

ÉCOLE DOCTORALE

MECANIQUE, THERMIQUE ET GENIE CIVIL

Année 2006

N° B.U. :

Thèse de DOCTORAT

***Diplôme délivré conjointement par
L'École Centrale de Nantes et l'Université de Nantes***

Spécialité : GENIE MECANIQUE
Présentée et soutenue publiquement par :

MAMY - POULIQUEN

le 28 septembre 2006
à l'École Centrale de Nantes

**PROPOSITION D'UN MODELE DE MAIN POUR LA SIMULATION DES INTERACTIONS
HOMME – MACHINE EN ENVIRONNEMENT VIRTUEL
APPLICATION A LA PREVENTION DES RISQUES D'ACCIDENTS AUX MAINS**

JURY

Président :	Daniel THALMANN	Professeur, Ecole Polytechnique Fédérale de Lausanne, Directeur du VRLab
Rapporteurs :	Daniel COUTELLIER Georges DUMONT	Professeur, Université de Valenciennes, laboratoire LAMIH Maître de Conférence, Ecole Normale Supérieure de Cachan, laboratoire IRISA
Examineurs :	Joseph CICCOTELLI Claude ANDRIOT Alain BERNARD	Chef adjoint du département IET, Institut National de Recherche et de Sécurité Expert Senior, Commissariat à l'Energie Atomique, laboratoire LSI Professeur, Ecole Centrale de Nantes, laboratoire IRCCyN

A ma famille,

A Martin,

Remerciements

En premier lieu, je tiens à remercier tout particulièrement le professeur Daniel Thalmann qui m'a fait l'honneur de présider mon jury de thèse ainsi que Daniel Coutellier et Georges Dumont pour avoir accepté d'être mes rapporteurs. Leurs remarques et l'intérêt qu'ils ont porté à ce travail m'ont énormément touchée.

Je remercie également MM. Alain Bernard, Joseph Ciccotelli et Claude Andriot de m'avoir permis de réaliser cette thèse sur un sujet tant ambitieux que passionnant qu'est la modélisation de la main humaine. Merci également d'avoir participé au jury.

Je tiens à signaler que la contribution de l'INRS a été indispensable pour la réalisation des simulations grâce à leurs interfaces et à leur plateforme de Réalité Virtuelle. Les vidéos réalisées ont été déterminantes pour le succès de cette recherche. Je remercie à nouveau Joseph Ciccotelli ainsi que Jacques Marsot de m'avoir soutenue et d'avoir mis à ma disposition le matériel de l'INRS au cours de mes nombreuses missions pour tester mes différents modèles.

Je remercie vivement Yann Pinczon du Sel (ex-INRS) et Fabien Marie (ex-CEA) avec qui j'ai eu le plaisir de travailler afin de réaliser les simulations présentées dans ce manuscrit. Les deux bibliothèques de programmes que nous avons développées ont permis de mettre en évidence les intérêts du modèle de main proposé pour des applications en Réalité Virtuelle.

Je remercie également toutes les personnes contactées au cours de ma thèse ou rencontrées lors des conférences avec qui j'ai apprécié discuter de nos travaux respectifs. Leurs réponses à toutes mes questions et leurs connaissances ont permis d'enrichir mon travail.

Mes remerciements chaleureux vont à l'ensemble des membres de l'équipe IVGI pour leur disponibilité, leur soutien, leur aide et leur bonne humeur. Je n'oublie pas les signatures des dossiers de réinscription ni les répétitions pour les conférences. Encore merci d'avoir été toujours là pour moi malgré la distance...

Merci également aux personnels des secrétariats de l'IRCCyN et de l'Ecole Doctorale pour leur patience et leur aide. Grâce à vous, j'ai pu gérer toutes les formalités administratives et les missions à distance sans le moindre problème.

Je remercie enfin tous ceux avec qui j'ai partagé ces années de dur labeur : les membres du bureau 18, les abonnés à la pause café et au repas de midi, les autres thésards anonymes du CEA, les secrétaires... Leur amitié, leur soutien et leur humour m'ont aidée à avancer.

Je remercie mes fidèles amies et amis pour leur compréhension et leur tolérance pendant ces années de thèse. Vous avez su m'entourer et me soutenir malgré mon « asociabilité croissante » due à ma vie en vase clos pendant cette thèse. Vos encouragements m'ont aidée à garder le moral et à m'accrocher.

Une tendre pensée à ma grand-mère qui m'a toujours encouragée...

Et bien sûr, je remercie mes parents Myra et Marcel, mon frère Maël, et surtout Martin, pour m'avoir supportée (dans tous les sens du terme :-)) tout au long de cette thèse et surtout pour avoir accepté mes retours tardifs, mes absences fréquentes ainsi que les soirs, les nuits, les week-ends et les vacances consacrés à mon travail plutôt qu'à eux.

Martin, merci à toi pour ton soutien sans faille.

Table des Matières

Remerciements	v
Table des Matières	vii
Table des Figures	xii
Table des Tableaux.....	xvii
Notations et Symboles utilisés	xix
INTRODUCTION.....	1
CHAPITRE I : La main, un organe complexe	5
1. Introduction	5
2. La main humaine	5
2.1 Définitions et conventions en anatomie.....	6
2.2 Le squelette de la main	8
3. Le réseau musculaire de la main	10
3.1 Présentation du squelette fibreux.....	10
3.2 La commande musculaire	10
3.3 La mécanique du mouvement.....	15
4. La biomécanique de la main.....	17
4.1 Les contraintes statiques.....	17
4.2 Les contraintes dynamiques.....	18
4.2.1 Les contraintes internes au doigt	18
4.2.2 Les contraintes entre les doigts.....	19
5. La physiologie de la main	20
5.1 La structure de la peau palmaire	20
5.2 Les récepteurs de la peau	21
5.3 Le système des coussinets palmaires	22
6. Conclusion.....	24
6.1 Le squelette de la main	24
6.2 Le réseau musculaire de la main.....	24
6.3 La biomécanique de la main.....	24
6.4 La physiologie de la main.....	25
6.5 Le modèle retenu	25
CHAPITRE II : Présentation des modèles de main existants	29
1 Introduction	29
2 Les modèles de mains pour la simulation et l’animation	29
2.1 Les modèles développés pour communiquer.....	30
2.1.1 Les modèles squelettiques	30
2.1.2 Les modèles surfaciques.....	33
2.1.2.1 Les surfaces polygonales.....	33
2.1.2.2 Les surfaces paramétriques	34

2.1.2.2.1	Définitions	34
2.1.2.2.2	Les courbes de Bézier	35
2.1.2.2.3	Les courbes B-splines	35
2.1.2.3	Les déformations de forme libre	37
2.1.2.4	La méthode de skinning	38
2.1.3	Les modèles volumiques	39
2.1.3.1	Les primitives simples.....	39
2.1.3.2	Les volumes construits à partir de surfaces implicites	39
2.1.3.3	Les volumes construits à partir des surfaces de convolution	41
2.1.3.4	Les volumes paramétriques.....	42
2.1.3.4.1	Les surfaces de Bézier	42
2.1.3.4.2	Les surfaces NURBS	42
2.1.4	Conclusion.....	43
2.2	Les modèles développés pour manipuler.....	44
2.2.1	Les modèles pour une préhension automatique.....	44
2.2.2	Les modèles pour une préhension interactive.....	46
2.2.3	Les modèles basés sur la physique	49
2.2.3.1	Les modèles masses/ressorts	49
2.2.3.2	La méthode des éléments finis	52
2.2.4	Conclusion.....	56
2.3	Conclusion.....	57
3	Les modèles physiologiques de la main.....	58
3.1	Définitions	58
3.2	Les modèles de pulpe existants.....	59
3.2.1	Les modèles de déformation uni-axiaux.....	60
3.2.2	Les modèles de déformation physiques.....	62
3.2.2.1	Les modèles ressorts/amortisseurs	62
3.2.2.2	Les modèles éléments finis	65
3.3	Conclusion.....	67
4	Définition de notre modèle.....	68
CHAPITRE III : Pilotage de la main en interactif		71
1	Introduction	71
2	Le modèle squelettique proposé	71
3	Modélisation géométrique et cinématique	74
3.1	Définitions et notations.....	74
3.1.1	Matrice de position d'un corps rigide.....	75
3.1.2	Torseur cinématique	75
3.1.3	Torseur des efforts.....	76
3.1.4	Théorème de Chasles.....	77
3.1.5	Description des chaînes cinématiques arborescentes.....	77
3.2	Description du modèle géométrique direct de la main virtuelle.....	77
3.2.1	Les matrices de changement de base.....	78
3.2.2	Les matrices homogènes.....	81
3.3	Les équations de la cinématique	86
3.3.1	Détermination de la matrice Jacobienne pour un doigt	86
3.3.2	Détermination de la matrice Jacobienne pour la main virtuelle.....	87
4	Utilisation de la capture de mouvements	89
4.1	Les systèmes de localisation existants.....	89
4.1.1	Les capteurs électromagnétiques	89
4.1.2	Les capteurs acoustiques	89
4.1.3	Les capteurs optiques	90
4.2	Les gants sensitifs.....	90
4.2.1	Les gants à fibre optique.....	90
4.2.2	Les gants à variation de résistances	91
4.3	Choix du matériel	91

4.4	Couplage entre le monde réel et le monde virtuel	92
5	Description du contrôle 6D	93
5.1	Description et paramétrages	94
5.2	Le contrôle-commande	95
5.2.1	Les données du système de capture de mouvements	95
5.2.2	Les données du moteur physique.....	96
5.2.3	L'asservissement	96
5.2.3.1	Contrôle de la main	96
5.2.3.1.1	Torseur d'efforts associé à la raideur	97
5.2.3.1.2	Torseur d'efforts associé à l'amortisseur	97
5.2.3.1.3	Torseur d'efforts total appliqué à la main virtuelle	98
5.2.3.2	Contrôle des doigts.....	98
5.2.3.2.1	Contrôle articulaire des doigts virtuels.....	98
5.2.3.2.2	Contrôle cartésien pour l'extrémité des doigts.....	99
5.2.3.2.3	Exemple d'application	100
5.3	Conclusion.....	102
6	Stabilité.....	102
6.1	Stabilité numérique.....	103
6.2	Etude de la stabilité de la simulation	103
6.2.1	Résolution de l'équation de la dynamique dans le domaine temporel.....	103
6.2.1.1	Calcul des solutions de l'équation différentielle	104
6.2.1.2	Critères de stabilité des systèmes analogiques	105
6.2.1.2.1	Critère de Routh.....	105
6.2.1.2.2	Critère de Hürwitz	106
6.2.1.3	Exemple d'application.....	107
6.2.2	Résolution de l'équation de la dynamique dans le domaine discret	108
6.2.2.1	Discretisation de l'équation différentielle	109
6.2.2.2	Etude de la stabilité du système discret.....	109
6.3	Conclusion	111
7	Mise en oeuvre	112
7.1	Structure de données pour la construction de l'environnement.....	112
7.2	Structure de données pour les fonctions d'asservissements	113
7.2.1	Les informations issues de la capture de mouvements	114
7.2.2	Les informations issues des contrôleurs	114
8	Conclusion.....	116
CHAPITRE IV : Proposition d'un modèle de préhension		119
1	Introduction	119
2	La détection de collision	119
2.1	La détection du contact.....	120
2.1.1	Le calcul de distances.....	120
2.1.2	La recherche d'un plan séparateur.....	120
2.1.3	Conclusion.....	120
2.2	La recherche de proximité et la détection approximative du contact	121
2.2.1	Les volumes englobants	121
2.2.2	La partition de l'espace	122
2.2.3	Les tests d'intersection	123
2.2.4	Conclusion.....	124
2.3	La détection de collision pour des objets déformables	124
2.3.1	La mise à jour des hiérarchies de volumes englobants	125
2.3.2	Le découpage spatial	125
2.3.3	L'auto-collision	125
2.3.4	Les champs de distances.....	125
2.3.5	Conclusion sur la détection de collision pour objets déformables.....	126
2.4	La solution proposée pour la détection de collision	126
2.4.1	Les conditions expérimentales.....	126

2.4.2	La construction de l'arbre.....	126
2.4.3	Le calcul de distances entre les objets de la scène.....	129
2.4.4	La détection d'interférences.....	130
2.4.5	Conclusion sur la méthode programmée.....	134
3	Le traitement du contact.....	134
3.1	Définitions.....	135
3.2	Les méthodes de gestion de contraintes existantes.....	136
3.2.1	La méthode des pénalités.....	136
3.2.2	La méthode à base de contraintes.....	137
3.2.3	La méthode des impulsions.....	137
3.2.4	La méthode pour des chaînes articulées issue de la Robotique.....	138
3.2.5	La méthode utilisée en Mécanique pour des milieux granulaires.....	139
3.2.6	Conclusion.....	139
3.3	Le modèle de contact proposé.....	139
3.3.1	Définition de la configuration initiale.....	140
3.3.2	Définition de l'espace de contact.....	141
3.3.3	Intégration du mouvement rigide d'une chaîne arborescente.....	144
3.3.4	Résolution des contacts frottants.....	145
3.3.5	Evaluation expérimentale.....	147
3.3.5.1	Les conditions expérimentales.....	147
3.3.5.2	Principe de l'algorithme.....	147
3.3.5.3	Exemple d'application.....	148
3.3.6	Conclusion sur le modèle de contact mis en oeuvre.....	150
4	Le modèle de déformations.....	151
4.1	Le principe des éléments finis.....	151
4.2	Coussinets déformables en petites déformations.....	152
4.2.1	Définitions et notations.....	152
4.2.2	Formulation variationnelle du problème.....	153
4.2.3	Discrétisation des coussinets.....	154
4.2.4	Calcul de la contrainte à partir de la loi de comportement.....	157
4.2.5	Résolution du problème discrétisé.....	158
4.3	Principe des algorithmes développés.....	160
4.4	Les conditions expérimentales.....	161
4.5	Le modèle dans le cas 2D.....	161
4.5.1	Description du modèle.....	161
4.5.2	Simulations et résultats.....	162
4.5.2.1	Flexion d'un doigt.....	162
4.5.2.2	Préhension d'un cylindre rigide dans le cas 2D sans frottement.....	162
4.5.2.3	Préhension d'un cylindre rigide dans le cas 2D avec frottement.....	163
4.5.2.4	Résultats des tests.....	163
4.5.3	Conclusion dans le cas 2D.....	165
4.6	Le modèle dans le cas 3D.....	165
4.6.1	Simulations et résultats.....	165
4.6.2	Conclusion dans le cas 3D.....	166
4.7	Conclusion pour le modèle déformable développé.....	166
5	Conclusion.....	167
CHAPITRE V : Pour une meilleure intégration de l'Homme à la conception		
d'équipements de travail..... 171		
1	Introduction.....	171
2	La simulation d'une presse plieuse.....	172
2.1	Principe de fonctionnement.....	172
2.2	La presse plieuse virtuelle.....	173
3	La plateforme de Réalité Virtuelle de l'INRS.....	174

4	L'intégration de la prévention des risques	179
4.1	Utilisation de la réalité virtuelle pour la prévention des risques professionnels.....	179
4.2	Les modèles de situation de travail.....	181
4.3	L'indice de risque dynamique	182
5	Les applications développées	184
5.1	La simulation d'une opération de pliage de tôle avec du retour d'efforts	185
5.1.1	Description	185
5.1.2	Objectif(s).....	185
5.1.3	Résultats	185
5.2	L'interaction avec l'environnement virtuel	186
5.2.1	Description	186
5.2.2	Objectif(s).....	186
5.2.3	Résultats	187
5.3	Le pliage avec du retour d'effort et de la capture de mouvements.....	188
5.3.1	Description	188
5.3.2	Objectif(s).....	188
5.3.3	Résultats	188
5.4	Conclusions des applications.....	190
6	La validation du simulateur de presse plieuse.....	190
7	Conclusion.....	191
	CONCLUSION.....	193
	REFERENCES BIBLIOGRAPHIQUES.....	197
	PUBLICATIONS PERSONNELLES	209
	ANNEXES.....	211
	A.1. Compléments sur les caractéristiques de la main.....	213
	A.2. Les modèles de contact et de distribution de pression	226
	A.3. L'Analyse en Composantes Principales (ACP).....	234
	A.4. Compléments sur les Eléments Finis	237
	A.5. Diagrammes des Classes de la Bibliothèque « Collision »	242

Table des Figures

INTRODUCTION	1
CHAPITRE I : La main, un organe complexe	5
Figure I.1 : La main humaine transparente (version simplifiée)	6
Figure I.2 : Le repère corporel pour l'étude des mouvements	7
Figure I.3 : Les mouvements des doigts	7
Figure I.4 : L'anatomie de la main	8
Figure I.5 : Les mouvements du poignet	8
Figure I.6 : Les degrés de liberté de la main	9
Figure I.7 : Génération de mouvements	11
Figure I.8 : Le modèle de Hill	12
Figure I.9 : La relation force/longueur pour un muscle	13
Figure I.10 : La relation force/vitesse pour un muscle	13
Figure I.11 : Puissance développée par un muscle actif	14
Figure I.12 : Force exercée quand l'activation musculaire est maximale	14
Figure I.13 : Les tendons fléchisseurs des doigts	15
Figure I.14 : Les poulies proximales et distales des doigts	16
Figure I.15 : La modélisation de l'index	17
Figure I.16 : Les butées articulaires pour l'index	18
Figure I.17 : Coupe de peau palmaire	21
Figure I.18 : Reconstruction d'un fragment de peau avec les récepteurs	21
Figure I.19 : Coussinets adipeux palmaires	23
Figure I.20 : Coupe d'un doigt	23
Figure I.21 : Les éléments qui empêchent le glissement lors des prises	23
Figure I.22 : Les couches étudiées pour une modélisation réaliste	26
CHAPITRE II : Présentation des modèles de main existants	29
Figure II.23 : Le modèle cinématique utilisé par le système DigitEyes	31
Figure II.24 : Un modèle cinématique de main à 20DDL	31
Figure II.25 : Le modèle cinématique utilisé par le système GREFIT	32
Figure II.26 : Une main volumique rigide utilisée pour la langue des signes	32

Figure II.27 : Le modèle cinématique de la main de MAN3D	32
Figure II.28 : Maillage triangulaire de la main	34
Figure II.29 : Une courbe B- spline cubique	36
Figure II.30 : Paramétrage de l'objet pour les FFD	37
Figure II.31 : Principe des déformations libres	37
Figure II.32 : Effondrement du coude en flexion	38
Figure II.33 : Rétrécissement du bras en rotation	38
Figure II.34 : Modélisation de la main par des surfaces implicites	40
Figure II.35 : Principe des metaballs	41
Figure II.36 : Modélisation de la main par 16 metaballs	41
Figure II.37 : La main modélisée par des surfaces de convolution	42
Figure II.38 : Le modèle cinématique rigide	45
Figure II.39 : Différentes prises d'objets simulées automatiquement	45
Figure II.40 : Le modèle de main avec des capteurs aux articulations	46
Figure II.41 : Le modèle de main déformable géométriquement	47
Figure II.42 : Serrage de mains entre deux acteurs virtuels	47
Figure II.43 : Progression de la main avec une guitare	48
Figure II.44 : Serrage de deux mains par une méthode basée sur l'exemple	49
Figure II.45 : Illustration d'une partie d'un système masse/ressort	50
Figure II.46 : Modélisation des muscles de la main par un système masses/ressorts	52
Figure II.47 : Modélisation du doigt par un maillage d'éléments finis volumiques	54
Figure II.48 : Saisie d'une balle déformable modélisée par des éléments finis	55
Figure II.49 : Modélisation d'un doigt par trois coussinets élasto-statiques individuels	56
Figure II.50 : Simulation interactive de contact entre le doigt et des objets rigides	56
Figure II.51 : Les trois modèles de base	59
Figure II.52 : Le modèle Waterbed pour la pulpe d'un primate	60
Figure II.53 : Membrane remplie d'un fluide incompressible	61
Figure II.54 : Relation force/déplacement obtenue avec le modèle de Serina et al.	61
Figure II.55 : Principe des modèles ressorts/amortisseurs	62
Figure II.56 : Le modèle non linéaire de Gulati et Srinivasan	63
Figure II.57 : Relation force totale/déplacement obtenue avec le modèle de Pawluk	64
Figure II.58 : Relations pression/déplacement estimées avec le modèle de Pawluk	64
Figure II.59 : Modèle éléments finis 2D élastique linéaire de la pulpe de Dandekar	65
Figure II.60 : Modèle éléments finis 3D élastique linéaire de la pulpe de Dandekar	66

Figure II.61 : Modèle éléments finis 3D élastique linéaire déformé de Dandekar	66
Figure II.62 : Modèle EF 2D non-linéaire viscoélastique de la pulpe de Wu et al.	66
Figure II.63 : Premier mode de déformation à $t=1s$	67
Figure II.64 : Relation force/déplacement en fonction des modes de vibration	67
CHAPITRE III : Pilotage de la main en interactif	71
Figure III.65 : Le modèle cinématique de la main virtuelle.....	72
Figure III.66 : La main virtuelle.....	73
Figure III.67 : Les repères de la main virtuelle	78
Figure III.68 : Position du repère $\mathcal{R}_{\text{doigt}}^1$ par rapport au repère d'origine.....	79
Figure III.69 : Position des repères pour les mouvement de flexion/extension	80
Figure III.70 : Schéma cinématique d'un doigt de la main virtuelle.....	81
Figure III.71 : Le système Vicon	91
Figure III.72 : Le CyberGlove.....	92
Figure III.73 : Le principe du couplage virtuel	92
Figure III.74 : Le contrôleur.....	94
Figure III.75 : Repères utilisés	95
Figure III.76 : Schéma bloc pour l'asservissement d'un doigt	102
Figure III.77 : Structure de données utilisée pour la scène et le modèle de main.....	113
Figure III.78 : Structure de données utilisée pour la capture de mouvements	115
Figure III.79 : Structure de données utilisée pour les asservissements	116
CHAPITRE IV : Proposition d'un modèle de préhension	119
Figure IV.80 : Les différents types de volumes englobants	121
Figure IV.81 : Construction d'un arbre Sphère par la méthode descendante	122
Figure IV.82 : Différents types de décomposition spatiale.....	122
Figure IV.83 : Hiérarchies construites lors de la partition de l'espace	123
Figure IV.84 : Tests d'interférence entre deux objets.....	124
Figure IV.85 : Sphère englobante calculée à partir du centre géométrique	126
Figure IV.86 : Hiérarchie de volumes englobants pour une sphère.....	128
Figure IV.87 : Temps de construction des hiérarchies de sphères (pré-calcul)	129
Figure IV.88 : Calcul de distance entre deux sphères	130
Figure IV.89 : Exemple de descente d'arbres lors d'une collision	131
Figure IV.90 : Test de triangles entre deux sphères maillées identiques	133
Figure IV.91 : Temps de calcul de la détection de collision en fonction des noeuds	133

Figure IV.92 : Exemple de liaison unilatérale.....	136
Figure IV.93 : Exemple de liaison bilatérale.....	136
Figure IV.94 : Méthode analytique pour les impulsions.....	138
Figure IV.95 : Contact entre deux objets A et B.....	141
Figure IV.96 : Exemple utilisé pour évaluer le modèle de contact.....	148
Figure IV.97 : Temps de construction de l'espace des contacts.....	149
Figure IV.98 : Temps d'exécution pour la résolution du LCP.....	150
Figure IV.99 : Définition de la déformation.....	152
Figure IV.100 : Définition de la contrainte.....	153
Figure IV.101 : Fonctions d'interpolation pour un triangle.....	155
Figure IV.102 : Fonctions d'interpolation pour un tétraèdre.....	156
Figure IV.103 : Description du modèle de doigt utilisé.....	161
Figure IV.104 : Pilotage d'un doigt déformable et contraintes internes.....	162
Figure IV.105 : Saisie d'un cylindre avec deux doigts.....	162
Figure IV.106 : Saisie d'un cylindre par le bout des doigts.....	163
Figure IV.107 : Temps de calcul en fonction du coefficient de frottement.....	163
Figure IV.108 : Force en fonction de déplacement et du module d'Young.....	164
Figure IV.109 : Expérience simulée de compression d'une sphère élastique linéaire sur un plan.....	166
CHAPITRE V : Pour une meilleure intégration de l'Homme à la conception d'équipements de travail.....	171
Figure V.110 : Vue schématique d'une presse plieuse.....	172
Figure V.111 : La presse virtuelle.....	173
Figure V.112 : L'atelier virtuel.....	173
Figure V.113 : Le bras à retour d'effort Virtuose 6D.....	175
Figure V.114 : Simulation dynamique d'un pliage de tôle avec le bras à retour d'effort..	175
Figure V.115 : Les marqueurs utilisés pour le suivi des mouvements de l'opérateur.....	176
Figure V.116 : Utilisation du CyberGlove pour animer la main virtuelle.....	176
Figure V.117 : Utilisation du CyberGlove dans l'environnement de la presse plieuse....	177
Figure V.118 : La plateforme de RV de l'INRS.....	177
Figure V.119 : Architecture de la simulation de la presse plieuse.....	178
Figure V.120 : Utilisation des faisceaux laser pour protéger les mains de l'opérateur.....	182
Figure V.121 : Estimation du risque pour une configuration donnée.....	183
Figure V.122 : Affichage des facteurs de risque.....	183

Figure V.123 : Illustration de l'indice de risque dynamique avec la main	184
Figure V.124 : Activation du dispositif de sécurité avec la main	184
Figure V.125 : Chargement de la configuration de la presse plieuse.....	185
Figure V.126 : Simulation d'un pliage de tôle	186
Figure V.127 : Manipulation avec les deux mains.....	187
Figure V.128 : Saisie de la tôle avec une main virtuelle articulée	187
Figure V.129 : Maintien de la tôle dans la main et lâcher	188
Figure V.130 : Pliage de tôle simulé avec les mains virtuelles et le retour d'effort	189
Figure V.131 : Suivi des mains de l'utilisateur dans la simulation.....	189
CONCLUSION.....	193
REFERENCES BIBLIOGRAPHIQUES.....	197
PUBLICATIONS PERSONNELLES	209
ANNEXES.....	211

Table des Tableaux

INTRODUCTION	1
CHAPITRE I : La main, un organe complexe	5
Tableau I.1 : Les limites articulaires	17
CHAPITRE II : Présentation des modèles de main existants	29
CHAPITRE III : Pilotage de la main en interactif	71
Tableau III.2 : Débattements angulaires de notre modèle squelettique	73
Tableau III.3 : Longueurs des phalanges pour un majeur de 12 cm	73
Tableau III.4 : Résultats des analyses de stabilité en continu	108
CHAPITRE IV : Proposition d'un modèle de préhension	119
Tableau IV.5 : Les temps de construction des arbres en fonction des maillages.....	128
Tableau IV.6 : Les temps d'exécution pour la résolution des contacts.....	149
Tableau IV.7 : Influence du module d'Young E sur les forces et les déplacements.....	164
Tableau IV.8 : Les temps d'exécution moyens en fonction du module d'Young E	165
CHAPITRE V : Pour une meilleure intégration de l'Homme à la conception d'équipements de travail	171
CONCLUSION	193
REFERENCES BIBLIOGRAPHIQUES	197
PUBLICATIONS PERSONNELLES	209
ANNEXES	211

Notations et Symboles utilisés

Notations générales :

- a : une lettre minuscule représente un scalaire
- \mathbf{v} : une lettre minuscule en gras représente un vecteur
- x_i : élément constitutif d'un vecteur
- \mathbf{M} : une lettre majuscule en gras représente une matrice ou un tenseur
- a_{ij} : élément constitutif d'une matrice ou d'un tenseur
- \mathbf{I}_n : matrice identité de dimension $n \times n$

- $\mathbf{a} \cdot \mathbf{b}$: produit scalaire de deux vecteurs
- $\mathbf{a} \wedge \mathbf{b}$: produit vectoriel de deux vecteurs
- $\text{tr}(\mathbf{M})$: trace d'une matrice
- $\det(\mathbf{M})$: déterminant d'une matrice
- \mathbf{M}^T : transposée d'une matrice (même écriture pour un vecteur)

Notations spécifiques au Chapitre III :

- \mathcal{R} : repère
- P : point de l'espace
- m : masse
- \mathfrak{I} : matrice d'inertie
- J : matrice Jacobienne
- F : force
- τ : couple

- \mathbf{X} : vecteur position $\dot{\mathbf{X}} = \frac{d\mathbf{X}}{dt}$: vecteur vitesse $\ddot{\mathbf{X}} = \frac{d^2\mathbf{X}}{dt^2} = \frac{d\dot{\mathbf{X}}}{dt}$: accélération

Notations spécifiques au Chapitre IV :

- \mathbf{q} : vecteur position
- \mathbf{u} : vecteur déplacement
- ε_{ij} : élément constitutif du tenseur des déformations, tenseur d'ordre 2
- σ_{ij} : élément constitutif du tenseur des contraintes, tenseur d'ordre 2
- E : module de Young
- ν : coefficient de Poisson
- λ et μ : coefficients de Lamé
- δ_{ij} : symbole de Kronecker, égal à 1 si $i=j$, égal à 0 sinon

Soit E un espace euclidien de dimension n . On définit un tenseur T d'ordre p par une application p -linéaire de $E \times E \times \dots \times E$ (p fois) dans \mathbb{R} .

- Par exemple, un tenseur d'ordre 2 est défini par l'opérateur G tel que :

$$G : E \times E \rightarrow \mathbb{R}$$

$$\{X, Y\} \mapsto X \cdot Y$$

où le produit scalaire de deux vecteurs $X \cdot Y$ est un tenseur du second ordre.

- Dans le cas d'un tenseur d'ordre 3, on obtient l'application T telle que :

$$T : E \times E \times E \rightarrow \mathbb{R}$$

$$\{X, Y, Z\} \mapsto T(X, Y, Z)$$

Si on choisit d'écrire les vecteurs X , Y et Z sur une base $\{e_i\}$, on a alors :

$$T(X, Y, Z) = T(e_i, e_j, e_k) X^i Y^j Z^k$$

$$= T_{ijk} X^i Y^j Z^k$$

\otimes : produit tensoriel

Soient P un tenseur d'ordre p et Q un tenseur d'ordre q . On appelle produit tensoriel de P et Q , noté $P \otimes Q$, le tenseur d'ordre $p+q$ défini par :

$$(P \otimes Q)(X_1, X_2, \dots, X_{p+q}) = P(X_1, X_2, \dots, X_p) Q(X_{p+1}, X_{p+2}, \dots, X_{p+q})$$

$$= P_{ij\dots p} Q_{p+i\ p+j\dots q} (e_i \otimes e_j \otimes \dots \otimes e_p \otimes \dots \otimes e_q)$$

$\bar{\otimes}$: produit tensoriel contracté

Soient P un tenseur d'ordre $p \geq 1$ et Q un tenseur d'ordre $q \geq 1$. On appelle produit tensoriel contracté de P et Q , noté $P \bar{\otimes} Q$, le tenseur d'ordre $p+q-2$ dont les composantes sont obtenues par sommation du dernier indice de P avec le premier indice de Q .

Par exemple, le produit contracté de deux vecteurs est un tenseur d'ordre 0 : il est équivalent au produit scalaire. Ou encore, le produit contracté d'un tenseur d'ordre 2 et d'un vecteur donne un vecteur.

$\overline{\overline{\otimes}}$: produit tensoriel doublement contracté

Soient P un tenseur d'ordre $p \geq 2$ et Q un tenseur d'ordre $q \geq 2$. On appelle produit tensoriel doublement contracté de P et Q , noté $P \overline{\overline{\otimes}} Q$, le tenseur d'ordre $p+q-4$ dont les composantes sont obtenues par sommation des deux derniers indices de P avec les deux premiers indices de Q .

Par exemple, le produit doublement contracté de deux tenseurs du second ordre est un scalaire.

INTRODUCTION

La Réalité Virtuelle est un domaine en plein essor. Elle offre de nouveaux outils pour de nombreuses applications dans des domaines variés, comme par exemple la conception avec le prototypage virtuel, la médecine avec les simulateurs chirurgicaux, la formation et l'entraînement en environnement virtuel ou encore la prévention des risques d'accidents.

Cette nouvelle technologie permet d'expérimenter et d'interagir avec des modèles et des environnements proches de la réalité et en toute sécurité. Elle offre un bon degré de contrôle sur la simulation car elle permet de planifier tous les événements simulés, ce qui est d'habitude impossible dans la vie réelle.

De nombreux progrès ont été réalisés ces dernières années pour concevoir des produits, des machines ou des environnements complexes avec les techniques de la Réalité Virtuelle. Les modèles développés sont de plus en plus complets et précis. La modélisation de l'Homme, qui évolue et interagit avec cet environnement, a par contre progressé beaucoup moins rapidement. La plupart des simulateurs n'utilisent que des avatars ou des mannequins virtuels possédant un nombre restreint de degrés de liberté en comparaison avec la réalité. Ce sont des modèles simplifiés qui ne permettent pas d'interagir avec l'environnement de manière réaliste. Dans ce travail, nous nous intéressons plus particulièrement à la main car elle est le principal organe d'interaction de l'Homme avec son environnement. Elle sert à prendre de manière active de l'information et à agir.

De nouveaux modèles et algorithmes sont donc nécessaires afin de répondre à ce besoin de réalisme et ainsi simuler le comportement de l'opérateur en situation de travail. L'Institut National de Recherche en Sécurité (INRS) réalise de nombreuses études pour tenter de répondre à ces exigences. Ainsi, les recherches menées au sein de l'INRS ont montré que le risque d'accidents était maximal lors des interactions des opérateurs avec leurs machines. Il apparaît donc essentiel de pouvoir simuler le comportement des opérateurs en situation de travail afin de diminuer les risques physiques qu'ils encourent voire d'empêcher l'accident. De plus, ces études ont mis en évidence que dans 26% des cas les accidents de travail concernaient les mains. D'où l'intérêt de simuler les interactions de l'opérateur avec son équipement de travail en développant un modèle de main permettant de réaliser les tâches manuelles de manière réaliste.

C'est pour répondre à ce besoin que l'INRS s'est intéressé aux systèmes de Réalité Virtuelle et a engagé le projet de recherche EVICS – Environnement Virtuel pour la Conception de Systèmes sûrs. Celui-ci permettra d'évaluer l'intérêt de ces nouvelles technologies à des fins de prévention des risques professionnels ainsi que de simuler les futures interactions de l'opérateur avec son équipement de travail.

Comme le pourcentage des accidents aux mains augmente considérablement dans le cas du travail avec des machines à métaux, une presse plieuse a été choisie comme équipement de travail à simuler en environnement virtuel. L'objectif est donc de développer un simulateur informatique mettant en œuvre une simulation de pliage de tôle sur une plateforme de Réalité Virtuelle. Il s'agira de modéliser la presse plieuse dans l'atelier, de choisir et de coupler les périphériques nécessaires à la réalisation de cette opération et de générer en images de synthèse les mains de l'opérateur et leur comportement sous l'action des forces de contact.

Les travaux menés au cours de cette thèse s'inscrivent précisément dans cet objectif et

présentent un modèle de main destiné à la simulation des interactions Homme-machine en environnement virtuel. La conception d'un tel modèle nécessite de disposer de données anatomiques, biomécaniques et mécaniques précises de la main humaine, ainsi que de méthodes de calcul suffisamment rapides pour permettre l'utilisation de cette main virtuelle dans des conditions de temps réel. Dans ce mémoire, nous proposons un modèle de main ayant la même structure et le même comportement que la main humaine et permettant de réaliser les mêmes tâches manuelles en environnement virtuel. Pour cela, nous avons intégré à notre modèle de main les caractéristiques physiques et physiologiques de la main humaine et nous l'avons couplé à différents périphériques afin de permettre à l'utilisateur d'interagir en temps réel avec l'environnement virtuel. Bien que ce travail de recherche s'effectue dans le cadre de la prévention des accidents aux mains, le modèle proposé est suffisamment générique pour pouvoir être utilisé dans une grande variété d'applications.

Ce manuscrit est organisé en cinq chapitres. Le premier aborde différents aspects de la main humaine. Nous décrivons tout d'abord l'anatomie et la biomécanique de la main. Cette description met en évidence la complexité de sa structure et de sa commande qui lui confère une grande dextérité en autorisant une multitude de configurations. Ensuite, nous présentons les propriétés physiologiques et mécaniques de la main qui font d'elle un outil de préhension et de manipulation aussi puissant que précis. Tout au long de ce mémoire, nous tenterons de restituer un comportement de notre main virtuelle le plus réaliste possible au cours de la simulation des tâches manuelles en environnement virtuel.

Le Chapitre II est consacré aux modèles de mains et de pulpes des doigts existants à ce jour. Il permet de présenter les structures, les cinématiques et les propriétés mécaniques pouvant être retenues pour notre étude ou pouvant orienter nos développements. Ainsi, après la description des modèles géométriques très répandus en vision, nous nous intéresserons aux modèles permettant la saisie et/ou la manipulation d'objets, et plus particulièrement aux modèles basés sur la physique. En effet, il est important que la main virtuelle se comporte de manière réaliste, tant sur la cinématique qu'au niveau des déformations au contact.

Le Chapitre III propose une méthode d'animation de la main dans un environnement virtuel basée sur la capture de mouvements. Elle consiste plus précisément en deux types de contrôle en position. Le premier est utilisé pour positionner le modèle de main (position et orientation de la paume) dans l'environnement virtuel en fonction des déplacements et des gestes de l'opérateur qui sont traqués par un système de capture de mouvements optique. Ce contrôle des six degrés de liberté de la main dans l'espace, est basé sur un couplage de type ressort-amortisseur entre les configurations des mains de l'opérateur et celles des mains virtuelles. Le second contrôle en position permet de simuler la fermeture des doigts sur un objet virtuel à partir des mesures angulaires des phalanges de l'opérateur obtenues par un gant de données et du modèle géométrique de la main virtuelle. Les asservissements permettent d'interagir de manière stable et en temps réel avec l'environnement virtuel au cours de la simulation physique.

Le Chapitre IV présente la méthode de calcul mise en œuvre afin de résoudre les contacts multiples entre la main – la paume et/ou les doigts - et l'objet saisi. Le problème de préhension est résolu en développant une méthode basée sur une détection de collision, un modèle de contact (avec et sans frottement) et un modèle de déformation. Elle met en œuvre le frottement de Coulomb et la méthode des éléments finis afin que la main ait un comportement physique lorsqu'elle est en contact avec l'environnement virtuel.

Des exemples sont présentés dans le Chapitre V et ils permettent d'apprécier les développements réalisés. Après la description du matériel et des logiciels utilisés sur la plateforme de Réalité Virtuelle, nous présentons la presse plieuse virtuelle ainsi que les outils développés par l'INRS pour la prévention des risques d'accidents. Non seulement les simulations réalisées montrent que l'opérateur peut interagir en temps réel et de manière stable avec la scène virtuelle (presse, tôle métallique ou outil) par le biais de périphériques comme un système de capture de mouvements ou un bras à retour d'efforts. Mais, elles mettent également en évidence l'intérêt de la Réalité Virtuelle pour la prévention des risques d'accidents.

Enfin, nous terminerons ce manuscrit par une conclusion générale dressant le bilan des travaux réalisés. Nous dégagerons également les perspectives possibles de cette étude dans le domaine des simulations des interactions Homme-machine en environnement virtuel et dans celui de la simulation de la préhension en temps réel.

CHAPITRE I : La main, un organe complexe

La main est le principal organe d'interaction de l'Homme avec son environnement. Elle permet en effet de toucher, saisir et manipuler des objets dans l'environnement. La diversité des tâches réalisables avec la main est rendue possible grâce à sa structure et à ses propriétés mécaniques et physiologiques. Du fait de cette complexité biologique, la modélisation de la main humaine demeure une tâche difficile. Jusqu'à ce jour, il n'existe aucun modèle qui tienne compte de tous ces éléments, et encore moins pour des applications en temps réel. L'objectif de cette thèse étant la simulation des interactions Homme-machine en environnement virtuel, nous nous intéressons à la modélisation de la main pour des applications en Réalité Virtuelle. Pour ce faire, nous étudions les caractéristiques de la main humaine afin d'obtenir un modèle qui soit le plus réaliste possible.

1. Introduction

Dans ce chapitre, nous présentons les principales caractéristiques de la main humaine. Dans un premier temps, nous décrirons le squelette qui possède de nombreux degrés de liberté ; ce qui confère à la main sa grande mobilité. Ensuite, nous détaillerons le système musculo-tendineux qui se compose de nombreux muscles et tendons. Ils se révèlent être importants pour la précision, la force ou encore la dextérité des gestes. Enfin, nous terminerons par les caractéristiques physiologiques de la main en décrivant son système tactile (i.e. les coussinets et les capteurs sensoriels) [Tubiana 90] [Rouviere 03] [Kapandji 05] qui permet à l'Homme de ressentir le contact avec les objets et l'environnement. Nous en déduisons alors les caractéristiques que devra respecter notre modèle de main afin d'être le plus conforme possible à la main humaine compte tenu des technologies utilisées.

2. La main humaine

Dans ce paragraphe, nous commencerons par l'analyse du squelette en détaillant la cinématique et les degrés de liberté. Puis, nous présenterons le système musculo-tendineux – autrement appelé « squelette fibreux » - et sa fonction d'actionnement. Enfin, nous nous intéresserons aux propriétés mécaniques de la main comme par exemple les coussinets palmaires qui jouent un rôle important dans les tâches manuelles.

Cet organe complexe - constitué entre autres d'os, de muscles, de tendons, de ligaments, de nerfs et de vaisseaux sanguins – est l'instrument de la préhension. Nous sommes en effet capables de réaliser avec notre main des prises en force ainsi que des manipulations délicates exigeant beaucoup de précision dans le geste. Elle nous offre aussi la possibilité d'interagir avec l'environnement et de réaliser de nombreuses prises de par le nombre des doigts et les particularités osseuses du pouce et de l'auriculaire.

La Figure I.1, qui est une version simplifiée de la main transparente de Kapandji [Kapandji 05], met en évidence la complexité de la main humaine en présentant un certain nombre de détails anatomiques :

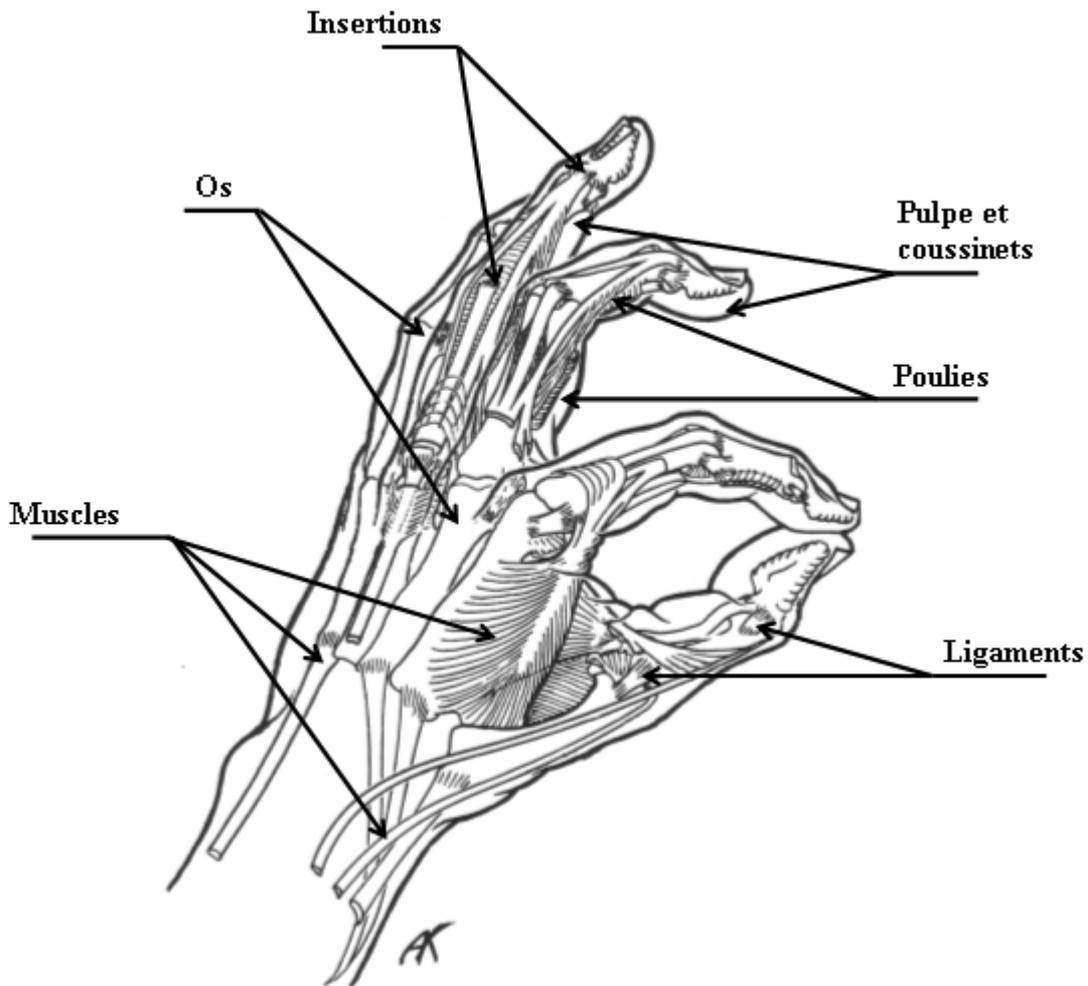


Figure I.1 : La main humaine transparente (version simplifiée)

Afin de mieux connaître la main et de mieux comprendre son fonctionnement, nous allons décrire ses principales fonctions. Nous commencerons par donner des définitions des termes spécifiques de l'anatomie descriptive et fonctionnelle utilisés dans les descriptions de la main.

2.1 Définitions et conventions en anatomie

Pour étudier les mouvements, on utilise différents repères. Le système d'axe de référence est le repère galiléen. On définit un repère corporel dont les axes sont parallèles à ceux du repère galiléen et dont l'origine est située au centre de gravité du corps. La Figure I.2 [Bouisset 02] illustre ce repère corporel. On remarque que pour définir ces plans de référence, le corps est dans une position appelée « position anatomique » : c'est une entité représentée debout, pieds réunis et parallèles et les bras le long du corps.

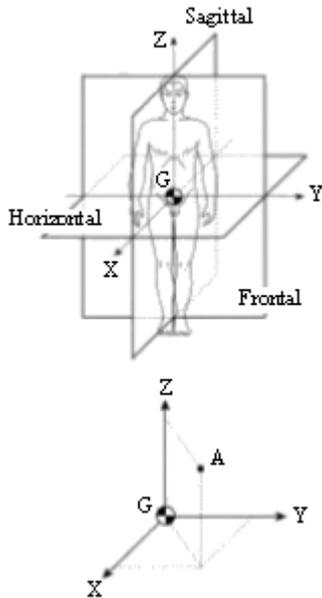


Figure I.2 : Le repère corporel pour l'étude des mouvements

On définit trois plans orthogonaux dans le repère corporel \mathcal{R}_G lié au centre de gravité du corps :

- le plan frontal (GYZ) : il divise le corps en corps ventral et corps dorsal,
- le plan sagittal ou antéro-postérieur (GXZ) : il divise le corps en corps droit et corps gauche,
- le plan horizontal ou transverse (GXY) : il divise le corps en parties supérieure et inférieure.

Sur la base de ce repère et de ces différents plans, peuvent être définis les deux principaux mouvements du corps :

- le mouvement de flexion/extension (noté **f/e**) : ce mouvement s'effectue dans le plan sagittal. La flexion consiste à ramener deux os l'un contre l'autre tandis que l'extension les éloigne.
- le mouvement d'adduction/abduction (noté **a/a**) : ce mouvement est situé dans le plan frontal. Le mouvement d'adduction consiste à ramener les membres vers l'axe médian du corps tandis que l'abduction les écarte.

Les mouvements de la main sont illustrés par la Figure I.3 [Bernier 97] :

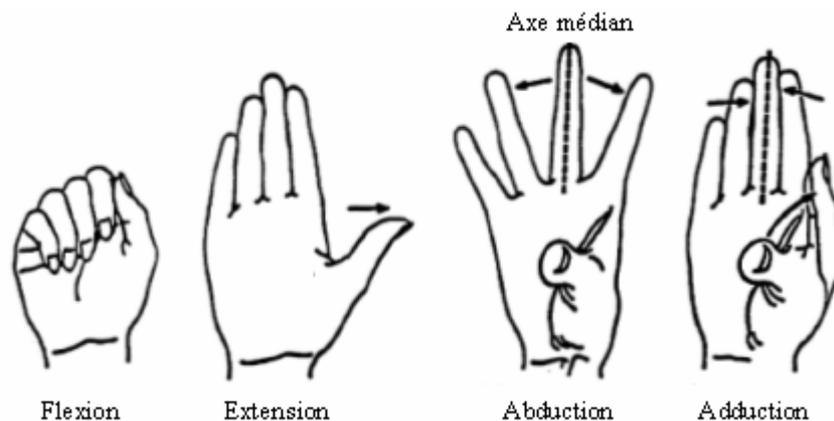


Figure I.3 : Les mouvements des doigts

On utilise également des termes spécifiques pour décrire le squelette. Ainsi, dans une structure arborescente telle que la main, deux os qui ont le même nom sont distingués comme suit :

- proximal : situé en amont de la chaîne,
- distal : situé en aval de la chaîne.

Après ces définitions, nous détaillons dans le paragraphe suivant le squelette de la main.

2.2 Le squelette de la main

Ce squelette est une structure arborescente composée de vingt-sept os [Rouvière 02] [Castaing 99a]. Elle est décrite par la Figure I.4 [Ouhaddi 98] comme suit :

- huit os courts pour le poignet (le carpe),
- cinq métacarpiens qui forment le squelette de la paume de la main,
- 14 phalanges qui constituent le squelette des doigts et qui se décomposent en :
 - cinq phalanges proximales, appelées « phalanges »,
 - quatre phalanges intermédiaires, appelées « phalanges »,
 - cinq phalanges distales, appelées « phalangettes ».

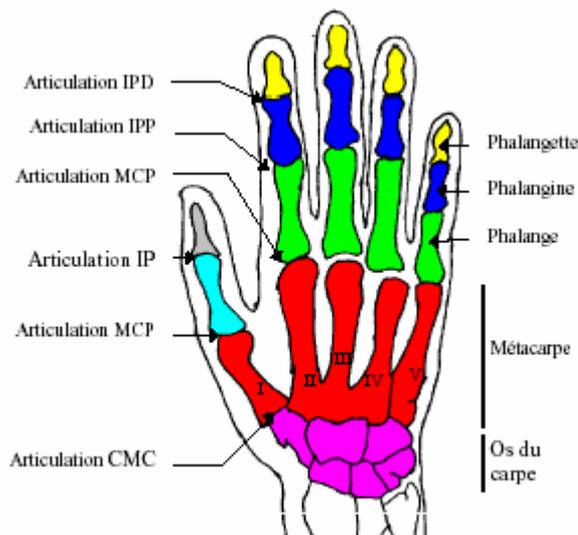


Figure I.4 : L'anatomie de la main

La main possède également de nombreux degrés de liberté dus à son architecture qui lui confèrent une grande mobilité.

Dans ce paragraphe, nous décrivons les principales mobilités du poignet et des doigts mais une description plus complète des mouvements est donnée dans l'Annexe A.1.

Le poignet comporte huit os courts et les deux extrémités (épiphyses) de l'ulna (ou cubitus) et du radius. Les os sont disposés en deux rangées horizontales superposées de quatre os chacune. La main a six degrés de liberté au niveau du poignet (Cf. Figure I.5) [Khale 94] :

- les mouvements de flexion, extension et inclinaison latérale,
- les mouvements de rotation.

