky [GC96] ont ainsi présenté un contrôleur capable de compenser les fuites d'énergies introduites par le bloqueur d'ordre zéro (voir Figure 1.7) ainsi que, dans le cas un mur virtuel, le problème de passage de l'état libre à l'état contraint. En effet, la position de l'interface haptique peut dépasser la position du

mur virtuel entre deux pas de temps. La transition entre le mode libre et contraint n'est pas synchronisée entre le l'interface haptique qui évolue de manière continue et l'environnement virtuel qui évolue de manière discret. C'est écart temporel est générateur d'énergie.

Le terme observateur de passivité (OP) et contrôleur de passivité (CP) a été introduit par Hannaford et Ryu [HR02]. Les auteurs ont décrit un algorithme permettant d'implémenter cette méthode pour un système à un port. Dans le cas de la Figure 1.8 l'algorithme est décrit par l'algorithme 1.

---

**algorithme 1** Calculer  $f_{CP}(k)$

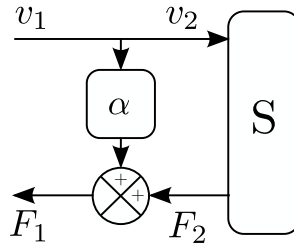
---

```

1: tantque CP actif faire
2:    $v_2(k) \leftarrow v_1(k)$ 
3:    $E_{\text{obs}}(k) \leftarrow E_{\text{obs}}(k-1) + f_2(k)v_2(k)T + \alpha(k-1)v_2(k-1)^2T$ 
4:   si  $E_{\text{obs}}(k) < 0$  alors
5:      $\alpha(k) \leftarrow -\frac{E_{\text{obs}}(k)}{v_2(k)T}$ 
6:   sinon
7:      $\alpha(k) \leftarrow 0$ 
8:   fin
9:    $f_1(k) \leftarrow f_2(k) + \alpha(k)v_2(k)$ 
10:   $k \leftarrow k + 1$ 
11: fin tantque

```

---



**Figure 1.8** – Système à un port avec contrôleur de passivité en parallèle. Dans ce cas, l'impédance du système est ajustée en dissipant l'énergie supplémentaire à l'aide de la viscosité  $\alpha$ .

Un algorithme similaire peut être établi pour le cas d'un contrôle en admittance. Le contrôleur peut également être disposé en série plutôt que parallèle.

Ryu *et al.* [RKH04a] ont ensuite étendu cette méthode au cas où la vitesse  $v_c(k)$  (dans le cas de la commande en impédance) n'est pas constante pendant le pas de temps  $k$ , l'énergie observée devient alors :

$$E_{\text{obs}}(k) = \left[ \sum_{j=1}^k f_2(t_{j-1}) (x_2(t_j) - x_2(t_{j-1})) \right] + f_2(t_k)(x_2(t_k) - x_2(t_{k-1})) \quad (1.14)$$

$x_2$  correspond à la consigne en position associée à la vitesse  $v_2$ , le terme entre crochet est l'énergie effectivement échangée avec l'environnement virtuel. Le terme suivant correspond à une estimation de l'énergie entrante au pas de temps suivant. La méthode OP/CP peut ainsi s'interfacer entre un système continu et un système discret. Stramigioli *et al.* [SSvdSF05] ont formulé ce principe à l'aide du formalisme des ports hamiltoniens afin d'interfacer des systèmes discrets et continus. Borghesan *et al.* [BMM07] ont utilisés le même formalisme pour appliquer la méthode OP/CP à des systèmes multi-fréquentiels.



Enfin, toujours dans le cas de systèmes discret et continu couplés, Ryu *et al.* [RPHH05] ont proposé de dissiper l'énergie de manière plus souple, en utilisant un modèle de l'environnement discret. Cette méthode évite une dissipation trop brusque de l'énergie. L'énergie observée du système devient donc :

$$E_{\text{obs}}(k) = \left[ \sum_{j=0}^k f_2(t_{j-1}) (x_2(t_j) - x_2(t_{j-1})) \right] - S(k) - D(k) \quad (1.15)$$

où  $S$  est l'énergie absorbée par le système et  $D$  est l'énergie dissipée par le système. La condition est plus restrictive et la correction du contrôleur de passivité intervient plus tôt. Dans le cas où l'environnement virtuel est non-linéaire,  $S$  et  $D$  peuvent être numériquement trop coûteux à calculer. Dans ce cas, l'intégration numérique de la puissance échangée avec l'environnement virtuel peut servir de modèle. Les premières formulations des systèmes OP/CP posaient problème lorsque l'utilisateur travaillait dans une région très dissipative. Dans l'algorithme 1, le terme  $E_{\text{obs}}$  peut devenir très grand lorsque l'utilisateur évolue longtemps dans une zone très dissipative. Si l'utilisateur se déplace vers une zone active,  $E_{\text{obs}}$  peut mettre plusieurs pas de temps avant de devenir nulle, et le système devient ainsi instable pendant cet intervalle de temps. En utilisant un modèle de l'environnement virtuel, le CP s'activera même si l'énergie échangée avec le système ( $E_{\text{obs}}(k) + S(k) + D(k)$  dans l'Equation 1.15) ne devient jamais nulle.

Néanmoins, Gosline et Hayward [GH07] ont remarqué que la dissipation via un amortisseur virtuel possède des limitations. En effet, le contrôleur de passivité utilise une estimation de la vitesse fournie par l'interface haptique. Une amplification importante, lors d'une forte dissipation, amplifie également le bruit inhérent à l'estimation de la vitesse. Des oscillations sont susceptibles d'apparaître en cours de manipulation. De plus, sans correcteur de passivité, un contact peut entraîner un régime permanent de petites oscillations. Dans le cas de vitesses faibles, un correcteur de passivité peut ne pas réussir à corriger ces oscillations. En utilisant un amortissement physique réglable à base d'un frein à courant de Foucault au lieu d'un amortissement virtuel, les auteurs ont proposé un contrôleur de passivité permettant effectivement de stabiliser une interface haptique liée à un environnement virtuel.

### 1.2.2.2 Couplage virtuel

Lorsque que l'environnement virtuel se complexifie, il devient difficile de démontrer la stabilité de la méthode de rendu haptique direct. Notamment dans le cas de tâches d'insertion où le nombre de contraintes unilatérales est élevé. Colgate *et al.* [CSB95] ont pour cela introduit la notion de couplage virtuel. Le couplage virtuel est un élément rajouté entre l'interface haptique et la simulation physique (Figure 1.9). Ainsi l'outil virtuel manipulé dans l'univers virtuel est relié à l'interface haptique par ce couplage virtuel.

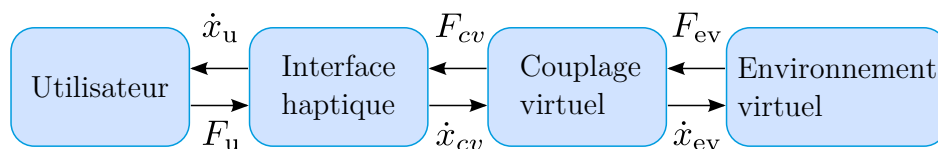
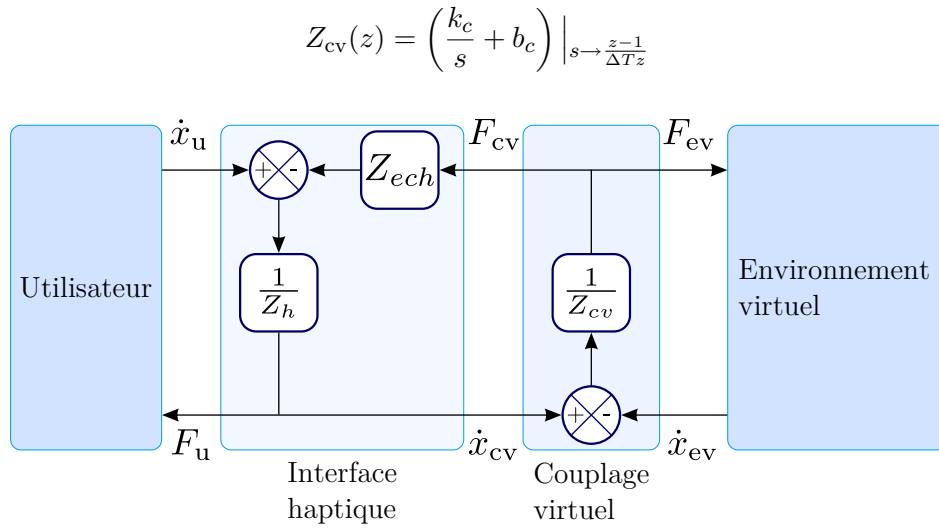


Figure 1.9 – Schéma bloc de rendu haptique avec un couplage virtuel en impédance.

Le couplage virtuel retenu est généralement un contrôleur proportionnel dérivé (ou ressort-amortisseur) dans le cas d'un pilotage en impédance (voir Figure 1.10) :



**Figure 1.10** – Schéma bloc de rendu haptique avec un couplage virtuel.

Le bloc  $Z_{ech}$  correspond au convertisseur numérique analogique en entrée de l'interface haptique. L'analyse de stabilité permet ainsi de séparer les problèmes de stabilité de la simulation physique (abordés dans la Section 1.1) de ceux de l'interface haptique. En pratique, ce couplage virtuel adapte l'impédance de l'environnement afin de respecter la contrainte de stabilité de l'interface haptique.

En se basant sur l'analyse de systèmes linéaire non dépendant du temps (*LTI : Linear Time-Invariant*), Adams et Hannaford [AMH99] ont généralisé l'utilisation de couplages virtuels quelque soit la méthode de commande de l'interface haptiques (impédance, admittance). Le système est *inconditionnellement stable* ce qui est moins restrictif que la condition de passivité abordée précédemment. Afin de pouvoir utiliser cette méthode, l'utilisateur doit être supposé passif, ce qui n'est en réalité pas le cas. Néanmoins, l'être humain à un comportement favorable pour la passivité, c'est à dire qu'il peut aider à rendre un système passif. Considérer l'être humain comme passif est donc une approximation raisonnable.

Adams et Hannaford [AMH99] ont posé le couplage virtuel comme un système à deux ports pour lequel existe une matrice d'immittance ( $I(s)$ ) permettant de décrire la relation entre les entrées et les sorties du couplage virtuel :

$$I(s) = \begin{pmatrix} i_{11}(s) & i_{12}(s) \\ i_{21}(s) & i_{22}(s) \end{pmatrix} \quad (1.16)$$

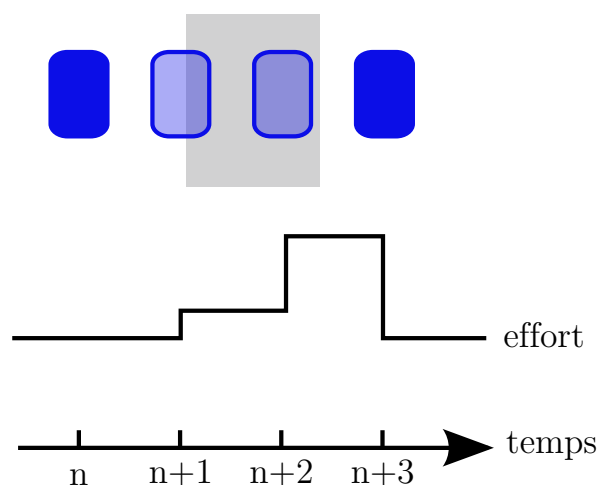
Une matrice d'immittance permet de considérer indifféremment les matrices correspondant à un asservissement en admittance, impédance ou hybride au sein d'un même formalisme. Les différents types d'interface et d'asservissements peuvent donc être analysés d'une même manière. Pour un système linéaire indépendant du temps, le critère de Llewellyn [Lle52] permet de garantir qu'un système à 2 ports est inconditionnellement stable s'il est relié à deux éléments passifs, quelque soit son mode de couplage :

$$\forall \omega \in \mathbb{R}^+ \left\{ \begin{array}{l} \Re(i_{11}), \Re(i_{22}) \in \mathbb{R}^+ \\ -\frac{\Re(i_{12}i_{21})}{|i_{12}i_{21}|} + 2\frac{\Re(i_{11})\Re(i_{22})}{|i_{12}i_{21}|} \geq 1 \end{array} \right. \quad (1.17)$$

Miller *et al.* [MCF00] ont étendu ces travaux et ont établi un critère en fonction de la viscosité de l'interface haptique et de la raideur du couplage virtuel, d'une manière similaire

à celle utilisée pour le rendu direct.

Zilles et Salisbury [ZS95] ont également introduit le principe de couplage virtuel de manière parallèle à Colgate *et al.*, sous le nom de *god-object* afin de résoudre des problèmes posés par le rendu direct, problèmes distincts de la stabilité. En effet, en utilisant un rendu direct associé à une méthode par pénalités (Section 1.1.2.3.a) et en exerçant un effort suffisamment important, des objets fins peuvent être traversés. En exerçant un effort important, la position de l'interface haptique se situe à l'intérieur de l'objet virtuel qui est entrain d'être touché. L'utilisateur voit donc l'objet qu'il manipule situé à l'intérieur de l'objet qu'il touche. Si cet objet est fin, la position de l'interface haptique peut se situer de l'autre coté de l'objet touché. L'objet manipulé se situe donc de l'autre coté de cet objet, donnant l'impression que l'objet vient d'être traversé (voir Figure 1.11).



**Figure 1.11** – Effet passe-muraille avec un rendu direct. Lorsque l'objet à atteint la position de droite, l'utilisateur à l'impression d'avoir traversé l'objet gris.

Le god-object est équivalent au principe du couplage virtuel lorsque l'objet manipulé est un point, de ce fait, les simulations sont limitées à trois degrés de libertés. En mouvement libre, la position du god-object reflète celle de l'interface haptique. Lorsque que la position de l'interface haptique est détectée à l'intérieur d'un objet de la scène virtuelle, le god-object est contraint<sup>3</sup> à la surface de celui-ci. En retour, un effort est appliqué à l'interface haptique en fonction de la distance entre le god-object et la position de l'interface haptique.

Le rendu direct pose en outre des problèmes de discontinués des efforts rendus. Le problème de la discontinuité de l'effort se manifeste lors du passage de l'interaction d'une face à une autre d'un polyèdre. Les normales aux surfaces n'étant pas continues, l'effort calculé n'est pas continu. En se basant sur les techniques d'ombrages développés pour l'informatique graphique, Ruspini *et al.* [RKK97] ont proposé de *lisser* l'effort calculé par l'intermédiaire du god-object. D'autre part, les effets tels que les frottements peuvent être intégrés au niveau du god-object au lieu d'être traités au niveau du modèle de contact. Cette dernière méthode peut s'avérer avantageuse en termes de temps de calcul.

Cette méthode a été étendue par McNeely *et al.* [MPT99] pour les interfaces à six degrés de liberté en utilisant des ressort-amortisseurs exprimés en trois dimensions avec une composante linéaire et une composante rotationnelle. Ortega *et al.* [ORC06] ont utilisé également cette méthode pour réaliser le rendu haptique avec des objets rigides complexes en contraignant

3. Zilles et Salisbury utilisent des multiplicateurs de Lagrange, une formulation équivalente, considérant la contrainte unilatérale peut être résolue avec un LCP (Section 1.1.2.3.b).

l'accélération du god-object au lieu de sa vitesse comme proposé par Zilles et Salisbury. Cette méthode permet d'éviter l'apparition d'artefacts haptiques introduit par la méthode initiale du god-object en six degrés de liberté.

### 1.2.2.3 Rendu haptique événementiel

Les différentes méthodes permettant d'interfacer de manière stable un environnement virtuel et une interface haptique, ont tous comme point commun de limiter d'une manière ou d'une autre l'impédance de l'environnement virtuel, afin de garantir la stabilité de du système. L'impédance perçue par les utilisateurs d'un tel système peut donc être diminuée.

Une des manières d'augmenter l'impédance transmise ou d'élargir le Z-width est de modifier l'interface haptique, en modifiant sa viscosité  $b$  par exemple (voir Section 1.2.2.1.a). Cette viscosité peut être rajoutée au niveau électrique de l'interface haptique sous la forme d'une résistance et d'une capacité en parallèle de l'actionneur électrique comme proposé par Melhing *et al.* [MCP05].

Sans modifier le système, il est possible d'augmenter la raideur perçue de l'environnement virtuel par l'utilisateur en appliquant des illusions psychophysiques, en particulier au moment du contact. Dans le cadre d'un rendu haptique direct, Salcudean *et al.* [SV97] ont utilisé une force de freinage afin d'immobiliser l'interface haptique le plus rapidement possible lors d'un contact. Idéalement, cette force est appliquée pendant un pas de temps. Cette force est alors une *impulsion de freinage*. Elle correspond à une viscosité qui est transitoire et ne compromet donc pas la stabilité du système.

D'une manière similaire, c'est à dire en ajoutant à la force calculée une seconde force, Okamura *et al.* [ODH98] ont suggéré d'améliorer la perception d'un objet rigide au moment du contact. Cette seconde force est calculée à partir d'une sinusoïde décroissante (modèle empirique proposé par Wellman et Howe [WH95]).

$$f_h = f_{ve} + A(\dot{x}_h)e^{-Bt} \sin(\omega t) \quad (1.18)$$

Les différents paramètres de cette force (amplitude  $A$ , atténuation  $B$  et fréquence  $w$ ) sont fonction du matériau caractérisant l'objet virtuel. Pour les déterminer, des mesures ont été réalisées avec différents matériaux. Un utilisateur a tapé de manière répétée un objet rigide à l'aide d'un stylet. Les profils de l'accélération ont permis d'identifier les paramètres précédents pour différents matériaux. L'atténuation varie de  $300 \text{ s}^{-1}$  pour du bois à  $1300 \text{ s}^{-1}$  pour de l'acier. En outre, ce phénomène est transitoire, il est donc similaire à l'approche de Salcudean *et al.* . Cette méthode est appelée haptique événementielle.

Cette approche peut entraîner des instabilités dans le système, car de l'énergie est introduite dans la boucle d'asservissement. Hwang *et al.* [HWN04] ont introduit le principe d'haptique en boucle ouverte. L'information de l'utilisateur vers l'environnement virtuel a une fréquence plutôt faible (quelques Hz). La partie du signal introduite par l'Equation 1.18 peut donc être filtrée, l'énergie ne boucle plus dans le système asservi. Le système demeure ainsi stable. Cette méthode peut donc s'appliquer à des interfaces à retour d'effort, tant que leur bande passante n'est pas trop atténuée dans les hautes fréquences.

Cette information haute fréquence peut être calculée à chaque pas de temps à partir d'un modèle prédéterminé empiriquement [OHDC00]. Hwang *et al.* [HWN04] ont utilisé des trames pré-calculées à partir de contacts réels. Kuchenbercker *et al.* [KFN06] ont introduit plusieurs modèles en fonction de la vitesse de l'outil ou de la partie de l'objet touché.

### 1.3 Interactions haptiques distantes

Pour réaliser une tâche collaborative distante avec du retour d'effort, les différentes interfaces haptiques doivent être reliées entre elles afin de permettre aux utilisateurs d'agir sur le même univers virtuel. Les informations échangées entre ces interfaces haptiques subissent un retard dû au temps de transmission des données. Ce temps de transmission dépend du réseau utilisé par la plateforme collaborative pour simuler l'univers virtuel. Dans le meilleur des cas, les informations se déplacent à la vitesse de la lumière entre chacune des interfaces haptiques. Ainsi, lors d'une transmission transatlantique les informations subiraient idéalement un retard de 20 ms. En pratique, les informations sont traitées par un ensemble d'éléments électroniques ou opto-électroniques qui augmentent ce retard et peut le rendre variable. C'est ce qui se passe pour un réseau informatique ATM ou Ethernet. Dans ce cas, le retard lors d'une connexion transatlantique est variable et est légèrement supérieur à 100 ms [KKT<sup>+</sup>04].

Comme nous l'avons vu dans la section précédente, la stabilité d'un système à retour d'effort est primordiale. Or, le retard sur la transmission d'information est un facteur bien connu d'instabilité des systèmes en boucle fermée.

L'étude de la stabilité de système à retour d'effort soumis à des retards de transmissions est au cœur du domaine de la téléopération. Un système de téléopération classique est constitué d'un robot maître (une interface haptique) et d'un robot esclave (voir Figure 1.12), chacun des deux étant associé à un contrôleur. Une analogie peut être réalisée avec un univers virtuel avec retour d'effort (Figure 1.6) où le robot maître correspond à l'interface à retour d'effort et le robot esclave et son environnement correspondent à l'univers virtuel. D'ailleurs, de nombreuses méthodes décrites dans la Section 1.2 sont également appliquées voir empruntées du domaine de la téléopération comme les couplages virtuels ou la méthode OP/CP.

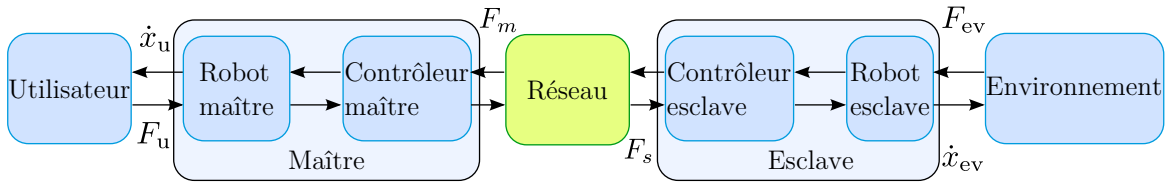


Figure 1.12 – Schéma bloc d'asservissement en téléopération

Un tel système est décrit par le système d'équation suivant :

$$\mathbf{M}_m \ddot{\mathbf{x}}_m + \mathbf{B}_m \dot{\mathbf{x}}_m = \mathbf{f}_m + \mathbf{f}_u \mathbf{M}_s \ddot{\mathbf{x}}_s + \mathbf{B}_s \dot{\mathbf{x}}_s = \mathbf{f}_s - \mathbf{f}_{ev} \quad (1.19)$$

où  $\mathbf{x}_m$  et  $\mathbf{x}_s$  sont les coordonnées généralisées de l'interface maître et de l'interface esclave.  $\mathbf{M}_*$  est une matrice d'inertie et  $\mathbf{B}_*$  une matrice d'amortissement.  $\mathbf{f}_u$  et  $\mathbf{f}_{ev}$  sont les forces exercées par l'utilisateur sur le maître et par l'environnement sur l'esclave. Les efforts  $\mathbf{f}_s$  et  $\mathbf{f}_m$  permettent d'asservir le maître et l'esclave.

Les sections suivantes présentent les différentes méthodes permettant de stabiliser un système de téléopération soumis à un retard fixe ou variable. De la même manière que pour le rendu haptique précédemment abordé, la stabilité d'un système de téléopération peut être analysée globalement en formulant le système sous la forme d'une matrice d'immittance et en utilisant des contrôleurs proportionnel dérivé comme ceux rencontrés dans les couplages virtuels. L'autre approche consiste à utiliser le théorème de la passivité afin de rendre passif les différents éléments constitutifs du système de téléopération. Les contrôleurs de passivité et la transformation en variables d'onde font parti de cette dernière approche.

Des éléments plus spécifiques concernant des cas particuliers (téléopération spatiale, etc.) peuvent être trouvés dans [She93, AM01, HS05]

### 1.3.1 Couplage virtuel

L'approche utilisant des couplages virtuels [CSB95] consiste à utiliser un asservissement proportionnel dérivé comme contrôleur. Cette approche présentée précédemment dans le cadre des environnements virtuels peut être aussi utilisé dans le domaine de la téléopération.

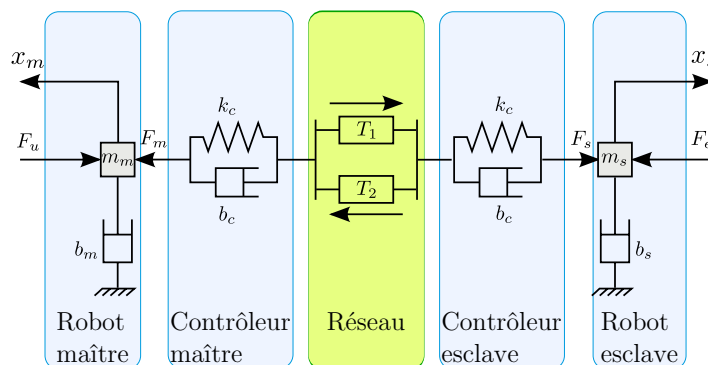
La représentation sous forme de matrice d'immittance du système de téléopération est à la base de plusieurs approches théoriques de l'analyse de stabilité d'un système téléopéré. Par exemple, dans le cas des couplages virtuels précédemment abordé (voir Section 1.2), le critère de Llewellyn permet d'établir une condition de stabilité.

Cette approche permet, pour un retard fixe, d'établir des gains de stabilité pour un système couplé simple via des contrôleurs proportionnel-dérivé après avoir exprimé le système sous la forme d'une matrice d'immittance. Une telle analyse a été réalisée par Hashtrudi-Zaad et Salcudean [HZS01] dans le cas d'une architecture *4 channel* [Law92, YY94]. Ce type d'architecture consiste à échanger les efforts et les vitesses des deux contrôleurs au lieu des efforts de l'un et la vitesse de l'autre, notamment, lors d'une commande en impédance. Un critère sur les gains des différents asservissements proportionnel dérivé est alors obtenu en fonction du retard introduit par le médium de communication.

En utilisant ce type de système appliqué aux environnements virtuels, Cheong *et al.* [CNAS05] ont proposé de diminuer l'écart de suivi entre le maître et l'esclave en ajoutant un prédicteur de Smith au niveau du contrôleur.

Ce type de système peut être également stabilisé de manière plus restrictive en utilisant la théorie de la passivité. Lee et Spong [LS06] ont proposé de dissiper l'énergie introduite par l'ajout d'un amortisseur dans chacun des contrôleurs (maître et esclave) en se basant sur un schéma de téléopération asservi en position (voir Figure 1.13). Les efforts de couplages sont alors :

$$\begin{aligned} \mathbf{f}_m(t) &= \mathbf{B}_c [\dot{\mathbf{x}}_s(t - T_1) - \dot{\mathbf{x}}_m(t)] + \mathbf{K}_c [\mathbf{x}_s(t - T_1) - \mathbf{x}_m(t)] - \mathbf{B}_{\text{diss}} \dot{\mathbf{x}}_m(t) \\ \mathbf{f}_s(t) &= \mathbf{B}_c [\dot{\mathbf{x}}_m(t - T_2) - \dot{\mathbf{x}}_s(t)] + \mathbf{K}_c [\mathbf{x}_m(t - T_2) - \mathbf{x}_s(t)] - \mathbf{B}_{\text{diss}} \dot{\mathbf{x}}_s(t) \end{aligned} \quad (1.20)$$



**Figure 1.13** – Représentation mécanique d'un asservissement position-position pour un cas de téléopération

Le système est passif si :

$$\mathbf{B}_{\text{diss}} - \left( \frac{\sin(\omega \frac{T_1+T_2}{2})}{\omega} \right)^2 \mathbf{K}_c \mathbf{B}_{\text{diss}}^{-1} \mathbf{K}_c \text{ est semi-défini positive, } \forall \omega \in \mathbb{R} \quad (1.21)$$

Une solution possible, avec  $T_{t_{\max}} \geq T_t$  une borne supérieure du délai est :

$$b_{\text{diss}} = \frac{T_{t_{\max}}}{2} k \quad (1.22)$$

Cette relation est très similaire à la condition de stabilité proposée par Colgate et Schenkel [CS94] dans le cas d'un environnement virtuel avec rendu haptique direct (Section 1.2.2.1.a). Les schémas de téléopération utilisant des couplages virtuels ne sont pas robustes dans le cas où le retard est variable. Une manière d'adapter ces méthodes au cas de retard variable est d'ajouter un tampon permettant d'augmenter le retard afin de compenser la part variable du retard.

### 1.3.2 Observateur et contrôleur de passivité

La méthode OP/CP décrite dans la Section 1.2.2.1.b peut être également appliquée au cas de la téléopération.

Le robot maître et le robot esclave sont des éléments continus pouvant être modélisés par des éléments dissipatifs (voir Equation 1.19), De la même manière que pour les environnements virtuels précédemment abordés, l'environnement et l'utilisateur peuvent être considéré comme passif. Les éléments non passifs sont alors le réseau, où le retard introduit rend ce bloc actif, et les deux contrôleurs qui sont des éléments discrets et potentiellement non passif. Le bloc formé par les deux contrôleurs maître et esclave ainsi que le réseau est donc le seul élément non passif du système considéré.

Les observateurs et contrôleurs de passivité peuvent être utilisés pour des systèmes à  $n$ -ports et donc en particulier appliqués au bloc formé par les deux contrôleurs maître et esclave ainsi que le réseau. Le principe est de dissiper l'énergie générée par le couplage bilatéral afin de garantir la condition de passivité.

Ryu *et al.* [RKH04b] ont ainsi étendu le principe de OP/CP au cas de la téléopération sans retard en utilisant deux observateurs et deux contrôleurs (Figure 1.14) :

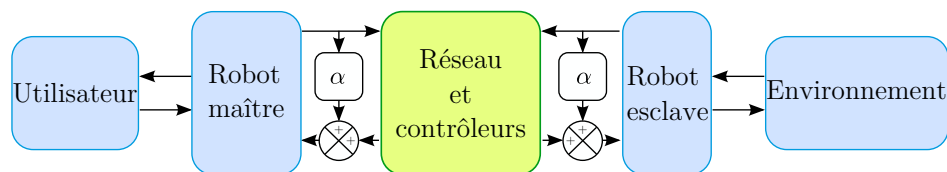


Figure 1.14 – Extension de schéma OP/CP au cas d'un système à deux ports

L'algorithme 1 est légèrement modifié, il doit prendre en compte le calcul de deux énergies à chacun des ports. Le port qui génère de l'énergie doit la dissiper. Dans le cas où les deux ports généreraient de l'énergie, le choix est laissé de dissiper à l'un ou l'autre des ports ou d'utiliser un algorithme plus complexe.

Dans le cas de la téléopération avec retard, Artigas *et al.* [AVPH06] ont eu recours à deux OP/PC de manière découplée. Il n'y a plus, dans ce cas, un algorithme central qui doit mesurer à tout instant l'énergie échangée à chaque port. Dès lors, cet algorithme peut être utilisé aisément en présence de retard.

La méthode utilisant les OP/CP peut être appliquée pour un retard fixe comme pour un retard constant. Néanmoins, appliquer cette méthode à un système fortement instable se traduira par un système globalement très visqueux, il est donc préférable d'appliquer cette méthode à un système faiblement instable ou présentant seulement des zones d'instabilités limitées. De plus, comme remarqué dans la Section 1.2.2.1.b, l'utilisation de forte dissipation pose des problèmes car la vitesse de l'interface haptique est estimée et donc bruitée [GH07].

### 1.3.3 Variables d'onde

Les *variables d'onde* sont directement issues de la théorie de la passivité. Le principe est de rendre passif l'élément de communication soumis à des retards. Ces variables d'ondes sont en fait l'application au domaine mécanique d'éléments issus de l'analyse des circuits électriques, en particulier de la théorie de la répartition (*scattering theory*).

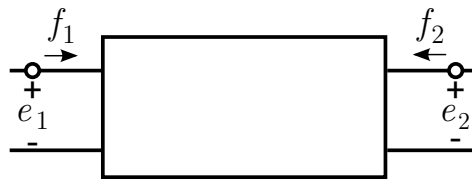


Figure 1.15 – Quadripôle

Le bloc réseau de la Figure 1.12 peut être formalisé sous la forme d'un quadripôle (Figure 1.15),  $e_1$ ,  $e_2$  sont des efforts et  $f_1$  et  $f_2$  des flux au sens des graphes de liaisons (*bond graph*), le produit des deux étant une puissance. Ainsi  $e$  et  $f$  sont respectivement la tension et l'intensité dans le domaine électrique et l'effort et la vitesse dans le domaine mécanique. Les efforts et les flux sont reliés par la relation d'impédance suivante :

$$\mathbf{f} = \begin{pmatrix} f_1(s) \\ f_2(s) \end{pmatrix} = Z(s) \begin{pmatrix} e_1(s) \\ e_2(s) \end{pmatrix} = Z(s)\mathbf{e} \quad (1.23)$$

Dans le domaine électrique, les *ondes de puissance* (*power waves*) sont une construction mathématique introduite par Penfield Jr. [Pen60] pour étudier le bruit dans un circuit électrique. Kurokawa [Kur65] remarqua que l'introduction de ces ondes de puissance permet de représenter la puissance incidente et réfléchiées lors de l'échange d'énergie avec un système. Le système défini par l'Equation 1.23 peut être alors réécrit en utilisant des ondes de puissances et la matrice de répartition  $S$  :

$$\mathbf{u}(s) = S(s)\mathbf{v}(s) \quad (1.24)$$

$$\mathbf{u}(s) = \frac{\mathbf{e}(s) + z(s)\mathbf{f}(s)}{2\sqrt{\|\Re(z(s))\|}} \quad (1.25)$$

$$\mathbf{v}(s) = \frac{\mathbf{e}(s) - \overline{z(s)}\mathbf{f}(s)}{2\sqrt{\|\Re(z(s))\|}} \quad (1.26)$$

$S$  est l'opérateur de répartition,  $\mathbf{u}$  représente l'onde réfléchiée et  $\mathbf{v}$  l'onde incidente,  $z(s)$  est une impédance donnée.

La puissance est exprimée par substitution dans Equation 1.26 :



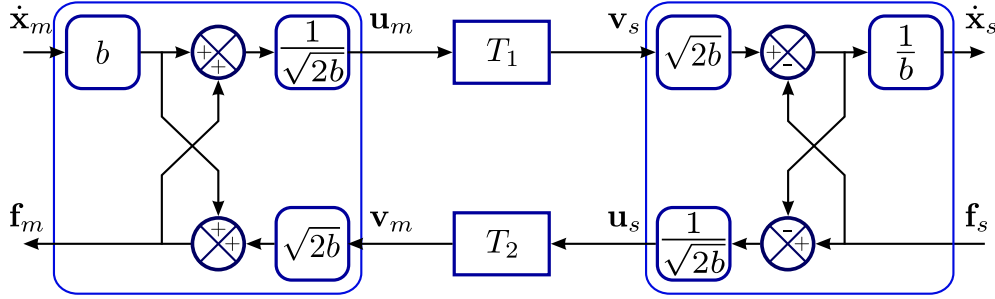


Figure 1.16 – Transformation en variable d'ondes

$$P = \Re(\mathbf{e}(s)\mathbf{f}(s)) \quad (1.27)$$

$$= \Re \left( \frac{(\mathbf{u}(s) - \mathbf{v}(s))(\overline{z(s)\mathbf{u}(s)} - z(s)\mathbf{v}(s))}{\|\Re(z(s))\|} \right) \quad (1.28)$$

$$= \|\mathbf{u}(s)\|^2 - \|\mathbf{v}(s)\|^2 \quad (1.29)$$

$\|\mathbf{u}\|^2$  est l'énergie incidente.  $\|\mathbf{v}\|^2$  est l'énergie réfléchi. La condition de passivité simplifiée (Equation 1.9 sans le terme d'énergie stockée) se traduit, à partir de Equation 1.26, par :

$$\begin{aligned} \int_0^t P(\tau) d\tau &\geq 0 \\ \Leftrightarrow \int_0^t \|\mathbf{u}\|^2 d\tau &\geq \int_0^t \|\mathbf{v}\|^2 d\tau \\ \Leftrightarrow \forall \omega, \|S(j\omega)\| &\leq 1 \end{aligned} \quad (1.30)$$

Cette condition de stabilité se traduit par le fait que le système ne peut renvoyer plus d'énergie dans l'onde réfléchi qu'il n'en a reçu, vérifiant ainsi la condition de passivité (Equation 1.9).

Si le retard est fixe et que la ligne de communication est sans perte, Hannaford et Spong [AS89] ont été les premiers à utiliser l'opérateur de répartition dans le domaine mécanique. Ils ont posé  $z(s) = 1$  dans leur approche, soit la transformation  $\mathbf{e} - \mathbf{f}$  et  $e + \mathbf{f}$ . Par la suite, Niemeyer et Slotine [NS91] ont introduit les variables d'ondes comme étant des ondes de puissance ayant pour impédance caractéristique  $b$  constante. Cette impédance est équivalente à une viscosité (Figure 1.16). La formulation classique des variables d'onde en mécanique est la suivante :

$$\mathbf{u}_m = \frac{b\dot{\mathbf{x}}_m + \mathbf{f}_m}{\sqrt{2b}} \quad (1.31)$$

$$\mathbf{v}_m = \frac{b\dot{\mathbf{x}}_m - \mathbf{f}_m}{\sqrt{2b}} \quad (1.32)$$

où  $\dot{\mathbf{x}}_m$  et  $\mathbf{f}_m$  sont définis dans l'Equation 1.19. La vitesse  $\dot{\mathbf{x}}_m$  correspond à un flux dans les graphes de liaisons et la force  $\mathbf{f}_m$  correspond à un effort.

Dans le cas d'un retard  $T_r$ , la condition de passivité se vérifie :

$$\mathbf{v}_s(t) = \mathbf{u}_m(t - T_1) \quad (1.33)$$

$$\mathbf{v}_m(t) = \mathbf{u}_s(t - T_2) \quad (1.34)$$

$$\int_0^t \frac{1}{2} \mathbf{v}\mathbf{v}^T d\tau \leq \int_0^t \frac{1}{2} \mathbf{u}\mathbf{u}^T d\tau + E_{\text{stocké}}(0) \quad (1.35)$$

$$\Leftrightarrow \int_0^{t-T_r} \frac{1}{2} \mathbf{u}\mathbf{u}^T d\tau \leq \frac{1}{2} \int_0^t \mathbf{u}\mathbf{u}^T d\tau + E_{\text{stocké}}(0) \quad (1.36)$$

$$\Leftrightarrow 0 \leq \int_{t-T_r}^t \frac{1}{2} \mathbf{u}\mathbf{u}^T d\tau \quad (1.37)$$

La condition est donc vérifiée. Transmettre les variables d'ondes sur une ligne de communication sans perte soumise à un retard constant est donc bien passif. Ce résultat peut s'exprimer directement à partir de la matrice de répartition pour des retards constants différents dans les deux directions  $T_1$  et  $T_2$  :

$$S_{T_r}(s) = \begin{pmatrix} 0 & e^{-sT_1} \\ e^{-sT_2} & 0 \end{pmatrix} \quad (1.38)$$

$$\Rightarrow \forall \omega, \|S_{T_r}(j\omega)\| = 1$$

A partir de la définition discrète de l'énergie, Kosuge et Murayama ont formulé les variables dans le domaine discret par une transformation bilinéaire de  $S(s)$  en  $S(z)$ . Pour un retard  $n_1 = \frac{T_1}{\Delta T}$  et  $n_2 = \frac{T_2}{\Delta T}$  avec  $\Delta T$  le pas de temps, la formulation est très similaire au cas continu :

$$S_{\text{retard}}(z) = \begin{pmatrix} 0 & z^{-n_1} \\ z^{-n_2} & 0 \end{pmatrix}$$

$$\mathbf{u}(z) = \frac{b\dot{\mathbf{x}}(z) + \mathbf{f}(z)}{\sqrt{2b}} \quad (1.39)$$

$$\mathbf{v}(z) = \frac{b\dot{\mathbf{x}}(z) - \mathbf{f}(z)}{\sqrt{2b}} \quad (1.40)$$

Dans l'Equation 1.32, l'impédance est la même pour toutes composantes des variables d'onde. Munir et Book [MB02] proposent une formulation équivalente permettant d'utiliser une impédance différente pour chacune des composantes, si elles sont indépendantes. Ce cas est intéressant lors de l'asservissement dans le domaine articulaire d'un robot esclave sur un robot maître. Soit  $\dot{\mathbf{q}}$  les vitesses articulaires,  $\mathbf{\Gamma}$ , les efforts articulaires.  $\dot{\mathbf{q}}, \mathbf{\Gamma}, \mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ , les variables d'onde s'expriment simplement par :

$$\exists A, B \in \mathbb{R}^{n \times n}, I = 2AB \quad (1.41)$$

$$\mathbf{u} = A\dot{\mathbf{q}} + B\mathbf{\Gamma} \quad (1.42)$$

$$\mathbf{v} = A\dot{\mathbf{q}} - B\mathbf{\Gamma} \quad (1.43)$$

Dans le cas où l'asservissement est réalisé dans le domaine cartésien plutôt que le domaine articulaire, Stramigioli *et al.* [SAJvdSM02] ont démontré que la décomposition en variables d'onde existe. Ils l'ont appelée *geometric scattering*. Leur formulation est plus compacte que celle présentée plus haut.

Soit deux espaces vectoriels duaux  $\mathcal{V}$  et  $\mathcal{V}^*$  et  $\mathcal{D}$ , tels que

$$\mathcal{D} = \mathcal{V} \times \mathcal{V}^* \quad (1.44)$$

Pour une métrique donnée  $Z$  de  $\mathcal{V}$ , il existe une décomposition de  $\mathcal{D}$  en deux sous espaces propres complémentaires  $\mathcal{S}_Z^+$  et  $\mathcal{S}_Z^-$  :

$$\mathcal{D} = \mathcal{S}_Z^+ \oplus \mathcal{S}_Z^- \quad (1.45)$$

Soit  $\mathbf{e}, \mathbf{f} \in \mathcal{D}$ , respectivement un effort et un flux et  $\mathbf{s}_Z^+ \in \mathcal{S}_Z^+, \mathbf{s}_Z^- \in \mathcal{S}_Z^-$ , l'énergie se définit par :

$$\langle \mathbf{e}, \mathbf{f} \rangle = \frac{1}{2} \|\mathbf{s}_Z^+\|_+^2 - \frac{1}{2} \|\mathbf{s}_Z^-\|_-^2 \quad (1.46)$$

où  $\|\cdot\|_+$  et  $\|\cdot\|_-$  sont des normes définies par le produit hermitien sur  $\mathcal{S}_Z^+$  et  $\mathcal{S}_Z^-$  respectivement.

$Z$  est une matrice semi-définie positive, elle possède donc une racine carré symétrique  $P$ , les variables d'ondes s'expriment donc par :

$$\begin{aligned} \mathbf{s}_Z^+ &= \frac{P^{-1}}{\sqrt{2}} (\mathbf{e} + Z\mathbf{f}) \\ \mathbf{s}_Z^- &= \frac{P^{-1}}{\sqrt{2}} (\mathbf{e} - Z\mathbf{f}) \\ Z &= P^2 \end{aligned} \quad (1.47)$$

Dans le domaine cartésien les forces  $W$  et les vitesses  $T$  appartiennent à deux algèbres de Lie complémentaire  $\mathfrak{se}(3)$  et  $\mathfrak{se}(3)^*$ .

Les différentes définitions des variables d'onde proposés introduisent un terme d'amortissement,  $b$  pour la formulation de Niemeyer et  $P$  pour celles de Stramigioli *et al.* . Le choix de ce terme permet de changer le comportement du système. De plus, la transformation en variables d'onde ne fait pas intervenir la position. En pratique la position est récupérée en intégrant la vitesse, mais des dérives peuvent apparaître.

### 1.3.3.1 Influence de l'impédance

La valeur de l'impédance caractéristique  $b$  n'influe pas sur la stabilité du système mais modifie le comportement d'un système utilisant une transformation en variables d'onde en donnant une importance plus grande à la vitesse ou à l'effort dans la variable d'onde. Afin de mettre en lumière ce comportement, la transformation en variables d'onde peut être exprimée sous la forme d'un ressort équivalent de raideur  $k_b$  à l'équilibre et d'une inertie de masse  $m_b$  en mouvement libre [NS97b] :

$$k_b = \frac{b}{T_r} \quad (1.48)$$

$$m_b = bT_r \quad (1.49)$$

Ainsi une grande valeur de  $b$  se traduit par une inertie et une raideur perçues importantes et donc des mouvements de plus faible amplitude. A l'inverse une valeur de  $b$  faible se traduit par des efforts faibles et donc un ressenti des efforts distants moins importants.

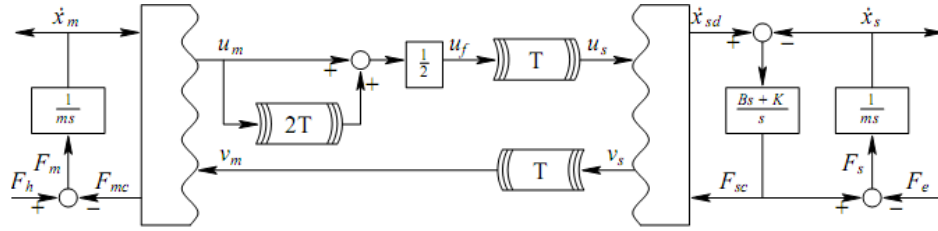


Figure 1.17 – Transformation en variable d'onde avec filtre à horizon fini (issu de [TN05])

La figure 1.16 montre que pour réaliser une transformation en variable d'onde, l'onde émise dépend de l'onde incidente. Si l'impédance caractéristique  $b$  est différente de l'impédance du système qui se situe après la transformation en variable d'onde (contrôleur esclave ou maître), une partie de l'onde incidente est réfléchiée et repart via l'onde émise. Ce mécanisme est similaire aux ondes électromagnétiques lors d'un changement de milieu. Cette réflexion d'onde peut, dans le cas de la téléopération, s'effectuer au niveau des deux transformations. Cette réflexion pouvant se produire de chaque côté de la transformation, l'onde réfléchiée peut entrer dans une boucle. Ce phénomène va se traduire par des oscillations au niveau du robot maître et du robot esclave. Il existe deux méthodes pour parer à ce problème.

Niemeyer et Slotine [NS97a] ont adapté l'impédance du contrôleur du maître et de l'esclave à l'impédance de la transformation en variables d'onde. Cette adaptation a pour effet d'annuler les réflexions en mouvement libre. Pour un contact rigide, l'adaptation d'impédance crée un filtre passe bas dépendant de  $b$ . Un réglage approprié de  $b$  permet donc de limiter les réflexions d'onde dans ce dernier cas. Cependant, il n'existe pas de réglage permettant de traiter les deux cas.

La seconde approche consiste à filtrer ces oscillations au niveau des ondes  $\mathbf{u}$  et  $\mathbf{v}$  [TN05]. La réflexion d'onde a une période qui est égale au retard total  $T_r$ , ainsi pour de faibles retards un filtre passe-bas peut être utilisé. En effet, l'interaction haptique n'est pas symétrique. La commande et donc l'information émise par le robot maître vers le robot esclave ne contient pas d'information utile dans les hautes fréquences, car un utilisateur ne pourra pas appliquer une consigne ayant une fréquence supérieure à quelques Hz. Si le retard est plus important, Tanner et Niemeyer [TN05] ont introduit deux filtres - filtre à horizon fini et filtre à horizon infini - permettant de filtrer les variables d'ondes. Ces filtres permettent d'éliminer la fréquence de résonance de la réflexion d'onde ainsi que ses harmoniques :

$$G_{\text{fini}}(s) = \frac{1 + e^{-2sT_r}}{2} \quad G_{\text{infini}}(s) = \frac{1}{2 - e^{-2sT_r}} \quad (1.50)$$

où  $T_r = T_1 + T_2$  est le retard total.

### 1.3.3.2 Suivi en position

La transformation en variables d'onde ne comporte pas explicitement d'information en position puisque d'après les équations 1.31 et 1.32 les variables d'onde dépendent de la vitesse et de la force. L'écart de position  $\Delta x$  entre le maître et l'esclave vaut :

$$\Delta x(t) = x_m(t) - x_s(t) \quad (1.51)$$

$$= \frac{1}{\sqrt{2b}} \int_{t-T_r}^t u_s(\tau) - v_s(\tau) d\tau \quad (1.52)$$

Au repos, l'écart de position est théoriquement nul. En mouvement, cet écart dépend de l'impédance  $b$ , c'est un critère supplémentaire à ajouter aux remarques précédentes (adaptation d'impédance, etc.) pour choisir  $b$ .

En pratique, la position dérive au cours du temps et l'erreur de position n'est plus nulle au cours d'une manipulation pour plusieurs raisons. L'utilisation des variables d'ondes ne se fait pas de manière continue, les contrôleurs sont des éléments numériques (ordinateurs, etc.), l'intégration est remplacée par une somme finie. De plus, des erreurs d'approximation peuvent apparaître lors de l'intégration numérique, bien que faible pour une manipulation longue avec des efforts ou des vitesses importantes, les erreurs numériques s'accumulent. Si le maître et l'esclave ne sont pas bien synchronisés, des données peuvent être perdues et augmenter d'autant l'écart de position. Enfin, un écart de position initial ne peut pas être corrigé.