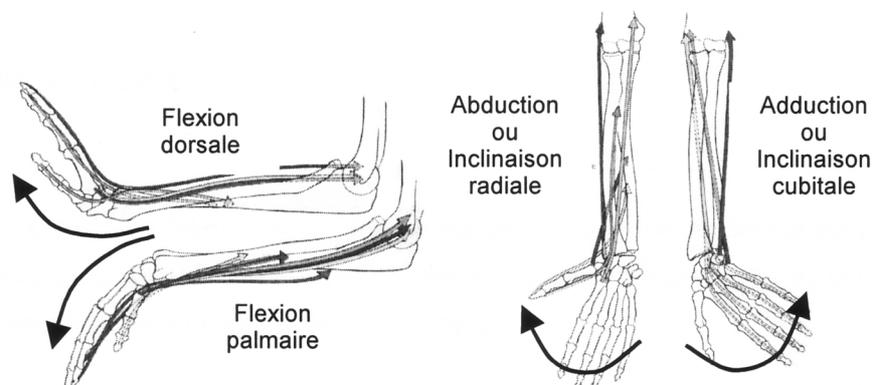


Figure I.5 : Les mouvements du poignet

Tandis que les doigts sont représentés par cinq chaînes polyarticulées [Rouvière 02] [Castaing 99a]. Chaque doigt, à l'exception du pouce, est composé de quatre os (Cf. Figure I.4). Il est modélisé par une série de trois phalanges reliées entre elles par quatre liaisons rotoïdes (pivots). Les degrés de liberté des doigts (à l'exception du pouce) se répartissent ainsi :

- deux degrés de liberté pour les articulations à la base de la première phalange (MCP) : flexion/extension (f/e) et abduction/adduction (a/a),
- un degré de liberté pour les articulations à la base de la deuxième phalange (IPP) : flexion/extension (f/e),
- un degré de liberté pour les articulations à la base de la dernière phalange (IPD) : flexion/extension (f/e).

Le pouce, quant à lui, n'est composé que de trois os [Rouvière 02] [Castaing 99a] car il n'a pas de phalange intermédiaire (Cf. Figure I.4). Il possède une très grande mobilité en comparaison avec les autres doigts grâce à ses cinq degrés de liberté :

- deux degrés de liberté pour l'articulation au niveau du poignet (CMC),
- deux degrés de liberté pour l'articulation à la base de la première phalange (MCP) : flexion/extension (f/e), et très faible amplitude en abduction/adduction (a/a),
- un degré de liberté pour l'articulation de la dernière phalange (IP) : flexion/extension (f/e).

Ainsi, la main est un objet articulé qui possède vingt-sept degrés de liberté avec six degrés de liberté pour le poignet et vingt-et-un degrés de liberté pour les articulations des doigts (Cf. Figure I.6) :



Figure I.6 : Les degrés de liberté de la main

Après avoir détaillé le squelette complet de la main ainsi que ses degrés de liberté, nous présentons maintenant le modèle musculaire qui actionne ce squelette.

3. Le réseau musculaire de la main

Le réseau des muscles et des tendons qui animent le squelette de la main est un système complexe (Cf. Figure I.1 et Annexe A.1) [Tubiana 90] [Castaing 99b] [Kapandji 05]. Dans les paragraphes qui suivent, nous présentons les différents groupes musculaires, puis le modèle de commande musculaire le plus usité – le modèle de Hill - et enfin le mécanisme d'actionnement par des poulies qui permet la transmission des mouvements.

3.1 Présentation du squelette fibreux

En général, trois groupes de muscles sont identifiés pour la main :

- les muscles intrinsèques : ils ont leurs insertions proximales et distales au niveau de la main. Ils sont au nombre de dix-neuf.
- les muscles extrinsèques : ils ont leur origine dans l'avant-bras. Ils se divisent en faisceaux et se prolongent par des tendons pour s'insérer dans les phalanges de différents doigts. Ils sont au nombre de vingt.
- les muscles du poignet : ils permettent d'effectuer les mouvements de rotation, de flexion et d'extension du poignet. Ils sont au nombre de six.

Les muscles intrinsèques commandent les mouvements fins de la main. Ils jouent un rôle important pour la préhension d'un point de vue de la précision et de la coordination. Quant aux muscles extrinsèques, ils interviennent dans le geste en terme de force : ils « verrouillent » les prises. Enfin, les muscles du poignet permettent d'orienter la main lors de la phase d'approche de la saisie d'un objet.

Au total, une cinquantaine de muscles permettent d'actionner la main en réalisant une grande diversité de prises [Brand 81] [Castaing 99b] [Albrecht 03].

Ce réseau musculaire complexe commande une chaîne cinématique qui possède un nombre élevé de degrés de liberté. Par conséquent, plusieurs configurations sont possibles pour réaliser une même tâche manuelle. Dans ce cas de figure, la chaîne d'actionneurs est redondante. Cette flexibilité à la réalisation d'une tâche pose le problème de la commande musculaire. Dans le paragraphe qui suit, les principes de base de la commande musculaire sont exposés.

3.2 La commande musculaire

Depuis plusieurs siècles, de nombreuses études ont été et sont réalisées afin de déterminer l'action des muscles et de comprendre la genèse des mouvements du corps. Elles ont conduit à définir deux familles de modèles décrivant les phénomènes mis en oeuvre lors de l'actionnement des muscles :

- les modèles chimiques,
- les modèles mécaniques.

Dans le cadre de notre modélisation de la main, nous nous intéressons plus particulièrement aux modèles dits « mécaniques ». Dans la suite de ce paragraphe, nous décrivons l'enchaînement des phénomènes mécaniques complexes qui entraînent les mouvements du corps humain.

Tout d'abord, l'excitation musculaire - que l'on appelle également « excitation électrique » - crée un signal nerveux. Ce dernier active les fibres musculaires qui génèrent alors les forces musculaires (modèle de Hill). Ces dernières, couplées avec le modèle géométrique, entraînent des couples articulaires au niveau des articulations où se croisent les muscles et les tendons actionnés. Cette dynamique entraîne les mouvements de la chaîne squelettique [Chèze 05] (Cf. Figure I.7).

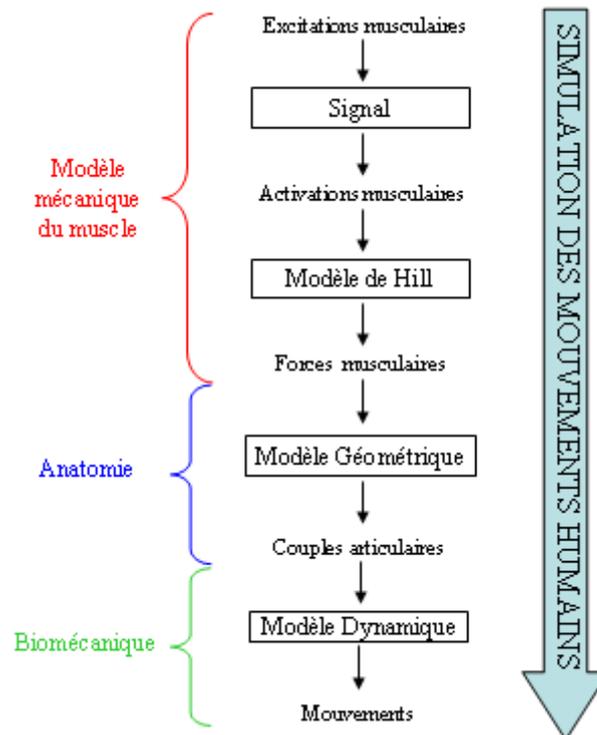


Figure I.7 : Génération de mouvements

En résumé, les mouvements du corps humain, même les plus simples, sont un agencement de contractions de divers muscles qui entrent en jeu à différents stades du mouvement. Comme un muscle ne travaille pas seul mais en collaboration avec d'autres muscles, il est difficile d'identifier la force exercée par un muscle lors d'un mouvement précis.

Nous nous intéressons maintenant aux lois de comportement des muscles qui permettent de générer les forces nécessaires aux déplacements. Ces relations caractéristiques de la mécanique musculaire sont la plupart du temps des relations mathématiques entre l'allongement du muscle et la variation de la force musculaire. Ainsi, les propriétés du muscle, système biologique complexe, sont modélisées par des éléments assemblés comme par exemple un ressort ou un amortisseur. Les modèles obtenus sont alors appelés « modèles rhéologiques ».

Ainsi, les premiers modèles assimilent le muscle activé à un simple ressort, rendant compte de l'effet de la longueur sur la force, ou, à un ressort associé en parallèle à un élément visqueux traduisant l'influence de la vitesse sur la force produite. Grâce à ses travaux sur la contraction musculaire, A.V. Hill a établi le premier modèle d'actionnement des muscles à trois éléments (Cf. Figure I.8) [Hill 38] : deux éléments en série et un troisième élément en parallèle avec les deux précédents. Ce modèle sert encore de base pour l'interprétation de nombreuses expériences.

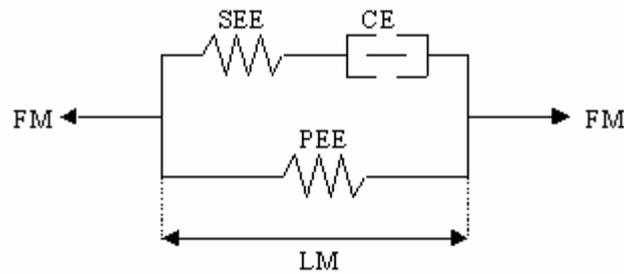


Figure I.8 : Le modèle de Hill

- L'élément contractile (CE : *contractile element*) non activé n'offre aucune résistance à l'étirement. Il est alors extensible au repos mais il se raccourcit quand il est activé par une stimulation électrique. Dans ce cas de figure, la force développée est fonction de la longueur et de la vitesse du raccourcissement du muscle. Cet élément produit la force musculaire active car il est responsable du changement de longueur du muscle ;
- Cet élément contractile est connecté en série à un élément élastique (SEE : *serial elastic element*). Celui-ci explique la chute brutale de tension lorsque le muscle activé en conditions de forces isométriques (à longueur de muscle constante) subit un raccourcissement rapide. Il correspond à l'élasticité musculaire dans ces conditions précises. Son influence est en fait négligeable devant celle des tendons, sauf pour des tendons très courts par rapport au muscle ;
- Ces deux éléments sont en parallèle à un élément élastique (PEE : *parallel elastic element*) qui correspond à l'élasticité musculaire au repos. Il induit une force passive lors de l'allongement des fibres musculaires.

Le modèle de Hill a permis d'identifier les effets de la charge sur les vitesses d'allongement (ou de raccourcissement) des muscles [Hill 38]. Ce modèle prend en compte la longueur du muscle ainsi que la variation de cette longueur, c'est-à-dire la vitesse de contraction musculaire. Ainsi, les propriétés des trois éléments du modèle de Hill responsables de la génération de la force sont définies par les relations suivantes :

- La relation force/longueur est définie pour la contraction musculaire isométrique¹. Au cours d'une contraction isométrique, l'élément série s'allonge alors que l'élément contractile se raccourcit. Lorsque l'élément série s'allonge, l'élément parallèle se raidit et il développe une tension de manière non linéaire [Winter 90]. La force totale développée à l'intérieur du muscle est alors la somme d'une force active générée par l'élément contractile et d'une force passive due à l'étirement de l'élément parallèle. Pour obtenir la courbe force/longueur, il faut déduire de la force mesurée la contribution de l'élément parallèle. Un exemple de courbe force/longueur est donné par la Figure I.9 [Hill 38] [Zajac 89] [Winter 90] [Chèze 05]. La force est proportionnelle à la valeur de l'activation musculaire : plus l'activation est grande, plus la force musculaire développée est importante :

¹ Contraction isométrique : i.e. à longueur constante. On mesure le développement de la tension engendrée par l'excitation musculaire.

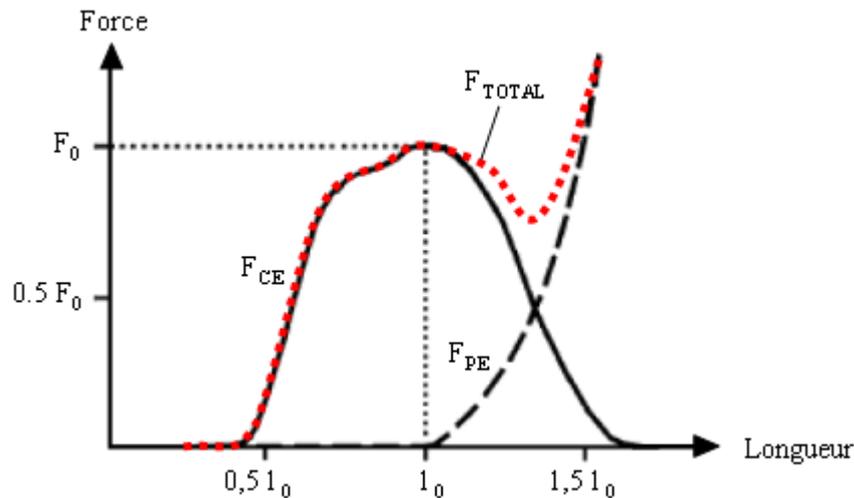


Figure I.9 : La relation force/longueur pour un muscle

Avec F_{CE} : la force active et F_{PE} : la force passive

- La relation force/vitesse est définie pour la contraction musculaire isotonique². Dans ce cas de figure, l'élément série ne subit pas de variation de force et il ne participe pas aux variations de longueur du muscle. La participation de l'élément parallèle est éliminée en se plaçant à une longueur inférieure ou égale à la longueur l_0 . Lorsqu'une force de tension est appliquée au muscle, celui-ci se contracte et s'immobilise. La dépendance de la force du muscle par rapport à sa vitesse de contraction est illustrée par la courbe force/vitesse de la Figure I.10 [Hill 38] [Zajac 89] [Winter 90] [Chèze 05] : on obtient une hyperbole. Cette courbe montre que si le muscle se contracte rapidement l'effort qu'il produit est plus faible.

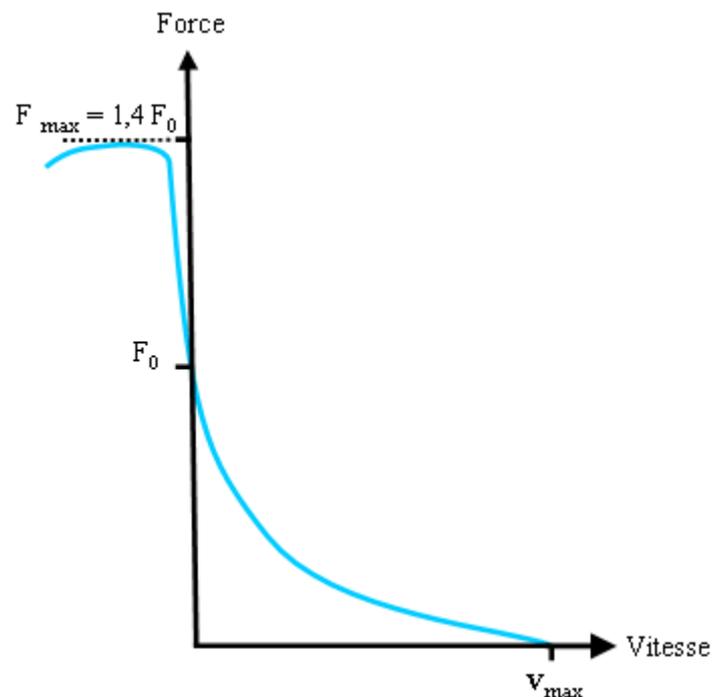


Figure I.10 : La relation force/vitesse pour un muscle

$F_0 = F_{\max}$: force maximale produite à la longueur optimale

² Contraction isotonique : i.e. à charge constante (force ou couple). On mesure les variations de longueur provoquées par l'excitation musculaire.

Ces deux relations permettent de déduire la puissance musculaire par multiplication des deux courbes (Cf. Figure I.11) [Hill 38] [Zajac 89] [Winter 90] [Chèze 05] :

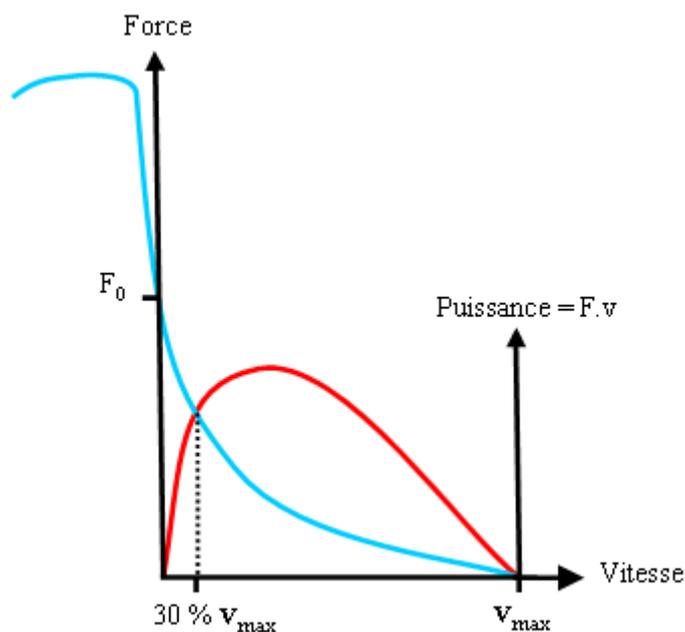


Figure I.11 : Puissance développée par un muscle actif

En général le maximum de la puissance motrice est obtenu autour de 30% de la vitesse maximale de raccourcissement.

Il est difficile d'évaluer la force développée par un muscle au cours d'un mouvement quelconque car de nombreux paramètres interviennent. Lorsque l'activation est maximale, la force exercée se trouve à l'intersection des courbes Force/Longueur et Force/ Vitesse (Cf. Figure I.12) [Chèze 05] :

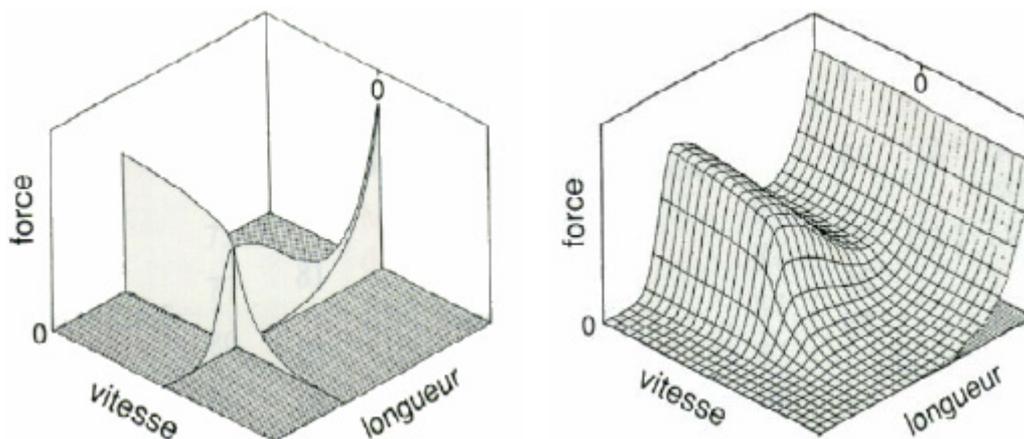


Figure I.12 : Force exercée quand l'activation musculaire est maximale

Nous venons de détailler le modèle de Hill ainsi que les résultats qu'il a obtenus. Bien que ce soit le modèle le plus répandu, d'autres modèles ont été développés par la suite comme par exemple ceux de Huxley [Huxley 57], Zajac [Zajac 89], Winter [Winter 90], Fung [Fung 93] ou encore de Stroeve [Stroeve 97].

Après la description du réseau musculaire et de quelques lois d'actionnement pour générer les mouvements, nous allons nous intéresser à la transmission du mouvement, et plus particulièrement au niveau des doigts de la main.

3.3 La mécanique du mouvement

Comme nous l'avons vu précédemment, les forces développées par la main sont les forces engendrées par la contraction des muscles intrinsèques et extrinsèques. Elles entraînent les mouvements de flexion, d'extension, d'adduction ou d'abduction.

Dans le cas particulier de la flexion (ou de l'extension), le (s) mouvement est (sont) le produit de deux forces :

- les forces actives dues aux contractions musculaires,
- les forces passives dues à la résistance visco-élastiques des parties molles qui sont les tendons et les muscles. Ces derniers freinent ou arrêtent le mouvement opposé.

Dans ce paragraphe, nous allons expliquer la mécanique du mouvement qui résulte de l'action des muscles, de celle des ligaments, mais aussi de la forme des os des phalanges – surtout au niveau des articulations. Nous donnerons l'exemple de la flexion en détaillant l'appareil long fléchisseur qui est le muscle le plus actif dans ce mouvement. La Figure I.13 met en évidence l'existence de différentes formes de fibres (annulaires et cruciformes) pour les tendons fléchisseurs des doigts ainsi que leur mouvement coulissant [Tubiana 80] :

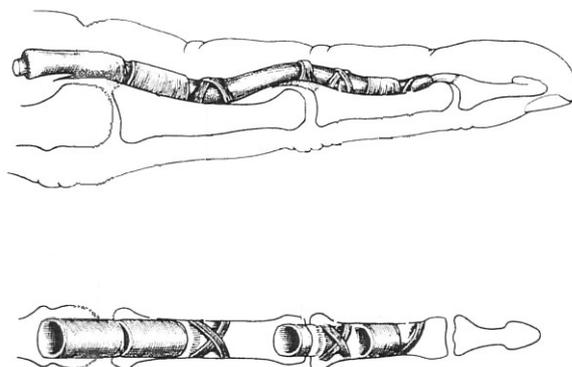


Figure I.13 : Les tendons fléchisseurs des doigts

Les muscles longs fléchisseurs développent une force qui est d'environ 3,5 kg au cm². Ils communiquent cette force à l'appareil tendineux qui la transforme au niveau du doigt en amplitude de flexion articulaire avec ou sans résistance par l'intermédiaire d'une mécanique spéciale : les poulies.

Dans le cas d'un muscle long fléchisseur, il existe trois poulies pour appliquer les efforts sur la face palmaire du squelette de la main :

- le ligament annulaire du carpe en face du poignet,

- la poulie proximale en regard de l'articulation au niveau de la première phalange (MCP),
- la poulie distale en regard de la deuxième phalange.

La Figure I.14 [Tubiana 80] présente les morphologies des poulies digitales, i.e. proximale et distale. Nous remarquons que deux aspects différents de la poulie proximale peuvent exister dans une même main : celle-ci peut être homogène ou alors divisée en deux parties bien distinctes :

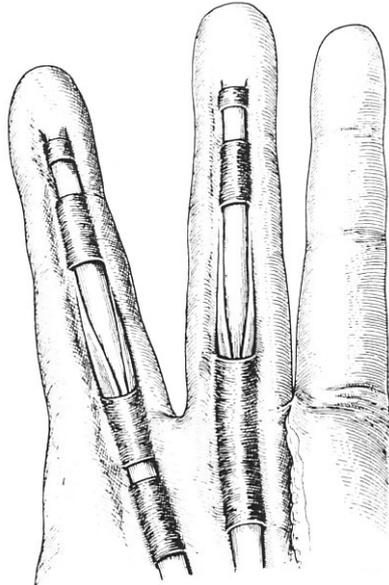


Figure I.14 : Les poulies proximales et distales des doigts

Les longueurs des poulies dépendent des longueurs des doigts : elles sont plus longues sur le majeur et plus courtes sur l'index et l'auriculaire. La distance entre les poulies proximale et distale est proportionnelle à la longueur du doigt considéré.

Les mouvements accomplis par la poulie proximale et la poulie distale suivent respectivement ceux de la première et de la deuxième phalanges.

L'énergie transformée se fait par l'intermédiaire d'un système polyarticulé. Les muscles fléchisseurs sont donc des muscles polyarticulés dont la mécanique est spéciale. D'où le rôle essentiel des différentes poulies qui appliquent et dirigent l'appareil tendineux sur cette chaîne polyarticulée mais sans perdre la moindre énergie grâce à un système avantageux de leviers et de gaines synoviales qui diminuent le frottement.

Les leviers squelettiques peuvent être maintenus en position (fonction statique) ou mis en mouvement (fonction dynamique) grâce aux muscles. Une partie de la complexité vient du fait que ces fonctions nécessitent toujours des actions en opposition (muscles agonistes et antagonistes). Le contrôle de ces fonctions suppose donc une connaissance parfaite de l'état des muscles selon deux paramètres :

- leur longueur permettant de déduire la position du levier squelettique,
- leur vitesse de contraction permettant de connaître les caractéristiques mécaniques du déplacement (vitesse, accélération).

Après avoir décrit la transmission des efforts générés par les muscles aux articulations des doigts par le biais des poulies, nous détaillons les nombreuses contraintes articulaires du squelette de la main dans le paragraphe suivant.

4. La biomécanique de la main

Les mouvements des doigts ne s'effectuent pas indépendamment les uns des autres en raison du réseau complexe de muscles et de tendons. Il existe de nombreuses contraintes articulaires entre les articulations de la main [Lansdmeer 55] [Vardy 98] [Ouhaddi 98]. Il faut tenir compte de ces contraintes pour obtenir un modèle réaliste de la main.

Ces contraintes sont réparties en deux catégories :

- les contraintes statiques qui correspondent aux butées articulaires,
- les contraintes dynamiques qui modélisent les différents couplages articulaires liés à la transmission du mouvement par les muscles.

4.1 Les contraintes statiques

Elles traduisent les limites des angles des articulations en flexion/extension et en adduction/abduction. La Figure I.15 présente les repères locaux d'un doigt ; dans notre cas de figure, l'index.

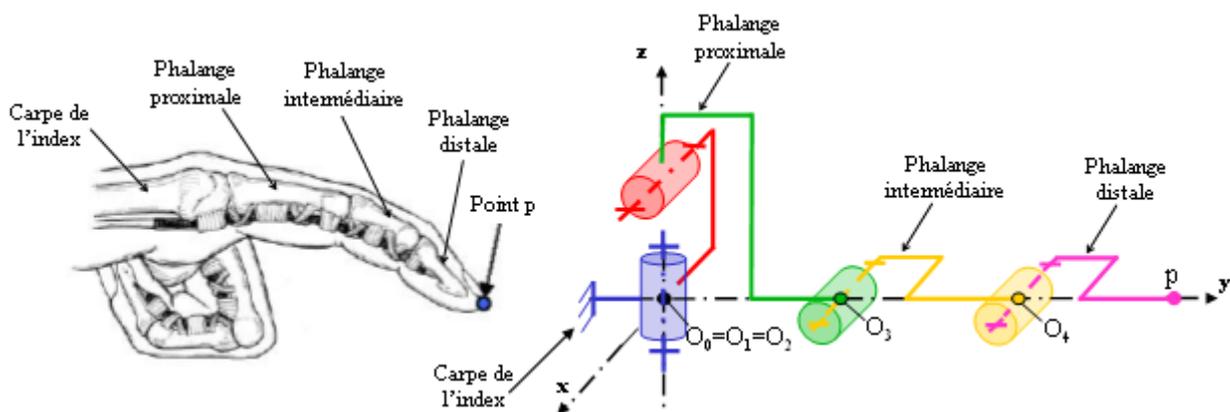


Figure I.15 : La modélisation de l'index

Le Tableau I.1 ci-dessous récapitule les débattements articulaires de chaque articulation :

	CMC f/e	CMC a/a	MCP (PP) f/e	MCP (PP) a/a	IPP (MP) f/e	IPD (DP) f/e
Auriculaire	$[-2^\circ; 2^\circ]$	$[0^\circ; 5^\circ-10^\circ]$	$[0^\circ; 90^\circ]$	$[-15^\circ; 15^\circ]$	$[0^\circ; 110^\circ-130^\circ]$	$[0^\circ; 65^\circ-90^\circ]$
Annulaire	$[-1^\circ; 1^\circ]$	$[0^\circ; 2^\circ]$	$[0^\circ; 90^\circ]$	$[-15^\circ; 15^\circ]$	$[0^\circ; 110^\circ-130^\circ]$	$[0^\circ; 65^\circ-90^\circ]$
Majeur	0°	0°	$[0^\circ; 90^\circ]$	0° (axe)	$[0^\circ; 110^\circ-130^\circ]$	$[0^\circ; 65^\circ-90^\circ]$
Index	0°	0°	$[0^\circ; 90^\circ]$	$[-30^\circ; 30^\circ]$	$[0^\circ; 110^\circ-130^\circ]$	$[0^\circ; 65^\circ-90^\circ]$
Pouce	$[-25^\circ; 120^\circ]$	$[0^\circ; 35^\circ-40^\circ]$	$[10^\circ; 90^\circ]$	0°	-	$[0^\circ; 65^\circ-90^\circ]$

f/e : flexion / extension
a/a : adduction/abduction

Tableau I.1 : Les limites articulaires

Un exemple de butées articulaires est donné dans le cas de l'index sur la Figure I.16 :

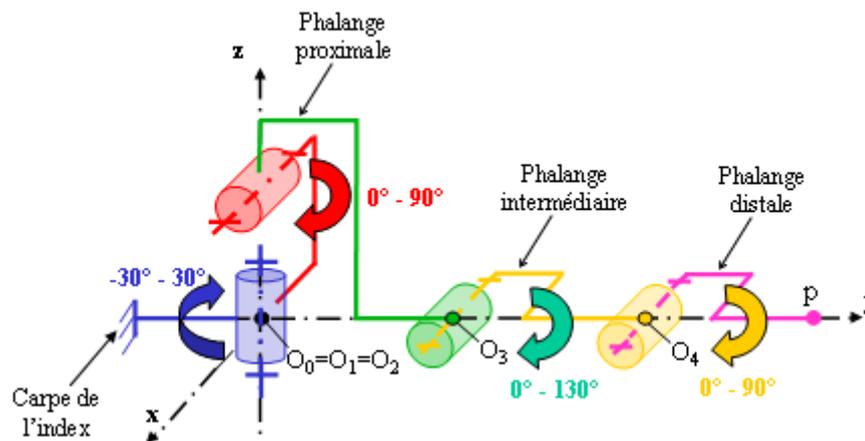


Figure I.16 : Les butées articulaires pour l'index

Nous venons de présenter les limites articulaires des doigts de la main, i.e. en statique. Mais lorsque la main est en mouvement, des contraintes supplémentaires apparaissent : elles sont appelées « contraintes dynamiques » et elles font l'objet du paragraphe suivant.

4.2 Les contraintes dynamiques

Ces contraintes représentent les interdépendances existant entre les degrés de liberté d'un même doigt ou entre plusieurs doigts [Kuch 95] [Ouhaddi 98]. Ces relations entre les angles aux articulations traduisent – en simplifiant un peu – les actions des muscles et des tendons [Rijpkema 91]. Dans le cas de l'animation de la main, i.e. sans action extérieure exercée sur la main, ces relations permettent d'obtenir rapidement une simulation réaliste de la main d'un point de vue visuel.

4.2.1 Les contraintes internes au doigt

Lorsqu'un doigt se plie ou se tend – i.e. un mouvement de flexion/extension - au niveau de l'articulation de la première phalange (MCP), cela entraîne une rotation dans l'articulation correspondante (CMC) au niveau du poignet. Ceci est dû au fait que les muscles agissant sur ces deux articulations sont identiques et naturellement couplés. Une relation linéaire entre ces deux rotations a été démontrée à la suite de nombreuses études [Lansmeer 55] [Rijpkema 91]. La contrainte à appliquer pour reproduire ce couplage est donnée par :

$$\theta_{\text{MCP}}(f/e) = 2 \times \theta_{\text{CMC}}(f/e) \quad (\text{Eq. I-1})$$

De même, entre les articulations des première et deuxième phalanges (respectivement MCP et IPP), on a la relation suivante :

$$\theta_{\text{IPP}}(f/e) = 2 \times \theta_{\text{MCP}}(f/e) \quad (\text{Eq. I-2})$$

Une contrainte semblable existe entre les articulations des deuxième et troisième phalanges (respectivement IPP et IPD) :

$$\theta_{\text{IPD}}(f/e) = \frac{2}{3} \times \theta_{\text{IPP}}(f/e) \quad (\text{Eq. I-3})$$

Pour le pouce, une relation entre les articulations des deuxième et troisième phalanges (respectivement IPP et IPD) est donnée par :

$$\theta_{IPD_pouce}(f/e) = \frac{7}{5} \times \theta_{IPP_pouce}(f/e) \quad (\text{Eq. I-4})$$

En appliquant ces trois contraintes, un pliage « naturel » du doigt est obtenu. Et surtout, ces relations permettent de réduire le nombre de degrés de liberté de la main.

Il existe également une relation au niveau de l'articulation de la première phalange (MCP) entre les mouvements de flexion/extension et d'abduction/adduction. Ainsi, si cette articulation est très fléchie, alors aucun mouvement d'adduction/abduction n'est possible et inversement. Des relations entre les limites angulaires des deux rotations peuvent être établies dans cette articulation. Dans le cas où l'angle de flexion/extension n'est pas nul, la valeur maximale de l'angle d'abduction/adduction de la première phalange est donnée par la relation suivante :

$$|\theta_{MCP}(a/a)| \leq \left(1 - \frac{1}{\theta_{MCP}(f/e)_{\max}} \right) \theta_{MCP}(f/e) \times \theta_{MCP}(a/a) \quad (\text{Eq. I-5})$$

avec $\theta_{MCP}(f/e)_{\max}$: la valeur maximale de l'angle de flexion/extension de la phalange.

Après les relations angulaires entre les phalanges d'un même doigt au cours du mouvement, nous détaillons les contraintes entre des doigts voisins.

4.2.2 Les contraintes entre les doigts

Dans certaines configurations, il est également impossible de mouvoir les doigts indépendamment les uns des autres [Lansmeer 55] [Rijkema 91]. Afin d'obtenir en modélisation le pliage simultané de deux doigts voisins, une distance angulaire minimale est conservée entre les articulations des premières phalanges au cours du mouvement. Par exemple, la flexion de la première phalange (MCP) de l'index ou celle de l'annulaire entraîne la flexion de la première phalange du majeur :

$$\theta_{MCP}(f/e)_{\text{majeur}} \geq \sup \left(\theta_{MCP}(f/e)_{\text{index}} - 25, \theta_{MCP}(f/e)_{\text{annulaire}} - 45, \theta_{MCP}(f/e)_{\text{majeur}}^{\min} \right) \quad (\text{Eq. I-6})$$

avec $\theta_{MCP}(f/e)_{\text{majeur}}^{\min}$: la valeur minimale de l'angle de flexion/extension de la première phalange du majeur.

Toutes les relations précédemment décrites s'appliquent aux quatre doigts de la main. En raison de sa morphologie particulière, le pouce est soumis à des contraintes différentes. Ainsi, il existe des contraintes qui relient les mouvements de flexion/extension et d'abduction/adduction des deux premières articulations du pouce :

$$\theta_{MCP}(f/e) = 2 \left(\theta_{CMC}(f/e) - \frac{\pi}{6} \right) \quad (\text{Eq. I-7})$$

$$\theta_{MCP}(a/a) = \frac{7}{5} \theta_{CMC}(a/a) \quad (\text{Eq. I-8})$$

Toutes ces relations nous permettent de simplifier le modèle de la main, tant au niveau du nombre de degrés de liberté que dans sa représentation dans l'espace des configurations. Ainsi, les contraintes décrites dans ce paragraphe permettent de simuler de manière réaliste les mouvements de la main humaine. Non seulement elles améliorent le rendu visuel, mais elles réduisent également le nombre de degrés de liberté à modéliser. Elles ne sont donc pas à négliger dans le modèle que nous développons, et cela est d'autant plus vrai que notre modèle est animé.

Après la description des squelettes osseux et fibreux de la main ainsi que celle des contraintes biomécaniques, nous détaillons dans le paragraphe suivant les propriétés physiologiques de la main qui nous permettent de toucher, saisir et manipuler des objets dans notre environnement.

5. La physiologie de la main

Nous étudions maintenant les caractéristiques de la peau humaine au niveau de la main. Sur le plan anatomique comme sur le plan fonctionnel, des différences entre la peau dorsale et la peau palmaire de la main existent. Dans le cadre des applications de saisie et de manipulation d'objets en Réalité Virtuelle, nous nous intéresserons plus particulièrement à la peau et aux coussinets palmaires - i.e. les coussinets phalangiens et la pulpe - des doigts puisqu'ils sont en contact avec les objets.

5.1 La structure de la peau palmaire

La peau palmaire est épaisse et résistante ce qui lui permet de supporter d'importantes contraintes de pression. Elle est constituée essentiellement de trois couches (Cf. Figure I.17)

[Tubiana 80] :

- La couche cornée : elle se compose d'une épaisse couche de cellules. C'est un réseau filamenteux favorable à l'hydratation ;
- l'épiderme : il comporte quatre couches. De la profondeur vers la surface, on a : la couche basale, le stratum Malpighi, la couche granuleuse et le stratum lucidum. C'est à ce niveau que s'effectue la régénération cutanée (multiplication cellulaire) ;
- et le derme : il comporte également quatre couches. De la surface vers l'épiderme, on a : une couche superficielle, un derme moyen, un derme profond et un plexus vasculaire superficiel. C'est un tissu conjonctif constitué par une architecture de fibres collagènes, de fibres élastiques et de fibres réticulaires. On y trouve surtout des terminaisons nerveuses et des vaisseaux.

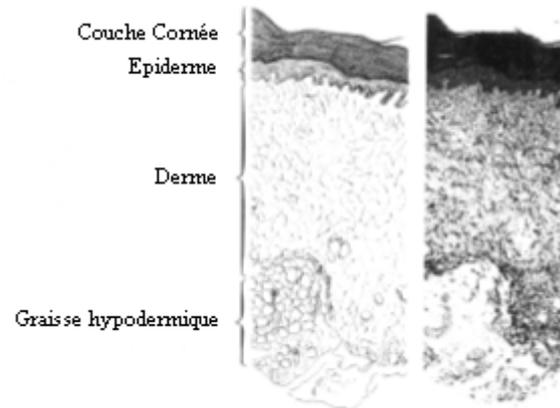


Figure I.17 : Coupe de peau palmaire

La paume possède de nombreuses propriétés comme une résistance mécanique importante en raison de sa structure histologique (i.e. la structure des tissus composés de cellules) et de son épaisseur. De plus, elle participe au sens tactile par la présence de ses nombreux récepteurs.

5.2 Les récepteurs de la peau

La peau se présente comme une matrice de capteurs placés verticalement et horizontalement. On obtient donc une résolution horizontale fine et une disposition de récepteurs en profondeur qui confère une continuité à la perception. Il existe quatre structures de récepteurs cutanés (Cf. Figure I.18) [Tubiana 80], qui sont :

- au niveau de l'épiderme : les terminaisons nerveuses libres et les complexes cellulaires de Merkel autrement appelés corpuscules de Merkel.
- au niveau du derme : les corpuscules de Meissner, les corpuscules de Pacini, les corpuscules de Ruffini et les corpuscules de Golgi-Mazzoni.

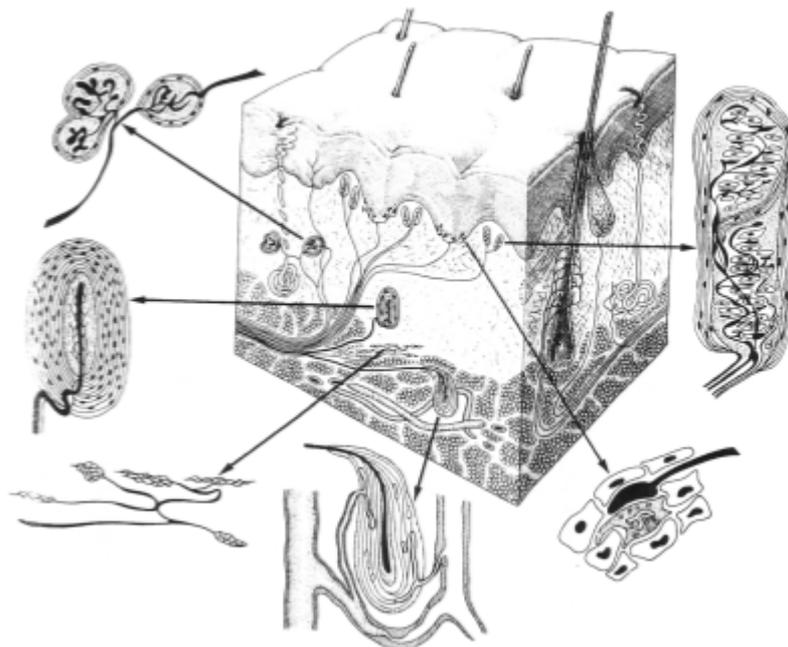


Figure I.18 : Reconstruction d'un fragment de peau avec les récepteurs

Du haut à gauche vers le haut à droite : corpuscules de Golgi-Mazzoni, corpuscules de Vater-Pacini, corpuscules de Ruffini, corpuscules de Pacini paravasculaires, corpuscules de Merkel, corpuscules de Meissner

Ceux qui nous intéressent plus particulièrement sont les mécanorécepteurs. Ce sont les récepteurs sollicités lorsqu'un contact est établi entre la peau et un objet (ou une matière). La distribution de ces mécanorécepteurs et la combinaison de leurs signaux en réponse aux stimuli induisent leur spécialisation d'un point de vue sensoriel. Il existe deux catégories de mécanorécepteurs (Cf. Figure I.18) [Tubiana 80] [Pawluk 97] :

- les mécanorécepteurs proprioceptifs : ils informent sur la position des membres supérieurs ainsi que sur l'intensité des forces de pression appliquées.
- les mécanorécepteurs extéroceptifs : ils informent sur le type d'interaction qu'a la peau avec le monde extérieur (variations thermiques, texture, pression, réactions chimiques, etc). Les variations mécaniques sont transformées en des influx nerveux par ces mécanorécepteurs.

Quatre types de mécanorécepteurs extéroceptifs ont été identifiés :

- les terminaisons de Meissner qui se situent au niveau des empreintes de la peau de la main entre les arêtes papillaires (i.e. les reliefs des empreintes) et le derme,
- les terminaisons de Merkel, qui se situent plutôt à la fin de ces arêtes,
- les corpuscules de Pacini,
- les terminaisons de Ruffini.

Des informations peuvent également être mesurées et transmises par la peau palmaire. En effet, elle possède de nombreuses propriétés mécaniques et physiologiques comme les coussinets, qui permettent une bonne adhérence lors de la réalisation des tâches manuelles.

5.3 Le système des coussinets palmaires

La peau palmaire est fixée aux membranes fibreuses enveloppant les muscles et les tendons sous-jacents par un faisceau de fibres. Cette liaison se fait par accolement direct. Ainsi se forme un système de coussinets adipeux déformables mais fixes par rapport à la peau et également par rapport à la membrane fibreuse sous-jacente. La paume comporte quatre coussinets principaux :

- le coussinet thénarien, situé au niveau du pouce et qui est proche du centre de la paume,
- le coussinet hypothénarien, situé au niveau de l'auriculaire et qui déborde sur l'extérieur de la main,
- le coussinet métacarpo-phalangien, situé devant la base des doigts transversalement. Il se situe entre les plis palmaires des différents doigts et le pli de flexion au niveau de la paume.

Les doigts possèdent également deux types de coussinets :

- les coussinets phalangiens, situés entre les plis de flexion proximal et intermédiaire pour la première phalange et les plis de flexion intermédiaire et distal pour la deuxième phalange,
- et, la pulpe du doigt, située à l'extrémité du doigt. Cette structure alvéolaire constituée d'un tissu très spécialisé s'attache d'une part sur le périoste de la phalangette et d'autre part à la face profonde du derme. Elle est épaisse, glabre et riche en glandes sudoripares. Les propriétés physiques et la richesse de ses terminaisons nerveuses font apparaître la pulpe des doigts comme un organe adapté à la perception active de l'environnement et à la préhension fine des objets.

Les coussinets et la pulpe sont illustrés par la Figure I.19 et la Figure I.20 [Tubiana 80] [Tubiana 90] :

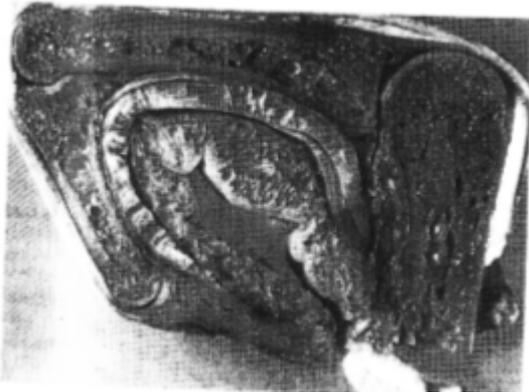


Figure I.19 : Coussinets adipeux palmaires

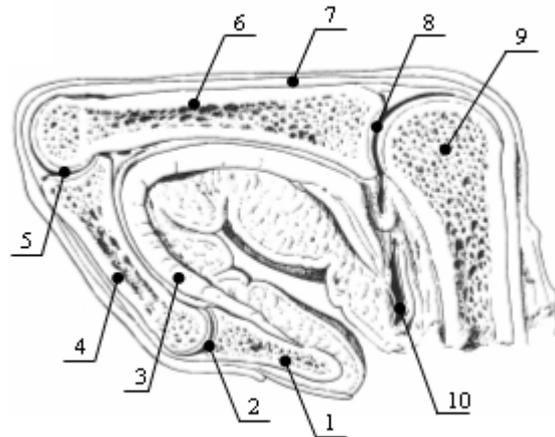


Figure I.20 : Coupe d'un doigt

1–Phalange distale ; 2–Articulation interphalangienne distale (IPD) ; 3–Muscle fléchisseur profond ; 4–Phalange intermédiaire ; 5–Articulation interphalangienne proximale (IPP) ; 6–Phalange proximale ; 7–Muscle extenseur ; 8–Capsule articulaire (MCP) ; 9–Tête de métacarpien ; 10–Récessus palmaire

D'un point de vue fonctionnel, ces coussinets constituent des structures essentielles à la préhension. Non seulement ils possèdent une bonne résistance et un amortissement aux pressions exercées mais, ils permettent de modeler la face palmaire sur l'objet à saisir de par leur caractère déformable. Ainsi, ils augmentent la surface de contact et ils améliorent la capacité de rétention. De plus, la fixité de la peau par rapport au squelette permet une transmission directe des efforts de serrage sur l'objet à saisir : c'est un élément fondamental de la stabilité des prises.

Lorsqu'il existe une possibilité de glissement de l'objet sur la peau, une action de résistance apparaît : le frottement. Le coefficient de frottement est proportionnel à l'étendue des surfaces en contact et il est variable suivant les zones cutanées. En effet, il est plus important au niveau des coussinets palmaires et il l'est davantage au niveau des pulpes des doigts (Cf. Figure I.21) [Tubiana 80] :

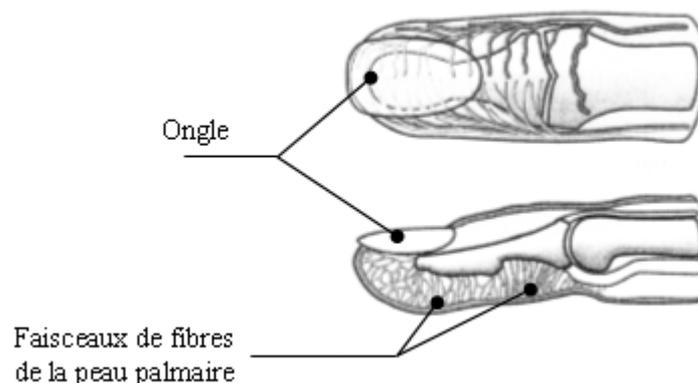


Figure I.21 : Les éléments qui empêchent le glissement lors des prises

Les paragraphes précédents ont présenté les principales caractéristiques et propriétés de la main humaine. Différentes fonctions ont été identifiées et détaillées comme le squelette, le réseau musculaire, les contraintes biomécaniques ou encore les propriétés physiologiques.

Afin d'obtenir une simulation réaliste d'un point de vue physique, il est nécessaire de respecter certaines spécificités de la main humaine.

6. Conclusion

Dans le cadre de ces travaux de recherche, l'objectif est de développer un modèle de main afin de simuler les interactions de l'opérateur avec son environnement de travail. Comme nous venons de le voir, la main est un organe complexe et donc difficile à modéliser dans son ensemble. Nous allons donc identifier les caractéristiques nécessaires à la réalisation des tâches manuelles de manière stable en environnement virtuel. Ainsi, à chaque couche décrite précédemment, nous établirons sa pertinence ainsi que les propriétés importantes à simuler dans notre cas de figure.

6.1 Le squelette de la main

Il définit la cinématique de la main humaine et forme donc la base de notre main virtuelle. Afin d'avoir une grande mobilité dans l'environnement virtuel, nous tiendrons compte de tous les degrés de liberté de la main humaine. Nous modéliserons également les doigts par cinq chaînes polyarticulées rigides afin de simuler les différentes prises et de conserver un réalisme visuel pour les utilisateurs.

6.2 Le réseau musculaire de la main

Comme dans notre cas de figure nous sommes dans le contexte temps réel, nous avons choisi de piloter la main par la capture de mouvements. De plus, les interfaces disponibles pour que nous interagissions avec l'environnement virtuel ne nous permettent pas de mesurer les efforts exercés sur les doigts. En fait, nous n'avons qu'une information de position (coordonnées cartésiennes) pour les doigts de l'utilisateur. Or ces données ne sont pas suffisantes pour identifier les forces générées par les muscles lorsqu'ils effectuent les mouvements des doigts. Par conséquent, nous n'avons pas retenu l'actionnement par commande musculaire et donc nous n'avons pas intégré de modèle rhéologique.

6.3 La biomécanique de la main

La biomécanique de la main est définie par les butées articulaires (contraintes statiques) et par les couplages entre les phalanges et entre les doigts (contraintes dynamiques).

Dans notre cas de figure, nous prenons en compte les butées articulaires car elles agissent comme des contraintes supplémentaires pour notre modèle. En effet, la chaîne cinématique étant redondante, il existe plusieurs configurations possibles pour une prise donnée. Les limites articulaires permettent donc de réduire l'espace des solutions. De plus, ces limites se révèlent être utiles pour palier aux perturbations ou occultations du système de capture de mouvements utilisé pour suivre les mains de l'utilisateur. En effet, dans le cas où les informations de positions ne seraient plus reçues, ces butées empêchent les mains virtuelles de se trouver dans des positions invraisemblables. Elles se révèlent également utiles pour adapter les mains virtuelles aux différentes morphologies des utilisateurs.

Quant aux couplages articulaires, nous n'en avons pas tenu compte. Les angles mesurés avec les gants de données correspondent aux positions angulaires des doigts de l'utilisateur et par conséquent ils respectent automatiquement et naturellement ces couplages.

6.4 La physiologie de la main

La main est le seul organe de préhension ainsi que l'organe récepteur principal du toucher chez l'Homme. Elle possède des propriétés mécaniques comme la souplesse, l'élasticité ou encore la résistance mécanique, qui se révèlent être des qualités indispensables pour le contact lors des prises. En effet, les parties molles des doigts – i.e. la pulpe et les coussinets - permettent d'augmenter l'adhérence lors de la prise par leur déformation. De plus, la présence de nombreux récepteurs dans les tissus lui confère une très grande sensibilité. Ces propriétés physiologiques permettent d'obtenir de nombreuses informations sur l'environnement avec lequel la main interagit.

Dans notre cas de figure, il est essentiel que notre modèle se comporte de la même manière que la main humaine lorsqu'il est en contact avec son environnement : il devra se déformer sous l'action des efforts exercés sur la peau. De plus, il est nécessaire d'assurer des prises stables vu que nous devons manipuler des objets virtuels. Pour ce faire, nous intégrons les caractéristiques mécaniques et physiologiques des coussinets et des pulpes des doigts dans notre main virtuelle.

Ni l'innervation, ni la vascularisation, ni le système lymphatique ne seront modélisés car ils ne répondent pas à notre besoin.

6.5 Le modèle retenu

Afin de simuler les interactions Homme-machine en environnement virtuel, notre modèle de main doit respecter les contraintes principales suivantes :

- respect de la cinématique du squelette et des degrés de liberté pour reproduire les nombreux mouvements de la main humaine ;
- actionnement de la main virtuelle en temps réel par le biais de la capture de mouvements ;
- simulation des déformations de la peau palmaire suite aux efforts de contact.

Comme ces contraintes correspondent à trois fonctions distinctes de la main humaine, nous avons choisi de développer un modèle multicouches pour notre main virtuelle. La Figure I.22 [Kapandji 05] présente les caractéristiques de la main humaine que nous avons identifiées comme essentielles pour simuler des tâches de préhension.

Les trois fonctions retenues correspondent aux principales étapes de la thèse. A chaque étape, nous intégrerons les caractéristiques mécaniques et/ou physiologiques décrites dans les paragraphes précédents afin d'obtenir une modélisation réaliste de la main humaine.

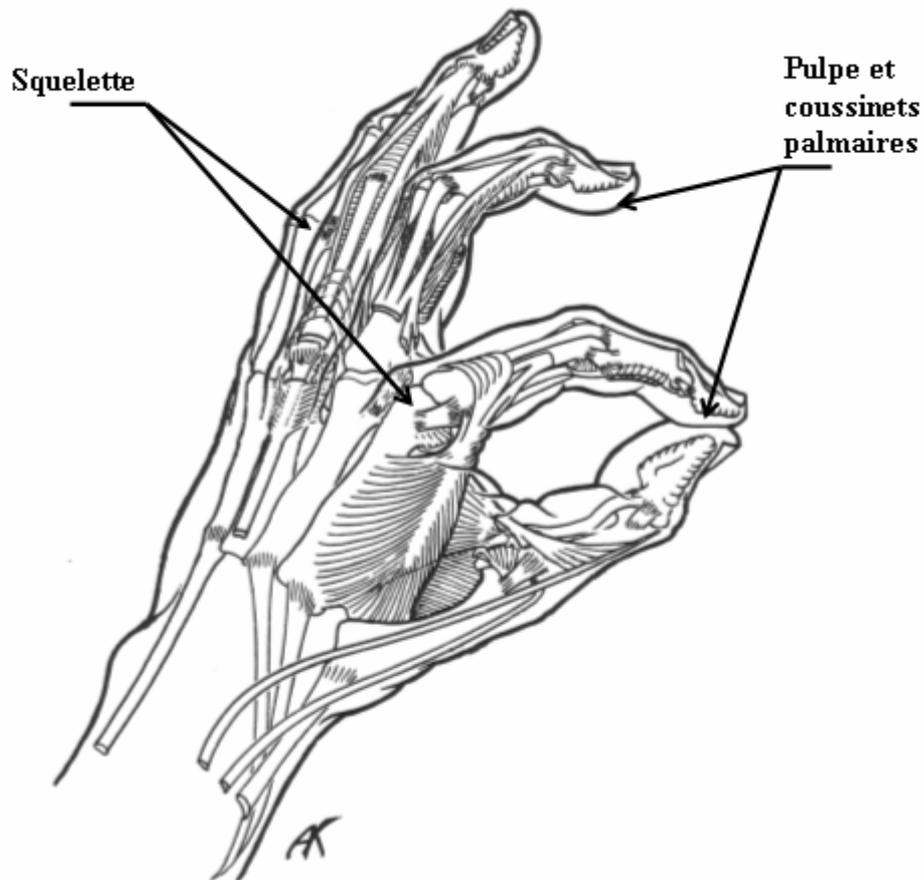


Figure I.22 : Les couches étudiées pour une modélisation réaliste

Ainsi, nous nous intéressons aux modèles de mains existants pour définir plus précisément notre main virtuelle. Afin de respecter les contraintes décrites précédemment et dans le but de simuler des tâches de préhension, nous avons focalisé nos recherches bibliographiques sur la cinématique ainsi que sur les propriétés mécaniques des coussinets et de la pulpe de la main humaine. Le Chapitre II présente un état de l'art des modèles de mains utilisés pour l'animation graphique et pour la simulation des interactions Homme-machine.

CHAPITRE II : Présentation des modèles de main existants

La main humaine est au cœur de nombreuses études dans des domaines variés comme la médecine (en anatomie, en physiologie ou en biomécanique), la mécanique et la robotique, ou encore la simulation et l'animation graphique. En effet, de nombreux chercheurs souhaitent connaître la complexité de sa structure et comprendre la genèse des multiples mouvements réalisables.

1 Introduction

Dans les paragraphes qui suivent, nous présentons les modèles de mains existants. Comme leurs caractéristiques dépendent fortement des applications, nous classifions les modèles en fonction de leur utilisation.

Tout d'abord, nous présenterons les modèles de mains utilisés pour la communication. Ils sont utilisés essentiellement pour la reconnaissance des gestes par vision par ordinateur ou encore pour le suivi des mouvements des mains dans l'espace avec ou sans système spécifique de capture de mouvements.

Puis, nous exposerons les mains développées pour manipuler des objets. Ces modèles sont ceux qui nous intéressent le plus puisque notre objectif est de simuler les interactions Homme-machine en environnement virtuel.

Nous décrirons également les techniques développées pour animer et déformer les modèles de mains au cours des mouvements et des manipulations.

Puis, nous présenterons les résultats des recherches menées sur les caractéristiques mécaniques des parties déformables de la main humaine – coussinets et pulpes des doigts - et plus particulièrement par leur modélisation.

Enfin, nous concluons par la description de notre modèle de main.

2 Les modèles de mains pour la simulation et l'animation

L'étude des mains a conduit à définir deux catégories de mouvements : les gestes et les mouvements involontaires. Dans notre cas de figure, nous nous focalisons sur les gestes car ils possèdent deux modalités principales [Pavlovic 97] :

- la communication en mettant en œuvre des actions spécifiques comme l'action de pointer un objet pour l'identifier ou encore des mouvements prédéfinis pour aider à naviguer dans un environnement, ainsi que des symboles comme ceux utilisés pour la langue des signes ;
- la manipulation grâce à la fonction de préhension de la main (Cf. Chapitre I) qui permet d'interagir naturellement avec des objets virtuels comme par exemple dans le cas de la téléopération ou celui de l'assemblage en environnement virtuel.

Nous décrivons les différents modèles dans la suite de ce paragraphe.

2.1 Les modèles développés pour communiquer

Les gestes sont un moyen naturel de communication entre personnes car ils permettent d'illustrer une conversation entre deux ou plusieurs individus. La langue des signes est d'ailleurs l'exemple le plus significatif. En effet, c'est un langage structuré qui est basé sur des positions définies des mains et des doigts. Ces dernières sont d'ailleurs souvent utilisées pour la validation d'un modèle de main et/ou d'une technique de reconnaissance de gestes. De nos jours, avec le développement des nouvelles technologies, les gestes peuvent également aider à la communication Homme-machine comme par exemple dans le cadre des applications en Réalité Virtuelle ou en Réalité augmentée.

Dans ce paragraphe, nous présentons les modèles qui nous aident à communiquer avec autrui ou avec une machine. Il existe deux cas de figure :

- soit les modèles sont couplés à des marqueurs ou des gants de données pour la capture de mouvements,
- soit ils sont utilisés avec des caméras et une méthode de reconnaissance de gestes pour la vision artificielle.

Ces applications requièrent différents types de modèles de main comme les modèles basés sur l'apparence ou encore les modèles génériques.

Les modèles basés sur l'apparence sont des modèles 2D constitués d'images ou de contours. Ce sont donc des apparences possibles de la main avec différents points de vue et sous différentes conditions [Pavlovic 97]. Nous ne développerons pas ces modèles car ils ne répondent pas à notre besoin.

Quant aux modèles génériques, ils peuvent être classifiés en trois catégories : les modèles squelettiques, les modèles surfaciques et les modèles volumiques. Ils consistent d'une part à modéliser la cinématique afin que le modèle de main se comporte de la même manière que la main humaine. D'autre part, ils enveloppent la main par des surfaces ou des volumes afin d'en modifier sa forme au cours des différents mouvements.

Dans les paragraphes qui suivent, nous présentons les différents modèles génériques de la main également appelés « modèles géométriques ». Ils s'attachent tout particulièrement à une modélisation la plus réaliste possible d'un point de vue visuel. Nous commencerons par les modèles cinématiques. Puis, nous décrirons les différentes enveloppes possibles pour les objets 3D, i.e. les modèles surfaciques, et nous généraliserons aux modèles volumiques. Nous détaillerons également les méthodes de déformation développées pour animer ces modèles.

2.1.1 Les modèles squelettiques

Dans ce paragraphe, nous décrivons les modèles squelettiques de la main. Ils servent à décrire la cinématique de la main et à animer le modèle tout en respectant l'anatomie et la biomécanique de la main humaine.

Rehg et Kanade [Rehg 93] ont utilisé un modèle cinématique de la main pour leur système de reconnaissance de gestes nommé *DigitEyes*. La main est modélisée par 16 cylindres (Cf. Figure II.23) : 3 pour les phalanges des cinq doigts et un cylindre pour la paume. Le modèle possède 27 degrés de liberté (DDL) avec la cinématique suivante :

- 4 DDL pour chacun des 4 doigts,
- 5 DDL pour le pouce,
- et 6 DDL pour la paume de la main.

En utilisant le paramétrage de Denavit-Hartenberg et en intégrant les contraintes biomécaniques, ils reproduisent les postures de la main par la cinématique inverse pour la communication Homme-machine.

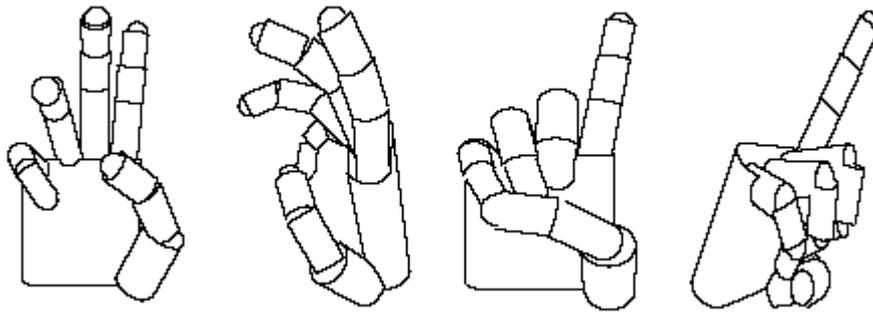


Figure II.23 : Le modèle cinématique utilisé par le système DigitEyes

Nirei et al. [Nirei 96] ont représenté la main par un modèle 3D pour l'étude des mouvements chez les sportifs et l'animation graphique à l'aide de deux caméras optiques. Chaque primitive correspond à un ellipsoïde tronqué. Ce modèle est constitué de 21 primitives et de 20 articulations avec (Figure II.24) :

- 3 segments pour le pouce
- 4 segments par doigt
- 1 pour le poignet (ou l'avant-bras).

Ils ont mis en œuvre des techniques d'optimisation – un algorithme génétique et un recuit simulé - pour faire correspondre le modèle 3D de la main aux images vidéos de la main humaine. Ces postures clés sont utilisées pour l'étude des mouvements des sportifs.

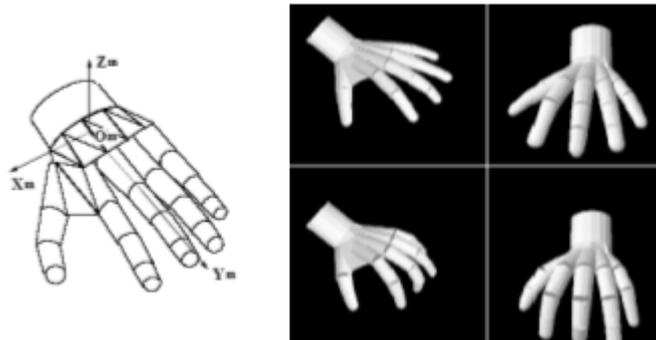


Figure II.24 : Un modèle cinématique de main à 20DDL

Le système GREFIT, pour *Gesture Recognition based on Finger Tips*, a été développé par Nölker et Ritter [Nölker 99] afin d'extraire les postures 3D de la main à partir d'images vidéos. Ils ont développé une main virtuelle composée de 16 éléments rigides - 3 pour chaque doigt et un pour la paume – et de 20 degrés de liberté avec 4DDL par doigt. Chaque élément est constitué de deux parties cylindriques reliées par une pièce prismatique. Des limites articulaires et des contraintes dynamiques ont été incluses afin de réduire l'espace des postures possibles. La Figure II.25 montre les résultats obtenus avec la main virtuelle pour certaines postures :

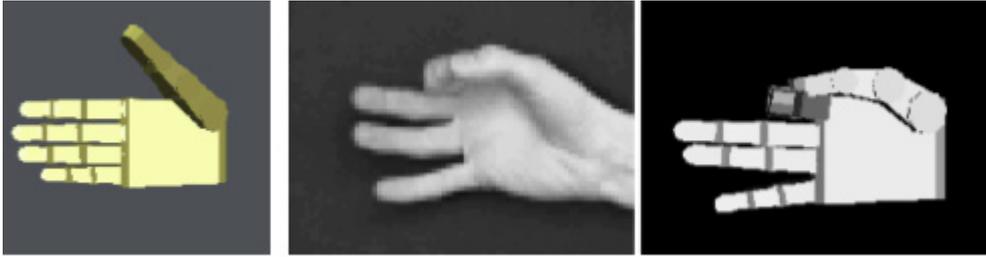


Figure II.25 : Le modèle cinématique utilisé par le système GREFIT

Ouhaddi et Horain [Ouhaddi 98] ont proposé un modèle de main volumique rigide plus complexe. Celui-ci est constitué de cylindres, de sphères et d'un parallélépipède. Il possède 27 degrés de liberté et il inclut les contraintes statiques et dynamiques de la main humaine présentées dans le paragraphe 4 du Chapitre I. Leur modèle tient également compte des différences morphologiques des mains humaines car la taille de la main est corrélée aux longueurs des doigts et des phalanges par le biais de contraintes anthropométriques. Ce dernier est utilisé pour le suivi de la main par vision artificielle dans des séquences vidéo (Cf. Figure II.26).



Figure II.26 : Une main volumique rigide utilisée pour la langue des signes

Savescu et al. [Savescu 04] présentent un modèle de main rigide qui simule les postures de préhension avec plus de degrés de liberté pour la paume. Ainsi, ils utilisent un modèle géométrique rigide constitué de 28 os et de 25 degrés de liberté dont 2 degrés de liberté pour l'arche de la paume (courbure). Ce modèle améliore les représentations des postures de la main et permet de simuler des tâches de préhension complexes de manière plus réaliste comme le montre la Figure II.27 :

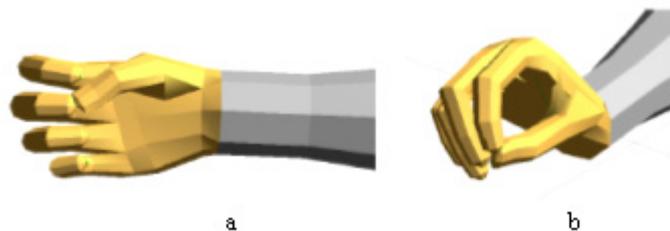


Figure II.27 : Le modèle cinématique de la main de MAN3D

a) Paume rigide – b) Ajout des DDL de l'arche de la paume : les extrémités des doigts sont en contact

Il existe d'autres modèles cinématiques utilisés par des systèmes de reconnaissance de gestes, pour l'étude de l'ergonomie des postures ou pour l'évaluation des performances des gestes sportifs. Ainsi, Vardy [Vardy 98] et McDonald et al. [McDonald 01] ont proposé des modèles cinématiques pour simuler les différents gestes de la langue des signes. Quant à Lin et al. [Lin 00], ils ont simulé 32 postures discrètes pour interagir avec une machine en respectant les mêmes contraintes.

Afin de modéliser le squelette de la main de la manière la plus fidèle et la plus naturelle, les modèles cinématiques tiennent compte de l'anatomie et de la biomécanique de la main humaine. L'intégration des débattements angulaires des phalanges et des contraintes de dépendance des doigts permet de réduire le nombre de degrés de liberté de la main virtuelle ainsi que l'espace des configurations possibles. Ce qui a pour résultat d'accélérer le calcul des positions du modèle de main par la cinématique inverse et de synthétiser des mouvements naturels. Ces modèles permettent donc de générer des animations réalistes pour la main.

Les modèles squelettiques décrits précédemment sont des modèles rigides. Afin d'améliorer le réalisme visuel de ces modèles cinématiques, des méthodes ont été développées pour modéliser la forme de la main et déformer celle-ci de manière naturelle.

Dans les paragraphes qui suivent, nous présentons les différents modèles utilisés pour obtenir des animations plus réalistes, c'est-à-dire des modèles déformables.

2.1.2 Les modèles surfaciques

Les modèles surfaciques consistent à envelopper les formes des objets par des maillages. Nous présentons les méthodes utilisées pour modéliser des objets complexes comme les parties du corps humain.

2.1.2.1 Les surfaces polygonales

Les surfaces polygonales sont constituées d'un ensemble de primitives géométriques planes comme par exemple des triangles ou des quadrilatères. Ce sont donc des approximations linéaires par morceaux du 1^{er} degré d'une courbe ou d'une forme et leur expression mathématique est donnée par l'équation d'un plan :

$$ax + by + cz + d = 0 \quad (\text{Eq. II-9})$$

L'ensemble des polygones interconnectés forme alors un maillage dit « polygonal ». En infographie, les objets sont souvent représentés par des maillages triangulaires car ce type de surface est directement affiché par les cartes graphiques ce qui économise du temps CPU [w3DCafe].

Wu a modélisé la main par 16 primitives articulées par 21 degrés de liberté [Wu 01]. Ce modèle squelettique est ensuite enveloppé d'un maillage polygonal. Ainsi, les mouvements du squelette sous-jacent induisent les déformations du maillage. La Figure II.28 présente une main maillée par des triangles :

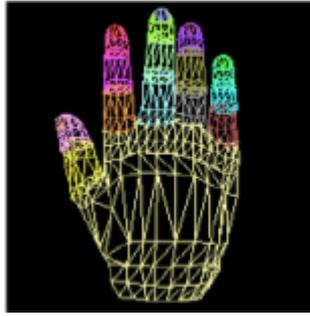


Figure II.28 : Maillage triangulaire de la main

Une méthode de déformation locale a été proposée par Magnenat-Thalmann et al. [Magnenat Thalmann 88]. Ils ont modélisé la main par un squelette et une enveloppe déformable pour la peau. Ils ont développé une méthode de déformation algorithmique de la peau appelée « déformation locale dépendant de l'articulation » ou *Joint-dependent Local Deformation* (JLD) en anglais. Tout d'abord, les sommets du maillage représentant la peau sont projetés sur les segments du squelette : ceci restreint l'influence d'une articulation aux deux segments connectés. Puis, les sommets sont déplacés en fonction des angles aux articulations. Des paramètres additionnels permettent d'ajuster les courbures aux articulations et de simuler les gonflements des muscles.

Bien que le résultat soit réaliste, cette méthode devient inadaptée à des modèles complexes constitués de nombreux points de contrôle en raison des multiples projections dans les repères locaux.

2.1.2.2 Les surfaces paramétriques

Pour réaliser de meilleures approximations que les surfaces polygonales décrites précédemment, des courbes paramétriques et des carreaux de surface ont été spécifiés. Ainsi, ces courbes et surfaces paramétriques enveloppent l'objet 3D.

Il existe une famille de courbes paramétriques regroupant entre autres les courbes de Bézier, les courbes bi-quadratiques, les B-splines rationnelles ou les B-splines rationnelles non uniformes (*NURBS* en anglais). Ces courbes se différencient par leur expression mathématique utilisée pour l'approximation³ ou pour l'interpolation⁴ des points de contrôle.

Dans ce paragraphe, nous présentons les courbes de Bézier utilisées en CAO et les courbes B-splines cubiques qui sont très répandues en infographie [Boyer 05].

2.1.2.2.1 Définitions

Cette approche consiste à utiliser des modèles d'ordre supérieur aux surfaces polygonales. Les coordonnées cartésiennes x , y et z d'un point P sont données par des fonctions à valeurs réelles d'un paramètre réel t telles que :

$$\begin{cases} x = x(t) \\ y = y(t) \\ z = z(t) \end{cases} \quad (\text{Eq. II-10})$$

³ Approximation : la courbe ne passe pas par les points de contrôle mais elle est incluse dans le polygone formé par ces points.

⁴ Interpolation : la courbe passe par les différents points de contrôle et elle est constituée de segments de continuité C^1 .

Ces fonctions sont de continuité C^0 sur l'intervalle parcouru par t et elles définissent une courbe continue sur ce même intervalle. Une telle courbe est dite « lisse » si la dérivée première est continue sur l'intervalle parcouru par t . Ainsi, pour interpoler ou approcher un nombre fini de points d'une courbe, on utilise des courbes de degré 3, dites « cubiques », dont l'équation notée $C(t)$ a la forme suivante :

$$C(t) = (x(t), y(t), z(t)) = \sum_{i=0}^n f_i(t) P_i \quad (\text{Eq. II-11})$$

avec $f_i(t)$ des fonction de pondérations et $P_i = (x_i, y_i, z_i)$ des points de données.

2.1.2.2.2 Les courbes de Bézier

L'équation $C(t)$ est donnée par :

$$C(t) = \sum_{i=0}^n B_{i,n}(t) P_i \quad (\text{Eq. II-12})$$

avec n le nombre de points de contrôle utilisés, $B_{i,n}$ les polynômes de Bernstein, P_i les points de contrôle, et $t \in [0;1]$.

Ainsi, pour deux points de contrôle, l'approximation correspond à une interpolation linéaire :

$$C(t) = (1-t)P_0 + tP_1 \quad (\text{Eq. II-13})$$

Enfin, pour quatre points de contrôle, l'équation est la suivante :

$$C(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3 \quad (\text{Eq. II-14})$$

Cette courbe est connue sous le nom de Bézier cubique. Elle est très utilisée en conception.

2.1.2.2.3 Les courbes B-splines

L'équation $C(t)$ d'une courbe B-spline est constituée d'une combinaison linéaire de fonctions de base B_i de degré au moins trois. La forme générale est donnée par :

$$C(t) = \sum_{i=0}^n B_i(t) P_i \quad (\text{Eq. II-15})$$

avec P_i : les points de contrôle et B_i : les fonctions de base normalisées telles que

$$\sum_{i=0}^n B_i(t) = 1 \quad \forall t.$$

Par exemple, dans le cas 1D, une courbe lisse est calculée en approchant des points de contrôle spécifiés par l'utilisateur. La Figure II.29 [Gibson 97] présente une courbe B-spline cubique définie par quatre points de contrôle a, b, c et d . C'est une fonction polynomiale de degré 3 par morceaux et de continuité C^2 :

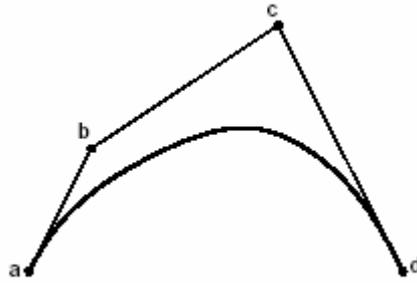


Figure II.29 : Une courbe B-spline cubique

Lorsqu'un point de contrôle est déplacé, la courbe est modifiée localement grâce aux conditions de continuité imposées au niveau de chaque point de contrôle et cela avec un temps de calcul faible.

Les courbes B-splines les plus répandues sont les courbes B-splines cubiques. Ce sont des courbes polynomiales cubiques de continuité C^2 . Elles approximent un ensemble de points de contrôle P_i avec $i \in [0; n]$ par une courbe constituée de $n-2$ segments de courbes Q_i avec $i \in [3; n]$. Chaque segment polynomial Q_i est fonction de quatre points de contrôle P_{i-3} , P_{i-2} , P_{i-1} et P_i ainsi que d'un paramètre t variant de t_i à t_{i+1} .

On distingue deux courbes types : les B-splines cubiques uniformes et les B-splines cubiques non uniformes.

Ainsi, pour les B-splines cubiques uniformes, les paramètres t_i sont uniformément répartis sur l'ensemble des points de contrôle tels que :

$$t_i = t_{i-1} + 1$$

Dans ce cas de figure, la position d'un point de contrôle P est définie par $C(t)$ avec $t \in [0; 1[$:

$$C(t) = \frac{1}{6} \left[(1-t)^3 P_{i-3} + (3t^3 - 6t^2 + 4) P_{i-2} + (-3t^3 + 3t^2 + 3t + 1) P_{i-1} + t^3 P_i \right] \quad (\text{Eq. II-16})$$

Soit sous forme matricielle :

$$C(t) = (t^3, t^2, t, 1) \cdot \frac{1}{6} \begin{pmatrix} -3 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-3} \\ P_{i-2} \\ P_{i-1} \\ P_i \end{pmatrix} \quad (\text{Eq. II-17})$$

La courbe passe par approximation par les points de contrôle et elle est contenue dans l'enveloppe convexe des points de contrôle.

Quant aux B-splines cubiques non uniformes, le paramètre t ne se répartissent pas uniformément de t_i à t_{i+1} . La courbe est donc contrôlée localement par les paramètres.

Il est également possible de pondérer l'influence de chaque point de contrôle sur la forme de la courbe par le biais de poids : ce sont les courbes B-splines rationnelles non uniformes -

NURBS en anglais. Une telle courbe est donc définie par des points de contrôle P_i et des poids w_i associés à chacun de ces points de contrôle. Dans ce cas de figure, les segments de courbe ne sont plus définis par des polynômes mais par des fractions rationnelles de polynômes. La position d'un point P de la courbe est alors définie par l'équation $C(t)$ suivante :

$$C(t) = \frac{\sum_{i=0}^n w_i B_{i,n}(t) P_i}{\sum_{i=0}^n w_i B_{i,n}(t)} \quad (\text{Eq. II-18})$$

avec : P_i : les points de contrôle, $B_{i,n}$: les fonctions de base normalisées et w_i : les pondérations associées aux points de contrôle.

Ainsi J.J. Kuch [Kuch 95] représente la main humaine par des splines qui définissent une approximation arbitraire de sa forme. De nombreux points de contrôle ont été nécessaires pour spécifier le modèle afin que les courbes paramétriques s'ajustent aux formes des mains. Ces surfaces se révèlent être très réalistes au niveau visuel mais leur utilisation devient complexe lorsqu'un tel modèle est déformé. Dans le cas d'un modèle de main pour simuler l'interaction Homme-machine, les nombreux paramètres utilisés conduisent à un temps de calcul long ; ce qui pose le problème du rendu temps réel.

2.1.2.3 Les déformations de forme libre

C'est une méthode globale pour déformer des objets qui nécessite un meilleur niveau de contrôle que l'ajustement individuel des points de contrôle. Elle a été adaptée par Sederberg et Parry [Sederberg 86] pour l'animation de personnages.

Ainsi, les déformations de formes libres – *Free Form Deformations* en anglais ou FFD – modifient la forme d'un objet en déformant l'espace auquel celui-ci est lié. La méthode consiste à envelopper un objet dans un treillis composé d'une grille de points à géométrie standard tel qu'un cube ou un cylindre (Cf. Figure II.30). En manipulant les nœuds de cette grille, on introduit des déformations de l'espace qui transforment les primitives graphiques sous-jacentes qui forment l'objet (Cf. Figure II.31). La déformation est appliquée sur l'objet à l'aide de polynômes de Bernstein trivariants, dont l'ordre dépend du nombre de points de contrôle demandé par l'utilisateur. Ces polynômes donnent de meilleurs résultats lors de la manipulation du maillage par les points de contrôle. Cette forme de déformation est dite indirecte.

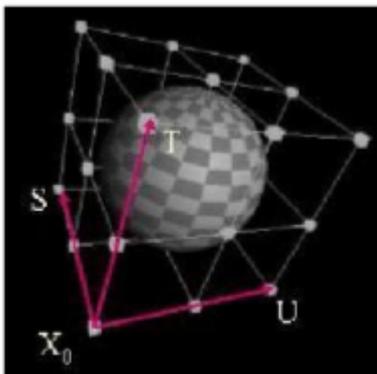


Figure II.30 : Paramétrage de l'objet pour les FFD

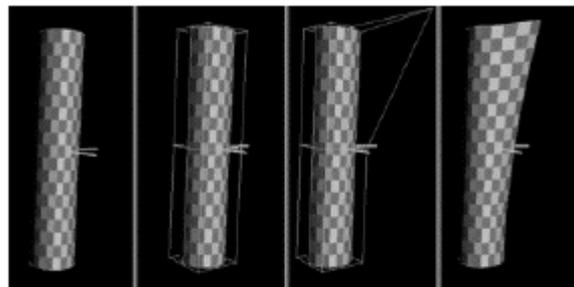


Figure II.31 : Principe des déformations libres

Avec cette méthode, il est possible de réaliser des transformations locales sur un objet en ne paramétrant qu'une partie de cet objet. Elles rendent donc la création de modèles complexes beaucoup plus aisée que les méthodes précédentes. C'est pourquoi les déformations de forme libre sont largement utilisées pour la modélisation de personnages ou celle du corps humain. Mais, il ne faut pas introduire de discontinuités de surfaces entre les parties non paramétrées et les autres.

Des améliorations et des extensions ont été apportées à la méthode de Sederberg et Parry. Par exemple, les EFFD – déformations de forme libre étendues ou *Extended Free Form Deformations* - généralisent les formes des grilles de points de contrôle [Coquillart 90]. Ou encore, les FFD directes permettent de modifier simultanément et directement plusieurs points de l'objet en utilisant des courbes B-splines plutôt que les polynômes de Bernstein [Hsu 92]. Cette méthode rend l'interface plus simple à manipuler et elle améliore la définition des déformations. Enfin, les déformations de forme libre de Dirichlet - appelée *Dirichlet Free Form Deformation* en anglais ou DFFD – généralisent les FFD en supprimant les limitations imposées sur les formes des grilles et en autorisant un contrôle plus local de la déformation. Moccozet et Magnenat-Thalmann ont utilisé cette méthode pour modéliser la main [Moccozet 97]. Des points de contrôle disposés sur les lignes de la main permettent de déformer la paume et les doigts.

Les méthodes décrites dans les paragraphes précédents permettent de modéliser et déformer les objets en les enveloppant dans un maillage surfacique. Dans les cas de la modélisation et de l'animation de la main, les déformations sont dues aux mouvements du squelette sous-jacent et aux actions de contact. Une méthode géométrique spécifique existe et est décrite ci-dessous.

2.1.2.4 La méthode de skinning

Le *skinning* ou les déformations dans l'espace du squelette - *Skeleton Subspace Deformations* en anglais ou SSD - sont une technique de déformation locale de la surface d'un objet en réponse aux mouvements du squelette sous-jacent. C'est un algorithme d'interpolation qui peut être utilisé sur tout type d'articulation ou de surface. Chaque sommet du maillage est soumis à la même transformation rigide que le repère du squelette auquel il est rattaché. C'est une technique simple qui donne des résultats corrects en statique et en dynamique. C'est pourquoi elle est très répandue en animation graphique.

Cependant, cette méthode peut conduire à des déformations irréalistes dues à l'étirement, la superposition ou l'intersection du maillage au niveau des articulations. La Figure II.32 et la Figure II.33 montrent les défauts constatés au niveau du coude [Lewis 00] :

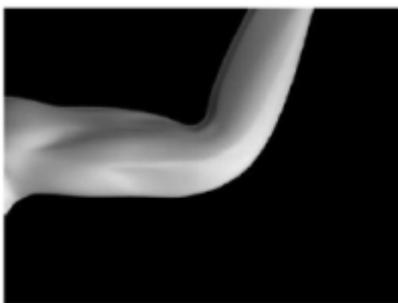


Figure II.32 : Effondrement du coude en flexion



Figure II.33 : Rétrécissement du bras en rotation

Différentes techniques ont été adoptées afin d'améliorer les performances de cet algorithme comme par exemple Singh et Kokkevis [Singh 00] qui ont introduit des FFD, Lewis et al. [Lewis 00] qui ont proposé une méthode pour calculer les pondérations ou encore Kry et al. [Kry 02] qui ont utilisé l'analyse modale pour définir les principaux modes de déformation.

Après la description des différents maillages possibles pour envelopper un objet ainsi que les techniques de déformation existantes lorsque cet objet est animé, nous nous intéressons aux modèles volumiques.

2.1.3 Les modèles volumiques

Certains modèles volumiques sont construits à partir des modèles surfaciques décrits précédemment : c'est le cas des volumes paramétriques. D'autres sont spécifiques comme les primitives ou les surfaces implicites. Dans ce paragraphe, nous présentons les principaux modèles utilisés en animation graphique.

2.1.3.1 Les primitives simples

Une approche mieux adaptée à la vision par ordinateur ou l'animation en temps réel repose sur l'utilisation de structures géométriques 3D simples paramétrées ou encore des super quadriques. Les parties du corps peuvent être modélisées à l'aide de cylindres, de sphères, d'ellipsoïdes ou de rectangles. Les parties plus complexes sont alors obtenues en couplant plusieurs de ces primitives.

L'avantage de l'utilisation des formes géométriques simples est que l'on peut obtenir une bonne approximation de la forme extérieure de la main avec une méthode moins complexe.

Davis et Shah [Davis 94] ont développé un modèle volumique de la main afin de suivre les mouvements des doigts en vision artificielle. Après identification des doigts sur l'image à l'aide de cylindres, ils modélisent les extrémités des doigts par des cylindres généralisés maillés. Les données anatomiques et biomécaniques sont introduites afin de faciliter la reconnaissance et le suivi des doigts (Cf. § 2 et § 4 du Chapitre I).

En règle générale, les modèles squelettiques présentés dans le paragraphe 2.1.1 appartiennent également à cette catégorie.

2.1.3.2 Les volumes construits à partir de surfaces implicites

Les surfaces implicites sont une généralisation des modèles décrits précédemment. Une surface implicite est définie par une fonction mathématique qui attribue une valeur scalaire à chaque point de l'espace. Une iso-surface est alors une surface contenant tous les points qui ont la même valeur scalaire. Un exemple de surface implicite couramment utilisée est l'ellipsoïde. Il est défini par une fonction f par :

$$f : \mathbb{R}^3 \rightarrow \mathbb{R}$$

$$f(x, y, z) = \frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} \quad (\text{Eq. II-19})$$

Le champ scalaire est généralement créé à partir de primitives simples, appelées « sources discrètes » ou « squelettes », telles que des points, des lignes ou des surfaces. Chaque

squelette émet un champ de potentiel avec une certaine distribution décrite par une fonction champ. On obtient de cette manière une épaisseur autour de ce squelette.

On commence avec une forme brute à base de quelques primitives et on ajoute les détails grâce aux fonctions suivantes : ajouter, supprimer, transformer, ajuster, maximum, minimum, etc. On obtient la forme finale en reliant toutes les surfaces créées.

Ainsi, Bloomenthal [Bloomenthal 93] a modélisé la main par des surfaces implicites. Pour ce faire, il a utilisé 85 primitives qui se décomposent en :

- une paume formée de 15 triangles,
- des doigts formés de 48 triangles,
- des tendons qui correspondent à 10 segments,
- des veines qui sont représentées par 11 segments,
- le muscle abducteur du pouce qui est modélisé par un segment.

En modifiant l'échelle, la position ou l'orientation de ces primitives, ainsi qu'en les additionnant, Bloomenthal obtient un modèle de main réaliste (Cf. Figure II.34) :

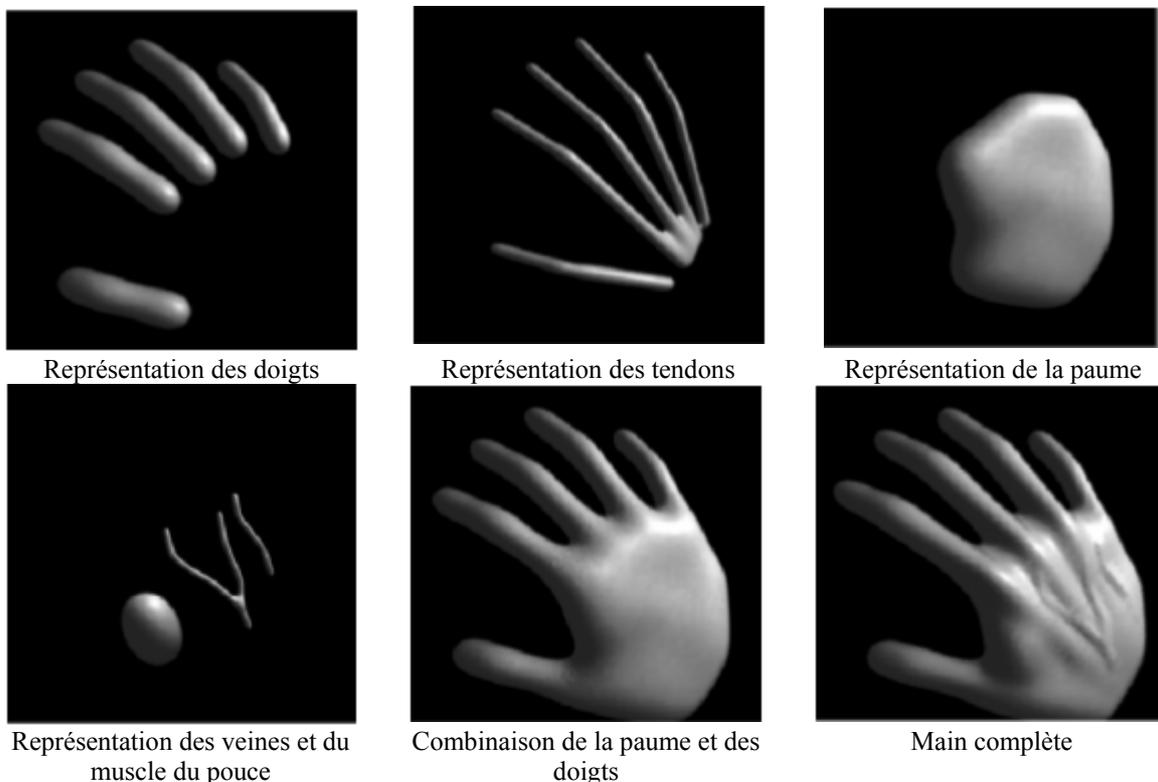


Figure II.34 : Modélisation de la main par des surfaces implicites

Les surfaces implicites possèdent de nombreuses propriétés qui les rendent propices à la modélisation du corps humain. Grâce à leur fonction champ, elles peuvent interagir de manière naturelle les unes avec les autres. De plus, la définition des iso-surfaces permet de localiser rapidement des points – à l'intérieur ou l'extérieur de l'objet - ce qui facilite la détection de collision. Un autre avantage est qu'elles sont définies par des points ou des polygones simples à éditer ce qui rend leur formulation très compacte et ne requiert que peu de place en mémoire.

Mais, lors de l'animation d'un tel modèle, il arrive que des déformations et des contacts non souhaités se fassent comme par exemple la fusion entre plusieurs surfaces implicites. Pour

remédier à ce problème, il faut subdiviser le corps en plusieurs parties et limiter les influences des primitives localement. La difficulté majeure est l’affichage de ces surfaces. En effet, il est nécessaire de créer des surfaces triangulées pour les voir à l’écran mais les algorithmes existants sont coûteux en temps de calcul.

Il existe une sous-catégorie des surfaces implicites appelée « surfaces de convolution » ou *metaballs* en anglais, que nous présentons ci-dessous.

2.1.3.3 Les volumes construits à partir des surfaces de convolution

Les surfaces de convolution - *metaballs* en anglais - sont des sphères spéciales obtenues en ajoutant des distributions de densité. Un paramètre définit un seuil pour le rayon de la surface d’iso-densité à visualiser. Ainsi, un *metaball* isolé apparaît comme une sphère normale. En considérant qu’au voisinage d’autres *metaballs* les fonctions de densité s’ajoutent pour générer une transition souple et naturelle entre eux, on obtient des ellipsoïdes quand ils se rapprochent ou s’éloignent (Cf. Figure II.35) [Bloomenthal 91] :

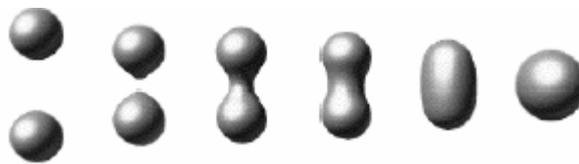


Figure II.35 : Principe des metaballs

Ces surfaces sont fréquemment utilisées pour représenter des formes organiques en raison de leur courbure naturelle.

Dewaele et al. [Dewaele 04] utilisent un modèle volumique de main pour la capture de mouvements. Celui-ci est composé d’un squelette de 27 degrés de liberté et de surfaces de convolution. La paume et les phalanges sont symbolisées par 16 primitives à forme elliptique. La fonction mathématique choisie permet de fusionner les phalanges pour former les doigts ainsi que de fusionner les doigts et la paume pour construire le modèle de main (Cf. Figure II.36) :



Figure II.36 : Modélisation de la main par 16 metaballs

Wan et al. ont développé un modèle de main qui tient compte de l’anatomie et de la forme de la main humaine [Wan 2004]. Ils ont construit un modèle géométrique de la main à l’aide de 27 *metaballs*. Ceci leur permet de modéliser la paume, les phalanges et les coussinets déformables de manière réaliste (Cf. Figure II.37).

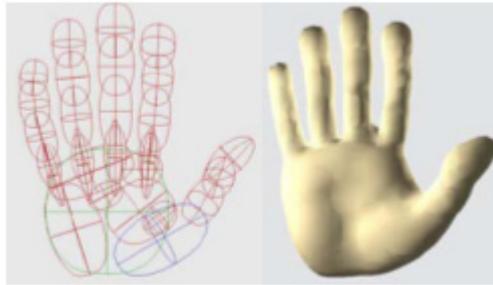


Figure II.37 : La main modélisée par des surfaces de convolution

2.1.3.4 Les volumes paramétriques

En augmentant l'espace paramétrique d'une dimension, on passe des courbes paramétriques à des surfaces bi-paramétriques. Ces surfaces 3D sont appelées « carreau de surface » ou *patch* en anglais.

Les volumes paramétriques sont le résultat des surfaces paramétriques fermées. Ils sont définis dans l'espace à partir des équations du § 2.1.2.2. Dans ce paragraphe, nous détaillons les carreaux de surface issus des courbes de Bézier et des courbes splines.

2.1.3.4.1 Les surfaces de Bézier

Les surfaces paramétriques sont formées à partir de grilles de points de contrôle. L'équation d'une surface de Bézier est la suivante :

$$C(s, t) = \sum_{i=0}^n \sum_{j=0}^m P_{ij} B_{i,n}(s) B_{j,m}(t) \quad (\text{Eq. II-20})$$

avec $(m+1)(n+1)$ points de contrôle P_{ij} et $s, t \in [0; 1]$

Dans le cas de surfaces de Bézier cubiques, les grilles sont formées de seize points de contrôle.

Comme nous l'avons vu précédemment, les courbes B-splines rationnelles non uniformes sont souvent utilisées en animation graphique. C'est également le cas des surfaces de *Nurbs* pour décrire l'apparence visuelle spatiale des objets.

2.1.3.4.2 Les surfaces NURBS

Les B-splines rationnelles non uniformes sont souvent mises en œuvre pour représenter des surfaces lisses à géométrie complexe. Ces surfaces paramétrées sont d'ailleurs le standard de nombreux logiciels commercialisés comme Maya [wAlias], 3DS Max [w3ds], Blender [wBlender] ou encore Rhino3D [wRhino].

Il existe plusieurs avantages à utiliser ces surfaces. Tout d'abord, les équations restent les mêmes quelle que soit la forme de l'objet à modéliser. De plus, les déplacements des points de contrôle sont calculés rapidement ce qui permet d'avoir un rendu visuel en temps réel.

Bien que les résultats des surfaces paramétriques soient très réalistes, il existe des inconvénients à leur utilisation. En effet, les surfaces et les volumes créés sont limités : il est

difficile d'ajouter des détails ou d'obtenir une topologie originale avec ces surfaces paramétriques. Ceci rend impossible la création de surfaces ou de volumes trop complexes.

Forsey [Forsey 91] a proposé une méthode pour déformer dynamiquement des surfaces paramétriques et trouver un compromis entre réalisme visuel et temps de calcul. Pour ce faire, il a utilisé des carreaux de surface formés de B-splines hiérarchiques. Ces surfaces autorisent l'insertion locale de nœuds dans le maillage ainsi que la manipulation multi-résolution qui sont des propriétés essentielles pour l'animation. Il a réussi à animer un personnage géométriquement en couplant un squelette à des surfaces de B-splines hiérarchiques pour modéliser la peau. Cette représentation permet de modifier directement les points de contrôle en fonction des angles des articulations et par conséquent, de générer des petites et des larges déformations avec un même modèle.

2.1.4 Conclusion

Nous avons présenté dans les paragraphes précédents les principaux modèles géométriques de la main utilisés en animation graphique ainsi que certaines techniques de déformation.

Ainsi, les modèles squelettiques permettent de simuler le comportement d'un objet articulé. Dans le cas de la main, ces modèles reproduisent la cinématique de la main humaine ce qui permet de reproduire les postures et mouvements de celle-ci. Afin d'améliorer le réalisme du comportement, de nombreux modèles intègrent en plus les contraintes biomécaniques décrites dans le paragraphe 4 du Chapitre I.

Bien que les modèles squelettiques simulent correctement tous les gestes de la main, ils sont pénalisés par leur manque de réalisme visuel. D'où le développement des modèles surfaciques et volumiques.

Ces derniers permettent de représenter la main par sa forme extérieure (maillage polygonal ou surface paramétrique) ou alors de la reconstruire (formes extérieure et intérieure) à partir de volumes (primitives, surfaces de convolution, ...). Dans le cas statique, ces modèles se révèlent être très réalistes visuellement : il suffit d'augmenter le nombre de points de contrôle ou le nombre de volumes.

Non seulement il faut modéliser la main, mais il est également nécessaire de l'animer ; c'est-à-dire de la déformer au cours du temps de manière réaliste. En effet, modifier la forme de ces modèles revient à manipuler les points de leur maillage ou à modifier des fonctions mathématiques plus complexes. Cette manipulation est intuitive dans le cas de surfaces paramétrées mais elle demeure néanmoins fastidieuse pour déformer des objets complexes tel que le corps humain et a fortiori la main.

C'est pourquoi des techniques de déformation spécifiques – telles que les déformations de forme libre (FFD), le *skinning* ou les déformations aux articulations (JLD) - ont été proposées afin de conserver le réalisme visuel des modèles.

Dans notre cas de figure, la main est considérée comme un corps déformable articulé. Les méthodes de déformations les mieux adaptées sont donc les déformations JLD et éventuellement le *skinning*. Cette technique est à améliorer car la main possède de nombreuses articulations.

Les modèles et les techniques de déformation cités sont en règle générale faciles à utiliser. Elles permettent de modéliser et d'animer des objets complexes mais leur principal inconvénient est qu'elles ne permettent pas de créer une réponse dynamique aux mouvements. En effet, ces méthodes dédiées à l'animation graphique permettent de générer des déformations cinématiques et non dynamiques. Leur utilisation est donc limitée.

Il est donc important d'étudier d'autres modèles car les méthodes géométriques ne répondront pas à notre besoin. Dans le paragraphe suivant, nous présentons les modèles qui permettent d'interagir avec les objets environnants.

2.2 Les modèles développés pour manipuler

Les modèles géométriques sont très répandus car ils permettent de communiquer entre deux humains, entre un humain et un système ou encore entre deux humains virtuels.

Dans le Chapitre I, nous avons vu que la principale fonction de la main est d'interagir avec l'environnement. Comme l'objectif de cette thèse est de simuler des interactions Homme-machine, il est nécessaire de connaître les modèles existants qui permettent d'interagir avec l'environnement et de manipuler des objets.

Ces modèles peuvent être classés en trois catégories :

- les modèles dits « heuristiques » pour simuler une préhension automatiquement. Ils sont basés sur une bibliothèque de gestes et des stratégies de prises pré-définies ;
- les modèles pour simuler une préhension interactive. Ils utilisent les données issues de la capture de mouvements pour piloter la main et des méthodes de détection de collision pour identifier le contact entre la main virtuelle et les objets. La stratégie de saisie est définie suivant la configuration en cours ;
- les modèles dits « physiques » qui tiennent compte des propriétés mécaniques de la main et des objets pour réaliser des préhensions avec un réalisme physique.

Nous présentons ci-dessous les modèles développés pour chacune des catégories.

2.2.1 Les modèles pour une préhension automatique

Ces modèles reposent sur une bibliothèque de postures – clés de la main virtuelle qui ont été enregistrées au préalable à partir d'un modèle générique de la main, d'images vidéos ou d'images obtenues par scanner ou IRM⁵. Au cours de la simulation, diverses méthodes sont utilisées pour choisir les postures adéquates et pour les animer dans le but de saisir des objets en environnement virtuel. Ces modèles permettent de réaliser une préhension dite « automatique » avec un bon rendu visuel.

La première approche semi-automatique de la préhension a été proposée par Magnenat-Thalmann et al. [Magnenat Thalmann 88] avec un modèle de main paramétrique et dans le cadre de la simulation des interactions en environnement virtuel. Tout d'abord, l'animateur positionne la main de l'acteur synthétique dans l'espace et il définit les points de contact possibles avec l'objet à saisir. Les angles des phalanges sont ensuite calculés par l'ordinateur par une méthode dichotomique afin que les doigts se trouvent en contact avec l'objet.

Cette méthode est à la base de toutes les méthodes utilisées pour simuler des tâches de préhension. Bien que le positionnement manuel de la main de l'acteur synthétique soit fastidieux, le principe était ce que l'on pouvait faire de mieux avec les moyens techniques de l'époque. Ainsi, cette approche a été reprise et améliorée au fur et à mesure des développements technologiques et des progrès algorithmiques.

Rijkema et Girard [Rijkema 91] ont mis en œuvre une approche basée sur la connaissance pour simuler des tâches de préhension. Pour cela, ils utilisent un modèle cinématique rigide de la main qui se compose de primitives simples et qui possède 21 degrés de liberté. Des

⁵ IRM : Imagerie par Résonance Magnétique

contraintes biomécaniques sont intégrées afin de réduire le nombre de configurations possibles.

La première étape consiste à créer la base de données pour simuler les tâches de préhension. Ainsi, des postures - clés du modèle de main sont pré-calculées et sont mémorisées. Les objets à saisir sont également identifiés et classés suivant leur forme géométrique. Puis, au cours de la simulation, les solutions possibles sont cherchées dans la base de connaissances soit par l'animateur – i.e. une préhension semi-automatique – soit par un système expert – i.e. une préhension automatique. Par cette approche, Rijpkema et Girard sont parvenus à simuler plusieurs stratégies de préhension visuellement réalistes avec leur modèle de main.

Avec les technologies actuelles comme la capture de mouvements, il est d'ailleurs plus facile et plus rapide de mémoriser les positions-clés pour réaliser une animation. C'est pourquoi l'approche basée sur la connaissance reste d'actualité comme nous le verrons par la suite.

Une autre méthode heuristique a été proposée par Mas Sanso [Mas Sanso 94]. La main est modélisée par des primitives simples rigides articulées par 21 degrés de liberté. L'intégration des contraintes biomécaniques permet de simplifier la cinématique du modèle (Cf. Figure II.38).