Niemeyer et Slotine [NS97b] ont suggéré de transmettre en parallèle des variables d'onde, ce qu'ils appellent l'intégrale des variables d'onde. Ces dernières sont définies par :

$$\mathbf{U}(t) = \int_0^t \mathbf{u}(\tau) d\tau = \frac{b\mathbf{x}(t) + \mathbf{p}(t)}{\sqrt{2b}} \quad (1.53)$$

$$\mathbf{p} = \int_0^t \mathbf{f}(\tau) d\tau \quad (1.54)$$

La définition de  $\mathbf{V}(t)$  est similaire. La position de l'esclave peut être explicitement déduite de l'Equation 1.54 :

$$\mathbf{x}_s(t) = \mathbf{x}_m(t - T_1) - \mathbf{x}_s(t - T_r) - \frac{1}{b} \int_{t-T_r}^t \mathbf{f}_s(\tau) d\tau \quad (1.55)$$

L'intégration de la force se fait entre  $t$  et  $t - T_r$ . Au repos, la force étant nulle, les positions de l'esclave et du maître convergent. Cependant cette formulation est valable si les différentes composantes de  $\mathbf{x}$  et  $\mathbf{p}$  sont indépendantes, la formulation des variables d'onde proposées par Stramigioli *et al.* [SAJvdSM02] dans l'espace cartésien n'admet pas directement une telle décomposition en intégrale de variables d'onde.

Deux autres méthodes permettant d'assurer le suivi en position existe également.

Niemeyer et Slotine [NS97a] ont suggéré une alternative aux intégrales de variables d'onde. Il s'agit d'observer l'écart de position et de corriger cet écart tant que la condition de passivité est respectée, l'écart peut donc ne pas être totalement corrigé si la condition de passivité ne peut être respectée.

Chopra *et al.* [CSOB06] ont proposé d'utiliser les variables d'onde pour échanger les informations de vitesse et d'effort et assurer le suivi en position à l'aide d'un correcteur proportionnel entre les deux contrôleurs (un ressort de raideur  $k$ ). Pour garantir la stabilité du système,  $k$  doit satisfaire la condition suivante :  $k^2 T_r^2 < b_m b_s$  où  $b_m, b_s$  sont respectivement l'amortissement du contrôleur maître et celui du contrôleur esclave.

### 1.3.3.3 Retard variable

Les transformations en variables d'onde présentée précédemment, ne garantissent la stabilité que dans le cas d'un retard constant. Or, un retard variable déforme les variables d'onde (voir Figure 1.18). Le suivi en position n'est plus alors garanti, ni la passivité du système. Niemeyer et Slotine [NS98] ont proposé d'utiliser les intégrales de variables d'ondes pour assurer le suivi en position, comme dans le cas des retards fixes et l'énergie  $\int_0^t u(\tau) d\tau$  portée

par la variable d'onde afin de reconstruire des variables d'ondes en conservant la passivité. Le filtre de reconstruction doit respecter :

$$\int_0^t \mathbf{u}_{\text{filtre}}(\tau)^2 d\tau \leq \int_0^{t-T_r(t)} \mathbf{u}(\tau)^2 d\tau \quad (1.56)$$

où  $\mathbf{u}_{\text{filtre}}$  est l'onde reconstruite et  $\mathbf{u}$  l'onde reçue par le filtre de reconstruction.

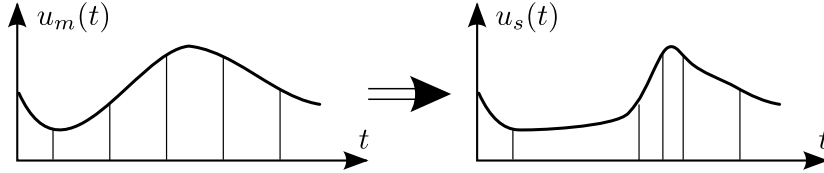


Figure 1.18 – Variable d'onde déformée

L'inégalité préserve la condition de passivité en garantissant que le filtre ne génère pas d'énergie. Mirfakhrai et Payandeh [MP02] ont ajouté un prédicteur à la transformation en variable d'onde afin d'améliorer les performances de cet asservissement.

Chopra *et al.* [CSHB03] proposent d'utiliser des gains sur les variables d'onde  $\mathbf{u}$  et  $\mathbf{v}$  en fonction du retard afin de respecter la condition de passivité. Le suivi en position est assuré par un correcteur proportionnel comme expliqué Chopra *et al.* [CSOB06]. La transformation en variable d'ondes doit satisfaire :

$$\mathbf{u}_s(t) = \mathbf{f}_s(T_1(t))\mathbf{u}_m(t - T_1(t)) \quad (1.57)$$

$$\mathbf{v}_m(t) = \mathbf{f}_m(T_2(t))\mathbf{v}_s(t - T_2(t)) \quad (1.58)$$

$$\mathbf{f}_s(T_1(t))^2 \leq 1 - \frac{dT_1}{dt} \quad (1.59)$$

$$\mathbf{f}_m(T_2(t))^2 \leq 1 - \frac{dT_2}{dt} \quad (1.60)$$

$$\frac{dT_r}{dt} \leq 1 \quad (1.61)$$

L'Equation 1.61 traduit le fait que le retard croît au pire aussi vite que le temps, dans le cas d'une perte de donnée.

La méthode OP/CP peut être associée à la transformation en variable d'onde pour rendre passif un système soumis à un retard variable. Yokohohki *et al.* [YIY00, YTY02] proposent de corriger la variable d'onde reconstruite lors d'un retard variable en rajoutant une boucle de contrôle sur la variable d'onde reçue. Une variable d'onde non déformée est alors reconstruite :

$$\mathbf{u}'_s(t) = \tilde{\mathbf{u}}_s(t) + \mathbf{K} \left( \int_0^t \mathbf{u}'_s(\tau) d\tau - \int_0^t \mathbf{u}_s(\tau) d\tau \right) \quad (1.62)$$

$$\tilde{\mathbf{u}}_s(t) = \mathbf{u}_m(t - T_r(t)) \quad (1.63)$$

$$\int_0^t \tilde{\mathbf{u}}_s(\tau) d\tau = \int_0^{t-T_r(t)} \mathbf{u}_m(\tau) d\tau \quad (1.64)$$

où  $\mathbf{u}_s(t)$  est la variable d'onde corrigée du côté esclave.  $\tilde{\mathbf{u}}_s$  est la variable d'onde déformée et  $\mathbf{u}'_s$  la variable d'onde restaurée à partir des estampilles. Cependant cette solution viole la condition de passivité, les auteurs proposent de rajouter un contrôleur de passivité qui renvoie une variable d'onde nulle ( $\mathbf{u}_s(t) = 0$ ) dans le cas où la condition de passivité est violée.

Lors de l'utilisation d'un réseau sans qualité de service, les données émises peuvent ne pas être reçue par la partie distante. Dans ce cas, Hirche et Buss [HB04] ont montré que l'utilisation d'un algorithme passif permettant de reconstruire les données perdues permettait de conserver la passivité du système. On peut noter que Hirche *et al.* [HHSB05] ont utilisé ce type de méthode utilisant la reconstruction de données afin de réduire (ou *compresser*) la quantité de donnée émise sur le réseau. Secchi et Fantuzzi [SF07] ont étendu ces résultats en utilisant le formalisme port hamiltonien dans le cas d'asservissements exprimé dans le domaine cartésien.

## 1.4 Collaboration et coopération distantes

Le retard introduit dans une manipulation distante a donc une très forte influence sur la stabilité d'un système à retour d'effort. Ces retards ont également une influence sur la perception qu'ont les utilisateurs d'un événement. Dans le cas d'une plateforme de collaboration, le choix de la topologie de l'application peut également modifier le comportement du système en modifiant le lieu où se situent les retards. Certaines architectures peuvent mener à l'existence de plusieurs simulations physiques. Les méthodes précédemment décrites peuvent être utilisées afin de lier entre elles les simulations physiques. Des couplages virtuels peuvent être ainsi mis en œuvre dans un cas sans retard [MABA02] ou des transformations en variables d'onde en présence de retard [SNF05].

Dans cette partie nous présentons tout d'abord les différentes architectures utilisées pour permettre une interaction distante avec retour d'effort puis l'influence de l'haptique sur des tâches collaboratives en enfin l'influence du retard sur ces tâches collaboratives avec retour d'effort.

En premier lieu, il convient de définir les termes *collaboration* et *coopération* qui sont souvent rencontrés lorsque plusieurs personnes réalisent une même tâche ou participent à une même tâche. Ainsi d'après la définition qu'en donne Broll [Bro95] pour un univers virtuel :

- la **coopération** est l'édition jointe d'un objet ou élément partagé.
- la **collaboration** est l'édition jointe et **simultanée** d'un objet ou élément partagé.

La contrainte de simultanéité implique que l'environnement virtuel soit capable de gérer plusieurs accès à une ressource partagée tout en garantissant la cohérence de l'univers virtuel. A contrario permettre la coopération est plus simple, des mécanismes d'exclusions mutuelles associés à des stratégies de manipulations tour par tour permettent parfaitement de réaliser la coopération.

### 1.4.1 Architecture réseau des plateformes collaboratives

Les différents sites entre lesquels se déroulent la collaboration doivent s'échanger des informations au sujet de la manipulation afin de permettre de réaliser des tâches collaboratives. Les différents ordinateurs reliés entre eux s'échangent deux types d'information.

- Des **Messages** sont des données qui ne sont pas échangées de manière régulière. Ils sont généralement associés à un événement comme un impact par exemple.
- Les **flux** sont des données qui sont échangées de manière régulière (à chaque pas de simulation par exemple), comme la position d'un objet virtuel. Malgré la nature discrète de l'échange de données, le terme « continu » peuvent être appliqué à ce genre de données.

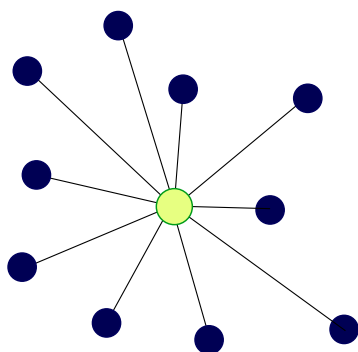
Ces données sont échangées par un réseau informatique qui peut être un réseau local comme pour les expérimentations précédentes jusqu'à l'Internet ou des réseaux satellitaires. Ces médiums de communication introduisent des retards, la vitesse de transmissions de données est soumise à des limites physiques et d'autre part par les éléments (répéteur, commutateur, routeur, etc.) constitutifs des ces réseaux. Dans le cas d'un réseau local, les retards induits par les réseaux sont très faibles (inférieur à la milliseconde). Alors que pour le cas d'une liaison transatlantique utilisant l'Internet, ce retard peut avoisiner 120 ms dans des conditions correctes [GJF<sup>+</sup>07]. Dans le cas de retard faible, l'architecture réseau retenue pour une plateforme collaborative a peu d'effet. Dans le cas de retard plus importants, suivant l'architecture retenue, les retards ne se situent pas entre les mêmes éléments de la plateforme collaborative ce qui provoque des comportements différents de l'univers virtuel.

La section suivante aborde les architectures utilisées dans les applications utilisant du retour d'effort. Ces univers virtuels sont généralement de petite taille et donc un certains nombres de problèmes liés à l'architecture rencontrés dans les grand environnements virtuels (tel NSPNET [MZP<sup>+</sup>95], MASSIVE [GB95], DIVE [CH93] ou plus récemment ceux issus de jeux de rôle en ligne massivement multi-joueurs *MMORPG*) ne sont pas abordés. Des précisions sur ces éléments peuvent être trouvées dans [ZS99]

Les principales plateformes permettant de réaliser des tâches collaboratives ou coopératives sont basées sur deux architectures principales, l'architecture client-serveur et l'architecture distribuée.

#### 1.4.1.1 Architecture client-serveur

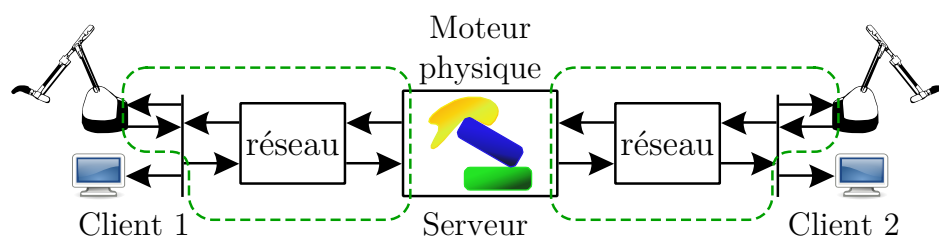
Historiquement la première utilisée en informatique, l'architecture client-serveur est constituée d'un serveur auquel sont connectés des clients (voir Figure 1.19). Cette architecture est aussi appelée étoilée ou centralisée. Le serveur héberge un moteur physique qui simule l'environnement virtuel. Les clients proposent aux utilisateurs une vue de cet univers virtuel. Les utilisateurs agissent sur le client qui transmet ces commandes au serveur. Ce dernier fait évoluer l'univers virtuel et renvoie aux clients le nouvel état de l'univers virtuel. L'information est donc centralisée sur le serveur, chaque client accède exactement aux mêmes informations, l'univers virtuel est unique et donc intrinsèquement cohérent.



**Figure 1.19** – Architecture client-serveur. Les clients sont connectés à un serveur central qui est chargé de simuler l'univers virtuel. Les clients transmettent au serveur les consignes des utilisateurs et proposent à ces derniers une vue de l'univers virtuel, un retour d'effort, etc.

Pour cette architecture les retards dus au réseau se situent entre chaque client et le serveur. Les consignes de l'utilisateur transmises du client au serveur sont retardées ainsi que les informations envoyées depuis le serveur aux clients. Dans le cas d'un environnement virtuel il y a un retard équivalent à un aller-retour client-serveur entre le moment où un utilisateur





**Figure 1.20** – Architecture clients-serveur avec retour d'effort. Les blocs pointillés peuvent générer des instabilités à cause du retard dû au réseau. Les méthodes issues de la téléopération doivent être utilisées pour corriger ces instabilités.

commence un mouvement et le moment où il observe ce mouvement via son affichage graphique. Ainsi, chaque utilisateur observe son action ainsi que les actions des autres personnes retardées. Dans le cas du retour d'effort, ce cas s'assimile directement à de la téléopération où le client ferait office de maître et le serveur d'esclave. Cette architecture est celle proposée par Buttolo *et al.* [BOH97] pour réaliser des tâches collaborative entre des sites distants.

La plateforme *Spring* [MBB<sup>+</sup>02] dédiée à la simulation chirurgicale est ainsi basée sur une architecture client-serveur. En utilisant ce type d'architecture, Gunn *et al.* [GHS<sup>+</sup>05] ont proposé une plateforme dédiée à l'entraînement de gestes chirurgicaux où un des intervenants travaille directement sur le serveur. Pour cette personne le retard introduit est quasi inexistant. Tang *et al.* [TCO06, OCT04] ont utilisés une architecture centralisée pour créer une simulation de canoë basée sur les variables d'ondes. Bien qu'un des utilisateurs agissaient directement sur le serveur, un retard était introduit afin que le retard soit similaire pour chacun des intervenants contrairement à Gunn *et al.* [GHS<sup>+</sup>05].

La Figure 1.20 présente une architecture clients-serveur avec un retour d'effort.

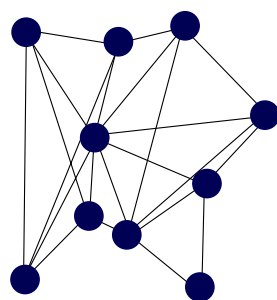
L'architecture client-serveur introduit un retard entre l'action d'un utilisateur sur le client et la réaction calculée par le moteur physique sur le serveur qui doit être renvoyée au client. En utilisant plusieurs moteurs physiques, l'architecture distribuée permet de résoudre ce problème.

#### 1.4.1.2 Architecture distribuée

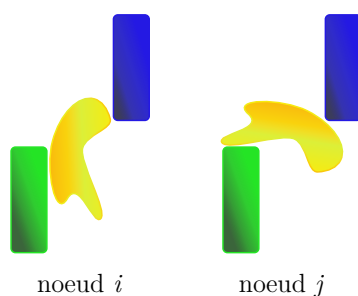
Les réseaux distribués reposent sur plusieurs ordinateurs ayant un rôle équivalent à la différence de l'architecture client-serveur. Chacun des nœud héberge un simulateur qui simule l'environnement virtuel. Ainsi les éléments de l'environnement virtuel sont répliqués à chaque nœud.

Les retards de communications se situent, dans ce cas, au niveau des informations échangées entre les nœuds. Dans de telles architectures distribuées, les différents nœuds s'échangent l'état de leur simulation afin que chacun simule le même univers virtuel. Si les états des différents nœuds sont échangés instantanément et continûment chacun des nœuds simulerait exactement le même univers virtuel. Cependant, l'information est retardées à cause du médium de communication, les nœuds peuvent donc fournir une vue différente de l'univers virtuel. Les différences entre les différents nœuds peuvent conduire à des incohérences, c'est à dire que deux états de l'univers virtuel à deux nœuds distincts ne puissent plus converger vers un état unique (voir Figure 1.22).

Le principal avantage de cette architecture est d'avoir un système plus réactif. Chaque nœud possédant une simulation, il n'y a pas de retard entre les commandes issues des interfaces haptiques et les retours visuels, sonore ou haptique autre que le temps de traitement



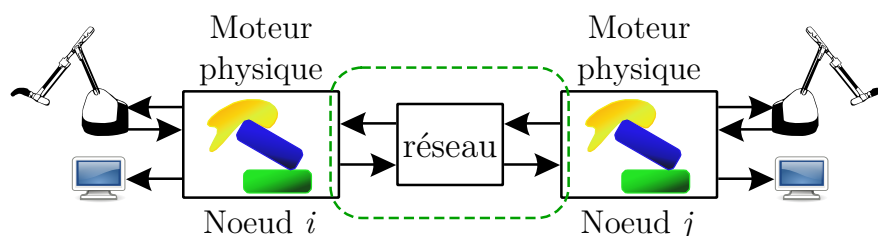
**Figure 1.21** – Architecture distribuée. Chaque nœud a un rôle équivalent, les différents nœuds sont directement reliés les uns aux autres.



**Figure 1.22** – Cas d'incohérence entre deux nœuds.

du moteur physique. Ainsi un nœud est équivalent à un environnement virtuel avec retour haptique abordé dans la section 1.2.

La Figure 1.23 présente une architecture distribuée avec un retour d'effort.



**Figure 1.23** – Architecture distribuée avec retour d'effort. Le bloc pointillé peut générer des instabilités à cause du retard dû au réseau. Les méthodes issues de la téléopération doivent être utilisées pour corriger ces instabilités.

Footohi *et al.* [FSC07] ont comparé les architectures centralisées et distribuées sur un réseau local (retard faible, 2 à 3 ms maximum). Dans le cas où la fréquence d'échantillonnage de l'interface haptique est supérieure à celle de la simulation physique, l'architecture client-serveur permet d'obtenir des domaines de stabilité plus importants.

En utilisant cette architecture, Kim *et al.* [KKT<sup>+</sup>04] ont réalisés entre le MIT et L'University College London une plateforme permettant de manipuler un cube à deux personnes avec un rendu haptique direct. La stabilité du système a été assurée en ajoutant des viscosités relativement importantes sur les différents éléments de la plateforme (entre les interfaces haptiques et les simulations physiques, au sein des simulations physiques) afin d'avoir un système très dissipatif et donc peu sujet aux instabilités.

L'apparition d'incohérences entre les états des différents nœuds peut être prévenue de deux manières. Premièrement en verrouillant l'espace où se déroule la manipulation afin qu'un seul nœud puisse agir dans cette zone verrouiller. La seconde méthode consiste à corriger l'état des nœuds au fur et à mesure de la manipulation.

#### 1.4.1.2.a Contrôle de la cohérence par verrous

Afin de limiter les incohérences, les interactions des utilisateurs peuvent être limitées. Takemura et Kishino [TK92] ont proposés une architecture utilisant des verrous gérés par un unique serveur. Dans le cas où les objets ne sont pas rigides des verrous régionalisés sur une partie de l'objet peuvent être employés, Bogsanyi et Miller [BM02] ont proposé une architecture distribuée pour une application de sculpture virtuelle utilisant ce principe. Ces plateformes ne permettent plus d'interactions collaboratives mais uniquement coopératives. L'utilisation de tels verrous délègue à un nœud une partie de l'environnement virtuel. Il ne peut y avoir d'incohérence car les autres nœuds ne peuvent agir dans cette partie.

Hespanha *et al.* [HMS<sup>+</sup>00] ont proposé un compromis entre collaboration et coopération en fonction de la grandeur des retards entre les différents nœuds. Ce système est basé sur un système de verrous de sous-région de l'environnement virtuel. Deux nœuds peuvent collaborer si le retard et la variation du retard (ou gigue) entre les deux sont faibles. Ces retards faibles diminuent les risques d'occurrence d'une incohérence. Entre deux groupes, les utilisateurs ne peuvent que coopérer.

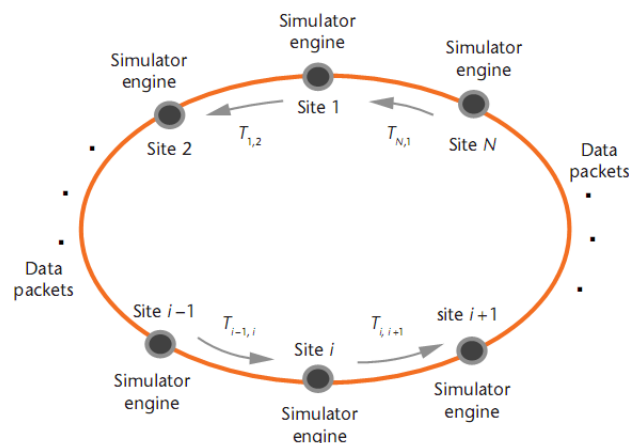
L'utilisation de verrous tend généralement à limiter voir interdire la collaboration au profit d'une coopération. D'autres méthodes doivent donc être mise en œuvre.

#### 1.4.1.2.b Contrôle de la cohérence intrinsèque

Une des techniques pour diminuer les incohérences dans une simulation est d'utiliser des algorithmes de prédictions. En utilisant une information extrapolée des données reçues, ces algorithmes permettent de compenser partiellement les effets du retard. Hiki-chi *et al.* [HMF<sup>+</sup>01] ont présenté une telle plateforme utilisant une architecture distribuée. En utilisant des algorithmes se basant sur une prédiction plus complexe des gestes des utilisateurs, Shen *et al.* [SBNG03] ont proposé une plateforme distribuée similaire. Boukerche *et al.* [BSH05] ont choisi également d'extrapoler la position des objets manipulés associé à de la qualité de service.

La prédiction ne permet pas de garantir la cohérence d'un univers ni de corriger un univers incohérent. Pour corriger la cohérence d'un univers, des algorithmes plus complexes doivent être mis en œuvre. Une autre approche vise à garantir que chacun des nœuds traitent les mêmes informations dans le même ordre. Pour ce faire il faut que les différents nœuds puissent se synchroniser. Lamport [Lam78] a proposé une horloge logique permettant d'établir un ordre partiel dans une application distribuée. Le principe est d'associer un entier ou *estampille* à chaque événement échangé entre les nœuds de l'application. Chaque nœud possède une horloge qui est incrémentée avant chaque événement. Lorsqu'un nœud reçoit un événement d'un autre nœud, il synchronise son horloge sur l'estampille de l'événement si celle ci est plus grande que l'horloge. Les horloges logiques permettent donc de synchroniser les différents nœuds ensemble.

En utilisant ce principe, Mauve *et al.* [MVHE04] ont proposé deux algorithmes complémentaires permettant de garantir la cohérence d'un univers virtuel. Le premier *Local Lag* est un retard supplémentaire entre la réception des données et leur application à la simulation. Il permet d'attendre que les données qui subissent un retard variable, arrivent. Comme ce retard



**Figure 1.24** – Organisation des moteurs physiques suivant un anneau. Issu de [CPN09]

supplémentaire est borné et fixe il ne peut compenser que les incohérences intervenant sur un intervalle de temps court. Les auteurs lui ont donc associé un algorithme appelé *Timewrap*. Le principe est de recalculer un état cohérent au moment où les données manquantes sont reçues. Sur chaque nœud, les états précédents de la simulation et les données reçues sont conservés en mémoire et associés à une estampille. De la même manière les données échangées sont chacune associées à une estampille. Lorsqu'un nœud reçoit une donnée qui a subi un retard trop important, la simulation est recalculée à partir de l'état de l'univers virtuel antérieur à l'émission de la donnée reçue et de toutes les données reçues dans l'intervalle jusqu'au pas de temps courant. Ce nouvel état devient alors l'état courant.

En se basant sur le principe du *timewrap*, Glencross *et al.* [GJF<sup>+</sup>07] ont réalisé une plateforme de collaboration avec retour d'effort permettant de manipuler des objets rigides issus de la CAO sur trois degrés de liberté entre l'University of Manchester en Angleterre et l'University of North Carolina at Chapter Hill aux États-Unis. De plus, afin de limiter la charge de calcul lorsque des données sont retardées, les auteurs ont utilisé des algorithmes de prédiction. La simulation est recalculée que si la prédiction s'avère fautive. Ces algorithmes sont inspirés de ceux utilisés pour les bases de données relationnelles. La plupart des bases de données peuvent en effet effectuer des *rollback*, c'est à dire annuler une succession d'action suite à une incohérence pour ensuite réappliquer une suite d'instruction (ou transaction) et obtenir un état cohérent.

Deux autres approches permettant de lier des simulations physiques sans retour d'effort peuvent être aussi détaillées :

Afin de d'obtenir la convergence des états des différents moteurs physiques, Dequidt *et al.* [DGC05] ont proposé de calculer l'état de chaque objet sur un nœud comme une somme pondérée des états des instances de ces objets partagés sur les autres nœuds. Le choix de la fonction de pondération permet d'obtenir un comportement client-serveur ou de privilégier les états des nœuds les plus proches, c'est à dire pour lesquels le retard de communication est le plus faible.

Enfin, Cheong *et al.* [CPN09] ont suggéré de lier les différents nœuds suivant un anneau. Chaque nœud est donc lié à seulement deux autres nœuds (voir Figure 1.24). Chacun des nœuds envoie l'état (positions et vitesses) des objets qu'il simule au nœud suivant sur l'anneau. Ces données reçues permettent, à l'aide d'un prédicteur de Smith, d'obtenir une convergence des états des différents nœuds.

### 1.4.1.3 Architectures hybrides

Les deux architectures précédentes ont chacune leurs avantages et leur inconvénients. En combinant ces deux architectures il est possible pour certaines applications de réduire une partie des inconvénients. Le principe est d'utiliser une architecture client-serveur où le serveur peut se déplacer en fonction des besoins. Buttolo *et al.* [BOH97] ont proposés d'utiliser une architecture distribuée où le serveur migre sur chacun des clients en utilisant un anneau à jeton (*token ring*) pour gérer les priorités. Une telle architecture permet donc une coopération au sein de l'univers virtuel en éliminant les retards de communications lors d'une interaction. Un serveur centralisé est chargé de gérer les requêtes. Lorsqu'un client a fini d'agir, le serveur migre vers un autre client. Marsh *et al.* [MGPH06] ont proposé une solution sensiblement similaire. L'architecture dispose de plusieurs serveurs où un seul serveur est actif à la fois, seul les clients reliés à ce serveur peuvent agir sur la simulation. Les serveurs sont répartis de telle manière que les retards présents entre les clients et leur serveur soit très faible. Ces architectures sont particulièrement intéressantes pour les tâches de coopération.

Dans le cas de retard faible (maximum 5 ms, Morris *et al.* [MSB<sup>+</sup>04] ont utilisé une approche légèrement différente pour une simulation chirurgicale, mais aussi basée sur une architecture distribuée. Lorsque que les utilisateurs agissent indépendamment sur la simulation, chacun travaille localement sur le moteur physique situé sur le nœud auquel il est connecté. Lorsque les utilisateurs souhaitent collaborer sur une même tâche, ils se connectent au même nœud et agissent sur le même moteur physique, l'architecture équivalente est alors une architecture clients-serveur.

## 1.4.2 Retour haptique sur une tâche collaborative

L'ajout de rendu haptique pour des tâches collaboratives doit permettre aux utilisateurs une meilleure immersion dans l'univers virtuel et une interaction plus efficace avec ce dernier.

En effet, Basdogan *et al.* [BHSS00] ont montré que l'haptique permet d'améliorer les performances lors d'une tâche collaborative. Dans cette étude, deux utilisateurs, un sujet et un expert, déplacent un anneau le long d'un fil d'un point de départ à un point d'arrivée. L'anneau ne peut se déplacer que si les deux utilisateurs exercent chacun simultanément un effort de direction opposée. Avec du retour d'effort, les deux personnes sont plus précises et l'anneau entre un contact moins souvent avec le fil. Ils ont, également, plus de facilité à maintenir l'anneau en mouvement, c'est à dire à maîtriser l'effort qu'ils appliquent chacun à l'anneau. Sallnäs *et al.* [SRGS00] ont étudié une tâche relativement similaire où des sujets doivent empiler des cubes. Comme l'étude précédente, les sujets devaient pincer un des cubes pour le déplacer. Lorsque cette tâche est effectuée avec un retour d'effort via un PHANTOM OMNI, les temps de réalisation sont moins longs que sans retour d'effort. En outre, le retour d'effort permet aux utilisateurs de plus facilement coordonner leurs actions quand ils soulèvent un cube. Shen *et al.* [SBNG03] ont également constaté une diminution du temps de réalisation d'une tâche avec l'ajout d'un retour d'effort. Dans leur cas il s'agissait de déplacer un cube au travers d'un labyrinthe.

Hubbold [Hub02] a étudié le cas d'un déplacement d'un brancard dans une usine (environnement moyennement encombré). Il a aussi observé une amélioration de la coordination des mouvements des utilisateurs lors du déplacement du brancard. Il a également noté que l'utilisation d'interface haptique ayant des espaces de travail différent (en l'occurrence PHANTOM OMNI et un Argonn Arm) peut tendre à déséquilibrer les utilisateurs.

L'ajout de retour d'effort dans une tâche collaborative n'améliore cependant pas systématiquement les temps de réalisation de la tâche, ainsi Sallnäs et Zhai [SZ03] ont testé une tâche de pointage basée sur la loi de Fitts [Fit54] avec deux sujets. Les essais avec un retour

d'effort ne se caractérisent pas par un temps d'exécution plus rapide contrairement aux deux expériences précédentes. Par contre, les personnes ont laissé tomber le cube moins souvent lorsqu'ils se l'échangeaient. Ainsi, comme pour les deux expériences précédentes les sujets coordonnent mieux leurs mouvements avec un retour d'effort.

Enfin, Sallnäs *et al.* [SRGS00] ont observé que l'ajout d'un retour d'effort n'améliore pas le temps de réalisation ni la précision d'une tâche si les sujets peuvent communiquer entre eux via un canal audio.

L'ajout d'un retour haptique dans une tâche collaborative au sein d'un environnement virtuel n'influence donc pas la tâche de la même façon. Dans certains cas, la tâche est effectuée plus rapidement, dans d'autres cas il n'y a pas de différence de temps mais les sujets sont plus précis. Ces résultats sont concordants avec les expériences menées sur des tâches non collaboratives ([GWZ97, LMB<sup>+</sup>02, DHM<sup>+</sup>06]) où le retour d'effort permet tantôt d'effectuer des tâches plus rapidement, tantôt d'être plus précis. Néanmoins, Edwards [Edw00] a montré un cas où le retour d'effort augmente le temps de la tâche et diminue la précision et les performances des sujets.

### 1.4.3 Influence du retard sur la coopération

Il y a deux éléments à considérer dans le retard que subissent les informations, une partie fixe et une partie variable, la gigue. Ces deux éléments n'influencent pas de la même manière la perception et les comportements des utilisateurs lors d'une tâche collaborative avec retour d'effort. En effet, un retard variable déforme l'information transmise en plus de la retarder.

Lorsque qu'une information (audio, visuelle, haptique, etc.) est retardée, cela influe sur la perception de l'action observée. La causalité entre deux événements est perçue par l'être humain si le temps qui s'écoule entre deux événements est en dessous d'un certain seuil [Mic63]. En plus d'influencer la perception, ce retard peut aussi influencer de manière consciente ou non le comportement et les performances des utilisateurs.

Sheridan et Farell [SF63] ont testé l'influence du retard dans une tâche de positionnement en téléopération. La durée de la tâche croît avec le retard entre le maître et l'esclave. De plus, avec un retard élevé les temps de positionnement sont plus importants. D'une manière similaire, MacKenzie et Ware [MW93] ont montré que l'ajout d'un retard dans une tâche de Fitts entraîne un temps de réalisation plus long et augmente le nombre d'erreurs (le sujet manque la cible). Les auteurs proposent un modèle similaire à la loi de Fitts où l'ajout de retard est équivalent à augmenter l'indice de difficulté de la tâche.

Lors de l'exécution d'une tâche collaborative, les différentes personnes interagissant doivent coordonner leurs actions. Afin d'être plus efficace, ils se basent sur les mouvements des autres personnes pour prédire les actions des autres utilisateurs et mieux adapter les leurs. Gutwin [Gut01] a étudié l'influence du retard et de la gigue sur des tâches de prédiction et de coordination. Comme pour les études précédentes il a constaté un allongement du temps de réalisation et une augmentation du nombre d'erreur lors de tâche préhension.

Jordan *et al.* [JMO<sup>+</sup>02] ont étendu les travaux de Basdogan *et al.* [BHSS00] dans le cas où les sujets sont sur des sites distants. L'ajout de l'haptique permet dans ce cas d'augmenter le sentiment de présence, les performances des utilisateurs n'ont pas été évaluées.

Les données peuvent être désynchronisées. Si les données haptiques et visuelles concernant un pas de temps sont émises dans des paquets de données différents, rien ne garantit qu'elles subissent le même retard. De plus, les dispositifs visuels et haptiques peuvent ajouter des retards supplémentaires.

Jay et Hubbold [JH06] ont montré que si le retour visuel et le retour haptique ne sont pas synchronisés (retards différents sur chacune des informations), l'information visuelle devient prépondérante. Dans ce dernier cas le retard sur le retour haptique n'a que peu d'influence sur la performance des utilisateurs, alors que le retard visuel augmente nettement le temps de réalisation de la tâche. Dans une seconde tâche, où le sujet doit alternativement venir taper deux cibles, le retour visuel est toujours prépondérant dans le cas d'un retard sur l'un des deux retours. Néanmoins si le retour visuel est retardé de plus de 100 ms alors que le retour haptique ne l'est pas, la tâche est réalisée légèrement plus rapidement. Thompson *et al.* [TOS98] ont également constaté en téléchirurgie que pour des tâches collaboratives, une asynchronie entre les informations visuelles et haptiques ne modifie pas le temps de réalisation de la tâche. Dans ce cas, les informations visuelles sont retardées par rapport aux informations haptiques car elles doivent être décompressées avant d'être affichée. A contrario lorsque le chirurgien manipule seul un outil, le temps de réalisation est plus faible si les données sont asynchrones, c'est à dire si elles sont fournies à l'utilisateur dès qu'elles sont reçues. Le couplage haptique est également plus stable car il n'y a pas de délai artificiel nécessaire à la synchronisation.

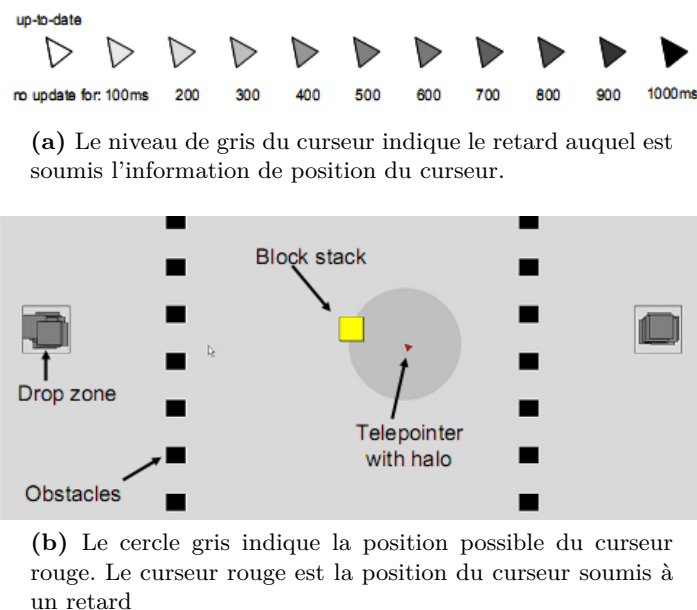
Enfin, le retard n'influence pas de manière linéaire sur le comportement des utilisateurs. Dans une tâche collaborative de déplacement d'objet, Jay *et al.* [JGH07] ont constaté plusieurs comportements en fonction du retard :

- lorsque celui-ci est faible, inférieur à 25 ms, la performance est stable, il y a peu d'erreur.
- pour un retard compris entre 25 ms et 50 ms, le nombre d'erreur est proportionnel au retard. Cependant, les personnes n'ont pas conscience de ce retard.
- pour un retard compris entre 50 et 100 ms, les sujets ont conscience du retard, mais ne modifient pas leur stratégie de manipulation car les mouvements d'un sujet semblent cohérents à l'autre sujet, le nombre d'erreur est toujours croissant et proportionnel au retard.
- au delà de 100 ms, les sujets changent de stratégie et le nombre d'erreur diminue très légèrement puis augmente de manière légère avec une augmentation du retard

Ce changement de comportement a aussi été observé par Sheridan et Farell [SF63] sur une architecture client-serveur. Pour des retards importants, les sujets adoptent une stratégie tour par tour. Ils attendent en effet d'avoir un retour d'information sur le mouvement qu'ils viennent d'effectuer avant de commencer une seconde tâche. Une seconde stratégie a été observée par Gutwin [Gut01]. Dans le cas de retards importants, les utilisateurs peuvent effectuer leur tâche en cadence. Ils ne se fient donc plus au retour d'information visuel ou haptique. Les deux sujets adoptent spontanément un rythme permettant d'effectuer leur tâche de préhension sans interférence.

#### 1.4.3.0.a Influence de la gigue

En déformant l'information transmise, la gigue influence le comportement des utilisateurs d'une manière différente du retard fixe. Son aspect imprédictible (en particulier sur des réseaux du type Internet) contraint les sujets à observer une action en entier avant d'agir. Gutwin [Gut01] a ainsi observé que les sujets ne peuvent pas faire la différence entre un arrêt volontaire d'un mouvement d'un sujet distant et une information temporairement retardée sous l'effet de la gigue. L'information retardée a pour conséquence d'arrêter temporairement le mouvement perçu de l'utilisateur distant. Les utilisateurs adoptent alors une stratégie tour par tout et la collaboration est rompue pour laisser place à une coopération. Lors d'un accès



**Figure 1.25** – Deux exemples de décorateurs permettant d'indiquer le retard et la gigue.

concurrent à une ressource (prise de pièce par exemple), une gigue importante à tendance à augmenter le nombre d'erreur, c'est à dire de prise simultanée.

Pour diminuer la gigue, il est possible d'augmenter le retard par un retard artificiel comme le propose Mauve *et al.* [MVHE04] avec l'utilisation d'un *Local Lag*. Hikichi *et al.* [HMY02] ont étudié le compromis entre un retard faible possédant une gigue importante, retard important possédant une gigue faible jusqu'à une gigue nulle et un retard fixe. Ils ont interrogés les sujets sur le compromis qu'ils préféreraient. Les auteurs n'ont pas constaté de différence si le retard est la gigue sont faibles. Pour des retards et giges plus importants (un premier retard suivant une distribution gaussienne de moyenne 50 ms et 60 ms et second retard suivant également une distribution gaussienne de moyenne 100 ms et 100 ms), les sujets semblent préférer un compromis médian où le retard absorbe une partie de la gigue. Une gigue importante avec un retard faible comme un retard important avec une gigue nulle sont les deux cas qui sont les plus gênants pour les sujets.

Pour palier à ces effets, Gutwin *et al.* [GBF<sup>+</sup>04] ont suggéré d'ajouter des aides visuelles ou *décorateurs* afin d'indiquer aux utilisateurs le retard présent dans l'univers virtuel (voir Figure 1.25). L'ajout de ces métaphores visuelles permet de diminuer le taux d'erreurs sur des expériences similaires à celle réalisée par Gutwin [Gut01].

## Conclusion

Dans ce chapitre, nous avons tout d'abord présenté les aspects techniques permettant de simuler des univers virtuels physiques. Il existe une certaine variété de choix possibles entre les détecteurs de collisions, les schémas d'intégration disponible et les méthodes de résolution du contact. Il y a donc un compromis à définir entre les performances attendues du moteur physique et les méthodes utilisées. Ces éléments nous serviront en particulier de base pour la réalisation d'une plateforme physique dans le chapitre 2.

Dans un second temps, nous avons détaillé le problème de stabilité lors d'ajout de retour haptique à un environnement virtuel. En particulier, le comportement actif de certains éléments de ces systèmes lors du rendu d'un mur virtuel engendre des instabilités. L'énergie



du système peut être alors modifiée par dissipation afin de conserver un système passif. Les couplages virtuels permettent quand à eux de moduler l'impédance transmise afin de conserver un environnement stable. À ces méthodes, des techniques basées sur des phénomènes transitoires peuvent créer l'illusion d'une meilleure performance du rendu haptique.

Le retard introduit par la distance entre les sites de manipulation tend à déstabiliser de tels systèmes. Les méthodes issues de la téléopération, en se basant sur les principes de passivité ou de couplages virtuels, sont utilisées pour réaliser des tâches avec retour d'effort à distance. Nous avons souligné plusieurs approches, dont certaines permettent de stabiliser un système de téléopération indépendamment du retard, comme pour les variables d'onde ou les contrôleurs de passivité.

Enfin, nous avons exposé l'apport du retour d'effort sur des tâches collaboratives et coopératives. Lorsque l'application relie plusieurs lieux géographiques nous avons soulevé l'importance du choix de l'architecture de la plateforme. En particulier, les problèmes de retard et d'asymétrie pour l'architecture client-serveur et les problèmes de cohérence pour l'architecture distribuée. Nous avons également présenté l'influence négative du retard sur les tâches collaboratives avec retour d'effort.

A partir de cette synthèse nous pouvons faire les remarques suivantes :

- Les retards introduits ont plusieurs effets négatifs. Premièrement au niveau de la téléopération, ils introduisent des problèmes importants de stabilités. Un système de téléopération peut être rendu stable mais au prix d'une perte sensible de transparence du système. Pour les environnements virtuels collaboratifs, le retard peut entraîner une perte de causalité dans le cas de client-serveur ou des problèmes d'incohérence dans le cas d'architecture distribuée. Enfin, ces retards peuvent engendrer une perte de performance lors de la réalisation d'une tâche. Ils peuvent amener les utilisateurs à modifier leur comportement, au delà d'un certain seuil. Il est, en outre, à noter que la gigue (part variable du retard) aggrave les problèmes de stabilité d'une part de perception d'autre part lorsqu'elle devient importante. Dans ces différentes études, en particulier au niveau des interactions collaboratives, il n'y a pas de considération sur l'influence des schémas de couplage retenus. C'est à dire de l'influence des efforts parasites, comme, par exemple, l'inertie supplémentaire dans le cas des variables d'onde.
- D'une part, les études menées dans le domaine de téléopération permettent de réaliser des manipulations haptiques distantes de manière stable. D'autre part, il existe plusieurs plateformes permettant de réaliser des tâches collaboratives avec retour d'effort. Néanmoins, la plupart de ces plateformes sont limitées à 3 degré de liberté ([CNAS05, FSC07]) et n'utilisent pas forcément de simulateur physique associé avec un détecteur de collision. De plus la majorité d'entre eux [HMF+01, KKT+04, ICG+06, GJF+07] ne garantissent pas la stabilité du rendu haptique. La qualité du rendu haptique n'est en outre généralement pas évaluée. Le lien entre, d'une part, les aspects stabilités et qualité du rendu haptique étudié en téléopération et d'autre part, la possibilité de réaliser des tâches collaboratives à partir d'environnements virtuels n'est pas réellement traité dans la littérature existante.
- On peut également remarquer que certaine technique utilisé au sein des environnements virtuels, tel le rendu haptique événementiel, n'est actuellement pas mis à profit dans les environnements collaboratifs avec retour d'effort. Ces méthodes peuvent être intéres-

santes dans le sens où elles ne sont pas sujettes aux problèmes de stabilité rencontrés par le retour haptique en boucle fermée.



# Plateforme de collaboration

## Sommaire

<b>2.1 Simulation physique à 6 degrés de liberté</b> . . . . .	<b>50</b>
2.1.1 Formulation . . . . .	50
2.1.2 Formulation quasi-statique . . . . .	50
<b>2.2 Architecture réseau pair-à-pair</b> . . . . .	<b>52</b>
2.2.1 Motivations . . . . .	52
2.2.2 Conception . . . . .	53
<b>2.3 Architecture logicielle proposée</b> . . . . .	<b>54</b>
2.3.1 Architecture des nœuds . . . . .	54
2.3.2 Les aides visuelles . . . . .	56
<b>2.4 Divergence et cohérence</b> . . . . .	<b>56</b>
2.4.1 Point de préhension et point de couplage . . . . .	57
2.4.2 Point de couplage et manipulation multi-utilisateurs . . . . .	61
2.4.3 Gestion de la cohérence . . . . .	62
<b>Conclusion</b> . . . . .	<b>64</b>

LE comportement du retour d'effort lors de la réalisation d'une tâche collaborative dépend de l'architecture retenue lors de la conception de la plateforme collaborative (voir Section 1.4.1). Dans le cas du prototypage virtuel, nous proposons dans ce chapitre, une plateforme de coopération distribuée permettant d'agir sur un environnement virtuel composé d'objets rigides potentiellement complexes pouvant être rencontrés dans des scénarios de prototypage virtuel. Bien que pouvant simuler des objets polyarticulés (robots, humain virtuel), nous avons décidé de nous focaliser sur les objets rigides uniquement. L'architecture distribuée introduit une divergence entre les états des différents nœuds. Ces divergences peuvent mener à l'apparition d'incohérences (voir les sections 1.4.1.2.a et 1.4.1.2.b). Nous avons néanmoins choisi de tolérer l'apparition de ces incohérences.

L'architecture choisie et la plateforme collaborative présentée dans ce chapitre seront utilisées dans la suite de ce manuscrit où nous étudierons la stabilité du système ainsi formé et la qualité du retour d'effort proposé.

La Section 2.1 présente la formulation de la loi d'évolution de l'environnement virtuel que nous avons adoptée. Dans la Section 2.2, nous présenterons les motivations présidant au choix d'une architecture distribuée ainsi que l'organisation de la plateforme de manière générale. Ensuite, la Section 2.3 décrit les aspects logiciels propres à cette plateforme. Enfin, dans la Section 2.4 nous présenterons l'influence du choix des points de préhension et des points d'asservissement sur la manipulation et particulièrement sur la divergence entre les instances d'un objet partagé. Nous proposerons également une méthode permettant d'indiquer aux utilisateurs l'apparition d'incohérences.

## 2.1 Simulation physique à 6 degrés de liberté

Le moteur physique que nous avons utilisé repose sur les principes abordés dans la Section 1.1. Il est donc constitué d'un simulateur physique calculant l'évolution de l'environnement virtuel et d'un détecteur de collision fournissant les informations de contacts.

Un moteur physique est chargé de simuler l'environnement virtuel sur chacun des nœuds de la plateforme collaborative. Pour cela, il résout la loi d'évolution Equation 1.2 issue du principe fondamental de la dynamique. Les sections suivantes permettent de définir la formulation adoptée ainsi que les notations qui seront utilisées dans les chapitres suivants.

### 2.1.1 Formulation

L'univers virtuel que nous souhaitons simuler est un espace à 3 dimensions dans lequel évoluent des objets virtuels rigides. La position d'un objet virtuel peut être ainsi décrite par une matrice homogène  $H$  qui appartient au groupe de Lie  $SE(3)$ . Les vitesses angulaires et linéaires de cet objet peuvent être décrites par un torseur cinématique  $T$  qui est un élément de l'algèbre de Lie  $\mathfrak{se}(3)$ . Les torseurs d'effort cinématique, notés  $W$ , sont quand à eux des éléments de l'algèbre  $\mathfrak{se}^*(3)$  duale de la précédente. Les notations de [Str01] sont adoptées par la suite pour ces différents éléments (voir A.1).