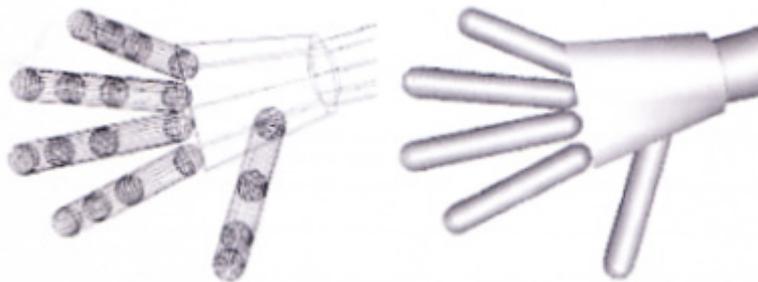


Figure II.38 : Le modèle cinématique rigide

Comme dans la méthode précédente, Mas Sanso mémorise les objets à saisir sous une forme simplifiée telle que la sphère ou le cube. Mais, il identifie et enregistre également les zones de préhension possibles de ces objets. Ceci l'a conduit à construire une taxonomie des prises possibles basée sur les formes des objets.

Ainsi, quand l'animateur choisit l'objet à saisir, le système trouve automatiquement la stratégie qui convient en se basant sur la forme et la masse de cet objet et en respectant la taxonomie définie précédemment. La position finale des doigts sur l'objet est calculée par cinématique inverse et par une méthode de détection de collision algébrique (Cf. Figure II.39).

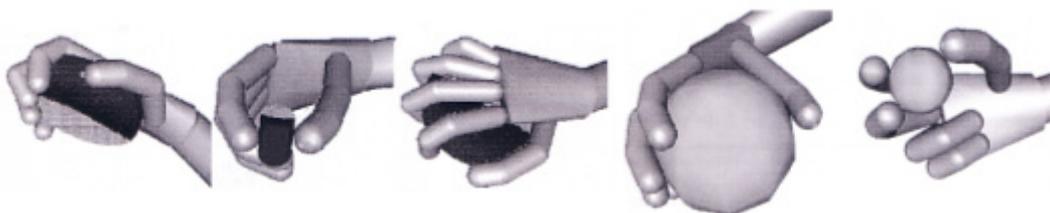


Figure II.39 : Différentes prises d'objets simulées automatiquement

Ce modèle inspiré de la robotique permet de réaliser un grand nombre de prises entre une main et des objets virtuels rigides. Dans les paragraphes suivants, nous verrons comment ce modèle a évolué pour que les interactions se fassent entre une main virtuelle rigide et des objets déformables.

Avec les progrès techniques des ordinateurs et avec le développement de nouvelles technologies comme les interfaces utilisées en environnement virtuel, de nouvelles approches ont vu le jour. Ce sont des méthodes qui permettent de simuler des tâches de préhension en interactif.

2.2.2 Les modèles pour une préhension interactive

Rezzonico et al. ont amélioré le modèle de Mas Sanso en le couplant à un gant de données et en ajoutant une approche multi-capteurs pour contrôler la prise [Rezzonico 95]. Ils ont construit un modèle de main rigide avec des capteurs sphériques associés à chaque articulation (Cf. Figure II.40) :

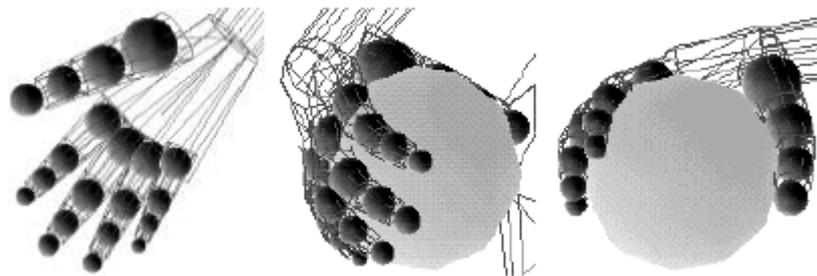


Figure II.40 : Le modèle de main avec des capteurs aux articulations

Cette configuration permet de calculer les points de contact entre les doigts de la main et l'objet de manière plus précise. Elle améliore donc la détection de collision et elle permet à la main virtuelle d'envelopper naturellement l'objet saisi. De plus, des déplacements normaux et tangentiels peuvent être générés au moment de la collision ce qui permet à l'utilisateur de manipuler l'objet [Boulic 96].

Ce modèle s'inspire des réseaux d'actionneurs et des approches automatiques de saisie existant en robotique et électronique. Il est également le premier à autoriser la manipulation des objets par une main virtuelle avec des contacts frottants.

Des extensions ont été apportées à ce modèle. Ainsi Huang [Huang 96] et Moccozet [Moccozet 96] ont amélioré le réalisme visuel de la main virtuelle en ajoutant une enveloppe surfacique déformée par la technique des déformations libres de Dirichlet (Cf. § 2.1.2.3) (Cf. Figure II.41) :

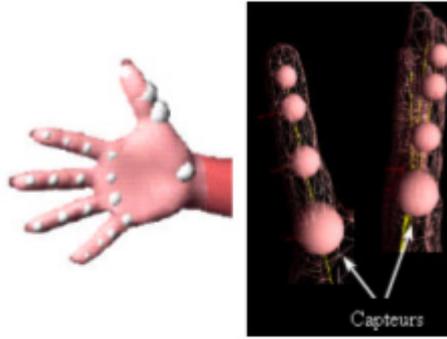


Figure II.41 : Le modèle de main déformable géométriquement

Ils ont également élargi la méthode de saisie à l'interaction entre humains en gérant l'auto-collision (Cf. Figure II.42) :



Figure II.42 : Serrage de mains entre deux acteurs virtuels

Les équipes de recherche de l'Ecole Polytechnique Fédérale de Lausanne et du laboratoire MIRALAB de l'Université de Genève sont très actives dans la modélisation des environnements virtuels et des acteurs synthétiques ainsi que dans la simulation des interactions entre les deux. Les résultats qu'ils ont obtenus sont toujours pertinents et sont à l'origine des nombreux développements actuels. C'est pourquoi la plupart des modèles de mains et des modèles de préhension présentés décrivent les résultats obtenus par ces équipes de recherche suisses qui ont été et qui restent en avance sur les autres en Réalité Virtuelle.

Mais, l'animation graphique de personnages et a fortiori celle de la main ainsi que les interactions Homme-machine sont devenues des thèmes très étudiés depuis le développement de la Réalité Virtuelle. Les modèles qui suivent présentent des résultats récents d'autres équipes sur ces sujets de recherche.

Zachmann a développé une approche semblable aux précédentes mais pour simuler des interactions Homme-machine en environnement virtuel et plus particulièrement dans le cadre de tâches de montage/démontage [Zachmann 00]. Son modèle pour la préhension repose sur une méthode de détection de collision par hiérarchie de volumes englobants.

Une main virtuelle articulée rigide est pilotée par un système de capture de mouvements électromagnétique et permet à l'utilisateur d'interagir avec l'environnement. Grâce à la détection de collision et grâce à l'identification de quatre stratégies de prise, l'utilisateur est capable de saisir et manipuler un objet ou un outil afin de réaliser un assemblage virtuel.

L'efficacité et la rapidité de la détection de collision permettent de simuler des tâches de préhension de manière naturelle et intuitive pour l'utilisateur tout en respectant la contrainte de temps réel.

Cette approche est intéressante pour nos travaux de recherche car elle a le même objectif que nous : c'est-à-dire, simuler les interactions Homme-machine dans des environnements virtuels contraints. Les solutions algorithmiques apportées par Zachmann répondent en partie à notre besoin.

Un autre modèle intéressant a été développé par ElKoura et Singh [ElKoura 03] pour l'apprentissage musical. Les mouvements de la main du guitariste sont traqués par 24 marqueurs électromagnétiques. Ces mouvements sont reproduits par une main virtuelle possédant 27 degrés de liberté en tenant compte d'hypothèses simplificatrices (comme l'indépendance entre les mouvements d'abduction/adduction entre les doigts), en utilisant une détection de collision algébrique et en appliquant la méthode de cinématique inverse. ElKoura et Singh sont parvenus à rejouer virtuellement des notes et de courts morceaux de guitare avec ce modèle (Cf. Figure II.43).



Figure II.43 : Progression de la main avec une guitare

Bien que l'application consiste à jouer de la guitare, ce modèle est intéressant car il montre qu'il est possible de réaliser des mouvements complexes en temps réel avec une main virtuelle de 27 degrés de liberté.

Wan et al. [Wan 04] ont développé un schéma de saisie automatique spécifique aux assemblages en environnement virtuel. Pour ce faire, ils ont classé les différents composants d'un assemblage – outils et pièces – en trois groupes de primitives simples (cube, cylindre et sphère). Ensuite, ils ont défini différentes configurations de la main virtuelle pour saisir ces objets : ils ont ainsi obtenu une méthode heuristique.

Comme les mouvements de la main de l'utilisateur sont suivis par un gant de données avec retour d'effort, toute interaction – collision ou contact - avec l'environnement virtuel est directement détectée. Une configuration pour saisir l'objet virtuel est alors automatiquement générée à partir des stratégies pré-enregistrées.

Cette méthode automatique est adaptée aux simulations temps-réel mais elle reste limitée car elle ne propose que quelques prises pour interagir avec l'environnement. De plus, le travail de simplification de l'environnement – appelé « abstraction » - peut se révéler long et fastidieux dans le cas d'environnements complexes.

Récemment, une nouvelle approche a été proposée par Pollard et Zordan [Pollard 05]. Elle repose sur une base de prises réalistes construite à partir des données issues de la capture de mouvements. Une large gamme de mouvements libres, de prises d'objets ou encore d'interactions bi-manuelles a ainsi été simulée et mémorisée.

La main virtuelle est asservie à la main de l'utilisateur par le biais d'un contrôleur. Ce dernier permet de choisir la stratégie de saisie adéquate et de l'adapter à la situation dès que la main virtuelle entre en contact avec son environnement. Il est ainsi possible de simuler des tâches de préhension avec de nombreux objets et de serrer une autre main (Cf. Figure II.44) :



Figure II.44 : Serrage de deux mains par une méthode basée sur l'exemple

L'inconvénient de cette approche est qu'elle n'est pas réaliste physiquement puisqu'il est impossible de contrôler les forces de préhension. Néanmoins, elle offre de nombreuses stratégies de prises pour des applications temps-réel.

Bien que ces modèles permettent de simuler les contacts entre une main virtuelle et son environnement, ils ne sont pas réalistes physiquement parlant. En effet, la plupart des modèles simulent la préhension et/ou la manipulation d'objets rigides avec une main virtuelle rigide.

Afin d'être plus proche de la réalité, il est important de simuler des objets déformables et/ou une main qui tienne compte des caractéristiques mécaniques de la main humaine.

Le paragraphe qui suit présente des modèles de main basés sur la physique.

2.2.3 Les modèles basés sur la physique

Lorsque la main interagit avec son environnement, elle applique des efforts et elle subit des actions extérieures. Il est donc important de prendre en compte les actions de contact qui s'exercent entre la main et l'objet lors d'une tâche de préhension. C'est pourquoi les modèles présentés ci-après sont déformables.

Avec le progrès technologique, la communauté informatique a commencé d'explorer des méthodes dites « physiques » pour l'animation et la modélisation. Ces méthodes utilisent des principes physiques et la puissance de calcul pour la simulation réaliste de procédés physiques complexes qu'il serait impossible de modéliser avec des techniques géométriques pures [Debunne 00].

Les deux modèles de déformation basés sur la physique les plus répandus sont :

- les modèles appelés « masses/ressorts »,
- les modèles basés sur la méthode des éléments finis ou MEF.

Nous décrivons brièvement les propriétés de ces modèles.

2.2.3.1 Les modèles masses/ressorts

Un système masses/ressorts est une technique simple et intuitive pour modéliser les objets déformables. La méthode consiste à discrétiser l'espace en un ensemble de valeurs ponctuelles, si possible uniformément réparties pour une meilleure représentation.

L'objet est donc modélisé par une collection de masses ponctuelles reliées entre elles par des relations de type « ressort » dans une structure multicouches [Gibson 97] (Cf. Figure II.45) :

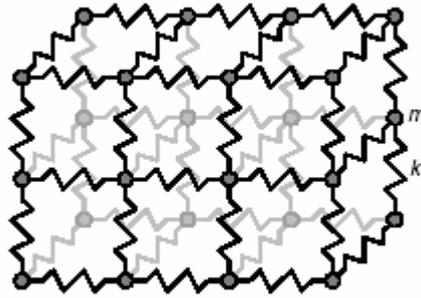


Figure II.45 : Illustration d'une partie d'un système masse/ressort

La plupart du temps, les forces des ressorts sont considérées comme linéaires : ils sont appelés « ressorts de Hooke ». Dans ce cas de figure, la force subie par chaque masse à chaque instant en fonction des distances à ses voisines est calculée par l'équation suivante [Debunne 00] :

$$f_i = \sum_j k_{ij} (l_{ij} - l_{ij}^0) \quad (\text{Eq. II-21})$$

avec f_i la somme des forces extérieures agissant sur la masse i , k_{ij} la raideur du ressort entre la masse i et sa voisine j , l_{ij} étant la longueur entre les deux, l_{ij}^0 leur longueur initiale.

Les ressorts peuvent aussi être plus complexes et comporter des termes d'amortissements.

Dans le cas d'un système dynamique, le mouvement d'une unique masse ponctuelle d'un maillage est régi par la 2nde loi de Newton. L'équation sera donc de la forme [Debunne 00] :

$$m_i \ddot{x}_i + d_i \dot{x}_i + k_i x_i = f_i^{\text{ext}} \quad (\text{Eq. II-22})$$

avec m_i , d_i et k_i trois scalaires désignant respectivement la masse, l'amortissement et la raideur, x_i sa position dans \mathcal{R}^3 et f_i^{ext} la somme des forces extérieures agissant sur la masse i .

L'algorithme consistera à intégrer ces équations au cours du temps pour chaque point. Pour toutes les masses ponctuelles du maillage, on obtient le système matriciel suivant à résoudre [Debunne 00] :

$$M\ddot{x} + D\dot{x} + Kx = f_{\text{ext}} \quad (\text{Eq. II-23})$$

avec M , D , K les matrices $3n \times 3n$ respectives de masse, d'amortissement et de raideur.

Ces modèles sont faciles à construire et ils permettent de réaliser des simulations en temps-réel. C'est pourquoi ils sont souvent utilisés en animation graphique ou pour la simulation chirurgicale [Brown 02].

Des améliorations et des extensions ont été apportées au modèle de base. Ainsi, le modèle proposé par Terzopoulos et Waters [Terzopoulos 90] permet d'utiliser des ressorts de raideurs différentes pour animer les différentes couches du visage.

Il est également possible d'utiliser des ressorts non linéaires pour modéliser des objets avec un comportement inélastique comme l'ont réalisé Provot [Provot 95], Baraff et Witkin [Baraff 98] dans le cas des tissus ou celui de la peau humaine.

De nouveaux comportements plus complexes ont également vu le jour par l'ajout de ressorts angulaires en plus des ressorts linéaires classiques comme par exemple pour simuler les déformations des muscles [Nedel 00].

Le principal inconvénient est que les constantes des ressorts ne sont pas toujours évidentes à déterminer pour obtenir un comportement homogène et réaliste.

Des modèles de mains utilisant un maillage masses/ressorts existent et nous les présentons ci-dessous.

Ainsi, Sibille et al. [Sibille 02] ont développé un modèle multi-couches de la main. Celui-ci est constitué d'un squelette polyarticulé rigide de 29 os, de tissus mous et d'une peau. Les doigts sont positionnés automatiquement par la minimisation de l'énergie.

Bien que les tendons ne soient pas modélisés géométriquement, ils appliquent des couples aux os du squelette par le biais de ressorts rotationnels positionnés à chaque articulation. Leurs actions s'opposent aux mouvements des doigts. Quant aux tissus mous, ils sont modélisés par des maillages masses/ressorts de forme elliptique positionnés sur les phalanges et sur la paume. Ces systèmes masses/ressorts totalisent 700 masses ponctuelles et 1400 ressorts. Leur déformation volumique est calculée par une méthode prédictive-corrective. Au cours de la simulation, les collisions entre les maillages adjacents sont résolues en respectant la contrainte d'incompressibilité des tissus mous. Un maillage surfacique enveloppant les deux modèles sous-jacents représente la peau de la main. Cette dernière s'ajuste automatiquement aux formes des coussinets elliptiques.

Ce modèle de la main humaine est animé de manière réaliste interactivement et quasiment en temps réel. Mais pour obtenir ce résultat, il a fallu régler les raideurs au cas par cas de manière expérimentale compte tenu du fait que le modèle masses/ressorts est discret. Ce travail de préparation et de réglage des paramètres est d'autant plus long et fastidieux que le modèle de main est complexe. Malgré la prise en compte de la biomécanique et de l'anatomie de la main humaine, ce modèle ne permet pas d'interagir avec l'environnement ni de manipuler des objets.

Albrecht et al. [Albrecht 03] ont également proposé un modèle multi-couches de la main basé sur l'anatomie. Celui-ci est constitué d'un squelette rigide articulé de 29 os, de muscles et d'une peau maillée par 3000 triangles. Le modèle cinématique est simplifié par l'intégration des butées articulaires et par la suppression des degrés de liberté à faible mobilité comme ceux situés au niveau des os du carpe. Ce squelette est animé par les contractions musculaires. Pour ce faire, ils ont développé un modèle de muscles hybride constitué (Cf. Figure II.46) :

- de pseudo-muscles qui actionnent la chaîne cinématique mais qui ne sont pas représentés explicitement,
- de muscles géométriques qui modélisent les déformations dues aux mouvements du squelette sous-jacent.

Les pseudo-muscles se comportent comme les muscles anatomiques de la main. Cela signifie qu'ils génèrent des couples moteurs aux articulations à partir des contractions musculaires de leurs fibres. Quant aux muscles géométriques, ils sont situés entre le squelette et la peau et ils sont modélisés par des systèmes masses/ressorts qui totalisent 1500 nœuds et 4000 ressorts. Ainsi, les mouvements des os provoquent le déplacement des nœuds des maillages masses/ressorts, ce qui entraîne la déformation et le gonflement du maillage surfacique, i.e. la peau.

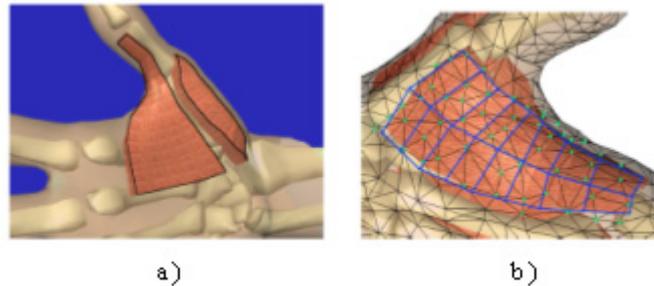


Figure II.46 : Modélisation des muscles de la main par un système masses/ressorts

a) Modélisation des muscles du pouce – b) Les muscles géométriques sont créés automatiquement à partir d'un maillage surfacique (en bleu). Les nœuds en vert sont les sommets du maillage surfacique de la peau qui sont influencés par les muscles sous-jacents.

Ce modèle est animé interactivement. Comme Albrecht et al. ont intégré des propriétés anatomiques et biomécaniques à leur modèle, les déformations sont réalistes d'un point de vue physique. Néanmoins, il est impossible d'interagir avec l'environnement avec ce modèle.

Récemment, il a été possible de saisir et manipuler des objets virtuels avec un modèle de main déformable et avec du retour d'effort. Ainsi, Borst et Indugula [Borst 05] ont présenté un modèle de main constitué d'un squelette articulé rigide et d'un système masses/ressorts avec des ressorts linéaires et rotationnels pour symboliser les tissus mous et la peau. A l'aide d'un gant de données avec retour d'effort, ils suivent les mouvements de la main réelle. Ces configurations sont ensuite utilisées par un contrôleur proportionnel-dérivé pour asservir la main virtuelle. Ils mesurent également les forces exercées sur les objets virtuels lors de leur préhension. Ceci leur permet de calculer la distance de pénétration entre l'objet et la main lors du contact. La résolution des contacts permet de connaître le déplacement des nœuds du maillage et donc de déformer la main.

Le taux de rafraîchissement de la simulation est suffisamment élevé pour simuler et contrôler en temps-réel les saisies d'objets virtuels avec de la capture de mouvements et du retour d'effort. Avec cette approche basée sur la physique, Borst et Indugula ont obtenu une simulation avec un rendu haptique réaliste pour des tâches de préhension en environnement virtuel. Ce modèle est très proche de celui que nous proposons mais contrairement à nous, il met en œuvre un modèle masses/ressorts pour les parties déformables.

Dans le paragraphe qui suit, nous présentons le principe de base de la méthode des éléments finis mais une description complète est donnée par Bathe [Bathe 96], Garrigues [Garrigues 02] ou Salençon [Salençon 05]. Nous apporterons également plus de précisions sur ces modèles dans le paragraphe 4 du Chapitre IV.

2.2.3.2 La méthode des éléments finis

Par définition, le modèle continu d'un objet déformable considère l'équilibre général d'un corps qui subit des forces extérieures. La déformation de l'objet est une fonction de ces forces d'action et des propriétés des matériaux de l'objet. L'objet est en équilibre quand son énergie potentielle est au minimum. La méthode des éléments finis permet de représenter la fonction continue recherchée par une approximation discrète. Plus précisément, elle permet de trouver analytiquement la valeur de la déformation qui satisfait la position d'équilibre. C'est donc une méthode spécifique à la simulation d'objets déformables.

Les modèles éléments finis (EF) considèrent l'objet déformable comme un milieu continu. Celui est subdivisé en un nombre fini d'éléments discrets qui sont reliés entre eux par des nœuds - i.e. les sommets des éléments. Le comportement du matériau est représenté au niveau de l'élément par un nombre fini de paramètres. Ces derniers sont en fait assignés à chacun des nœuds de l'élément.

Une fonction solution de l'équation d'équilibre est trouvée pour chaque élément. La solution dépend des conditions limites ; c'est-à-dire des contraintes aux nœuds et des éléments à la frontière. En regroupant toutes ces fonctions dans une matrice, on obtient le système matriciel suivant à résoudre pour obtenir l'approximation de la déformation sur le maillage :

$$K U = F \quad (\text{Eq. II-24})$$

avec K , la matrice de rigidité du système de dimension $n \times n$, U et F les vecteurs rassemblant respectivement les déplacements et les forces extérieures appliquées aux n sommets du maillage.

La définition précédente est une analyse statique de l'objet : ce dernier n'est soumis qu'à des forces ponctuelles distribuées. Comme en animation l'objet suit une trajectoire dynamique, on cherche généralement à trouver ses déformations au cours du temps et non plus ses positions d'équilibre. Dans ce cas de figure, il faut considérer les forces d'inertie et l'énergie de dissipation, d'où le système matriciel à résoudre :

$$M \ddot{U} + D \dot{U} + K U = F \quad (\text{Eq. II-25})$$

avec M , D et K , respectivement les matrices de masse, d'amortissement, de raideur et F , les forces extérieures. Ces matrices sont obtenues en regroupant les équations de chaque sommet : cette opération est appelée « assemblage ». Elles peuvent dépendre de la géométrie des éléments. Dans ce cas, elles doivent être recalculées dès que l'objet se déforme ce qui est très coûteux en temps de calcul.

En général, l'objet est considéré comme se déformant suffisamment peu pour que ces matrices soient constantes au cours du temps. Mais cette approximation n'est vraie que pour les petites déformations : le modèle éléments finis est dit « modèle linéaire ». Dans ce cas de figure, la théorie élastique linéaire donne de bons résultats.

La méthode des éléments finis nécessite moins de nœuds pour estimer les déformations et elle est plus réaliste que les systèmes masses/ressorts car elle tient compte du comportement mécanique du matériau : les équations intègrent la loi de comportement du matériau simulé. C'est pourquoi elle est souvent utilisée dans des simulations interactives avec du retour d'effort comme les simulateurs chirurgicaux. Cependant, les temps de calcul sont relativement longs en raison de l'inversion de la matrice de raideur K et de l'intégration sur le volume pour trouver la solution globale.

De nombreuses techniques ont été développées afin d'optimiser le temps de calcul [Delingette 98] comme par exemple le fait de considérer un système quasi-statique avec des intervalles de temps fixés ou de pré-calculer les matrices M , D et K . Une approche intéressante a été proposée par Bro Nielsen [Bro Nielsen 96] et Cotin [Cotin 99] dans laquelle ils ne considèrent que les déplacements des nœuds à la surface. L'équation II-24 est alors résolue en séparant les nœuds à l'intérieur de l'objet des nœuds à la surface. Le système matriciel à résoudre est donc de plus petite taille. Cette méthode est appelée « condensation ».

La méthode développée par Debunne [Debunne 00] est dans la même lignée. Il représente le modèle par des maillages avec différentes résolutions : le maillage est raffiné dans les zones

où les déformations sont les plus importantes. Ces éléments finis dits « adaptatifs » permettent de concentrer la puissance de calcul sur les zones déformées ce qui optimise le temps de résolution. Ces approches permettent de simuler des déformations en temps réel avec la méthode des éléments finis en respectant certaines hypothèses simplificatrices (modèle élastique linéaire ou petites déformations). Mais des modèles complexes comme les modèles non linéaires sont également simulés pour des applications temps-réel.

Dans la suite du paragraphe, nous décrivons les modèles de mains mettant en œuvre la méthode des éléments finis pour simuler des déformations réalistes de la main et/ou de l'objet en contact.

Gourret et al. [Gourret 89] ont utilisé une méthode numérique basée sur la théorie des éléments finis pour simuler de manière réaliste les déformations dues au contact entre la main et un objet lors d'une tâche de préhension.

Le modèle mis en œuvre est un modèle en couches constitué d'os à géométrie réaliste, avec une chair et une peau déformables. La peau et la chair volumique des doigts sont discrétisées en éléments finis mais par un maillage non uniforme qui est plus fin aux articulations. Gourret et al. imposent l'hypothèse des petites déformations et utilisent un modèle élastique linéaire pour la chair et l'objet déformables (i.e. avec la loi de Hooke comme loi de comportement) (Cf. Figure II.47). Ces derniers sont donc discrétisés par des éléments finis cubiques.

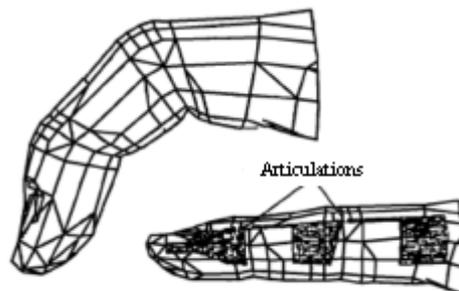


Figure II.47 : Modélisation du doigt par un maillage d'éléments finis volumiques

Comme le mouvement de saisie de la balle est considéré comme pseudo-statique, l'équilibre du système est calculé en résolvant l'équation II-24. Les déplacements des nœuds des maillages EF sont obtenus en se servant de fonctions d'interpolation et de la loi de comportement du matériau.

La méthode développée par Gourret et al. permet de détecter et résoudre les contacts de manière exacte ce qui donne des déformations très réalistes physiquement et visuellement. Mais les inconvénients de cet algorithme sont un temps de calcul élevé et une convergence non garantie qui rendent l'algorithme non temps-réel.

Toutefois, il est important de noter que ce modèle est le seul modèle de préhension entre une main et un objet déformables utilisant la MEF.

Magenat-Thalmann et Thalmann [Magenat-Thalmann 90] ont amélioré l'utilisation de la méthode précédente en la couplant à la méthode JLD [Magenat-Thalmann 88]. Ils séparent la simulation des déformations en deux processus. Le premier met en œuvre la méthode JLD dans le cas de déformations dues aux mouvements du corps. Le second utilise la méthode des éléments finis et l'algorithme précédent au cours de tâches de préhension pour calculer les déformations résultant du contact entre la main et l'objet.

Cette approche permet d'optimiser le temps de calcul et de simuler des animations et des interactions avec la main virtuelle.

Les deux méthodes précédentes ont cherché à modéliser des tâches de préhension avec une main et un objet virtuels déformables. Pour simuler les interactions Homme-machine en temps réel, il est nécessaire de simplifier les modèles. En effet, l'utilisation de la capture de mouvements et/ou du retour d'effort exigent des temps de calcul de l'ordre de 10 ms. C'est avec l'objectif d'optimiser les temps de calcul que les modèles récents simulent la préhension d'objets déformables avec une main rigide ou celle d'objets rigides avec une main déformable.

Ainsi Hui et al. ont simulé la préhension d'une balle déformable avec une main rigide [Hui 02] en utilisant la méthode des éléments finis (Cf. Figure II.48). La main rigide articulée est pilotée par un gant de données. L'utilisateur peut ainsi saisir l'objet virtuel avec ses doigts. L'objet est modélisé par des éléments tétraédriques. Ils utilisent un modèle élastique linéaire pour calculer les déformations aux nœuds du maillage et ils font l'hypothèse des petites déformations. Les points et les actions de contact de la main sur l'objet sont donnés par une méthode de détection de collision utilisant une hiérarchie de volumes englobants. Pour optimiser le temps de calcul, ils utilisent la méthode de condensation pour inverser la matrice de raideur et la parallélisation.

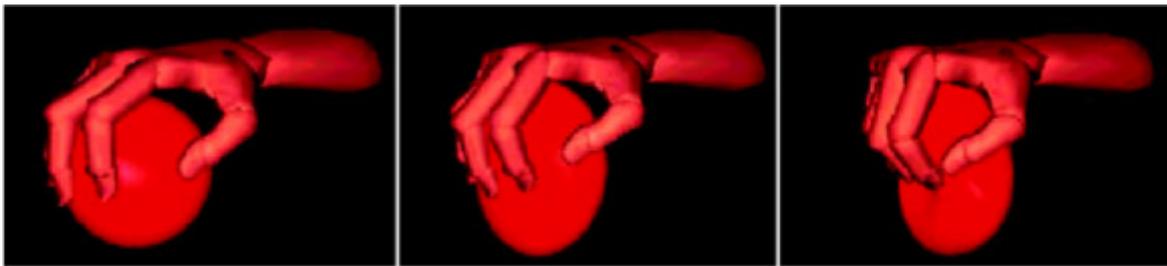


Figure II.48 : Saisie d'une balle déformable modélisée par des éléments finis

Cette approche permet de saisir des objets avec une main virtuelle interactivement en utilisant la capture de mouvements. Mais, les prises sont limitées et elles ne sont pas temps-réel.

Une méthode temps-réel a été développée par James et Pai pour simuler des objets déformables [James 01] et un doigt déformable [James 02] pour des applications avec de la capture de mouvements et du retour d'effort. Dans les deux cas de figure, ils utilisent des éléments élastiques linéaires pour la discrétisation. Les objets sont décrits par les valeurs limites aux nœuds. Calculer la matrice de raideur globale consiste à calculer les raideurs aux nœuds puis à obtenir la raideur de la surface de l'objet par une fonction d'interpolation.

Dans le cas particulier du doigt, celui-ci est décomposé en trois zones distinctes qui forment donc trois domaines linéaires élastiques avec des conditions de continuité aux jonctions (Cf. Figure II.49 et Figure II.49) :

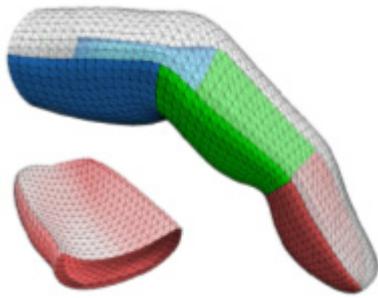


Figure II.49 : Modélisation d'un doigt par trois coussinets élasto-statiques individuels

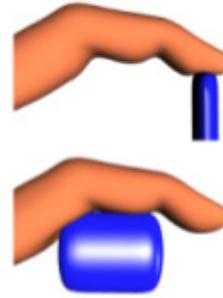


Figure II.50 : Simulation interactive de contact entre le doigt et des objets rigides

Les avantages de ce découpage sont la résolution de matrices de raideur plus petites car la condensation se fait sur chaque domaine ainsi que la possibilité de gérer des déformations non homogènes car chaque zone a son comportement propre. Mais l'inconvénient est qu'ils n'ont modélisé qu'un seul doigt ce qui ne permet pas la prise d'objets et surtout pose le problème du temps de calcul pour deux voire plusieurs doigts.

Nous présentons ci-dessous une autre utilisation de la MEF qui n'est pas dédiée à la préhension mais qui offre des solutions pour la modélisation et l'animation en temps-réel d'une main virtuelle déformable de manière physique.

Cette approche - appelée *Skeleton Driven Deformation* ou SSD - a été proposée par Capell et al. [Capell 02]. Ils présentent un environnement pour animer des personnages déformables en se basant sur les mouvements du squelette. Celui-ci est formé de segments rigides sur lesquels est fixé un maillage non régulier d'éléments finis volumiques. L'objet est divisé en plusieurs zones et les nœuds du maillage coïncident avec les segments rigides. Ils utilisent un modèle linéaire élastique avec l'hypothèse des petites déformations. Comme précédemment, les matrices de raideur sont calculées pour chaque zone ce qui optimise le temps de calcul.

Cette méthode permet de simuler de manière dynamique des personnages déformables articulés par un squelette sous-jacent en temps réel. Mais, l'hypothèse des petites déformations n'est pas toujours vérifiée en particulier dans le cas des mouvements de rotation.

2.2.4 Conclusion

Nous avons présenté dans les paragraphes précédents les différents modèles développés pour simuler le contact entre la main virtuelle et son environnement.

Ainsi les modèles cinématiques rigides couplés à une base de stratégies de saisie furent les premiers modèles à simuler la préhension d'objets rigides en animation graphique. Ces modèles ont été inspirés par ceux utilisés en Robotique.

Bien que ces modèles rigides soient capables de saisir et maintenir en position un objet, ils ne sont pas interactifs. Avec le développement des technologies de la Réalité Virtuelle, des mains virtuelles ont été développées afin d'interagir avec l'environnement ou avec un acteur virtuels en temps-réel. Ces modèles, rigides ou avec une géométrie déformable, sont pilotés par un système de capture de mouvements tels qu'un gant de données ou des marqueurs. Afin de connaître les actions de contact nécessaires à une simulation visuellement réaliste de la préhension, ils sont couplés à une méthode de détection de collisions.

Malgré leur possibilité d'interagir en temps réel avec l'environnement virtuel, ces modèles ne respectent pas la réalité dans la mesure où ils ne se déforment pas au cours du contact entre

l'environnement et la main virtuelle. Afin d'obtenir une préhension physiquement réaliste, il est nécessaire de déterminer les contacts exacts et les forces de réaction. Les modèles basés sur la physique permettent de calculer ces éléments pour évaluer les déformations réelles.

Ainsi, les modèles masses/ressorts sont très répandus en animation graphique car ils sont faciles à mettre en œuvre et rapides en temps de calcul. Mais la définition des raideurs de ressorts dans le cas d'objets complexes peut se révéler être longue et fastidieuse si le maillage est constitué d'un trop grand nombre de points et donc de ressorts.

C'est pourquoi la méthode des éléments finis est utilisée car elle permet d'obtenir un comportement du matériau homogène quelle que soit la forme de l'objet. Bien que physiquement réaliste, cette méthode est très coûteuse en temps de calcul : elle est donc limitée par la puissance de calcul des ordinateurs utilisés. En conséquence, il faut trouver la bonne combinaison des principes physiques, de la modélisation géométrique, des outils de calcul et des hypothèses simplificatrices pour permettre une simulation dynamique dans le domaine du temps réel. Plusieurs techniques ont été développées pour accélérer la résolution des équations telles que les pré-calculs, la méthode de condensation, la parallélisation ou encore la multirésolution.

2.3 Conclusion

Dans les paragraphes précédents, nous avons présenté les différents modèles de main existants. Ils ont été classés en deux catégories :

- Les modèles pour la communication qui permettent essentiellement de visualiser la main et de l'animer afin d'interagir de manière symbolique (gestes, signes,...) avec un système ou une personne.
- Les modèles pour la manipulation qui permettent d'interagir avec l'environnement de manière automatique ou physique.

Les premiers modèles mettent en œuvre des méthodes géométriques qui permettent de modéliser et d'animer des objets complexes mais qui n'autorisent que des déformations cinématiques et non dynamiques. Leur utilisation est limitée car il n'y a pas de réponse dynamique possible aux mouvements.

Or, l'objectif de ces travaux de recherche est de développer un modèle de main afin de simuler les interactions de l'opérateur avec son environnement de travail en utilisant les technologies de la Réalité Virtuelle.

Comme décrit dans le Chapitre I, notre main virtuelle devra respecter le modèle cinématique de la main humaine et il sera contrôlé par les données issues de la capture de mouvements. Au regard des modèles décrits précédemment, nous développons un modèle multicouches constitué :

- d'une chaîne arborescente polyarticulée rigide pour le squelette avec les contraintes biomécaniques statiques ;
- d'un modèle déformable pour les tissus mous afin de gérer les déformations dynamiques dues aux mouvements du squelette sous-jacent et les multiples contacts dus à l'interaction avec l'environnement;
- d'un maillage surfacique représentant la peau.

Afin d'obtenir un modèle physiquement réaliste, nous avons choisi de modéliser les tissus mous par un modèle éléments finis. En effet, les déformations obtenues par cette méthode sont les plus réalistes mais elles requièrent beaucoup de ressources en temps de calcul. D'après notre état de l'art, la plupart des modèles éléments finis décrits dans ce chapitre ont été simplifiés au niveau des lois de comportement, des zones de déformation, ou encore des

méthodes d'inversion de matrice afin d'optimiser le temps de calcul. Dans le cas de la simulation de la main, organe complexe, ces aspects sont particulièrement importants afin de satisfaire la contrainte temps-réel.

Nous avons décrit les modèles de mains cinématiques, rigides et déformables pour l'animation graphique et la simulation de tâches de saisie en environnement virtuel, en interactif ou non. Comme nous le verrons dans la suite de ce manuscrit, nous nous sommes inspirés de certains de ces modèles pour construire notre modèle de main virtuelle ou pour l'animer.

Dans le paragraphe suivant, nous nous intéressons plus particulièrement aux propriétés physiologiques de la main c'est-à-dire au comportement des coussinets et de la pulpe des doigts – situés du côté palmaire des phalanges - qui sont des acteurs importants dans la tâche de préhension. En effet, ils agissent comme un élément de couplage entre la main et les objets saisis. Nous présentons les modèles de pulpes de la main existants, appelés « modèles physiologiques ». Ces modèles nous permettront par la suite de définir les propriétés mécaniques à intégrer dans notre modèle éléments finis afin que les déformations simulées soient les plus réalistes possibles.

3 Les modèles physiologiques de la main

La main est le seul organe de préhension chez l'Homme : elle lui permet d'interagir avec son environnement. Comme décrit dans le Chapitre I au paragraphe 5, la peau et les tissus mous de la main possèdent de nombreuses propriétés mécaniques qui jouent un rôle important dans le contrôle des conditions de contact lors de l'exploration par le toucher ou la manipulation d'objets. Ils possèdent également de nombreux récepteurs pour percevoir les propriétés physiques (i.e. la forme et la texture) de l'environnement en contact.

Dans ce paragraphe, nous allons nous intéresser à la modélisation du comportement mécanique de la pulpe dans le but d'étudier la préhension.

3.1 Définitions

Dans des conditions réelles, i.e. in vivo, la pulpe du doigt répond à une indentation par une force qui augmente avec le déplacement. Dans ce cas de figure, la pulpe se comporte comme un ressort avec une grande rigidité. Lors du déchargement, la pulpe reprend sa forme initiale avec un temps de relaxation. Ces observations ont ainsi mis en évidence la nature visqueuse des coussinets. Il a donc été supposé que les modèles des extrémités des doigts contiennent des ressorts et des amortisseurs car ce sont des éléments qui produisent respectivement des forces en réponse aux déformations ainsi que des taux de déformation.

Sans réelles expérimentations, trois modèles de base ont été définis à partir de combinaisons de ressorts k et d'amortisseurs b (Cf. Figure II.51) [Gulati 97] :

- le modèle de Voigt,
- le modèle de Maxwell,
- le modèle de Kelvin.

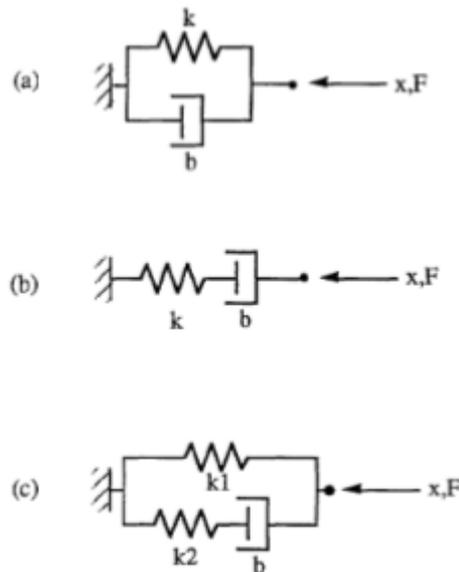


Figure II.51 : Les trois modèles de base

(a) Le modèle de Voigt – (b) Le modèle de Maxwell – (c) Le modèle de Kelvin

Pour chacun de ces modèles, l'équation différentielle qui relie la force F et le déplacement x est donnée par [Gulati 97] :

- dans le cas du modèle de Voigt :

$$F = kx + b\dot{x} \quad (\text{Eq. II-26})$$

- dans le cas du modèle de Maxwell :

$$\frac{\dot{F}}{k} + \frac{F}{b} = \dot{x} \quad (\text{Eq. II-27})$$

- dans le cas du modèle de Kelvin :

$$\frac{b}{k_2} \dot{F} + F = k_1 \cdot x + b \left(1 + \frac{k_1}{k_2} \right) \dot{x} \quad (\text{Eq. II-28})$$

Cependant, ces modèles ne prennent pas en compte les effets de l'inertie de la masse des tissus.

Ainsi, le modèle de Kelvin est considéré comme le modèle linéaire standard. Il traduit une relation générale qui tient compte de la charge, de la déformation et de leurs dérivées premières, d'où sa qualification de modèle « viscoélastique ». Ce modèle a été utilisé par Singh [Singh 97] afin de modéliser le comportement viscoélastique de la pulpe du doigt pour l'étude des caractéristiques thermiques de la peau.

Dans le paragraphe suivant, nous présentons les différents modèles développés pour étudier le comportement de la peau et des tissus mous des doigts sous l'action de forces ainsi que pour comprendre le fonctionnement des mécanorécepteurs.

3.2 Les modèles de pulpe existants

De nombreuses études ont été menées afin d'évaluer les déformations de la pulpe des doigts sous l'action des forces de contact. Elles ont permis d'identifier les lois de comportement

ainsi que les paramètres nécessaires à une simulation réaliste de la pulpe des doigts en contact avec l'environnement. Nous présentons les modèles développés et les résultats obtenus à l'issue de ces études.

Pour plus de clarté, nous avons classé les modèles en deux catégories :

- les modèles dits « uni-axiaux » qui prédisent la déformation de la pulpe des doigts suivant une direction ;
- les modèles dits « physiques » qui permettent d'étudier les déformations de la pulpe selon plusieurs directions.

3.2.1 Les modèles de déformation uni-axiaux

Ces modèles reposent sur une approximation géométrique de la pulpe des doigts. Comme les tissus mous sont constitués de 60%-70% d'eau, l'extrémité du doigt est modélisée par une membrane élastique contenant un fluide incompressible. En mettant en œuvre la théorie des membranes élastiques et celle des déformations finies, il est alors possible de simuler les déformations de la pulpe sous l'action des forces extérieures.

Ainsi, Srinivasan a développé un modèle 2D du comportement mécanique des tissus de la pulpe du doigt appelé le modèle *Waterbed* [Srinivasan 89]. La peau a été modélisée par une membrane élastique contenant un fluide incompressible représentant les tissus mous (Cf. Figure II.52) :

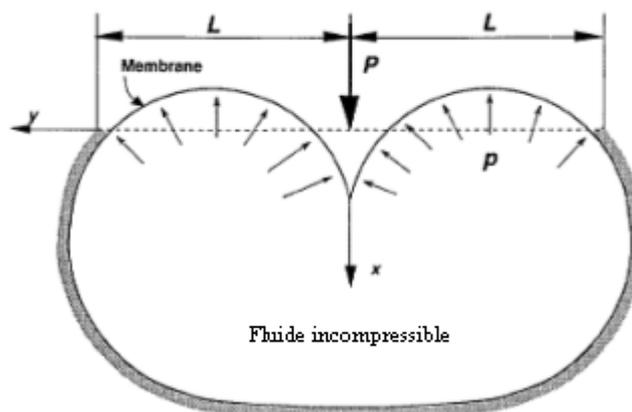


Figure II.52 : Le modèle *Waterbed* pour la pulpe d'un primate

----- : Etat de repos de la membrane ; _____ : Etat déformé sous l'action d'une charge linéique

En raison de la tension uniforme de la membrane et du champ de pression uniforme à l'intérieur du fluide, ce modèle a donné des profils très précis de la déformation de la surface de la peau sous l'action de chargements linéiques. Ces données ont été vérifiées par des expériences sur des doigts de singes et des doigts d'humains.

Mais, ce modèle ne permet pas de prédire de façon précise la relation contrainte/déformation à l'intérieur du doigt (i.e. pour les tissus mous) en raison des hypothèses posées (i.e. tension uniforme et pression uniforme).

Serina et al. ont amélioré le modèle de Srinivasan décrit précédemment en proposant un modèle géométrique 3D de la pulpe du doigt [Serina 98]. Celle-ci est modélisée par une membrane ellipsoïdale axisymétrique (qui représente la peau) contenant un fluide non

visqueux incompressible (qui représente les tissus mous). La membrane est donc gonflée par la pression interne du fluide (Cf. Figure II.53) :

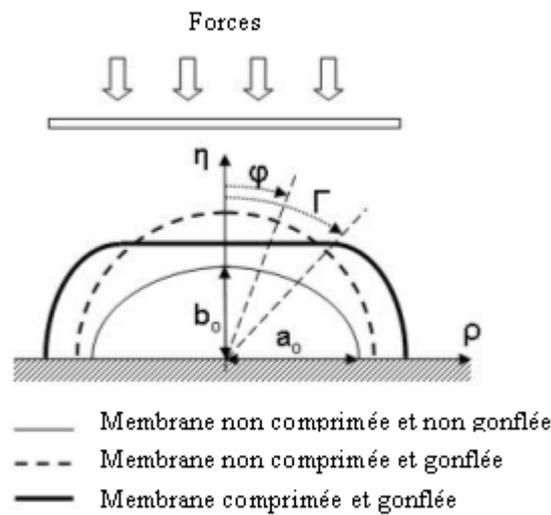


Figure II.53 : Membrane remplie d'un fluide incompressible

La membrane est déformée sous l'action de deux plans parallèles infiniment rigides en compression quasi-statique. La relation force/déplacement de la surface de la peau obtenue par ce modèle est montrée par la Figure II.54 :

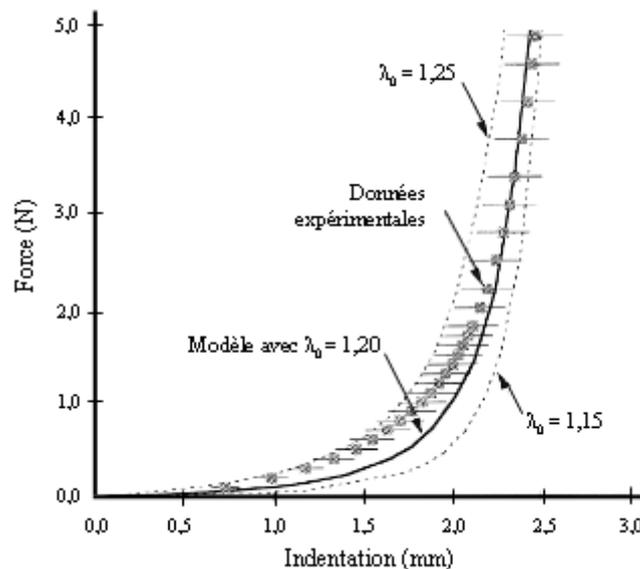


Figure II.54 : Relation force/déplacement obtenue avec le modèle de Serina et al.

Les prédictions des déformations à la surface de la zone de contact se sont révélées être proches des mesures expérimentales réalisées lors d'un contact de la pulpe avec une surface plane rigide. De plus, Serina et al. ont montré que le déplacement et la zone de contact augmentaient rapidement avant que la charge n'atteigne 1 N (magnitude de force faibles) et que la raideur augmente de manière non-linéaire avec le déplacement.

Ce modèle structurel prend en compte la non homogénéité du matériau ainsi que la géométrie de la pulpe du doigt. Mais, il est limité par les hypothèses simplificatrices (fluide uniforme) et l'angle d'incidence des forces extérieures qui est fixé à 45° . Si cet angle n'est pas respecté, les zones de contact ne sont plus circulaires et la contrainte axisymétrique n'est plus respectée. Dans ce cas de figure, les équations ne sont plus valables.

Ainsi, les modèles précédents ne tiennent pas compte des propriétés mécaniques de la pulpe et ils ne permettent pas de connaître les contraintes et les déformations dans les couches sous-cutanées. De plus, ils ne simulent que des réponses en force en statique. Afin d'autoriser des simulations plus réalistes et surtout d'estimer la propagation des efforts dans la pulpe des doigts, des modèles physiques ont été développés.

3.2.2 Les modèles de déformation physiques

De nombreux travaux de recherche sur les tissus mous ont conclu que les tissus biologiques ne sont pas élastiques. De plus, dans la vie réelle les directions des sollicitations ne sont pas contrôlables. Ces observations montrent qu'il est donc nécessaire d'utiliser des modèles basés sur la physique pour simuler les parties déformables des doigts de la main. Afin d'intégrer les lois de comportement des matériaux biologiques dans la modélisation des coussinets et de la pulpe, deux approches ont été développées :

- les modèles composés de ressorts et d'amortisseurs,
- les modèles continus.

Nous décrivons dans la suite les modèles développés dans chaque catégorie.

3.2.2.1 Les modèles ressorts/amortisseurs

Ces modèles sont basés sur les modèles de Voigt, Maxwell et Kelvin décrits dans le paragraphe 3.1. Ils permettent de prendre en compte les propriétés mécaniques de la peau et de la pulpe, et plus particulièrement leur nature visqueuse en intégrant des ressorts et des amortisseurs comme le montre la Figure II.55 [Gulati 97] :

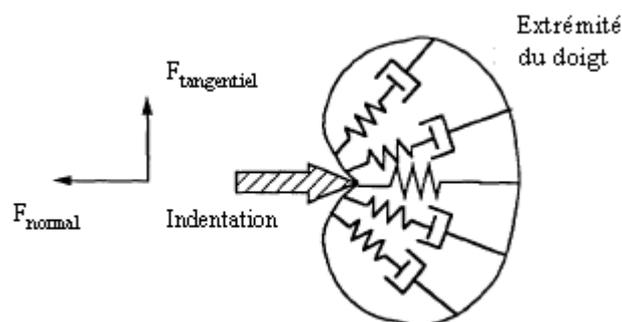


Figure II.55 : Principe des modèles ressorts/amortisseurs

Ainsi, Fung a montré que le modèle de Kelvin est le modèle de base qui traduit le mieux les propriétés mécaniques des tissus vivants [Fung 93]. Suite à ses expérimentations, il a développé un modèle de la pulpe du doigt qui consiste en deux composantes :

- une réponse élastique qui est la réponse instantanée du doigt à une consigne en force ;

- une fonction de relaxation réduite qui est la réponse normée en fonction du temps du doigt à une consigne en position.

Ce modèle viscoélastique quasi-linéaire est à la base de nombreuses recherches et de nombreux modèles.

Gulati et Srinivasan ont amélioré le modèle de Kelvin pour mieux représenter le comportement des tissus biologiques [Gulati 97]. En mesurant les réponses temporelles en force de la pulpe des doigts sous l'action d'excitateurs de différentes formes (ponctuel, plan et circulaire plan), ils ont développé un modèle de type Kelvin avec un ressort non-linéaire (Cf. Figure II.56) :

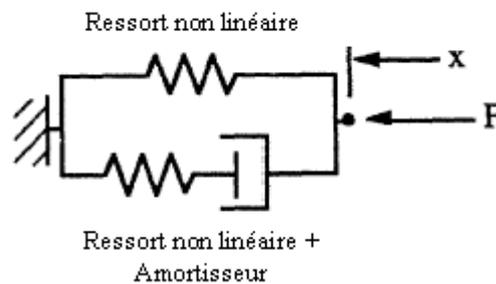


Figure II.56 : Le modèle non linéaire de Gulati et Srinivasan

Ce modèle a permis d'expliquer les réponses en force viscoélastiques non-linéaires de la pulpe observées au cours de leurs expérimentations. Bien qu'il simule de manière réaliste le comportement global de l'extrémité du doigt, il ne permet pas de prendre en compte la variation spatiale des contraintes ou celle des déformations à l'intérieur du doigt, ni d'obtenir les profils de déformation de la surface de la peau.

De la même manière, le modèle développé par Pawluk tient compte des caractéristiques non linéaires de la relation contrainte/déformation des tissus biologiques [Pawluk 97]. Ainsi, elle a proposé un modèle visco-élastique quasi-linéaire basé sur ceux de Kelvin et Fung. Le modèle global est composé d'une séquence de plusieurs ressorts de différentes longueurs, avec un nombre de ressorts qui augmente lorsque la déformation augmente. Celui-ci a permis de simuler la réponse de la pulpe du doigt à une déformation mécanique en mettant en évidence une relation non-linéaire, de type exponentiel, entre la force et la raideur.

Pawluk et Howe ont également étudié l'interaction mécanique dynamique entre la pulpe et un plan [Pawluk 99a]. La Figure II.57 montre la relation force/déplacement non linéaire obtenue à l'issue de leurs expérimentations :

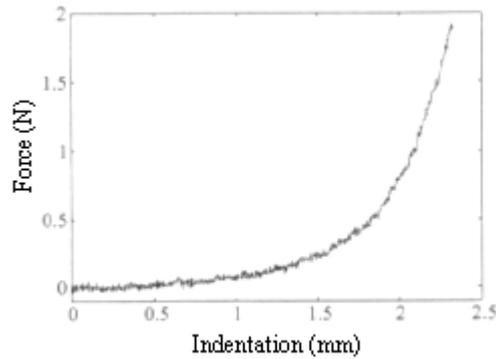


Figure II.57 : Relation force totale/déplacement obtenue avec le modèle de Pawluk

En appliquant des trajectoires contrôlées en position à l'extrémité du doigt, ils ont également obtenu la distribution des forces de pression sur la pulpe [Pawluk 99b]. La Figure II.58 montre les relations pression/déplacement de la pulpe obtenue à différentes localisations du point de mesure par rapport au point de stimulation :

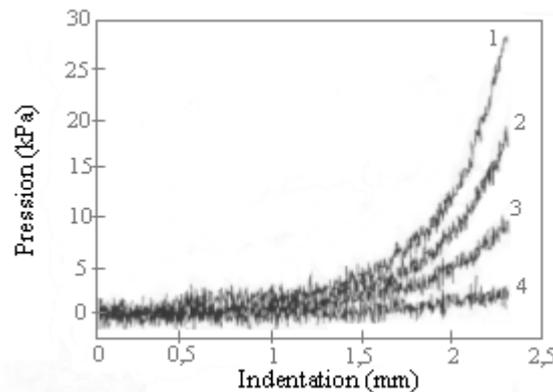


Figure II.58 : Relations pression/déplacement estimées avec le modèle de Pawluk

1 : mesure la plus proche de la stimulation, 4 : la plus éloignée.
Les réponses en force diminuent quand la distance au contact augmente.

A partir de leurs mesures, ils ont défini un modèle basé sur le modèle de contact de Hertz. Ainsi, le modèle développé par Pawluk et Howe permet de prédire les profils de la déformation globale de la pulpe ainsi que la distribution spatiale des forces de pression lors d'un contact en réponse à des forces dynamiques.

Ce modèle a été mis en œuvre par Jindrich et al. afin de caractériser les chargements mécaniques et les déplacements de la pulpe au cours de tâches dynamiques comme la dactylographie [Jindrich 03]. Ainsi, ils ont pu évaluer l'aptitude du modèle viscoélastique non linéaire à décrire les relations force/déplacement de la pulpe sollicitée en compression verticale. Jindrich et al. ont montré que ce modèle permet de caractériser la réponse en force au cours des impacts en tant que fonction dépendant du déplacement de la pulpe et du temps.

Bien que les modèles ressorts/amortisseurs décrits précédemment simulent de manière réaliste le comportement global de l'extrémité du doigt, ils ne permettent pas de prendre en compte la variation spatiale des contraintes ou des déformations des différentes couches du doigt. C'est pour remédier à ce problème que des modèles continus ont été définis.

3.2.2.2 Les modèles éléments finis

L'un des premiers modèles mécaniques continus de la peau humaine a été proposé par Phillips et Johnson [Phillips 81]. Ils ont représenté l'extrémité du doigt comme un espace semi-infini homogène, isotrope, incompressible et élastique linéaire avec l'hypothèse d'un état de contrainte plane ou de petites déformations. Ce modèle analytique leur a permis d'étudier les relations entre les mesures de déformations à la surface de la peau et les enregistrements des réponses neuronales des mécanorécepteurs.

Bien que ce type de modèles puisse prédire les réponses neuronales de manière approximative, il ne tient pas compte de la forme géométrique (courbure) ni de la structure non-homogène de la pulpe du doigt. De plus, ce modèle n'est pas capable de prendre en compte les modifications de la surface de contact ou le comportement non-linéaire de la pulpe. En conséquence, les réponses en force obtenues par un tel modèle sont différentes de celles mesurées sur la peau des doigts.

Afin d'être plus réaliste du point de vue physique, Dandekar et Srinivasan ont développé des modèles éléments finis 2D [Dandekar 97] et 3D [Dandekar 03] des extrémités des doigts chez l'Homme et chez le singe.

L'extrémité du doigt a été modélisée en 2D comme un matériau homogène, isotrope, linéaire élastique comme l'illustre la Figure II.59 [Dandekar 97] :

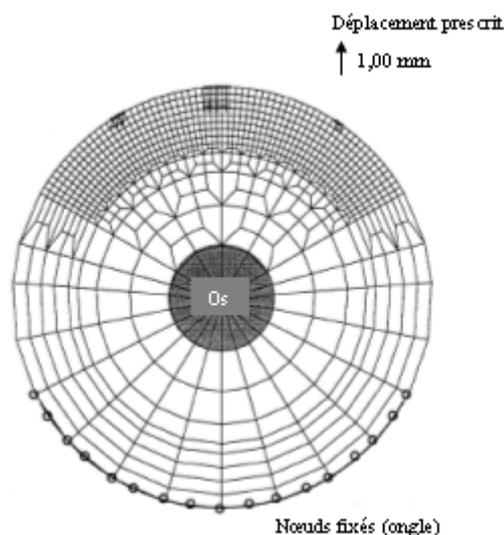


Figure II.59 : Modèle éléments finis 2D élastique linéaire de la pulpe de Dandekar

En généralisant au cas 3D, ils ont défini un modèle éléments finis multi-couches constitué de la peau, des tissus mous sous-cutanés et de l'os (Cf. Figure II.60). Lorsque ce modèle est déformé par un déplacement constant, on obtient les déformations de la pulpe suivantes (Figure II.61) [Dandekar 03] :

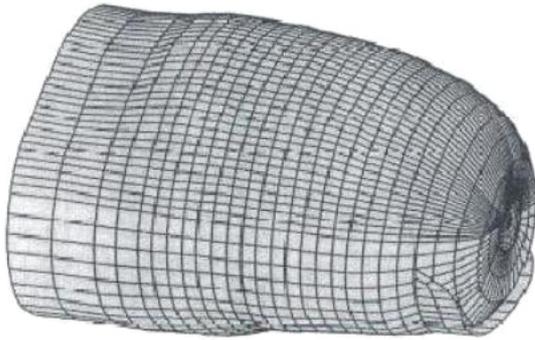


Figure II.60 : Modèle éléments finis 3D élastique linéaire de la pulpe de Dandekar

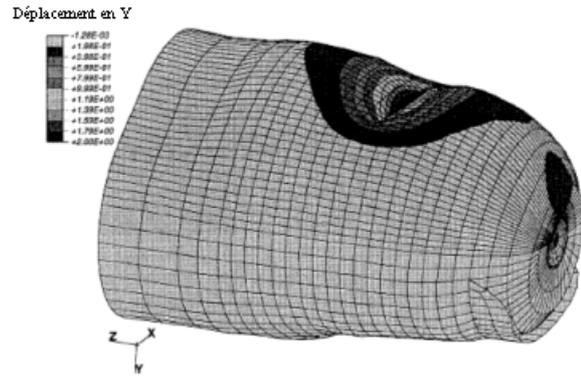


Figure II.61 : Modèle éléments finis 3D élastique linéaire déformé de Dandekar

Le modèle continu proposé par Dandekar et Srinivasan prédit les distributions de contraintes et de déformations à l'intérieur de l'extrémité du doigt, i.e. au niveau des différentes couches de la pulpe. Mais, la relation force/déformation et le profil de déformation calculés à la surface de la peau avec ce modèle ne correspondent pas aux données expérimentales : les lois de comportement mises en œuvre ne sont donc pas correctes. Bien que ce modèle soit réaliste géométriquement, il est nécessaire de connaître les propriétés des matériaux (module d'Young et coefficient de Poisson) pour simuler un modèle avec la méthode des éléments finis qui soit réaliste.

Un autre modèle éléments finis 2D a été proposé récemment par Wu et al. pour étudier le comportement de l'extrémité des doigts sous l'action de vibrations libres [Wu 02] [Wu 04]. Le modèle se compose d'un ongle et d'un os élastiques linéaires et d'une couche intermédiaire visco-élastique non linéaire qui représente la peau et les tissus mous sous-jacents. La section de l'extrémité du doigt est considérée comme elliptique comme le montre la Figure II.62 [Wu 02] [Wu 04] :

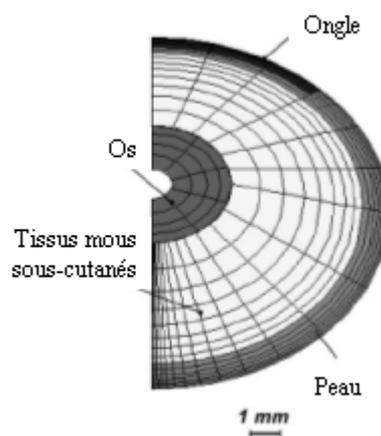


Figure II.62 : Modèle EF 2D non-linéaire viscoélastique de la pulpe de Wu et al.

En excitant ce modèle par un plan qui vibre, i.e. en compression, Wu et al. ont obtenu les différents modes de déformation de la pulpe du doigt. La Figure II.63 montre le premier mode de déformation ainsi que la distribution spatiale de la pression quand le modèle EF 2D est

stimulé par une pointe qui se déplace verticalement. Ils ont également calculé la relation force/déplacement en fonction des différents modes de déformation (Cf. Figure II.64) [Wu 02] [Wu 04] :

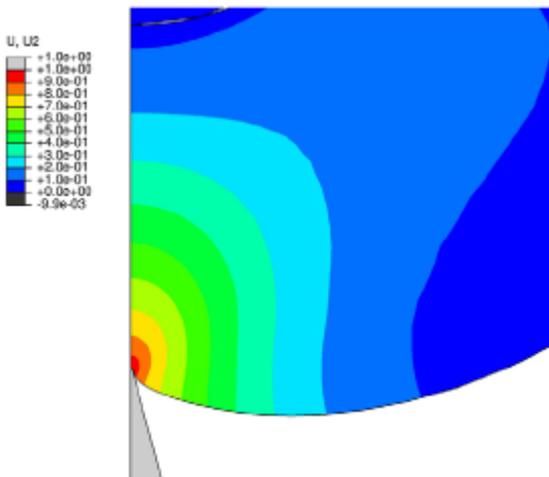


Figure II.63 : Premier mode de déformation à $t=1s$

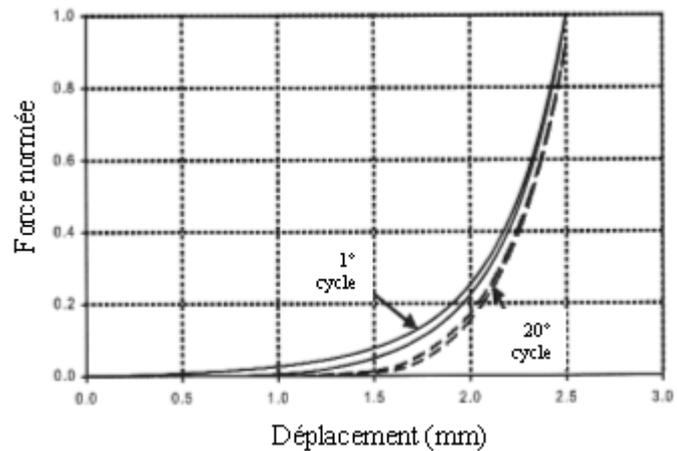


Figure II.64 : Relation force/déplacement en fonction des modes de vibration

Le modèle EF 2D développé par Wu et al. tient compte de la structure anatomique et des propriétés mécaniques des tissus mous (viscoélasticité et élasticité non linéaire). Il permet de prédire les caractéristiques modales sous différentes conditions de déformation ou de pré-chargement. Les résultats simulés ont été comparés à des résultats de mesures et ils se sont révélés être proches de la réalité.

Bien que ce soit l'un des modèles les plus réalistes physiquement parlant, il ne modélise la pulpe du doigt et ses déformations que dans un plan.

3.3 Conclusion

Nous avons décrit dans les paragraphes précédents les différents modèles d'extrémité des doigts développés pour simuler les déformations subies par les coussinets et la pulpe lorsqu'ils sont sollicités par des actions de contact. De tels modèles cherchent essentiellement à atteindre deux objectifs :

- connaître le comportement des pulpes des doigts en identifiant les relations force/déplacement, force/zone de contact ou encore la distribution de pression à la surface de la pulpe. Toutes ces courbes sont destinées à comprendre la « mécanique du toucher »;
- connaître les réponses des mécanorécepteurs aux forces de contact pour identifier l'information neuronale à partir des relations contrainte/déformation obtenues à la surface de la peau et dans les tissus mous sous-jacents.

Bien que les modèles géométriques donnent des profils de réponse en force réalistes à la surface de la peau, ils ne tiennent pas compte des propriétés mécaniques de la main humaine. Or, les chargements ou les distributions de pression le long de la surface palmaire créent une contrainte qui varie en fonction du temps et elles génèrent des densités de déformation à l'intérieur de la peau et des couches sous-cutanées. Il est donc essentiel d'intégrer les lois de comportement des matériaux biologiques dans un modèle de pulpe.

Afin d'obtenir une déformation surfacique et volumique des doigts physiquement réalistes, il est nécessaire de déterminer les propriétés mécaniques de chaque couche du doigt, et plus

particulièrement de la pulpe. Des modèles basés sur la physique ont été définis pour répondre à ce besoin.

Ainsi, les modèles ressorts/amortisseurs ont été développés afin de simuler la viscoélasticité du doigt. Mais, ils ne permettent de donner qu'une déformation globale de la pulpe sans prendre en compte la variation spatiale des contraintes ou des déformations des couches internes du doigt.

C'est pourquoi la méthode des éléments finis a été mise en oeuvre. En effet, elle permet de modéliser l'extrémité du doigt comme un milieu continu afin d'obtenir un comportement homogène du matériau sous des sollicitations extérieures. Il est donc possible de modéliser des lois de comportement complexes telles que l'hyper élasticité ou la viscoélasticité des tissus mous. Seulement, cette modélisation réaliste est coûteuse en temps de calcul.

Il faut donc résoudre deux types de déformations : d'une part celles dues à la cinématique du modèle (comme par exemple aux articulations), et d'autre part celles dues aux contacts simultanés lors de la saisie de l'objet. La méthode des éléments finis semble être la mieux adaptée à notre problématique. Mais, le choix de la loi de comportement du matériau biologique est primordial pour permettre une simulation dynamique dans le domaine du temps réel.

4 Définition de notre modèle

L'objectif de ces travaux de recherche est de développer un modèle de main afin de simuler les interactions Homme-machine en environnement virtuel.

La description de la main humaine faite dans le Chapitre I a mis en évidence la complexité de cet organe. Elle nous a également conduit à définir les contraintes principales que doit respecter notre modèle.

Ainsi, notre main virtuelle devra respecter le modèle cinématique de la main humaine et il sera contrôlé par les données issues de la capture de mouvements. Au regard des modèles décrits dans le paragraphe 2, nous développons un modèle multicouches constitué :

- d'une chaîne arborescente polyarticulée rigide pour le squelette avec les contraintes biomécaniques statiques ;
- d'un modèle déformable pour les tissus mous afin de gérer les déformations dynamiques dues aux mouvements du squelette sous-jacent et aux contacts multiples inhérents à la manipulation ;
- d'un maillage surfacique représentant la peau.

Afin d'évaluer les déformations réelles à la surface de la peau et à l'intérieur du doigt, nous avons choisi de modéliser les tissus mous par un modèle élément fini. En effet, un tel modèle permet d'étudier :

- la mécanique tactile entre l'objet et la surface de la pulpe ;
- les réponses statiques et dynamiques des tissus mous sous l'action de mouvements statiques et cycliques ;
- les déformations et les champs de contraintes et de déformations à l'intérieur des tissus mous.

Les modèles physiques de l'extrémité du doigt détaillés dans le paragraphe 3.2.2 ont mis en évidence le comportement viscoélastique non linéaire de ce matériau biologique. Mais, dans notre cas de figure, en raison de la complexité technique à modéliser les tissus mous en chargement dynamique et surtout de la contrainte temps-réel, les coussinets et la pulpe des doigts seront considérés comme élastiques, linéaires et isotropes.

Le terme « élastique » indique que l'état des déformations ne dépend que de l'état présent des contraintes. Cela signifie que les forces internes au matériau n'agissent que par contact immédiat et que leur rayon d'action est proche de zéro. Ainsi, un matériau élastique qui a été déformé sous l'action de certaines forces regagne son état initial une fois que ces forces ont disparu. Il peut donc restituer toute l'énergie qu'il a absorbée et par conséquent sa forme ne dépend pas de l'historique de ses déformations.

Le terme « linéaire » indique que la relation entre contraintes et déformations est du premier ordre.

Le terme « isotrope » signifie que les propriétés du matériau utilisé sont les mêmes dans toutes les directions. Cette condition introduit un certain nombre de liens et donc de simplifications entre les contraintes et les déformations. La relation du premier ordre entre les contraintes et les déformations se réduit à deux paramètres indépendants : les coefficients de Lamé λ et μ (Cf. Chapitre IV).

Bien que la loi de comportement que nous intégrons soit une simplification de celle issue des différentes expériences réalisées pour identifier les modèles décrits précédemment, elle correspond au comportement d'un grand nombre de matériaux réels soumis à de faibles déformations. De plus, les simulations devraient être réalistes du point de vue physique car la distribution spatio-temporelle des chargements mécaniques sur la peau à l'interface de contact et la distribution des contraintes et des déformations à l'intérieur de la pulpe seront homogènes. Nous reviendrons plus en détails sur le modèle éléments finis dans le Chapitre IV.

Nos recherches bibliographiques nous ont permis de définir les caractéristiques de notre modèle de main ainsi que les propriétés physiques à intégrer.

Dans le Chapitre III, nous décrivons le squelette du modèle de main ainsi que les méthodes mises en œuvre afin de le contrôler en position et en orientation en temps-réel dans un environnement virtuel contraint.

Puis, nous détaillons la modélisation des tissus mous par la méthode des éléments finis dans le Chapitre IV.

Enfin, nous présentons les interactions Homme-machine simulées en environnement virtuel dans le cas de la prévention des risques d'accidents dans le Chapitre V.

CHAPITRE III : Pilotage de la main en interactif

Dans les chapitres précédents, nous avons décrit la main humaine et nous avons présenté les modèles de mains existants qui ont été développés dans différents domaines de recherche. Afin de réduire la complexité de cet organe de préhension pour des applications temps-réel, tout en conservant un comportement réaliste, nous avons choisi de développer un modèle multicouches. Celui-ci comporte trois couches principales :

- une chaîne arborescente polyarticulée rigide pour le squelette et ses contraintes biomécaniques statiques ;
- un modèle déformable pour les tissus mous, afin de gérer les déformations dynamiques dues aux mouvements du squelette sous-jacent et aux contacts multiples inhérents à la manipulation ;
- un maillage surfacique représentant la peau.

Dans ce chapitre, nous nous intéressons à la première couche de notre main virtuelle, c'est-à-dire au modèle cinématique et plus particulièrement à son animation en temps-réel.

1 Introduction

Dans notre cas de figure, nous avons choisi de contrôler la main en utilisant les données issues de la capture de mouvements. Ainsi, la position de la main est donnée par le biais de cibles optiques pour la position et l'orientation du poignet et par des gants de données pour les positions des doigts. Avec cette méthode, les mouvements de la main humaine sont mesurés en temps-réel et naturellement reproduits par la main virtuelle.

Tout d'abord, nous présentons le modèle squelettique de main proposé pour les applications en Réalité Virtuelle. Ce modèle tient compte des caractéristiques cinématiques et biomécaniques de la main humaine présentées dans le Chapitre I.

Ensuite, nous décrivons les algorithmes développés pour piloter la main en temps-réel dans l'environnement virtuel. Nous détaillerons ainsi les asservissements développés pour les mouvements du poignet et des doigts et nous présenterons les deux méthodes de résolution utilisées : une méthode du premier ordre (Cf. paragraphe 5) puis une méthode du second ordre (Cf. paragraphe 6).

2 Le modèle squelettique proposé

En se référant à la description anatomique et fonctionnelle de la main humaine donnée au Chapitre I (Cf. [Tubiana 80] [Castaing 99a] [Rouvière 02] [Kapandji 05]) ainsi qu'aux modèles de mains squelettiques existants décrits au Chapitre II (Cf. les modèles de Rehg et Kanade [Rehg 93], de Mas Sanso [Mas Sanso 94] ou encore de Ouhaddi et Horain [Ouhaddi 98]), nous avons défini le squelette de notre main virtuelle par une structure polyarticulée rigide. Celle-ci se compose de quatre doigts longs et d'un pouce en opposition. Chaque doigt est une chaîne arborescente qui possède quatre degrés de liberté (Cf. Figure III.65) :

- trois degrés de liberté pour le mouvement de flexion/extension pour les trois articulations des doigts (MCP, IPP et IPD),
- un quatrième degré de liberté pour le mouvement d'abduction/adduction pour les articulations situées à la base des doigts (MCP).

Le pouce a la chaîne cinématique suivante :

- trois degrés de liberté pour le mouvement de flexion/extension pour les articulations situées au niveau du métacarpe (CMC) et des phalanges (MCP et IP),
- deux degrés de liberté pour le mouvement d'abduction/adduction pour les articulations au niveau du métacarpe (CMC) et de la première phalange (MCP).

Chacun de ces degrés de liberté est modélisé par une liaison parfaite rotoïde (sans frottement).

Afin de simplifier le modèle squelettique, nous éliminons des degrés de liberté au niveau des os du carpe car ces segments osseux n'ont pas une très grande amplitude de mouvements (Cf. Chapitre I, paragraphe 4). Par conséquent, les os du carpe ont été associés et les métacarpiens des quatre doigts (index à auriculaire) ont été considérés comme rigidement liés à ces segments osseux. Seul le métacarpe du pouce conserve ses deux degrés de liberté au niveau du poignet car son amplitude de mouvement d'abduction/adduction est beaucoup plus importante et elle lui permet son opposition avec les autres doigts lors des tâches manuelles. Nous éliminons également le degré de liberté de la première phalange (MCP) du pouce car le mouvement d'abduction/adduction a une amplitude très faible.

De plus, nous faisons l'hypothèse que la paume est toujours dans un plan dans les mouvements que nous simulons. Nous modélisons donc la paume, les os du carpe, les métacarpes et les articulations situées au niveau du poignet (CMC) de la main humaine par un unique parallépipède rectangle. Les mouvements du poignet correspondront donc aux six degrés de liberté de la paume virtuelle.

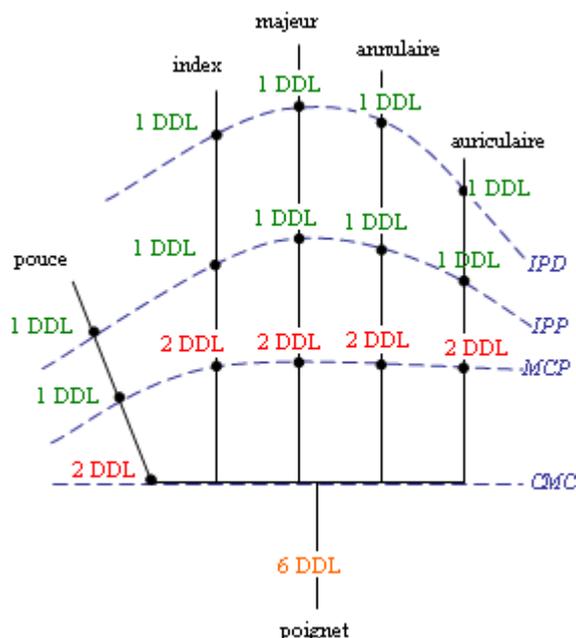


Figure III.65 : Le modèle cinématique de la main virtuelle

Au final, nous avons retenu un modèle avec vingt degrés de liberté pour les doigts et six degrés de liberté pour le poignet. Les débattements angulaires des phalanges de notre modèle sont donnés par le tableau suivant (Cf. Tableau III.2) :

	CMC f/e	CMC a/a	MCP f/e	MCP a/a	IPP f/e	IPD f/e
Auriculaire			[0°; 90°]	[-15°;15°]	[0°;110°-130°]	[0°;65°-90°]
Annulaire			[0°; 90°]	[-15°;15°]	[0°;110°-130°]	[0°;65°-90°]
Majeur			[0°; 90°]	0° (axe)	[0°;110°-130°]	[0°;65°-90°]
Index			[0°; 90°]	[-30°;30°]	[0°;110°-130°]	[0°;65°-90°]
Pouce	[-25°;120°]	[0°;35°-40°]	[10°;90°]		-	[0°;65°-90°]

f/e : flexion / extension

a/a : adduction/abduction

Tableau III.2 : Débattements angulaires de notre modèle squelettique

Des simplifications au niveau des formes géométriques ont été réalisées pour le squelette car il ne sera pas visible. Les doigts de la main sont donc modélisés par des cylindres et des sphères et par un parallélépipède pour la paume et le poignet. Ces primitives simples ont également l'avantage de faciliter la détection de collision (Cf. Chapitre IV).

Le choix des dimensions de la main (longueur et diamètre) respecte les longueurs moyennes des différents os de la main humaine. Nous intégrons les valeurs obtenues à partir des données anthropométriques pour un majeur de 12 cm (Cf. Annexe A.1). Nous obtenons les mesures suivantes (Cf. Tableau III.3) :

Doigts	Longueur totale (mm)	Longueur de la phalange (mm)	Longueur de la phalangine (mm)	Longueur de la phalangette (mm)
Pouce	74	52	-	22
Majeur	120	58,7	37,5	23,8
Index	107	54	31,5	21,4
Annulaire	113,8	53,8	36,4	23,6
Auriculaire	91,3	45	25,8	20,5

Tableau III.3 : Longueurs des phalanges pour un majeur de 12 cm

Le modèle de main virtuelle est développé sous un logiciel de simulation physique (Cf. Figure III.66).

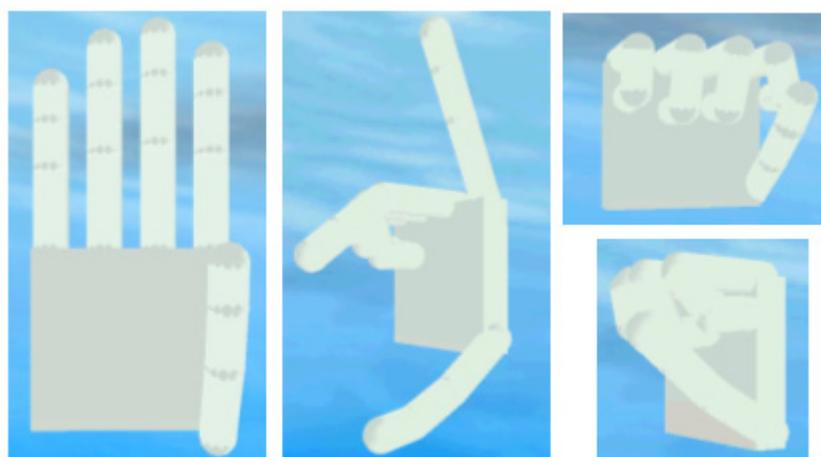


Figure III.66 : La main virtuelle

Les données sont enregistrées dans un fichier XML qui permet ainsi de modifier rapidement la taille des phalanges ou encore les limites articulaires. Tous ces paramètres sont donc accessibles à l'utilisateur et modifiables.

Après la description de la main virtuelle développée pour nos applications, nous présentons le modèle géométrique dans le paragraphe suivant. Celui-ci est utilisé pour piloter la main virtuelle à partir des données issues de la capture de mouvements.

3 Modélisation géométrique et cinématique

Dans ce paragraphe, nous décrivons le modèle géométrique et le modèle cinématique de la main. Nous commençons par donner les définitions et les notations utilisées pour décrire le modèle géométrique. Ensuite, nous détaillons les repères utilisés dans cette structure polyarticulée. Puis, à partir de ces paramètres, nous décrivons les mouvements réalisés par le modèle de main dans l'environnement virtuel. Enfin, nous détaillons les équations de la cinématique de notre main virtuelle.

3.1 Définitions et notations

Nous nous basons sur la théorie des groupes de Lie car elle se révèle être bien adaptée à l'étude des mécanismes rigides. Dans la suite de ce chapitre, nous utilisons les éléments du groupe spécial euclidien $SE(3)$ pour représenter les matrices de notre main virtuelle [Stramigioli 01a] [Stramigioli 01b] [Ravani 02].

Nous utilisons donc les notations suivantes :

- $p : [x \ y \ z]^T$: un vecteur avec $p \in \mathbb{R}^3$
- P : un point de coordonnées $[p \ 1]^T$
- R : matrice de rotation 3×3
- $SO(3)$: groupe spécial orthonormal, groupe de Lie : c'est l'espace des rotations
On a : $SO(3) = \left\{ R \in \mathbb{R}^{3 \times 3} ; R^{-1} = R^T, \det R = 1 \right\}$
- $so(3)$: espace des vitesses angulaires
- $SE(3)$: groupe spécial euclidien, groupe de Lie : c'est l'espace des matrices
On a : $SE(3) = \left\{ \begin{pmatrix} R & p \\ 0 & 1 \end{pmatrix} ; R \in SO(3), p \in \mathbb{R}^3 \right\}$

Un espace Euclidien est un ensemble continu de points qui a une structure supplémentaire qui permet de décrire l'orthogonalité et de mesurer les distances : le produit scalaire.

- $se(3)$: espace des vitesses linéaires

$$\text{On a : } \mathfrak{se}(3) = \left\{ \begin{pmatrix} \omega & v \\ 0 & 0 \end{pmatrix}; \omega \in \mathfrak{so}(3), v \in \mathbb{R}^3 \right\}$$

- $\mathfrak{se}^*(3)$: espace dual de $\mathfrak{se}(3)$. C'est l'espace des torseurs d'efforts et l'espace vectoriel des opérateurs linéaires de $\mathfrak{se}(3)$.
- \mathfrak{R}_i : repère orthonormé direct lié au corps i
- $H_i^j \in \mathbb{R}^{4 \times 4}$: matrice de transformation en coordonnées homogènes pour passer du repère \mathfrak{R}_i au repère \mathfrak{R}_j
- $T_i^j \in \mathfrak{se}(3)$: torseur cinématique représentant le mouvement de \mathfrak{R}_i au repère \mathfrak{R}_j et exprimé dans \mathfrak{R}_j
- $T_i^{k,j}$: vecteur 6×1 projeté dans \mathfrak{R}_k représentant le torseur cinématique qui permet de passer du repère \mathfrak{R}_i au repère \mathfrak{R}_j
- $W_i = [\tau_i \quad f_i]^T$: torseur des efforts appliqués à l'objet rigide dans le repère \mathfrak{R}_i . τ_i représente un couple et f_i une force linéaire.
- W_i^j : vecteur 6×1 projeté dans \mathfrak{R}_j représentant le torseur d'effort.
On note ${}^K W_i^j$ et ${}^B W_i^j$ les torseurs d'efforts appliqués respectivement à un ressort et à un amortisseur qui connectent le repère \mathfrak{R}_i au repère \mathfrak{R}_j .

3.1.1 Matrice de position d'un corps rigide

La configuration d'un corps rigide peut être représentée par une matrice de transformation homogène ou par une position et un quaternion :

$$H_i^j = \begin{bmatrix} R_i^j & p_i^j \\ 0_3^T & 1 \end{bmatrix} \quad \text{ou bien} \quad H(p_i^j, Q_i^j)$$

avec : $H_i^j \in \text{SE}(3)$, $R_i^j \in \text{SO}(3)$ et $p_i^j \in \mathbb{R}^3$

- Changement de coordonnées d'un point P du repère \mathfrak{R}_1 au \mathfrak{R}_2 :

$$\begin{bmatrix} p^2 \\ 1 \end{bmatrix} = \begin{bmatrix} R_1^2 & p_1^2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p^1 \\ 1 \end{bmatrix} \quad \text{ou} \quad P^2 = H_1^2 P^1$$

3.1.2 Torseur cinématique

Le torseur cinématique représente la vitesse d'un corps rigide. Il peut être décrit sous deux formes :

$$T = \begin{pmatrix} \omega \\ v \end{pmatrix} \quad \text{ou} \quad \tilde{T} = \begin{bmatrix} \tilde{\omega} & v \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\omega_z & \omega_y & v_x \\ \omega_z & 0 & -\omega_x & v_y \\ -\omega_y & \omega_x & 0 & v_z \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

avec : $\omega \in \mathfrak{so}(3)$ la vitesse angulaire et $v \in \mathfrak{se}(3)$ la vitesse linéaire

A partir du changement de coordonnées et en dérivant, on obtient :

$$\dot{P}^j = \dot{H}_i^j P^i$$

avec : $\dot{H}_i^j = H_i^j \tilde{T}$ ou $\dot{H}_i^j = \tilde{T} H_i^j$

Il existe alors deux possibilités pour écrire le torseur cinématique entre deux repères \mathfrak{R}_i et \mathfrak{R}_j en fonction du point de réduction choisi :

$$\tilde{T}_i^{i,j} = H_j^i \dot{H}_i^j \quad \text{ou} \quad \tilde{T}_i^{j,j} = \dot{H}_i^j H_j^i$$

avec : \tilde{T}_i^j le torseur cinématique du mouvement du repère \mathfrak{R}_i par rapport au repère \mathfrak{R}_j

On remarque que : $\tilde{T}_i^{j,j} = H_j^i \tilde{T}_i^{i,j} H_j^i$

- Changement de repère pour un torseur cinématique :

En utilisant la forme vectorielle du torseur cinématique, on obtient :

$$T_i^{j,j} = \text{Ad}_{H_i^j} T_i^{i,j}$$

avec :

$\text{Ad}_{H_i^j}$ la matrice adjointe qui est de la forme $\text{Ad}_{H_i^j} = \begin{pmatrix} R_i^j & 0 \\ \tilde{p}_i^j R_i^j & R_i^j \end{pmatrix}$ avec $R_i^j \in \text{SO}(3)$ et $\tilde{p}_i^j \in \mathbb{R}^3$

3.1.3 Torseur des efforts

Ce torseur représente les efforts et les moments qui s'appliquent sur un corps rigide. Il peut être écrit sous deux formes :

$$W = \begin{pmatrix} \tau \\ f \end{pmatrix} \quad \text{ou} \quad \tilde{W} = \begin{pmatrix} \tilde{f} & \tilde{\tau} \\ 0 & 0 \end{pmatrix}$$

avec : τ un couple et f une force linéaire

- Changement de coordonnées pour les torseurs d'effort :

Afin que le torseur d'efforts reste constant en passant d'un repère \mathfrak{R}_i à un repère \mathfrak{R}_j , on applique la relation suivante avec la forme vectorielle :

$$W^i = \text{Ad}_{H_i^j}^T W^j$$

avec :

$\text{Ad}_{H_i^j}^T$ la matrice adjointe qui est de la forme $\text{Ad}_{H_i^j}^T = \begin{pmatrix} R_j^i & -R_j^i \tilde{p}_i^j \\ 0 & R_j^i \end{pmatrix}$ avec $R_j^i \in \text{SO}(3)$ et $\tilde{p}_i^j \in \mathbb{R}^3$.

3.1.4 Théorème de Chasles

Un mouvement peut être décomposé en une rotation autour d'un axe et une translation le long de cet axe. Une des propriétés des groupes de Lie est la possibilité de relier des vecteurs de l'algèbre de Lie à des éléments du groupe de Lie.

Pour toute matrice H , il existe un torseur cinématique \tilde{T} tel que :

$$H = e^{\tilde{T}}$$

avec : $H \in \text{SE}(3)$ et $\tilde{T} \in \text{se}(3)$

3.1.5 Description des chaînes cinématiques arborescentes

Après la description de la cinématique d'un objet rigide, nous nous intéressons aux chaînes cinématiques.

Les liaisons utilisées ont un degré de liberté (liaison pivot ou liaison glissière). Les torseurs cinématiques unitaires ont respectivement la forme suivante :

$$\hat{T} = \begin{pmatrix} \hat{\omega} \\ \bullet \end{pmatrix} \quad \text{ou} \quad \hat{T} = \begin{pmatrix} 0 \\ \hat{v} \end{pmatrix}$$

- Ainsi, le déplacement relatif entre deux objets peut s'écrire $T_i^{j,j} = \theta \hat{T}_i^{j,j}$ avec $\theta \in \mathbb{R}$ et $\hat{T}_i^{j,j}$ un torseur unitaire constant.

Pour une liaison pivot, on a $\hat{T}_i^{j,j} = \begin{pmatrix} \hat{\omega} \\ r \wedge \hat{\omega} \end{pmatrix}$. Si l'axe de rotation passe par le centre de la

liaison, on a $\hat{T}_i^{j,j} = \begin{pmatrix} \hat{\omega} \\ 0 \end{pmatrix}$.

- Dans le cas de deux solides i et j liés par une liaison pivot, la position de i par rapport à j est donnée par :

$$H_i^j(\theta_i) = \exp\left(\tilde{T}_i^{j,j} \theta_i\right) H_i^j(0)$$

Cette formule permet de décrire une chaîne cinématique mais elle peut également être utilisée pour réaliser plusieurs changements de repères.

Maintenant que nous avons expliqué les différentes notations, nous décrivons dans le paragraphe suivant le modèle géométrique de la main virtuelle.

3.2 Description du modèle géométrique direct de la main virtuelle

Dans le paragraphe 2, nous avons décomposé la main en (Cf. Figure III.65) :

- un ensemble formé par le poignet et les métacarpes qui possède six degrés de liberté. Il est modélisé par un parallélépipède rectangle,

- cinq doigts (quatre doigts longs et le pouce) qui possèdent 20 degrés de liberté. Ils sont modélisés par des cylindres et des sphères formant cinq chaînes cinématiques série ouvertes.

Tout d'abord, nous précisons les différents repères utilisés ainsi que les matrices de changement de base nécessaires à l'établissement du modèle géométrique direct.

3.2.1 Les matrices de changement de base

Soient les repères orthonormés directs \mathcal{R}_{ed} (resp. \mathcal{R}_{eg}) de vecteurs unitaires \mathbf{x}_0 , \mathbf{y}_0 et \mathbf{z}_0 dont les origines O_0 se situent respectivement au centre des paumes virtuelles droite et gauche.

Nous définissons également un repère orthonormé direct pour chaque phalange noté $\mathcal{R}_{doigt}^i(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)$ dont l'origine O_i est située au centre de chaque articulation, avec $1 \leq i \leq 4$ pour les quatre degrés de mobilité de chaque doigt (Cf. Figure III.65 et Tableau III.2). Par exemple, les repères \mathcal{R}_{index}^1 et \mathcal{R}_{index}^2 sont liés à la première phalange de l'index.

Le repère global de la main virtuelle et les repères locaux des doigts sont modélisés sur la Figure III.67 :

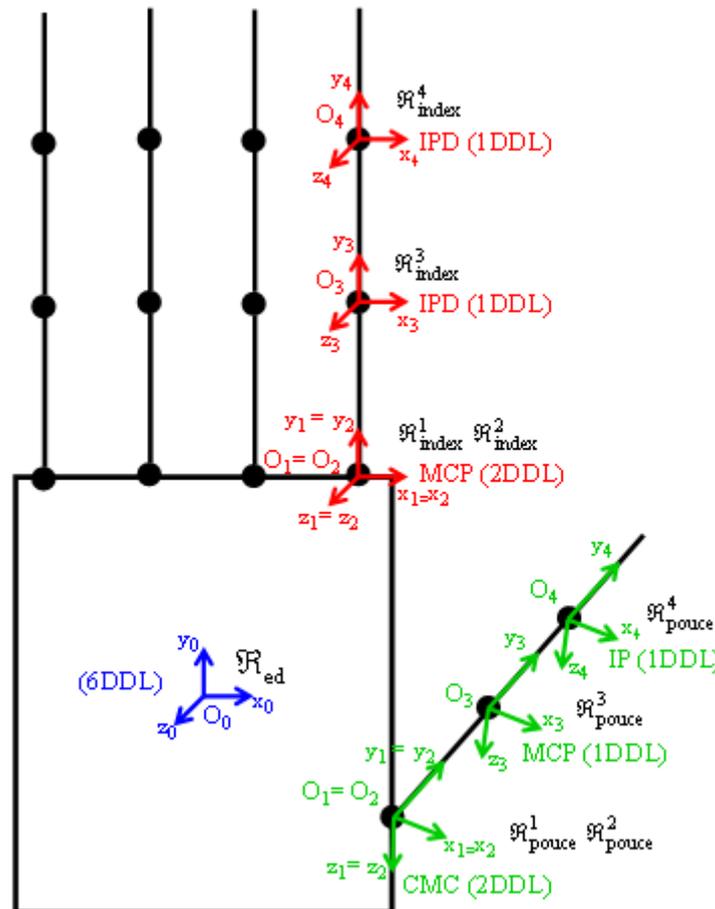


Figure III.67 : Les repères de la main virtuelle

Dans le cas d'un doigt long (index à auriculaire), la première phalange effectue un mouvement d'abduction/adduction (Cf. Chapitre I). Le repère \mathcal{R}_{doigt}^1 se déduit de \mathcal{R}_{ed} (resp.

\mathcal{R}_{eg}) par une rotation d'angle θ_{MCP_AA} autour de l'axe \mathbf{z}_0 - comme représenté sur la Figure III.68 - suivie d'une translation d'un vecteur $\lambda_1(\lambda_{ix}, \lambda_{iy}, 0)$

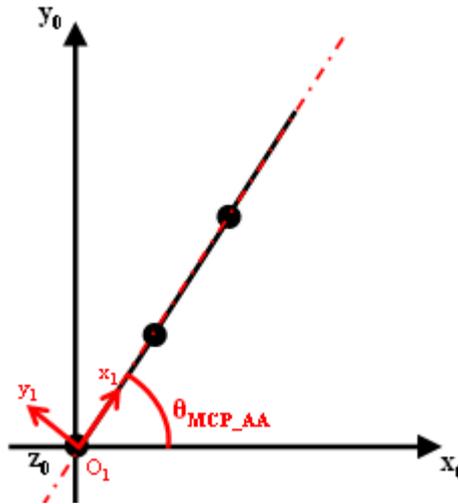


Figure III.68 : Position du repère \mathcal{R}_{doigt}^1 par rapport au repère d'origine

On obtient donc la relation suivante sous forme matricielle :

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{y}_1 \\ \mathbf{z}_1 \end{bmatrix} = \begin{bmatrix} \cos \theta_{MCP_AA} & -\sin \theta_{MCP_AA} & 0 \\ \sin \theta_{MCP_AA} & \cos \theta_{MCP_AA} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{y}_0 \\ \mathbf{z}_0 \end{bmatrix} \quad (\text{Eq. III-29})$$

On note \mathbf{M}_1^0 la matrice de passage de \mathcal{R}_{doigt}^1 à \mathcal{R}_{ed} (resp. \mathcal{R}_{eg}) telle que :

$$\mathbf{M}_1^0 = \begin{bmatrix} \cos \theta_{MCP_AA} & -\sin \theta_{MCP_AA} & 0 \\ \sin \theta_{MCP_AA} & \cos \theta_{MCP_AA} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{Eq. III-30})$$

De plus, les trois phalanges d'un doigt long (index à auriculaire) réalisent des mouvements de flexion/extension. En faisant abstraction des translations entre les repères locaux, les mouvements des premières, deuxièmes et troisièmes phalanges sont donc des rotations d'angles θ_{MCP} , θ_{IPP} et θ_{IPD} respectivement autour des axes \mathbf{x}_1 , \mathbf{x}_2 et \mathbf{x}_3 comme le modélise la Figure III.69 :

Le repère \mathcal{R}_{doigt}^2 se déduit de \mathcal{R}_{doigt}^1 en utilisant la relation suivante :

$$\begin{bmatrix} \mathbf{x}_2 \\ \mathbf{y}_2 \\ \mathbf{z}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_{MCP} & \sin \theta_{MCP} \\ 0 & -\sin \theta_{MCP} & \cos \theta_{MCP} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{y}_1 \\ \mathbf{z}_1 \end{bmatrix} \quad (\text{Eq. III-31})$$

On note \mathbf{M}_2^1 la matrice de passage de \mathcal{R}_{doigt}^2 à \mathcal{R}_{doigt}^1 telle que :

$$\mathbf{M}_2^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_{\text{MCP}} & \sin \theta_{\text{MCP}} \\ 0 & -\sin \theta_{\text{MCP}} & \cos \theta_{\text{MCP}} \end{bmatrix} \quad (\text{Eq. III-32})$$

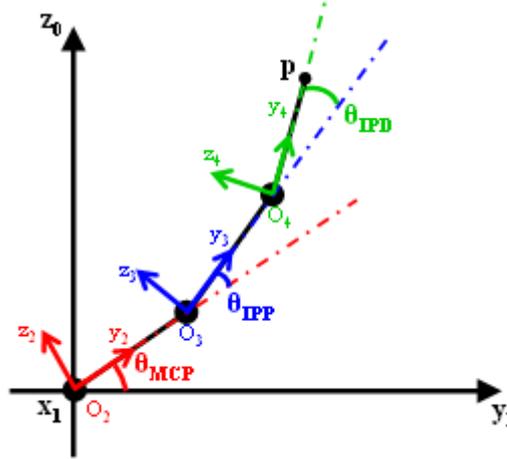


Figure III.69 : Position des repères pour les mouvements de flexion/extension

Le repère $\mathcal{R}_{\text{doigt}}^3$ se déduit de $\mathcal{R}_{\text{doigt}}^2$ en utilisant la matrice de passage \mathbf{M}_3^2 suivante :

$$\mathbf{M}_3^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_{\text{IPP}} & \sin \theta_{\text{IPP}} \\ 0 & -\sin \theta_{\text{IPP}} & \cos \theta_{\text{IPP}} \end{bmatrix} \quad (\text{Eq. III-33})$$

Le repère $\mathcal{R}_{\text{doigt}}^4$ se déduit de $\mathcal{R}_{\text{doigt}}^3$ en utilisant la matrice de passage \mathbf{M}_4^3 suivante :

$$\mathbf{M}_4^3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_{\text{IPD}} & \sin \theta_{\text{IPD}} \\ 0 & -\sin \theta_{\text{IPD}} & \cos \theta_{\text{IPD}} \end{bmatrix} \quad (\text{Eq. III-34})$$

De la même manière, il nous est possible de définir les matrices de passage pour le pouce. Dans ce cas de figure, nous obtenons des matrices de passage de même forme que précédemment mais pour des rotations d'angles $\theta_{\text{CMC_AA}}$, θ_{CMC} , θ_{MCP} et θ_{IP} respectivement autour des axes \mathbf{z}_0 , \mathbf{x}_1 , \mathbf{x}_2 et \mathbf{x}_3 du pouce.

Le modèle géométrique direct de chaque doigt est d'abord déterminé dans son repère local à partir des transformations :

$$\mathbf{M}_4^1 = \mathbf{M}_2^1 \cdot \mathbf{M}_3^2 \cdot \mathbf{M}_4^3 \quad (\text{Eq. III-35})$$

Nous utilisons les éléments du groupe spécial euclidien SE(3) pour décrire le modèle géométrique direct dans le repère global [Stramigioli 01a] [Stramigioli 01b] [Ravani 02]. Nous représentons ces éléments par les matrices de transformation à coordonnées homogènes.