L'expression de l'équation d'évolution d'un objet virtuel résolue par la simulation physique au sein du moteur physique peut être exprimé dans de tels espaces au centre de masse  $C$  de l'objet virtuel par [MLS94] :

$$\mathbf{M}\dot{T}_C^C + \mathbf{N}T_C^C = \sum_{i=0}^n W_{C,i}^C + W_{C,g}^C \quad (2.1)$$

$\mathbf{M}$  et  $\mathbf{N}$  sont respectivement les matrices d'inertie généralisées et la matrice contenant les effets centrifuges et les effets Coriolis.

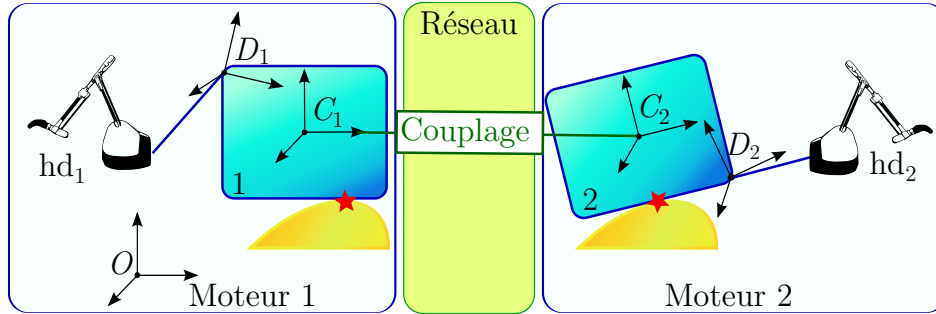
$T_C^C$  est une vitesse cartésienne (ou torseur cinématique) de l'objet au point  $C$  dans un repère lié à l'objet en  $C$ .  $W_{C,i}^C$  est un des torseurs d'effort externe. Ces contacts sont composés d'efforts d'asservissement entre les instances des objets répliqués et d'efforts de contacts entre les différents objets d'une simulation. Les asservissements que nous avons utilisés sont détaillés dans les chapitres 3 et 4. Les informations de contacts sont issues d'un détecteur de collision, comme ceux décrits dans la Section 1.1.1. En particulier, nous avons utilisés les deux algorithmes LMD et VPS. Nous avons utilisé VPS pour de grande scène virtuelle où le temps de calcul nécessaire à LMD pour détecter les contacts peut dépasser le temps qui est alloué au moteur physique.

### 2.1.2 Formulation quasi-statique

Bien que la formulation soit indépendante du nombre de nœuds et de l'interface haptique utilisée, dans la suite du document, seuls deux nœuds (notés 1 et 2, voir Figure 2.1), et donc deux moteurs physiques, seront considérés, afin de simplifier l'écriture des équations et de clarifier l'exposé. . Lors des développements suivants, le point vue adopté sera celui d'un des deux nœuds. Ce nœud sera noté  $i$  et le nœud distant  $j$ . Les objets virtuels, les interfaces haptiques et les utilisateurs sont repérés par l'indice  $i$  (respectivement  $j$ ) selon leur présence sur le nœud  $i$  (respectivement  $j$ ).

Lors d'une tâche de prototypage virtuel, la gravité et les effets inertiels sont souvent négligés. En effet, ces efforts ne sont pas nécessaires dans le cadre d'une tâche de prototypage virtuel. Notamment lorsqu'on cherche à vérifier la possibilité d'assembler des éléments ensembles ou à vérifier l'accessibilité d'une pièce.

Pour cette plateforme, on choisit donc une formulation quasi-statique peut être adoptée [MPT99]. L'équation d'évolution Equation 2.1 résolu par le moteur physique  $i$  peut être ainsi reformulée (voir Fig. 2.1) :



**Figure 2.1** – Deux utilisateurs (1 et 2) manipulent le même objet via deux interfaces haptiques  $hd_1$ ,  $hd_2$  à deux points de préhension distincts ( $H_{C_1}^{D_1} \neq H_{C_2}^{D_2}$ ). Chaque objet est soumis à trois efforts. Le premier est lié à l'interface haptique, le second au couplage entre les deux simulations et le dernier aux collisions de l'objet avec son environnement (indiquées par des étoiles rouges).

$$\mathbf{B}_s T_{C_i}^{C_i} = W_{C_i, j}^{C_i} + \text{Ad}_{H_{C_i}^{D_i}}^T W_{D_i, hd_i}^{D_i} + W_{\text{contacts}}^{C_i} \quad (2.2)$$

où  $(i, j) = (1, 2)$  du côté du moteur physique 1 et  $(i, j) = (2, 1)$  du côté du moteur physique 2.

Dans cette formulation quasi-statique, le terme de gauche,  $\mathbf{B}_s T_{C_i}^{C_i}$ , est un effort visqueux. Il est introduit afin de pouvoir résoudre l'équation d'évolution dans tous les cas.  $B$  est une matrice d'amortissement.  $\mathbf{B}_s$  est choisie comme étant une matrice diagonale avec des coefficients faibles. Si les termes de droite de l'Equation 2.2 sont non nuls et dépendent de  $T_{C_i}^{C_i}$ , ce qui est le cas habituel, la matrice  $B_s$  peut être nulle. Annuler la matrice  $\mathbf{B}_s$  permet d'éviter l'ajout d'un terme de viscosité globale pouvant gêner les utilisateurs lors de la manipulation.

Le torseur d'effort  $W_{C_i, j}^{C_i}$  permet d'asservir l'instance  $i$  sur l'instance  $j$  de l'objet partagé en prenant en compte le délai entre les moteurs physiques et les problèmes de stabilité. Ce torseur dépend de la position et de la vitesse des deux instances de l'objet partagé. Il est créé par le couplage existant entre les deux nœuds  $i$  et  $j$ . Au repos, cet effort doit permettre la convergence en position des instances des objets partagés.

$\text{Ad}_{H_{C_i}^{D_i}}^T W_{D_i, hd_i}^{D_i}$  est un torseur d'effort généré par l'asservissement à l'interface haptique manipulant l'objet  $i$  au point de préhension  $D_i$ . L'effort  $W_{D_i, hd_i}^{D_i}$  est généré par un couplage virtuel permettant d'asservir la position de l'objet sur la position de l'interface haptique. En l'occurrence, l'interface haptique est pilotée en impédance, le couplage virtuel utilisé est un ressort amortisseur [CSB95, AH98]. L'expression de l'effort généré par le ressort-amortisseur dans un espace à 6 degré liberté est disponible dans [FB98] :

$$W_{D_i, hd_i}^{D_i} = W_{\text{ressort}}(H_{hd_i}^O, H_{D_i}^O, \mathbf{K}) + \mathbf{B} \left( T_{hd_i}^{D_i, O} - T_{D_i(t)}^{D_i, O} \right) \quad (2.3)$$

où  $\mathbf{K} = \mathbf{K}^T = \begin{pmatrix} \mathbf{K}_r & \mathbf{K}_c \\ \mathbf{K}_c^T & \mathbf{K}_t \end{pmatrix}$  et  $\mathbf{B} = \mathbf{B}^T = \begin{pmatrix} \mathbf{B}_r & \mathbf{B}_c \\ \mathbf{B}_c^T & \mathbf{B}_t \end{pmatrix}$ ,  $\mathbf{K}, \mathbf{B} \in \mathbb{R}^6$

Le terme  $W_{\text{ressort}}(H_{hd_i}^O, H_{D_i}^O, \mathbf{K})$  dérive d'un potentiel d'énergie, il est fonction de la position de l'objet  $H_{D_i}^O$ , de la position de l'interface haptique  $H_{hd_i}^O$  et d'une matrice de raideur

**K**. **K** est la matrice de raideur où  $\mathbf{K}_t$  est la partie linéaire du ressort,  $\mathbf{K}_r$  la partie angulaire et  $\mathbf{K}_c$  le terme liant les rotations aux translations. **B**, la matrice d'amortissement est définie de manière analogue à **K**. Les termes extra-diagonaux ( $\mathbf{K}_c$ ,  $\mathbf{B}_c$ ) sont généralement nuls. Le couplage étant généralement isotrope, les blocs diagonaux sont des multiples de la matrice identité. Les termes de couplage d'un tel ressort-amortisseur à 6 degrés de liberté sont donc les scalaires  $k_t$ ,  $k_r$ ,  $b_t$  et  $b_r$  correspondant aux matrices  $\mathbf{K}_t$ ,  $\mathbf{K}_r$ ,  $\mathbf{B}_t$ ,  $\mathbf{B}_r$  respectivement.

L'adjoint  $\text{Ad}_{H_{C_i}^{D_i}}$  permet de déplacer le torseur d'effort du point  $D_i$  au centre de masse  $C_i$  de l'objet. Dans le cas où aucune interface haptique n'est liée à l'objet virtuel ce terme est nul.

Le dernier torseur d'effort,  $W_{\text{contacts}}^{C_i}$ , est calculé à partir des collisions fournies par un algorithme de détection comme ceux présentés dans la section 1.1.1.

Les états des instances  $i$  et  $j$  de l'objet partagé sont transmis avec des retards, ainsi l'état  $S_j^i(t)$  de  $j$  sur le nœud  $i$  est égal à celui sur la simulation  $j$  retardé du temps de communication de  $j$  vers  $i$ , noté  $T_j$ . D'où  $S_j^i(t) = S_j^j(t - T_j)$  et réciproquement,  $S_i^j(t) = S_i^i(t - T_i)$ . Le retard total est noté  $T_r = T_i + T_j$ .

## 2.2 Architecture réseau distribuée

### 2.2.1 Motivations

La plateforme de collaboration doit permettre de réaliser des tâches collaboratives dans le cadre du prototypage virtuel. C'est à dire, dans notre cas, de manipuler simultanément un même objet virtuel rigide ou de manipuler des objets virtuels différents pouvant rentrer potentiellement en contact. Nous nous focaliserons donc sur le cas des tâches collaboratives, car elles sont plus exigeantes que les tâches coopératives ou les tâches d'exploration d'un environnement fixe [BOH97].

L'architecture que nous avons retenue pour réaliser des tâches collaboratives est une architecture distribuée. Ce choix est dû à trois considérations principales :

- L'utilisateur observe directement le résultat de ses actions sur l'environnement virtuel [GJF<sup>+</sup>07].
- Des effets additionnels (haptique événementielle) peuvent être réalisés sur chacun des nœuds sans être soumis à un retard (voir 5).
- Une architecture distribuée peut être très facilement être transformée en architecture hybride et ainsi proposer de meilleures performances pour une tâche coopérative [MGPH06].

Le fait que la réactivité soit importante est avantageux si un utilisateur explore ou entre en collision avec un élément fixe de la scène. Dans ces cas, la simulation qui se situe sur le nœud où agit l'utilisateur est équivalente à une simulation locale, c'est à dire, non soumis à un retard de communication. Il n'y a donc pas de retard entre l'action de l'utilisateur et la réaction (le retour d'effort) ressenti par ce dernier contrairement à ce que provoquer une architecture client-serveur. Pour les tâches de prototypage virtuel, les éléments fixes de l'environnement virtuel représentent souvent une partie importante de la scène (voir par exemple la manipulation de brancard dans une usine dans [Hub02]).

L'utilisation éventuelle de l'haptique événementielle bénéficie aussi d'une réactivité plus importante. Le temps écoulé entre une action d'un utilisateur et le retour d'information via l'haptique événementielle est uniquement dû au temps de calcul de la simulation. Dans le cas d'une architecture clients-serveur, le retard de communication introduit pourrait rompre le

lien de cause à effet entre deux événements [Mic63]. Avec une architecture client-serveur, une telle information devraient être émise par le serveur et être envoyée aux clients concernés par l'événement.

Enfin, une architecture hybride [MGPH06] peut être facilement dérivée d'une architecture distribuée. Les architectures hybrides (voir Section 1.4.1.3) sont particulièrement adaptées aux tâches coopératives qui sont couramment rencontrés dans des tâches de prototypage virtuel.

Cependant, cette architecture ne garantit pas de cohérence intrinsèque de l'univers virtuel comme évoqué précédemment (voir section 1.4.1). À un instant donné, les états des différents nœuds ne sont pas forcément identiques. La cohérence d'un environnement virtuel est un point critique dans l'utilisation d'une plateforme de réalité virtuelle. Les problèmes relatifs aux incohérences sont abordés dans la section 2.4.

### 2.2.2 Conception

La plateforme distribuée est constituée de différents nœuds hébergeant chacun un moteur physique. Afin que chaque utilisateur évolue dans le même univers virtuel, ce dernier est répliqué sur chaque moteur physique. Les objets virtuels constituant l'univers virtuel sont ainsi représentés par une instance sur chacun des moteurs physiques. La manipulation d'un objet virtuel se fait via une instance de celui-ci sur un nœud donné. Les différents nœuds échangent les états des différentes instances entre eux à travers un réseau informatique (voir Figure 2.3).

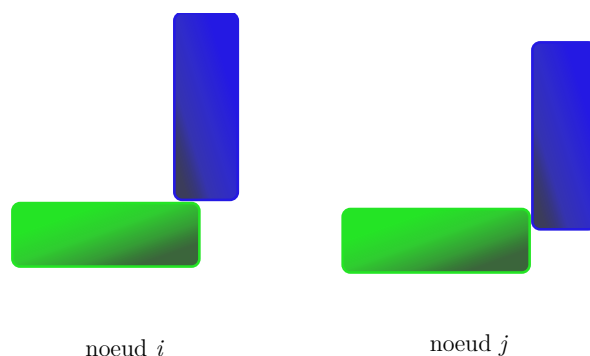
Les différentes simulations physiques utilisent un même repère commun dans lequel sont exprimées les positions, vitesses et accélérations des différentes instances représentant les objets partagés. Chaque instance d'un objet partagé est asservie sur les instances répliquées sur les autres moteurs physiques situés sur les différents nœuds de la plateforme collaborative. Ces asservissements permettent à chaque objet partagé de converger au repos vers une position commune  $H_{\text{objet}}^0$ . Ces asservissements sont traités dans les chapitres 4 et 3.

Ces asservissements permettent une certaine divergence en position et en vitesse entre les instances des objets partagé lorsque ceux-ci sont en mouvement ou soumis à des efforts. Cette divergence est susceptible d'entraîner des incohérences entre les différents nœuds. Nous avons également choisis de tolérer la présence des ces incohérences. Néanmoins, les incohérences sont signalées de manière explicite aux utilisateurs (voir Section 2.4.3) afin que ces derniers puissent adopter une stratégie adéquate (absence de correction, correction manuelle, ...). En effet une légère incohérence peut être ignorée, se corrigeant d'elle même lorsque la manipulation continue, alors que d'autres types d'incohérences peuvent nécessiter une correction spécifique de l'utilisateur. Le choix de ne pas corriger automatiquement les incohérences tient à trois considérations principales :

- Toutes les incohérences ne sont pas forcément dérangeante (voir Figure 2.2).
- L'ajout d'un mécanisme de correction est fortement susceptible de générer de l'énergie et donc de rendre instable la simulation.
- Certains mécanismes de correction basés sur des techniques de *rollback* [GJF<sup>+</sup>07] ne sont pas forcément réalisables compte tenu du temps de calcul nécessaire dans un cadre interactif sur des tâches complexes.

Il n'existe donc pas actuellement, d'après ce que nous avons rencontré dans la littérature, de moyen de corriger les incohérences lors d'une tâche collaborative avec retour d'effort, sans soit interrompre l'interaction, soit déstabiliser le système (ou introduire des artefacts haptique dans le meilleur des cas).

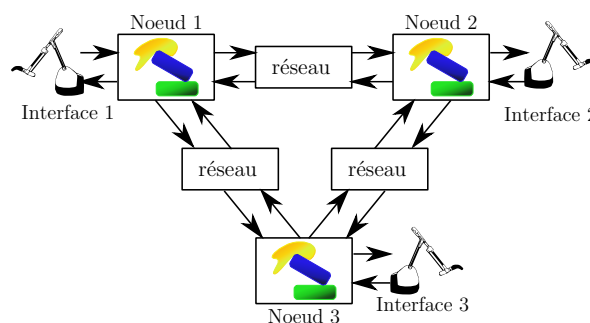




**Figure 2.2** – Cas d'une légère incohérence

Afin de diminuer la quantité de données échangées, les objets immobiles de la scène ne sont pas partagés comme les objets potentiellement mobiles.

Sur chaque simulation peuvent intervenir localement un ou plusieurs utilisateurs par l'intermédiaire d'une ou plusieurs interfaces haptiques. En effet, le simulateur physique utilisé permet à plusieurs personnes d'interagir au sein d'une même simulation.



**Figure 2.3** – Architecture de la plateforme collaborative à trois nœuds. Chacun des nœuds héberge une simulation de l'univers virtuel où les objets partagés sont répliqués. Les états de ces différents univers virtuels peuvent diverger en cours de manipulation.

Enfin, un des avantages de cette architecture est de pouvoir facilement être transformé en architecture hybride comme proposé par Marsh *et al.* [MGPH06] en faisant *migrer* un unique simulateur physique entre les différents nœuds. En pratique il suffit d'utiliser un asservissement unilatéral entre les nœuds et de ne permettre l'interaction (détection de collision, ...) que sur un seul nœud (voir Figure 2.4). Les nœuds n'hébergeant pas la simulation ne font que *rejouer* l'évolution de l'univers virtuel se déroulant sur le serveur. Dans le cas de tâches coopératives, couramment rencontrées dans l'industrie (revue de projet, formation, etc.), cette méthode permet d'éliminer les problèmes d'incohérence due à l'architecture distribuée et de réduire drastiquement les problèmes de stabilité pour une architecture client-serveur.

## 2.3 Architecture logicielle proposée

### 2.3.1 Architecture des nœuds

Les nœuds sont identiques en termes logiciel. Ils sont composés classiquement d'une simulation physique à 6 degrés de libertés associée à un moteur graphique et d'une interface haptique. Ces éléments sont cadencés à des fréquences différentes dans des fils d'exécutions différents (voir Figure 2.5).

Les données échangées entre les différents fils sont de deux natures :

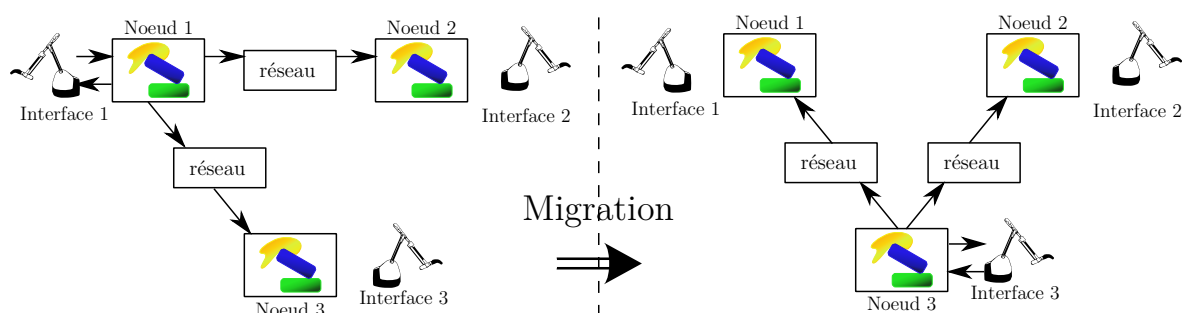


Figure 2.4 – Transformation en architecture hybride

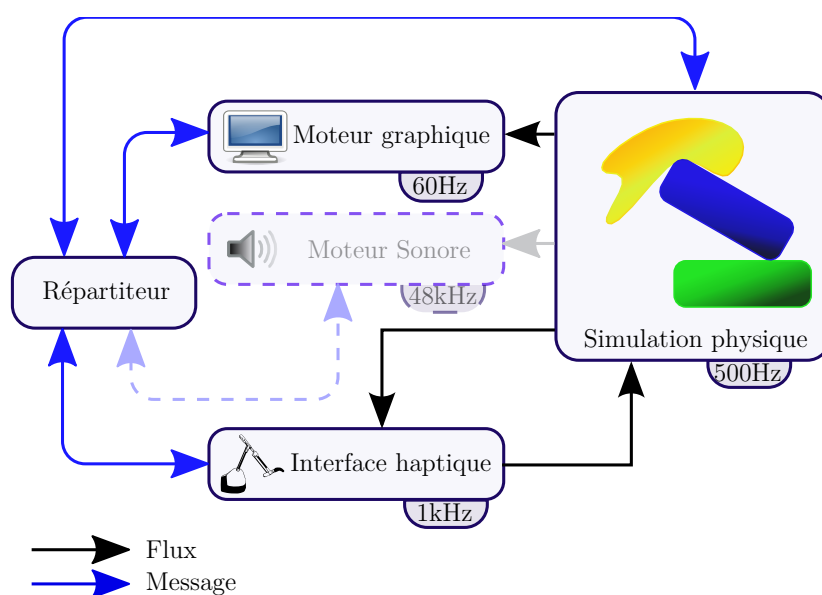
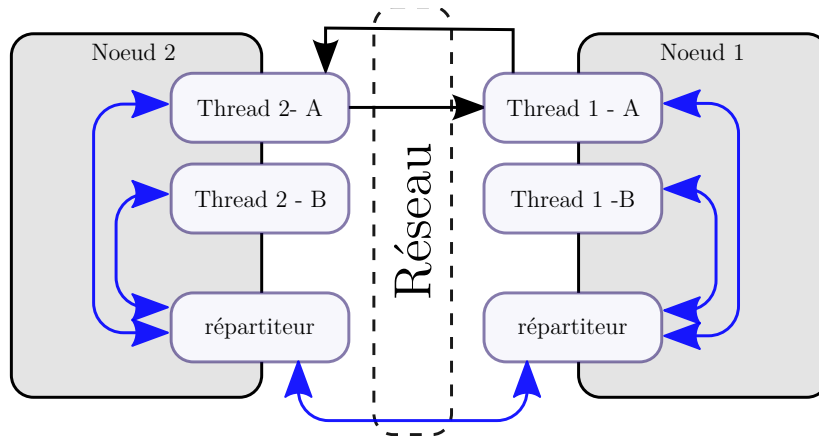


Figure 2.5 – Chaque nœud est constitué d'une simulation physique, d'un moteur graphique et d'une interface haptique. Un moteur sonore peut aussi être ajouté. Chacun de ces éléments est exécuté dans un fil spécifique. Ils peuvent être exécutés à des fréquences différentes.

- Les flux ou données synchrones : ils sont échangés à la fréquence des fils émetteur. Si l'information n'est pas lue par le fils récepteur qui à une fréquence plus faible, l'information est écrasée par la suivante. Les positions et vitesses des instances des objets partagés sont échangées via des flux.
- Les messages ou données asynchrones : ce sont des données asynchrones qui correspondent à des événements comme la pression d'un bouton sur l'interface haptique ou à l'initialisation de la scène virtuelle. Ces messages sont traités par un répartiteur dans un fil d'exécution indépendant qui se charge de les distribuer aux différents fils. Ces messages sont stockés ensuite dans des queues jusqu'à ce que le fils récepteur les traite.

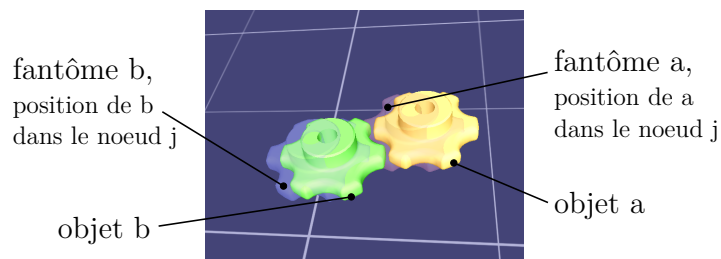
La communication entre les nœuds est effectuée de la même manière que la communication entre les différents fils, les données échangées sont de même nature, flux et message. Les flux utilisent le protocole UDP/IP alors que les messages utilisent le protocole TCP/IP afin de garantir leur réception. Les fils d'exécution qui doivent échanger des flux entre eux sont directement connectés. Les messages échangés entre les nœuds sont traités sur chaque nœud par un répartiteur qui les répartit ensuite vers les fils destinataires (voir Figure 2.6).



**Figure 2.6** – Les flux sont directement échangés entre les fils d'exécution cibles. Les messages sont échangés via le répartiteur.

### 2.3.2 Les aides visuelles

Afin d'améliorer la compréhension de l'univers virtuel par les utilisateurs, nous avons choisi d'utiliser une aide visuelle simple. Un fantôme des objets distants est affiché dans le rendu graphique Figure 2.7. Ce principe de fantôme [Gut01] ou d'écho [EZ05] permet d'une part aux utilisateurs de constater le retard présent sur le réseau par l'écart en position entre une instance locale d'un objet partagé et son fantôme issu des autres simulations et d'autre part de visualiser d'éventuelles incohérences.



**Figure 2.7** – Chaque objet du nœud  $i$  possède un fantôme indiquant la position de l'instance de l'objet dans le nœud  $j$

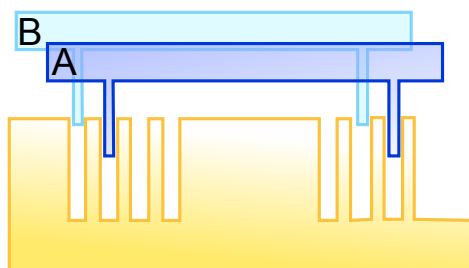
L'ajout de ces fantômes permet ainsi aux utilisateurs d'adapter leur comportement à la situation. En présence de retards importants, les utilisateurs peuvent ainsi contrôler que la divergence entre les états des différentes simulations ne croît pas dans de trop grande proportion et ainsi limiter l'apparition d'incohérences.

## 2.4 Divergence et cohérence

En utilisant une architecture distribuée, les simulations en chaque nœud ont de fortes chances d'être dans des états différents. La divergence entre les états de plusieurs nœuds peut être limitée. L'asservissement des différentes instances des objets partagés entre les nœuds permet un suivi en position et en vitesse à l'instar des asservissements utilisés en téléopération. Ces asservissements peuvent garantir que les instances d'un objet partagé convergent vers une même position au repos. Si l'asservissement ne permet pas cette convergence au repos,

l'univers virtuel devient alors incohérent. L'utilisation directe des transformations en variables d'onde sont un exemple de ce dernier cas.

Lorsque deux états de deux simulations divergent, des incohérences peuvent apparaître. Une incohérence peut être définie comme le fait que deux états divergeant ne puissent pas converger naturellement. Une telle incohérence peut être due à la présence d'un objet entre deux instances d'un objet partagé (voir Figure 2.8). Ainsi, plus les états des simulations constitutives d'une plateforme distribuée divergent plus la probabilité d'occurrence d'une incohérence peut apparaître (voir 2.8).



**Figure 2.8** – les deux objets A et B ne peuvent pas converger uniquement avec un correcteur proportionnel dérivé à cause de la présence de l'objet jaune.

Nous décrirons dans la section suivante l'importance du choix du point de couplage par rapport au point de préhension. Ensuite, nous proposerons une méthode permettant de signaler aux utilisateurs l'existence d'incohérence dans l'univers virtuel puisque nous avons choisi de les tolérer.

### 2.4.1 Point de préhension et point de couplage

La convergence en position et en vitesse est garantie par l'effort de couplage  $W_{C_{i,j}}^{C_i}$  dans l'Equation 2.2. Le choix du point d'application de cet effort par rapport au point de préhension des utilisateurs a une influence importante sur la divergence en position entre deux objets répliqués, avec des effets différents en rotation et en translation. Dans les deux parties suivantes, nous présenterons les effets du choix du point de couplage pour un cas où un utilisateur applique un effort  $F$  et le cas où un utilisateur souhaite faire pivoter un objet autour du point de préhension.

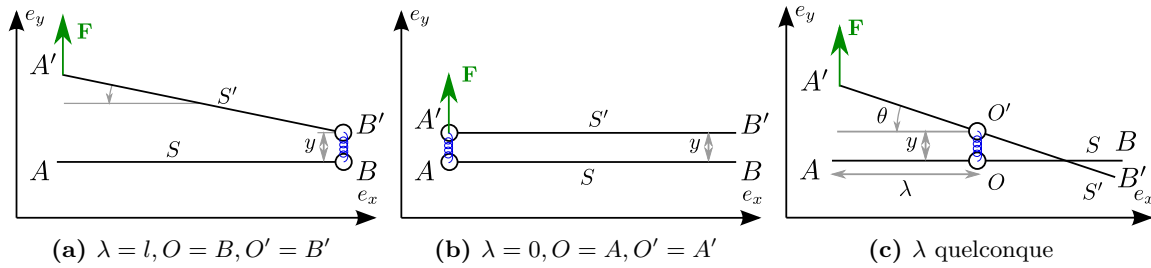
Les différents cas sont présentés avec un asservissement proportionnel à l'équilibre. Ce modèle permet également de prendre en compte la transformation en variables d'onde car elle possède à l'équilibre le comportement d'un ressort (voir Equation 1.49).

#### 2.4.1.1 Cas des translations et définition d'une mesure de la divergence

Pour le cas des translations, considérons une tige de longueur  $l$  répliquée entre 2 nœuds en 2D (voir Figure 2.9) où le couplage est réalisé par un ressort amortisseur de raideur  $k_t$  en translation et  $k_r$  en rotation.

Sur le premier nœud, un utilisateur applique un effort  $\mathbf{F}$  parallèlement à la tige au repos. La tige sur le second nœud est fixe. A l'équilibre statique, les vitesses sont nulles et le système peut s'écrire au point d'application de l'effort :

$$\begin{aligned} F_x &= 0 \\ F_y &= k_t y \\ k_r \theta &= \lambda \cos \theta k_t y \end{aligned} \quad (2.4)$$



**Figure 2.9** – Les différents choix du lieu couplage pour un ressort amortisseur à 6 degrés de liberté dans le cas d'une force appliquée à une extrémité.

où  $y = \overline{OO'}$  est l'écart en position selon l'axe  $e_y$  entre les deux points de couplages. L'écart suivant  $e_x$  est nul, car il n'y a pas d'effort selon cet axe.  $\theta$  est l'angle entre les deux tiges et  $\lambda$ , la distance entre le point d'application de l'effort et le lieu du couplage. L'angle  $\theta$  est défini sur  $[0; \frac{\pi}{2}]$ .

$$y = \frac{k_t}{F} \quad (2.5)$$

$$\frac{\theta}{\cos \theta} = \frac{\lambda F}{k_r} \quad (2.6)$$

La fonction  $\frac{\theta}{\cos \theta}$  est monotone et croissante sur  $[0; \frac{\pi}{2}]$ . Donc, l'angle  $\theta$  est d'autant plus grand que le bras de levier  $\lambda$  est important. La raideur  $k_r$  peut être augmentée pour compenser cette divergence, mais pour des problèmes de stabilité, elle est bornée par une valeur maximale (voir chapitre 3).

Une mesure de la divergence peut être calculée en mesurant l'ensemble des distances entre les points d'un objet répliqué. Il faut pour cela définir la distance séparant deux tiges. Soit l'espace  $E$  des segments de longueur  $l$  :

$$E = \{(a, b) \in \mathbb{C}, |a - b| = l\} \quad (2.7)$$

Une paramétrisation triviale d'un segment  $S \in E$  peut être définie par :

$$E = \{p_s(t) = ta + (1 - t)b, t \in [0; 1]\} \quad (2.8)$$

Une distance entre deux segments  $S, S' \in E$  permettant de mesurer la divergence peut être définie :

$$d^2(S, S') = \int_0^1 |p_s(t) - p_{s'}(t)|^2 dt \quad (2.9)$$

Un point  $C'$  de la tige répliquée  $S'$  est obtenu par une isométrie  $f$  du plan à partir d'un point  $C$  du segment  $S$ .

$$f: \mathbb{C} \rightarrow \mathbb{C} \\ x \mapsto v + xr \quad (2.10)$$

$$v = a + ib \in \mathbb{C} \text{ et } r = e^{i\theta} \in \mathbb{U}$$

$$a, b, \theta \in \mathbb{R}$$

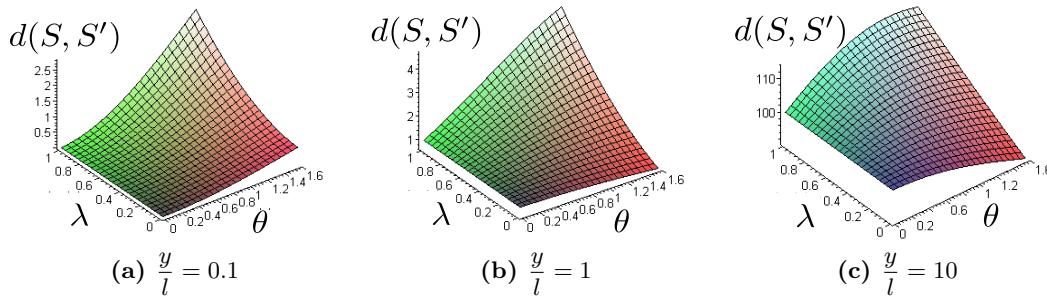


Figure 2.10 – Mesure de la divergence suivant  $\lambda$  et  $\theta$  pour différentes valeurs de  $y$

Dans le cas de la Figure 2.9, on considère la tige en position initiale selon l'axe des réels. Les caractéristiques de la transformation  $v = iy$  et  $r = e^{i\theta}$ . La mesure de la divergence vaut donc :

$$d^2(S, S') = \frac{2}{3}l^3(1 - \cos \theta) - yl^2(1 - \cos \theta) + l \left( (y + \lambda \sin^2 \theta + \lambda^2(1 - \cos \theta))^2 \right) \quad (2.11)$$

$$f_d: \mathbb{R}^+ \times [0; \pi] \times \mathbb{R}^+ \rightarrow [0; l]$$

$$(l, \theta, y, \lambda) \mapsto d^2(S, S')$$

où  $f_d$  est l'application qui associe à la configuration de la préhension une mesure de la divergence. Afin d'étudier l'évolution de cette divergence, le gradient de  $f_d$  peut être calculé :

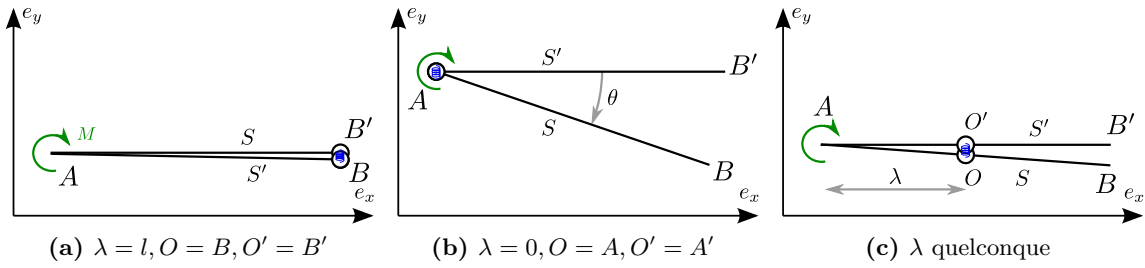
$$\vec{\nabla} f_d(l, \theta, y, \lambda) = \begin{pmatrix} 2l^2(1 - \cos \theta) - 2yl(1 - \cos \theta) + ((y + \lambda \sin^2 \theta + \lambda^2(1 - \cos \theta))^2) \\ \arctan \left( \frac{-6\lambda y}{-\lambda^2 63yl + 2l^2} \right) \\ -l^2(1 - \cos \theta) + 2ly + 2l\lambda \sin \theta \\ 2ly \sin \theta + 4\lambda l(1 - \cos \theta) \end{pmatrix} \quad (2.12)$$

Le premier polynôme n'admet pas de racine réelle sauf en 0. La divergence est donc monotone en fonction  $l$ . Elle est en outre intuitivement croissante. Plus la tige est longue plus la divergence est importante pour un déplacement donné.

Le terme sur  $\lambda$  est strictement positif. La divergence est donc minimale lorsque le point de préhension et le point de couplage sont confondus. Le terme sur  $y$  s'annule en  $y = \frac{l(1 - \cos \theta) - 2\lambda \sin \theta}{2}$ , soit pour  $\theta$  et  $\lambda$  nul, donc également lorsque le point de couplage et le point de préhension sont confondus. Le terme sur  $\theta$  change de signe pour  $y = \frac{6\lambda^2 + 2l^2}{l}$ . Donc pour une grande valeur  $y$  devant  $l$  et  $\lambda$ . La fonction  $f_d$  est tracée en fonction de  $\lambda$  et  $\theta$  pour différentes valeurs de  $y$  sur la Figure 2.10.

Pour  $\theta$  donné, la divergence est minimale lorsque  $\lambda$  est nul. Le lieu idéal du couplage est donc le lieu du point d'application de l'effort. Ensuite, la divergence, avec cette mesure de distance, diminue pour des valeurs de  $y$  importantes devant  $l$  dès lors que  $\theta$  croît. Une grande valeur de  $y$  ou de  $\theta$  pour une force  $\mathbf{F}$  donnée correspond à des gains  $k_t$  ou  $k_r$  faibles.

Une autre mesure de la divergence possible est la distance maximale entre deux points de deux objets répliqués,  $d_{\max}(S, S')$ . Dans le cas des tiges, cette mesure correspond à la taille de l'objet le plus grand pouvant s'insérer entre les deux objets répliqués. Plus cette distance



**Figure 2.11** – Différents points d’application du point de couplage pour un ressort amortisseur 6D dans le cas d’un utilisateur effectuant une rotation.

est faible, plus le nombre d’objets de la scène pouvant provoquer une incohérence est faible. Dans le cas de la tige :

$$d_{\max}(S, S') = \max_{u \in [0;1]} \left| 2p_s(u)^2(1 - \cos \theta) - 2p_s(u)y(1 - \cos \theta) + (y + \lambda \sin \theta)^2 + \lambda^2(1 - \cos \theta)^2 \right| \quad (2.13)$$

Cette distance admet un minimum pour  $p_s(u) = \frac{y}{2}$ . Le maximum est soit atteint pour  $u = 0$  où :

$$d_{\max}(S, S') = \left| (y + \lambda \sin \theta)^2 + \lambda^2(1 - \cos \theta)^2 \right| \quad (2.14)$$

soit pour  $u = 1$  où :

$$d_{\max}(S, S') = \left| 2l(1 - \cos \theta)(l - y) + y^2 + \lambda^2(1 - \cos \theta)^2 + \lambda \sin \theta(y + \lambda \sin \theta) \right| \quad (2.15)$$

Dans le premier cas, la distance est minimale pour  $\lambda$  nul. Dans le deuxième cas, le polynôme  $2l(1 - \cos \theta)(l - y) + y^2$  est positif sur son domaine de définition, donc la distance  $d_{\max}(S, S')$  est minimale si  $\lambda$  est nul.

Outre le fait d’augmenter la divergence selon les deux distances proposées au dessus, le fait de choisir  $\lambda > 0$  introduit un centre de rotation dans le mouvement de la tige soumise à l’effort  $\mathbf{F}$ . Pour  $\mathbf{F}$  donné la distance  $y$  est constante alors que l’angle  $\theta$  diffère en fonction du point de couplage (voir Figure 2.9). Si l’utilisateur applique uniquement une force, la tige répliquée pivote autour du point  $O'$ . Dans le cas où il est seul à manipuler une tige sans contrainte extérieure, ce mouvement de rotation ne correspond pas à ce que le même utilisateur obtiendrait en réalité ou dans le cas d’un univers virtuel à un seul nœud. Ce comportement peut être très déconcertant car il ne correspond pas à une situation réelle.

#### 2.4.1.2 Cas des rotations

En prenant le cas similaire des translations, un utilisateur déplace en rotation la tige répliquée autour de  $A$  (voir Figure 2.11). A l’équilibre statique :

$$F_x = k_t \lambda (1 - \cos \theta) \quad (2.16)$$

$$F_y = k_t \lambda \sin \theta \quad (2.17)$$

$$M = k_r \theta + k_t \lambda^2 \sin \theta \quad (2.18)$$

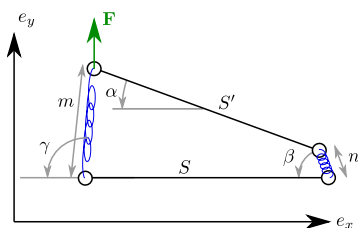


Figure 2.12 – Exemple de deux couplages en  $A, A'$  et  $B, B'$

L'utilisateur doit alors appliquer un moment  $M$  et un effort  $\mathbf{F}$  si le point de couplage diffère du point d'application de l'effort,  $O \neq A$ . En outre, les fonctions  $\theta \mapsto \theta$  et  $\theta \mapsto \sin \theta$  sont toutes deux strictement croissantes sur  $[0, \frac{\pi}{2}]$ , L'angle  $\theta$  est donc d'autant plus faible que  $\lambda$  est grand pour  $M$  donné. En utilisant les deux mesures définies plus haut, on obtient :

$$d^2(S, S') = \frac{\theta l^2}{2} \quad (2.19)$$

$$d_{\max} = 2l \sin \frac{\theta}{2} \quad (2.20)$$

$$(2.21)$$

Dans les deux cas, la divergence est d'autant plus faible que l'angle  $\theta$  est petit. Pour une rotation, plus la distance du point de préhension au point de couplage est importante, plus la divergence est faible, contrairement au cas de la translation.

Il faut donc établir un compromis lors de la manipulation à un seul utilisateur pour le choix du point de couplage entre la divergence tolérée pour la rotation et la translation, et les effets induits tels les centres de rotation au court d'un mouvement. Un bon compromis est de manipuler la pièce en son barycentre et de réaliser le couplage en ce point également.

### 2.4.2 Point de couplage et manipulation multi-utilisateurs

Dans un cas d'une manipulation collaborative, le lieu du couplage est le même pour chacun des éléments répliqués d'un objet. En utilisant un seul couplage, si les utilisateurs ne saisissent pas l'objet au même point, il y a forcément des moments induit par les couplages.

Il peut être intéressant d'utiliser plusieurs couplages en les plaçant aux différents points de manipulation (voir Figure 2.12). Dans ce cas, comme le rôle des différents utilisateurs est supposé équivalent, les gains des différents couplages sont égaux. En reprenant le cas des tiges de la section précédente, si les tiges sont couplées aux points  $A, A'$  et  $B, B'$ , le système est régi par le système d'équation suivant :

$$0 = k_t(m \cos \gamma + n \cos \beta) \quad (2.22)$$

$$F = k_r(m \sin \gamma + n \sin \beta) \quad (2.23)$$

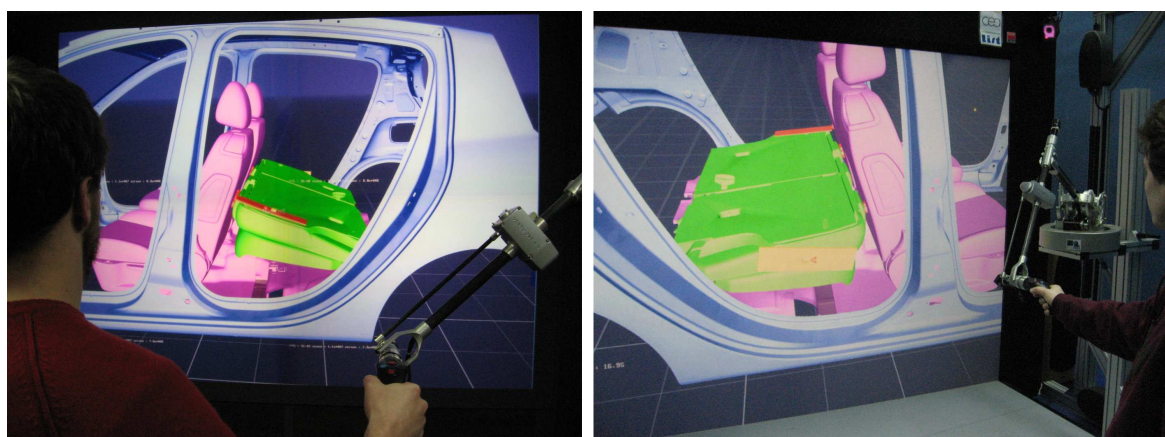
$$0 = 2k_r \alpha - \sin(\beta - \alpha)nk_t l \quad (2.24)$$

Le moment  $-\sin(\beta - \alpha)nk_t l$  induit par le couplage  $B, B'$  empêche ces deux points de s'éloigner l'un de l'autre. L'utilisateur appliquant un effort  $\mathbf{F}$  à l'extrémité  $A$  de la tige est donc grossièrement contraint sur un cercle dont le centre de rotation se situe autour des points  $B$  et  $B'$ .



Dans le cas où un second utilisateur manipule la tige au point  $B$ , ce comportement est cohérent. L'utilisateur au point  $A$  ne peut facilement déplacer le point  $B$  sans l'aide du second utilisateur, sauf à appliquer un moment d'autant plus important que la distance  $l$  entre les points de préhension est grande. Dans un cas réel, lorsque deux utilisateurs manipulent une poutre  $AB$  de longueur  $l$ , le point de préhension d'un des utilisateurs contraint le mouvement du second utilisateur. Si ce second utilisateur maintient son point de préhension  $B$  fixe, le premier utilisateur ne peut que faire décrire à son point de préhension  $A$ , un cercle de longueur  $l$  autour du point de préhension  $B$  du premier utilisateur.

Les deux ressorts linéaires aux points  $A, A'$  et  $B, B'$  participent aussi à la limitation de la divergence en rotation comme c'est le cas pour les rotations dans le cas mono-utilisateur, lorsque le point de couplage est choisi loin du point de préhension (voir Figure 2.11). Cependant, le fait de rajouter des couplages virtuels a un effet sur la stabilité du système. L'ajout de points de couplage est similaire au fait de rajouter des couplages virtuels en parallèle, ce qui augmente globalement la raideur du système asservi et peut le rendre instable. La divergence en position n'est donc pas nécessairement plus faible avec plus de couplages virtuels, mais dans le cas de multiple point de préhension, elle est plus naturelle car elle évite l'apparition de point de rotation lors du mouvement.



**Figure 2.13** – Chaque utilisateur manipule l'objet à une des extrémités (pièces rouge et jaune respectivement). Un asservissement est placé à chaque point de préhension.

### 2.4.3 Gestion de la cohérence

Nous avons choisi de tolérer les incohérences dans notre plateforme tout en les présentant aux utilisateurs de manière explicite afin qu'ils puissent choisir la meilleure stratégie à adopter. Ces incohérences peuvent être présentées aux utilisateurs sous la forme d'aides visuelles.

L'utilisation supplémentaire de fantôme dans l'environnement virtuel permet de donner une indication de la position des objets aux utilisateurs. D'une manière similaire, nous avons choisi de présenter les incohérences potentielles en indiquant aux utilisateurs la présence éventuelle d'objets entre deux instances d'un objet partagé. Nous laissons ainsi le soin aux utilisateurs d'intégrer cette information pour continuer la manipulation ou pour résoudre l'incohérence avant de continuer la manipulation. Le principe est de détecter la présence d'objets pouvant se trouver éventuellement entre les deux instances de l'objet partagé le long de la trajectoire reliant les deux instances de cet objet. Ce principe est similaire à celui permettant de déterminer le moment d'un contact à l'aide d'un détecteur de collision continue entre deux pas de temps. Au lieu de considérer un objet à deux pas de temps distinct, nous

considérons deux instances d'un objet partagé. Ensuite, il s'agit de détecter une collision avec d'autres objets lors du parcours de la trajectoire reliant ces deux instances.

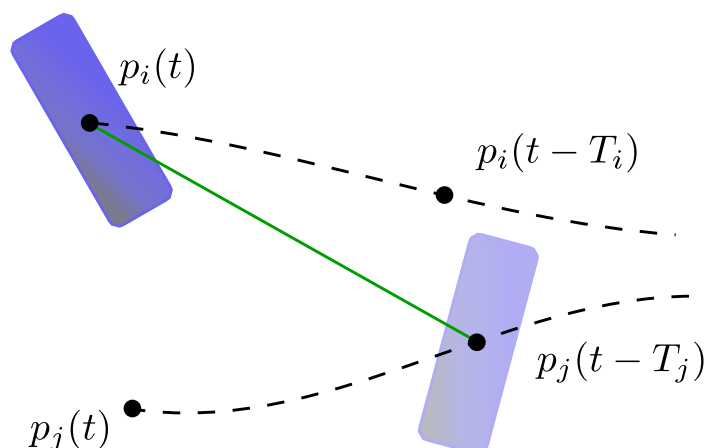


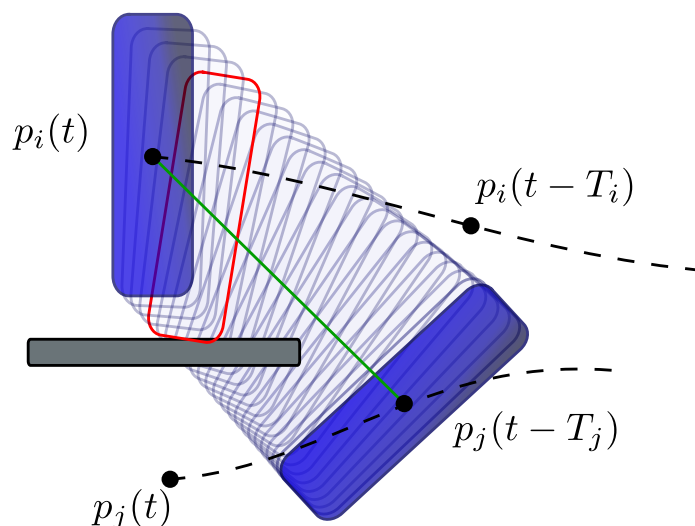
Figure 2.14 – Exemple de trajectoire de deux instances d'un objet répliqué

La trajectoire permettant de relier les deux instances n'a pas de réalité physique. Parmi l'infinité de trajectoire reliant ces deux positions, il en faut une le long de laquelle la détection de collision pourra être réalisée. La trajectoire la plus simple et la plus intuitive reliant les deux objets est la droite passant par les positions des deux instances. Cette trajectoire correspond à celle que parcourraient les instances ayant une vitesse initiale nulle, s'ils pouvaient converger l'un vers l'autre grâce à un asservissement proportionnel dérivé.

La trajectoire à considérer relie donc  $p_i(t)$  et  $p_j(t)$ , les positions des instances  $i$  et  $j$  de l'objet partagé dans les nœuds  $i$  et  $j$  au temps  $t$ . Or, le nœud  $i$  n'a connaissance que de  $p_i(t)$  et  $p_j(t - T_j)$ , cette dernière position n'étant pas forcément prédictible au temps  $t$  à cause des contacts entre objets ou du mouvement de l'opérateur distant entre  $t$  et  $t - T_j$  (voir Figure 2.14). Cependant dans un environnement encombré les utilisateurs ont tendance à déplacer les objets doucement, donc la position  $p_j(t - T_j)$  peut être considérée comme une approximation grossière de  $p_j(t)$ . Nous avons donc retenu la trajectoire  $p_i(t)p_j(t - T_j)$  comme trajectoire permettant de tester d'éventuelles incohérences.

Plusieurs choix peuvent être adoptés pour détecter la présence d'objet le long de cette trajectoire. Les algorithmes de détection continue sont de bons candidats pour cette tâche. Néanmoins, d'un point de vue logiciel il est plus pertinent d'utiliser le même algorithme pour cette tâche que pour la détection de collision réalisée par le moteur physique. En effet, les objets sont déjà modélisés, ce qui évite une trop grande consommation de mémoire en particulier pour de grandes scènes. La parallélisation des différentes boucles s'en trouve également simplifiée. En pratique, nous utilisons un moteur de collision discret de type LMD ou VPS (voir Section 1.1.1). Afin d'utiliser cet algorithme, il faut discrétiser la trajectoire et effectuer un test sur chacune des positions de la trajectoire discrétisées. Ce principe est équivalent au *backtrack* effectué pour détecter l'instant de la collision avec des moteurs de collision discret [MW88]. En pratique, un test uniquement d'intersection plutôt que de collision permet de fournir des résultats suffisant pour indiquer la présence d'un objet sur la trajectoire (voir Figure 2.15). Le test d'intersection est beaucoup plus rapide car l'information de la position précise du contact n'a pas besoin d'être extraite.

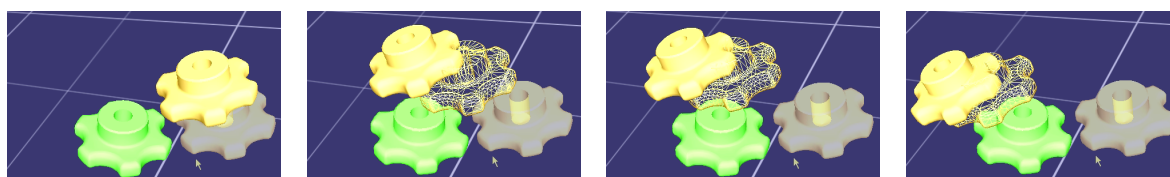
En pratique le fil d'exécution chargé de cette détection est ajouté à l'application aux cotés des autres fils d'exécution (voir Figure 2.5). Cette tâche n'a pas besoin d'une fréquence de rafraîchissement rapide. En pratique; elle est cadencée à 2 Hz, ce qui permet de laisser la majorité de la puissance de calcul au reste du moteur physique. Un exemple d'aide visuelle



**Figure 2.15** – Exemple de discrétisation de la trajectoire et de détection d'incohérence

utilisée pour rendre compte de l'incohérence est présentée Figure 2.16. Dans cet exemple, la première position fournissant une intersection avec un objet de la scène est affichée en fils de fer.

Dans le cas où la scène comporte de multiples objets mobiles, chacune des trajectoires des objets mobiles est discrétisée et possède le même nombre d'éléments. Chacune des trajectoires n'est donc pas discrétisée avec le même pas. Afin d'être suffisamment précis, le pas de discrétisation est basé sur la trajectoire la plus longue.



**Figure 2.16** – Implémentation de la détection d'incohérence

## Conclusion

Dans ce chapitre nous avons proposé une architecture de plateforme collaborative permettant de simuler une tâche de prototypage virtuel. L'architecture adoptée est une architecture distribuée qui permet une bonne réactivité locale et peut être facilement convertie en architecture hybride (il suffit d'utiliser un seul simulateur à la fois et de faire *migrer* la simulation en activant un autre simulateur et désactivant le précédent) Cette architecture est constituée de différents nœuds hébergeant chacun un moteur physique. Ces moteurs physiques sont très similaires à ceux rencontrés dans le cas de tâche de prototypage virtuel non collaboratif. L'architecture logicielle permettant à ces différents moteurs physiques d'échanger leurs états a été présentée.

Nous avons fait le choix de ne pas contrôler la divergence entre les états des différents moteurs physiques autrement que par les efforts d'asservissements reliant les différentes instances des objets partagés. L'importance du lieu de couplage entre ces instances par rapport aux points de préhension a été en particulier soulignée. Nous pouvons distinguer d'une part ce qui concerne son influence sur la divergence entre les instances et d'autre part l'appari-

tion de centres de rotation lors de la manipulation. Enfin, nous avons présenté une méthode basée sur les détecteurs de collision permettant de souligner l'apparition d'incohérences dans l'univers virtuel partagé.

L'implémentation de cette architecture constitue la base des évaluations des deux asservissements proposés dans les chapitres 3 et 4.

### Perspectives

Afin de pouvoir simuler une plus grande variété de tâches de prototypage, il serait intéressant d'étudier la simulation d'objets polyarticulés dans une telle architecture, particulièrement en ce qui concerne les problématiques de divergence et d'incohérence. Afin de partager un objet polyarticulé, il faudrait remplacer l'asservissement d'un point quelconque par un asservissement articulaire entre les instances de l'objet polyarticulé. Ce type d'asservissement pourrait permettre de limiter de manière efficace la divergence entre les instances des objets polyarticulés.

D'un point de vue expérimental, nous avons essayé cette plateforme uniquement dans le cas d'une collaboration à deux personnes et donc uniquement avec deux nœuds. Il serait intéressant d'évaluer cette plateforme avec un plus grand nombre de nœuds et d'utilisateurs afin d'étudier les problèmes de collaboration sur une tâche nécessitant la coordination de plusieurs personnes.



# Asservissement par couplage virtuel proportionnel dérivé

## Sommaire

<b>3.1 Couplage Proportionnel-Dérivé et retour haptique distant</b> . . . . .	<b>68</b>
3.1.1 Motivation . . . . .	68
3.1.2 Couplage en 6 degré de liberté . . . . .	69
<b>3.2 Analyse en 1 degré de liberté</b> . . . . .	<b>70</b>
3.2.1 Les différents cas de manipulation . . . . .	70
3.2.2 Formulation discrète . . . . .	73
3.2.3 Matrices d'immittance des différents cas de manipulations . . . . .	73
3.2.4 Étude de stabilité . . . . .	76
3.2.5 Efforts de traînée . . . . .	78
<b>3.3 Compensation des efforts de traînée</b> . . . . .	<b>81</b>
3.3.1 Boucle de compensation . . . . .	81
3.3.2 Effet de la boucle de compensation . . . . .	83
<b>Conclusion</b> . . . . .	<b>85</b>

UNE architecture distribuée comme celle présentée au chapitre précédent implique l'utilisation de plusieurs moteurs physique et donc de plusieurs instances d'un même objet virtuel. Ces différentes instances doivent donc être asservies entre elles. Parmi les méthodes d'asservissements issues de la téléopération (Section 1.3), nous avons choisi, dans le cas de retards faibles d'étudier les asservissements proportionnels dérivés et leur influence en termes d'efforts sur la manipulation. Ce type d'asservissement a l'avantage, par rapport aux variables d'onde ou aux observateurs de passivité, de ne pas introduire de viscosités ni d'inerties supplémentaires.

Cependant, l'asservissement proportionnel-dérivé entre une position d'une instance et une position retardée d'une autre instance introduit un effort. Celui-ci est ressenti par les utilisateurs, car il s'oppose aux mouvements tendant à séparer les différentes instances. Nous proposons une manière de compenser ces efforts, appelés efforts de *traînée*, lors de mouvements libres.

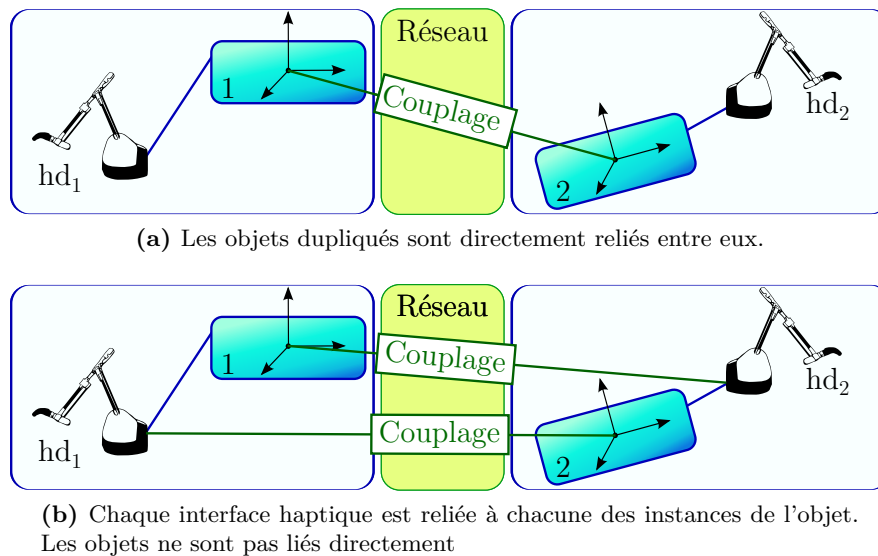
Ce chapitre présente dans une première section (Section 3.1) les motivations nous ayant conduit à privilégier une architecture distribuée. La formulation de l'asservissement à 6 degrés de liberté est également présentée. La section suivante (Section 3.2) étudie la stabilité du problème au travers du cas simplifié à 1 degré de liberté aux travers des cas de manipulations les plus couramment répandus. Enfin, dans la dernière partie (Section 3.3) nous proposons d'ajouter une boucle de compensation pour compenser les efforts de traînée. Les différents résultats issus de l'analyse de stabilité sont en parallèle utilisés pour régler la plateforme de collaboration à 6 degrés de liberté. Les résultats en termes de suivi et d'effort en 6 degré de liberté sont ensuite présentés sur des cas de manipulations simples.

### 3.1 Couplage Proportionnel-Dérivé et retour haptique distant

#### 3.1.1 Motivation

Le domaine de la téléopération offre plusieurs approches permettant de stabiliser un système avec retard (voir 1.3). Parmi ceux-ci, Sankaranarayanan et Hannaford [SH06] ont testé différents schémas de couplages, les schémas ressort-amortisseur, ressort-amortisseur avec observateur et correcteur de passivité ainsi que les transformations en variables d'onde dans un cas de téléopération. Les auteurs ont étudié l'écart en position d'un objet répliqué entre deux nœuds lors de la manipulation collaborative d'un cube en un degré de liberté ainsi que les efforts générés par le couplage. Le schéma ressort-amortisseur est celui qui a minimisé les écarts en positions, alors que le schéma en variables d'onde assure un meilleur suivi en effort. Pour ce dernier, l'amortissement  $b$  de la transformation était faible, favorisant de fait les efforts par rapport au suivi en position (voir Section 1.3.3). L'ajout d'un observateur de passivité augmente la divergence entre les instances des objets répliqués mais permet de garantir la stabilité du système lors de retards variables.

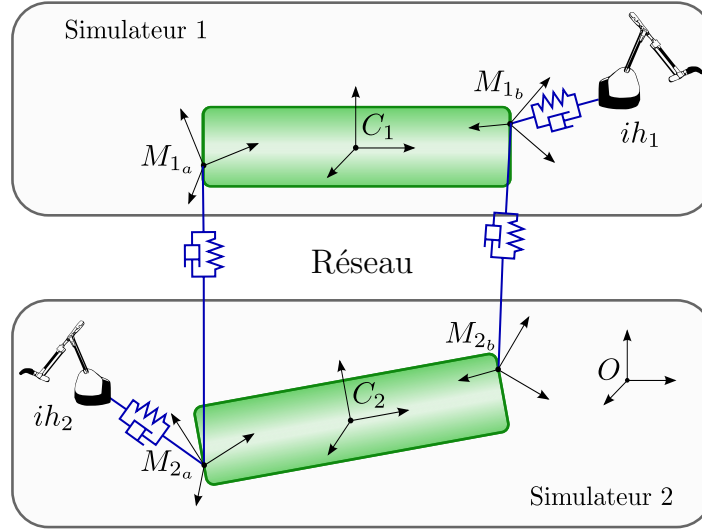
Les mêmes auteurs ont ensuite étudié l'influence de l'architecture sur le suivi en position des objets répliqués [SH08a, SH08b]. Pour une architecture distribuée avec un schéma ressort-amortisseur il y a deux choix principaux, le premier consiste à relier chaque interface haptique à un objet local partagé. Les deux instances de l'objet partagé sont liées via un ressort amortisseur (Figure 3.1a). Le second type de couplage consiste à lier chaque interface haptique aux deux instances de l'objet virtuel partagé. Il n'y a dans ce dernier cas pas d'asservissement direct entre les instances de l'objet partagé (Figure 3.1b).



**Figure 3.1** – L'asservissement proportionnel dérivé peut être mis en œuvre de deux façons : uniquement au niveau des objets virtuels ou entre les interfaces (issu de [SH08b]).

Le schéma où les deux objets répliqués sont asservis en position (Figure 3.1a) est celui où la divergence est la moins importante [SH08a]. Ce schéma est donc celui pour lequel les incohérences ont le moins de chance d'apparaître indépendamment d'un éventuel mécanisme permettant de contrôler la convergence. Ce résultat est du à l'asservissement présent entre les deux instances à la différence de la seconde architecture.

Les schémas ressort-amortisseurs entre deux instances d'un objet partagé sans contrôleur de passivité sont donc les plus performants sur le suivi en position. Néanmoins, la stabilité de ce schéma dépend du délai. En téléopération, les schémas utilisant des couplages virtuels sont



**Figure 3.2** – Manipulation d'un objet entre deux nœuds. L'objet partagé est répliqué en deux instances sur chacun des deux nœuds 1 et 2. Les deux utilisateurs manipulent ces instances aux positions  $M_{2a}$  pour l'utilisateur  $a$  et  $M_{1b}$  pour le second utilisateur  $b$ . Les deux instances de l'objet partagé sont liées par un système ressort-amortisseur à chacun des points de préhension.

de moins en moins stables avec un retard croissant [HZS01]. Nous montrerons lors de l'analyse du système que notre architecture à également ce comportement. Nous avons donc, dans le cas de retard relativement faible, étudié les schémas ressort-amortisseurs pour la plateforme de collaboration.

### 3.1.2 Couplage en 6 degré de liberté

Le ressort-amortisseur permettant de lier deux objets répliqués génère un effort permettant l'asservissement en position des objets entre eux. Dans l'Equation 2.2, le terme de couplage entre l'objet  $i$  et l'objet répliqué  $j$ ,  $W_{C_i,j}^{C_i}$  correspond au torseur d'effort du couplage. Il est exprimé à partir de l'Equation 2.3 par :

$$W_{C_i,j}^{C_i} = W_{\text{ressort}}(H_{C_j(t-T_j)}^O, H_{C_i(t)}^O, K_{cv}) + B_{cv} \left( T_{C_j(t-T_j)}^{C_i(t),O} - T_{C_i(t)}^{C_i(t),O} \right) \quad (3.1)$$

L'expression de cet asservissement est le même que celui entre l'interface haptique et le moteur physique (Equation 2.3). L'asservissement est réalisé entre d'une part la position  $H_{C_i}^O(t)$  et la vitesse  $T_{C_i(t)}^{C_i(t),O}$  de l'instance locale  $i$  et d'autre part la position retardée  $H_{C_j(t-T_j)}^O$  et la vitesse retardée  $T_{C_j(t-T_j)}^{C_i(t),O}$  de l'instance distante  $j$ . Les gains du couplage virtuel  $K_{cv}$ ,  $B_{cv}$  sont définis comme précédemment, c'est à dire diagonaux.

Dans l'Equation 2.2, un seul couplage est considéré entre l'objet  $i$  et l'objet  $j$ . Afin de réduire les écarts en rotation, particulièrement lorsque les points de préhension sont éloignés, il peut être plus pertinent d'utiliser plusieurs points de couplages comme évoqué dans la Section 2.4.2 (voir Figure 3.2). Le terme  $W_{C_i,j}^{C_i}$  de l'Equation 3.1 devient donc :

$$W_{C_i,j}^{C_i} = \sum_l \text{Ad}_{H_{C_i}^{M_i}}^T W_{M_{il},j}^{M_i} \quad (3.2)$$

où  $l$  est successivement l'ensemble des points où s'appliquent chacun des asservissements proportionnels dérivés. La stabilité du système est modifiée en conséquence. En effet les gains globaux de l'asservissement équivalent entre les différents objets augmentent avec le nombre



d'asservissements. De plus, l'asservissement à 6 degrés de liberté introduit des bras de levier qui impactent également la stabilité du système.

## 3.2 Analyse en 1 degré de liberté

Afin de simplifier l'analyse nous considérons le cas à 1 degré de liberté avec 2 nœuds (voir Figure 3.3) où deux utilisateurs peuvent interagir. Les domaines de stabilité obtenus par cette analyse ont servi de base aux réglages expérimentaux de la plateforme dans le cas à 6 degrés de liberté.

### 3.2.1 Les différents cas de manipulation

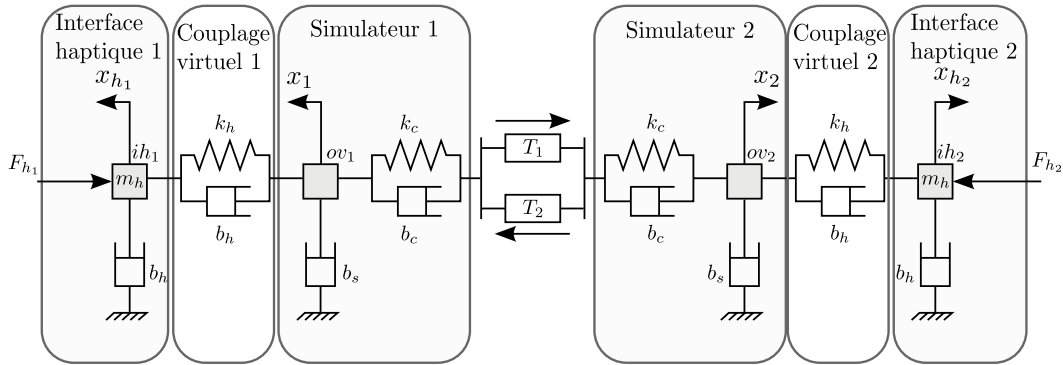


Figure 3.3 – Manipulation à deux personnes d'un objet répliqué en espace libre

Un objet virtuel ( $ov$ ) est dupliqué sur les deux nœuds en deux instances,  $ov_1$  sur le nœud 1 et  $ov_2$  sur le second nœud. La formulation adoptée étant quasi-statique (voir Section 2.1.2 et Equation 2.2), ils n'ont pas de masse. Le système est considéré comme symétrique, avec les deux mêmes interfaces haptiques. Les valeurs des couplages entre les simulations et les interfaces haptiques sont égales sur chacun des simulateurs et ont pour raideur  $k_h$  et pour amortissement  $b_h$ . Les couplages entre les objets répliqués sont également égaux et ont pour gains  $k_c$  et  $b_c$ . Ce schéma est équivalent à celui proposé par Sankaranarayanan et Hannaford [SH08b] (voir Figure 3.1a) aux termes de masse près.

Le mouvement des objets virtuels est donc régi par les équations suivantes dans le domaine continu :

$$\begin{cases} b_s \dot{x}_1(t) = b_c (\dot{x}_2(t - T_2) - \dot{x}_1(t)) + k_c (x_2(t - T_2) - x_1(t)) \\ \quad + b_h (\dot{x}_{h1}(t) - \dot{x}_1(t)) + k_h (x_{h1}(t) - x_1(t)) \\ b_s \dot{x}_2(t) = b_c (\dot{x}_1(t - T_1) - \dot{x}_2(t)) + k_c (x_1(t - T_1) - x_2(t)) \\ \quad + b_h (\dot{x}_{h2}(t) - \dot{x}_2(t)) + k_h (x_{h2}(t) - x_2(t)) \\ m_h \ddot{x}_{h1}(t) = b_h (\dot{x}_1(t) - \dot{x}_{h1}(t)) + k_h (x_1(t) - x_{h1}(t)) + F_{h1}(t) \\ m_h \ddot{x}_{h2}(t) = b_h (\dot{x}_2(t) - \dot{x}_{h2}(t)) + k_h (x_2(t) - x_{h2}(t)) + F_{h2}(t) \end{cases} \quad (3.3)$$

avec  $x_1, \dot{x}_1, x_2, \dot{x}_2$  les positions et vitesses de l'objet répliqués au nœud 1 et 2. ,  $x_{h1}, \dot{x}_{h1}, x_{h2}, \dot{x}_{h2}$ , les positions et vitesses des interfaces haptiques. Les interfaces haptiques sont modélisées par une masse  $m_h$  amortie par une viscosité  $b_h$ .  $b_s$  est un une viscosité (voir la formulation quasi-statique Equation 2.2) et  $b_c, k_c, b_h, b_h$  sont les termes de raideur et d'amortissements des couplages virtuels entre d'une part l'interface haptique et la simulation

et d'autre part les simulations entre elle.

Dans le cas d'une tâche collaborative, la comanipulation d'un objet dans un espace libre est seulement un des cas possibles. Les différents cas de manipulations sont modélisés par différentes configurations de la plateforme et possèdent donc des domaines de stabilité différents les uns des autres. C'est pourquoi, nous considérons les différents cas pouvant être rencontré à 1 degré de liberté en l'absence d'incohérence. En pratique, sur la plateforme de collaboration, un contrôleur de passivité peut être ajouté afin de prendre en compte les phases transitoires entre les différents cas de manipulation ainsi que les cas issus des incohérences. Cette analyse de stabilité, nous permet par la suite de réaliser une évaluation dans des cas canoniques dans la Section 3.3. En un degré de liberté, les différents cas de manipulations sont les suivants (voir Figure 3.4) :

**Co-manipulation d'un objet** – Deux personnes manipulent deux instances d'un objet partagé dans un espace libre (voir Figure 3.4a). C'est le cas précédemment présenté.

- Deux personnes manipulent deux instances d'un objet partagé qui peut rentrer en contact avec l'environnement sur chacun des nœuds (voir Figure 3.4b).
- Deux personnes manipulent deux instances d'un objet partagé en contact avec l'environnement dans chacun des nœuds (voir Figure 3.4c).

**Manipulation d'un seul objet** – Deux personnes manipulent deux objets différents qui sont en contact entre eux (voir Figure 3.4d).

- Une personne manipule un objet libre (voir Figure 3.4e).
- Une personne manipule un objet en contact avec l'environnement dans le simulateur distant (voir Figure 3.4f).
- Une personne manipule un objet en contact avec l'environnement dans le simulateur local (voir Figure 3.4g).

Les différents cas où l'objet est en contact dans une seule simulation correspondent à des divergences d'état entre les moteurs physiques. Dans des tâches de pointage, particulièrement à une personne, ce genre de divergence à de forte probabilité d'apparaître. En effet, lors d'un mouvement balistique, typique d'une tâche de pointage, la vitesse de déplacement de l'objet peut être élevée ce qui favorise des divergences importantes. A contrario, nous n'avons pas retenu ces cas pour la manipulation de deux objets (Figure 3.4d) car dans ces cas les vitesses de déplacement des objets sont plus faibles et ces cas on moins de chance d'apparaître. Enfin le cas d'une manipulation de deux objets en contact avec l'environnement n'est pas non plus explicité car il est équivalent au cas d'une manipulation locale en contact avec l'environnement sur chacun des nœuds, les objets en contacts ayant une vitesse nulle.

Il y a donc 7 cas à étudier. Cependant, certains cas peuvent être regroupés. Notamment le second cas de la manipulation par deux personnes d'un objet peut être ramené aux deux cas de manipulation simple par un seul utilisateur car au contact la vitesse de l'objet est nulle (voir Figure 3.4b).

Le cas de deux objets manipulés chacun par un utilisateur (voir Figure 3.4d) peut être ramené au cas d'une comanipulation sur un même objet. Il suffit de considérer que les deux objets en contact forment un seul objet équivalent. Le couplage virtuel entre les deux instances de l'objet équivalent possède ainsi des gains deux fois plus élevés que dans le cas de la comanipulation. Ces 7 cas se simplifient donc en 5 cas.

L'étude des différents cas principaux présentés ci-dessus permet de définir un domaine de stabilité pouvant servir de base au réglage des gains des différents couplages virtuels de la plateforme collaborative. Néanmoins, le réglage de la plateforme reste expérimental. De plus

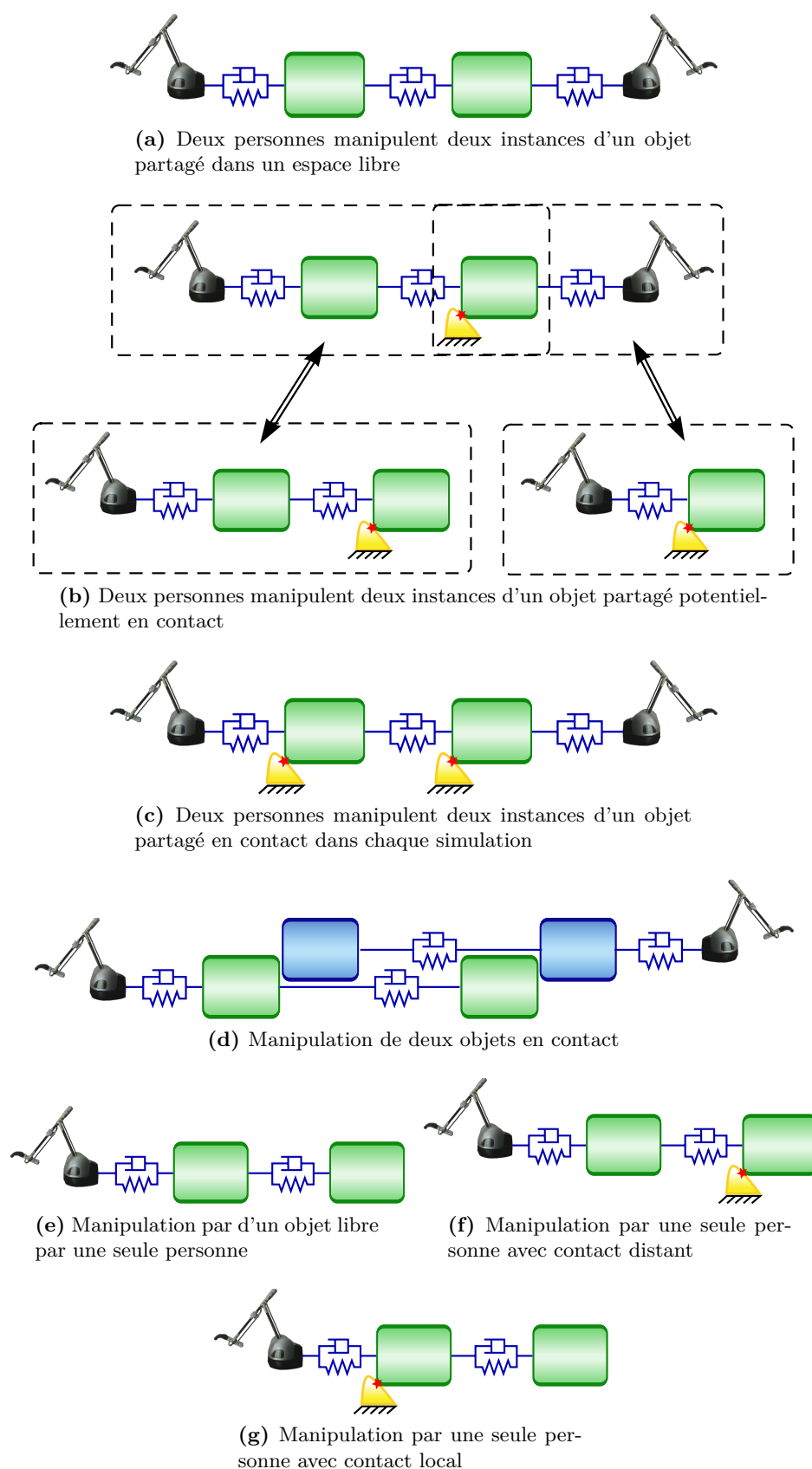


Figure 3.4 – Les différents cas de manipulation rencontrés lors d'une tâche collaborative.

ces cas sont les cas principaux pouvant apparaître, et ne représente pas de manière exhaustive l'ensemble des configurations que peut adopter la plateforme. Afin de garantir la stabilité de la plateforme dans le cas complexe à 6 degrés de liberté, un observateur de passivité peut être ajouté. Ce type d'ajout dégrade la qualité du retour haptique car la condition de passivité sur laquelle il est basé est plus restrictive que la condition de stabilité utilisée dans la présente analyse. Cependant, en pratique l'utilisation des gains issus de cette analyse nous a permis de réaliser des interactions collaboratives et coopératives stables au cours desquelles nous n'avons pas constaté d'instabilité.

### 3.2.2 Formulation discrète

Le simulateur est un système discret, la stabilité du système est donc étudiée dans le domaine discret. En exprimant le système, pour les différents cas présentés ci-dessus, sous la forme d'une matrice d'immittance, le critère de Llewellyn (voir Section 1.2) permet d'obtenir les conditions sur les différents gains permettant de garantir la stabilité du système.

Les ressorts virtuels sont calculés par le simulateur selon un schéma d'intégration Euler implicite (Section 1.1.2.4), leur expression dans le domaine discret est la suivante :

$$Z_{cv}(z) = b + \frac{k}{s} \Big|_s \rightarrow \frac{z-1}{\Delta T z} \quad (3.4)$$

L'interface haptique est un système continu, son impédance est obtenue dans le domaine discret en appliquant une transformation bilinéaire (ou transformation de Tustin) à l'impédance exprimée dans le domaine de Laplace. Cette transformation permet de conserver les mêmes conditions de stabilité pour le domaine discret que pour le domaine continu :

$$Z_h(z) = m_h \frac{2(z-1)}{\Delta T(z+1)} + b_h \Big|_s \rightarrow \frac{2(z-1)}{\Delta T(z+1)} \quad (3.5)$$

La conversion des signaux entre la simulation, qui est discrète, est réalisée par des convertisseurs analogique numérique et numérique-analogique. Ce dernier, un échantillonneur, peut être modélisé par un filtre passe-bas :

$$Z_{ech} = \frac{z+1}{2z} \quad (3.6)$$

Le retard  $T_i$  est exprimé dans le domaine discret par :

$$n_i = \frac{T_i}{\Delta T} \quad (3.7)$$

Afin de simplifier l'écriture, les termes dépendant des ressorts-amortisseurs sont regroupés tels que,  $C_c(s) = b_c + \frac{k_c \Delta T(z+1)}{z}$ ,  $C_h(s) = b_h + \frac{k_h \Delta T(z+1)}{z}$ .

### 3.2.3 Matrices d'immittance des différents cas de manipulations

Les cinq cas présentés précédemment correspondent à cinq configurations différentes de la plateforme collaborative. Chacun des cas peut être exprimé sous la forme d'une matrice d'immittance. La superposition des domaines de stabilité fournit par le critère de Llewellyn appliqué à ces différentes matrices fournit un domaine de stabilité permettant de réaliser l'ensemble de ces tâches de manière stable.

Le premier cas correspond au schéma 3.2 et aux cas où deux utilisateurs manipulent deux instances d'un objet répliqué (voir Figure 3.4a et Figure 3.4d). Le cas où deux objets manipulés sont en contact relève également de ce cas.

Ensuite, les cas où un utilisateur manipule un objet libre (voir Figure 3.4e) et où un utilisateur manipule un objet en contact dans la simulation distante (voir Figure 3.4f) peuvent être étudiés ensemble. En effet, en exprimant ces deux cas par comme la manipulation d'un objet distant évoluant dans un environnement discret passif, les deux cas sont étudiées simultanément.

Enfin les cas restant où les instances de l'objet partagé sont en contact dans deux simulations (voir Figure 3.4c) et où l'objet est en contact local (voir Figure 3.4g) sont similaire au cas d'un contact local. Ils sont donc étudiés de la même manière que le couplage virtuel associé à une unique simulation physique [AMH99].

### 3.2.3.1 Manipulation d'un objet par deux personnes

Le système utilisé lors d'une manipulation collaborative d'un objet partagé (voir Figure 3.3) peut être mis sous la forme de la matrice d'admittance suivante :

$$\begin{bmatrix} V_{h_1}(z) \\ -V_{h_2}(z) \end{bmatrix} = \begin{pmatrix} Y_{\text{coll}11}(z) & Y_{\text{coll}12}(z) \\ Y_{\text{coll}21}(z) & Y_{\text{coll}22}(z) \end{pmatrix} \begin{bmatrix} F_{h_1}(z) \\ -F_{h_2}(z) \end{bmatrix} \quad (3.8)$$

$V_{h_1}$  et  $V_{h_2}$  sont les vitesses des interfaces haptiques 1 et 2 respectivement. Les deux utilisateurs sont considérés comme étant des opérateurs passifs [AH98]. L'expression de la matrice d'admittance est sensiblement différente de l'écriture couramment rencontrée car le système est symétrique. La vitesse  $V_{h_2}$  possède donc un signe opposé au cas maître-esclave. La formulation adoptée est similaire à celle utilisée par Adams et Hannaford [AH98] pour l'étude de la stabilité d'un environnement avec retour d'effort à l'exception du signe de  $F_{h_2}$ . En effet, en téléopération, ce terme correspond à l'effort du robot esclave sur l'environnement alors qu'ici nous considérons la plateforme symétrique et donc la force de l'utilisateur 2 sur l'interface haptique. Cette matrice décrit le schéma 3.5a. Ce schéma peut être simplifié en considérant les deux simulateurs ensembles (voir Figure 3.5b).

Les paramètres de la matrice d'admittance  $Y_{\text{coll}}$  sont déduits du système d'équations 3.3 :

$$Y_{\text{coll}11}(z) = \frac{(b_s + C_h + C_c)(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c)) - C_c^2(Z_h + C_h Z_{\text{ech}})z^{-n_r}}{(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c))^2 - C_c^2(Z_h + C_h Z_{\text{ech}})^2 z^{-n_r}} \quad (3.9)$$

$$Y_{\text{coll}12}(z) = \frac{C_c C_h^2 Z_{\text{ech}} z^{-n_1}}{(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c))^2 - C_c^2(Z_h + C_h Z_{\text{ech}})^2 z^{-n_r}} \quad (3.10)$$

$$Y_{\text{coll}21}(z) = \frac{C_c C_h^2 Z_{\text{ech}} z^{-n_2}}{(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c))^2 - C_c^2(Z_h + C_h Z_{\text{ech}})^2 z^{-n_r}} \quad (3.11)$$

$$Y_{\text{coll}22}(z) = \frac{(b_s + C_h + C_c)(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c)) - C_c^2(Z_h + C_h Z_{\text{ech}})z^{-n_r}}{(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c))^2 - C_c^2(Z_h + C_h Z_{\text{ech}})^2 z^{-n_r}} \quad (3.12)$$

Le système possède en effet un comportement symétrique. Le comportement local d'un nœud est équivalent à celui des autres nœuds, car  $Y_{\text{coll}11} = Y_{\text{coll}22}$ . En outre si  $T_1 = T_2$  les termes de couplages entre les simulations sont également équivalentes. Il peut être noté que la stabilité du système dépend du retard aller-retour  $T_r$ , car le critère de Llewellyn dépend des termes diagonaux  $Y_{\text{coll}11}$ ,  $Y_{\text{coll}22}$  et du produit  $Y_{\text{coll}12} Y_{\text{coll}21}$ , or les retards unidirectionnels  $T_1$  et  $T_2$  apparaissent uniquement au numérateur des termes non diagonaux de la matrice d'admittance et se simplifient donc en  $T_r$ .

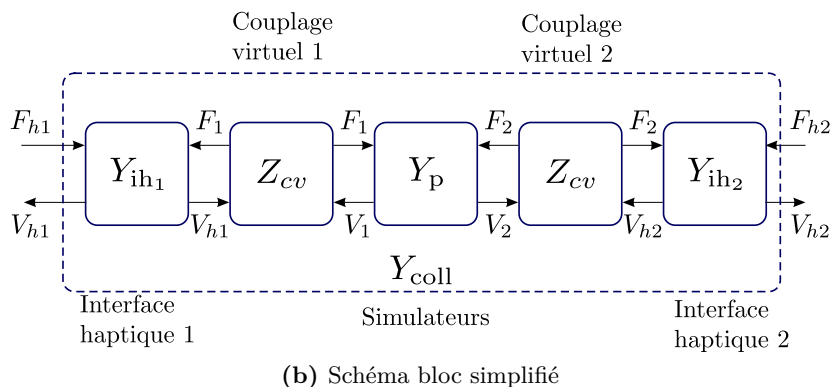
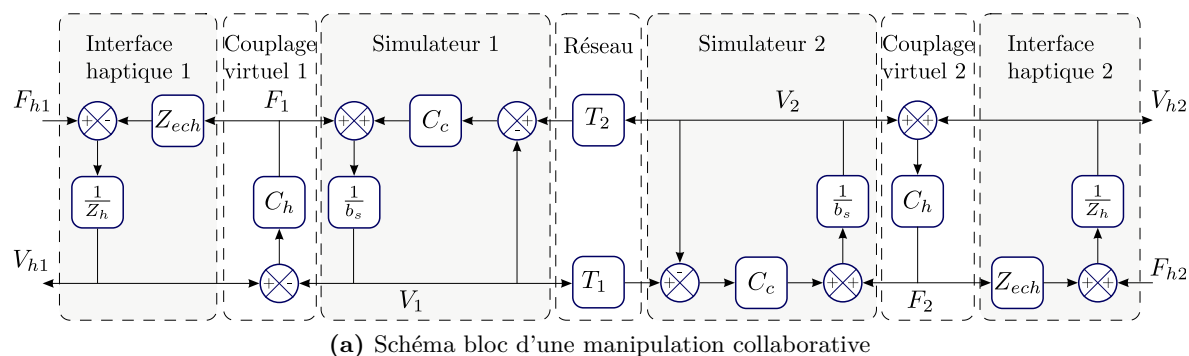


Figure 3.5 – Schéma bloc de la manipulation par deux personnes d'un objet répliqué

Lorsque deux personnes manipulent deux objets distincts en contacts entre eux (voir Figure 3.4d), le système est strictement équivalent, hormis le terme de couplage  $C_c$  entre les objets virtuels. Il est en effet doublé à cause des deux asservissements dus aux deux objets partagés :  $C_{c_{contact}} = 2C_{c_{coll}}$ . La matrice d'admittance est notée  $Y_{coll-contact}$

### 3.2.3.2 Manipulation d'un objet libre et manipulation avec contact distant

Les deux cas de manipulation que sont la manipulation libre et la manipulation avec contact distant sont analysés ensemble. Ces cas peuvent être généralisés en considérant que l'objet distant interagit avec un environnement passif (voir schéma Figure 3.6). En pratique, si la simulation n'est pas passive, elle peut être rendue passive en utilisant des méthodes telles que les observateurs et contrôleurs de passivité [HR02]. Comme précédemment, la personne manipulant est considérée comme un opérateur passif. En prenant les deux cas extrêmes  $Z_{EV_2} = 0$  et  $Z_{EV_2} = \infty$ , on retrouve bien les deux cas de manipulations, objet libre et contact distant.

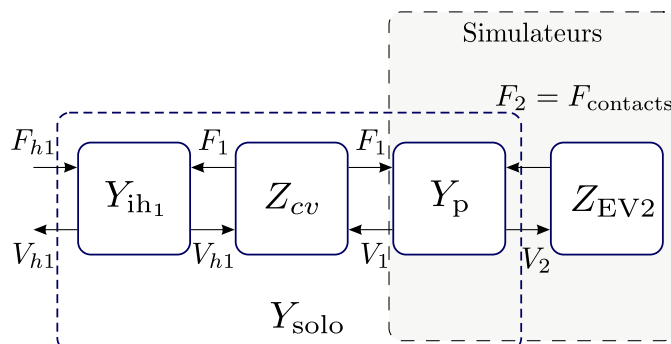


Figure 3.6 – Manipulation par un seul manipulateur

Ce système peut être ainsi exprimé sous la forme d'une matrice d'impédance  $Y_{\text{solo}}$  :

$$\begin{bmatrix} V_{h_1}(z) \\ -V_2(z) \end{bmatrix} = \begin{pmatrix} Y_{\text{solo}_{11}}(z) & Y_{\text{solo}_{12}}(z) \\ Y_{\text{solo}_{21}}(z) & Y_{\text{solo}_{22}}(z) \end{pmatrix} \begin{bmatrix} F_{h_1}(z) \\ -F_2(z) \end{bmatrix} \quad (3.13)$$

Les éléments de cette matrice d'admittance sont détaillées dans l'annexe A.2.1.1.

### 3.2.3.3 Manipulation d'un objet avec contact local

Le cas d'un contact local est similaire au mur virtuel [CSB95]. Comme pour le cas précédent, le mur virtuel peut être généralisé en étudiant la stabilité du système formé du couplage virtuel et de l'interface haptique [AH98] (voir schéma Figure 3.7). L'utilisateur est toujours considéré comme un opérateur passif. La simulation est également considérée comme étant passive.

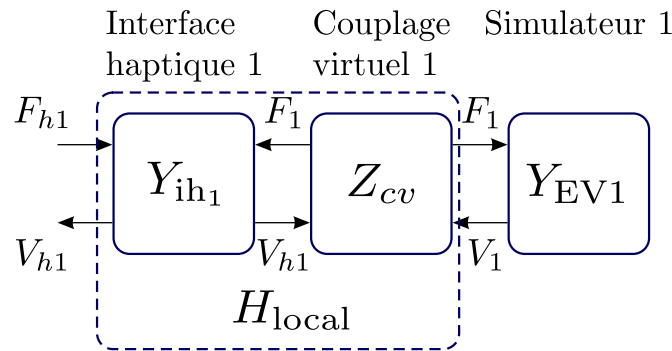


Figure 3.7 – Cas d'un contact local

Ce système peut être exprimé sous forme d'une matrice hybride  $H_{\text{local}}$  :

$$\begin{bmatrix} V_{h_1}(z) \\ F_1(z) \end{bmatrix} = \begin{pmatrix} H_{\text{local}_{11}}(z) & H_{\text{local}_{12}}(z) \\ H_{\text{local}_{21}}(z) & H_{\text{local}_{22}}(z) \end{pmatrix} \begin{bmatrix} F_{h_1}(z) \\ -V_1(z) \end{bmatrix} \quad (3.14)$$

L'expression des éléments de cette matrice sont décrit dans [AH98].

### 3.2.4 Étude de stabilité

La plateforme est stable si les quatre matrices précédentes  $Y_{\text{coll}}$ ,  $Y_{\text{coll-contact}}$ ,  $Y_{\text{solo}}$  et  $H_{\text{local}}$  respectent chacune le critère de stabilité de Llewellyn (voir Equation 1.17). Le domaine de stabilité correspond à la superposition des domaines de stabilité issus de l'analyse des matrices présentées.

Le terme d'amortissement de l'objet virtuel  $b_s$  est généralement choisi petit ou nul afin de réduire la viscosité globale ressentie par les utilisateurs.  $b_s$  est donc négligeable devant  $b_h$  ou  $b_c$ , sauf lorsque ces derniers sont très faibles. Des valeurs très faibles de  $b_h$  et  $b_c$  correspondent à des gains faibles et donc un asservissement permettant une grande divergence. Afin d'être utilisable, les gains des différents asservissements ne doivent pas être trop faibles.  $b_s$  est donc toujours considéré négligeable devant les autres amortissements.

Le critère de stabilité de Llewellyn appliqué à chacun des systèmes décrit précédemment, permet d'obtenir des conditions sur les gains  $b_h$ ,  $k_h$ ,  $b_c$ ,  $k_c$  permettant de garantir la stabilité du système. Les interfaces haptiques considérées sont des Virtuose6D d'Haption. En translation, la masse  $m_h$  vaut environ 1 kg. L'amortissement est compris entre 0.25 et 0.4  $\text{Nm}^{-1}\text{s}$  selon les axes. Nous avons retenu la valeur la plus faible d'amortissement afin de considérer le cas le moins stable.

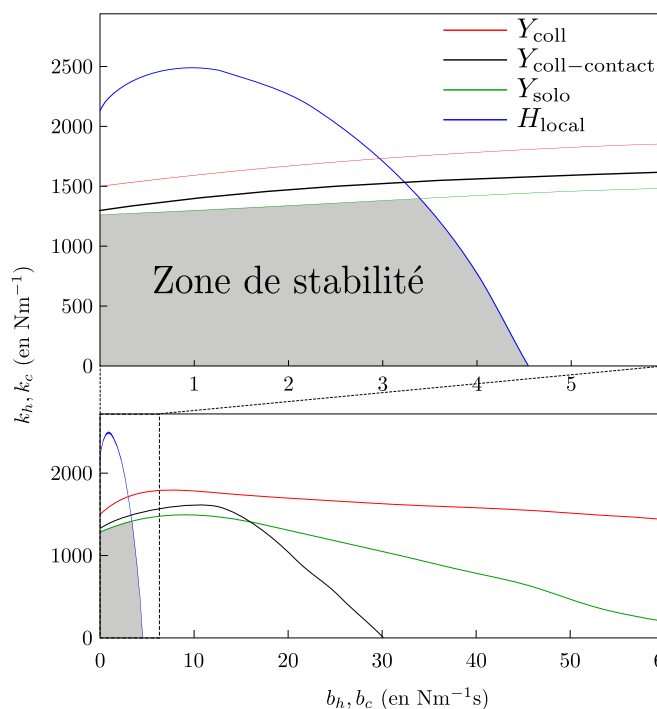


Figure 3.8 – Domaine de stabilité pour un retard minimal pour  $k_c = k_h$  et  $b_c = b_h$

### 3.2.4.1 Influence du retard

Pour cette évaluation, nous avons considéré les différents gains des couplages virtuels comme égaux,  $b_c = b_h$  et  $k_c = k_h$ . Ce choix traduit le fait que le retour d'effort n'est pas privilégié par rapport à l'asservissement entre les instances de l'objet partagé et réciproquement. Un rapport  $\frac{b_h}{b_c}$  et  $\frac{k_c}{k_h}$  grand autoriserait un retour d'effort faible. L'inverse permettrait aux instances des diverger largement les unes des autres. L'amortissement  $b_s$  est considéré nul.