3.2.2 Les matrices homogènes

Chaque doigt est modélisé par une chaîne cinématique arborescente possédant quatre degrés de mobilité. Les liaisons entre les corps rigides sont des liaisons rotoïdes. La Figure III.70 présente le schéma cinématique d'un doigt de la main avec les différents repères (0 à 4), les angles aux articulations (θ_{MCP_AA} , θ_{MCP} , θ_{IPP} et θ_{IPD}) et les longueurs des phalanges (L_{prox} , L_{inter} et L_{dist}) :

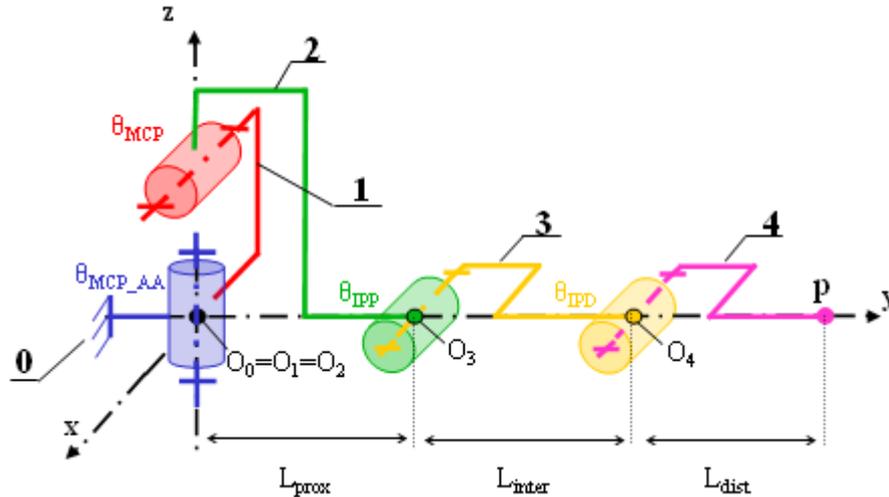


Figure III.70 : Schéma cinématique d'un doigt de la main virtuelle

Nous établissons l'expression des matrices de passage homogènes de chaque transformation qui vont nous permettre de décrire les cinq chaînes cinématiques de la main virtuelle. Ainsi, pour un doigt de la main virtuelle, nous obtenons les matrices homogènes suivantes :

- La matrice de transformation pour passer du repère \mathcal{R}_{doigt}^1 au repère \mathcal{R}_{ed} (resp. \mathcal{R}_{eg}) est :

$$\mathbf{H}_1^0 = \begin{bmatrix} \cos \theta_{MCP_AA} & -\sin \theta_{MCP_AA} & 0 & \lambda_{ix} \\ \sin \theta_{MCP_AA} & \cos \theta_{MCP_AA} & 0 & \lambda_{iy} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{Eq. III-36})$$

- La matrice de transformation pour passer du repère \mathcal{R}_{doigt}^2 au repère \mathcal{R}_{doigt}^1 est :

$$\mathbf{H}_2^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta_{MCP} & \sin \theta_{MCP} & 0 \\ 0 & -\sin \theta_{MCP} & \cos \theta_{MCP} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{Eq. III-37})$$

- La matrice de transformation pour passer du repère \mathcal{R}_{doigt}^3 au repère \mathcal{R}_{doigt}^2 est :

$$\mathbf{H}_3^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta_{IPP} & \sin \theta_{IPP} & L_{\text{prox}} \\ 0 & -\sin \theta_{IPP} & \cos \theta_{IPP} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{Eq. III-38})$$

- Le repère $\mathcal{R}_{\text{doigt}}^4$ se déduit de $\mathcal{R}_{\text{doigt}}^3$ en utilisant la matrice suivante :

$$\mathbf{H}_4^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta_{IPD} & \sin \theta_{IPD} & L_{\text{inter}} \\ 0 & -\sin \theta_{IPD} & \cos \theta_{IPD} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{Eq. III-39})$$

- La position de l'extrémité du doigt, notée p, est donnée par la matrice suivante :

$$\mathbf{H}_P^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_{\text{dist}} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{Eq. III-40})$$

Pour passer dans le repère global de la main virtuelle, nous appliquons donc la transformation suivante :

$$\mathbf{H}_4^0 = \mathbf{H}_1^0 \cdot \mathbf{H}_2^1 \cdot \mathbf{H}_3^2 \cdot \mathbf{H}_4^3 \quad (\text{Eq. III-41})$$

Nous en déduisons le modèle géométrique complet d'un doigt en notant p l'extrémité :

$$\begin{bmatrix} x_{0p} \\ y_{0p} \\ z_{0p} \\ 1 \end{bmatrix}_{\mathcal{R}_{\text{ed}}} = \mathbf{H}_1^0 \cdot \mathbf{H}_2^1 \cdot \mathbf{H}_3^2 \cdot \mathbf{H}_4^3 \cdot \begin{bmatrix} 0 \\ L_{\text{dist}} \\ 0 \\ 1 \end{bmatrix} \quad (\text{Eq. III-42})$$

D'où la position de l'extrémité d'un doigt dans le repère de référence de la main virtuelle donnée par la relation (Eq. III-43) :

$$\begin{bmatrix} x_{0p} \\ y_{0p} \\ z_{0p} \\ 1 \end{bmatrix}_{\mathcal{R}_{\text{ed}}} = \begin{bmatrix} -\sin \theta_{\text{MCP_AA}} (L_{\text{dist}} \cos(\theta_{\text{MCP}} + \theta_{\text{IPP}} + \theta_{\text{IPD}}) + L_{\text{inter}} \cos(\theta_{\text{MCP}} + \theta_{\text{IPP}}) + L_{\text{prox}} \cos \theta_{\text{MCP}}) + \lambda_{ix} \\ \cos \theta_{\text{MCP_AA}} (L_{\text{dist}} \cos(\theta_{\text{MCP}} + \theta_{\text{IPP}} + \theta_{\text{IPD}}) + L_{\text{inter}} \cos(\theta_{\text{MCP}} + \theta_{\text{IPP}}) + L_{\text{prox}} \cos \theta_{\text{MCP}}) + \lambda_{iy} \\ -L_{\text{dist}} \sin(\theta_{\text{MCP}} + \theta_{\text{IPP}} + \theta_{\text{IPD}}) - L_{\text{inter}} \sin(\theta_{\text{MCP}} + \theta_{\text{IPP}}) - L_{\text{prox}} \sin \theta_{\text{MCP}} \\ 1 \end{bmatrix}$$

La position géométrique zéro, i.e. que tous les angles aux articulations $\theta_{\text{MCP_AA}}$, θ_{MCP} , θ_{IPP} et θ_{IPD} sont nuls, correspond à la position des doigts tendus (Cf. Figure III.70). C'est la configuration de référence.

De la même manière, nous définissons le modèle géométrique du pouce dans le repère \mathcal{R}_{ed} à partir de l'équation III-43 en remplaçant les angles θ_{MCP_AA} , θ_{MCP} , θ_{IPP} et θ_{IPD} respectivement par les angles θ_{CMC_AA} , θ_{CMC} , θ_{MCP} et θ_{IP} .

Ces équations définissent les modèles géométriques directs complets des doigts de la main virtuelle dans le repère de référence.

Nous exprimons ces équations sous forme d'exponentielles dans la suite du paragraphe afin d'obtenir le modèle géométrique direct par la formule dite de « Brockett » [Stramigioli 01a] [Stramigioli 01b] [Ravani 02]. Pour cela, il est nécessaire de calculer les torseurs cinématiques unitaires.

Dans la configuration de référence, nous obtenons les torseurs cinématiques unitaires suivants pour un doigt :

- pour le mouvement d'abduction / adduction autour de l'axe \mathbf{z}_0 de la première phalange par rapport à la paume, nous avons le torseur suivant :

$$\hat{T}_1^{0,0} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{ou} \quad \tilde{T}_1^{0,0} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{Eq. III-44})$$

- pour les mouvements de flexion / extension des trois phalanges du doigt respectivement autour des axes \mathbf{x}_1 , \mathbf{x}_2 et \mathbf{x}_3 , nous obtenons :

$$\hat{T}_2^{1,1} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{ou} \quad \tilde{T}_2^{1,1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{Eq. III-45})$$

$$\hat{T}_3^{2,2} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -L_{\text{prox}} \end{pmatrix} \quad \text{ou} \quad \tilde{T}_3^{2,2} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & -L_{\text{prox}} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{Eq. III-46})$$

$$\hat{\mathbf{T}}_4^{3,3} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -L_{\text{inter}} \end{pmatrix} \quad \text{ou} \quad \tilde{\mathbf{T}}_4^{3,3} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & -L_{\text{inter}} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{Eq. III-47})$$

En utilisant l'équation III-41, nous obtenons la position de P dans le repère de base par la relation suivante :

$$\mathbf{H}_p^0 = \mathbf{H}_4^0 \cdot \mathbf{H}_p^4 \quad (\text{Eq. III-48})$$

avec \mathbf{H}_4^0 la matrice de passage telle que :

$$\mathbf{H}_4^0 = \mathbf{H}_1^0(\theta_{\text{MCP_AA}}) \cdot \mathbf{H}_2^1(\theta_{\text{MCP}}) \cdot \mathbf{H}_3^2(\theta_{\text{IPP}}) \cdot \mathbf{H}_4^3(\theta_{\text{IPD}}) \quad (\text{Eq. III-49})$$

Or par définition, nous avons (Cf. § 3.1.5) [Stramigioli 01a] [Stramigioli 01b] :

$$\mathbf{H}_i^j(\theta_i) = \exp\left(\tilde{\mathbf{T}}_i^{j,j} \cdot \theta_i\right) \mathbf{H}_i^j(0)$$

Nous obtenons donc les matrices de transformation suivantes :

$$\mathbf{H}_1^0(\theta_{\text{MCP_AA}}) = \exp\left(\tilde{\mathbf{T}}_1^{0,0} \cdot \theta_{\text{MCP_AA}}\right) \mathbf{H}_1^0(0) \quad (\text{Eq. III-50})$$

$$\mathbf{H}_2^1(\theta_{\text{MCP}}) = \exp\left(\tilde{\mathbf{T}}_2^{1,1} \cdot \theta_{\text{MCP}}\right) \mathbf{H}_2^1(0) \quad (\text{Eq. III-51})$$

$$\mathbf{H}_3^2(\theta_{\text{IPP}}) = \exp\left(\tilde{\mathbf{T}}_3^{2,2} \cdot \theta_{\text{IPP}}\right) \mathbf{H}_3^2(0) \quad (\text{Eq. III-52})$$

$$\mathbf{H}_4^3(\theta_{\text{IPD}}) = \exp\left(\tilde{\mathbf{T}}_4^{3,3} \cdot \theta_{\text{IPD}}\right) \mathbf{H}_4^3(0) \quad (\text{Eq. III-53})$$

avec $\mathbf{H}_i^{i-1}(0)$ la position pour $\theta_i = 0$.

En remplaçant dans l'équation III-49, nous obtenons alors la matrice de passage suivante :

$$\begin{aligned} \mathbf{H}_4^0 &= \exp\left(\tilde{\mathbf{T}}_1^{0,0} \cdot \theta_{\text{MCP_AA}}\right) \cdot \mathbf{H}_1^0(0) \times \exp\left(\tilde{\mathbf{T}}_2^{1,1} \cdot \theta_{\text{MCP}}\right) \cdot \mathbf{H}_2^1(0) \times \exp\left(\tilde{\mathbf{T}}_3^{2,2} \cdot \theta_{\text{IPP}}\right) \cdot \mathbf{H}_3^2(0) \times \\ &\quad \exp\left(\tilde{\mathbf{T}}_4^{3,3} \cdot \theta_{\text{IPD}}\right) \cdot \mathbf{H}_4^3(0) \\ &= \exp\left(\tilde{\mathbf{T}}_1^{0,0} \cdot \theta_{\text{MCP_AA}}\right) \times \left[\mathbf{H}_1^0(0) \cdot \exp\left(\tilde{\mathbf{T}}_2^{1,1} \cdot \theta_{\text{MCP}}\right) \cdot \mathbf{H}_2^1(0) \cdot \mathbf{H}_1^0(0) \right] \times \\ &\quad \left[\mathbf{H}_2^1(0) \cdot \exp\left(\tilde{\mathbf{T}}_3^{2,2} \cdot \theta_{\text{IPP}}\right) \cdot \mathbf{H}_3^2(0) \cdot \mathbf{H}_2^1(0) \right] \times \left[\mathbf{H}_3^2(0) \cdot \exp\left(\tilde{\mathbf{T}}_4^{3,3} \cdot \theta_{\text{IPD}}\right) \cdot \mathbf{H}_4^3(0) \cdot \mathbf{H}_3^2(0) \right] \times \mathbf{H}_4^3(0) \end{aligned}$$

$$\begin{aligned}
&= \exp\left(\tilde{\hat{T}}_1^{0,0} \cdot \theta_{\text{MCP_AA}}\right) \times \left[\exp\left(\tilde{\hat{T}}_2^{0,1} \cdot \theta_{\text{MCP}}\right) \cdot \mathbf{H}_1^0(0) \right] \times \left[\exp\left(\tilde{\hat{T}}_3^{1,2} \cdot \theta_{\text{IPP}}\right) \cdot \mathbf{H}_2^1(0) \right] \times \\
&\quad \left[\exp\left(\tilde{\hat{T}}_4^{2,3} \cdot \theta_{\text{IPD}}\right) \cdot \mathbf{H}_3^2(0) \right] \times \mathbf{H}_4^3(0) \\
&= \exp\left(\tilde{\hat{T}}_1^{0,0} \cdot \theta_{\text{MCP_AA}}\right) \times \exp\left(\tilde{\hat{T}}_2^{0,1} \cdot \theta_{\text{MCP}}\right) \times \left[\mathbf{H}_1^0(0) \cdot \exp\left(\tilde{\hat{T}}_3^{1,2} \cdot \theta_{\text{IPP}}\right) \cdot \mathbf{H}_0^1(0) \cdot \mathbf{H}_1^0(0) \right] \times \\
&\quad \left[\mathbf{H}_2^1(0) \cdot \exp\left(\tilde{\hat{T}}_4^{2,3} \cdot \theta_{\text{IPD}}\right) \cdot \mathbf{H}_1^2(0) \cdot \mathbf{H}_2^1(0) \right] \times \mathbf{H}_4^3(0) \\
&= \exp\left(\tilde{\hat{T}}_1^{0,0} \cdot \theta_{\text{MCP_AA}}\right) \times \exp\left(\tilde{\hat{T}}_2^{0,1} \cdot \theta_{\text{MCP}}\right) \times \exp\left(\tilde{\hat{T}}_3^{0,2} \cdot \theta_{\text{IPP}}\right) \cdot \mathbf{H}_1^0(0) \times \exp\left(\tilde{\hat{T}}_4^{1,3} \cdot \theta_{\text{IPD}}\right) \cdot \mathbf{H}_2^1(0) \times \mathbf{H}_4^3(0) \\
&= \exp\left(\tilde{\hat{T}}_1^{0,0} \cdot \theta_{\text{MCP_AA}}\right) \times \exp\left(\tilde{\hat{T}}_2^{0,1} \cdot \theta_{\text{MCP}}\right) \times \exp\left(\tilde{\hat{T}}_3^{0,2} \cdot \theta_{\text{IPP}}\right) \times \left[\mathbf{H}_1^0(0) \cdot \exp\left(\tilde{\hat{T}}_4^{1,3} \cdot \theta_{\text{IPD}}\right) \cdot \mathbf{H}_0^1(0) \cdot \mathbf{H}_1^0(0) \right] \times \\
&\quad \mathbf{H}_2^1(0) \times \mathbf{H}_4^3(0) \\
&= \exp\left(\tilde{\hat{T}}_1^{0,0} \cdot \theta_{\text{MCP_AA}}\right) \times \exp\left(\tilde{\hat{T}}_2^{0,1} \cdot \theta_{\text{MCP}}\right) \times \exp\left(\tilde{\hat{T}}_3^{0,2} \cdot \theta_{\text{IPP}}\right) \times \exp\left(\tilde{\hat{T}}_4^{0,3} \cdot \theta_{\text{IPD}}\right) \times \mathbf{H}_1^0(0) \times \mathbf{H}_2^1(0) \times \mathbf{H}_4^3(0)
\end{aligned}$$

D'où la relation (Eq. III-54) :

$$\mathbf{H}_4^0(\theta_{\text{MCP_AA}}, \dots, \theta_{\text{IPD}}) = \exp\left(\tilde{\hat{T}}_1^{0,0} \cdot \theta_{\text{MCP_AA}}\right) \times \exp\left(\tilde{\hat{T}}_2^{0,1} \cdot \theta_{\text{MCP}}\right) \times \exp\left(\tilde{\hat{T}}_3^{0,2} \cdot \theta_{\text{IPP}}\right) \times \exp\left(\tilde{\hat{T}}_4^{0,3} \cdot \theta_{\text{IPD}}\right) \times \mathbf{H}_4^0(0)$$

Cette matrice de passage nous permet donc d'obtenir la position des extrémités des doigts de la main virtuelle dans le repère de base de la paume. Elle utilise une expression sous forme d'exponentielles appelée « formule de Brockett » [Stramigioli 01a] [Stramigioli 01b] [Ravani 02].

Dans ce paragraphe, nous avons décrit le modèle géométrique direct d'un doigt de notre main virtuelle en utilisant les éléments du groupe de Lie. La méthode utilisée permet d'obtenir la position des extrémités des doigts dans le repère de référence à partir des angles aux articulations des phalanges : c'est la méthode dite de « cinématique directe ».

Il est également possible d'obtenir le modèle géométrique inverse pour les doigts de la main. Cela consiste à résoudre un système non linéaire de 3 équations à 4 inconnues $\theta_{\text{MCP_AA}}$, θ_{MCP} , θ_{IPP} et θ_{IPD} (ou $\theta_{\text{CMC_AA}}$, θ_{CMC} , θ_{MCP} et θ_{IP} pour le pouce). Dans le Chapitre I au paragraphe 3.2., nous avons présenté les relations de dépendance qui existent entre les phalanges d'un doigt [Lansmeer 55] [Rijpkema 91] [Kuch 95] [Ouhaddi 98] :

$$\theta_{\text{IPP}} = 2 \times \theta_{\text{MCP}} \quad \text{et} \quad \theta_{\text{IPD}} = \frac{2}{3} \times \theta_{\text{IPP}}$$

En intégrant ces relations, la solution du modèle géométrique inverse d'un doigt long est de la forme :

$$\theta = \left(\theta_{\text{MCP_AA}}, \theta_{\text{MCP}}, 2 \cdot \theta_{\text{MCP}}, \frac{4}{3} \cdot \theta_{\text{MCP}} \right) \quad (\text{Eq. III-55})$$

Ces relations de couplage nous permettent de traiter la redondance des chaînes cinématiques en réduisant le nombre de degrés de mobilité de chaque doigt. Il est alors possible d'obtenir le modèle géométrique inverse de chaque doigt.

Après avoir décrit le modèle géométrique direct des doigts, nous détaillons le modèle cinématique de la main virtuelle dans le paragraphe suivant.

Par définition, le modèle cinématique est l'écriture des vitesses des extrémités des doigts en fonction des vitesses articulaires. Nous exprimons cette relation par la matrice Jacobienne notée $\mathbf{J} = \mathbf{J}(\boldsymbol{\theta})$. Après avoir défini la matrice Jacobienne pour un doigt, nous chercherons à calculer la matrice Jacobienne de la main virtuelle.

3.3 Les équations de la cinématique

Nous déterminons les équations de la cinématique et plus particulièrement la matrice Jacobienne de notre main à partir de l'équation III-54.

3.3.1 Détermination de la matrice Jacobienne pour un doigt

Nous savons que :

$$\tilde{\mathbf{T}}_n^{0,0} = \dot{\mathbf{H}}_n^0 \mathbf{H}_0^n \quad \text{et} \quad \mathbf{H}_n^0 = \mathbf{H}_1^0 \mathbf{H}_2^1 \dots \mathbf{H}_n^{n-1}$$

En utilisant ces relations dans notre cas de figure, nous obtenons la relation (Eq. III-56) suivante :

$$\begin{aligned} \dot{\mathbf{H}}_4^0 &= \frac{d}{dt} (\mathbf{H}_1^0 \cdot \mathbf{H}_2^1 \cdot \mathbf{H}_3^2 \cdot \mathbf{H}_4^3) \\ &= \left(\frac{d\mathbf{H}_1^0}{dt} \cdot \mathbf{H}_2^1 \cdot \mathbf{H}_3^2 \cdot \mathbf{H}_4^3 + \mathbf{H}_1^0 \cdot \frac{d\mathbf{H}_2^1}{dt} \cdot \mathbf{H}_3^2 \cdot \mathbf{H}_4^3 + \mathbf{H}_1^0 \cdot \mathbf{H}_2^1 \cdot \frac{d\mathbf{H}_3^2}{dt} \cdot \mathbf{H}_4^3 + \mathbf{H}_1^0 \cdot \mathbf{H}_2^1 \cdot \mathbf{H}_3^2 \cdot \frac{d\mathbf{H}_4^3}{dt} \right) \end{aligned}$$

Or par définition, nous avons :

$$\mathbf{H}_i^j(\theta_i) = \exp\left(\tilde{\mathbf{T}}_i^{j,j} \theta_i\right) \mathbf{H}_i^j(0)$$

En dérivant cette relation, on obtiens :

$$\dot{\mathbf{H}}_i^j(\theta_i) = \tilde{\mathbf{T}}_i^{j,j} \dot{\theta}_i \times \exp\left(\tilde{\mathbf{T}}_i^{j,j} \theta_i\right) \mathbf{H}_i^j(0)$$

En remplaçant dans l'équation III-56, nous obtenons alors :

$$\begin{aligned}
 \dot{\mathbf{H}}_4^0 &= \left[\tilde{\mathbf{T}}_1^{0,0} \cdot \dot{\theta}_{\text{MCP_AA}} \cdot \exp\left(\tilde{\mathbf{T}}_1^{0,0} \cdot \theta_{\text{MCP_AA}}\right) \times \exp\left(\tilde{\mathbf{T}}_2^{1,1} \cdot \theta_{\text{MCP}}\right) \times \exp\left(\tilde{\mathbf{T}}_3^{2,2} \cdot \theta_{\text{IPP}}\right) \times \exp\left(\tilde{\mathbf{T}}_4^{3,3} \cdot \theta_{\text{IPD}}\right) \times \mathbf{H}_4^0(0) \right] + \\
 &= \left[\exp\left(\tilde{\mathbf{T}}_1^{0,0} \cdot \theta_{\text{MCP_AA}}\right) \times \tilde{\mathbf{T}}_2^{1,1} \cdot \dot{\theta}_{\text{MCP}} \cdot \exp\left(\tilde{\mathbf{T}}_2^{1,1} \cdot \theta_{\text{MCP}}\right) \times \exp\left(\tilde{\mathbf{T}}_3^{2,2} \cdot \theta_{\text{IPP}}\right) \times \exp\left(\tilde{\mathbf{T}}_4^{3,3} \cdot \theta_{\text{IPD}}\right) \times \mathbf{H}_4^0(0) \right] + \\
 &= \left[\exp\left(\tilde{\mathbf{T}}_1^{0,0} \cdot \theta_{\text{MCP_AA}}\right) \times \exp\left(\tilde{\mathbf{T}}_2^{1,1} \cdot \theta_{\text{MCP}}\right) \times \tilde{\mathbf{T}}_3^{2,2} \cdot \dot{\theta}_{\text{IPP}} \cdot \exp\left(\tilde{\mathbf{T}}_3^{2,2} \cdot \theta_{\text{IPP}}\right) \times \exp\left(\tilde{\mathbf{T}}_4^{3,3} \cdot \theta_{\text{IPD}}\right) \times \mathbf{H}_4^0(0) \right] + \\
 &= \left[\exp\left(\tilde{\mathbf{T}}_1^{0,0} \cdot \theta_{\text{MCP_AA}}\right) \times \exp\left(\tilde{\mathbf{T}}_2^{1,1} \cdot \theta_{\text{MCP}}\right) \times \exp\left(\tilde{\mathbf{T}}_3^{2,2} \cdot \theta_{\text{IPP}}\right) \times \tilde{\mathbf{T}}_4^{3,3} \cdot \dot{\theta}_{\text{IPD}} \cdot \exp\left(\tilde{\mathbf{T}}_4^{3,3} \cdot \theta_{\text{IPD}}\right) \times \mathbf{H}_4^0(0) \right] \\
 &= \left[\tilde{\mathbf{T}}_1^{0,0} \cdot \dot{\theta}_{\text{MCP_AA}} + \tilde{\mathbf{T}}_2^{1,1} \cdot \dot{\theta}_{\text{MCP}} + \tilde{\mathbf{T}}_3^{2,2} \cdot \dot{\theta}_{\text{IPP}} + \tilde{\mathbf{T}}_4^{3,3} \cdot \dot{\theta}_{\text{IPD}} \right] \times \\
 &\quad \left[\exp\left(\tilde{\mathbf{T}}_1^{0,0} \cdot \theta_{\text{MCP_AA}}\right) \times \exp\left(\tilde{\mathbf{T}}_2^{1,1} \cdot \theta_{\text{MCP}}\right) \times \exp\left(\tilde{\mathbf{T}}_3^{2,2} \cdot \theta_{\text{IPP}}\right) \times \exp\left(\tilde{\mathbf{T}}_4^{3,3} \cdot \theta_{\text{IPD}}\right) \right] \times \mathbf{H}_4^0(0) \\
 &= \begin{bmatrix} \tilde{\mathbf{T}}_1^{0,0} & \tilde{\mathbf{T}}_2^{1,1} & \tilde{\mathbf{T}}_3^{2,2} & \tilde{\mathbf{T}}_4^{3,3} \end{bmatrix} \times \begin{bmatrix} \dot{\theta}_{\text{MCP_AA}} & \dot{\theta}_{\text{MCP}} & \dot{\theta}_{\text{IPP}} & \dot{\theta}_{\text{IPD}} \end{bmatrix}^T \times \mathbf{H}_4^0
 \end{aligned}$$

En multipliant par la matrice \mathbf{H}_0^4 à droite, nous en déduisons la relation suivante :

$$\dot{\mathbf{H}}_4^0 \mathbf{H}_0^4 = \begin{bmatrix} \tilde{\mathbf{T}}_1^{0,0} & \tilde{\mathbf{T}}_2^{1,1} & \tilde{\mathbf{T}}_3^{2,2} & \tilde{\mathbf{T}}_4^{3,3} \end{bmatrix} \times \begin{bmatrix} \dot{\theta}_{\text{MCP_AA}} & \dot{\theta}_{\text{MCP}} & \dot{\theta}_{\text{IPP}} & \dot{\theta}_{\text{IPD}} \end{bmatrix}^T \quad (\text{Eq. III-57})$$

Or $\dot{\mathbf{H}}_4^0 \mathbf{H}_0^4 = \tilde{\mathbf{T}}_4^{0,0}$. En remplaçant dans l'équation III-57, nous avons la relation suivante :

$$\tilde{\mathbf{T}}_4^{0,0} = \begin{bmatrix} \tilde{\mathbf{T}}_1^{0,0} & \tilde{\mathbf{T}}_2^{1,1} & \tilde{\mathbf{T}}_3^{2,2} & \tilde{\mathbf{T}}_4^{3,3} \end{bmatrix} \times \begin{bmatrix} \dot{\theta}_{\text{MCP_AA}} & \dot{\theta}_{\text{MCP}} & \dot{\theta}_{\text{IPP}} & \dot{\theta}_{\text{IPD}} \end{bmatrix}^T$$

Au final, nous obtenons la relation suivante pour un doigt :

$$\mathbf{T}_4^{0,0} = \mathbf{J}(\boldsymbol{\theta}) \cdot \dot{\boldsymbol{\theta}} \quad (\text{Eq. III-58})$$

avec :

- $\dot{\boldsymbol{\theta}} = \begin{pmatrix} \dot{\theta}_{\text{MCP_AA}} & \dot{\theta}_{\text{MCP}} & \dot{\theta}_{\text{IPP}} & \dot{\theta}_{\text{IPD}} \end{pmatrix}^T$: les vitesses angulaires aux articulations des phalanges ;
- $\mathbf{J}(\boldsymbol{\theta}) = \begin{pmatrix} \mathbf{T}_1^{0,0} & \mathbf{T}_2^{0,1} & \mathbf{T}_3^{0,2} & \mathbf{T}_4^{0,3} \end{pmatrix}$: la matrice Jacobienne du doigt avec le torseur cinématique unitaire $\mathbf{T}_i^{0,i-1} = \text{Ad}_{\mathbf{H}_{i-1}^0} \tilde{\mathbf{T}}_i^{i-1,i-1}, \forall i \in [0;4]$.

La matrice Jacobienne $\mathbf{J}(\boldsymbol{\theta})$, dont l'expression est donnée par l'équation III-57, permet de calculer la vitesse de l'extrémité du doigt en fonction de la vitesse aux articulations.

Maintenant, nous généralisons les équations précédentes afin d'obtenir la matrice Jacobienne de la main virtuelle : i.e. en tenant compte des cinq doigts.

3.3.2 Détermination de la matrice Jacobienne pour la main virtuelle

Afin d'utiliser les notations du paragraphe 3.1 et afin d'exprimer les matrices Jacobiennes des cinq chaînes polyarticulées, nous numérotions les articulations et les corps rigides du doigt dans le sens de la profondeur. Pour chaque liaison k , nous utilisons les notations suivantes :

- $k.i$ pour un corps précédant la liaison k ,
- $k.o$ pour un corps suivant la liaison k .

D'où la position du corps dans le repère de base donnée par la matrice homogène suivante :

$$H_{k,o}^o = H_{k,i}^o H_{k,o}^{k,i} \quad (\text{Eq. III-59})$$

Ce qui donne la relation suivante pour réaliser plusieurs changements de repères :

$$H_{k,o}^{k,i}(\theta_k) = e^{\tilde{T}_{k,o}^{k,i} \theta_k} \cdot H_{k,o}^{k,i}(0) \quad (\text{Eq. III-60})$$

Cette relation permet de calculer le modèle géométrique direct par récurrence à partir des torseurs cinématiques unitaires $\tilde{T}_{k,o}^{k,i}$ de chaque liaison k de la chaîne cinématique.

Une sous-chaîne est définie par la chaîne série équivalente du corps 0 au corps i qui subit la force extérieure. Pour chaque sous-chaîne i , on calcule la matrice Jacobienne de la manière suivante :

$$\mathbf{J}_i(\boldsymbol{\theta}) = \underbrace{\begin{pmatrix} T_1 & T_2 & \dots & T_n \end{pmatrix}}_{\text{Nombre de liaisons}} \Downarrow \begin{matrix} 6 \text{ composantes : vitesse linéaire et vitesse angulaire} \end{matrix}$$

$$\text{avec : } T_k = \begin{cases} 0 & \text{si } k.o \notin \text{sous-chaîne } i \\ \text{Ad}_{H_{ki}^o} \tilde{T}_{k,o}^{k,i} & \text{sinon} \end{cases}$$

Le modèle cinématique de la main se compose de 16 corps rigides articulés par 20 liaisons parfaites (Cf. Figure III.65). Il se divise en cinq sous-chaînes arborescentes ouvertes. Nous obtenons pour chaque main virtuelle les matrices Jacobiennes $\mathbf{J}_i(\boldsymbol{\theta})$ suivantes (Eq. III-61) avec $i \in [1;5]$, où $i=1$ correspond au pouce :

$$\left\{ \begin{array}{l} \mathbf{J}_1(\boldsymbol{\theta}) = \begin{pmatrix} \tilde{T}_1^{0,0} & \text{Ad}_{H_1^o} \tilde{T}_2^{1,1} & \text{Ad}_{H_2^o} \tilde{T}_3^{2,2} & \text{Ad}_{H_3^o} \tilde{T}_4^{3,3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\ \mathbf{J}_2(\boldsymbol{\theta}) = \begin{pmatrix} 0 & 0 & 0 & 0 & \tilde{T}_5^{0,0} & \text{Ad}_{H_5^o} \tilde{T}_6^{5,5} & \text{Ad}_{H_6^o} \tilde{T}_7^{6,6} & \text{Ad}_{H_7^o} \tilde{T}_8^{7,7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\ \mathbf{J}_3(\boldsymbol{\theta}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{T}_9^{0,0} & \text{Ad}_{H_9^o} \tilde{T}_{10}^{9,9} & \text{Ad}_{H_{10}^o} \tilde{T}_{11}^{10,10} & \text{Ad}_{H_{11}^o} \tilde{T}_{12}^{11,11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\ \mathbf{J}_4(\boldsymbol{\theta}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{T}_{13}^{0,0} & \text{Ad}_{H_{13}^o} \tilde{T}_{14}^{13,13} & \text{Ad}_{H_{14}^o} \tilde{T}_{15}^{14,14} & \text{Ad}_{H_{15}^o} \tilde{T}_{16}^{15,15} & 0 & 0 & 0 & 0 \end{pmatrix} \\ \mathbf{J}_5(\boldsymbol{\theta}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{T}_{17}^{0,0} & \text{Ad}_{H_{17}^o} \tilde{T}_{18}^{17,17} & \text{Ad}_{H_{18}^o} \tilde{T}_{19}^{18,18} & \text{Ad}_{H_{19}^o} \tilde{T}_{20}^{19,19} \end{pmatrix} \end{array} \right.$$

Après avoir décrit le modèle cinématique et le modèle géométrique direct de notre main virtuelle, nous présentons les méthodes mises en œuvre afin de la contrôler en position et en orientation en temps-réel dans un environnement virtuel contraint. Dans le paragraphe 6 du Chapitre I, nous avons fait le choix d'utiliser la capture de mouvements pour piloter notre main virtuelle.

Dans le paragraphe suivant, nous présentons les différents systèmes de capture de mouvements existants ainsi que le système choisi pour nos applications. Puis, nous expliquerons le principe du couplage entre le monde réel et un environnement virtuel pour interagir en temps-réel.

4 Utilisation de la capture de mouvements

Dans le cadre de nos applications de saisie et de manipulation d'objets virtuels, il est important de connaître et de transmettre en temps-réel au simulateur physique la position et l'orientation des mains virtuelles par rapport à leurs positions d'origine et par rapport aux autres objets de la scène virtuelle. Pour ce faire, il est donc nécessaire de mesurer les six degrés de mobilité des mains de l'utilisateur ainsi que les angles aux articulations de ses doigts.

Dans ce paragraphe, nous présentons les interfaces qui permettent la localisation de l'utilisateur dans l'environnement et par conséquent le suivi des mouvements des mains de celui-ci. Les descriptions données ci-après sont issues de [Burdea 93], [Younblut 96] et [Fuchs 01].

4.1 Les systèmes de localisation existants

Il existe plusieurs systèmes qui permettent de mesurer la position spatiale d'un objet et de suivre sa trajectoire dans un environnement. Ils peuvent être répartis en quatre catégories :

- les capteurs mécaniques,
- les capteurs électromagnétiques,
- les capteurs acoustiques,
- les capteurs optiques.

Nous ne décrivons pas les capteurs mécaniques car ils peuvent gêner les mouvements en raison de leur liaison mécanique avec l'utilisateur. Ce manque d'ergonomie les rend difficilement utilisables. Nous nous focalisons donc sur les interfaces de localisation sans contact.

4.1.1 Les capteurs électromagnétiques

Ce type de traqueurs est composé d'un émetteur, d'un récepteur et d'une unité électronique. Des flux magnétiques sont émis dans l'espace. Le récepteur reçoit les signaux et les échantillonne dans son unité électronique. Un algorithme permet de déterminer la position et l'orientation du récepteur par rapport à l'émetteur. Ces traqueurs sont souvent utilisés pour suivre les mouvements de la tête et des mains.

Ces capteurs permettent de couvrir une zone importante. De plus, ils possèdent une bonne précision dans les mesures. Malgré leur coût élevé, ils sont souvent employés en Réalité Virtuelle. Leur inconvénient majeur est qu'ils peuvent être perturbés par la présence d'objets métalliques. Ils sont commercialisés entre autres par les sociétés Polhemus (*Fastrack*) [wPolhemus] et Ascension Technology Corporation (*Flock of Birds*) [wAscension].

4.1.2 Les capteurs acoustiques

Ce type de traqueurs est composé d'un émetteur à ultrasons, d'un récepteur lié à l'objet et d'une unité électronique. Avec ce capteur, la mesure de distances est déterminée par des temps de propagation des ultrasons dans l'air. Les mesures des différents temps de vol des ultrasons entre les émetteurs et les récepteurs permettent de déterminer la position de l'objet.

Ces capteurs permettent d'effectuer des mesures rapidement, mais ils doivent toujours être orientés vers les récepteurs : tout obstacle situé sur le parcours fait perdre le signal. Ce qui conduit à une zone d'étude limitée. Ils sont commercialisés entre autres par les sociétés Ascension Technology Corporation (*pcBird*) [wAscension] et InterSense (*IS 900*) [wIntersense].

4.1.3 Les capteurs optiques

Ce type de traqueurs est composé de plusieurs caméras fixes et de récepteurs ponctuels (LED à infrarouge ou cibles optiques réfléchissantes). En connaissant la position des caméras et en traitant les images acquises par celles-ci, on est capable de déterminer la position de l'objet traqué.

Ces capteurs sont précis mais il est important de bien positionner les caméras afin de couvrir les zones de déplacement. Il est également nécessaire d'utiliser trois caméras pour obtenir la position exacte d'une cible dans l'espace.

Les systèmes à infrarouge sont commercialisés entre autres par Qualisys (*active markers*) [wQualisys], A.R.T. (Advanced Realtime Tracking) GmbH (*finger tracking prototype*) [wArt], Northern Digital Inc. (*Polaris*) [wNorthern], Origin Instrument Corporation (*Dynasight*) [wOrigin] et Charnwood Dynamics (*Codamotion*) [wCharnwood].

Les systèmes réfléchissants sont commercialisés entre autres par Qualisys (*passive markers*) [wQualisys], A.R.T. GmbH (*hand target*) [wArt], Vicon [wVicon] et Northern Digital Inc. (*Polaris*) [wNorthern].

Les capteurs décrits précédemment permettent de mesurer la position et l'orientation d'un objet dans l'espace en temps réel et de suivre son mouvement au cours du temps. Nous les utiliserons pour traquer les mains de l'utilisateur. Dans notre application, il est également nécessaire de connaître la position des doigts afin d'effectuer les différentes tâches manuelles. Il existe des systèmes optiques utilisant des LED infrarouges développés par Charnwood Dynamics (*Coda motion capture*) [wCharnwood] et ARTracking (*DTrack*) [wArt] mais ils ne peuvent suivre les mouvements que de deux et trois doigts, respectivement. Ils ne répondent pas à notre besoin car nous devons piloter une main avec ses cinq doigts. Pour cela, nous utiliserons des gants de données. Quelques modèles sont présentés dans le paragraphe qui suit.

4.2 Les gants sensitifs

Ces gants permettent de mesurer les mouvements relatifs des doigts par rapport à la main. La difficulté principale est de mesurer les rotations des phalanges avec des dispositifs peu encombrants. Il existe plusieurs dispositifs :

- les gants à fibre optique,
- les gants à variation de résistances,
- les gants à effet Hall.

Ceux qui sont les mieux adaptés à notre problématique sont les deux premières catégories. Nous détaillons rapidement les principales caractéristiques de ces deux modèles de gants de données.

4.2.1 Les gants à fibre optique

Le DataGlove a été le premier gant développé pour interagir avec des environnements virtuels. Il est constitué d'un gant léger sur lequel sont montées des fibres optiques. A chaque articulation, une boucle en fibre optique permet la mesure de la position de la phalange. L'une des extrémités de la boucle est connectée à une LED. La lumière qui retourne à l'autre extrémité est collectée par un phototransistor. Lorsque l'utilisateur plie le doigt, la lumière sort. Ainsi, la valeur de l'angle articulaire se mesure indirectement par la mesure de l'intensité de la lumière réfléchiée par la fibre. Ce gant a été commercialisé par la société VPL Research, Inc. jusqu'à son rachat par Sun Microsystems, Inc.

Des gants moins coûteux basés sur la même technologie, les 5DT Datagloves, sont commercialisés par la société Fifth Dimension [w5dt]. La flexion des doigts est mesurée en

suivant le même principe que le gant précédent par 5, 14 ou 16 capteurs suivant le modèle utilisé.

4.2.2 Les gants à variation de résistances

Le gant P5 utilise la variation des jauges de contraintes pour mesurer le déplacement des doigts et pour positionner la main. Ainsi, quand le matériau se plie, la résistivité du matériau augmente. Après un calibrage fin de la main de l'utilisateur, le mouvement des doigts peut être déterminé. Il est développé par Essential Reality et commercialisé par Video Game Alliance [wP5].

Le CyberGlove mesure les trois flexions de chaque doigt par le biais de 18 capteurs. Il existe un modèle avec 22 capteurs angulaires qui mesure également les mouvements d'abduction/adduction entre les doigts. Le calibrage prédéfini donne directement les positions angulaires des doigts de l'utilisateur. Ce gant est commercialisé par Immersion Corporation [wImmersion].

4.3 Choix du matériel

Nous avons présenté les systèmes de capture de mouvements qui permettent de mesurer les déplacements d'un corps ou d'un objet en temps réel. Pour interagir avec l'environnement virtuel de la manière la plus naturelle possible, nous avons choisi un système de localisation sans contact afin que l'utilisateur ne soit pas gêné dans ses mouvements. Comme notre zone de travail est délimitée, nous nous sommes orientés vers un système de capteurs optiques : le système Vicon [wVicon].

Ce dernier dispose de quatre caméras placées en face de l'utilisateur. Nous diminuons ainsi les risques d'occultation des mains et/ou des doigts lors de la manipulation des objets ou des interfaces. Ce système offre également des fonctionnalités complémentaires – traitement d'images, module biomécanique, etc. - qui permettent à l'utilisateur de reconstruire les trajectoires suivies et d'analyser les mouvements (Cf. Figure III.71) [wStandard].



Figure III.71 : Le système Vicon

Quant au suivi des mouvements des doigts, nous avons opté pour le CyberGlove (Cf. Figure III.72) après l'avoir testé et comparé avec le gant P5 et le 5DT DataGlove. En effet, les mesures obtenues avec ce gant étaient beaucoup plus précises que celles données par les deux autres modèles. Comme nous avons pour objectif la simulation de tâches de préhension d'objets virtuels, il est essentiel de connaître les positions exactes des phalanges afin d'être le

plus réaliste possible des points de vue physique et visuel dans l'estimation des forces de contact.



Figure III.72 : Le CyberGlove

Maintenant que nous avons choisi les différents périphériques pour suivre les mouvements des mains et des doigts de l'utilisateur, nous devons piloter les mains virtuelles à partir des informations données par la capture de mouvements. Nous commençons par expliquer le principe du couplage entre le monde réel et un monde virtuel dans le paragraphe suivant.

4.4 Couplage entre le monde réel et le monde virtuel

Une simulation en Réalité Virtuelle fait cohabiter deux environnements distincts :

- le monde réel dans lequel l'opérateur se déplace et interagit avec les différents périphériques comme par exemple le système de capture de mouvements,
- le monde virtuel qui est l'espace de simulation et dans lequel ont lieu les interactions entre les objets virtuels et les mains virtuelles.

Afin que l'utilisateur se sente immergé dans la simulation et qu'il puisse interagir avec l'environnement virtuel, il est nécessaire de faire la liaison entre ses deux mondes. Pour cela, nous utilisons un couplage en position à l'aide d'un ressort-amortisseur virtuel. Ce type de couplage est équivalent à un contrôleur proportionnel-dérivé (noté PD) [Duidam 01] [Stramigioli 01b]. Le principe de cet asservissement est expliqué ci-dessous par un exemple d'application dans le cas 1D.

Considérons la main de l'utilisateur qui se déplace en translation suivant l'axe x (Cf. Figure III.73) :

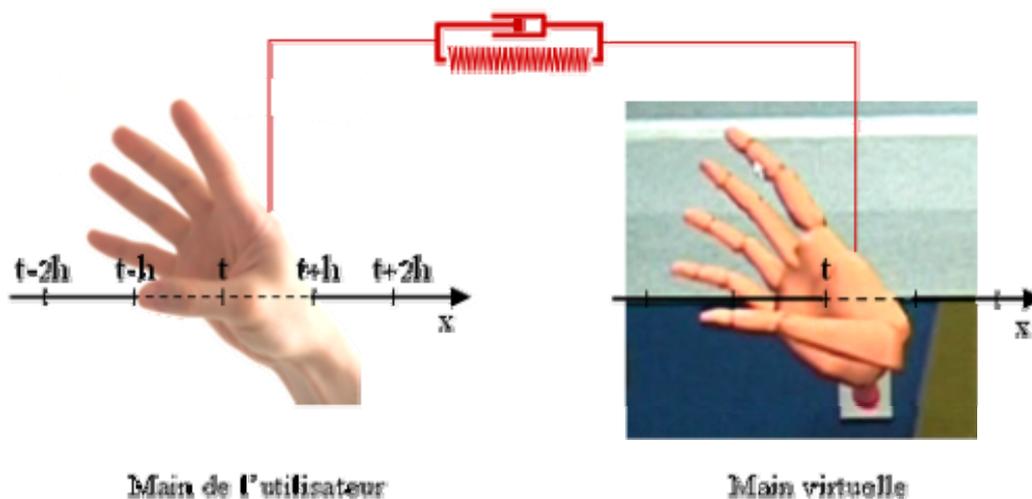


Figure III.73 : Le principe du couplage virtuel

Pour déplacer la main virtuelle en fonction des mouvements de translation de la main dans l'espace réel, nous calculons un effort \mathbf{F} à partir de la position et de la vitesse de la main réelle. En se plaçant dans \mathbb{R}^n , nous obtenons alors la relation suivante dans le cas 1D :

$$\mathbf{F} = \mathbf{k} \Delta \mathbf{x} + \mathbf{b} \Delta \dot{\mathbf{x}} \quad (\text{Eq. III-62})$$

avec :

- \mathbf{k} et \mathbf{b} qui sont respectivement les coefficients de raideur et d'amortissement du couplage virtuel ;
- $\Delta \mathbf{x}$ la différence entre la position de la main de l'utilisateur et la position de la main virtuelle ;
- $\Delta \dot{\mathbf{x}}$ la différence entre la vitesse de la main de l'utilisateur et la vitesse de la main virtuelle.

L'effort \mathbf{F} appliqué à la main virtuelle permet de diminuer les erreurs en position de celle-ci par rapport à la position de la main dans l'espace réel. L'utilisation de ce couplage permet d'avoir un asservissement en position et en vitesse d'un objet virtuel sur un objet réel.

Après avoir expliqué le principe du couplage virtuel de type ressort-amortisseur entre le monde réel et un environnement virtuel, nous détaillons les algorithmes mis en place pour piloter la main et les doigts dans les paragraphes qui suivent. Bien que les tests aient été réalisés avec du matériel spécifique (i.e. le système Vicon et le CyberGlove), les lois de commande mises en œuvre sont génériques. En effet, elles peuvent être utilisées avec n'importe quel système de capture de mouvements et n'importe quel type de gants de données.

5 Description du contrôle 6D

Dans le cas du couplage à notre simulation physique, nous généralisons le principe décrit précédemment au cas d'un objet 3D avec six degrés de liberté. Ainsi, nous utilisons les matrices à coordonnées homogènes pour décrire les configurations et les torseurs cinématiques de la main virtuelle. Par exemple, un déplacement se caractérise par une vitesse linéaire et une vitesse angulaire. Nous utilisons les éléments du groupe spécial euclidien SE(3) pour la description de nos algorithmes [Stramigioli 01b] [Ravani 02]. La Figure III.74 présente le principe de notre contrôleur :

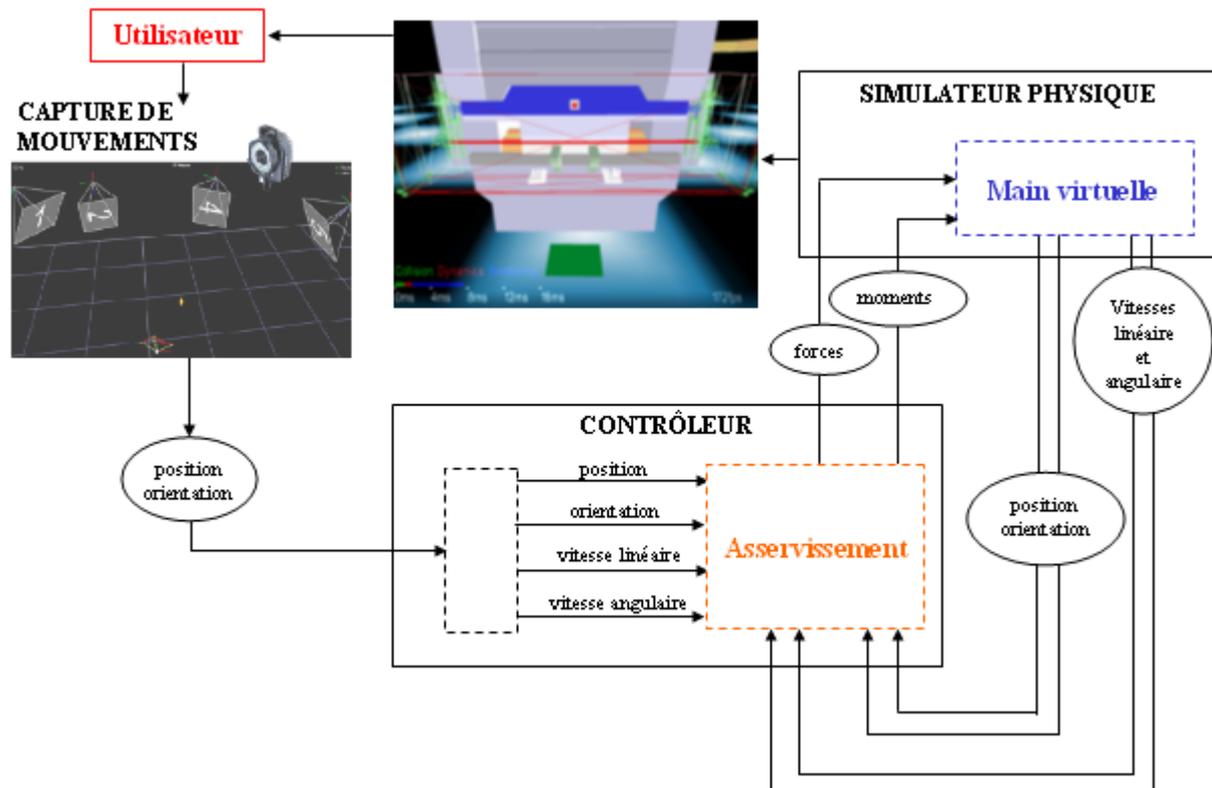


Figure III.74 : Le contrôleur

Dans la suite du paragraphe, nous détaillons la commande des mains virtuelles dans l'environnement à partir des données fournies par le système de capture de mouvements – i.e. la consigne) – et celles données par le moteur de simulation physique. Comme décrit précédemment, nous allons utiliser les informations de positions et de vitesses de la simulation pour réaliser notre asservissement.

Après la description des repères utilisés dans notre simulation, nous présentons l'asservissement de la main qui permet d'interagir avec l'environnement virtuel en temps réel. Ensuite, nous expliquerons les lois de commande qui ont été développées pour asservir les doigts afin de saisir et manipuler des objets virtuels.

5.1 Description et paramétrages

L'utilisateur évolue dans la plate-forme de Réalité Virtuelle. Le repère global de la plate-forme est noté \mathcal{R}_{pf} . C'est un repère fixe.

Les déplacements de ses mains sont mesurés en temps réel par un système de capture de mouvements. On définit deux repères mobiles notés \mathcal{R}_{md} pour la main droite et \mathcal{R}_{mg} pour la main gauche de l'utilisateur. Les positions des mains sont les consignes de notre asservissement.

Les mains virtuelles modélisées par un logiciel de simulation physique sont affichées sur l'écran. On définit deux repères mobiles notés \mathcal{R}_{ed} et \mathcal{R}_{eg} , respectivement pour la main virtuelle droite et la main virtuelle gauche.