Le cas d'un retard minimal est tout d'abord étudié (voir Figure 3.8). Dans le cas de la plateforme, le retard minimal est de l'ordre de 1 à 3 ms. En effet, le temps de communication du réseau entre deux nœuds ajouté au temps de traitement des données (traitement par les couches réseaux du système d'exploitation) peut prendre entre 1 et 2 ms dans le pire des cas. A cela, il faut ajouter le fait que les moteurs physiques des deux nœuds ne sont pas synchronisés. Un pas de temps de décalage, soit 1 ms peut apparaître entre les simulations. Afin d'obtenir un retard fixe, ce dernier est majoré par la borne maximal du retard variable. Les marges de stabilité sont donc calculées dans le pire des cas, soit pour un retard de 3 ms.

Dans le cas où le retard est minimal, le domaine de stabilité est borné par le cas d'une manipulation simple (courbe bleue de la Figure 3.8) et le cas d'une manipulation à un seul utilisateur entre deux nœuds (courbe verte). La comanipulation d'un objet à deux utilisateurs (courbe rouge) et le cas où deux objets sont en collision (courbe noire) sont plus stables que le cas de la manipulation à un seul utilisateur. L'ajout d'un autre opérateur pour une manipulation collaborative via une interface haptique supplémentaire tend donc à stabiliser le système.

Le cas de figure d'un retard minimal n'est pas forcément intéressant, dans le sens où le délai est suffisamment faible pour qu'une solution client-serveur soit à privilégier. En effet, l'architecture distribuée introduit un ressort amortisseur entre les objets répliqués, l'impédance transmise par le système entre les deux interfaces haptiques est plus faible que dans le cas client-serveur et donc les efforts transmis moins importants. De plus un certain



Site	Adresse	Retard total ( $T_r$ ) (en ms)
University of Oxford	ox.ac.uk	15.5
Universidad Politécnica de Madrid	upm.es	25
Technische Universität München	mytum.de	31
Uniwersytet Warszawski	uv.edu.pl	40
New York University	nyu.edu	105
The University of Tokyo	u-tokyo.ac.jp	297

**Table 3.1** – Ordre de grandeur de retard en le CEA Fontenay aux Roses et des universités.

retard est introduit entre les deux simulateurs, ce qui n'est pas le cas pour une architecture client-serveur. L'impédance transmise de l'utilisateur distant sera donc plus faible. Néanmoins, dans le cas multi-fréquentiel, l'architecture distribuée peut se révéler plus stable que l'architecture client-serveur en l'absence de retard [FSC07].

Le cas de retards supérieur au retard minimal est donc étudié avec plus d'attention (voir Figure 3.9). De plus, afin de simplifier l'étude, les retards sont égaux dans les deux sens,  $T_1 = T_2 = \frac{T_r}{2}$ .

De la même manière que pour le retard minimal, les domaines de stabilité sont bornés par le cas de la manipulation simple et le cas de la manipulation à un seul utilisateur. L'ajout d'un second utilisateur permet également de stabiliser le système lorsque le retard croît. Par contre, les gains diminuent de manière considérable lorsque le retard augmente. En pratique, le retour d'effort commence à être réellement faible pour des retards supérieurs à 50 ms. De plus, des gains faibles favorisent une grande divergence entre les objets répliqués, en présence de retards importants l'utilisation d'asservissements ressort-amortisseurs peut donc facilement générer des incohérences.

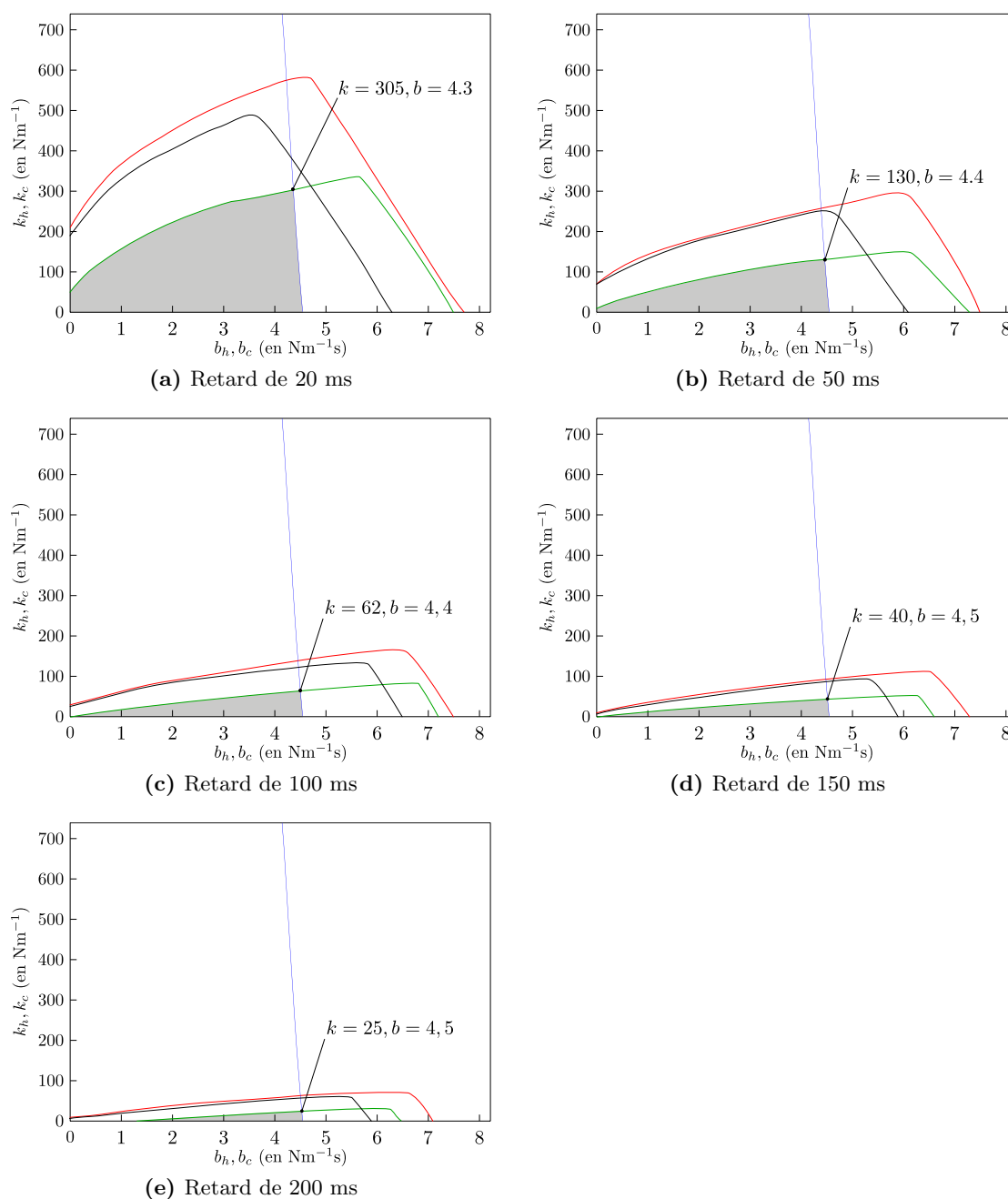
L'ordre de grandeur de 50 ms est comparable aux ordres de grandeur de retards rencontrés lors de communication sur internet pour une distance allant jusqu'à 2000 km en utilisant des réseaux de bonne qualité (réseaux universitaire et de recherche de type RENATER) (voir table 3.1).

### 3.2.5 Efforts de traînée

Le schéma Figure 3.10 présente des mesures d'efforts et de divergence en position et rotation pour un mouvement libre. L'écart en rotation  $\theta_{j(t-T_j)}^{i(t)}$  (voir Figure 3.10b) est calculé à partir de la matrice de rotation  $R_{j(t-T_j)}^{i(t)}$  entre les objets  $j$  et  $i$  par :

$$\theta_{j(t-T_j)}^{i(t)} = \arccos \left( \frac{\text{tr}(R_{j(t-T_j)}^{i(t)}) - 1}{2} \right) \quad (3.15)$$

Lors d'un mouvement libre, des forces et des moments de traînée apparaissent (voir Figure 3.10c et 3.10d). Le mouvement libre n'est donc pas transparent. Ces efforts permettent de limiter la divergence en position entre les objets répliqués mais peuvent devenir rapidement fatiguant pour les utilisateurs. Par exemple, dans le cas d'un mouvement libre avec un retard de 50 ms (voir Figure 3.11) d'une amplitude maximum de 70 cm, la norme maximum de l'effort peut atteindre 6 N et la norme du moment atteint 0.5 Nm au niveau de l'interface haptique. L'écart en position reste lui relativement petit devant l'amplitude du mouvement, avec 3 cm au maximum et 0.3 rad en rotation.

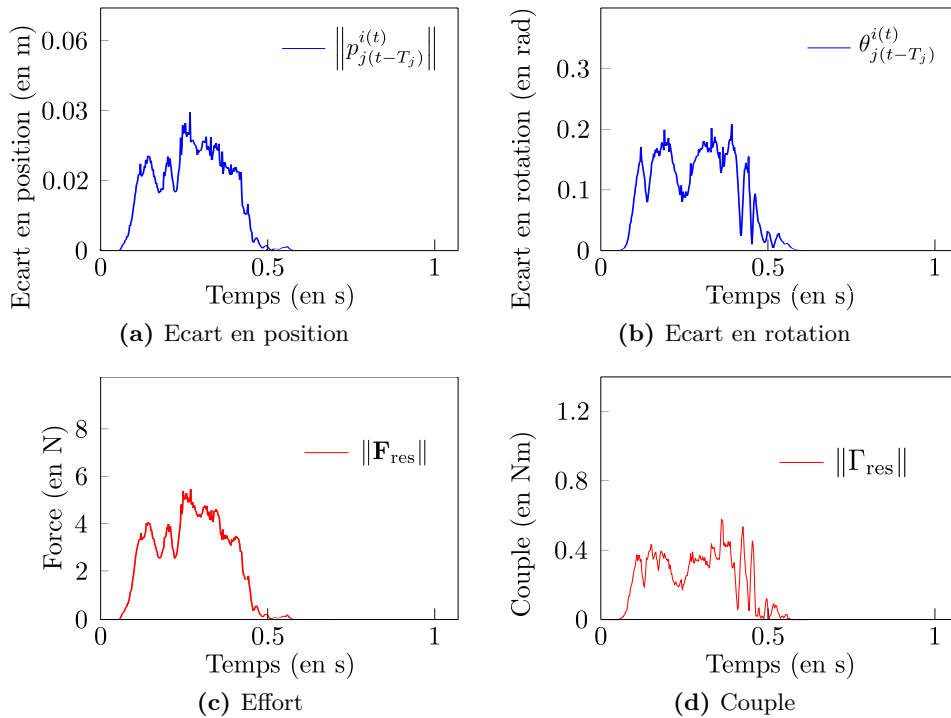


**Figure 3.9** – Domaine de stabilité en fonction de retards croissants

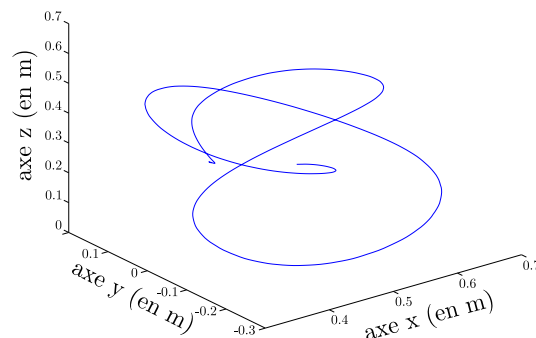
L'impédance du système formé des deux moteurs physiques et du couplage virtuel lors d'une manipulation libre (voir Figure 3.12) est exprimé à partir de  $Y_{\text{solo}}$  par :

$$F_1 = Z_{\text{libre}} V_{h_1}$$

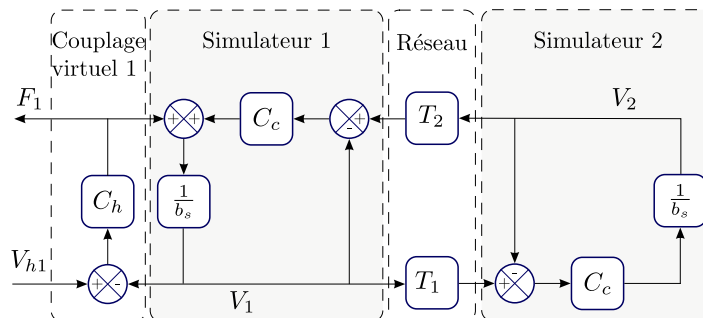
$$Z_{\text{libre}} = \frac{C_h(C_c^2 z^{-n_r} - C_c^2 - 2C_c b_s - b_s^2)}{C_c^2 z^{-n_r} - C_c^2 - 2C_c b_s - C_h C_c - C_h b_s - b_s^2} \quad (3.16)$$



**Figure 3.10** – Efforts de traînée lors d'un mouvement libre, avec un retard  $T_1 = T_2 = 25$  ms,  $k_c = k_h = 80 \text{ Nm}^{-1}$  et  $b_c = b_h = 3.6 \text{ Nm}^{-1}\text{s}$ .  $\mathbf{F}_{\text{res}}$  et  $\mathbf{\Gamma}_{\text{res}}$  sont les forces et les moments transmis à l'interface haptique



**Figure 3.11** – Trajectoire de test, position de l'objet



**Figure 3.12** – Schéma bloc d'une manipulation libre, sans l'interface haptique

or  $b_s$  est soit nul soit très petit devant  $b_h$  ou  $b_c$ , d'où

$$Z_{\text{libre}} = \frac{C_h C_c (z^{-n_r} - 1)}{C_c (z^{-n_r} - 1) - C_h} \quad (3.17)$$

L'effort de traînée dépend donc de la vitesse  $V_{h1}(t)$  et de cette vitesse retardée  $V_{h1}(t - n_r \Delta T)$ . En effet, lors d'un mouvement libre  $V_2(t) = V_1(t - n_1 \Delta T)$ , car l'instance distance de l'objet partagé dans la simulation 2 n'est pas contraint. Il suit donc sa consigne. La consigne reçue par l'instance de l'objet dans la simulation 1 depuis la simulation 2 vaut  $V_2(t - n_2 \Delta T) = V_1(t - n_r \Delta T)$ . L'effort de traînée subit par l'objet dans la simulation 1 vaut donc :

$$F = k_c(x_1(t - n_r \Delta T) - x_1(t)) + b_c(\dot{x}_1(t - n_r \Delta T) - \dot{x}_1(t)) \quad (3.18)$$

L'effort de traînée peut être réduit en diminuant les gains du couplage  $b_c$  et  $k_c$  au détriment du suivi en position. Cet effort est nul pour des gains nuls. Afin de ne pas pénaliser le suivi en position et d'annuler l'effort de traînée, nous proposons de compenser ce dernier au niveau du couplage virtuel.

### 3.3 Compensation des efforts de traînée

#### 3.3.1 Boucle de compensation

Afin de compenser les efforts de traînée mis en évidence dans la section précédente, nous proposons, dans le cas des asservissements par couplage proportionnel dérivé, d'introduire une boucle de compensation. Nous souhaitons ainsi annuler les efforts de traînée avant qu'ils ne soient transmis à l'interface haptique. Nous suggérons donc d'ajouter une boucle de compensation au niveau de chaque couplage virtuel entre l'interface haptique et la simulation physique (voir schéma 3.13). L'expression des matrices d'immittance précédemment décrite avec la boucle de compensation sont détaillées dans l'annexe A.2.2.

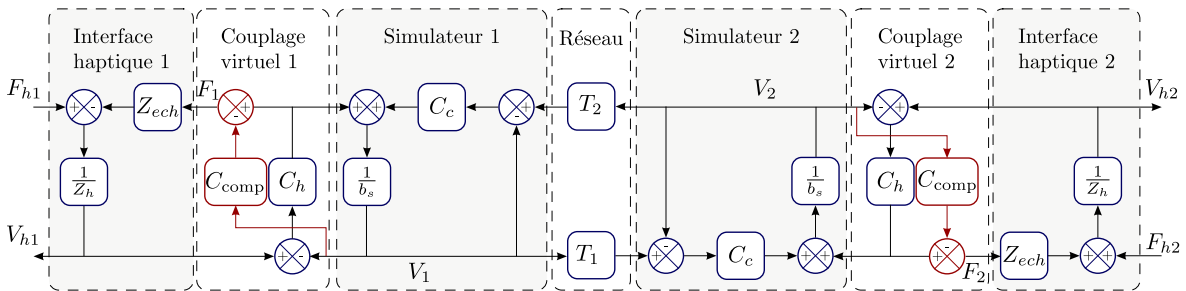


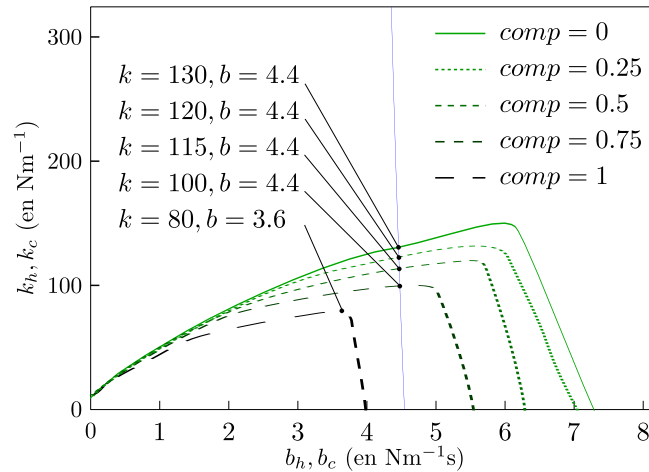
Figure 3.13 – Schéma bloc avec compensation

Dans le cas d'un mouvement libre, l'Equation 3.17 devient alors :

$$Z_{\text{libre}} = \frac{C_h (C_c (z^{-n_r} - 1) + C_{\text{comp}})}{C_c (z^{-n_r} - 1) - C_h} \quad (3.19)$$

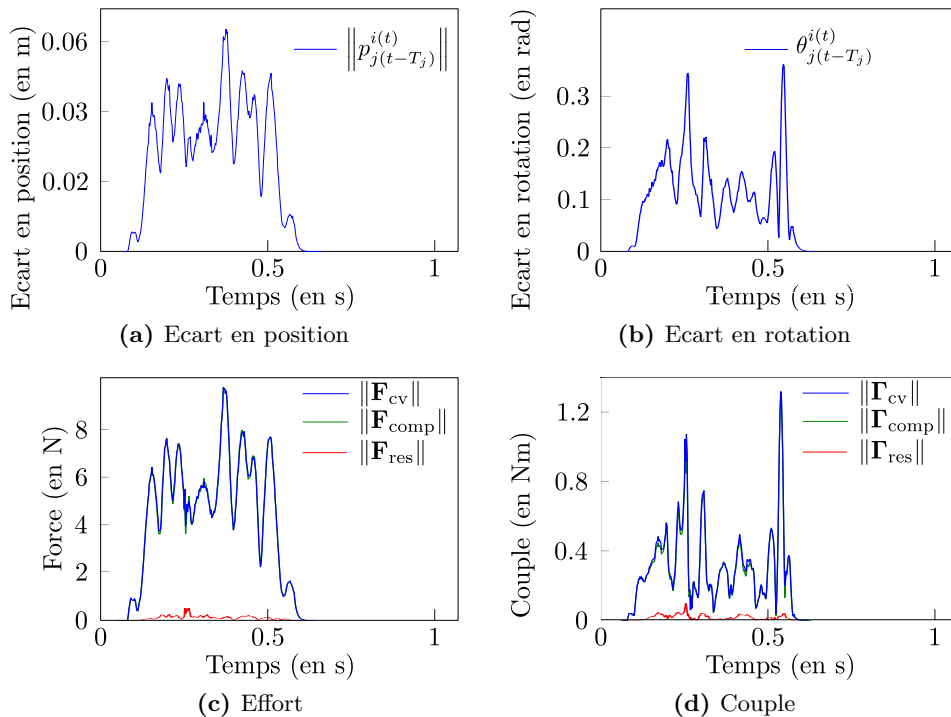
Le mouvement est libre si  $Z_{\text{libre}} = 0$ . En posant  $C_{\text{comp}} = C_d(1 - z^{-n_r})$  et  $C_d = C_c$  l'effort de traînée est ainsi compensé.

Le domaine de stabilité du système est calculé de la même manière que précédemment. Ce domaine est présenté sur le schéma 3.14. Seules les courbes limitant de domaine de stabilité sont présentes. L'ajout d'une compensation réduit le domaine de stabilité du système. La réduction des gains a tendance à augmenter la divergence entre les deux instances de l'objet

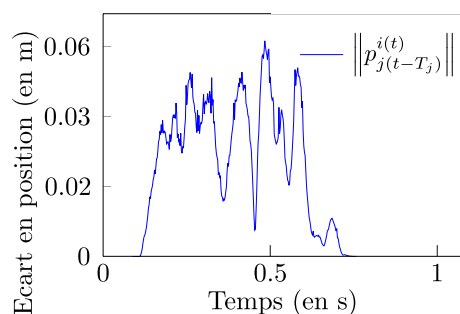


**Figure 3.14** – Effet de la compensation sur les marges de stabilité pour un retard de 50 ms et  $k_h = k_c$  et  $b_h = b_c$ .  $\mathbf{F}_{cv}$  et  $\mathbf{\Gamma}_{cv}$  sont les forces et les moments calculés par le couplage virtuel liant la simulation physique à l’interface haptique.  $\mathbf{F}_{comp}$  et  $\mathbf{\Gamma}_{comp}$  sont calculés par la boucle de compensation.

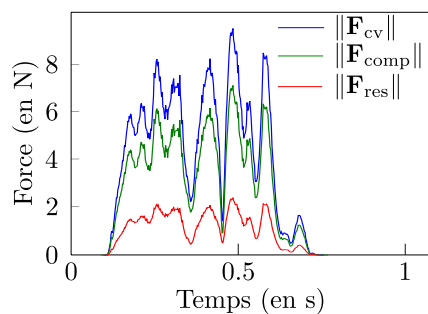
répliqué. Par contre, la compensation d’effort élimine les efforts de traînée qui s’opposent au mouvement lorsque le mouvement tend à augmenter l’écart entre les deux instances de l’objet partagé. L’utilisateur peut ainsi appliquer une vitesse plus importante à l’instance manipulée, augmentant encore la divergence entre les instances de l’objet. Par exemple avec la compensation activée et des gains  $k_h = k_c = 80 \text{ Nm}^{-1}$  et  $b_c = b_h = 3.6 \text{ Nm}^{-1}\text{s}$  pour un retard total  $T_r = 50 \text{ ms}$  l’écart de position peut être deux fois plus grand pour le cas avec compensation (voir Figure 3.15) que sans compensation (voir Figure 3.10) (0.06 m contre 0.03 m maximum).



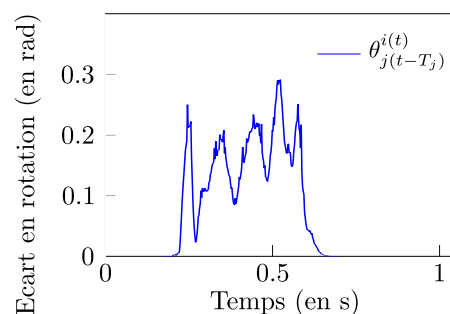
**Figure 3.15** – Effet de la compensation sur un mouvement libre, avec un retard  $T_1 = T_2 = 25 \text{ ms}$ ,  $k_c = k_h = 80 \text{ Nm}^{-1}$  et  $b_c = b_h = 3.6 \text{ Nm}^{-1}\text{s}$



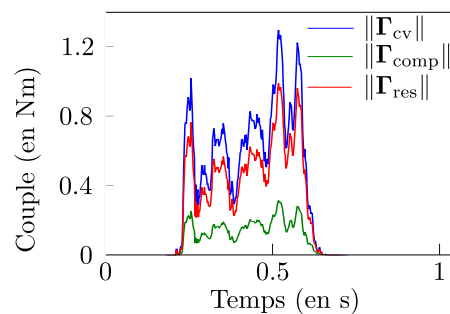
(a) Ecart en position



(b) Effort

**Figure 3.16** – Effet d’une compensation de 0.75 sur les translations

(a) Ecart en rotation



(b) Couple

**Figure 3.17** – Effet d’une compensation de 0.25 sur les rotations

### 3.3.2 Effet de la boucle de compensation

Afin de réduire ces effets, une compensation partielle peut être appliquée en posant :

$$C_d = \alpha C_c \text{ avec } \alpha \in [0; 1] \quad (3.20)$$

Des gains plus importants peuvent alors être utilisés (voir Figure 3.14). Il y a donc un compromis entre à établir entre la compensation désirée, les gains voulus et la divergence maximale admissible. Des exemples du comportement du système sont fournis pour de tels compromis sur les figures 3.16 et 3.17.

L’ajout de la boucle de compensation peut être interprété comme un point de vue *optimiste* de l’interaction distante, alors que la version sans compensation correspond à un point de vue *pessimiste*. Avec une compensation complète ( $\alpha = 1$ ), le système se comporte comme si l’instance distante suivait l’instance locale de l’objet partagé, et qu’il n’y a pas lieu d’empêcher le mouvement sauf indication contraire de la part de la simulation distante. Au contraire sans la compensation, le système se comporte comme s’il attendait d’obtenir la position de l’instance distante avant *d’autoriser* le déplacement de l’instance locale. Les forces de traînée sont là pour empêcher l’instance locale de s’écarter de la dernière position connue de l’instance distante. Avec la compensation en mouvement libre, puisque l’instance distante est sensée suivre l’instance locale, il n’y a pas lieu de limiter le mouvement de l’utilisateur.

Les implications de la compensation sont observables lorsque l’instance distante est fixe dans l’environnement (voir Figure 3.18). L’effort de rappel entre l’instance distante fixe et l’instance locale libre n’est effectivement présent que 50 ms après le début de la divergence en position des deux instances, soit le temps que le retard total  $T_r$  s’écoule. Pendant les 50 premières ms l’effort de traînée est compensé, comme si le système considérait que l’instance distante suivait effectivement la position de l’instance locale. Donc, dans le cas sans compensation les efforts de traînée sont annulés, mais les efforts issus des interactions sur le

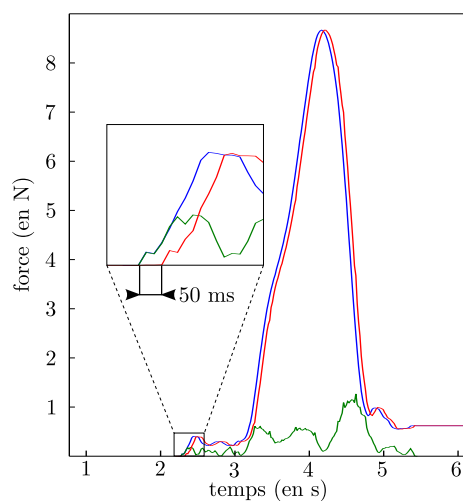


Figure 3.18 – Effet de la compensation lors d'un contact distant

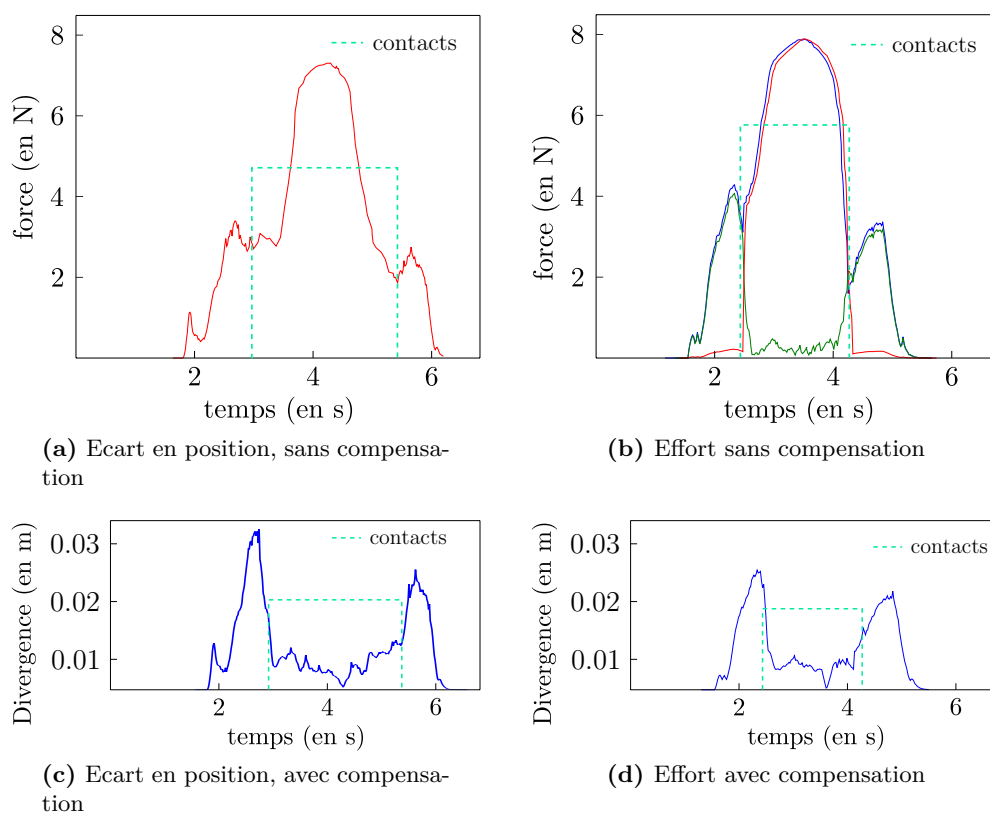


Figure 3.19 – Effet de la compensation lors d'un contact, avec un retard  $T_1 = T_2 = 25$  ms,  $k_c = k_h = 80$  Nm<sup>-1</sup> et  $b_c = b_h = 3.6$  Nm<sup>-1</sup>s

noeud distant sont retardés. Sans compensation ils sont retardés du retard ( $T_j$ ) entre le noeud distant  $j$  et le noeud local  $i$ , le temps que l'information se propage, avec la compensation ce retard vaut  $T_r$ , alors que les interactions locales comme le contact avec l'environnement ne sont pas soumis au retard. Néanmoins, la compensation permet de percevoir de manière plus franche les contacts (voir Figure 3.19) en supprimant les efforts de traînée. Les ordres de grandeur des efforts en contact (8 N maximum) sont à comparer à ceux des efforts de traînée (voir Figure 3.10) (entre 3,5 et 4 N maximum). Si l'utilisateur applique un effort plus faible, la différence ne sera pas forcément nette.

## Conclusion

Dans cette partie nous avons étudié l'utilisation d'asservissement proportionnel dérivé pour asservir les différentes instances des objets partagés. Une analyse à 1 degré de liberté des différents cas de manipulation nous a permis de déterminer des domaines stabilité de ce système. Ces domaines de stabilité sont utilisés pour régler expérimentalement les gains de la plateforme à 6 degrés de liberté.

L'utilisation de contrôleur proportionnel dérivé fait apparaître des efforts de traînée qui s'opposent au mouvement et peuvent d'une part rendre la manipulation fatigante et d'autre part altérer la perception des contacts. Nous avons proposé d'introduire une boucle de compensation permettant d'annuler partiellement ou complètement ces efforts. Les résultats de manipulations menées à 6 degrés de liberté montrent que le mouvement libre est effectivement quasi-transparent. Cependant, cette compensation introduit un retard supplémentaire et peut être gênant d'un point de vue perceptif. Il y a donc un compromis à obtenir sur l'utilisation de cette compensation en fonction du retard, et de la grandeur des efforts de traînée. De plus cette méthode reste valable tant que les retards sont faibles. Dans le chapitre suivant, nous proposerons d'utiliser les variables d'onde afin de garantir la stabilité du système quelque soit le retard.

## Perspectives

Dans cette étude, nous avons utilisé des gains équivalents pour chacun des couplages virtuels. Une perspective intéressante est d'étudier l'influence du rapport  $\frac{C_c}{C_h}$  sur la stabilité ainsi que sur le comportement en suivi et en effort. L'approche utilisant la compensation reste valide dans ce cas.

Un autre point intéressant, serait d'évaluer l'influence de l'amortissement  $b_s$  sur la stabilité du système. Un amortissement faible pourrait permettre d'utiliser des gains plus importants tout en conservant un système stable. Cependant, ce terme  $b_s$  doit rester faible afin de ne pas introduire d'efforts visqueux trop importants, ce qui annulerait les bénéfices de la boucle de compensation.





# Asservissement par variables d'onde

## Sommaire

<b>4.1</b>	<b>Couplage par variables d'onde articulaire</b>	<b>87</b>
4.1.1	Motivation	87
4.1.2	Formulation articulaire	89
4.1.3	Contrôle de la dérive	90
4.1.4	Filtrage des réflexions d'onde	92
<b>4.2</b>	<b>Expérimentation sur la plateforme collaborative</b>	<b>93</b>
4.2.1	Simulation	93
4.2.2	Plateforme collaborative	93
4.2.3	Influence du retard	95
	<b>Conclusion</b>	<b>100</b>

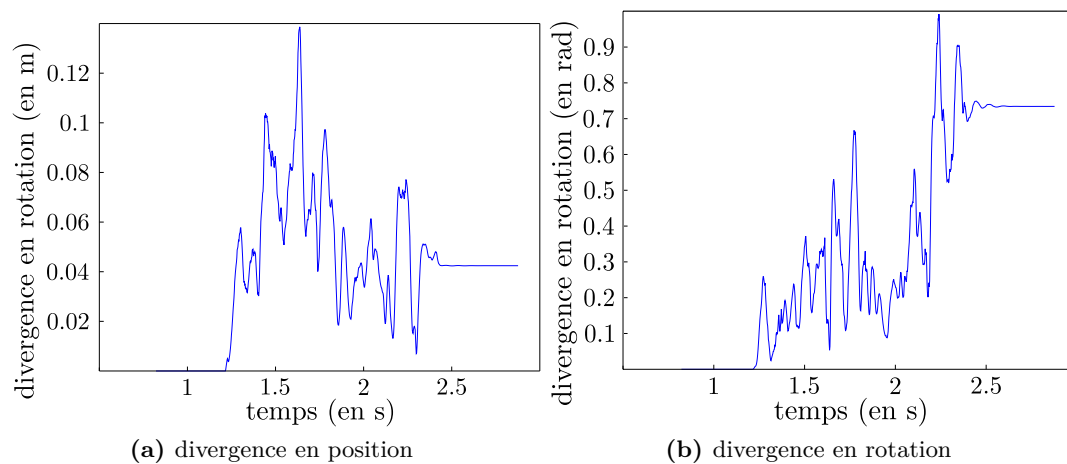
L'utilisation d'asservissements de type proportionnel dérivé devient instable lorsque le retard croît. Dans le cas d'un retard constant, la transformation en variables d'onde permet de garantir la passivité d'un système soumis à un retard. Cependant, les variables d'onde sont sujettes à différentes limitations. Utilisées directement, elles ne permettent pas d'assurer un suivi en position, ce qui est critique dans le cas d'une application collaborative distribuée. Ensuite, la différence d'impédance entre les variables d'onde et les environnements auxquels elles sont connectées peut entraîner des oscillations.

Nous proposons dans ce chapitre d'adapter la transformation en variables d'onde ainsi que les différentes techniques permettant de corriger les problèmes issus de cette transformation, et d'étudier les effets de ce couplage dans le cas d'une plateforme à 6 degrés de liberté. La première partie (Section 4.1) décrit l'utilisation des variables d'onde dans l'espace articulaire ainsi que la manière de corriger la dérive en position et les oscillations dues aux réflexions d'onde. La seconde partie (Section 4.2) présente les résultats de simulations et des résultats expérimentaux permettant d'évaluer l'influence du retard sur le comportement de la plateforme avec un asservissement utilisant des variables d'onde.

## 4.1 Couplage par variables d'onde articulaire

### 4.1.1 Motivation

La transformation en variables d'onde souffre d'un problème de dérive [NS97b, NS97a] (voir Section 1.3.3.2). Parmi les différentes méthodes permettant de corriger cet écart, l'utilisation d'intégrale d'onde [NS97b] et d'un ressort [CSOB06] permettent à tout instant de garantir la convergence indépendamment des problèmes de cohérence de l'univers virtuel. En effet la méthode consistant à observer la divergence et à la corriger est valide tant que la condition de passivité est respectée. Si l'univers n'est plus passif, la correction de la divergence



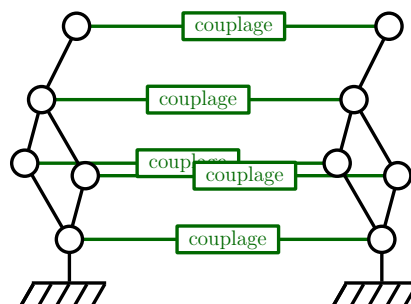
**Figure 4.1** – divergence induit par la transformation en variables d'onde pour un retard total de 100 ms.

ne peut être effectuée. La méthode visant à ajouter un ressort pour garantir la convergence en position est stable si la raideur  $k$  du ressort vérifie :

$$k^2 T_r^2 < b_m b_s \quad (4.1)$$

où  $b_m, b_s$  sont les amortissements présents au niveau du maître et de l'esclave. Une valeur trop faible de  $k$  ne permettra qu'une convergence très lente. Or, pour un retard  $T_r$  élevé, il faudrait des amortissements  $b_m$  et  $b_s$  élevés au niveau des objets virtuels. L'environnement virtuel résultant serait donc très visqueux. Il est intéressant de remarquer que cette condition est similaire à la condition de stabilité donnée dans [LS06] pour le cas d'un couplage basé sur un ressort-amortisseur.

La méthode que nous avons donc utilisée est basée sur l'intégrale des variables d'onde. Elle n'est cependant pas idéale car il n'existe pas de décomposition en intégrales de variables d'onde des valeurs considérées (intégrales des torseurs cinématiques et torseurs statiques) dans l'espace cartésien, contrairement à la décomposition des torseurs cinématiques et torseurs statiques [SAJvdSM02] (voir Section 1.3.3). En effet, les vitesses et les efforts sont exprimés dans des espaces vectoriels ( $\mathbb{R}^6$ ) alors que leurs intégrales ne le sont pas ( $SE(3)$  pour les positions). En choisissant un espace articulaire bien approprié [MB02], il est possible d'utiliser les intégrales de variables d'onde dans un espace vectoriel. C'est l'approche que nous avons retenu ici.



**Figure 4.2** – Exemple d'un asservissement articulaire pour deux robots redondants identiques

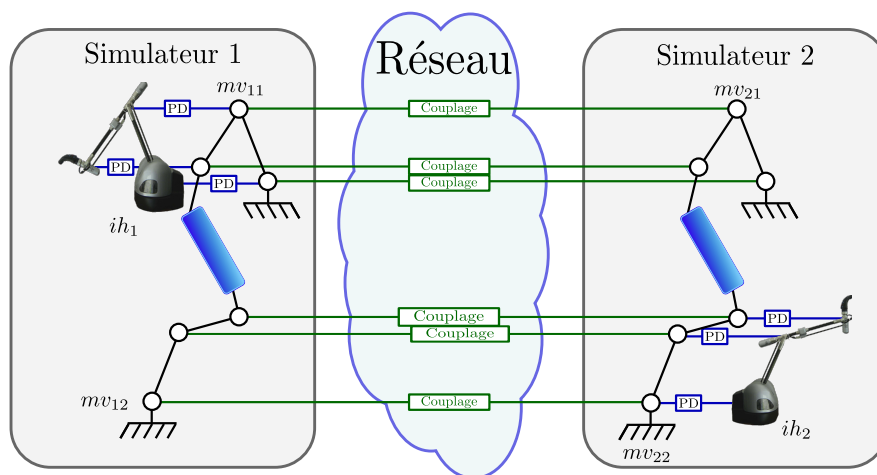
L'asservissement dans l'espace articulaire revient à asservir les différentes articulations de deux mécanismes entre eux. Chaque articulation est indépendante des autres (voir Figure 4.2),

chaque asservissement devient un asservissement à 1 degré de liberté contrairement au cas cartésien. Il peut être remarqué qu'un asservissement articulaire peut être directement appliqué pour asservir deux objets polyarticulés entre deux nœuds de la plateforme collaborative.

#### 4.1.2 Formulation articulaire

Pour cela il faut modifier sensiblement la plateforme utilisée dans la partie précédente (voir Figure 3.2). Nous introduisons des mécanismes virtuels dans la simulation physique pour pouvoir exprimer les variables d'onde dans le domaine articulaire. Grâce à ces mécanismes virtuels, les vitesses et positions cartésiennes des objets virtuels sont échangées par l'intermédiaire des vitesses et positions des articulations des mécanismes virtuels. Pour pouvoir utiliser les intégrales de variables d'onde, ces positions et vitesses doivent appartenir à un espace vectoriel. Les trois translations de l'espace associées aux trois rotations exprimées par les angles d'Euler forment un tel espace. Afin d'éviter les problèmes de singularité, ces mécanismes virtuels sont basés sur les mécanismes des interfaces haptiques (voir Figure 4.3). Entre une interface et son mécanisme virtuel associé au sein d'un nœud, l'asservissement est réalisé par un contrôleur proportionnel dérivé au niveau de chaque articulation. Entre deux mécanismes virtuels situés sur deux nœuds différents, l'asservissement est réalisé grâce à la transformation en variables d'onde, pour garantir la stabilité et l'intégrale de variables d'onde, pour garantir la convergence en position.

Outre garantir la convergence en position, cette approche permet d'utiliser un contrôleur simplifié pour l'interface haptique. En effet, dans le cas d'un couplage cartésien, le contrôleur de l'interface haptique doit posséder un mécanisme virtuel permettant de convertir l'expression de son état dans le domaine articulaire vers le domaine cartésien. Dans le cas d'un contrôle articulaire, cet élément n'est plus nécessaire. Ainsi, les gains (raideurs et amortissements) de l'interface haptique sont réglés directement en fonction des caractéristiques physiques de l'interface.



**Figure 4.3** – Modification de la plate-forme de collaboration afin d'utiliser la transformation en variables d'onde. Des mécanismes virtuels sont introduits. Les asservissements sont réalisés au niveau des articulations. Pour chaque nouvelle interface haptique, un mécanisme virtuel correspondant devra être ajouté dans chaque nœud.

L'équation d'évolution en quasi-statique Equation 1.3 doit donc être reformulée dans l'espace articulaire. Pour cela, les positions et vitesses articulaires du mécanisme virtuel  $mv_{\alpha\beta}$  représentant l'interface  $\alpha$  dans le moteur  $\beta$  sont noté respectivement  $\mathbf{q}_{\alpha\beta}, \dot{\mathbf{q}}_{\alpha\beta} \in \mathbb{R}^n$ . De la

même manière, les positions et vitesses articulaires de l'interfaces haptiques  $ih_i$  sont notées  $\mathbf{q}_{ih_i}, \dot{\mathbf{q}}_{ih_i} \in \mathbb{R}^n$ . A chaque interface haptique  $ih_i$  est donc associé dans chacun des moteurs physiques  $\beta$  un mécanisme virtuel  $mv_{i\beta}$  comme indiqué sur la Figure 4.3.

Les positions articulaires sont liées aux coordonnées cartésiennes par la géométrie du système et les vitesses articulaires sont liées aux torseurs cinématiques par la géométrie différentielle du mécanisme qui s'exprime par la matrice Jacobienne suivante :

$$T_A^A = J_v^A(\mathbf{q}_v)\dot{\mathbf{q}}_v \quad (4.2)$$

Considérons un système composé de deux nœuds 1 et 2 possédant chacun une interface haptique  $hd_1$  et  $hd_2$ . Pour un moteur  $i$ , l'interface locale est  $hd_i$ . Le moteur distant est noté  $j$  et l'interface distance est notée  $hd_j$ . Dans le cas d'un objet manipulé par deux utilisateurs, l'Equation 1.3 devient, dans l'espace articulaire, le système d'équation suivant au nœud  $i$  :

$$\begin{aligned} B(\mathbf{q}_{ii})\dot{\mathbf{q}}_{ii}(t) &= K_{hd_i}(\mathbf{q}_{hd_i}(t) - \mathbf{q}_{ii}(t)) \\ &+ B_{hd_i}(\dot{\mathbf{q}}_{hd_i}(t) - \dot{\mathbf{q}}_{ii}(t)) \\ &+ K_i(\mathbf{q}_{ij}(t - T_j) - \mathbf{q}_{ii}(t)) \\ &+ B_i(\dot{\mathbf{q}}_{ij}(t - T_j) - \dot{\mathbf{q}}_{ii}(t)) \\ &+ J_{Collision}^T W_{Collision} \end{aligned} \quad (4.3)$$

$$\begin{aligned} B(\mathbf{q}_{ji})\dot{\mathbf{q}}_{ji}(t) &= K_j(\mathbf{q}_{jj}(t - T_j) - \mathbf{q}_{ji}(t)) \\ &+ B_j(\dot{\mathbf{q}}_{jj}(t - T_j) - \dot{\mathbf{q}}_{ji}(t)) \end{aligned} \quad (4.4)$$

$$T_{C_i}^C(t) = J_{C_i}^i \dot{\mathbf{q}}_{ii}(t) = J_{C_j}^j \dot{\mathbf{q}}_{jj}(t) \quad (4.5)$$

$T_i$  est le retard entre le moteur  $i$  et  $j$ ,  $T_j$  entre  $j$  et  $i$ , comme défini précédemment.

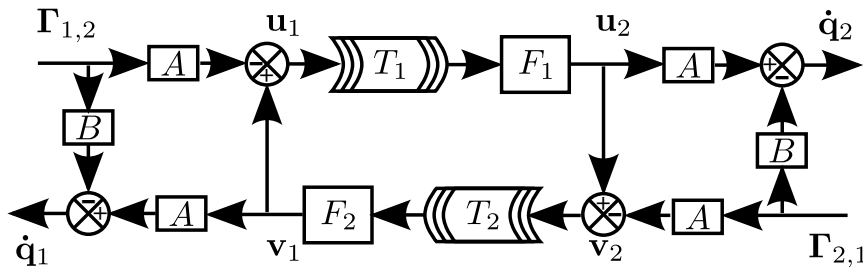
$\forall \alpha, K_\alpha, B_\alpha \in \mathbb{R}^n$  sont des vecteurs de raideurs et d'amortissements.  $K_{hd_i}, B_{hd_i}$  sont les gains des ressorts amortisseurs reliant l'interface  $hd_i$  au mécanisme virtuel  $mv_{hd_i}$ . Les gains  $K_i, B_i$  (resp.  $K_j, B_j$ ) sont ceux des ressorts amortisseurs reliant les mécanismes virtuels  $mv_{hd_i}$  et  $mv_{hd_{ij}}$  (resp.  $mv_{hd_{ji}}$  et  $mv_{hd_{jj}}$ ).

Lorsque chacun des utilisateurs manipule un objet distinct, les équations Equation 4.4 et Equation 4.5 sont éliminées du système d'équations précédent. Dans le cas plus complexe, où il y a plus de deux interfaces haptiques, les mécanismes virtuels correspondants doivent être ajoutés.

### 4.1.3 Contrôle de la dérive

Afin de garantir la stabilité du système présenté, les positions et les vitesses ( $\dot{\mathbf{q}}_{ij}, \mathbf{q}_{ij}, \dot{\mathbf{q}}_{jj}$  et  $\mathbf{q}_{jj}$ ) sont transmises en utilisant la transformation en variables d'onde. Nous supposons que tous les des nœuds (interfaces haptiques associées aux moteurs physiques) sont passifs ou peuvent être rendus passifs en utilisant un contrôleur de passivité par exemple. Seul le couplage bilatéral n'est pas considéré passif.

Nous utilisons la formulation des variables d'onde de Munir et Book [MB02] qui est la généralisation de la formulation des variables d'onde pour des mécanismes à  $n$  degrés de liberté. La transformation en variables d'onde n'étant pas symétrique (l'expression de l'onde incidente  $v$  et émise  $u$  est différente entre les différents nœuds), nous ne conservons plus les notations utilisant les indices  $i$  et  $j$  afin de distinguer le moteur physique 1 du nœud 1 du moteur 2 du second nœud.  $u$  est ainsi l'onde de 1 vers 2 et  $v$  l'onde de 2 vers 1. Les deux nœuds 1 et 2 ne sont désormais plus équivalents.