La Figure III.75 ci-dessous représente les différents repères de notre simulation :

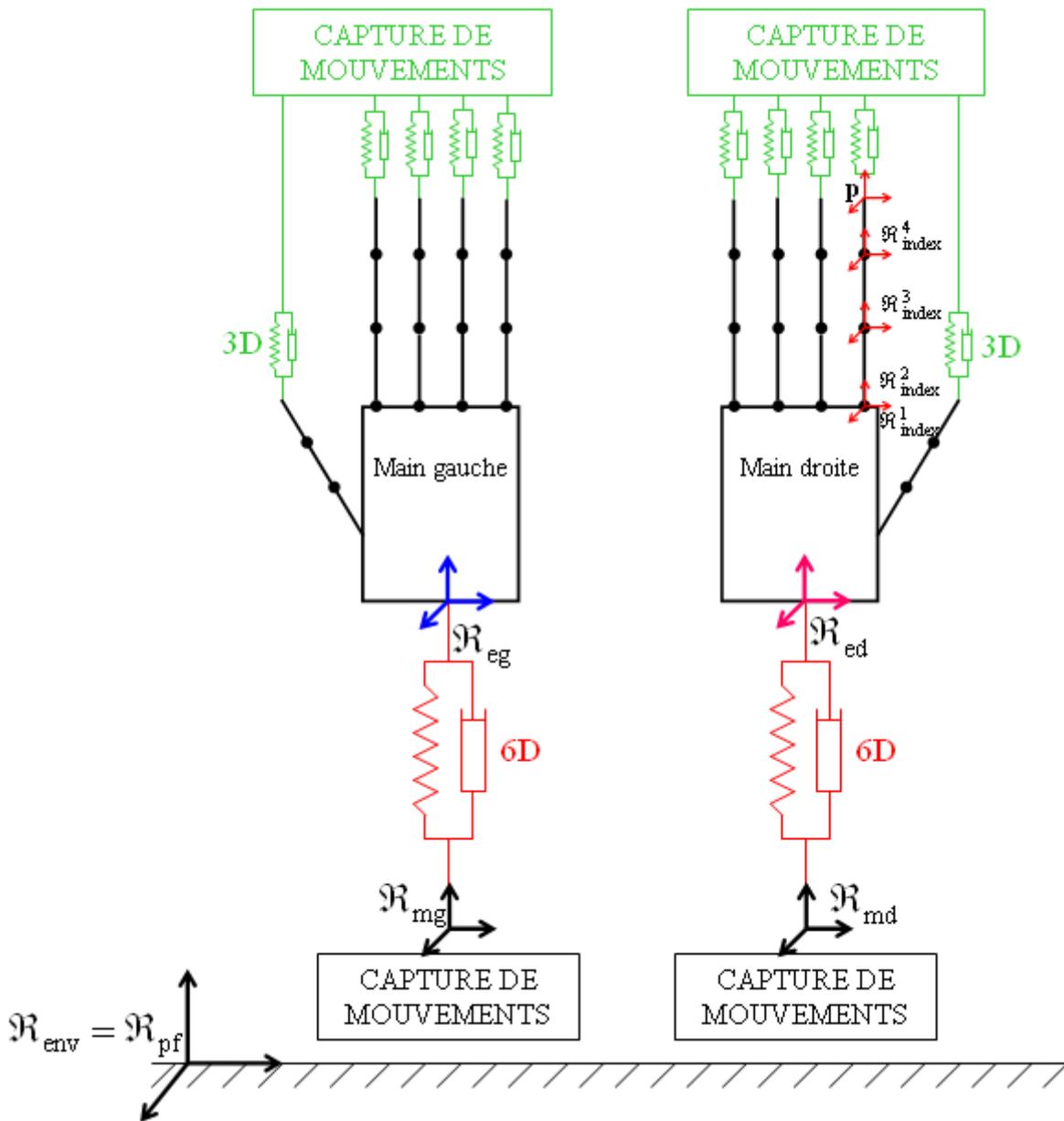


Figure III.75 : Repères utilisés

Dans les paragraphes qui suivent, nous présentons les lois de commande mises en place pour piloter la main virtuelle en temps réel.

5.2 Le contrôle-commande

5.2.1 Les données du système de capture de mouvements

Ce système mesure la position des mains à chaque pas de temps dans le repère \mathcal{R}_{pf} . Nous obtenons donc en consigne les matrices \mathbf{H}_{md}^{pf} et \mathbf{H}_{mg}^{pf} . A partir de ces déplacements, nous pouvons calculer les torseurs cinématiques $T_{md}^{pf,pf}$ et $T_{mg}^{pf,pf}$ des mains de l'utilisateur qui se

déplacent dans la plateforme de Réalité Virtuelle en changeant le repère de projection et de réduction de $T_{md}^{pf,md}$ et $T_{mg}^{pf,mg}$ par :

$$T_{md}^{pf,pf} = Ad_{H_{md}^{pf}} T_{md}^{pf,md} \quad (\text{Eq. III-63})$$

$$T_{mg}^{pf,pf} = Ad_{H_{mg}^{pf}} T_{mg}^{pf,mg} \quad (\text{Eq. III-64})$$

Nous connaissons donc les configurations et les déplacements des mains virtuelles dès l'initialisation de la démonstration et à chaque pas de temps de la simulation.

5.2.2 Les données du moteur physique

Nous associons des repères orthonormés directs \mathcal{R}_{ed} et \mathcal{R}_{eg} , respectivement à la main virtuelle droite et la main virtuelle gauche (Cf. § 3, Figure III.67 et Figure III.75).

A chaque pas de temps, le moteur de simulation physique nous donne la position, l'orientation, la vitesse linéaire et la vitesse angulaire des mains virtuelles dans le repère de base de ce moteur physique noté \mathcal{R}_{vortex} . Nous obtenons donc les matrices \mathbf{H}_{ed}^{vortex} et \mathbf{H}_{eg}^{vortex} , et les torseurs cinématiques $T_{ed}^{vortex,vortex}$ et $T_{eg}^{vortex,vortex}$ à chaque pas de temps.

De la même manière que précédemment, nous exprimons les matrices et les torseurs cinématiques dans le repère de base \mathcal{R}_{pf} en effectuant un changement de repère par multiplication de la matrice adjointe $Ad_{H_{vortex}^{pf}}$. Cette matrice est connue et elle est déterminée à l'initialisation de la simulation. Nous obtenons alors les positions des mains virtuelles notées \mathbf{H}_{ed}^{pf} et \mathbf{H}_{eg}^{pf} et les torseurs cinématiques $T_{ed}^{pf,pf}$ et $T_{eg}^{pf,pf}$.

5.2.3 L'asservissement

Les calculs doivent s'effectuer dans le repère de la plateforme \mathcal{R}_{pf} . Une fois que toutes les matrices sont exprimées dans ce repère, nous calculons les différences en position et en vitesse entre la main de l'utilisateur – notée md ou mg - et la main virtuelle – notée ed ou eg. Nous asservissons la main virtuelle à celle de l'utilisateur en utilisant un contrôleur de type proportionnel-dérivé (PD) qui consiste à établir un couplage masse-ressort entre les deux mains. Ainsi, ce contrôleur calcule les efforts à appliquer sur la main virtuelle à partir des erreurs en position et en vitesse mesurées entre la main de l'utilisateur et les données du moteur physique afin qu'elle atteigne la consigne.

Dans la suite, nous présentons le principe de la loi de commande mise en œuvre pour piloter la main virtuelle dans l'espace. Ensuite, nous décrivons les deux contrôleurs développés et testés pour asservir les doigts des mains virtuelles afin de simuler des tâches de préhension.

5.2.3.1 Contrôle de la main

Les positions des mains de l'utilisateur sont mesurées par le système de capture de mouvements à chaque pas de temps de la simulation. Les consignes de notre asservissement sont donc les coordonnées cartésiennes des cibles optiques positionnées sur les paumes de l'utilisateur. La loi de commande du contrôleur PD modélise un système ressort-

amortissement virtuel entre la main de l'utilisateur représentée par la matrice $\mathbf{H}_{\text{md}}^{\text{pf}}$ et la main virtuelle représentée par la matrice $\mathbf{H}_{\text{ed}}^{\text{pf}}$. Dans ce paragraphe, nous définissons les torseurs d'efforts associés à la raideur et à l'amortissement de ce couplage.

5.2.3.1.1 Torseur d'efforts associé à la raideur

Ainsi, nous définissons un ressort spatial qui permet de connecter la main virtuelle à la main de l'utilisateur [Fasse 99] [Duidam 01]. Nous pouvons alors définir une application qui associe à une certaine position du ressort le torseur d'effort ${}^{\text{K}}\mathbf{W}_{\text{ed}}$ [Fasse 98a] [Fasse 98b]. Ce torseur d'efforts est donc caractérisé par une fonction d'énergie potentielle notée f telle que :

$${}^{\text{K}}\mathbf{W}_{\text{ed}} = f(\mathbf{H}_{\text{ed}}^{\text{md}}) = \begin{pmatrix} {}^{\text{K}}\boldsymbol{\tau} \\ {}^{\text{K}}\mathbf{f} \end{pmatrix}$$

avec :

$${}^{\text{K}}\tilde{\boldsymbol{\tau}} = 2 \text{ as}(\mathbf{G}_o \mathbf{R}_{\text{ed}}^{\text{md}}) + \text{ as}(\mathbf{G}_t \mathbf{R}_{\text{md}}^{\text{ed}} \tilde{\mathbf{p}}_{\text{ed}}^{\text{md}} \tilde{\mathbf{p}}_{\text{ed}}^{\text{md}} \mathbf{R}_{\text{ed}}^{\text{md}}) + 2 \text{ as}(\mathbf{G}_c \tilde{\mathbf{p}}_{\text{ed}}^{\text{md}} \mathbf{R}_{\text{ed}}^{\text{md}}) \quad (\text{Eq. III-65})$$

$${}^{\text{K}}\tilde{\mathbf{f}} = \mathbf{R}_{\text{md}}^{\text{ed}} \text{ as}(\mathbf{G}_t \tilde{\mathbf{p}}_{\text{ed}}^{\text{md}}) \mathbf{R}_{\text{ed}}^{\text{md}} + \text{ as}(\mathbf{G}_t \mathbf{R}_{\text{md}}^{\text{ed}} \tilde{\mathbf{p}}_{\text{ed}}^{\text{md}} \mathbf{R}_{\text{ed}}^{\text{md}}) + 2 \text{ as}(\mathbf{G}_c \mathbf{R}_{\text{ed}}^{\text{md}}) \quad (\text{Eq. III-66})$$

où :

- $\text{as}(\cdot)$: est l'opérateur qui renvoie la matrice symétrique gauche d'une matrice,
- $\mathbf{R}_{\text{ed}}^{\text{md}}$ et $\tilde{\mathbf{p}}_{\text{ed}}^{\text{md}}$ sont les éléments de la matrice homogène $\mathbf{H}_{\text{ed}}^{\text{md}}$,
- \mathbf{G}_o , \mathbf{G}_t , \mathbf{G}_c sont respectivement les co-raideurs d'orientation, de translation et de couplage du ressort.

Les matrices de co-raideurs sont calculées à partir des matrices de raideurs d'orientation \mathbf{K}_o , de translation \mathbf{K}_t et de couplage \mathbf{K}_c du ressort. Pour calculer ces matrices de raideur, nous nous plaçons au centre de raideur [Fasse 99] [Duidam 01]. Dans ces conditions, l'expression de la matrice de raideur découple les rotations et les translations et nous obtenons donc une matrice de raideur \mathbf{K} de la forme :

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_t \mathbf{I}_3 & \mathbf{K}_c \mathbf{I}_3 \\ \mathbf{K}_c \mathbf{I}_3 & \mathbf{K}_o \mathbf{I}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{K}_t \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_o \mathbf{I}_3 \end{bmatrix}$$

A l'équilibre, quand le centre de raideur est l'origine du repère, nous obtenons les relations suivantes entre les matrices \mathbf{K}_α et \mathbf{G}_α , avec $\alpha = \{o, t, c\}$ [Fasse 98a] [Duidam 01] :

$$\mathbf{G}_\alpha = \frac{1}{2} \text{tr}(\mathbf{K}_\alpha) \mathbf{I} - \mathbf{K}_\alpha \quad (\text{Eq. III-67})$$

5.2.3.1.2 Torseur d'efforts associé à l'amortisseur

Puis, nous calculons le torseur d'efforts engendré par l'amortissement noté ${}^{\text{B}}\mathbf{W}_{\text{ed}}$ à partir des torseurs cinématiques et des gains en amortissement B_t et B_o respectivement pour les mouvements de translation et de rotation. Nous obtenons l'équation de la forme :

$$\begin{aligned} ({}^{\text{B}}\mathbf{W}_{\text{ed}})^{\text{T}} &= \mathbf{B} \cdot \mathbf{T}_{\text{ed}}^{0,\text{md}} \\ &= \mathbf{B} \cdot \text{Ad}_{\mathbf{H}_{\text{md}}^0} \tilde{\mathbf{T}}_{\text{ed}}^{\text{md},\text{md}} \end{aligned} \quad (\text{Eq. III-68})$$

Avec \mathbf{B} une matrice (6×6) telle que $\mathbf{B} = \begin{bmatrix} \mathbf{B}_t \cdot \mathbf{I}_3 & \mathbf{B}_c \mathbf{I}_3 \\ \mathbf{B}_c \mathbf{I}_3 & \mathbf{B}_o \cdot \mathbf{I}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{B}_t \cdot \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_o \cdot \mathbf{I}_3 \end{bmatrix}$ au centre de raideur.

5.2.3.1.3 Torseur d'efforts total appliqué à la main virtuelle

En sommant les torseurs d'efforts engendrés par le ressort et l'amortisseur, nous obtenons le torseur total \mathbf{W}_{ed} des efforts exercés à appliquer sur l'objet virtuel :

$$\begin{aligned} \mathbf{W}_{ed} &= {}^K \mathbf{W}_{ed} + {}^B \mathbf{W}_{ed} \\ &= f(\mathbf{H}_{ed}^{md}) + \mathbf{B} \cdot \mathbf{T}_{ed}^{0,md} \\ &= f(\mathbf{H}_{ed}^{md}) + \mathbf{B} \cdot \text{Ad}_{H_{md}^0} \tilde{\mathbf{T}}_{ed}^{md,md} \end{aligned} \quad (\text{Eq. III-69})$$

Nous en déduisons donc les efforts à appliquer à la main virtuelle : nous venons d'identifier les lois de commande du contrôleur de la main.

Les matrices de raideur et d'amortissement de l'objet virtuel dépendent de l'inertie, des dimensions et des caractéristiques mécaniques (matériau) de l'objet virtuel.

5.2.3.2 Contrôle des doigts

La main de l'utilisateur est suivie par un système de capture de mouvements. Quant aux doigts, leurs déplacements peuvent être traqués à l'aide de ce même système ou alors par un gant de données.

Dans ce paragraphe, nous présentons les méthodes utilisées pour asservir les doigts virtuels aux doigts de l'utilisateur dans chacun des cas de figure. Celles-ci reposent sur le même principe (i.e. un couplage virtuel de type masse-ressort) mais elles ne conduisent pas aux mêmes équations.

Dans les deux cas, nous cherchons à décrire le torseur des efforts appliqué aux doigts virtuels en fonction des positions et des vitesses des doigts de l'utilisateur. Comme détaillé dans le paragraphe précédent, nous devons donc définir deux torseurs distincts : l'un dédié aux efforts dus au ressort, noté ${}^K \mathbf{W}_{doigt}^i$, et l'autre à ceux exercés par l'amortisseur, noté ${}^B \mathbf{W}_{doigt}^i$.

5.2.3.2.1 Contrôle articulaire des doigts virtuels

Nous utilisons un gant de données pour mesurer les positions angulaires de chaque articulation de la main. Nous estimons à chaque pas de temps les différences de positions angulaires entre les mesures effectuées par le gant – i.e. la consigne - et les angles des phalanges virtuelles donnés par le moteur de simulation physique.

Dans le cas d'une articulation, nous ne considérons que les matrices de rotation, les vitesses angulaires et les couples. Nous obtenons le couple articulaire à appliquer par la relation suivante :

$$\boldsymbol{\tau}_i = \mathbf{K}_\theta (\boldsymbol{\theta}_{des} - \boldsymbol{\theta}) + \mathbf{B}_\theta (\dot{\boldsymbol{\theta}}_{des} - \dot{\boldsymbol{\theta}}) \quad (\text{Eq. III-70})$$

avec :

- θ_{des} et $\dot{\theta}_{des}$ qui sont respectivement le vecteur orientation (position angulaire) et vecteur de la vitesse angulaire à atteindre (i.e. la consigne) ;
- θ et $\dot{\theta}$ les vecteurs courants ;
- K_θ et B_θ sont les matrices de raideur et d'amortissement pour chaque articulation.

Cette relation écrite dans l'espace articulaire nous permet d'asservir les phalanges virtuelles en imposant des couples moteurs à chaque articulation. Nous pouvons donc contrôler les chaînes cinématiques des doigts en position.

Au lieu de contrôler les phalanges par des couples moteur directement calculés à partir des mesures angulaires d'un gant de données, il est également possible de les estimer à partir des efforts exercés en bout de chaîne, i.e. sur les extrémités des doigts. Pour ce faire, nous utilisons des marqueurs optiques pour suivre les déplacements et les orientations des extrémités des doigts de l'utilisateur.

5.2.3.2.2 Contrôle cartésien pour l'extrémité des doigts

Dans ce cas de figure, nous connaissons donc la position cartésienne des extrémités des doigts dans l'espace ainsi que la position et l'orientation de la paume à chaque pas de temps.

Nous avons donc comme données d'entrée de notre asservissement le modèle géométrique noté $H_j(\theta)$ et le torseur cinématique noté $T_j(\dot{\theta})$ avec j le doigt considéré ($j=1$ correspond au pouce).

De plus, nous avons déterminé la matrice Jacobienne d'un doigt dans le paragraphe 3.3.1. Elle est de la forme :

$$T_j(\dot{\theta}) = J_j(\theta) \cdot \dot{\theta}_j \quad (\text{Eq. III-71})$$

avec :

- $\dot{\theta}_j$ les vitesses angulaires aux articulations des phalanges du doigt j ;
- $J_j(\theta)$ la matrice Jacobienne du doigt j .

Pour rappel, nous obtenons pour l'index la matrice Jacobienne $J_2(\theta)$ suivante :

$$\begin{aligned} T_2(\dot{\theta}) &= \begin{bmatrix} T_1^{0,0} & T_2^{0,1} & T_3^{0,2} & T_4^{0,3} \end{bmatrix} \times \begin{bmatrix} \dot{\theta}_{MCP_AA} & \dot{\theta}_{MCP} & \dot{\theta}_{IPP} & \dot{\theta}_{IPD} \end{bmatrix}^T \\ &= J_2(\theta) \cdot \dot{\theta}_2 \end{aligned}$$

Nous estimons les couples nécessaires pour atteindre la configuration souhaitée à partir de la cinématique des doigts décrite dans le paragraphe 3. Pour ce faire, nous calculons le torseur d'efforts W_j de la sous-chaîne j qui s'exerce sur la phalange distale asservie. Comme pour le contrôle de la main (Cf. paragraphe 5.2.3.1), ce torseur correspond aux efforts exercés par un couplage virtuel de type ressort-amortisseur créé pour l'asservissement des extrémités des doigts. Celui-ci est de la forme :

$$W_j = {}^K W_j + {}^B W_j \quad (\text{Eq. III-72})$$

- Le torseur d'efforts engendré par le ressort, noté ${}^K W_j$, est caractérisé par une fonction d'énergie potentielle notée $f_j(H_j(\theta))$ telle que :

$${}^K W_j = f_j(H_j(\theta)) \quad (\text{Eq. III-73})$$

Les couples et les efforts exercés sur les extrémités des doigts ont respectivement leurs expressions identiques aux équations III-65 et III-66 dans le repère local du doigt.

- Le torseur d'efforts engendré par l'amortissement, noté ${}^B W_j$, est déterminé à partir du torseur cinématique $T_j(\dot{\theta})$ et de la matrice d'amortissement B_j :

$$\begin{aligned} {}^B W_j &= B_j \cdot T_j(\dot{\theta}) \\ &= B_j \cdot J_j(\theta) \cdot \dot{\theta}_j \end{aligned} \quad (\text{Eq. III-74})$$

Nous obtenons donc le torseur total W_j des efforts exercés sur les extrémités des doigts dans l'espace cartésien :

$$\begin{aligned} W_j &= {}^K W_j + {}^B W_j \\ &= f_j(H_j(\theta)) + B_j \cdot J_j(\theta) \cdot \dot{\theta} \end{aligned} \quad (\text{Eq. III-75})$$

A partir du torseur d'efforts, de la matrice Jacobienne et du modèle géométrique direct, nous calculons les couples internes de chaque doigt, i.e. les couples articulaires notés $\tau_j^{\text{doigt}}(\theta)$ avec j la liaison étudiée. Nous obtenons donc la relation suivante dans l'espace cartésien :

$$\begin{aligned} \tau_j^{\text{doigt}}(\theta) &= J_j^T(\theta) \cdot W_j \\ &= J_j^T(\theta) \cdot (f_j(H_j(\theta)) + B_j \cdot J_j(\theta) \cdot \dot{\theta}) \\ &= J_j^T(\theta) \cdot f_j(H_j(\theta)) + J_j^T(\theta) \cdot B_j \cdot J_j(\theta) \cdot \dot{\theta} \end{aligned} \quad (\text{Eq. III-76})$$

Comme les doigts s'apparentent à des chaînes en série ouvertes, le couple total appliqué à l'ensemble des articulations des doigts est égal à la somme des couples articulaires. Nous obtenons donc la relation suivante :

$$\tau_{\text{Total}} = \sum_j \tau_j^{\text{doigt}}(\theta, \dot{\theta}) \quad (\text{Eq. III-77})$$

D'où la relation pour le couple total en remplaçant $\tau_j^{\text{doigt}}(\theta)$ par son expression dans l'équation III-77 :

$$\begin{aligned} \tau_{\text{Total}} &= \sum_j \tau_j^{\text{doigt}}(\theta, \dot{\theta}) \\ &= \sum_j J_j^T(\theta) \cdot f_j(H_j(\theta)) + \sum_j J_j^T(\theta) \cdot B_j \cdot J_j(\theta) \cdot \dot{\theta} \end{aligned} \quad (\text{Eq. III-78})$$

Nous connaissons donc pour chaque articulation le couple résultant (i.e. le couple moteur) des torseurs d'efforts exercés sur la totalité de la chaîne. Celui-ci permet de contrôler les doigts en position angulaire ou encore les orientations des extrémités des doigts.

5.2.3.2.3 Exemple d'application

Afin d'expliquer le principe de fonctionnement de cette loi de commande, nous l'illustrons par un exemple simplifié de chaîne cinématique.

Nous considérons un modèle de chaîne cinématique polyarticulée sans inertie. Nous faisons donc l'hypothèse qu'il n'existe que des frottements visqueux au niveau des articulations. Dans ce cas de figure, nous obtenons l'expression suivante pour le couple articulaire total :

$$\boldsymbol{\tau}_{\text{Total}} = \mathbf{B}\dot{\boldsymbol{\theta}} \quad (\text{Eq. III-79})$$

avec \mathbf{B} la matrice des coefficients de frottement articulaire telle que :

$$\mathbf{B} = \begin{bmatrix} b_1 & 0 & 0 & 0 \\ 0 & b_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & b_n \end{bmatrix} = \sum_{i,j=1}^n \delta_{ij} b_j \quad \text{avec } n \text{ le nombre d'articulations.}$$

D'où l'équation :

$$\begin{aligned} \boldsymbol{\tau}_{\text{Total}} &= \sum_j \boldsymbol{\tau}_j^{\text{doigt}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \\ \Leftrightarrow \mathbf{B}\dot{\boldsymbol{\theta}} &= \sum_j \mathbf{J}_j^T(\boldsymbol{\theta}) \cdot f_j(\mathbf{H}_j(\boldsymbol{\theta})) + \sum_j \mathbf{J}_j^T(\boldsymbol{\theta}) \cdot \mathbf{B}_j \cdot \mathbf{J}_j(\boldsymbol{\theta}) \cdot \dot{\boldsymbol{\theta}} \\ \Leftrightarrow \dot{\boldsymbol{\theta}} \left(\mathbf{B} - \sum_j \mathbf{J}_j^T(\boldsymbol{\theta}) \cdot \mathbf{B}_j \cdot \mathbf{J}_j(\boldsymbol{\theta}) \right) &= \sum_j \left(\mathbf{J}_j^T(\boldsymbol{\theta}) \cdot f_j(\mathbf{H}_j(\boldsymbol{\theta})) \right) \quad (\text{Eq. III-80}) \\ \Leftrightarrow \dot{\boldsymbol{\theta}} &= \sum_j \left(\mathbf{J}_j^T(\boldsymbol{\theta}) \cdot f_j(\mathbf{H}_j(\boldsymbol{\theta})) \right) \times \left(\mathbf{B} - \sum_j \mathbf{J}_j^T(\boldsymbol{\theta}) \cdot \mathbf{B}_j \cdot \mathbf{J}_j(\boldsymbol{\theta}) \right)^{-1} \end{aligned}$$

Nous obtenons donc les vitesses articulaires à partir des matrices Jacobiennes des chaînes arborescentes et des torseurs d'efforts exercés sur les extrémités des doigts.

Pour intégrer les équations du mouvement et obtenir les valeurs angulaires aux articulations, nous devons discrétiser l'équation III-80. Pour ce faire, il existe différentes méthodes d'intégration : celles du premier ordre ou celles du second ordre. Dans cet exemple simplifié, nous avons utilisé un schéma d'intégration de type Euler explicite dont la forme est la suivante :

$$\boldsymbol{\theta}(t+h) = \boldsymbol{\theta}(t) + \mathbf{h}\dot{\boldsymbol{\theta}}(t)$$

C'est une méthode du premier ordre qui nous permet de calculer les angles aux articulations à partir des vitesses articulaires en un pas de temps (intégration).

Nous représentons la loi de commande proposée dans cet exemple par le schéma bloc suivant (Cf. Figure III.76) :

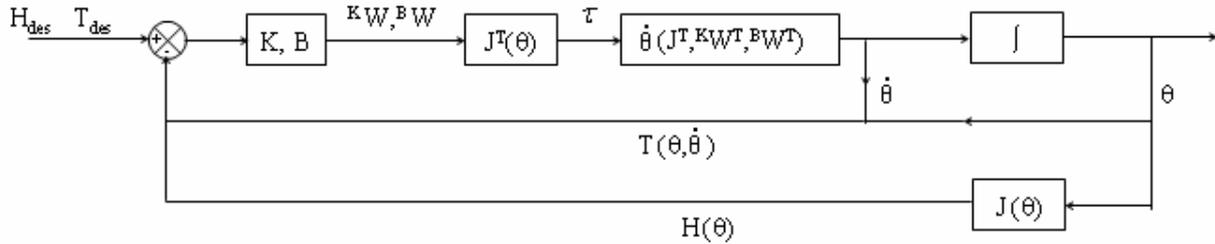


Figure III.76 : Schéma bloc pour l'asservissement d'un doigt

L'objectif de cet exemple est d'expliquer les algorithmes mis en œuvre dans ce paragraphe pour contrôler la main et les doigts virtuels. Pour que notre système d'équations modélise au mieux la main humaine, il est important de tenir compte de la masse et de l'inertie de celle-ci. Le couple total est donc fonction de la matrice d'inertie de chaque phalange.

5.3 Conclusion

Dans ce paragraphe, nous avons décrit les différents algorithmes développés pour piloter la main virtuelle à partir des données issues de la capture de mouvements. Le contrôle de la main virtuelle se décompose en deux asservissements distincts :

- Le premier est utilisé pour commander la main en position et en orientation dans l'environnement virtuel.
- Le second permet de piloter les mouvements des doigts afin de simuler des tâches de préhension.

Ces deux asservissements sont réalisés simultanément. Pour pouvoir interagir en temps réel avec l'environnement et avoir une simulation stable, il est donc essentiel de bien définir les matrices de raideur \mathbf{K} et d'amortissement \mathbf{B} .

Dans le paragraphe qui suit, nous étudions la stabilité numérique de notre modèle. Nous combinons deux méthodes distinctes – dans le domaine temporel puis dans le domaine discret – afin de caractériser le domaine de stabilité de notre asservissement. Les résultats obtenus par ces deux études ont pour objectif de vérifier les gains des contrôleurs de la main et des doigts voire de nous aider à déterminer les valeurs les plus pertinentes pour piloter la main dans la scène virtuelle.

6 Stabilité

La réussite du contrôle de la main en temps réel dépend principalement de la stabilité du modèle de main lorsqu'il interagit avec l'environnement virtuel comme par exemple lors des tâches de saisie et manipulation d'objets virtuels.

Dans ce paragraphe, nous cherchons à résoudre les problèmes complexes de la stabilité numérique. Tout d'abord, Nous donnons quelques définitions importantes qui vont nous être utiles dans la suite de ce paragraphe. Puis, nous résolvons les équations de la dynamique afin de définir les paramètres nécessaires à intégrer dans notre simulation.

6.1 Stabilité numérique

Le concept de stabilité numérique repose sur l'analyse du comportement des solutions numériques de l'équation en fonction des valeurs du pas de temps temporel d'itération noté h , autrement appelé période d'échantillonnage T . Ce dernier intervient dans la discrétisation des équations différentielles : la stabilité numérique est donc basée sur la solution discrète. C'est un problème récurrent pour toute résolution numérique d'un système dynamique.

Ainsi, pour que la solution approchée d'une équation reste bornée, positive et décroissante, i.e. qu'elle reproduise le comportement de la solution exacte, il faut imposer une condition sur h [Hoffmann 01] [Mohammadi 03]. Par exemple dans le cas de la résolution de l'équation différentielle du premier ordre $\dot{y}(t) = -\alpha y(t)$ avec $y(0) = y_0$ et $\alpha > 0$, le schéma d'Euler

explicite est stable si et seulement si $h < \frac{1}{\alpha}$ [Mohammadi 03] [Hoffmann 01]. En effet, lorsque le pas de temps h devient trop grand, l'approximation numérique ne converge plus vers la solution exacte de l'équation : elle tend généralement à osciller en prenant des valeurs de plus en plus élevées. Tandis que le schéma d'Euler implicite est inconditionnellement stable pour résoudre cette équation.

Pour étudier la stabilité de notre simulation, nous cherchons à déterminer le seuil de stabilité de notre modèle. C'est-à-dire que nous cherchons la valeur du pas d'itération au-delà de laquelle le système ne converge plus. Malheureusement ce seuil de stabilité dépend d'un très grand nombre de facteurs comme par exemple le schéma d'intégration numérique, la géométrie des objets, la valeur des masses (et des inerties), les coefficients de raideur et d'amortissement ou encore la loi mécanique modélisée. Dans la suite de ce paragraphe, nous cherchons à estimer ce seuil de stabilité en se basant sur la géométrie et sur les paramètres des contrôleurs proportionnel-dérivé mis en œuvre pour piloter la main virtuelle.

6.2 Etude de la stabilité de la simulation

Les mouvements de la main virtuelle et des objets dans la scène sont donnés par l'équation (équations de la dynamique en formulation de Lagrange). Dans notre cas de figure, nous considérons l'équation dynamique pour chaque corps de notre main virtuelle :

$$\mathbf{M} \ddot{\mathbf{X}}(t) + \mathbf{B} \dot{\mathbf{X}}(t) + \mathbf{K} \mathbf{X}(t) = \mathbf{F}(t) \quad (\text{Eq. III-81})$$

avec :

- \mathbf{M} la matrice de masse, \mathbf{B} la matrice d'amortissement, \mathbf{K} la matrice de raideur et \mathbf{F} la force exercée sur le système,
- $\ddot{\mathbf{x}}(t)$, $\dot{\mathbf{x}}(t)$ et $\mathbf{x}(t)$ respectivement les vecteurs accélération, vitesse et position à l'instant t ,
- $\mathbf{F}(t)$ le vecteur des efforts extérieurs.

6.2.1 Résolution de l'équation de la dynamique dans le domaine temporel

Afin d'expliquer la résolution du système matriciel différentiel III-81 dans le domaine temporel, nous avons recours à l'équation différentielle linéaire du second ordre non homogène correspondante :

$$m\ddot{x}(t) + b\dot{x}(t) + kx(t) = f(t) \quad (\text{Eq. III-82})$$

avec :

- m la masse, b le facteur d'amortissement, k la raideur et f la force extérieure exercée,

- $\ddot{x}(t)$, $\dot{x}(t)$ et $x(t)$ respectivement l'accélération, la vitesse et le déplacement à l'instant t .

Par définition, la solution générale d'une telle équation différentielle est une combinaison linéaire [Granjon 01] [Longchamp 06] :

- de la solution générale de l'équation sans second membre qui est de la forme $C_1 e^{r_1 x} + C_2 e^{r_2 x}$, avec C_1 et C_2 des constantes définies à partir des conditions initiales,
- d'une solution particulière de l'équation avec second membre.

6.2.1.1 Calcul des solutions de l'équation différentielle

Dans ce paragraphe, nous cherchons à résoudre l'équation différentielle III-82. Pour ce faire, nous introduisons les termes :

- ω_0 la pulsation propre du système ;
- ξ le facteur d'amortissement.

Nous écrivons l'équation différentielle III-82 sous la forme :

$$\frac{1}{\omega_0^2} \ddot{x}(t) + \frac{2\xi}{\omega_0} \dot{x}(t) + x(t) = S(t) \quad (\text{Eq. III-83})$$

En comparant les équations III-82 et III-83, nous obtenons les équivalences suivantes :

- $\omega_0^2 = \frac{k}{m}$
- $\xi = \frac{b}{2\sqrt{km}}$ ou $\xi = \frac{b}{2m\omega_0}$
- $S(t) = \frac{F(t)}{k}$

Pour définir les solutions de l'équation différentielle sans second membre, nous écrivons l'équation caractéristique associée à cette équation du mouvement. Nous obtenons la relation suivante :

$$\frac{1}{\omega_0^2} r^2 + \frac{2\xi}{\omega_0} r + 1 = 0 \quad (\text{Eq. III-84})$$

Le discriminant est donné par Δ :

$$\Delta = \frac{4}{\omega_0^2} (\xi^2 - 1) \quad (\text{Eq. III-85})$$

Nous obtenons donc trois solutions possibles selon le discriminant :

- si $\Delta > 0$, alors $\xi > 1$: les deux racines r_1 et r_2 sont réelles et simples
 $r_{1,2} = \omega_0 \left(-\xi \pm \sqrt{\xi^2 - 1} \right)$;
- si $\Delta = 0$, alors $\xi = 1$: la racine r_1 est double : $r_1 = -\omega_0$;

- si $\Delta < 0$, alors $0 < \xi < 1$: les deux racines r_1 et r_2 sont complexes conjuguées $r_{1,2} = \omega_0 \left(-\xi \pm j\sqrt{1-\xi^2} \right)$.

Ensuite, pour calculer la solution particulière de cette équation différentielle, nous remplaçons $x(t)$ dans l'équation III-82 par l'une des trois solutions décrites précédemment selon la valeur de Δ [Granjon 01] [Longchamp 06].

Nous déduisons la solution complète de l'équation du mouvement en sommant la solution de l'équation sans second membre avec la solution particulière dans chacun des cas de figure [Granjon 01] [Longchamp 06].

La résolution décrite précédemment concerne les systèmes à 1DDL. Dans notre cas de figure, la main est considérée comme un système composé de plusieurs corps rigides. Nous devons donc résoudre le système matriciel III-80 :

$$\mathbf{M} \ddot{\mathbf{X}}(t) + \mathbf{B} \dot{\mathbf{X}}(t) + \mathbf{K} \mathbf{X}(t) = \mathbf{F}(t)$$

avec \mathbf{K} la matrice de raideur, \mathbf{B} la matrice d'amortissement, \mathbf{M} la matrice des masses et $\mathbf{X} = [x_1(t) \dots x_n(t)]^T$ le vecteur position.

La résolution d'un tel système matriciel repose sur le fait que les matrices \mathbf{M} , \mathbf{B} et \mathbf{K} sont diagonalisables ou pas. Les solutions se mettent alors sous la forme d'exponentielles de matrices. Pour plus de détails, se référer à [Granjon 01] ou [Longchamp 06]

6.2.1.2 Critères de stabilité des systèmes analogiques

Pour étudier la stabilité d'un système différentiel linéaire à coefficients constants comme l'équation III-81, nous étudions les coefficients de l'équation sans second membre :

$$m \ddot{x}(t) + b \dot{x}(t) + k x(t) = 0$$

Nous nous focalisons donc sur l'équation caractéristique suivante :

$$m r^2 + b r + k = 0 \quad (\text{Eq. III-86})$$

Il existe différents critères que nous détaillons dans les paragraphes qui suivent [Granjon 01] [Longchamp 06].

6.2.1.2.1 Critère de Routh

Ce critère de stabilité est une méthode permettant de déterminer la stabilité d'un système à partir de son équation caractéristique.

Critère de Routh :

Toutes les racines de l'équation caractéristique ont leur partie réelle négative si et seulement si les éléments de la première colonne de la table de Routh ont le même signe. Dans le cas contraire, le nombre de racines à partie réelle positive est égal au nombre de changements de signes comptabilisés dans cette colonne.

Le critère de Routh écrit l'équation caractéristique d'un système sous la forme particulière :

$$R(p) = a_0 p^n + a_1 p^{n-1} + \dots + a_n$$

Nous obtenons donc la relation suivante :

$$a_0 p^2 + a_1 p + a_2 = 0 \Leftrightarrow m r^2 + b r + k = 0$$

Les coefficients sont utilisés pour remplir un tableau selon une loi précise. Ainsi, nous obtenons à partir de l'équation III-85 le tableau de Routh suivant :

R^2	m ($=a_0$)	k ($=a_2$)
R^1	b ($=a_1$)	0
R^0	k ($=(a_1 a_2 - a_3 a_0)/a_1$)	0

Dans notre cas de figure, les valeurs m , b et k sont toutes positives et elles sont donc toutes du même signe. Ainsi, d'après le critère de Routh notre système est stable car toutes les racines de l'équation ont leur partie réelle négative.

6.2.1.2.2 Critère de Hurwitz

Ce critère de stabilité est une autre méthode pour déterminer si toutes les racines de l'équation caractéristique ont leur partie réelle négative ou non. Dans ce cas de figure, on utilise les déterminants formés à partir des coefficients de l'équation caractéristique.

Critère de Hurwitz :

Toutes les racines de l'équation caractéristique ont leur partie réelle négative si et seulement si le déterminant Δ_i est positif pour $i=1, 2, \dots, n$

Le critère de Hurwitz écrit l'équation caractéristique d'un système sous la forme particulière :

$$H(p) = a_n p^n + a_{n-1} p^{n-1} + \dots + a_0$$

Nous obtenons donc la relation suivante :

$$a_2 p^2 + a_1 p + a_0 = 0 \Leftrightarrow m r^2 + b r + k = 0$$

Nous calculons le déterminant de l'équation caractéristique :

$$\Delta = b^2 - 4km \quad (\text{Eq. III-87})$$

Hurwitz a montré que la condition nécessaire et suffisante pour qu'il n'existe aucune racine à partie réelle positive est que le déterminant soit positif. Dans notre cas de figure, pour qu'il n'y ait pas d'instabilité, il faut que :

$$\begin{aligned} \Delta \geq 0 &\Leftrightarrow b^2 - 4km \geq 0 \\ &\Leftrightarrow b \geq 2\sqrt{km} \end{aligned} \quad (\text{Eq. III-88})$$

6.2.1.3 Exemple d'application

A partir des relations précédentes et des paramètres d'échantillonnage, nous proposons de déterminer la raideur et l'amortissement de notre asservissement. Pour nos simulations, nous avons choisi une période d'échantillonnage $T_e = 3 \text{ ms}$. Nous obtenons alors la fréquence d'échantillonnage f_e par :

$$f_e = \frac{1}{T_e} = \frac{1}{0,003} = 333,33 \text{ Hz}$$

La pulsation d'échantillonnage ω_e est égale à :

$$\omega_e = 2 \pi f_e = 2094,37 \text{ rad.s}^{-1}$$

Le théorème de Shannon nous impose une pulsation ω_e telle que :

$$\omega_e > 2 \times \omega_0$$

Mais en pratique, une valeur nettement plus élevée de la pulsation d'échantillonnage est nécessaire car aucun signal physique ne possède une bande passante bien délimitée : l'hypothèse du théorème de Shannon ne peut être rigoureusement respectée [Granjon 01] [Longchamp 06].

La règle adoptée par les automaticiens en matière de définition de pulsation d'échantillonnage consiste en fait à choisir une valeur pour ω_e telle que :

$$6 \times \omega_0 < \omega_e < 25 \times \omega_0$$

En règle générale, une valeur de ω_e est choisie dans l'intervalle $[10\omega_0; 20\omega_0]$ [Granjon 01] [Longchamp 06]. Comme dans notre cas de figure nous connaissons déjà la pulsation d'échantillonnage ω_e , nous pouvons en déduire la pulsation propre de notre système. Nous obtenons les valeurs suivantes pour ω_0 :

$$104,72 \text{ rad.s}^{-1} \leq \omega_0 \leq 209,44 \text{ rad.s}^{-1} \quad (\text{Eq. III-89})$$

Pour notre simulation, nous avons choisi $\omega_0 = 104,72 \text{ rad.s}^{-1}$. Nous sommes donc en mesure d'évaluer les paramètres k et b de notre couplage virtuel.

Nous obtenons une raideur k égale à :

$$\begin{aligned} k &= m\omega_0^2 \\ &= 1 \times 104,72^2 \\ &= 10966 \text{ N.m}^{-1} \end{aligned} \quad (\text{Eq. III-90})$$

Pour limiter le dépassement, nous choisissons une valeur pour le gain statique : $\xi = 1$.

Nous en déduisons donc l'amortissement b tel que :

$$\begin{aligned} b &= 2m\omega_0\xi \\ &= 2 \times 1 \times 104,72 \times 1 \\ &= 209,44 \text{ N.s.m}^{-1} \end{aligned} \quad (\text{Eq. III-91})$$

Les valeurs de ω_0 et ξ ont été choisies empiriquement et elles permettent d'obtenir les paramètres suivants pour l'asservissement : $k = 10\,966 \text{ N.m}^{-1}$ et $b = 209,44 \text{ N.s.m}^{-1}$.

Nous vérifions les critères de stabilité présentés dans les paragraphes 6.2.1.2.1 et 6.2.1.2.2 :

- les paramètres k , b et m étant tous positifs, ils satisfont donc bien au critère de Routh.
- pour vérifier le critère de Hürwitz, nous devons respecter la relation III-88. Nous obtenons donc : $2\sqrt{km} = 2\sqrt{10966 \times 1} = 209,44$. Ce qui correspond bien à la valeur de l'amortissement b .

En choisissant ces valeurs pour ω_0 et ξ , nous constatons que les paramètres k et b précédemment calculés respectent bien les critères de stabilité de Routh et de Hürwitz. Nous pouvons donc en conclure que notre système est stable.

Nous présentons dans le Tableau III.4 les résultats obtenus lors de l'analyse de la stabilité de l'équation de la dynamique dans le domaine continu quand ω_0 parcourt l'intervalle donné par l'équation III-89 :

ω_0 (rad/s)	k (N/m)	b (N.s/m)	Critère de Hurwitz (b)
104,72	10966,01	209,44	209,44
110,23	12150,70	220,46	220,46
116,35	13538,28	232,71	232,71
123,20	15177,87	246,40	246,40
130,90	17134,39	261,80	261,80
139,62	19495,12	279,25	279,25
149,60	22379,61	299,20	299,20
161,11	25955,05	322,21	322,21
174,53	30461,13	349,06	349,06
190,40	36251,27	380,80	380,80
209,44	43864,03	418,87	418,87

Tableau III.4 : Résultats des analyses de stabilité en continu

Nous constatons que le critère de Hürwitz est toujours vérifié. Par conséquent, notre système est toujours stable.

Nous avons également testé les valeurs k (eq. III-90) et b (eq. III-91) sur la plateforme de Réalité Virtuelle pour asservir en position la main virtuelle à la main de l'utilisateur. Le résultat était correct mais nous avons constaté que l'asservissement devenait instable lorsque les mouvements exécutés par l'opérateur réel étaient rapides et plus particulièrement en rotation.

6.2.2 Résolution de l'équation de la dynamique dans le domaine discret

Pour vérifier la stabilité de notre système, il est également possible de résoudre l'équation III-80 dans le domaine discret.

Dans ce paragraphe, nous résolvons les équations de la dynamique par une méthode des différences finies. Nous présentons ensuite le critère de stabilité en temps discret.

6.2.2.1 Discrétisation de l'équation différentielle

Nous proposons de discrétiser directement les équations de la mécanique (i.e. équation III-80) par les différences finies (méthode multipas) [Hoffman 01] [Mohammadi 03].

Nous choisissons les approximations centrées des dérivées première et seconde suivantes :

$$\dot{\mathbf{X}}(t) = \frac{\mathbf{X}(t+h) - \mathbf{X}(t-h)}{2h} + O(h^2) \quad (\text{Eq. III-92})$$

$$\ddot{\mathbf{X}}(t) = \frac{\mathbf{X}(t+h) - 2\mathbf{X}(t) + \mathbf{X}(t-h)}{h^2} + O(h^2) \quad (\text{Eq. III-93})$$

En remplaçant les dérivées de l'équation III-80 par leurs approximations III-92 et III-93 et en tronquant à l'ordre 2, nous obtenons le schéma d'approximation suivant de l'équation de la dynamique en différences finies :

$$\begin{aligned} \mathbf{M} \frac{\mathbf{X}(t+h) - 2\mathbf{X}(t) + \mathbf{X}(t-h)}{h^2} + \mathbf{B} \frac{\mathbf{X}(t+h) - \mathbf{X}(t-h)}{2h} + \mathbf{K} \mathbf{X}(t+h) &= \mathbf{F}(t+h) \\ \Leftrightarrow \mathbf{X}(t+h) \left[\frac{\mathbf{M}}{h^2} + \frac{\mathbf{B}}{2h} + \mathbf{K} \right] - \frac{2\mathbf{M}}{h^2} \mathbf{X}(t) + \mathbf{X}(t-h) \left[\frac{\mathbf{M}}{h^2} - \frac{\mathbf{B}}{2h} \right] &= \mathbf{F}(t+h) \\ \Leftrightarrow [2\mathbf{M} + \mathbf{B}h + 2\mathbf{K}h^2] \mathbf{X}(t+h) - 4\mathbf{M} \mathbf{X}(t) + [2\mathbf{M} - \mathbf{B}h] \mathbf{X}(t-h) &= 2h^2 \mathbf{F}(t+h) \end{aligned} \quad (\text{Eq. III-94})$$

Ce schéma est un schéma explicite d'ordre 2 en temps par développement de Taylor. Il donne une formule explicite de calcul de la solution au temps (t+h) en fonction des valeurs de la solution au pas de temps précédent.

Dans la suite du paragraphe, nous étudions les conditions de stabilité de ce schéma numérique.

6.2.2.2 Etude de la stabilité du système discret

Dans notre cas de figure, il existe deux méthodes pour analyser les conditions de stabilité de l'équation du mouvement avec ce type de schéma de discrétisation :

- une étude par l'analyse des coefficients de la suite numérique représentant l'équation caractéristique $\alpha_2 u_{n+1} + \alpha_1 u_n + \alpha_0 u_{n-1} = 0$,
- une étude par le critère de Jury de l'équation caractéristique.

Nous avons fait le choix d'analyser la stabilité par le critère de Jury. Nous étudions donc l'équation caractéristique de l'équation matricielle III-94 :

$$\alpha_2 z^2 + \alpha_1 z + \alpha_0 = 0 \quad (\text{Eq. III-95})$$

avec :

- $\alpha_2 = 2\mathbf{M} + \mathbf{B}h + 2\mathbf{K}h^2$
- $\alpha_1 = -4\mathbf{M}$
- $\alpha_0 = 2\mathbf{M} - \mathbf{B}h$

Le critère de Jury est un critère de stabilité pour les systèmes discrets qui est analogue au critère de Routh [Granjon 01] [Longchamp 06].

Critère de Jury :

La zone de stabilité est définie par le cercle unité $|z|=1$ du plan complexe. Une condition nécessaire et suffisante de stabilité du système est que toutes les racines de l'équation caractéristique aient un module inférieur à 1. Cela garantit que la réponse du système est décroissante dans le temps.

Le critère de Jury écrit l'équation caractéristique d'un système sous la forme particulière :

$$J(z) = a_n z^n + a_{n-1} z^{n-1} + \dots + a_1 z + a_0$$

Les conditions nécessaires et suffisantes pour que les modules des racines de l'équation $J(z)=0$ soient inférieurs à 1 sont les suivantes :

1 ^{er} test	2 ^{ème} test	...	Dernier test
$J(1) > 0$	$ a_0 < a_n$	$ c_0 > c_{n-2} $	$ r_0 > r_3 $
$\begin{cases} J(-1) > 0 & \text{pour } n \text{ pair} \\ J(-1) < 0 & \text{pour } n \text{ impair} \end{cases}$	$ b_0 > b_{n-1} $...	$ s_0 > s_2 $

Avec :

$$b_k = \begin{vmatrix} a_0 & a_{n-k} \\ a_n & a_k \end{vmatrix} \quad \dots \quad s_1 = \begin{vmatrix} r_0 & r_2 \\ r_3 & r_1 \end{vmatrix}$$

$$c_k = \begin{vmatrix} b_0 & b_{n-1-k} \\ b_{n-1} & b_k \end{vmatrix} \quad s_0 = \begin{vmatrix} r_0 & r_3 \\ r_3 & r_0 \end{vmatrix} \quad s_2 = \begin{vmatrix} r_0 & r_1 \\ r_3 & r_2 \end{vmatrix}$$

Ces coefficients sont utilisés pour remplir un tableau selon une loi précise. Ainsi, nous obtenons le tableau de Jury suivant à l'ordre 2 :

1	a_0	a_1	a_2
2	a_2	a_1	a_0
3	$b_0 = a_0^2 - a_2^2$	$b_1 = a_0 a_1 - a_2 a_1$	
4	$b_1 = a_0 a_1 - a_2 a_1$	$b_0 = a_0^2 - a_2^2$	
5	$c_0 = (a_0^2 - a_2^2)^2 - (a_0 a_1 - a_2 a_1)^2$		

En respectant le formalisme de Jury, nous obtenons la relation suivante :

$$a_2 z^2 + a_1 z + a_0 = 0 \Leftrightarrow \alpha_2 z^2 + \alpha_1 z + \alpha_0 = 0$$

Nous vérifions tout d'abord les 2 premières conditions de Jury. Nous obtenons donc :

$$J(1) = \alpha_2 + \alpha_1 + \alpha_0$$

- $= 2M + Bh + 2Kh^2 - 4M + 2M - Bh$
- $= 2Kh^2 > 0$

Cette relation est toujours vérifiée.

$$J(-1) = \alpha_2 - \alpha_1 + \alpha_0$$

- $= 2\mathbf{M} + \mathbf{B}h + 2\mathbf{K}h^2 + 4\mathbf{M} + 2\mathbf{M} - \mathbf{B}h$
 $= 8\mathbf{M} + 2\mathbf{K}h^2 > 0$

Cette relation est toujours vérifiée.

Ces deux conditions étant satisfaites, nous pouvons donc construire la table de Jury. Nous obtenons à partir des coefficients de l'équation caractéristique III-95 le tableau suivant :

1	$2\mathbf{M}-\mathbf{B}h$	$-4\mathbf{M}$	$2\mathbf{M}+\mathbf{B}h+2\mathbf{K}h^2$
2	$2\mathbf{M}+\mathbf{B}h+2\mathbf{K}h^2$	$-4\mathbf{M}$	$2\mathbf{M}-\mathbf{B}h$
3	$-4h(2\mathbf{B}\mathbf{M}+2\mathbf{K}\mathbf{M}h+\mathbf{B}\mathbf{K}h^2+\mathbf{K}^2h^3)$	$8\mathbf{M}h(\mathbf{B}+\mathbf{K}h)$	
4	$8\mathbf{M}h(\mathbf{B}+\mathbf{K}h)$	$-4h(2\mathbf{B}\mathbf{M}+2\mathbf{K}\mathbf{M}h+\mathbf{B}\mathbf{K}h^2+\mathbf{K}^2h^3)$	
5	$16\mathbf{K}h^4(\mathbf{B}+\mathbf{K}h)(4\mathbf{M}\mathbf{B}+4\mathbf{K}\mathbf{M}h+\mathbf{B}\mathbf{K}h^2+\mathbf{K}^2h^3)$		

Il nous reste à vérifier deux contraintes du test de Jury :

- $|a_0| < a_2 \Leftrightarrow |2\mathbf{M} - \mathbf{B}h| < 2\mathbf{M} + \mathbf{B}h + 2\mathbf{K}h^2$

Nous avons donc deux cas de figure :

$$\begin{cases} 2\mathbf{M} - \mathbf{B}h < 2\mathbf{M} + \mathbf{B}h + 2\mathbf{K}h^2 \\ \mathbf{B}h - 2\mathbf{M} < 2\mathbf{M} + \mathbf{B}h + 2\mathbf{K}h^2 \end{cases} \Leftrightarrow \begin{cases} \mathbf{B}h + \mathbf{K}h^2 > 0 \\ 2\mathbf{M} + \mathbf{K}h^2 > 0 \end{cases}$$

Ces relations sont toujours vérifiées car \mathbf{M} , \mathbf{B} , \mathbf{K} et h sont positifs.

$$|b_0| > |b_1| \Leftrightarrow \left| -4h(2\mathbf{B}\mathbf{M} + 2\mathbf{K}\mathbf{M}h + \mathbf{B}\mathbf{K}h^2 + \mathbf{K}^2h^3) \right| > |8\mathbf{M}(\mathbf{B}h + \mathbf{K}h^2)|$$

- $\Leftrightarrow 4h(2\mathbf{B}\mathbf{M} + 2\mathbf{K}\mathbf{M}h + \mathbf{B}\mathbf{K}h^2 + \mathbf{K}^2h^3) > 8\mathbf{M}(\mathbf{B}h + \mathbf{K}h^2)$
 $\Leftrightarrow \mathbf{B}h^3 + \mathbf{K}h^4 > 0$

Cette relation est toujours vraie.