**Figure 4.4** – Transformation en variables d'onde dans l'espace articulaire.  $F_1$  et  $F_2$  sont deux filtres dans l'espace des variables d'onde.

La transformation en variables d'onde est donc exprimée par (voir Figure 4.4) :

$$\dot{\mathbf{q}}_1 = A\mathbf{v}_1 - B\Gamma_{1,2} \quad \mathbf{u}_1 = \mathbf{v}_1 - A\Gamma_{1,2} \quad (4.6)$$

$$\dot{\mathbf{q}}_2 = A\mathbf{u}_2 - B\Gamma_{2,1} \quad \mathbf{v}_2 = \mathbf{u}_2 - A\Gamma_{2,1} \quad (4.7)$$

$$I = 2(A^{-1})^2 B \quad (4.8)$$

La formulation est sensiblement différente de celle issue de la littérature (voir Section 1.3.3 et Equation 1.43). En effet, contrairement au cas de la téléopération où le robot maître et le robot esclave ont des rôles différents, les différents moteurs physiques ont ici un comportement similaire. En inversant le signe de  $\Gamma_{2,1}$ , la formulation originale [MB02] est obtenue.

En l'absence de filtrage (soit  $F_1 = F_2 = 1$ ), les variables d'onde sont liées par :

$$\mathbf{u}_2(t) = \mathbf{u}_1(t - T_1) \quad \mathbf{v}_1(t) = \mathbf{v}_2(t - T_2) \quad (4.9)$$

Afin d'assurer la convergence en position des objets répliqués, l'intégrale des variables d'onde et les positions articulaires sont exprimées sur le moteur 1 à partir de l'Equation 4.6 par :

$$\begin{aligned} \int_0^t \mathbf{q}_1(\tau) d\tau &= \mathbf{q}_1 = A\mathbf{V}_1 - B \int_0^t \Gamma_{1,2}(\tau) d\tau \\ \int_0^t \mathbf{u}_1(\tau) d\tau &= \mathbf{U}_1 = \mathbf{V}_1 - A \int_0^t \Gamma_{1,2}(\tau) d\tau \end{aligned} \quad (4.10)$$

De la même manière, l'intégrale de variables d'onde  $\mathbf{V}$  et les positions articulaires  $\mathbf{q}_2$  sont facilement exprimées sur le moteur physique 2 à partir de Equation 4.7.

Avec ces formulations, les intégrales de variables d'onde dépendent des intégrales des efforts articulaires. De la même manière que pour l'intégration des positions, ces valeurs peuvent dériver de leurs valeurs théoriques à cause d'approximations numériques. Ce phénomène peut s'amplifier avec de grands couples articulaires ou sur de longues périodes.

Afin d'éviter une dérive due à l'intégration de l'effort, les positions articulaires dans un moteur physique  $i$  peuvent être exprimées à partir des positions articulaires dans le moteur  $j$  et une intégrale finie de  $\Gamma$  [NS98] (voir Figure 4.5). A partir des équations 4.9 et 4.10, les intégrales de variables d'onde sont éliminées :

$$\mathbf{q}_1(t) = \int_0^t \dot{\mathbf{q}}_1(\tau) d\tau \quad (4.11)$$

$$\mathbf{q}_2(t) = 2\mathbf{q}_1(t - T_1) - \mathbf{q}_2(t - T_R) - B \int_{t-T_R}^t \Gamma_2(\tau) d\tau \quad (4.12)$$

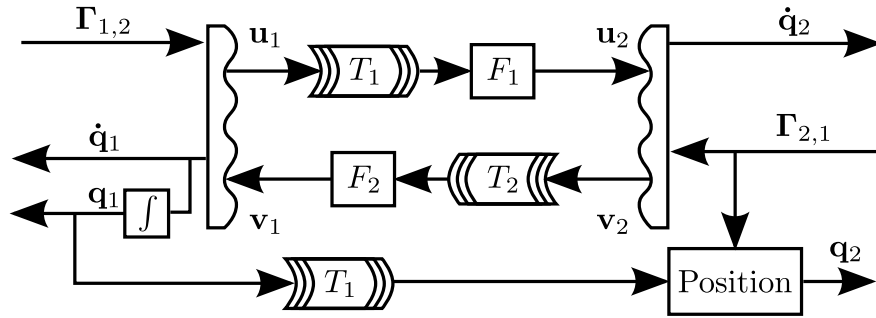


Figure 4.5 – Variables d'onde avec suivi en position.

Il y a donc une position issue d'une intégration,  $\mathbf{q}_1$  et une position *décodée*,  $\mathbf{q}_2$

#### 4.1.4 Filtrage des réflexions d'onde

Les variables d'onde sont sujettes à des oscillations, appelées réflexions d'onde (voir Section 1.3.3.1. Parmi les solutions proposées, celle utilisant l'adaptation de l'impédance [NS97a] est difficile à mettre en œuvre à cause du changement d'impédance des simulations (mouvement libre, contacts, etc.). En outre, une impédance donnée fixe le compromis effort/vitesse inhérent aux variables d'onde. Nous préférons conserver la possibilité de fixer ce compromis en fonction de la divergence tolérée et du retour d'effort voulu (voir Figure 4.11 pour les effets de l'impédance  $b$ ). Nous utilisons donc la seconde méthode qui consiste à filtrer les ondes échangées, car elle est indépendante de l'impédance du système [TN05].

Les résultats utilisant les filtres à horizon fini et infini sont similaires. Nous présentons par la suite les résultats avec le filtre à horizon fini.

Afin de garantir la passivité de la transformation en variables d'onde avec le contrôle de la dérive, le décodage de la position dans l'Equation 4.12 doit être écrit à l'aide de l'Equation 1.50. Trois dispositions du filtre sont possibles, il y a donc trois expressions différentes possibles de la position décodée (voir Figure 4.5) :

- Filtre proche du décodage de la position,  $F_1 = G$ ,  $F_2 = 1$  :

$$\mathbf{q}_2(t) = \mathbf{q}_1(t - T_1) - B \int_{t-T_R}^t \mathbf{\Gamma}_2(\tau) d\tau \quad (4.13)$$

- Filtre proche de l'intégration,  $F_1 = 1$ ,  $F_2 = G$  :

$$\begin{aligned} \mathbf{q}_2(t) = & 2\mathbf{q}_1(t - T_1) - \mathbf{q}_1(t - T_R - T_1) \\ & - B \int_{t-T_R}^t \mathbf{\Gamma}_2(\tau) d\tau \end{aligned} \quad (4.14)$$

- Filtrage bilatéral,  $F_1 = G$ ,  $F_2 = G$  :

$$\begin{aligned} \mathbf{q}_2(t) = & \mathbf{q}_1(t - T_1) - \frac{1}{2}\mathbf{q}_1(t - T_R - T_1) \\ & - B \int_{t-T_R}^t \mathbf{\Gamma}_2(\tau) d\tau - \frac{1}{4}B \int_{t-2T_R}^{t-T_R} \mathbf{\Gamma}_2(\tau) d\tau \\ & + \frac{3}{4}\mathbf{q}_2(t - T_R) - \frac{1}{4}\mathbf{q}_2(t - 2T_R) \end{aligned} \quad (4.15)$$

Seul le dernier cas permet d'obtenir un comportement symétrique du système lors d'un contact (voir Figure 4.6d), mais il nécessite plus de calcul de la part de la simulation. Afin de

retarder les variables  $\mathbf{q}$  et  $\int \mathbf{\Gamma}$ , Tanner et Niemeyer [TN05] ont proposé de les envoyer sur le réseau afin qu'elles soient exactement retardée du retard que subit le système. L'avantage de cette option est que le système n'a pas à avoir de connaissance a priori du retard. Néanmoins cette méthode consomme de la bande passante, particulièrement pour le cas du filtrage bilatéral où cinq variables doivent être retardées du retard  $T_r$ . En pratique, nous avons choisi de retarder ces variables en utilisant un tampon circulaire (*ring buffer*) associé à une estampille. L'estampille est associée à chaque donnée émise. Une copie de ces données est conservée en mémoire dans le tampon. L'estampille est lue lors de la réception des données, elle permet de mesurer le retard que subissent les données échangées et le retard  $T_r$  est ainsi évalué. Connaissant  $T_r$ , il est alors possible de récupérer les valeurs retardées de  $T_r$  en mémoire.

## 4.2 Expérimentation sur la plateforme collaborative

### 4.2.1 Simulation

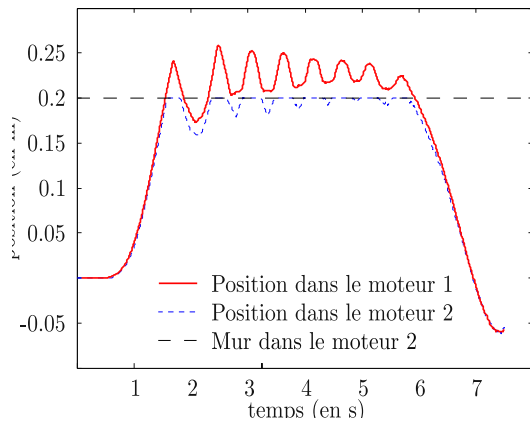
Tout d'abord, des simulations ont été réalisées (voir Figure 4.6) afin de valider l'apport du filtrage et d'évaluer l'impact des différents choix possibles quand au placement du ou des filtres. Le retard était de 100 ms dans les deux sens ( $T_r = 200$  ms). Lors de ces simulations, un objet est déplacé le long d'un axe dans l'un des deux moteurs physiques et il entre en contact dans le second moteur physique (ce cas présente le plus d'oscillations). Pour ce cas à 1 degré de liberté, toutes les raideurs ont été fixées à  $1500 \text{ Nm}^{-1}$  et l'amortissement à  $5 \text{ Nm}$ . La valeur de l'amortissement  $b$  de la transformation en variables d'onde a été fixée à  $6 \text{ Nm}^{-1}\text{s}$ . Ces valeurs ont été choisies afin d'être similaires à celles utilisées sur la plateforme collaborative. La valeur de  $b$  présente un bon compromis pour un retard total de 200 ms.

Sans aucun filtrage, bien que le suivi soit bon en environnement libre, de nombreux rebonds apparaissent entre le début d'un contact et la fin du maintien en contact (voir Figure 4.6a). En ajoutant un filtre à horizon infini à proximité du décodage de la position, (Figure 4.6b) les rebonds sont réduits. Ils disparaissent dans l'une des deux simulations et il y a un rebond dans la seconde simulation. Si le filtre est placé à coté de l'intégration (Figure 4.6c), les rebonds sont quasiment entièrement filtrés. Cependant, la transition entre le déplacement libre et le déplacement contraint est moins franche que dans le cas de l'utilisation de deux filtres (Figure 4.6d). Une pente peut être observé pour les deux cas à un seul filtre, entre 1,5 s et 2,5 s. Dans le cas de l'utilisation d'un seul filtre, le comportement du système n'est plus symétrique. Les résultats divergent selon la position du mur (dans le moteur physique 1 ou 2). De plus, le suivi en position est dégradé par rapport à la solution non filtrée. Dans le cas où deux filtres sont utilisés, il n'y a plus de rebonds et le comportement de la plateforme est symétrique (Figure 4.6d). Mais la disparition des rebonds se fait au dépend du suivi en position. En pratique nous avons privilégié cette dernière solution afin d'obtenir un comportement symétrique et d'éliminer les rebonds intempestifs, mais comme toujours, il s'agit d'un compromis qui peut ne pas être le plus adapté à toutes les tâches collaboratives.

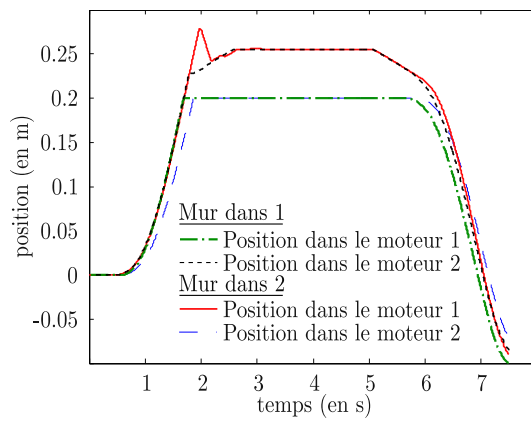
### 4.2.2 Plateforme collaborative

Le couplage utilisant les variables d'onde a été testé sur un scénario de prototypage virtuel consistant en deux pinces manipulées chacune par un utilisateur, et un capot de voiture fixe (Figure 4.7). Le système est constitué d'un moteur physique cadencé à 500 Hz. Les interfaces haptiques utilisées sont des virtuoses 6d. Le médium de communication utilisé est un réseau local où un retard de 200 ms est introduit.

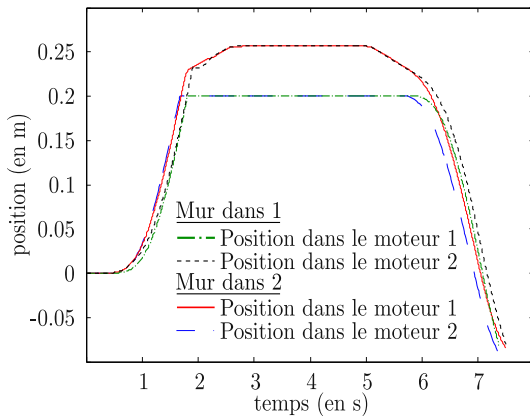




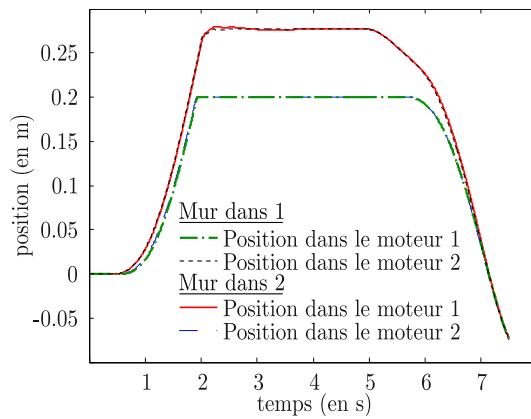
(a) Pas de filtre : Equation 4.10. La ligne pointillée représente la position du mur virtuel



(b) Filtre accolé au décodage de la position : Equation 4.13



(c) Filtrage proche de l'intégration : Equation 4.14



(d) Filtrage bilatéral : Equation 4.15

**Figure 4.6** – Effets du filtrage. Un objet est déplacé le long d'un axe dans un moteur physique  $i$ , il est arrêté par un mur virtuel dans le moteur physique distant  $j$ . Comme l'architecture de la plateforme n'est pas symétrique à cause des variables d'onde, deux essais ont été réalisés pour chaque filtre. La ligne rouge continue et la ligne pointillée bleue correspondent à un mur virtuel dans le moteur physique 2. Pour le second essai, ligne pointillée noire et ligne pointillée verte, le mur virtuel est dans le moteur 1. La position du mur est indiquée par une ligne noire horizontale.

Les variables d'onde sont connues pour modifier les gains de l'asservissement, ainsi qu'introduire une inertie et un amortissement supplémentaire. Lors du réglage de la transformation en variables d'onde, il y a donc toujours un compromis à trouver entre d'une part une grande inertie et une grande raideur, où la divergence en position reste faible, et d'autre part une faible inertie et une faible raideur où la divergence peut devenir grande et les forces difficiles à ressentir par un utilisateur. Les matrices  $A$  et  $B$  de la transformation (voir Equation 1.43) doivent donc être réglées de manière à conserver une raideur correcte tout en conservant un écart en position faible.

Quelques essais de manipulation ont été menés sur la plateforme, où deux utilisateurs ont essayé d'attraper la pince à souder de l'autre utilisateur puis de la pousser ou de la tirer. Avec les réglages précédents, les sujets pouvaient se déplacer en mouvement libre sans trop d'effort, mais ils ne pouvaient pas distinguer clairement les forces dues à l'inertie et à

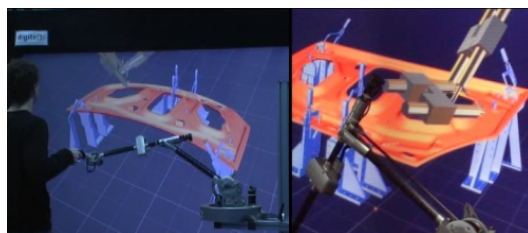


Figure 4.7 – Deux utilisateurs manipulent chacun une pince à souder

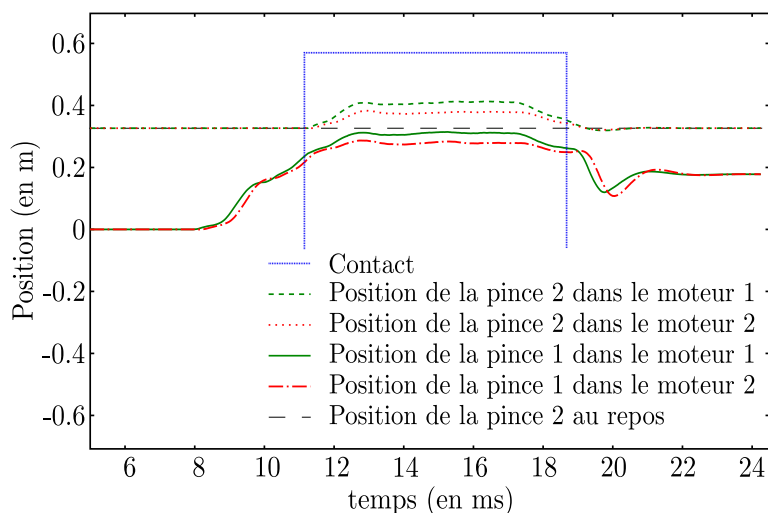
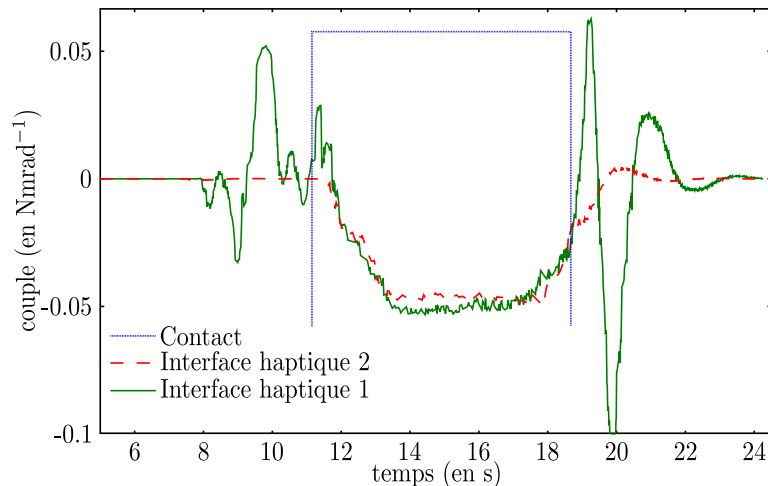


Figure 4.8 – Suivi en position selon l'axe  $x$  pour le cas des pinces à souder.

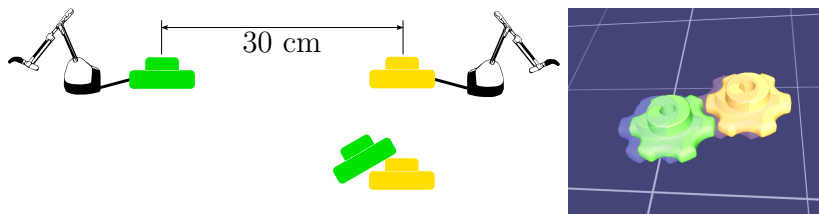
la viscosité de celles dues au contact entre les pinces. Sur la Figure 4.9, les amplitudes des efforts au contact et en mouvement libre sont sensiblement similaires. Si un utilisateur essaie de conserver son interface haptique en position fixe, il peut facilement ressentir les efforts d'un second utilisateur. De plus, le contrôle de la dérive permet bien de garantir la convergence des instances des objets partagés au repos (Figure 4.8). L'ordre de grandeur du retard (200 ms) est à comparer à ceux présents dans la table 3.1. Ce retard est notamment supérieur à ce qui peut être rencontré dans le cas d'une connexion transatlantique.

### 4.2.3 Influence du retard

Enfin, nous avons étudié l'influence du retard sur le comportement de la plateforme sur une tâche simple, afin d'évaluer l'évolution de la divergence et des efforts lorsque le retard croît. Dans cette tâche, deux utilisateurs manipulent un bouton de 5 cm de large. Ces boutons sont séparés de 30 cm. Alors qu'un des deux utilisateurs essaie de conserver la position du bouton qu'il manipule, le second utilisateur essaie de le bouger en appuyant dessus. Nous avons mesuré le torseur cinématique équivalent à celui transmis par l'environnement virtuel à l'interface haptique via le mécanisme virtuel sur le moteur physique de l'utilisateur qui déplace le bouton. Trois valeurs d'amortissement  $b$  de la transformation en variables d'onde ont été utilisées : une valeur centrale  $b = 6 \text{ Nm}^{-1}\text{s}$  correspondant aux réglages que nous avons utilisés dans le cas précédent et une valeur supérieure et inférieure,  $b = 0.5 \text{ Nm}^{-1}\text{s}$  et  $b = 60 \text{ Nm}^{-1}\text{s}$ . Les différents résultats pour des retards de 100, 200, 400 et 600 ms sont présentés dans les figures 4.11, 4.12, 4.13 et 4.14 respectivement. Les valeurs de divergence en position et d'effort sont présentées respectivement côte à côte sur les mêmes graphiques par soucis de lisibilité. Ces différents essais n'ont pas été réalisés de manière continue. La durée du contact entre les boutons est indiquée par la courbe verte sur chacune des figures.



**Figure 4.9** – Couples sur chacune des interfaces haptiques selon l'axe le plus actif. Les efforts en mouvement libre et au contact ont le même ordre de grandeur, ce qui peut rendre difficile pour un utilisateur de distinguer le contact lors d'un mouvement.

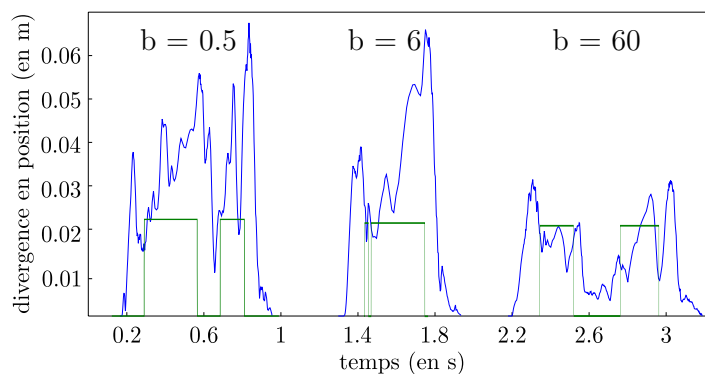


**Figure 4.10** – Un utilisateur maintient son bouton en position pendant que le second essaie de le pousser

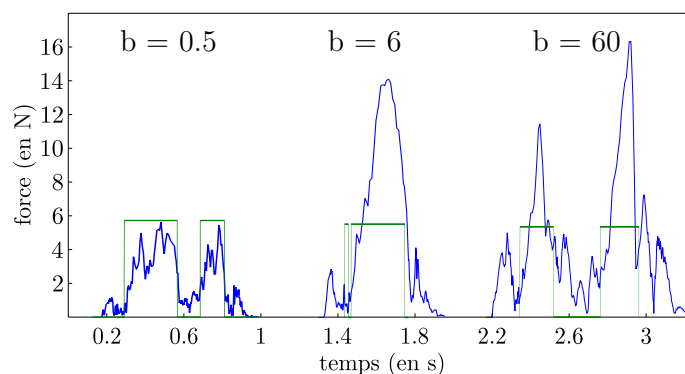
La divergence augmente avec le retard et diminue avec un amortissement  $b$  important. À partir d'un retard de 400 ms la divergence en position représente la moitié du chemin à parcourir entre les deux boutons. Les efforts nécessaires lors du mouvement libre dépassent 5 N au minimum pour la valeur  $b$  la plus faible. Le compromis entre des efforts de contacts importants et une faible divergence est net quelque soit le retard. Les efforts en mouvement libre deviennent aussi importants que les efforts au contact pour un retard total de 200 ms pour la valeur de  $b$  la plus élevée. Pour un retard de 600 ms, seules les conditions  $b = 6$  et  $b = 60$  sont présentées. En effet, la condition  $b = 0.6$  introduisait une divergence telle que le contact était difficile à établir.

Globalement, au delà de 200 ms le couplage utilisant des variables d'onde autorise une divergence très importante. Dans le cas d'une valeur  $b$  importante, la divergence reste limitée mais les efforts en mouvement libre provoquent rapidement une fatigue musculaire importante. Cet asservissement ne peut être facilement utilisé pour des retards trop importants dans le cadre d'une tâche de prototypage virtuel.

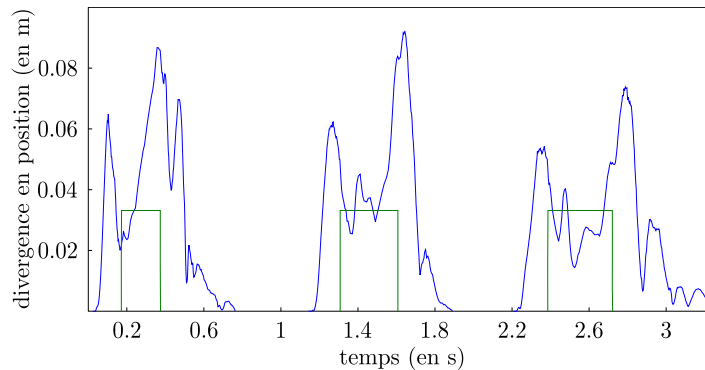
Enfin, nous avons comparé pour un retard de 50 ms, l'asservissement avec variables d'onde (Figure 4.15) avec l'asservissement proportionnel-dérivé proposé au chapitre précédent (Figure 4.16). Quelque soit la valeur de  $b$ , la divergence en position lors du contact est plus grande pour la transformation en variables d'onde que pour l'asservissement proportionnel-dérivé. En mouvement libre, la divergence est équivalente uniquement pour la valeur de  $b$  la plus importante. Pour des retards faibles de l'ordre de 50 ms l'asservissement présenté au chapitre précédent offre de meilleure performance en termes de suivi en position et de transparence lors d'un mouvement libre. Ces résultats rejoignent ceux obtenus dans [SH06] concernant les meilleures performances de suivi en position pour l'asservissement ressort amortisseur. En



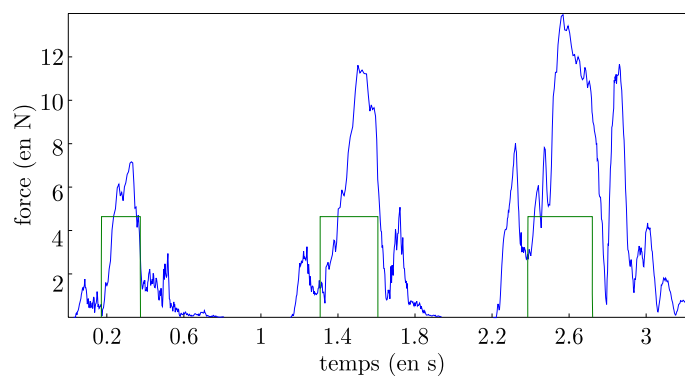
(a) Divergence en position



(b) Effort appliqué sur l'interface haptique

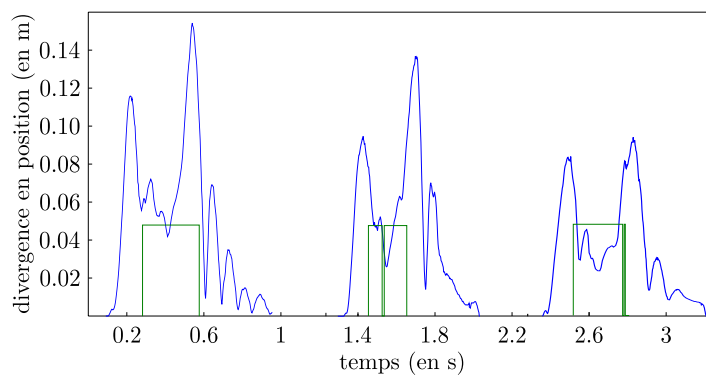
**Figure 4.11** – Divergence et effort pour un retard de 100ms

(a) Divergence en position

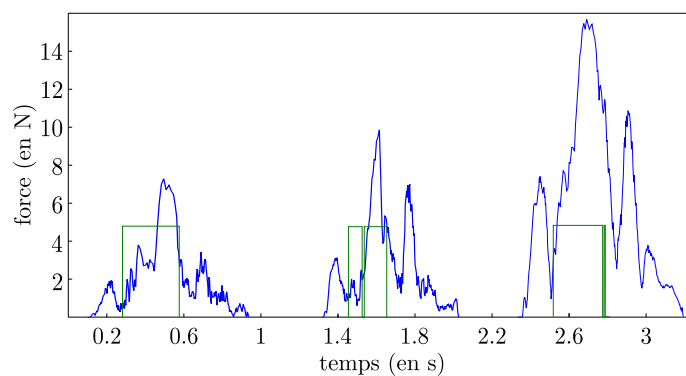


(b) Effort appliqué sur l'interface haptique

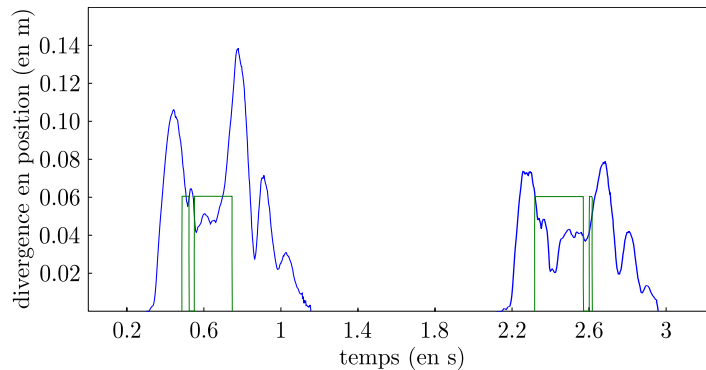
**Figure 4.12** – Divergence et effort pour un retard de 200ms



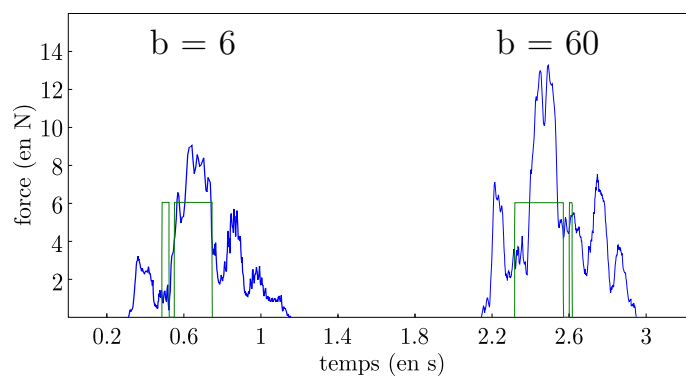
(a) Divergence en position



(b) Effort appliqué sur l'interface haptique

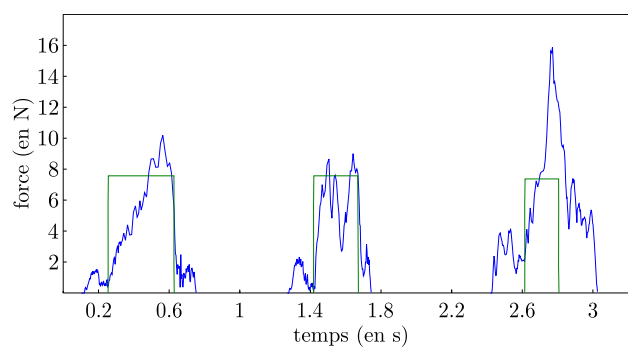
**Figure 4.13** – Divergence et effort pour un retard de 400ms

(a) Divergence en position

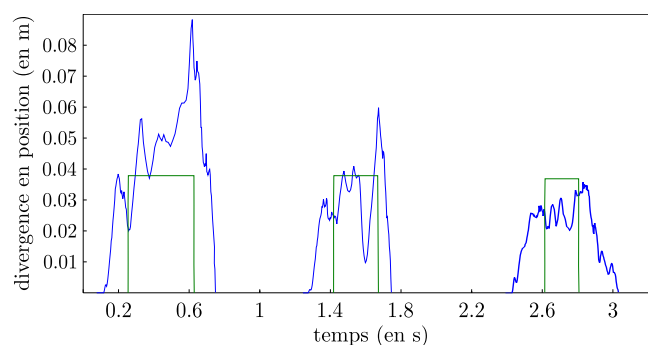


(b) Effort appliqué sur l'interface haptique

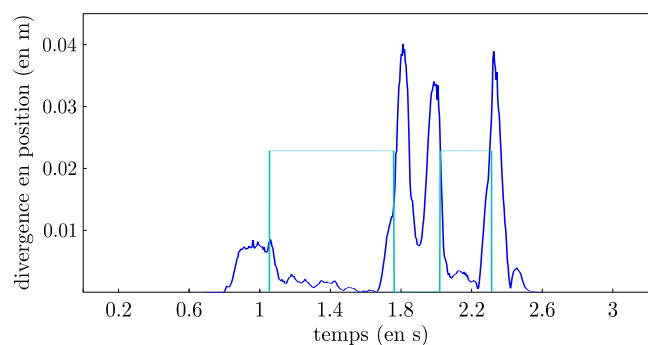
**Figure 4.14** – Divergence et effort pour un retard de 600ms



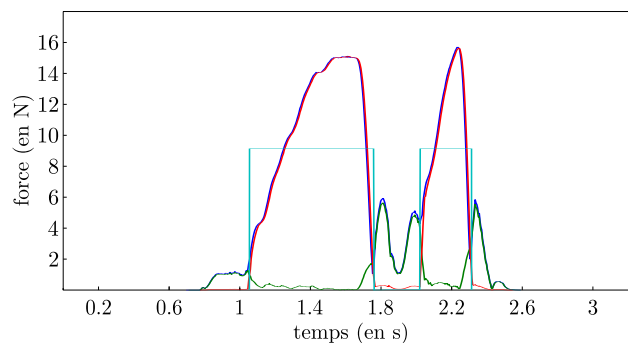
(a) Divergence en position



(b) Effort appliqué sur l'interface haptique

**Figure 4.15** – Divergence et effort pour un retard de 50ms avec variables d'onde

(a) Divergence en position



(b) Effort appliqué sur l'interface haptique

**Figure 4.16** – Divergence et effort pour un retard de 50ms avec un couplage ressort amortisseur et la compensation activée

ce qui concerne les efforts, Sankaranarayanan et Hannaford [SH06] ont observé un meilleur suivi en effort pour les variables d'onde lors de la comanipulation d'un objet. Dans le cas de la manipulation d'objet distinct, l'asservissement en variables d'onde présente de faible performance en mouvement libre, principalement à cause des effets inertiels.

## Conclusion

Dans ce chapitre, nous avons proposé un couplage entre des moteurs physiques permettant d'assurer la stabilité du système pour un retard constant quelconque. En utilisant une formulation dans l'espace articulaire de la transformation en variables d'onde ainsi que de son intégrale associée à un contrôle de la dérive, nous avons pu réaliser des manipulations sur des scénarios de prototypage virtuel complexes en présence de délais significatifs (200 ms au total). Néanmoins, lorsque le retard croît au delà de 200 ms, la divergence en position devient importante et les efforts en mouvement libre peuvent produire une fatigue physique importante. De plus, ils ont un ordre de grandeur similaire à ce qui peut être rencontré lors d'un contact. Un utilisateur peut alors avoir du mal à distinguer la transition entre le mouvement libre et le mouvement contraint.

Enfin, pour des retards faibles, les asservissements utilisant les variables d'onde semblent être moins performants que celui que nous avons proposé, basé sur des couplages virtuels ressorts-amortisseurs avec compensation d'effort. Cet asservissement permet donc la réalisation d'une tâche collaborative lorsque l'asservissement proportionnel dérivé présente un domaine de stabilité trop petit, mais il ne peut être utilisé pour des retards trop grands.

## Perspectives

Les variables d'onde ont bénéficié de plusieurs travaux leur permettant de stabiliser un réseau soumis à du retard variable voir à des pertes de paquets. L'implémentation de ces méthodes sur la plateforme collaborative actuelle permettrait d'étudier la validité d'une collaboration via Internet, dans la limite de retard de 200 ms où cette approche reste utilisable au sein d'environnements collaboratifs.

# Retour haptique événementiel

## Sommaire

<b>5.1 Motivation</b>	<b>101</b>
<b>5.2 Calcul du rendu haptique événementiel</b>	<b>103</b>
5.2.1 Surévaluation de l'impact	103
<b>5.3 Expérimentation</b>	<b>105</b>
5.3.1 Objectif	105
5.3.2 Sujets	106
5.3.3 Dispositif expérimental	106
5.3.4 Procédure	107
<b>5.4 Résultats</b>	<b>109</b>
5.4.1 Résultat des efforts	109
5.4.2 Résultat pour les erreurs de manipulation	111
5.4.3 Résultats pour le temps de réalisation	112
<b>5.5 Discussion</b>	<b>112</b>
5.5.1 Effets du retour haptique événementiel	113
<b>Conclusion</b>	<b>114</b>

L'utilisation de retour d'effort au sein d'une plateforme collaborative est toujours soumise à un compromis. Aucune méthode ne permet d'obtenir un retour transparent en présence de retard. D'une part les gains des asservissements peuvent diminuer avec le retard, d'autre part des efforts parasites sont susceptibles d'être introduits. Le retour haptique se trouve ainsi dégradé.

Partant de ce constat, nous proposons d'utiliser du retour haptique événementiel, afin de compléter l'information fournie par le retour d'effort traditionnel. Nous souhaitons, en utilisant cette méthode, permettre à l'utilisateur de mieux comprendre l'univers virtuel avec lequel il interagit, particulièrement en ce qui concerne l'information de contact.

Dans ce chapitre nous proposons d'évaluer le bénéfice que peut apporter l'haptique événementielle lors d'une manipulation dans un environnement où le retour d'effort est dégradé. Pour ce faire nous proposons une expérience simple où un seul utilisateur doit s'aider uniquement du retour d'effort afin de réaliser une tâche.

Dans une première partie (Section 5.1) nous présentons les motivations qui nous amènent à proposer ce type de retour dans le cas d'une plateforme collaborative. Ensuite, l'algorithme permettant de réaliser ce retour haptique événementiel est décrit (Section 5.2). Une expérimentation permettant d'évaluer le bénéfice de ce type de rendu a été réalisée (Section 5.3). Les résultats de cette expérience sont ensuite présentés (Section 5.4) et discutés (Section 5.5).

## 5.1 Motivation

Dans les deux chapitres précédents, nous avons introduits des asservissements (proportionnels dérivés dans le chapitre 3 et variables d'onde dans le chapitre 4) afin d'assurer la



convergence d'instances d'objets partagés entre différents nœuds. Ces différentes techniques ainsi que la méthode observateur de passivité et contrôleur de passivité (OP/PC) introduisent des efforts parasites, comme des efforts de traînée (voir Figure 3.10), des inerties (voir Figure 4.9) ou des viscosités supplémentaires. Dans le cas des contrôleurs proportionnel dérivé, les gains sont d'autant plus faibles que le retard est grand (voir Figure 3.9). Des gains faibles réduisent la raideur perçue par les utilisateurs.

Dans le cas d'un environnement non collaboratif, le retour d'effort peut aussi être dégradé pour deux raisons. Tout d'abord, si la scène virtuelle simulée s'avère particulièrement grande, il peut être nécessaire de diminuer la fréquence du simulateur. Si le retour d'effort est réalisé avec des couplages virtuels, les gains de ces derniers doivent être réduits. Si le retour d'effort est un rendu direct, la raideur du contact doit être réduite, car, la stabilité du système dépend du pas de temps de la simulation et cette raideur (Equation 1.11) [GSH<sup>+</sup>07]. Ensuite, les détecteurs de collision discrets peuvent introduire une viscosité afin de limiter la vitesse maximum des objets virtuels dans une scène. En effet, des objets trop rapide peuvent traverser d'autres objets de la scène. Cet ajout permet donc d'éviter des interpénétrations ou les effets passe-muraille (Section 1.1.1).

Les différents effets introduits par les méthodes précédemment décrites sont résumés dans la table 5.1.

Type	Effets
Variables d'onde [NS91]	ou { grande raideur, grande inertie faible raideur, faible inertie
OP/PC [HR02]	amortissement supplémentaire lorsque que le système est actif
Couplage virtuel [CSB95]	ou/et { raideur faible amortissement supplémentaire [LS06]
Rendu direct [CGSS93]	amortissement supplémentaire
Détection de collision discrète [MPT06, JW03]	amortissement supplémentaire

**Table 5.1** – Résumé des différents effets induit par les méthodes couramment rencontrée dans les environnements virtuels en particulier les environnements collaboratifs.

Des études ont été menées pour évaluer l'influence de ces méthodes d'un point de vue perceptif. Notamment, la manière de calculer le rendu haptique (rendu direct, couplage virtuel, etc.) ainsi que les gains utilisés par ces méthodes ont tendance à modifier la perception d'un mur virtuel [RA93]. Cependant, la perception de la dureté du contact n'est pas nécessairement induite par les propriétés mécaniques (raideur et amortissement) de l'interface haptique, mais est liée au rapport entre la variation initiale d'effort et la vitesse de pénétration au moment de l'impact. Lawrence *et al.* ont formalisé ce rapport qu'ils ont nommé *rate hardness* [LPD<sup>+</sup>00]. En se focalisant ainsi sur la perception du matériau, l'haptique événementielle permet d'augmenter la raideur perçue d'un objet en ajoutant une force transitoire haute fréquence au retour haptique traditionnel lors d'un impact [OHDC00] (voir Section 1.2.2.3). Ces forces transitoires permettent aussi de fournir une information réaliste sur le matériau d'un objet virtuel [KFN06] ou d'informer de la position d'un impact [SLAA09].

Les gains utilisés par l'interface haptique sont donc seulement un des éléments associés à la perception de la raideur ou du matériau d'un objet virtuel. Ils ont aussi une influence sur la perception de la taille d'un objet virtuel [OG02]. En particulier, des gains faibles dégradent la perception de la taille. Bien que de tels gains soient généralement considérés

comme non réalistes et ont tendance à dégrader les performances des utilisateurs, certaines tâches spécifiques, comme des tâches de pointages, semblent en tirer parti [PS09].

Nous souhaitons dans ce chapitre évaluer l'apport de ces techniques permettant de transmettre des informations hautes fréquences, dans des environnements où le retour d'effort est dégradé comme ceux rencontrés dans les chapitres précédents. L'approche que nous proposons se distingue de celle usuellement adoptée lors de l'utilisation d'haptique événementielle, car nous ne cherchons pas nécessairement à rendre la tâche plus réaliste mais à améliorer les performances des utilisateurs sur une tâche avec un retour d'effort dégradé. Cette approche ne passe donc pas forcément par un retour haptique plus réaliste mais par des informations mises en valeur de manière plus nette. Nous avons ainsi choisi d'utiliser cette méthode pour mettre en évidence les impacts.

## 5.2 Calcul du rendu haptique événementiel

L'haptique événementielle consiste à superposer un effort transitoire haute fréquence au retour haptique traditionnel au moment de l'un impact. Le torseur d'effort généré au moment de l'impact  $W_i$  et le torseur d'effort transitoire résultant  $W_t$ , sont calculés en utilisant l'algorithme proposé par [SBL<sup>+</sup>07] :

$$\begin{cases} W_i^h = \text{Ad}_{H_h^p}^T W_i^p \\ W_i^p = \begin{pmatrix} 0 \\ b_i \mathbf{v}_n \end{pmatrix} \end{cases} \quad (5.1)$$

où  $\mathbf{v}_n$  est la vitesse normale au point d'impact  $\mathbf{p}$ ,  $b_i$  est le coefficient de restitution de l'impact. L'effort est calculé au niveau du point d'attache grâce à l'adjoint  $\text{Ad}_{H_h^p}^T$ .  $H_h^p$  est la transformation homogène du repère  $p$  attaché au point d'impact au repère  $h$  lié à l'interface haptique.

Le torseur d'effort transitoire résultant  $W_t^h$  est ensuite obtenu en multipliant le torseur  $W_i^h$  par une sinusoïde décroissante. Le torseur transitoire obtenu est ensuite échantillonné à la fréquence de la boucle de contrôle haptique ( $\Delta t = 1$  kHz).

$$W_t^h(k) = W_i^h a e^{-k\lambda\Delta t} \sin(k\omega\Delta t) \quad (5.2)$$

où l'amplitude  $a$ , la fréquence  $\omega$  et le coefficient de décroissance  $\lambda$  sont des grandeurs caractéristiques de l'impact [OHDC01].

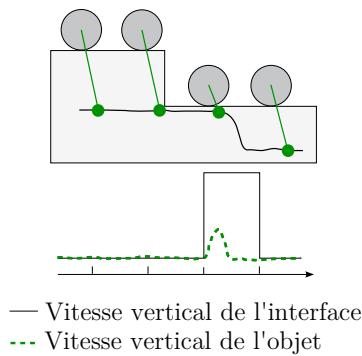
Le torseur d'effort résultant  $W_{hd}^h$  envoyé au contrôleur par la simulation physique est la somme du torseur d'effort  $W_{cv}^h$  calculé en boucle fermée par l'asservissement proportionnel dérivé et du torseur d'effort transitoire  $W_t^h(k)$  :

$$W_{hd}^h(k) = W_{cv}^h(k) + W_t^h(k) \quad (5.3)$$

### 5.2.1 Surévaluation de l'impact

Lors de la manipulation d'un objet virtuel, des artéfacts haptiques sont susceptibles d'apparaître en utilisant la méthode précédemment décrite. Ces artéfacts sont dus au couplage virtuel utilisé entre la simulation physique et l'interface haptique. Ce couplage peut en effet générer de grandes vitesses pour des mouvements de petites amplitudes. Ces grandes vitesses contribuent à générer des impacts importants (voir Equation 5.1) qui sont amenés à être plus fort que ceux auxquels s'attendent les utilisateurs. Ces impacts sont alors déconcertants.

Lorsqu'un impact se produit entre deux objets en mouvement libre dont l'un est manipulé via une interface haptique, la vitesse de l'objet manipulé est sensiblement similaire à la vitesse fournie par l'interface haptique<sup>1</sup>. Ce n'est plus forcément le cas lors d'un mouvement contraint. Notamment, lors d'un mouvement le long d'une surface, si l'utilisateur pousse l'objet virtuel contre la surface, les positions de l'objet virtuel et la consigne de l'interface haptique diffèrent. Dans ce cas, l'utilisateur ressent un effort généré par le couplage virtuel, le contraignant le long de la surface. S'il y a un décrochement net sur cette surface (une marche par exemple) (voir Figure 5.1), la position de l'objet tend à converger vers la position de l'interface haptique. Si l'objet possède une masse faible (ou pas de masse dans le cas d'une formulation quasi-statique [MPT06]), la convergence en position se fait presque instantanément ce qui crée une grande vitesse de l'objet. Ce phénomène est amplifié si les gains du couplage virtuel sont faibles ou si l'utilisateur exerce un effort important contre la surface.



**Figure 5.1** – Lors d'un mouvement le long d'une surface, la vitesse de l'objet virtuel peut devenir grande et générer un impact ayant une forte amplitude. Une amplitude trop forte par rapport au mouvement effectué par l'utilisateur peut être déconcertante.

Afin de limiter cet effet, la vitesse utilisée pour calculer le torseur d'effort d'impact  $W_i$  est bornée par une vitesse proportionnelle à la vitesse fournie par l'interface haptique au point d'impact. Cette vitesse est exprimée le long la normale  $\mathbf{n}$  au contact :

$$\begin{pmatrix} \boldsymbol{\omega}_{hd} \\ \mathbf{v}_{hd} \end{pmatrix} = \text{Ad}_{H_h^p} T_{hd}^h$$

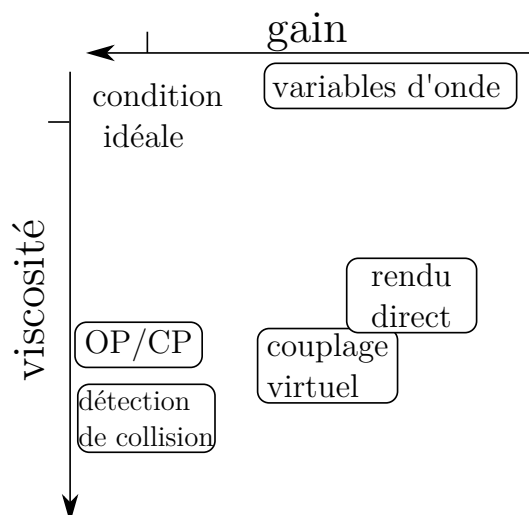
$$\mathbf{v}_{nhd} = (\mathbf{v}_{hd} \cdot \mathbf{n}) \mathbf{n}$$

$$\mathbf{v}_b = \max(\|\mathbf{v}_n\|, \alpha \|\mathbf{v}_{nhd}\|) \mathbf{n} \quad (5.4)$$

$$W_i^p = \begin{pmatrix} 0 \\ b_i \mathbf{v}_n \end{pmatrix} \quad (5.5)$$

En essayant cette méthode, nous avons réglé expérimentalement  $\alpha$  à 1.5. Ce réglage permet d'obtenir un retour événementiel qui nous semble réaliste.

1. Un coefficient de proportionnalité peut être utilisé au niveau du couplage virtuel. Bien que le cas ne soit pas considéré en l'occurrence, la surévaluation de l'impact est aussi susceptible de produire des artefacts haptiques



**Figure 5.2** – Résumé des effets de différentes méthodes utilisées en réalité virtuelle ou en téléopération sur les gains de l'interface haptique (axe horizontal) et sur la viscosité globale (axe vertical). La condition idéale correspond aux gains maximum et a une viscosité globale nulle.

## 5.3 Expérimentation

### 5.3.1 Objectif

Le but de cette expérience est d'évaluer les effets du rendu haptique événementiel lors de manipulations dans un lorsque ce retour d'effort est dégradé. Nous nous sommes limités aux environnements virtuels qui possèdent une viscosité globale non nulle ou qui utilisent des gains faibles au niveau de leur couplage virtuel. Les méthodes abordées dans la section 5.1 provoquant de tels effets sont résumés sur la Figure 5.2. L'influence de l'inertie modifiée des objets par les variables d'onde n'a pas été étudiée lors de cette expérience.

Plusieurs conditions correspondant à trois réglages de gains et trois viscosité globales différentes ont été testées avec et sans retour haptique événementiel. Ces conditions ont été choisies afin de refléter les conditions qui peuvent être rencontrées lors de l'utilisation des méthodes présentée dans la Figure 5.2.

Notre hypothèse principale est que l'ajout d'un retour haptique événementiel au retour d'effort traditionnel permet aux utilisateurs de mieux comprendre l'environnement virtuel avec lequel ils interagissent. En effet, ce retour événementiel permet de mieux percevoir les impacts lorsque le retour d'effort traditionnel est dégradé. Cette hypothèse peut être découpée en trois points :

1. Les sujets appliquent des efforts moins importants pour détecter une situation de contact avec un retour haptique événementiel.
2. Les différences de performance selon les conditions avec et sans haptique événementielle sont plus significatives lorsque le retour d'effort est plus dégradé (gain plus faible et/ou viscosité globale plus importante.)
3. Comme les sujets perçoivent mieux l'environnement virtuel, ils effectuent leur tâche plus rapidement.

Des hypothèses secondaires peuvent être faites concernant la viscosité globale et l'influence du réglage des gains du couplage virtuel :



Figure 5.3 – Dispositif technique

1. Un environnement plus visqueux augmente le temps de réalisation de la tâche. De plus les sujets ont plus de difficulté à ressentir les contacts.
2. Avec des gains faibles, les sujets peuvent appliquer des efforts moins importants pour détecter des contacts, car un mouvement ample génère des forces plus faibles. De plus, les sujets mettent plus de temps à détecter un contact et donc à réaliser la tâche.

### 5.3.2 Sujets

14 sujets (12 hommes et 2 femmes) âgés de 22 à 35 ans se sont portés volontaires pour cette expérience. Tous les sujets étaient droitiers et n'avaient pas de troubles de perception. Ils étaient tous ignorants quant aux buts de l'expérience. Parmi les sujets, certains possédaient une grande expérience de la manipulation d'objet virtuels via une interface haptique alors que d'autres étaient novices en la matière.

### 5.3.3 Dispositif expérimental

Le dispositif expérimental était constitué d'un moteur physique simulant un environnement virtuel associé à une interface haptique. L'interface haptique était un Virtuose6D™ d'Haption. La simulation utilisée était basée sur le moteur physique XDE développé par le CEA et tournait sur un Intel Core 2 Duo E6700 2.67 GHz. La visualisation était réalisée par Ogre3D 1.4. Le siège était réglé de telle façon que l'avant-bras des utilisateurs était horizontal au début de la manipulation et pas plus haut que la poitrine des sujets lors de la manipulation. Chaque sujet portait un casque permettant de bloquer le bruit ambiant (voir Figure 5.3)

Dans cette étude, seules les translations ont été considérées, de plus nous avons ignorés les effets inertiels. Une formulation quasi-statique a donc été retenue :

$$0 = -\mathbf{B}_g \dot{\mathbf{x}}_{ov} + \mathbf{B}(\dot{\mathbf{x}}_{ih} - \dot{\mathbf{x}}_{ov}) + \mathbf{K}(\mathbf{x}_{ih} - \mathbf{x}_{ov}) \quad (5.6)$$

où  $\dot{\mathbf{x}}_{ov}$ ,  $\mathbf{x}_{ov}$ ,  $\dot{\mathbf{x}}_{ih}$  et  $\mathbf{x}_{ih}$  sont respectivement les vitesses et positions de l'objet virtuel (ov) et de l'interface haptique (ih).  $\mathbf{B}_g$  est un amortissement permettant de simuler une viscosité globale.

$\mathbf{B}$  et  $\mathbf{K}$  sont les gains du couplage virtuel.  $\mathbf{B}_g$ ,  $\mathbf{B}$  et  $\mathbf{G}$  sont choisis afin que l'environnement soit isotrope :

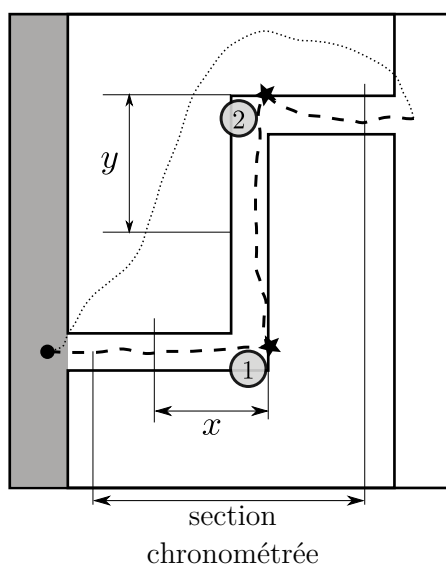
$$\mathbf{B}_g = b_g \mathbf{I}_3, \mathbf{B} = b \mathbf{I}_3, \mathbf{K} = k \mathbf{I}_3 \quad (5.7)$$

avec  $b_g, b, k \in \mathbb{R}$  et  $\mathbf{I}_3$ , la matrice identité.

### 5.3.4 Procédure

Chaque sujet devait déplacer une balle rouge au travers d'un parcours très simple, de gauche à droite (voir Figure 5.4). Il devait effectuer la tâche à l'aveugle en utilisant le retour d'effort. Les sujets ne devaient pas appliquer d'efforts trop importants sur la balle. Cette consigne est imposée afin d'éviter la stratégie la plus rapide qui est d'ignorer le retour d'effort et de pousser la balle en haut à droite. Elle se traduit par une limite d'effort de 5 N. Si un sujet applique plus de 5 N sur la balle, il réalise une erreur de manipulation et la balle devient noire. Cependant, les sujets peuvent voir la balle seulement au début et à la fin de la tâche (voir Figure 5.5). Ils ne peuvent donc réaliser leur erreur uniquement à la fin de chaque essai. En cas d'erreur, ils devaient recommencer l'essai jusqu'à ce qu'ils réussissent à le réaliser sans dépasser la limite d'effort de 5 N. Les forces appliquées par le sujet ne sont mesurées que dans le plan  $xy$ .

Les sujets avaient pour seconde consigne de réaliser le parcours le plus vite possible. Cette consigne est indiquée comme étant moins prioritaire. Le temps de réalisation a été mesuré pour chaque essai, la zone chronométrée est indiquée sur la Figure 5.4. A la fin de chaque essai, réussi ou non, le sujet doit ramener la balle au début du labyrinthe. Chaque essai est prêt à être effectué lorsque le sujet presse la balle contre la zone se situant à gauche de l'entrée du parcours (zone grise dans la Figure 5.4 et verte lors de l'essai, Figure 5.5). L'essai commence réellement lorsque la balle entre dans la section chronométrée. Le parcours a la forme d'un S et ses dimensions horizontale  $x$  et verticale  $y$  sont changées entre chaque essai. Les zones indiquées par 1 et 2 sur la Figure 5.4 signalent le premier et le second coin du parcours où les contacts apparaissent.



**Figure 5.4** – Chaque sujet doit déplacer la balle de gauche à droite au travers du parcours. Les étoiles le long de la trajectoire d'exemple notent l'emplacement des contacts entre la balle et les murs du parcours. La trajectoire en pointillés fins indique la trajectoire de retour de la balle à la fin de l'essai.

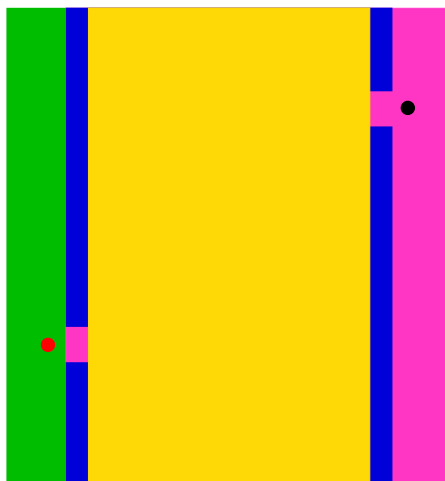
Trois conditions de gains ( $k$ ,  $b$ ) du couplage virtuel, trois conditions de viscosité globale  $b_g$  ainsi que deux conditions d'haptique événementielle (HE) ont été testées. La condition idéale correspond aux gains du couplage virtuel nominaux et à viscosité globale nulle ( $b_g = 0$ ). Les gains nominaux utilisés ont été  $k = 1000 \text{ Nm}^{-1}$  et  $b = 2 \text{ Nm}^{-1}\text{s}$  sur chacun des axes.

Les trois conditions de gains sont 100%, 50% et 10% des gains nominaux. Tous les gains ( $k$ ,  $b$ ) sont multipliés de manière égale par le coefficient de la condition. Dans la suite du chapitre, ces conditions sont notées  $g100$ ,  $g50$  et  $g10$ . Ces conditions de gains sont à rapprocher de celles qui peuvent être rencontrées sur la plateforme collaborative abordée dans le chapitre 3. Ainsi, sans retard, la raideur nominale est voisine de  $1200 \text{ Nm}^{-1}$ , alors qu'avec du retard, les gains utilisés étaient compris entre  $130 \text{ Nm}^{-1}$  et  $80 \text{ Nm}^{-1}$ .

Les trois conditions de viscosité  $b_g$  ont été fixées à 0, 0.05 et  $0.2 \text{ Nm}^{-1}\text{s}$  afin d'avoir une viscosité globale nulle, faible et forte. Ces trois conditions sont notées  $v0$ ,  $v25$  et  $v100$  pour 0%, 25% et 100% de la viscosité la plus élevée. L'haptique événementielle était soit activée HE doit désactivée NHE.  $3 \times 3 \times 2 = 18$  conditions ont été testées au total.

Les paramètres du rendu haptique événementiel ( $a$ ,  $\lambda$  and  $\omega$  dans l'Equation 5.2) ont été fixés afin d'obtenir un impact réaliste pour la condition idéale ( $g100, v0$ ). Nous avons choisi  $a = 20 \text{ s}^{-1}$ ,  $\lambda = 75 \text{ s}^{-1}$  and  $\omega = 80 \text{ Hz}$ . Ces réglages correspondent à ceux déterminés expérimentalement dans [OHDC01] pour un matériau virtuel correspondant à du bois. Seule l'amplitude  $a$  diffère, car le rendu de l'impact dépend à la fois de l'amplitude mais aussi de l'architecture de l'interface haptique. Ces réglages ont été conservés pour toutes les conditions de gains et de viscosité.

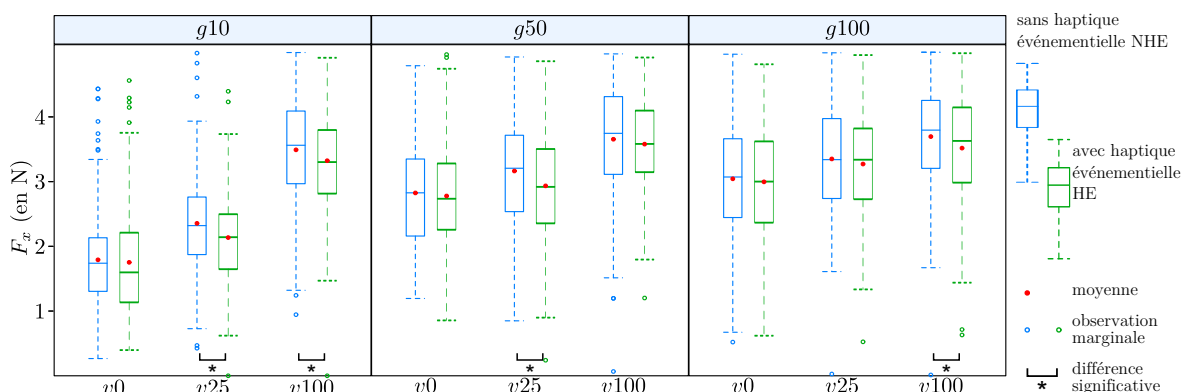
Chaque ensemble de conditions a été testé dans un ordre aléatoire, deux fois, sur 9 configuration du parcours ( $x \times y = \{0, 0.01, 0.02\} \text{ m} \times \{0, 0.01, 0.02\} \text{ m}$ , voir Figure 5.5). Les 18 conditions ont été testées 18 fois soit un total de 324 essais au minimum, c'est-à-dire si le sujet ne commet aucune erreur de manipulation.



**Figure 5.5** – Le parcours effectivement vu par le sujet. Sur la gauche une balle au début de la manipulation. A droite la même balle après une erreur de manipulation.

La limite d'effort de 5 N a été choisie afin que la tâche puisse être réalisée avec succès quelque soit l'ensemble de conditions testées.

Plusieurs paramètres ont été mesurés : la force appliquée à la balle par l'utilisateur le long de la normale au contact ainsi que les positions et vitesses de la balle et de l'interface haptique. La force appliquée par l'utilisateur le long de la normale au contact est déduite de la Figure 5.6 :



**Figure 5.6** – Moyennes et boîtes à moustache pour les efforts maximum le long de l’axe  $x$  pour le coin 1. Les différences significatives entre HE et NHE sont indiquées par un astérisque.

$$\begin{aligned} \mathbf{F}_n &= -\mathbf{B}\dot{\mathbf{x}}_{\text{hd}}\mathbf{n} + \mathbf{k}(\mathbf{x}_{\text{hd}} - \mathbf{x}_{\text{vo}})\mathbf{n} \\ \dot{\mathbf{x}}_{\text{vo}}\mathbf{n} &= 0 \end{aligned} \quad (5.8)$$

Le temps total de l’expérience a duré entre 60 min et 90 min par utilisateur. Cette durée inclut les explications, quelques essais préliminaires et le débriefing.

## 5.4 Résultats

### 5.4.1 Résultat des efforts

Les efforts ont sensiblement le même ordre de grandeur quelque soit les conditions testées. Le maximum des efforts appliqués est donc plus facile à comparer que la vitesse maximum ou l’écart maximum entre les positions de l’objet virtuel et de l’interface haptique. Il y a deux principales régions d’intérêts qui sont les coins 1 (voir Figure 5.4) le long de l’axe  $x$  et le coin 2 le long de l’axe  $y$ . Les boîtes à moustaches relatifs aux efforts sont exposées dans les figures 5.6 et 5.7.

Facteur	dl	x		y	
		valeur de $F$	valeur de $p$	valeur de $F$	valeur de $p$
gain	2	583.18*	$< 10^{-6}$	312.73*	$< 10^{-6}$
visc	2	779.77*	$< 10^{-6}$	463.57*	$< 10^{-6}$
HE	1	31.61*	$< 10^{-6}$	24.9*	$< 10^{-6}$
gain $\times$ HE	2	0.31	0.72	2.04	0.09
visc $\times$ HE	2	3.50	0.03	1.96	0.15
gain $\times$ visc	4	84.04*	$< 10^{-6}$	89.4*	$< 10^{-6}$
gain $\times$ visc $\times$ HE	4	1.11	0.35	0.26	0.26

**Table 5.2** – ANOVA à trois facteurs des efforts pour chaque coin selon les axes  $x$  et  $y$ . Les valeurs significatives sont indiquées par un astérisque.

Une ANOVA à trois facteurs a été réalisée sur les efforts maximum pour les conditions de gains de viscosités et d’haptique événementielle pour chaque coin du parcours. Le résumé de cette analyse est présenté dans la table 5.2.



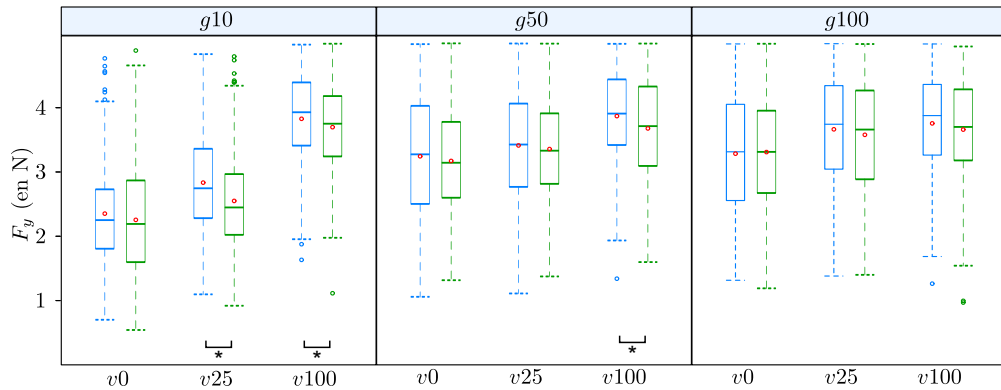


Figure 5.7 – Moyennes et boîtes à moustache pour les efforts maximum le long de l’axe  $y$  pour le coin 2. Les différences significatives entre HE et NHE sont indiquées par un astérisque.

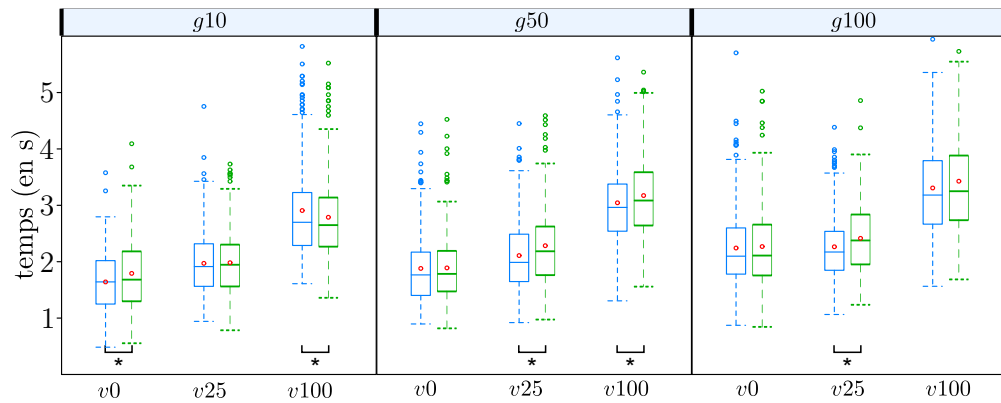


Figure 5.8 – Moyennes et boîtes à moustache pour le temps de réalisation de la tâche. Les différences significatives entre HE et NHE sont indiquées par un astérisque.

	g100	g50	g10		g100	g50	g10		g100	g50	g10		g100	g50	g10				
Time	v0	0.84	0.61	0.0005 ***	$F_x$	v0	0.49	0.45	0.52	$F_y$	v0	0.73	0.33	0.13	erreur de manipulation	v0	0.05	0.34	0.16
	v25	0.001 **	0.001 **	0.81		v25	0.23	0.0007 ***	0.0001 ***		v25	0.25	0.39	$< 10^{-5}$ ***		v25	0.09	0.60	0.58
	v100	0.12	0.043 *	0.048 *		v100	0.005 **	0.61	0.009 **		v100	0.14	0.004 **	0.018 *		v100	0.06	0.06	0.11

Figure 5.9 – Résumé des différences significatives pour des test- $t$  pour le temps,  $F_x$  et  $F_y$  et pour un test de Kruskal-Wallis pour le taux d’erreur de manipulation. Les nombres en gris sont non significatifs. Les cellules possédant un fond grisé indiquent les conditions où l’haptique événementielle améliore les performances.

Le maximum d'effort est plus élevé lorsque la viscosité globale est grande ou lorsque les gains se rapprochent des gains nominaux ( $g100$ ). Ces résultats sont corrélés aux moyennes d'erreur de manipulation présentées dans la table 5.3. Les sujets ont tendance à faire plus d'erreur de manipulation pour ces conditions, c'est à dire qu'ils ont tendance à appliquer plus de 5 N lors de contacts.

L'amplitude maximum des efforts appliqués pour détecter un contact est globalement plus faible lorsque l'haptique événementielle est activée. Il y a des différences significatives pour la condition HE lorsque la viscosité est non nulle et que les gains sont plus faibles que les gains nominaux. Par exemple, tant le temps de réalisation que l'amplitude des efforts appliqués et le taux d'erreur de manipulation ont été améliorés pour la condition la plus difficile ( $g10$  et  $v100$ ) dans chacun des coins. Néanmoins, les résultats diffèrent pour une condition donnée entre chacun des coins (voir Figure 5.9). La condition ( $g50, v100$ ) est ainsi significative pour le coin 2 mais pas pour le coin 1, alors que c'est le contraire pour la condition ( $g100, v100$ ). La différence pour la condition EH a été testée par un test- $t$ , les différences significatives sont indiquées sur les figures 5.6 et 5.7.

#### 5.4.2 Résultat pour les erreurs de manipulation

$m$	$v0$	$v25$	$v100$	$\sigma$	$v0$	$v25$	$v100$
$g10$				$g10$			
NHE	0.14	0.93	6.57	NHE	0.36	2.20	8.77
HE	0.5	0.36	3.0	HE	0.94	0.93	3.06
valeur $p$	0.16	0.58	0.11				
$g50$				$g50$			
NHE	4.14	3.57	15.93	NHE	4.50	4.27	16.71
HE	3.07	4.57	7.93	HE	3.04	5.59	6.69
valeur $p$	0.34	0.60	0.06				
$g100$				$g100$			
NHE	5.64	9.71	16.35	NHE	3.77	6.18	14.68
EH	8.28	7.35	11.42	HE	6.25	4.70	7.26
valeur $p$	0.05	0.09	0.06				

**Table 5.3** – Moyennes ( $m$ ) et écarts type ( $\sigma$ ) du nombre d'erreurs de manipulation pour chaque condition.

Le taux d'erreur de manipulation décroît avec des gains faibles et augmente avec une viscosité globale forte. Ce taux diminue également lorsque l'haptique événementielle est activée alors que les gains sont faibles et/ou la viscosité globale est importante. Des résultats inverses peuvent être observés pour des conditions plus proches de l'idéal ( $g10, v0$ ), ( $g50, v25$ ) et ( $g100, v0$ ).

Le nombre d'erreurs de manipulation ne suit pas une loi normale. Par exemple, il y a beaucoup de conditions avec quasiment aucune erreur, particulièrement celles avec la condition  $g10$  (49 conditions testées sans erreur pour 1 gain  $\times$  3 viscosités  $\times$  2 HE  $\times$  14 sujets = 84 cas). Il y a donc beaucoup de zéros et de liens entre chacune des populations. Les tests de Kruskal-Wallis menés sur le taux d'erreur montrent des différences marginalement significatives entre HE et NHE pour ( $v100, g50$ ) et pour toutes les conditions avec  $g100$ .

Les performances des utilisateurs sont loin d'être homogènes. Certains sujets ont réalisé

Facteur	dl	valeur de $F$	valeur de $p$
gain	2	264.09*	$< 10^{-6}$
visc	2	1768.68*	$< 10^{-6}$
HE	1	18.4*	$1.78 \cdot 10^{-6}$
gain $\times$ HE	2	2.98	0.05
visc $\times$ HE	2	1.49	0.22
gain $\times$ visc	4	3.36*	$4.21 \cdot 10^{-5}$
gain $\times$ visc $\times$ HE	4	5.10*	$4.20 \cdot 10^{-4}$

**Table 5.4** – Une ANOVA à trois facteurs a été réalisée sur le temps de réalisation de la tâche. Les différences significatives sont indiquées par un astérisque.

l'ensemble de la manipulation en ne commettant quasiment aucune erreur de manipulation (13 erreurs au mieux, 6 NEBH and 7 EBH), alors que d'autres sujets ont réalisé beaucoup d'erreur, 261 dans le pire des cas (159 NHE et 102 HE).

### 5.4.3 Résultats pour le temps de réalisation

Le temps écoulé pour réaliser le parcours de gauche à droite a été évalué le long de la section chronométrée (voir Figure 5.4). Les résultats sont présentés dans la Figure 5.8. Une ANOVA à trois facteurs a été réalisée sur le temps de parcours pour les conditions de gains, de viscosité et de HE. Les résultats sont présentés dans la table 5.4. Des tests- $t$  appariés ont été menés pour chaque conditions. Les différences significatives entre NHE et HE sont indiquées sur la Figure 5.8. Lorsque la viscosité croît, le temps de réalisation augmente. Au contraire, le temps décroît lorsque les gains deviennent faibles. Le temps de réalisation est globalement plus important avec l'haptique événementielle que sans, sauf pour la condition la plus difficile ( $g10, v100$ ).

Les différences significatives pour chacune des grandeurs mesurées sont résumées dans la Figure 5.9.

## 5.5 Discussion

Tout d'abord, nous notons que la viscosité a un effet important sur la manipulation, ce qui était attendu. Une grande viscosité augmente le temps de réalisation de la tâche et dégrade l'information fournis par le rendu haptique. Les nombres d'erreurs augmentent et le maximum d'effort appliqué augmente également (conditions  $v25$  et  $v100$ ).

Le taux d'erreurs de manipulation et le temps de réalisation de la tâche diminuent avec des gains faibles. Les sujets semblent plus confiant dans le fait de réaliser la tâche et arrivent à respecter les deux consignes que sont le respect de la limite de 5 N et la réalisation de la tâche le plus rapidement possible. Ce résultat contredit une de nos hypothèses secondaires, celle où le temps de réalisation doit augmenter avec les gains faibles. Ce résultat peut être dû à deux causes distinctes. Premièrement, avec des gains faibles, un sujet doit réaliser un mouvement plus large pour générer le même effort au contact qu'avec des gains plus élevés. Il a peut être ainsi plus de temps pour réagir au moment de l'impact et il peut éviter d'appliquer des efforts trop important. Comme les sujets réussissent mieux à respecter la limite d'efforts ils deviennent plus confiant et réalisent donc la tâche plus rapidement avec des gains faibles. La seconde explication peut être due à l'interface haptique utilisée. Les sujets ont manipulés les objets virtuels à l'aide d'une poignée (voir Figure 5.3), ils ont peut être été moins précis pour

contrôler leur mouvement et les efforts appliqués que si ils avaient pu manipuler l'interface avec leur doigts. Une précision moins grande sur le mouvement tire parti de gains plus faibles. En effet, un mouvement de plus grande amplitude est nécessaire avec des gains faibles pour générer une grande force. La limite de 5 N a été fixée de manière uniforme pour toutes les conditions de gain. Une condition d'effort adaptée pour chaque gain ou une condition sur la distance d'interpénétration aurait peut être pu fournir des résultats plus significatifs sur le nombre d'erreurs où sur le temps de réalisation de la tâche. Ce résultat concernant de meilleures performances pour des gains faibles semble similaire à celui de [PS09] où un retour haptique léger diminue le temps de réalisation d'une tâche de sélection.

Enfin, il peut être noté que les résultats en efforts le long des axes  $x$  et  $y$  sont sensiblement différents (voir figures 5.6 et 5.7) spécialement pour des gains faibles. Le bras humain est connu pour être un effecteur et un capteur anisotrope. C'est peut être pourquoi les sujets ont appliqué un effort globalement plus important selon l'axe  $y$  que selon l'axe  $x$  afin de détecter la paroi du parcours.

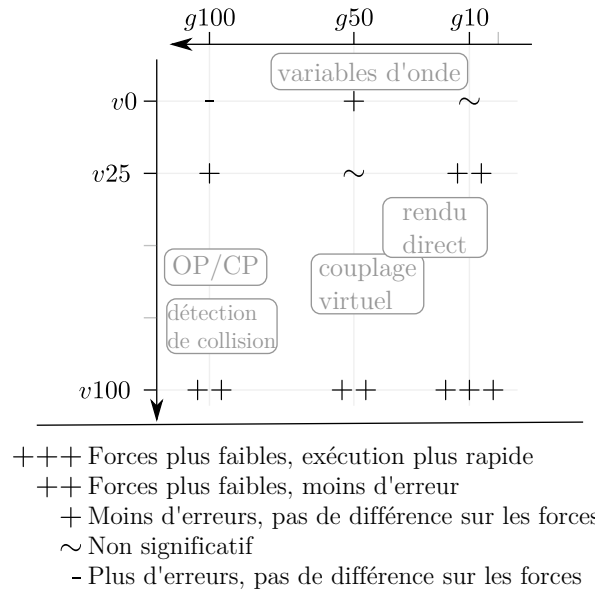
### 5.5.1 Effets du retour haptique événementiel

Les résultats semblent montrer que le retour haptique événementiel a un effet paradoxal sur la réalisation de la tâche. En effet, bien que les sujets aient tendance à appliquer des efforts plus faibles lorsque que l'haptique événementielle est activé, ils prennent plus de temps pour réaliser la tâche. Ce fait invalide notre troisième hypothèse au sujet de l'haptique événementielle, nous avons supposé que le temps de réalisation de la tâche diminue avec HE. Nous pensons que les sujets ont tendance à surestimer les efforts qu'ils appliquent au moment de l'impact lorsque l'haptique événementielle est activé. Cette surestimation a peut-être tendance à les rendre plus prudent et donc à réaliser la tâche plus lentement.

Pour les conditions extrêmes, la condition idéale ( $g100, v0$ ) et la condition la plus difficile ( $g10, v100$ ), les effets de l'haptique événementielle sont plus clairs. L'haptique événementielle semble améliorer les performances pour la condition la plus dégradée en diminuant le temps de réalisation de la tâche, en diminuant les efforts appliqués sur les axes  $x$  et  $y$  et en diminuant le nombre d'erreurs. A l'opposé, l'haptique événementielle a des effets négatifs sur la condition idéale. Bien qu'il n'y ait pas de différences significatives entre les efforts appliqués pour HE et NHE, le nombre d'erreurs de manipulation décroît lorsque l'haptique événementielle est désactivée. Les sujets ont peut-être sous-estimé les efforts qu'ils appliquaient et ont donc été amenés à réaliser plus d'erreurs.

La différence du nombre d'erreurs avec et sans retour haptique événementiel semble croître avec une viscosité importante. D'une manière similaire, les efforts sont plus faibles avec HE en présence de viscosité. Bien que la différence du nombre d'erreurs ne soit pas significative pour des gains faibles, les forces sont significativement plus faibles avec HE en présence de viscosité et de gains faibles (voir Figure 5.9). Les effets de l'haptique événementielle semblent donc plus notables pour des gains faibles et des viscosités importantes, ainsi que nous l'avions supposé. Notamment, le nombre d'erreurs de manipulation est deux fois moindre avec HE qu'avec NHE dans la condition la plus difficile ( $g10, v100$ ). Il est aussi intéressant de noter que les différences significatives sur le nombre d'erreurs et sur les efforts appliqués semblent complémentaires pour  $v100$  et  $v25$ . Cela doit être du à la limite d'effort de 5 N. Les forces sont bornées par cette limite. Ainsi, lorsque les efforts appliqués se rapprochent de cette limite, le nombre d'erreurs de manipulation augmente.

Notre hypothèse, selon laquelle le retour haptique événementiel associé avec un retour d'effort dégradé améliore la manipulation en permettant de mieux détecter les contacts, semble validée, particulièrement à partir du moment où il y a une viscosité. La compréhension



**Figure 5.10** – Résumé des effets du retour haptique événementiel pour les différentes conditions testées. Les méthodes présentées sur la Figure 5.2 sont indiquées sur le schéma.

de l'environnement virtuel semble améliorée en utilisant un retour haptique événementiel. Les effets de cette technique sont résumés sur la Figure 5.10 en fonction des conditions de gain et de viscosité testées.

Le débriefing à la fin de chaque séance de manipulation nous a permis de relever des éléments intéressants. Les utilisateurs expérimentés semblent préférer travailler avec le retour haptique événementiel activé, lorsqu'ils comparent les deux conditions NHE et HE. A part un sujet, les sujets n'ayant jamais utilisés d'interface haptique n'ont pas perçu de différence entre les conditions NHE et HE lorsqu'ils ont pu indiquer leurs impressions sur l'expérience au moment du débriefing. Ce sujet est celui qui a réalisé le minimum d'erreur de manipulation durant l'expérience. A la fin de l'expérience, seule une personne n'a pas pu distinguer les conditions NHE et HE même en lui expliquant la différence. Ce sujet est l'un des deux sujets pour lesquels l'haptique événementielle n'a eu aucune influence que ce soit sur le temps de réalisation de la tâche, le nombre d'erreurs ou les forces appliquées. Le retour haptique événementiel semble donc être grandement dépendent de la sensibilité de chaque sujet.

## Conclusion

Dans ce chapitre, nous avons présenté une évaluation simple de la contribution du retour haptique événementiel pour enrichir l'information de contact lorsque le retour d'effort traditionnel est dégradé. L'évaluation a été menée pour plusieurs gains du couplage virtuel et donc pour plusieurs raideurs transmises du contact ainsi que pour plusieurs viscosités globales. Les résultats semblent suggérer que l'ajout de retour haptique événementiel permet d'améliorer la perception du contact, bien que cette technique ait un effet paradoxal sur la manipulation. En effet, les sujets appliquent des efforts moins importants avec un retour d'effort dégradé. Cependant, dans une situation idéale, le retour haptique événementiel dégrade les performances. Enfin, l'efficacité de cette technique semble dépendre de chaque sujet.

### Perspectives

Cette étude a été menée pour une tâche à un seul opérateur. Une première perspective est de vérifier si le retour haptique événementiel permet d'améliorer les performances lors d'une tâche collaborative, et en particulier vis-à-vis des méthodes d'asservissement présentées dans les chapitres précédents. Nous pensons que les effets seraient plus notables pour l'asservissement avec variables d'ondes, car comme pour le cas d'un retour d'effort subissant une viscosité, des efforts parasites sont présents, en l'occurrence à cause d'une importante inertie. Or, l'haptique événementielle semble améliorer les performances lorsque le mouvement libre n'est pas transparent (conditions  $v25$  et  $v100$ ).

Un second point intéressant, serait d'étudier l'apport de retour haptique événementiel entre les différentes architectures distribuée et client-serveur. En effet, selon l'architecture retenue, le comportement de l'univers virtuel est différent. Le retour haptique événementiel doit aussi avoir une influence différente sur les performances de l'utilisateur en fonction de l'architecture proposée.

Enfin, l'ajout de retour haptique événementiel pourrait également modifier la manière dont une personne interagit avec un environnement virtuel et la représentation que la personne se fait de l'univers. Une utilisation plus longue que celle proposée lors de l'expérimentation pourrait également améliorer les résultats. Ces effets d'apprentissage sur le retour haptique événementiel peuvent donc être intéressants à développer.



# Conclusion et Perspectives

L'objet de nos travaux a été d'étudier la stabilité et les performances de retour d'effort pour la réalisation de tâche collaborative dans le cadre du prototypage virtuel en présence de retards constants.

Nous avons tout d'abord présenté une synthèse de l'état de l'art sur les environnements virtuels et sur le retour d'effort structurée en quatre axes. Nous avons ainsi décrit les simulations physiques rencontrées en réalité virtuelle et la manière de leur associer un retour d'effort stable. Dans le cas de retard, nous avons présenté les méthodes issues de la téléopération avec retour d'effort. Enfin, nous avons détaillé les différents choix d'architecture possibles ainsi que l'influence de ces choix et du retard sur la collaboration avec retour d'effort. Cette étude nous a permis de remarquer, que le problème de la stabilité et la qualité du retour d'effort n'était pas suffisamment exploré dans la littérature actuelle. En particulier, il n'existe pas d'études de l'influence des asservissements sur la collaboration et qu'il n'existe pas de plateformes collaboratives permettant une collaboration stable dans la littérature.

Afin d'étudier l'apport du retour d'effort dans un environnement virtuel collaboratif, nous avons proposé une architecture distribuée. En nous limitant aux cas des objets rigides, nous avons souligné l'importance du choix du point de couplage entre les instances d'un objet partagé. Tout en tolérant la divergence entre les états des différents nœuds de l'application distribuée, nous avons proposé une méthode basée sur les détecteurs de collision qui permet de révéler aux utilisateurs d'éventuelle incohérence lors de la manipulation. Ces différents points nous ont amené à mettre en place une plateforme collaborative à retour d'effort qui nous a permis de réaliser des tâches de prototypage virtuel avec un retour d'effort à 6 degrés de liberté. Cette plateforme a servi, par la suite, de base à l'évaluation des deux asservissements proposés.

Nous avons ensuite choisi d'utiliser des asservissements proportionnel dérivé pour asservir les instances de chaque objet partagé. Cette méthode est stable pour des retards faibles et constants. Cependant, l'utilisation de ces asservissements introduit des efforts de traînée gênants pour la manipulation. Nous avons alors suggéré d'ajouter une boucle de compensation. Cette boucle permet, en réduisant le domaine de stabilité des asservissements, de garantir un mouvement libre quasi transparent. Avec cette contribution, nous avons effectivement pu réaliser des tâches de manipulation avec un retard de 50 ms.

Dans le cas de retards constants de plus grande ampleur, nous avons développé une architecture basée sur la transformation en variables d'onde. Celle-ci nous a permis de garantir la stabilité de l'interaction quelque soit le retard. À l'aide d'une formulation articulaire et des différentes méthodes issues du domaine de la téléopération, nous avons pu effectivement réaliser des tâches collaboratives pour des retards de plus de 50 ms. Néanmoins, nous avons remarqué, pour des retards supérieurs à 200 ms, que cette méthode ne permet plus d'offrir un confort suffisant pour réaliser une tâche collaborative. En effet, les effets inertiels induits



deviennent dans ce cas particulièrement importants.

Dans le dernier chapitre, nous avons suggéré l'utilisation du retour haptique événementiel pour améliorer la qualité de l'interaction dans un environnement virtuel collaboratif dégradé. En effet, les asservissements précédemment proposés dégradent la qualité du retour d'effort. Afin de valider cette approche, nous avons réalisé une expérimentation en simulant certaines conditions pouvant être rencontrées dans de tels environnements. Les résultats semblent montrer que le retour haptique événementiel améliore les performances des sujets sur une tâche simple basée sur le retour haptique, particulièrement dans un environnement dégradé. Cette technique semble prometteuse car les sujets ont eu tendance à appliquer des efforts moins importants pour détecter un contact.

A partir de ces résultats, nous pouvons constater que l'utilisation de retour d'effort au sein d'une plateforme collaborative est toujours soumise à certains compromis, compromis également présents en téléopération. Quelques soient les méthodes d'asservissement retenues, la qualité du retour d'effort ne peut être vraiment idéal. De plus, le choix de l'architecture nous impose soit un retard entre l'action et la réaction dans le cas d'une architecture clients-serveur, soit la présence de divergence entre les nœuds et donc d'incohérence, en l'absence de mécanismes de contrôles. Pour ce dernier cas, il n'existe actuellement pas de méthode satisfaisante garantissant une manipulation stable sans interruption. Néanmoins, les résultats concernant le retour haptique événementiel semblent prometteurs.

## Perspectives

Plusieurs axes peuvent être dégagés à partir de ces conclusions.

Au sein des asservissements, en particulier pour le couplage virtuel proportionnel dérivé, le rôle des différents couplages virtuels (interface haptique - simulation, simulation - simulation) n'a pas été abordé. Un réglage fin des gains de ces différents paramètres pourrait permettre d'obtenir des domaines de stabilité plus importants. Un retour d'effort de meilleure qualité pourrait ainsi être fourni. Il faudrait également étudier l'influence de ces gains sur la qualité du suivi en position.

D'une manière un peu similaire, nous pensons, qu'un réglage donné des asservissements n'est pas forcément pertinent pour toutes les tâches collaboratives, en particulier en ce qui concerne l'utilisation des variables d'onde. Une tâche collaborative où les vitesses des objets virtuels sont faibles peut admettre une valeur d'amortissement  $b$  plus importante qu'une tâche nécessitant beaucoup de mouvement en environnement libre. En effet, avec une valeur d'amortissement  $b$  élevé, les effets inertiels sont importants. Des heuristiques pourraient être établies permettant d'aider le réglage de ces asservissements.

Actuellement, les deux asservissements présentés sont stables pour un retard constant. En l'état, il faudrait utiliser un réseau dédié pour réaliser une tâche collaborative à distance avec la plateforme que nous avons présentée. Des travaux ont proposé, notamment, des adaptations des variables d'onde pour le cas de retards variables ou de pertes de paquets. Un axe intéressant serait donc d'étudier l'influence de ces méthodes sur la qualité du retour haptique lors d'une tâche collaborative. Puis d'évaluer une telle plateforme sur Internet entre deux sites distants

Cette plateforme a permis d'étudier la qualité du retour d'effort pour le cas des objets rigides. L'ajout d'objets polyarticulés permettrait de traiter une gamme de scénarios issus du prototypage virtuel plus large. Dans le cas de retard faible, et donc de divergence faible, un asservissement articulaire entre les instances d'objets polyarticulés semble pertinent. Avec des retards élevés, des cas divergences et d'incohérences complexes pourraient se produire et ainsi limiter l'usage d'objets polyarticulés dans une plateforme collaborative avec retour d'effort.

L'étude de l'apport de l'haptique événementielle sur des environnements dégradés dans un cas simple nous a fourni des résultats prometteurs. Cette étude pourrait être prolongée par l'évaluation d'une tâche collaborative. Dans ce cas plusieurs effets sont à séparer. Par rapport à notre évaluation, le cas d'une évaluation collaborative introduit un retard entre les actions des sujets. Il faut alors distinguer les effets de l'environnement dégradé, des effets du retard sur les actions.

Enfin, et c'est un axe qui nous semble très prometteur, il s'agirait d'étudier l'apport de l'haptique événementielle sur une tâche collaborative en l'absence de retour haptique traditionnel. Cette approche serait plus simple à mettre en œuvre car elle permettrait de s'affranchir des problèmes de stabilité, et de pouvoir utiliser des méthodes de contrôle de cohérence potentiellement instables avec un retour d'effort classique [GJF<sup>+</sup>07]. De plus la contrainte d'un temps de calcul à 1 kHz pourrait être relâchée. Cette approche permet aussi de s'affranchir de toutes les forces parasites (forces de traîné, effet inertiel, etc.) introduites par les asservissements issus de la téléopération. Nous pensons que ce type de rendu associé à des aides visuelles pourrait effectivement permettre de réaliser des tâches collaboratives de manière au moins aussi efficace qu'avec un retour d'effort traditionnel.

L'utilisation de retour d'effort au sein d'environnements virtuels collaboratifs n'est encore que marginal. De nombreux compromis doivent être fait afin de pouvoir réaliser une tâche collaborative. Les perspectives proposées, en particulier au niveau du rendu haptique événementiel devrait permettre de pouvoir réaliser des interactions collaboratives bénéficiant d'un meilleur réalisme. Cette dernière méthode étant simple à mettre en œuvre, elle pourrait bénéficier à tout type d'environnement virtuel, même les environnements virtuels grand public.



# Annexe

## Sommaire

<b>A.1 Notations</b> . . . . .	<b>121</b>
A.1.1 Notation . . . . .	121
A.1.2 Cinématique des objets rigides . . . . .	121
<b>A.2 Expression des matrices d'immittances pour le couplage proportionnel dérivé</b>	<b>123</b>
A.2.1 Expression sans la boucle de compensation . . . . .	123
A.2.2 Ajout d'une boucle de compensation . . . . .	124

## A.1 Notations

Cette annexe décrit les notations utilisées dans le manuscrit.

### A.1.1 Notation

Les notations utilisées dans ce manuscrit sont issues de [Str01].

- $\Psi_i$  est un repère direct  $i$ ,  $\Psi_O$  est le repère de référence du système.
- $H_i^j$  est la transformation homogène du repère  $\Psi_i$  vers le repère  $\Psi_j$
- $T_i^{k,j}$  est le torseur cinématique attaché au repère  $\Psi_i$  par rapport au repère  $\Psi_j$ . Il est exprimé dans le repère  $\Psi_k$
- $W_{i,j}^k$  est le torseur d'effort appliqué sur le repère  $\Psi_j$  par le repère  $\Psi_i$  exprimé dans le repère  $\Psi_k$
- $W_i^k$  est l'ensemble des torseurs d'effort appliqués sur  $\Psi_i$  exprimé dans le repère  $\Psi_k$

### A.1.2 Cinématique des objets rigides

A chaque point  $i$  d'un objet rigide, un repère  $\Psi_i$  est associé. La position du point  $i$  d'un objet rigide peut être exprimée dans le repère  $\Psi_O$  par la transformation homogène  $H_i^O$ . Cette expression permet de décrire de manière unifiée une translation et une rotation. Cette matrice est un élément d'un groupe de Lie, elle est également une représentation sous forme de matrice d'un élément du groupe Euclidien Spécial  $SE(3)$  défini par :

$$SE(3) := \left\{ H = \begin{pmatrix} R & p \\ 0 & 1 \end{pmatrix} \in \mathbb{R}^{4 \times 4} \text{ avec } R, p \in SO(3) \times \mathbb{R}^3 \right\}$$

$$SO(3) := \left\{ R \in \mathbb{R}^{3 \times 3} \text{ avec } R^{-1} = R^T \text{ et } \det(R) = 1 \right\}$$

où  $R \in SO(3)$  est la représentation sous forme de matrice d'une rotation et  $p \in \mathbb{R}^3$  est la représentation vectorielle d'une translation. Ces deux éléments permettent donc bien de représenter une position d'un objet dans l'espace.

La trajectoire du repère  $i$  par rapport à  $j$  peut alors être définie par la matrice homogène dépendant du temps  $H_i^j(t) \in SE(3)$ .

Les vitesses en translation et en rotation d'un objet rigide peuvent être définies à partir d'un déplacement instantané relatif entre deux repères  $\Psi_i$  et  $\Psi_j$ ,  $\dot{H}_i^j(t)$ . Cette matrice peut être écrite à l'aide de la matrice  $\tilde{T}_i^{j,k} = H_j^k \dot{H}_j^i H_k^j$ . Cette dernière est un élément de l'algèbre de Lie  $\mathfrak{se}(3)$  correspondant au groupe de Lie  $SE(3)$  :

$$\mathfrak{se}(3) := \left\{ \tilde{T} = \begin{pmatrix} \tilde{\omega} & v \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{4 \times 4} \text{ avec } \tilde{\omega} \in \mathfrak{so}(3) \text{ et } v \in \mathbb{R}^3 \right\}$$

$$\mathfrak{so}(3) := \left\{ \tilde{\omega} \in \mathbb{R}^{3 \times 3} \text{ avec } \tilde{\omega}^T = -\tilde{\omega} \right\}$$

où  $\tilde{\omega} \in \mathfrak{so}(3)$  est la représentation sous forme de matrice d'une vitesse de rotation et  $v \in \mathbb{R}^3$  est la représentation vectorielle d'une vitesse de translation définissant ainsi une vitesse d'un objet dans l'espace. La matrice  $\tilde{T}$  est donc la représentation matricielle du torseur cinématique  $T$ .

La matrice  $\tilde{\omega}$  étant antisymétrique de dimension 3, elle peut également être écrite sous forme vectorielle. L'opérateur permet ainsi d'associer de manière bijective un élément de l'espace vectoriel  $\mathbb{R}^3$  à un élément de l'espace des matrices antisymétriques de dimension  $3 \times 3$  de la manière suivant :

$$\forall \omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \tilde{\omega} = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}$$

D'une manière similaire  $\tilde{T}$  et  $T$  sont lié par :

$$\forall T = \begin{pmatrix} \omega \\ v \end{pmatrix}, \tilde{T} = \begin{pmatrix} \tilde{\omega} & v \\ 0 & 0 \end{pmatrix}$$

Il est intéressant de remarquer que chaque élément de  $SO(3)$  peut à partir d'un élément de  $\mathfrak{so}(3)$  en utilisant une carte exponentielle :

$$\forall R \in SO(3), \exists \tilde{\omega} \in \mathfrak{so}(3), R = e^{\tilde{\omega}} \text{ avec } e^A = \sum_{n=0}^{\infty} \frac{A^n}{n!}$$

Il existe une relation similaire entre un élément de  $SE(3)$  et un élément de  $\mathfrak{se}(3)$  :

$$\forall H \in SE(3), \exists \tilde{T} \in \mathfrak{se}(3), H = e^{\tilde{T}}$$

Le changement de repère du torseur cinématique peut être réalisé avec l'adjoint  $\text{Ad}_{H_i^j}$  de la transformation homogène  $H_i^j$  :

$$\exists \text{Ad}_{H_i^j} = \begin{pmatrix} R_i^j & 0 \\ \tilde{p}_i^j R_i^j & R_i^j \end{pmatrix}, T_i^{k,j} = \text{Ad}_{H_k^l} T_i^{l,j}$$

Un effort s'exerçant sur un objet peut être représenté par un torseur d'effort  $W$ . Ce dernier est défini à partir de l'expression de la puissance  $P$  développé par cet effort :

$$P = \tilde{W}^T \tilde{T} \text{ avec } P \in \mathbb{R}$$

$\tilde{W}$  est alors un élément de l'algèbre de Lie  $\mathfrak{se}^*(3)$  dual de  $\mathfrak{se}(3)$ . Les forces  $f$  et les moments  $\tau$  sont définis par :

$$W = \begin{pmatrix} \tau \\ f \end{pmatrix}$$

$W$  est également la représentation d'un élément de l'espace dual  $\mathfrak{se}^*(3)$  de  $\mathfrak{se}(3)$ . L'adjoint de la matrice homogène permet également de changer le repère dans lequel est exprimé ce torseur d'effort :

$$W_{i,j}^k = \text{Ad}_{H_k}^T W_{i,j}^l$$

## A.2 Expression des matrices d'immittances pour le couplage proportionnel dérivé

Cette annexe détaille les expressions des matrices d'immittances rencontrées dans le chapitre 3 lors de l'utilisation d'asservissements proportionnel dérivé entre deux instances d'un objet partagé.

### A.2.1 Expression sans la boucle de compensation

Le cas de la manipulation collaborative d'un objet ou de deux objets est déjà détaillé dans la Section 3.2.3.1 au travers des Équations 3.9, 3.10, 3.11 et 3.12

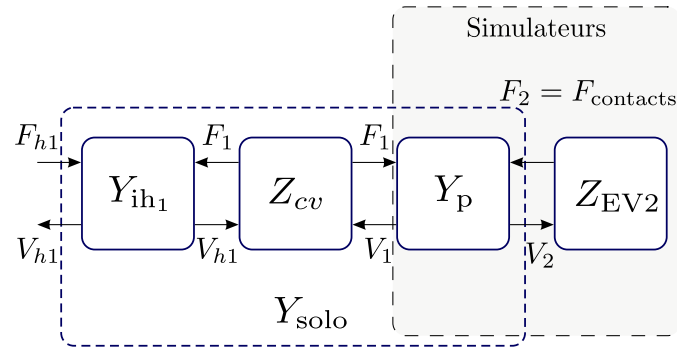
#### A.2.1.1 Cas mono-utilisateur

Le cas de la manipulation mono-utilisateur est résumé sur schéma A.1a et A.1b  
La matrice d'admittance  $Y_{\text{solo}}$  est la suivante :

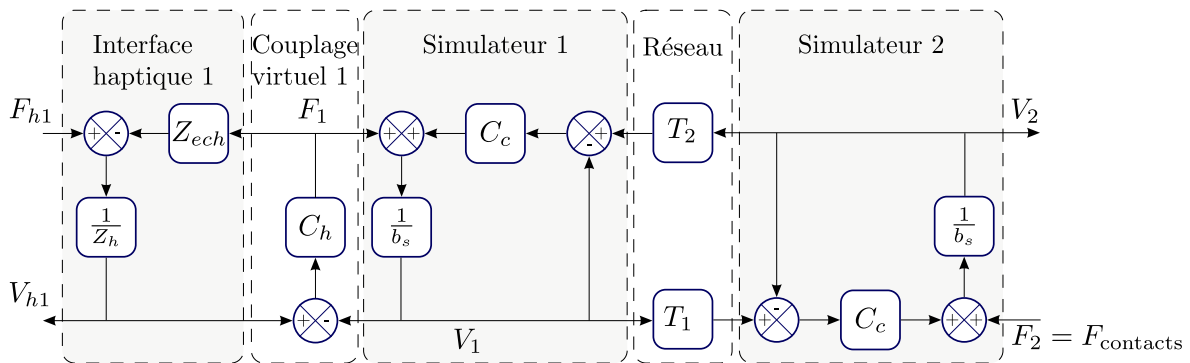
$$\begin{bmatrix} V_{h_1}(z) \\ -V_2(z) \end{bmatrix} = \begin{pmatrix} Y_{\text{solo}_{11}}(z) & Y_{\text{solo}_{12}}(z) \\ Y_{\text{solo}_{21}}(z) & Y_{\text{solo}_{22}}(z) \end{pmatrix} \begin{bmatrix} F_{h_1}(z) \\ -F_2(z) \end{bmatrix} \quad (\text{A.1})$$

où les éléments de la matrice valent :

$$\begin{aligned} Y_{\text{solo}_{11}}(z) &= \frac{(C_c + b_s)(b_s + C_h + C_c) - C_c^2 z^{-n_r}}{(C_c + b_s)(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c)) - C_c^2 (Z_h + C_h Z_{\text{ech}}) z^{-n_r}} \\ Y_{\text{solo}_{12}}(z) &= \frac{C_h Z_{\text{ech}} z^{-n_1}}{(C_c + b_s)(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c)) - C_c^2 (Z_h + C_h Z_{\text{ech}}) z^{-n_r}} \\ Y_{\text{solo}_{21}}(z) &= \frac{C_h C_c z^{-n_2}}{(C_c + b_s)(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c)) - C_c^2 (Z_h + C_h Z_{\text{ech}}) z^{-n_r}} \\ Y_{\text{solo}_{22}}(z) &= \frac{Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c)}{(C_c + b_s)(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c)) - C_c^2 (Z_h + C_h Z_{\text{ech}}) z^{-n_r}} \end{aligned} \quad (\text{A.2})$$



(a) Schéma bloc simplifié (Figure 3.6)



(b) Schéma bloc dans le cas mono-utilisateur

Figure A.1 – Schéma bloc détaillé

## A.2.2 Ajout d'une boucle de compensation

L'ajout de la boucle de compensation au niveau du couplage virtuel, fait apparaître un terme supplémentaire dans l'expression des matrices d'admittance  $Y_{coll}$  relative au cas collaboratif et  $Y_{solo}$  relative à la manipulation mono-utilisateur.

### A.2.2.1 Cas de manipulation collaborative

Le cas de la manipulation collaborative avec boucle de compensation est résumé sur les schémas A.1a et A.1b

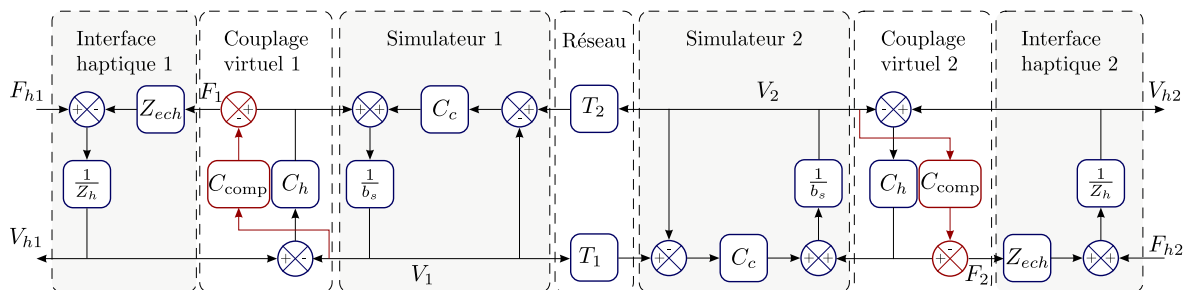


Figure A.2 – Schéma bloc détaillé du cas collaboratif avec boucle de compensation (Figure 3.13)

Les termes de la matrice d'admittance  $Y_{coll}$  valent alors :

$$Y_{\text{coll}11}(z) = \frac{(b_s + C_h + C_c)(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c - C_d)) - C_c^2(Z_h + C_h Z_{\text{ech}})z^{-n_r}}{(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c - C_d))^2 - C_c^2(Z_h + C_h Z_{\text{ech}})^2 z^{-n_r}}$$

$$Y_{\text{coll}12}(z) = \frac{C_c C_h (C_h + C_d) Z_{\text{ech}} z^{-n_1}}{(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c - C_d))^2 - C_c^2(Z_h + C_h Z_{\text{ech}})^2 z^{-n_r}}$$

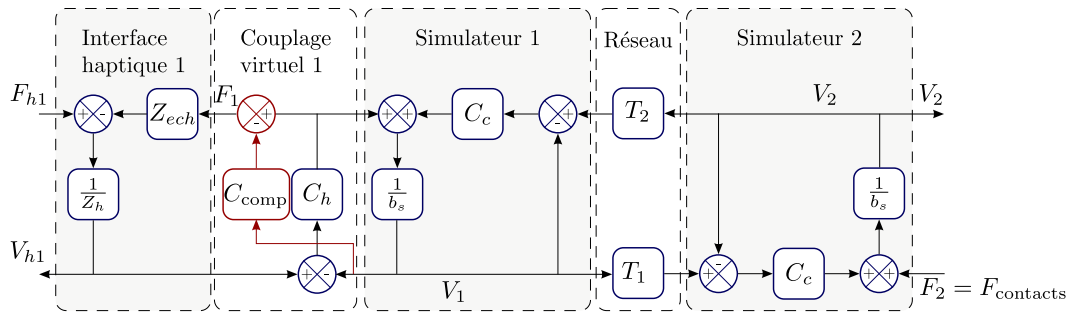
$$Y_{\text{coll}21}(z) = \frac{C_c C_h (C_h + C_d) Z_{\text{ech}} z^{-n_2}}{(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c - C_d))^2 - C_c^2(Z_h + C_h Z_{\text{ech}})^2 z^{-n_r}}$$

$$Y_{\text{coll}22}(z) = \frac{(b_s + C_h + C_c)(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c - C_d)) - C_c^2(Z_h + C_h Z_{\text{ech}})z^{-n_r}}{(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c - C_d))^2 - C_c^2(Z_h + C_h Z_{\text{ech}})^2 z^{-n_r}}$$

La symétrie remarquée dans la matrice  $Y_{\text{coll}}$  sans compensation (Section 3.2.3.1) est toujours présente avec l'ajout du terme de compensation  $C_d$ . La stabilité du système pour ce cas précis dépend également uniquement du retard total  $z^{-n_r}$ .

### A.2.2.2 Cas mono-utilisateur

De la même manière que pour la matrice  $Y_{\text{coll}}$ , le terme  $C_d$  apparaît dans les différents éléments de la matrice  $Y_{\text{solo}}$  lorsque que la boucle de compensation est ajoutée.



**Figure A.3** – Schéma bloc détaillé du cas collaboratif avec boucle de compensation (Figure 3.13)

$$Y_{\text{solo}11}(z) = \frac{(C_c + b_s)(b_s + C_h + C_c) - C_c^2 z^{-n_r}}{(C_c + b_s)(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c - C_d)) - C_c^2(Z_h + C_h Z_{\text{ech}})z^{-n_r}}$$

$$Y_{\text{solo}12}(z) = \frac{(C_h + C_d) Z_{\text{ech}} z^{-n_1}}{(C_c + b_s)(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c - C_d)) - C_c^2(Z_h + C_h Z_{\text{ech}})z^{-n_r}}$$

$$Y_{\text{solo}21}(z) = \frac{C_h C_c z^{-n_2}}{(C_c + b_s)(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c - C_d)) - C_c^2(Z_h + C_h Z_{\text{ech}})z^{-n_r}}$$

$$Y_{\text{solo}22}(z) = \frac{(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c - C_d))}{(C_c + b_s)(Z_h(b_s + C_h + C_c) + C_h Z_{\text{ech}}(b_s + C_c - C_d)) - C_c^2(Z_h + C_h Z_{\text{ech}})z^{-n_r}}$$

(A.3)





# Publications de l'auteur

- [P1] Mathieu Gautier, Claude Andriot and Pierre Ehanno. 6DOF Haptic Cooperative Virtual Prototyping over High Latency Networks. In *EuroHaptics '08 : Proceedings of the 6th international conference on Haptics*, pages 876–885, 2008
- [P2] Mathieu Gautier et Claude Andriot and Pierre Ehanno. Plateforme de coopération distribuée avec retour haptique à 6 degrés de liberté pour le prototypage virtuel. *Actes des journées de l'Association Française de Réalité Virtuelle*, 2008
- [P3] Mathieu Gautier and Claude Andriot. 6DOF haptic cooperation over large latency network with wave variables for virtual prototyping. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1208–1214, 2009
- [P4] Mathieu Gautier, Jean Sreng and Claude Andriot. Influence of Event-based Haptic on the Manipulation of Rigid Objects in Degraded Virtual Environment. In *Proceedings of ACM Symposium on Virtual Reality Software and Technology*, 2009 *papier soumis*



# Bibliographie

- [AH98] Richard J. Adams and Blake Hannaford. A two-port framework for the design of unconditionally stable haptic interfaces. In *IEEE/RSJ International Conference on Intelligent Robots and Systems 1998*, pages 1254–1259, 1998. 51, 74, 76
- [AM01] Paolo Arcara and Claudio Melchiorri. A comparison of control schemes for teleoperation with time delay. In *1st IFAC Conference on Telematics Applications in Automation and Robotics*, pages 505–510, 2001. 27
- [AMH99] Richard J. Adams, Manuel R. Moreyra, and Blake Hannaford. Stability and performance of haptic displays : Theory and experiments. In *Proceedings ASME International Mechanical Engineering Congress and Exhibition*, pages 227–234, 1999. 23, 74
- [AO05] Jake J. Abbott and Allison M. Okamura. Effects of position quantization and sampling rate on virtual-wall passivity. *IEEE Transactions on Robotics*, 21(5) :952–964, 2005. 20
- [APS99] Mihai Anitescu, Florian A. Potra, and David E. Stewart. Time-stepping for three-dimensional rigid body dynamics. In *Computer Methods in Applied Mechanics and Engineering*, pages 183–197, 1999. 13
- [AS89] Robert J. Anderson and Mark W. Spong. Bilateral control of teleoperators with time delay. *IEEE Transactions on Automatic Control*, 34(5) :494–501, 1989. 30
- [AVPH06] Jordi Artigas, Jordi Vilanova, Carsten Preusche, and Gerd Hirzinger. Time domain passivity control-based telepresence with time delay. In *IEEE/RSJ International Conference on Intelligent Robots and Systems 2006*, pages 4205–4210, 2006. 28
- [Bar93] David Baraff. Non-penetrating rigid body simulation. In *SIGGRAPH 95 Course Note 34. ACM SIGGRAPH*, 1993. 13
- [BBPB02] Mourad Bouzit, Grigore Burdea, George Popescu, and Rares Boian. The Rutgers master ii-new design force-feedback glove. *IEEE/ASME Transactions on Mechatronics*, 7(2) :256–263, 2002. 16
- [BHSS00] Catagay Basdogan, Chih-Hao Ho, Mandayam A. Srinivasan, and Mel Slater. An experimental study on the role of touch in shared virtual environments. *ACM Transactions on Computer-Human Interaction*, 7(4) :443–460, 2000. 42, 43

- [BJ08] Jernej Barbič and Doug L. James. Six-DoF haptic rendering of contact between geometrically complex reduced deformable models. *IEEE Transactions on Haptics*, 1(1), 2008. 9
- [BKHAK04] Mohamed Benali-Khoudja, Moustapha Hafez, Jean-Marc Alexandre, and Abderrahmane Kheddar. Tactile interfaces : a state-of-the-art survey. In *International Symposium on Robotics*, pages 721–726, 2004. 17
- [BM02] Francis Bogsanyi and Maike Luiken Miller. Tool and object based synchronization in collaborative haptics. In *The IEEE International Workshop on Haptic, Audio and Visual Environments*, pages 109–113, 2002. 40
- [BMM07] Gianni Borghesan, Alessandro Macchelli, and Claudio Melchiorri. Simulation issues in haptics. In *International Conference on Robotics and Automation 2007*, 2007. 21
- [BOH97] Pietro Buttolo, Roberto Oboe, and Blake Hannaford. Architectures for shared haptic virtual environments. *Special Issue of Computer and Graphics*, 1997. 38, 42, 52
- [Bro95] Wolfgang Broll. Interacting in distributed collaborative virtual environments. In *Proceedings of the IEEE Virtual Reality Annual International Symposium*, pages 148–155, 1995. 36
- [BSH05] Azzedine Boukerche, S. Shirmohammadi, and A. Hossain. A prediction algorithm for haptic collaboration. In *IEEE International Workshop on Haptic, Audio and Visual Environments and Their Applications*, pages 154–158, 2005. 40
- [Can86] John Canny. Collision detection for moving polyhedra. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2) :200–209, 1986. 10
- [CB94] J. Edward Colgate and J. Michael Brown. Factors affecting the z-width of a haptic display. In *IEEE Conference on Robotics and Automation*, pages 3205–3210, 1994. 16
- [CGSS93] J. Edward Colgate, Paul E. Grafing, Michael C. Stanley, and Gerd Schenkel. Implementation of stiff virtual walls in force-reflecting interfaces. In *IEEE Virtual Reality Annual International Symposium*, pages 202–208, 1993. 18, 19, 102
- [CH93] Christer Carlsson and Olaf Hagsand. Dive - a multi-user virtual reality system. In *IEEE Virtual Reality Annual International Symposium*, pages 394–400, 1993. 37
- [CLMP95] Jonathan D. Cohen, Ming C. Lin, Dinesh Manocha, and Madhav K. Ponamgi. I-COLLIDE : An interactive and exact collision detection system for large-scale environments. In *Proceedings of ACM SIGGRAPH'95*, pages 189–196, 1995. 7
- [CNAS05] Joono Cheong, Silviu-Iulian Niculescu, Anuradha Annaswamy, and Man-dayam A. Srinivasan. Motion synchronization in virtual environments with shared haptics and large time delays. In *WHC '05 : Proceedings of the First*

*Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 277–282, 2005. 27, 46

- [CPN09] Joono Cheong, Jongwoo Park, and Silviu-Iulian Niculescu. Physically based collaborative simulations under ring-like network configurations. *IEEE Transactions on Computer Graphics and Applications*, 29(4) :64–80, 2009. 41
- [CS94] J. Edward Colgate and Gerd Schenkel. Passivity of a class of sampled-data systems : application to haptic interfaces. In *Proceedings of the American Control Conference*, pages 3236–3240, 1994. 19, 28
- [CSB95] J. Edward Colgate, Michael C. Stanley, and J. Michael Brown. Issues in the haptic display of tool use. In *IEEE/RSJ International Conference on Intelligent Robots and Systems 1995*, pages 140–145, 1995. 22, 27, 51, 76, 102
- [CSHB03] Nikhil Chopra, Mark W. Spong, Sandra Hirche, and Martin Buss. Bilateral teleoperation over the internet : the time varying delay problem. In *Proceedings of the American Control Conference*, pages 155–160, 2003. 35
- [CSOB06] Nikhil Chopra, Mark W. Spong, Romeo Ortega, and Nikita Barbanov. On tracking performance in bilateral teleoperation. *IEEE Transactions on Robotics*, 22(4) :861–866, 2006. 34, 35, 87
- [DGC05] Jérémie Dequidt, Laurent Grisoni, and Christophe Chaillou. Collaborative interactive physical simulation. In *Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 147–150, 2005. 41
- [DHM<sup>+</sup>06] Iñaki Díaz, Josune Hernantes, Ignacio Mansa, Alberto Lozano, Diego Borro, Jorge Juan Gil, and Emilio Sánchez. Influence of multisensory feedback on haptic accessibility tasks. *Virtual Reality*, 10 :31–40, 2006. 43
- [DNBS06] Nicola Diolaiti, Günter Niemeyer, Federico Barbagli, and J. Kenneth Salisbury. Stability of haptic rendering : Discretization, quantization, time-delay and coulomb effects. *IEEE Transactions on Robotics*, 22(2) :256–268, 2006. 20
- [DR84] Philip J. Davis and Philip Rabinowitz. *Methods of Numerical Integration*. Academic Press, 1984. 14
- [DV75] Charles A. Desoer and M. Vidyasagar. *Feedback Systems : Input-Output Properties*. Academic Press, 1975. 17, 18
- [Edw00] Gregory W. Edwards. Performance and usability of force feedback and auditory substitutions in a virtual environment manipulation task. Master’s thesis, Faculty of the Virginia Polytechnic Institute, 2000. 43
- [EZ05] Chadi El Zammar. *Interactions coopératives 3D distantes en environnements virtuels : gestion des problèmes réseau*. PhD thesis, Institut National des Sciences Appliquées de Rennes, 2005. 56
- [FB98] E.D. Fasse and R.C. Breedveld. Modeling of elastically coupled bodies : Part i - general theory and geometric potential function method. *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, 120 :496–500, 1998. 51

- [Fit54] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6) :381–391, June 1954. 42
- [FSC07] Mahyar Fotoohi, Shahin Sirouspour, and David Capson. Stability and performance analysis of centralized and distributed multi-rate control architectures for multi-user haptic interaction. *The International Journal of Robotics Research*, 26 :977–994, 2007. 39, 46, 78
- [GB95] Chris Greenhalgh and Steve Benford. Massive : a distributed virtual reality system incorporating spatial trading. In *15th IEEE International Conference on Distributed Computing Systems (ICDCS'95)*, pages 27–34, 1995. 37
- [GBF<sup>+</sup>04] Carl Gutwin, Steve Benford, Mike Fraser, Ivan Vaghi, and Chris Greenhalgh. Revealing delay in collaborative environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 503–510, 2004. 45
- [GC96] R. Brent Gillespie and Mark R. Cutkosky. Stable user-specific haptic rendering of the virtual wall. In *Proceedings ASME International Mechanical Engineering Conference and Exposition*, pages 397–406, 1996. 20
- [GH07] Andrew H. C. Gosline and Vincent Hayward. Time-domain passivity control of haptic interfaces with tunable damping hardware. In *IEEE World Haptics Conference*, pages 164–169, 2007. 22, 29
- [GHS<sup>+</sup>05] Chris Gunn, Matthew Hutchins, Duncan Stevenson, Matt Adcock, and Patricia Youngblood. Using collaborative haptics in remote surgical training. In *the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 481–482, 2005. 38
- [GJF<sup>+</sup>07] Mashhuda Glencross, Caroline Jay, Jeff Feasel, Luv Kohli, Mary C. Whitton, and Roger Hubbard. Effective cooperative haptic interaction over the internet. In *IEEE Virtual Reality*, pages 115–122, 2007. 37, 41, 46, 52, 53, 119
- [GLM96] Stefan Gottschalk, Ming C. Lin, and Dinesh Manocha. OBB-Tree : A hierarchical structure for rapid interference detection. In *Proceedings of ACM SIGGRAPH'96*, 1996. 7
- [GME<sup>+</sup>00] Arthur Gregory, Ajith Mascarenhas, Stephen Ehmann, Ming C. Lin, and Dinesh Manocha. Six degree-of-freedom haptic display of polygonal models. In *in Proceedings IEEE Visualization*, pages 139–146, 2000. 20
- [GSH<sup>+</sup>07] Jorge Juan Gil, Emilio Sánchez, Thomas Hulin, Carsten Preusche, and Gerd Hirzinger. Stability boundary for haptic rendering :influence of damping and delay. In *International Conference on Robotics and Automation 2007*, pages 124–129, 2007. 20, 102
- [Gut01] Carl Gutwin. The effects of network delays on group work in real-time groupware. In *Proceedings of the Seventh European Conference on computer-Supported Cooperative World*, pages 299–318, 2001. 43, 44, 45, 56
- [GWZ97] Rakesh Gupta, Daniel Whitney, and David Zeltzer. Prototyping and design for assembly analysis using multimodal virtual environments. *Computer-Aided Design*, 29(8) :585–597, 1997. 43

- [HACH<sup>+</sup>04] Vincent Hayward, Oliver R. Astley, Manuel Cruz-Hernandez, Danny Grant, and Gabriel Robles-De-La-Torre. Haptic interfaces and devices. *Sensor Review*, 24(1) :16–29, 2004. 17
- [HB04] Sandra Hirche and Martin Buss. Packet loss effects in passive telepresence systems. In *Proceedings of the IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, december 2004. 36
- [HHSB05] Sandra Hirche, P. Hinterseer, E. Steinbach, and M. Buss. Network traffic reduction in haptic telepresence systems by deadband control. In *Proceedings of the World Congress of the International Federation of Automatic Control*, Prague, Czech, 2005. 36
- [HMF<sup>+</sup>01] Kenji Hikichi, Hironao Morino, I. Fukuda, S. Matsumoto, and Yasuhiko Yasuda. Architecture of haptics communication system for adaptation to network environments. In *IEEE International Conference on Multimedia and Expo*, pages 744–747, 2001. 40, 46
- [HMS<sup>+</sup>00] João P. Hespanha, Margaret McLaughlin, Gaurav S. Sukhatme, Mino Akbarian, Rajiv Garg, and Weirong Zhu. Haptic collaboration over the internet. The Fifth PHANTOM Users Group Workshop, 2000. 40
- [HMY02] Kenji Hikichi, Hironao Morino, and Yasuhiko Yasuda. The evaluation of adaptation control for haptics collaboration over the internet. In *IEEE Communications Quality and Reliability International Workshop.*, 2002. 45
- [HR02] Blake Hannaford and Jee-Hwan Ryu. Time domain passivity control of haptic interfaces. *IEEE Transactions on Robotics and Automation*, 18(1) :1–10, 2002. 21, 75, 102
- [HS05] Peter F. Hokayem and Mark W. Spong. Bilateral teleoperation : An historical survey. *Automatica*, 42 :2035–2057, 2005. 27
- [Hub95] Philip M. Hubbard. Collision detection for interactive graphics application. *IEEE Transactions on Visualization and Computer Graphics*, 1(3) :218–230, 1995. 7
- [Hub02] Roger Hubbard. Collaborative stretcher carrying : a case study. In *Eighth Eurographics Workshop on Virtual Environments*, pages 7–12, 2002. 42, 52
- [HWN04] Jesse D. Hwang, Michael D. Williams, and Günter Niemeyer. Toward event-based haptics : Rendering contact using open-loop force pulses. In *Proceedings of the 12th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 24–31, 2004. 25
- [HZZ01] Keyvan Hashtrudi-Zaad and Septimiu E. Salcudean. Analysis of control architectures for teleoperation systems with impedance/admittance master and slave manipulators. *The International Journal of Robotics Research*, 20 :419–445, 2001. 27, 69
- [ICG<sup>+</sup>06] Rosa Iglesias, Sara Casado, Teresa Gutiérrez, García-Alonso, Kian Meng Yap, Wai Yu, and Alan Marshall. A peer-to-peer architecture for collaborative haptic assembly. In *Proceedings of the 10th IEEE International Symposium on Distributed Simulation and Real-Time Applications*, 2006. 46



- [IS94] Masahiro Ishii and Makoto Sato. A 3d spatial interface device using tensed strings. *Presence*, 3(1), 1994. 17
- [Iwa94] Hiroo Iwata. Desktop force display. In *Proceedings of ACM SIGGRAPH'94*, page 215, 1994. 16
- [JC01] David E. Johnson and Elaine Cohen. Spatialized normal cone hierarchies. In *Proceedings of ACM Symposium on Interactive 3D Graphics*, pages 129–134, 2001. 9
- [JGH07] Caroline Jay, Mashhuda Glencross, and Roger Hubbard. Modeling the effects of delayed haptic and visual feedback in a collaborative virtual environment. *ACM Transactions on Computer-Human Interaction*, 14(2) :1073–0516, 2007. 44
- [JH06] Caroline Jay and Roger Hubbard. Quantifying the effects of latency on sensory feedback in distributed virtual environments. In *Proceedings of Virtual Images Seminar*, pages 9–16, 2006. 44
- [JMO<sup>+</sup>02] Joel Jordan, Jesper Mortensen, Manuel Oliveira, Mel Slater, Boon K. Tay, Jung Kim, and Mandayam A. Srinivasan. Collaboration in a mediated haptic environment. In *PRESENCE 2002, The 5th Annual International Workshop on Presence*, 2002. 43
- [JW03] David E. Johnson and Peter Willemsen. Six degree-of-freedom haptic rendering of complex polygonal models. In *11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2003. 20, 102
- [JWC05] David E. Johnson, Peter Willemsen, and Elaine Cohen. A haptic system for virtual prototyping of polygonal models. In *Proceedings of ACM SIGGRAPH'05*, 2005. 9
- [KFN06] Katherine J. Kuchenbecker, Jonathan Fiene, and Günter Niemeyer. Improving contact realism through event-based haptic feedback. *IEEE Transactions on Visualization and Computer Graphics*, 12(2) :219–230, 2006. 25, 102
- [KKT<sup>+</sup>04] Jung Kim, Hyun Kim, Boon K. Tay, Manivannan Muniyandi, Mandayam A. Srinivasan, Joel Jordan, Jesper Mortensen, Manuel Oliveira, and Mel Slater. Transatlantic touch : A study of haptic collaboration over long distance. *Presence*, 13(3) :328–337, 2004. 26, 39, 46
- [Kur65] K Kurokawa. Power waves and the scattering matrix. *IEEE Transactions on Microwave Theory and Techniques*, 13(2) :194–202, 1965. 29
- [Lam78] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communication of the ACM*, 21(7), July 1978. 40
- [Law92] D.A. Lawrence. Stability and transparency in bilateral teleoperation. In *Decision and Control, 1992., Proceedings of the 31st IEEE Conference on*, volume 3, pages 2649–2655, 1992. 27
- [LL88] Jack M. Loomis and Susan J. Lederman. Tactual perception. In Kenneth R. Boff and Janet E. Lincoln, editors, *Engineering Data Compendium : Human Perception and Performance*, chapter 31. Harry G. Armstrong Aerospace Medical Research Laboratory, 1988. 15

- [Lle52] F. B. Llewellyn. Some fundamental properties of transmission systems. *Proceedings of the Institute of Radio Engineers*, 40(3) :271–283, 1952. 23
- [LM08] Ming C. Lin and Dinesh Manocha. Overview on collision and proximity queries. In Ming C. Lin and Miguel A. Otaduy, editors, *Haptic Rendering : Foundations, Algorithms, and Applications*, chapter 9, pages 181–202. A K Peters, 1st edition, 2008. 6, 8
- [LMB<sup>+</sup>02] Anatole Lécuyer, Christine Mégard, Jean-Marie Burkhardt, Taegi Lim, Sabine Coquillart, Philippe Coiffet, and Ludovic Graux. The effect of haptic, visual and auditory feedback on an insertion task on a 2-screen workbench. In *Proceedings of Immersive Projection Technology Symposium*, 2002. 43
- [LPD<sup>+</sup>00] D.A. Lawrence, L.Y. Pao, A.M. Dougherty, M.A. Salada, and Y. Pavlou. Rate-hardness : a new performance metric for haptic interfaces. *IEEE Transactions on Robotics and Automation*, 16 :357–371, 2000. 102
- [LS06] Dongjun Lee and Mark W. Spong. Passive bilateral teleoperation with constant time-delays. In *International Conference on Robotics and Automation 2006*, pages 2902–2907, 2006. 27, 88, 102
- [MABA02] Tangi Meyer, Guillermo Andrade-Barroso, and Burno Arnaldi. Une architecture pour le retour d’efforts. In *Actes des 15 émes journée de l’AFIG*, 2002. 36
- [MB02] Saghir Munir and Wayne J. Book. Internet-based teleoperation using wavevariables with prediction. *IEEE/ASME transactions on mechatronics*, 7(2) :124–133, 2002. 31, 88, 90, 91
- [MBB<sup>+</sup>02] Kevin Montgomery, Cynthia Bruyns, Joel Brown, Stephen Sorkin, Frederic Mazzella, Guillaume Thonier, Arnaud Tellier, Benjamin Lerman, and Anil Menon. Spring : A general framework for collaborative, real-time surgical simulation. In *Medicine Meets Virtual Reality*, pages 296–303, 2002. 38
- [MC95] Brian Mirtich and John Canny. Impulse-based simulation of rigid bodies. In *Proceedings of ACM Symposium on Interactive 3D Graphics*, 1995. 13
- [MCF00] Brian E. Miller, J. Edward Colgate, and Randy A. Freeman. Guaranteed stability of haptic systems with nonlinear virtual environments. *IEEE Transactions on Robotics and Automation*, 16(6) :712–719, 2000. 20, 23
- [MCP05] Joshua S. Mehling, J. Edward Colgate, and Michael A. Peshkin. Increasing the impedance range of a haptic display by adding electrical damping. In *the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 257–262, 2005. 25
- [Mer07] Xavier Merlhiot. A robust, efficient and time-stepping compatible collision detection method for non-smooth contact between rigid bodies of arbitrary shape. In *Multibody Dynamics*, 2007. 9
- [MGPH06] James Marsh, Mashhuda Glencross, Steve Pettifer, and Roger Hubbard. A network architecture supporting consistent rich behavior in collaborative interactive applications. *IEEE Transactions on Visualization and Computer Graphics*, 12(3) :405–416, 2006. 42, 52, 53, 54

- [Mic63] Albert Michotte. *La perception de la causalité*. Publications universitaires de Louvain, deuxième édition, 1963. 43, 53
- [MLS94] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., Boca Raton, FL, USA, 1994. 50
- [MP02] Tissaphern Mirfakhrai and Shahram Payandeh. A delay prediction approach for teleoperation over the internet. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2002. 35
- [MPT99] William A. McNeely, Kevin D. Puterbaugh, and James J. Troy. Six degree-of-freedom haptic rendering using voxel sampling. In *Proceedings of ACM SIGGRAPH'99*, pages 401–408, 1999. 8, 12, 24, 51
- [MPT06] William A. McNeely, Kevin D. Puterbaugh, and James J. Troy. Voxel-based 6-dof haptic rendering improvements. *Haptics-e*, 3(7), 2006. 8, 102, 104
- [MSB<sup>+</sup>04] Dan Morris, Christopher Sewell, Nikolas Blevins, Federico Barbagli, and John Kenneth Salisbury. Collaborative virtual environment for the simulation of temporal bone surgery. In *MICCAI*, pages 319–327, 2004. 42
- [MVHE04] Martin Mauve, Jürgen Vogel, Volker Hilt, and Wolfgang Effelsberg. Local-lag and time- warp : Providing consistency for replicated continuous applications. *IEEE Transactions on Multimedia*, 6(1) :47–57, 2004. 40, 45
- [MW88] Matthew Moore and Jane Wilhelms. Collision detection and response for computer animation. *SIGGRAPH Computer Graphics*, 22(4) :289–298, 1988. 10, 12, 63
- [MW93] Scott MacKenzie and Colin Ware. Lag as a determinant of human performance in interactive systems. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 488–493, 1993. 43
- [MZP<sup>+</sup>95] M.R. Macedonia, M.J. Zyda, D.R. Pratt, D.P. Brutzman, and P.T. Barham. Exploiting reality with multicast groups : a network architecture for large-scale virtual environments. *IEEE Computer Graphics and Applications*, 15(5) :38–45, 1995. 37
- [NAT90] B. Naylor, J. Amanatides, and W. Thibault. Merging bsp trees yield polyhedral modeling results. In *Proceedings SIGGRAPH'90*, pages 115–125, 1990. 7
- [NS91] Günter Niemeyer and Jean-Jacques E. Slotine. Stable adaptive teleoperation. *IEEE Journal of Oceanic Engineering*, 16(1) :152–162, 1991. 30, 102
- [NS97a] Günter Niemeyer and Jean-Jacques E. Slotine. Designing force reflecting teleoperators with large time delays to appear as virtual tools. In *International Conference on Robotics and Automation 1997*, pages 2212–2218, 1997. 33, 34, 87, 92
- [NS97b] Günter Niemeyer and Jean-Jacques E. Slotine. Using wave variables for system analysis and robot control. In *International Conference on Robotics and Automation 1997*, pages 1619–1625, 1997. 32, 34, 87

- [NS98] Günter Niemeyer and Jean-Jacques E. Slotine. Towards force reflecting teleoperation over the internet. In *International Conference on Robotics and Automation 1998*, pages 1909–1915, 1998. 34, 91
- [OCT04] Pontus A. Ollson, Craig R. Carignan, and J. Tang. Cooperative control of virtual objects using haptic teleoperation over the internet. In *5th International Conference Series On Disability, Virtual Reality And Associated Technologies*, pages 149–156, 2004. 38
- [ODH98] Allison M. Okamura, Jack T. Dennerlein, and Robert D. Howe. Vibration feedback models for virtual environments. In *Proceedings of International Conference on Robotics and Automation*, pages 674–679, 1998. 25
- [OG02] Marcia Kilchenman O’Malley and Michael Goldfarb. The implications of surface stiffness for size identification and perceived surface hardness in haptic interfaces. In *IEEE International Conference on Robotics and Automation*, pages 1255–1260, 2002. 102
- [OHDC00] Allison M. Okamura, Matthew W. Hage, Jack T. Dennerlein, and Mark R. Cutkosky. Improving reality-based models for vibration feedback. In *Proceedings of ASME Dynamic Systems and Control Division*, pages 1117–1124, 2000. 25, 102
- [OHDC01] Allison M. Okamura, Matthew W. Hage, Jack T. Dennerlein, and Mark R. Cutkosky. Reality-based models for vibration feedback in virtual environments. *IEEE ASME Transactions on Mechatronics*, 6(2), 2001. 103, 108
- [OL03] Miguel A. Otaduy and Ming C. Lin. Clods : Dual hierarchies for multiresolution collision detection. In *Proceedings of Eurographics Symposium on Geometry Processing 2003*, pages 94–101, 2003. 15
- [OL06] Miguel A. Otaduy and Ming C. Lin. A modular haptic rendering algorithm for stable and transparent 6-dof manipulation. *IEEE Transactions on Robotics*, 22(4) :751–762, 2006. 14
- [ORC06] Michaël Ortega, Stéphane Redon, and Sabine Coquillart. A six degree-of-freedom god-object method for haptic display of rigid bodies. In *Proceedings of IEEE International Conference on Virtual Reality*, 2006. 24
- [Pen60] P. Jr. Penfield. Noise in negative resistance amplifiers. *IRE Transactions on Circuit Theory*, 7(2) :166–170, 1960. 29
- [Pop73] V. M. Popov. *Hyperstability of Control Systems*. Springer verlag, 1973. 17
- [PS09] Vijay M. Pawar and Anthony Steed. Evaluating the influence of haptic force-feedback on 3d selection tasks using natural egocentric gestures. In *Proceedings of the 2009 IEEE Virtual Reality Conference*, pages 11–18, Washington, DC, USA, 2009. IEEE Computer Society. 103, 113
- [RA93] L.B. Rosenberg and B.D. Adelstein. Perceptual decomposition of virtual haptic surfaces. In *IEEE Symposium on Research Frontiers in Virtual Reality*, pages 46–53, Oct 1993. 102

- [RKC00] Stéphane Redon, Abderrahmane Kheddar, and Sabine Coquillart. An algebraic solution to the problem of collision detection for rigid polyhedral objects. In *Proceedings of International Conference on Robotics and Automation*, pages 3733–3738, 2000. 10
- [RKC02] Stéphane Redon, Abderrahmane Kheddar, and Sabine Coquillart. Gauss’ least constraints principle and rigid body simulations. In *International Conference on Robotics and Automation 2002*, pages 517–522, 2002. 13
- [RKH04a] Jee-Hwan Ryu, Yoon Sang Kim, and Blake Hannaford. Sampled- and continuous-time passivity and stability of virtual environments. *IEEE Transactions on Robotics*, 20(4) :772–776, 2004. 21
- [RKH04b] Jee-Hwan Ryu, Dong-Soo Kwon, and Blake Hannaford. Stable teleoperation with time domain passivity control. *IEEE Transactions on Robotics and Automation*, 20(2) :365–373, 2004. 28
- [RKK97] Diego Ruspini, Krasimir Kolarov, and Oussama Khatib. The haptic display of complex graphical environments. In *Proceedings of SIGGRAPH’97*, pages 345–352, 1997. 24
- [RKLM04] Stéphane Redon, Young J. Kim, Ming C. Lin, and Dinesh Manocha. fast continuous collision detection for articulated body. In *Proceedings of ACM Symposium on Solid Modeling and Applications*, pages 117–283, 2004. 10, 12
- [RPHH05] Jee-Hwan Ryu, Carsten Preusche, Blake Hannaford, and Gerd Hirzinger. Time domain passivity control with reference energy following. *IEEE Transactions on Control Systems Technology*, 13(5) :737–742, 2005. 22
- [SAJvdSM02] Stefano Stramigioli, Bernhard Maschke Arjan J. van der Schaft, and Claudio Melchiorri. Geometric scattering in robotic telemanipulation. *IEEE Transactions on Robotics and Automation*, 18(4) :588–596, 2002. 31, 34, 88
- [SB97] Mandayam A. Srinivasan and Cagatay Basdogan. Haptics in virtual environments : Taxonomy, research status, and challenges. *Computers & Graphics*, 21(4) :393–404, 1997. 15
- [SBL<sup>+</sup>07] Jean Sreng, Florian Berge, Jérémie Legarrec, Anatole Lécuyer, and Claude Andriot. Using an event-based approach to improve the multimodal rendering of 6DOF virtual contact. In *Proceedings of the 2007 ACM symposium on Virtual reality software and technology*, pages 165–173. ACM, 2007. 103
- [SBNG03] Xiaojun Shen, Francis Bogsanyi, Liya Ni, and Nicolas D. Georganas. A heterogeneous scalable architecture for collaborative haptics environments. In *The 2nd IEEE International Workshop on Haptic, Audio and Visual Environments and Their Applications*, pages 113–118, 2003. 40, 42
- [SF63] Thomas B. Sheridan and William R. Ferrell. Remote manipulative control with transmission delay. *IEEE Transactions on Human Factors in Electronics*, 4(1) :25–29, 1963. 43, 44
- [SF07] Cristian Secchi and Cesare Fantuzzi. Energy shaping over networks for mechanical systems. In *46th IEEE Conference on Decision and Control*, pages 647–652, 2007. 36

- [SH06] Ganesh Sankaranarayanan and Blake Hannaford. Virtual coupling schemes for position coherency in networked haptic environments. In *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechanics*, pages 853–858, 2006. 68, 96, 100
- [SH08a] Ganesh Sankaranarayanan and Blake Hannaford. Experimental comparison of internet haptic collaboration with time-delay compensation techniques. In *International Conference on Robotics and Automation 2008*, 2008. 68
- [SH08b] Ganesh Sankaranarayanan and Blake Hannaford. Experimental internet haptic collaboration using virtual coupling schemes. In *Haptic interfaces for virtual environment and teleoperator systems*, pages 259–266, 2008. 68, 70
- [She93] Thomas B. Sheridan. Space teleoperation through time delay : review and prognosis. *IEEE Transactions on Automatic Control*, 9(5) :592–606, 1993. 27
- [Sig33] Antonio Signorini. Sopra alcune questioni di elastostatica. In *Proceedings of Atti della Società Italiana per il Progresso delle Scienze*, 1933. 11
- [SLAA09] Jean Sreng, Anatole Lecuyer, Claude Andriot, and Bruno Araldi. Spatialized haptic rendering : Providing impact position information in 6DOF haptic simulations using vibrations. In *Proceedings of the 2009 IEEE Virtual Reality Conference*, pages 3–9, Washington, DC, USA, 2009. IEEE Computer Society. 102
- [SNF05] Alexander Siemers, Iakov Nakhimovski, and Dag Fritzon. Meta-modelling of mechanical systems with transmission linejoints in modelica. In *Proceedings of the 4th International Modelica Conference*, pages 177–182, 2005. 36
- [SRGS00] Eva-Lotta Sallnäs, Kirsten Rasmus-Gröhn, and Calle Sjöström. Supporting presence in collaborative environments by haptic force feedback. *ACM Transactions on Computer-Human Interaction*, 7(4) :461–476, 2000. 42, 43
- [SSvdSF05] Stefano Stramigioli, Cristian Secchi, Arjan J. van der Schaft, and Cesare Fantuzzi. Sampled data systems passivity and discrete port-hamiltonian systems. *IEEE Transactions on Robotics*, 21(4) :574–587, 2005. 21
- [Str01] Stephano Stramigioli. *Modeling and IPC Control of Interactive Mechanical Systems : a Coordinate Free Approach*. Springer-Verlag, 2001. 50, 121
- [SV97] Septimiu E. Salcudean and T. D. Vlaar. On the emulation of stiff walls and static friction with a magnetically levitated input/output device. *Journal of Dynamic Systems, Measurement, and Control*, 119(1) :127–132, 1997. 25
- [SZ03] Eva-Lotta Sallnas and Shumin Zhai. Collaboration meets fitts law : Passing virtual objects with and without haptic force feedback. In *Proceedings of INTERACT 2003, IFIP Conference on Human-Computer Interaction*, pages 97–104, 2003. 42
- [TCO06] J. Tang, Craig R. Carignan, and Pontus A. Ollson. Tandem canoeing over the internet using haptic feedback. In *Haptic interfaces for virtual environment and teleoperator systems*, pages 281–285, 2006. 38

- [TK92] Haruo Takemura and Fumio Kishino. Cooperative work environment using virtual workspace. In *Proceedings of Computer Supported Cooperative Work*, pages 226–232, 1992. 40
- [TN05] Neal A. Tanner and Günter Niemeyer. Improving perception in time-delayed telerobotics. *The International Journal of Robotics Research*, 24(8) :631–644, 2005. v, 33, 92, 93
- [TOS98] James M. Thompson, Mark P. Ottensmeyer, and Thomas B. Sheridan. Human factors in tele-inspection and tele-surgery : Cooperative manipulation under asynchronous video and control feedback. In *Proceedings of the First International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 368–376, 1998. 44
- [WH95] Parris Wellman and Robert D. Howe. Towards realistic vibrotactile display in virtual environments. In *Proceedings of ASME Dynamic Systems and Control Division*, 1995. 25
- [YIY00] Yasuyoshi Yokokohji, Takashi Imaida, and Tsuneo Yoshikawa. Bilateral control with energy balance monitoring under time-varying communication delay. In *International Conference on Robotics and Automation 2000*, pages 2684–2689, 2000. 35
- [YTY02] Yasuyoshi Yokokohji, Teruhiro Tsujioka, and Tsuneo Yoshikawa. Bilateral control with time-varying delay including communication blackout. In *10th Symposium On Haptic Interfaces For Virtual Environment & Teleoperator Systems*, 2002. 35
- [YY94] Y. Yokokohji and T. Yoshikawa. Bilateral control of master-slave manipulators for ideal kinesthetic coupling-formulation and experiment. *Robotics and Automation, IEEE Transactions on*, 10(5) :605–620, Oct 1994. 27
- [ZS95] Craig B. Zilles and John Kenneth Salisbury. A constraint-based god-object method for haptic display. In *IEEE/RSJ International Conference on Intelligent Robots and Systems 1995*, pages 146–151, 1995. 24
- [ZS99] Michael Zyda and Sandeep Singhal. *Networked Virtual Environments : Design and Implementation (Siggraph Series)*. ACM Press SIGGRAPH Series, Addison-Wesley Professional, 1999. 37