Comme toutes les conditions du test de Jury sont satisfaites, toutes les racines de l'équation caractéristique ont leurs affixes à l'intérieur du cercle unité. Nous en concluons donc que le système est stable.

6.3 Conclusion

La condition de stabilité sur l'amortissement donnée par l'équation III-88 et les résultats des tests de Jury montrent que les paramètres choisis permettent d'asservir notre modèle de main de manière stable.

Les études de la stabilité dans les domaines continu et discret nous ont également permis de mettre en évidence la dépendance des conditions de stabilité de notre modèle par rapport aux différents paramètres dynamiques (la raideur, l'amortissement, la masse, etc.).

L'expérience acquise au cours des applications réalisées sur la plateforme de l'INRS a par ailleurs mis en évidence les limites de nos asservissements dans certains cas de figure. Afin d'obtenir un meilleur contrôle des mouvements en rotation, il serait intéressant de modifier le schéma de contrôle Proportionnel-Dérivé (PD) en un schéma de type Proportionnel, Intégral et Dérivé (PID).

Après avoir étudié la stabilité de notre modèle, nous décrivons dans le paragraphe suivant les différentes variables et méthodes développées pour simuler les interactions Homme-machine en environnement virtuel avec une main virtuelle rigide.

7 Mise en oeuvre

Le modèle géométrique de la main virtuelle et les algorithmes des asservissements décrits dans ce chapitre ont été programmés en langage C/C++. Ils sont disponibles sous la forme d'une bibliothèque appelée **VirtualHand**.

Dans cette section, nous présentons les informations concernant la structure de données adoptée pour cette bibliothèque ainsi que les principales classes utilisées. Ces dernières regroupent deux fonctions spécifiques :

- la construction de la scène et du modèle,
- les fonctions des contrôleurs de la main pour l'asservissement.

7.1 Structure de données pour la construction de l'environnement

Les principales classes mises en oeuvre pour la construction de la scène et du modèle de main sont :

- `st_VHvortexData` : cette classe représente la scène virtuelle (*universe*). Elle contient les liens vers le logiciel de simulation physique VortexTM ;
- `st_VHcreatedObjects` : cette classe donne la structure du modèle de main ;
- `st_hand` : cette classe contient toutes les informations nécessaires – données et fonctions - à la construction du modèle géométrique de la main (modèle physique). Celui-ci est constitué de corps rigides (*body*), de modèles (*model*) et d'articulations (*joint*) qui sont stockés dans des tables de hachage afin de permettre un accès rapide à toutes ces données ;
- `st_virtualHandState` : cette classe décrit les états de la main virtuelle (phase d'initialisation, phase de capture de mouvements, etc.) ;
- `st_offsetSettings` : cette classe initialise la scène et la position de la main dans l'environnement de la presse plieuse.

Une description plus détaillée de ces classes est donnée sur la Figure III.77 ci-dessous :

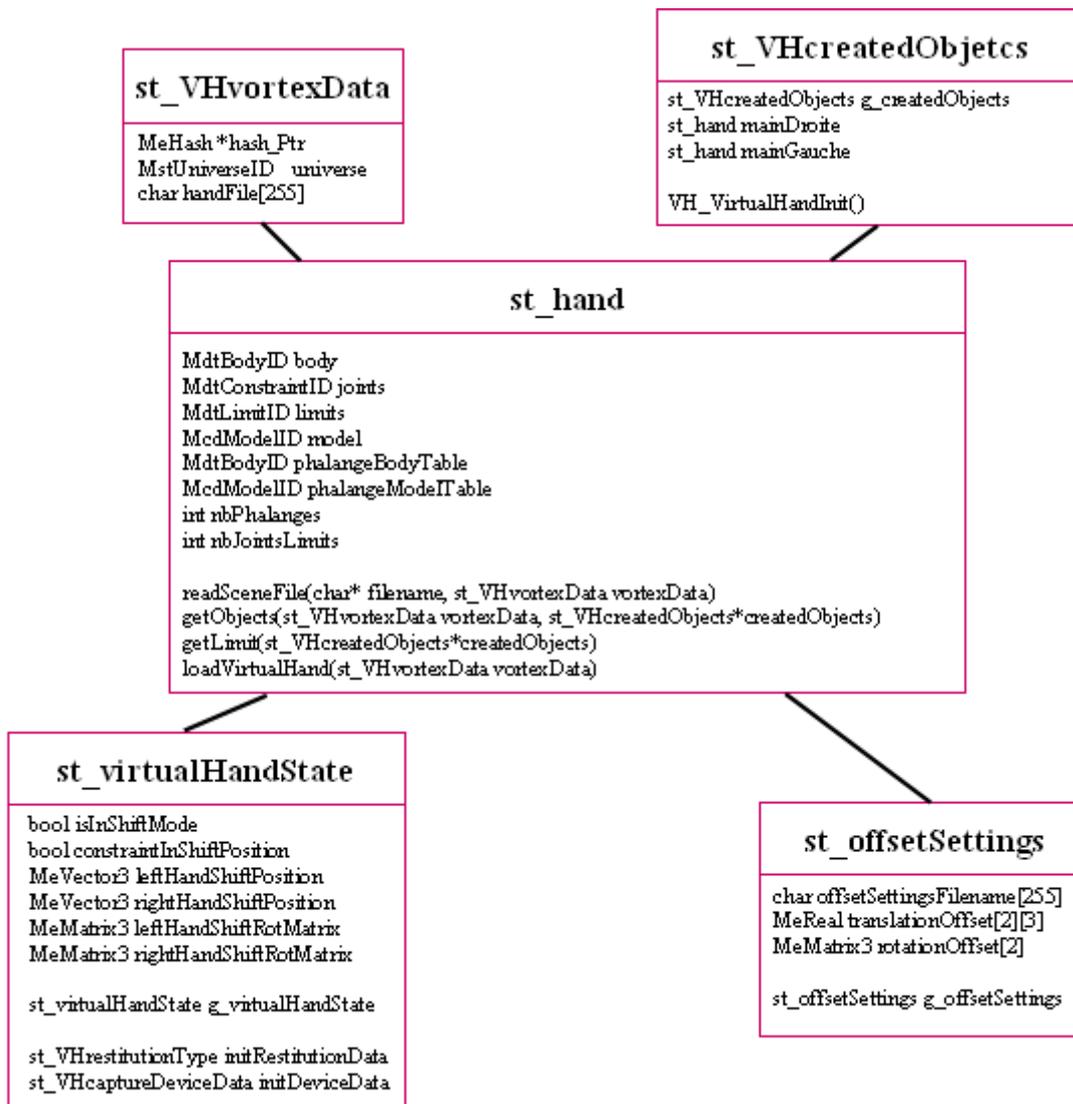


Figure III.77 : Structure de données utilisée pour la scène et le modèle de main

Grâce à ces variables et fonctions, nous pouvons construire le modèle géométrique de notre main virtuelle. Il est donc possible de représenter l'utilisateur dans la scène virtuelle et plus précisément dans l'environnement de la presse virtuelle.

Les variables et fonctions nécessaires au pilotage de la main virtuelle sont expliquées dans le paragraphe suivant.

7.2 Structure de données pour les fonctions d'asservissements

Le contrôle de la main virtuelle s'effectue en deux étapes :

- le stockage des données issues de la capture de mouvements tout au long du suivi des mouvements de l'utilisateur,
- le calcul et l'application des efforts à exercer sur les mains virtuelles pour que leur comportement soit identique à celui des mains réelles observé en temps réel.

7.2.1 Les informations issues de la capture de mouvements

La main virtuelle est contrôlée à partir des mouvements de la main de l'utilisateur qui sont traqués par un système de capture de mouvements. La classe `st_VHcaptureDeviceData` permet d'initialiser la phase de capture de mouvements, c'est-à-dire de sélectionner l'interface utilisée.

Ainsi le suivi des mains de l'utilisateur peut s'effectuer par le biais :

- soit d'un système optique : dans ce cas de figure les variables et fonctions utiles sont contenues dans la classe `st_VHdataReturnFromDevice`. Nous stockons pour chacune des mains virtuelles les informations mesurées sous la forme de vecteurs position pour les déplacements en translation ou de quaternions pour les déplacements en rotation ;
- soit d'un gant de données : la classe `st_VHdataReturnFromGloveDevice` contient toutes les informations nécessaires au stockage et au traitement des données angulaires.

Trois classes supplémentaires ont été spécifiquement définies afin de tenir compte des différents systèmes optiques (`st_ViconStaticData` et `st_ActiCMStaticData`) ainsi que du contrôle direct par des raccourcis clavier (`st_ClavierStaticData`).

La structure des données utilisées pour la capture de mouvement est détaillée sur la Figure III.78.

7.2.2 Les informations issues des contrôleurs

Dans le paragraphe 5, nous avons présenté les différents asservissements développés pour piloter la main virtuelle dans l'environnement virtuel. La Figure III.79 présente les deux classes principales utilisées :

- `st_VHrestitutionType` : cette classe permet de sélectionner le type d'asservissements ;
- `st_VHrestitutionMethods` : cette classe permet d'appliquer l'asservissement choisi précédemment (système optique ou gant de données). Nous retrouvons respectivement dans les fonctions `VH_applyNewConsigneByAsserv6D()` et `VH_applyNewConsigneCyberGlove()` les méthodes détaillées dans les paragraphes 5.2.3.1 et 5.2.3.2.

Nous avons ajouté la fonction `VH_applyNewConsigneByDirectPosOri()` pour contrôler la main dans la scène virtuelle à partir des touches du clavier.

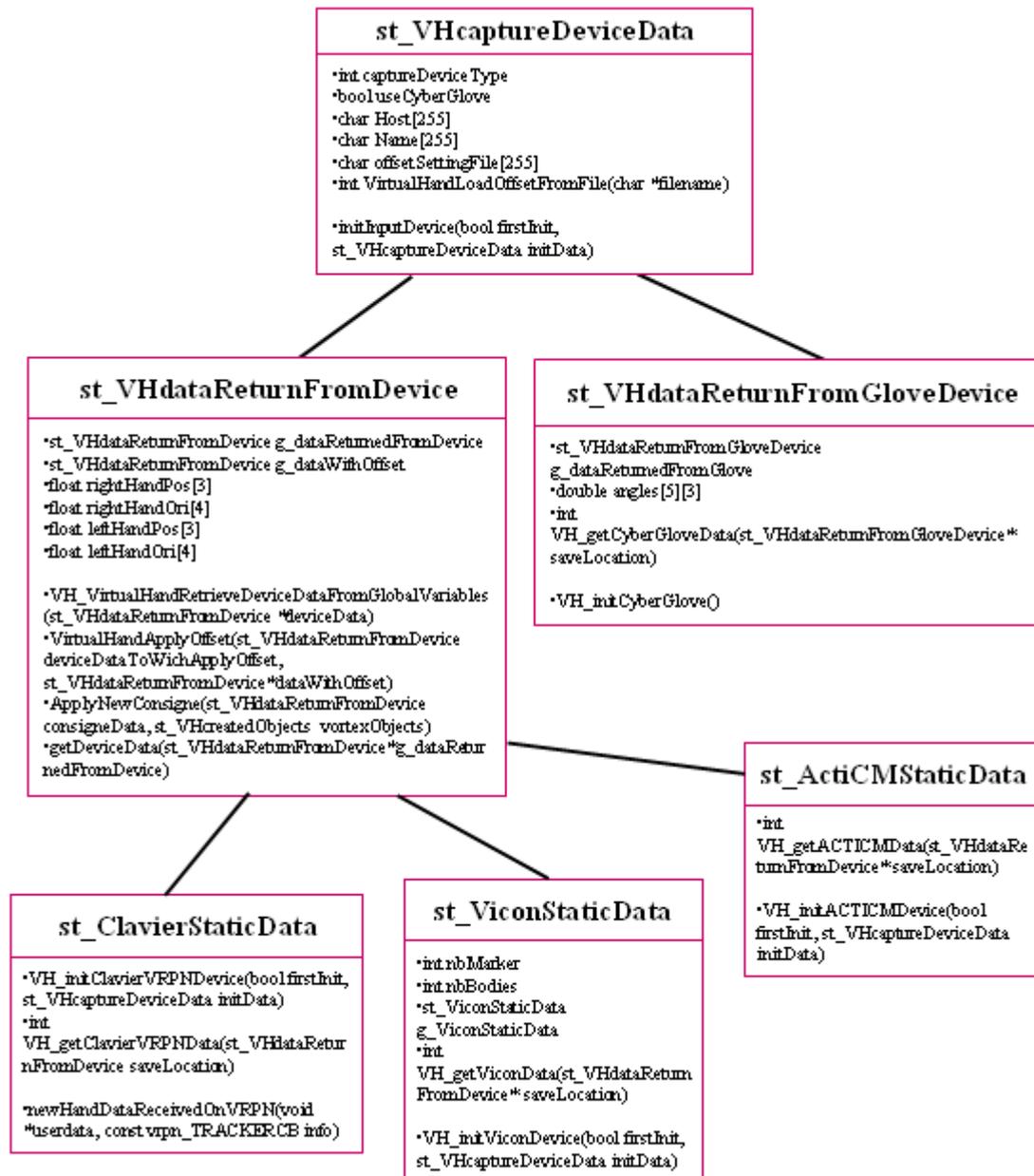


Figure III.78 : Structure de données utilisée pour la capture de mouvements



Figure III.79 : Structure de données utilisée pour les asservissements

Les algorithmes utilisés pour le traitement des données issues de la capture de mouvements ainsi que les lois de commande ont été publiés dans [Pouliquen 05b]. Bien que l'utilisation d'un contrôleur de type proportionnel-dérivé ne soit pas nouvelle en téléopération, nous avons dû développer des modalités spécifiques pour le mettre en place dans le cas d'applications de Réalité Virtuelle utilisant un système de capture de mouvements.

8 Conclusion

La description de la main humaine faite dans le Chapitre I a mis en évidence la complexité de cet organe. Au regard des caractéristiques de la main humaine et des modèles existants décrits dans le Chapitre II, nous avons fait le choix de développer un modèle de main multicouches afin de simuler des interactions Homme-machine en environnement virtuel. Dans ce chapitre, nous avons présenté la base de ce modèle : i.e. une chaîne arborescente polyarticulée rigide qui tient compte des contraintes biomécaniques statiques afin de modéliser le squelette.

Après l'étude du modèle géométrique et du modèle cinématique de notre main virtuelle, nous avons détaillé les méthodes mises en œuvre pour actionner cette main dans l'environnement virtuel. Suite au choix de la capture de mouvements pour piloter la main virtuelle, nous avons dû définir des lois de commande appropriées. Pour ce faire, nous avons donc développé trois contrôleurs de type proportionnel-dérivé pour la main et pour les doigts virtuels :

- un asservissement 6D pour piloter la main dans la scène (position et orientation),

- un asservissement articulaire pour contrôler les doigts virtuels à partir des positions angulaires mesurées par un gant de données,
- un asservissement cartésien basé sur la position cartésienne des phalanges distales, la configuration de la paume et la chaîne cinématique des doigts pour contrôler les doigts virtuels.

Ces asservissements permettent de calculer les efforts, couples et/ou forces, à appliquer aux objets virtuels afin de réduire les erreurs en position et en vitesse qui existent entre la main réelle et la main virtuelle. Ils sont équivalents à des couplages de type ressort-amortisseur entre la main de l'utilisateur et la main virtuelle.

Nous avons testé les deux types d'asservissement pour les doigts dans l'environnement de la presse plieuse. Dans les deux cas de figure, l'utilisateur a pu interagir avec la presse virtuelle ou la tôle virtuelle. Au final, nous avons opté pour l'asservissement cartésien car il offre une plus grande flexibilité en étant adaptable à tout type de gant de données ainsi qu'à toute forme de main (comme par exemple la main d'un enfant ou celle d'un adulte) sans pour autant modifier les équations utilisées.

Nous avons également étudié la stabilité numérique de notre simulation dans le domaine continu et dans le domaine discret. En choisissant un pas de temps h petit, notre système est stable par les critères de Routh-Hürwitz et de Jury. Cette étude a également mis en évidence le lien entre les coefficients des différents asservissements – raideur et amortissement – et les caractéristiques physiques de notre main virtuelle comme par exemple l'inertie.

Les modèles géométrique et cinématique de la main virtuelle ainsi que les lois de commande sont contenus dans la bibliothèque **VirtualHand** présentée au paragraphe précédent. Cette bibliothèque programmée en langage C/C++ propose un modèle de main qui permet à l'utilisateur de se déplacer dans et d'interagir avec la scène virtuelle en temps réel de manière réaliste. Dans le Chapitre V, nous présenterons des applications utilisant la bibliothèque **VirtualHand**.

CHAPITRE IV : Proposition d'un modèle de préhension

Dans les chapitres précédents, nous avons décrit le modèle de main proposé ainsi que les algorithmes utilisés pour l'animation de la main en temps réel. Dans le cas de la simulation de tâches manuelles pour des applications en Réalité Virtuelle, nous nous intéressons aux déformations résultant du contact entre les doigts et les objets saisis et manipulés.

1 Introduction

Comme nous l'avons vu dans le Chapitre I, les propriétés mécaniques des coussinets et de la pulpe permettent de saisir et de manipuler des objets de manière stable grâce au frottement et aux déformations au niveau des zones de contact. Afin d'obtenir une simulation réaliste de la main, nous proposons un modèle de préhension dédié à la Réalité Virtuelle. Il consiste à modéliser les coussinets phalangiens et la pulpe des doigts de la main humaine par des coussinets déformables placés sur la main virtuelle rigide décrite précédemment. Comme le logiciel de simulation physique utilisé ne traite que des objets rigides, nous avons besoin de mettre en place une méthode de détection et de gestion de collision pour traiter le cas de collisions entre objets rigides et déformables. Nous intégrerons également un modèle de déformation pour visualiser les actions des forces de pression exercées sur les mains virtuelles. La réalisation de notre moteur physique pour objets rigides et déformables, ainsi que la construction de notre modèle de préhension seront réalisées en trois étapes :

- la détection de collision : i.e. l'identification des points de contact exacts entre la main virtuelle et l'environnement ;
- la gestion de la collision : i.e. le calcul des paramètres de ce contact (normale, distance, coordonnées, etc.) ainsi que ses caractéristiques (contact avec ou sans frottement) ;
- la mise à jour du modèle déformable : i.e. le calcul des efforts de pression qui s'exercent sur la main virtuelle et qui la déforment.

Les paragraphes qui suivent présentent ces trois étapes.

2 La détection de collision

La détection de collision est étudiée par de nombreuses équipes dans divers domaines comme par exemple l'animation graphique, la simulation physique, la robotique pour la planification de trajectoires, les applications médicales et les applications en Réalité Virtuelle utilisant ou non le retour d'efforts [Lin 96], [Meseure 03], [Redon 04] et [Teschner 05].

Dans notre cas de figure, nous nous intéresserons plus particulièrement à la détection de collision entre objets polyédriques rigides et déformables. En effet, nous utilisons des objets convexes maillés qui deviennent parfois concaves après déformations.

Dans ce paragraphe, nous présentons les méthodes existantes pour déterminer les objets susceptibles d'entrer en contact ainsi que celles définissant le contact exact sur ces objets.

Puis, nous présenterons les approches qui permettent d'accélérer les tests d'interférences dans un environnement dynamique. De plus, dans notre contexte il est important de pouvoir détecter les collisions entre des objets qui se déforment au cours de leur mouvement. Par conséquent, nous présenterons les caractéristiques que doit respecter notre algorithme pour détecter les collisions entre une main déformable et un objet/environnement rigide en temps réel.

2.1 La détection du contact

Dans cette étape, on opère les détections entre paires d'objets (i.e. primitives bas niveau) avec la localisation du contact et l'identification de ses paramètres tels que les triangles en contact ou les normales au contact. Cette phase est appelée « noyau » ou « test d'interférence statique ». Ainsi, vérifier que deux objets ne sont pas en collision revient à trouver une distance ou un plan les séparant.

2.1.1 Le calcul de distances

Les algorithmes de calcul de distances commencent généralement par déterminer les plus proches éléments caractéristiques (sommet, arête ou facette) de deux polyèdres. On calcule alors la distance euclidienne qui les sépare. Si cette distance est négative, les objets s'interpénètrent. Trois méthodes existent :

- l'approche de Gilbert, Johnson et Canny [Gilbert 88], dite GJK : cet algorithme calcule la distance entre les enveloppes convexes de deux polyèdres en se basant sur la différence de Minkowski. Il donne la valeur de cette distance dans le cas où les deux enveloppes sont séparées et une approximation de la distance d'interpénétration dans le cas contraire. Ce type d'algorithmes a connu diverses améliorations (*Enhanced GJK* dite EGJK [Cameron 97], JL [Joukhadar 96], SOLID [Van den Bergen 99]).
- l'approche de Dobkin et Kirkpatrick [Dobkin 90], dite DK : ils proposent de construire une suite d'approximations de plus en plus grossières des polyèdres - i.e. une représentation hiérarchique - pour évaluer par itération la distance.
- l'approche de Lin et Canny [Lin 91], dite LC : cet algorithme calcule les deux plus proches éléments caractéristiques mais en utilisant des informations sur la topologie de voisinage. Ainsi, il se base sur une découpe en régions de Voronoï de l'espace environnant chaque polyèdre. Il a d'ailleurs été baptisé *Voronoi Marching*. Diverses améliorations ont également été proposées pour optimiser le temps de calcul (V-Clip [Mirtich 98]).

2.1.2 La recherche d'un plan séparateur

Deux objets ne sont pas en collision s'ils se situent de part et d'autre d'un plan. Les approches possibles construisent ce plan de différentes façons comme celle proposée par Van den Bergen qui consiste à calculer une borne minimale de la distance de séparation [Van den Bergen 99]. Lors d'un mouvement, Baraff propose d'exploiter la cohérence temporelle en testant d'abord le plan calculé au pas précédent [Baraff 90]. De ce fait, la recherche des plans séparateurs est généralement plus rapide que les méthodes d'évaluation des distances.

2.1.3 Conclusion

Les algorithmes décrits précédemment sont coûteux car ils ont une complexité quadratique. Comme les environnements virtuels comportent de nombreux objets, le nombre de couples d'objets à examiner a une complexité en $O(n^2)$, avec n le nombre d'objets de la scène. Il est

donc nécessaire de mettre en œuvre des techniques d'accélération pour respecter la contrainte de temps réel. Nous les détaillons dans le paragraphe suivant.

2.2 La recherche de proximité et la détection approximative du contact

En général, deux méthodes sont mises en œuvre : les volumes englobants ou la partition de l'espace. Ainsi, la détection de collision est organisée en pipeline [Meseure 03] [Teschner 05] :

- la phase de recherche de proximité, appelée *broad phase* : c'est une étape de détection permettant de trouver quels sont les couples d'objets susceptibles d'entrer en collision, parmi tous les couples possibles. Elle réduit le nombre de tests d'interférence à effectuer.
- la phase de détection approximative, appelée *narrow-phase* : c'est une étape qui permet de déterminer rapidement les zones de collision potentielles ou au contraire les collisions impossibles pour chacun des couples. Elle réduit la complexité de la détection exacte décrite dans le paragraphe précédent.

Tout d'abord, nous présentons l'intérêt des volumes englobants et des cellules spatiales pour identifier les objets susceptibles d'entrer en collision. Ensuite, nous expliquerons la construction des hiérarchies pour définir les zones de collision possibles entre chaque couple d'objets définis.

2.2.1 Les volumes englobants

L'objet est englobé par un premier volume dont les formes peuvent être variables. On délimite ainsi une zone d'intérêt autour de l'objet virtuel par des objets géométriques simples comme une sphère, une boîte alignée sur les axes (appelée AABB pour *Axis Aligned Bounding Box*), une boîte à orientation quelconque (appelée OBB pour *Oriented Bounding Box*) ou encore un polytope à orientation discrète (appelé DOP pour *Discrete Oriented Polytope*). La Figure IV.80 présente différentes possibilités dans le cas 2D [Teschner 05] :

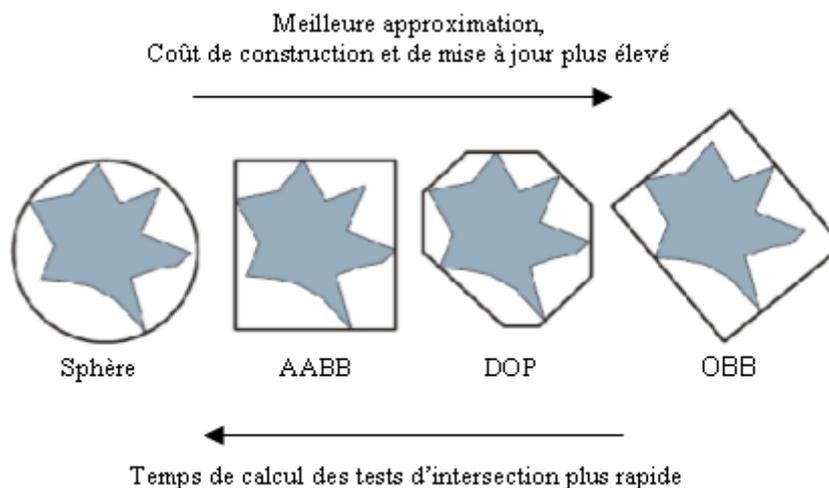


Figure IV.80 : Les différents types de volumes englobants

Quand la complexité d'un objet augmente, on utilise une hiérarchie de volumes englobants, autrement appelée « arbre ». Pour construire l'arbre, il existe deux méthodes [Lin96] :

- de bas en haut, méthode dite ascendante (*bottom-up*) : elle consiste à créer un petit volume pour chaque face/polygone, puis à les fusionner deux par deux suivant certains critères pour construire le niveau supérieur de la hiérarchie.

- de haut en bas, méthode dite descendante (*top-down*) : elle consiste à créer un volume recouvrant l'ensemble des faces/polygones, puis à le subdiviser jusqu'à ce que les nœuds des feuilles soient indivisibles (i.e. primitives de bas niveau). C'est la méthode la plus utilisée.

La Figure IV.81 illustre la construction d'un arbre de sphères [Lin96] :

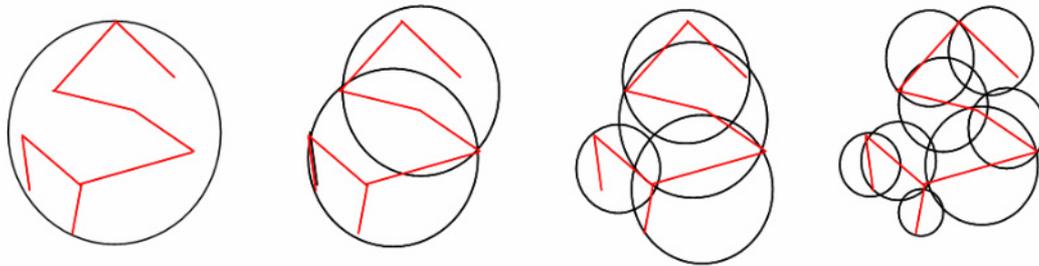


Figure IV.81 : Construction d'un arbre Sphère par la méthode descendante

Les nœuds des feuilles peuvent contenir une unique primitive géométrique ou une collection de primitives suivant les choix de conception. Pour des modèles polyèdres, on obtient une collection de triangles. Ainsi, dans notre cas de figure, nous utilisons des modèles polyédriques et nous obtenons donc des collections de triangles.

Des hiérarchies de sphères ont été décrites par Hubbard [Hubbard 96] ou Redon et al. [Redon 01], des hiérarchies d'AABB dans un repère global sont détaillées par Larsson et Möller [Larsson 01]. Une optimisation a été apportée par Van den Bergen [Van den Bergen 99] pour construire les hiérarchies d'AABB. Il propose d'avoir des coordonnées locales pour l'arbre AABB et donc de stocker la position générale et l'orientation de l'objet dans sa racine (base de l'arbre). Cette approche évite la reconstruction de l'arbre AABB à chaque déplacement de l'objet. Gottshalk et al. [Gottshalk 96] et Eberly [Eberly 98] décrivent des hiérarchies d'OBB. Les arbres de k-DOP sont présentés par Zachmann [Zachmann 00].

2.2.2 La partition de l'espace

Cette méthode consiste à créer un volume englobant tous les objets de la scène, puis à découper ce volume en n sous-volumes. On répartit les objets dans ces sous-volumes. On recommence récursivement. Cela revient à construire une hiérarchie de petits espaces contenus dans un plus grand espace. Diverses techniques existantes dans le cas 2D sont illustrées par la Figure IV.82 [Teschner 05].

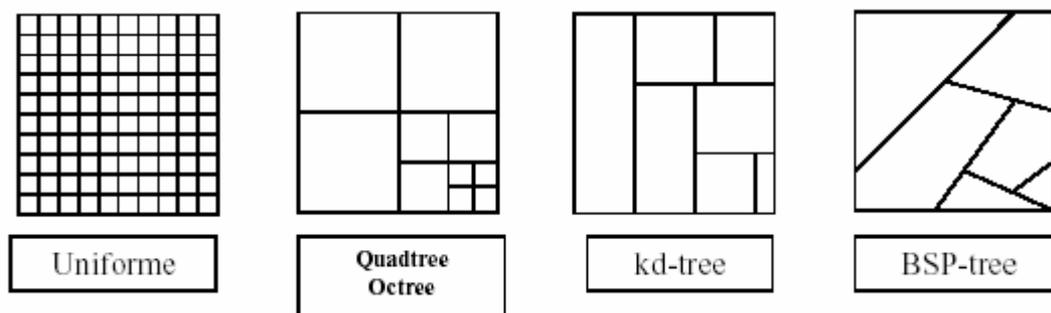


Figure IV.82 : Différents types de décomposition spatiale

Il existe différentes décompositions spatiales comme [Teschner 05] :

- les grilles uniformes : l'espace est découpé en $N \times N \times N$ cellules appelées *voxels*,
- les *quadrees* ou *octrees* (Cf. Figure IV.83) [Ponamgi 95] : l'espace est découpé en quatre ou huit voxels contigus et de même taille. Chacun de ces voxels est à son tour divisé en quatre ou huit sous-voxels. Les voxels vides ne sont pas subdivisés.
- les arbres binaires à k dimensions (appelés *k-d tree* pour *k-dimensional binary tree*) (Cf. Figure IV.83) : l'espace est divisé en cellules rectangulaires. La cellule de base correspond à un volume englobant l'objet de type AABB. De même que précédemment, les volumes englobants vides ne sont pas subdivisés.
- les arbres BSP (appelés *BSP-tree* pour *binary space partitioning tree*) : c'est un *k-d tree* généralisé. L'espace est découpé suivant des plans de coupe orientés arbitrairement.

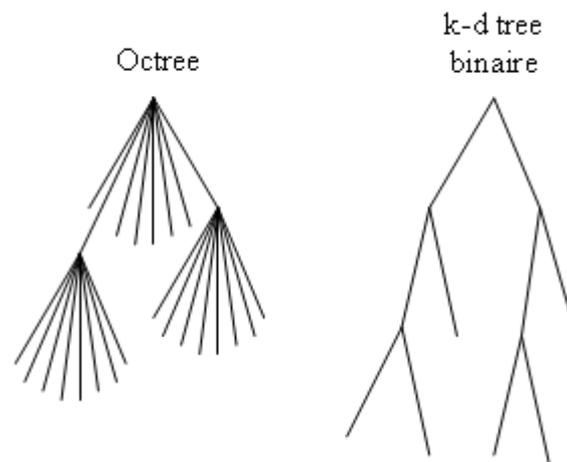


Figure IV.83 : Hiérarchies construites lors de la partition de l'espace

2.2.3 Les tests d'intersection

Une primitive peut être un volume englobant (sphère, boîte, etc.) ou un élément caractéristique de l'objet comme le triangle. Ainsi, l'interaction peut être détectée si au moins deux primitives s'interpénètrent. Ces tests se basent sur le même principe que les tests d'interférence statique : c'est-à-dire trouver un axe séparant les deux objets. Mais dans ce cas, le choix de l'axe est conditionné par le volume englobant ou l'élément caractéristique choisi.

Ainsi, pour détecter les collisions entre deux hiérarchies de volumes englobants, on inspecte dans un premier temps les volumes racines. S'il y a intersection, on descend récursivement dans la hiérarchie des deux objets ou dans celle de l'objet dont le volume englobant inspecté est le plus grand (Cf. Figure IV.84) [Teschner 05]. L'algorithme fournit alors un ensemble de facettes en probable collision.

Il existe différentes méthodes pour parcourir la hiérarchie comme par exemple celle de Quinlan [Quinlan94] qui propose de parcourir simultanément les deux arborescences de volumes englobant les objets pour calculer les distances les séparant. Cela génère une suite croissante convergeant vers la distance.

Cohen et al. ont proposé un algorithme qui permet de déterminer si deux boîtes de type AABB sont en collision : il est appelé *Sweep and Prune* [Cohen 95]. Chaque boîte est projetée sur les trois axes de l'espace : on obtient alors des intervalles. Si les projections se chevauchent sur les trois axes, alors les boîtes sont en collision. Si les intervalles ne se chevauchent pas, l'axe correspondant est appelé « axe séparateur » et les boîtes sont disjointes.

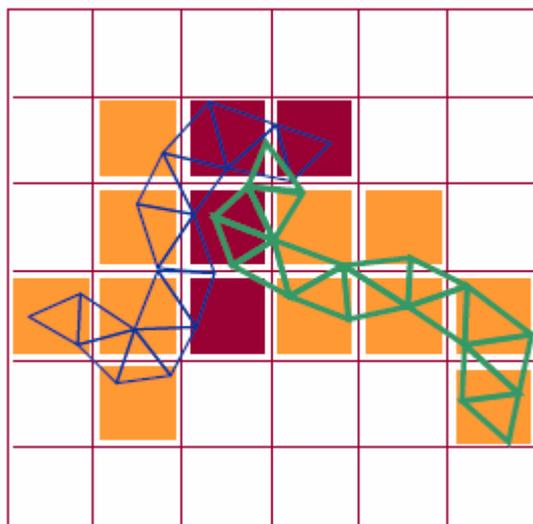


Figure IV.84 : Tests d'interférence entre deux objets

Pour toute autre boîte englobante, on utilise le test de l'axe séparateur – appelé *Separating Axis Test* (dit SAT) – développé par Gottshalk et al. [Gottshalk 96]. Si les boîtes ne sont pas en contact, on peut définir un axe séparateur. Quinze projections axiales sont suffisantes pour déterminer si les boîtes sont en contact ou pas : trois faces d'une boîte, trois faces de l'autre boîte, neuf paires de combinaisons de côtés. Mais, ce test reste relativement coûteux en raison du calcul des axes qui sont définis par des produits vectoriels. Néanmoins, tester les six premiers axes permet de trouver un axe séparateur dans la majorité des cas.

L'algorithme de Möller [Möller 97], quant à lui, permet de tester l'intersection entre deux facettes d'objets maillés (i.e. des primitives triangulaires). On obtient les triangles en contact et les deux points de contact. Des optimisations ont été apportées à cet algorithme par Guigue et Devillers [Guigue 03].

2.2.4 Conclusion

Les méthodes présentées permettent d'éliminer rapidement les parties des objets ou les zones de l'espace qui ne sont pas en collision et, d'identifier les couples de primitives – volumes englobants ou éléments caractéristiques – en collision ou les plus proches. Les volumes englobants sont largement utilisés pour la détection de collision car ils réduisent le nombre de tests de collision et ils optimisent le temps de calcul. Il est nécessaire de trouver un compromis entre l'enveloppe de l'objet et la rapidité des tests d'interférences. Ce choix du volume dépend beaucoup du contexte et plus particulièrement des formes et des mouvements des objets considérés.

Cette structure hiérarchique est plus adaptée aux corps rigides qu'aux corps déformables. En cas de déformation, il est nécessaire de recalculer les volumes en chaque noeud de l'arbre. Il faut donc se tourner vers d'autres méthodes ou faire appel à des techniques d'optimisation qui sont présentées dans le paragraphe suivant.

2.3 La détection de collision pour des objets déformables

La plupart des algorithmes de détection de collision existants concernent les objets rigides. Mais depuis la fin des années 90, de plus en plus de recherches sont réalisées pour détecter les interférences avec des objets déformables [Rodriguez 03] [Teschner 04].

2.3.1 La mise à jour des hiérarchies de volumes englobants

La déformation continue des objets dans un environnement dynamique implique la mise à jour de l'arbre. Les données stockées lors du prétraitement – comme par exemple la structure de l'arbre - doivent donc être recalculées à la volée.

Les travaux les plus récents et les plus rapides sont proposés par Van den Bergen [Van den Bergen 99] ainsi que par Larsson et Möller [Larsson 01]. Ils se basent sur des arborescences d'AABB.

Ainsi, Van den Bergen propose de remonter dans la hiérarchie de AABB en utilisant un tableau de nœuds [Van den Bergen 99]. Il s'assure que l'indice d'un nœud est toujours plus grand que celui de son fils et l'arbre est réajusté en parcourant ce tableau dans l'ordre inverse. La mise à jour se fait des feuilles vers la racine.

La méthode de Larsson et Möller est différente car elle a comme point de départ un niveau intermédiaire de la hiérarchie [Larsson 01]. Cette méthode est dite « hybride » car l'arbre est mis à jour à partir du point intermédiaire par un processus bas-haut incrémental et un processus haut-bas sélectif.

2.3.2 Le découpage spatial

Dans le cas où les corps déformables subissent des changements topologiques de l'objet, comme les découpes ou les fractures, Teschner et al. utilisent les découpages spatiaux comme les *octrees* ou les grilles uniformes avec une fonction de hachage qui se révèlent être plus intéressants que les hiérarchies de volumes englobants [Teschner 03]. En effet, ces dernières doivent être entièrement recalculées dans ce cas de figure puisque la structure a été modifiée par insertion ou suppression d'un (ou plusieurs) nœud(s).

2.3.3 L'auto-collision

Un problème supplémentaire est l'auto-collision des objets déformables : c'est-à-dire le fait que l'objet soit en collision avec lui-même. Cela nécessite de prendre en compte les liens de voisinage et la juxtaposition des parties de l'objet. Les contributions majeures viennent des travaux sur la simulation de vêtements et elles ont été apportées par Provot [Provot 97], Baraff et Witkin [Baraff 98] ainsi que Volino et Magnenat-Thalmann [Volino 00]. Mais, il est à noter une application de tests d'auto-collision pour la simulation chirurgicale présentée par Raghupathi et al. [Raghupathi 03].

2.3.4 Les champs de distances

Les champs de distances spécifient la distance minimale à une surface fermée de tous les points appartenant au champ. La distance doit avoir un signe afin de distinguer l'intérieur de l'extérieur. Cette méthode issue de la planification de trajectoires a été adaptée pour la détection de collision en temps réel par Fisher et Lin [Fisher 01] ainsi que par Furhmann et al. [Furhmann 03]. Il existe essentiellement trois approches pour calculer un champ de distances efficacement : les méthodes utilisant les diagrammes de Voronoï, les méthodes de propagation et les méthodes utilisant les hiérarchies. L'utilisation des champs de distances est avantageuse car il n'y a aucun problème de topologie et le calcul des distances pour la détection de collision est indépendant de la complexité de l'objet. Cette méthode permet de tester les collisions et les auto-collisions et elle permet donc de définir les points de contact ainsi que les distances de pénétration si elles existent.

2.3.5 Conclusion sur la détection de collision pour objets déformables

Les méthodes de détection compatibles avec les corps déformables sont plus lentes que les méthodes dédiées aux corps rigides en raison des modifications des formes des objets suite aux collisions. Certaines hypothèses permettront de négliger certaines déformations et les auto-collisions afin de diminuer le temps de calcul comme par exemple dans le cas d'objets élastiques peu déformables.

Les approches basées sur les hiérarchies de volumes englobants ont prouvé leur efficacité. Il est important de bien choisir la forme du volume englobant ainsi que la méthode de mise à jour de l'arbre afin de répondre à la volée aux changements de formes des objets dans l'environnement de simulation.

Dans le cas de collisions entre objets déformables et rigides, les champs de distances offrent les meilleures performances. Ces champs pré-calculés permettent d'avoir les distances entre les objets ainsi que les distances de pénétration qui sont essentielles pour réaliser une réponse à la collision qui soit physiquement réaliste.

2.4 La solution proposée pour la détection de collision

Dans notre cas de figure, nous traitons le cas d'objets polyédriques qui peuvent être convexes ou concaves, et déformables ou rigides. Comme l'avatar de main évolue dans l'environnement virtuel, il est donc nécessaire d'utiliser des tests d'interférence dynamiques. Les discrétisations de l'espace se révélant être coûteuses pour la mise à jour, nous avons opté pour les hiérarchies de volumes englobants. Nous avons choisi de modéliser les coussinets déformables par des géométries simples. Nous utilisons donc des sphères englobantes pour englober les coussinets sphériques et les objets de l'environnement virtuel. Par conséquent, nous construisons des hiérarchies de sphères pour la détection de collision [Marie 05].

Après avoir décrit la construction de notre arbre de sphères, nous présenterons les méthodes mises en œuvre pour le calcul de distances et la détection de collisions.

2.4.1 Les conditions expérimentales

Les algorithmes présentés dans ce paragraphe ont été programmés dans une librairie appelée **Collision** qui est liée dynamiquement à un moteur de rendu 3D OpenGL. La plate-forme d'exécution est un processeur Pentium M 1,60 GHz avec 512 Mo de RAM.

Le diagramme des classes de la librairie **Collision** est présenté en Annexe A.5.

2.4.2 La construction de l'arbre

Pour construire la sphère englobante d'un polyèdre à partir d'un ensemble de triangles, il est nécessaire de calculer son centre et son rayon. Ainsi, la première sphère englobante est calculée en utilisant l'algorithme présenté par Gärtner qui permet d'obtenir le plus petit cercle englobant un nuage de points (Cf. Figure IV.85) [Gartner 99].



Figure IV.85 : Sphère englobante calculée à partir du centre géométrique

L'algorithme développé pour construire la hiérarchie de sphères est l'amélioration de la méthode de Quinlan [Quinlan 94] par l'intégration de l'algorithme décrit par Gottschalk et al. [Gottschalk 96]. Ainsi, l'arbre de sphères est construit récursivement par une méthode descendante jusqu'à avoir englobé tous les triangles de nos coussinets des doigts. Nous obtenons alors un arbre binaire de sphères : c'est à dire que chaque volume englobant de la hiérarchie possède zéro ou deux fils. Cette méthode hybride permet d'améliorer les approximations des polyèdres par des sphères englobantes.

Afin de résoudre les difficultés liées à la division de la sphère en deux parties égales et à la répartition des triangles appartenant aux demi-sphères, nous utilisons une Analyse en Composantes Principales, appelée « ACP » (Cf. Annexe A.3.). Cette méthode permet de définir les directions principales d'un nuage de points en utilisant des statistiques et les valeurs propres d'un ellipsoïde englobant le nuage de points.

L'algorithme proposé construit un arbre binaire de manière récursive à partir d'un nœud dont le volume englobant a déjà été calculé. Cet algorithme est décrit ci-après :

Variables internes :

- A est la matrice des vecteurs propres avec $A = [V_i]$ pour $i \in \{1,2,3\}$,
- T est l'ensemble des triangles du nœud courant, t est un triangle $\in T$,
- $T_{\text{fils_gauche}}$ est l'ensemble des triangles du fils gauche et $T_{\text{fils_droit}}$ est l'ensemble des triangles du fils droit,

Algorithmes utilisés :

- ACP retournant la matrice des vecteurs propres ordonnés de manière décroissante suivant leur valeur propre.

```

Début
  Si card(T) = 2, avec  $T = \{t_1, t_2\}$ ,
     $t_1 \in T_{\text{fils\_gauche}}$ 
     $t_2 \in T_{\text{fils\_droit}}$ 
    On calcule le volume englobant du fils gauche
    On calcule le volume englobant du fils droit
  Sinon
    Si card(T) > 2,
       $A = \text{ACP}$ 
       $i = 0$ 
      Tant que card( $T_{\text{fils\_gauche}}$ ) = 0 ou que card( $T_{\text{fils\_droit}}$ ) = 0,
        Pour chaque  $t \in T$ 
           $X = \text{barycentre}(t)$ 
          Si  $\text{barycentre}(t) \cdot V_i > 0$ ,
             $t \in T_{\text{fils\_gauche}}$ 
          Sinon
             $t \in T_{\text{fils\_droit}}$ 
          Fin si
        Fin pour
         $i = i + 1$ 
      Fin tant que
      On calcule le volume englobant du fils gauche
      On calcule le volume englobant du fils droit
      On relance l'algorithme avec le fils gauche
      On relance l'algorithme avec le fils droit
    Fin si
  Fin si
Fin

```

Notons que cet algorithme fonctionne quel que soit le type de volume englobant utilisé.

Nous obtenons les résultats suivants dans le cas d'une sphère maillée (Cf. Figure IV.86) :

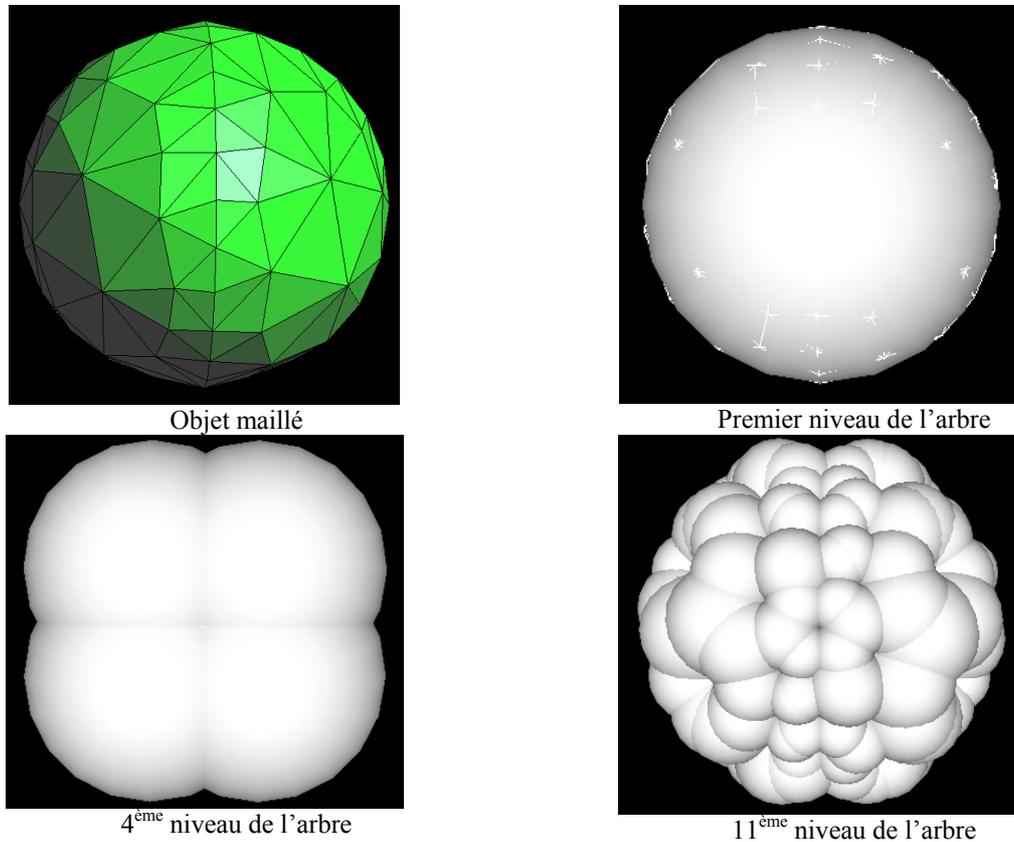


Figure IV.86 : Hiérarchie de volumes englobants pour une sphère

La sphère se compose de 180 nœuds, 200 triangles, 1137 tétraèdres

Les temps de construction des arbres binaires pour un rectangle et une sphère sont donnés dans le tableau suivant (Cf. Tableau I.1) :

	Nombre d'éléments	Nombre de nœuds	Temps (ms)
Rectangle	240	44	34,835
	994	154	120,305333
	3006	453	261,388667
	6096	940	403,049667
Sphere	406	70	33,401
	779	108	50,1976667
	1954	274	95,2093333
	6131	861	243,364333

Tableau IV.5 : Les temps de construction des arbres en fonction des maillages

Nous remarquons que plus les objets maillés sont complexes, plus la construction de la hiérarchie de sphères englobantes est longue (4 s au maximum pour un objet constitué de

6000 éléments). Mais, ceci n'a pas d'impact sur les performances de l'algorithme de détection de collision puisque les arbres binaires sont pré-calculés.

Les courbes de la Figure IV.87 montrent que le temps d'exécution varie presque linéairement en fonction du nombre de nœuds des objets.

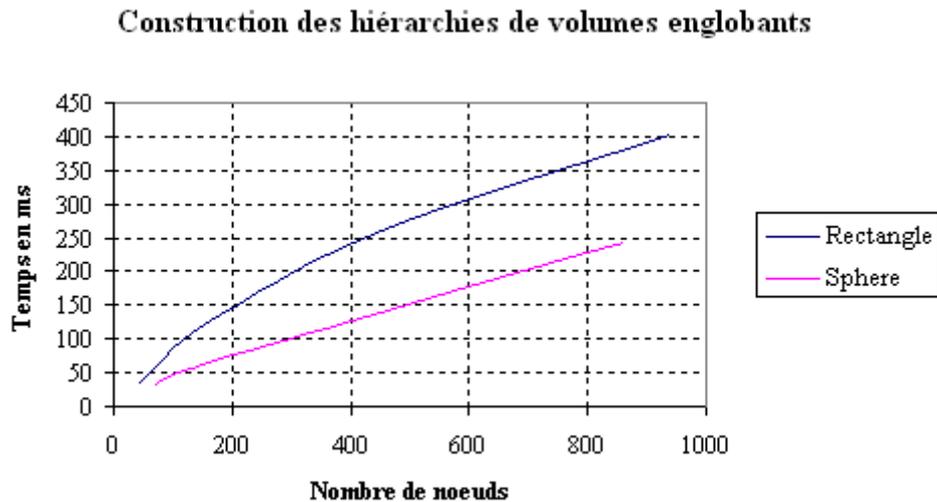


Figure IV.87 : Temps de construction des hiérarchies de sphères (pré-calcul)

Après la construction des arbres de sphères englobantes pour chaque objet de la scène, nous devons déterminer s'ils sont en contact ou pas.

Pour ce faire, nous calculons tout d'abord la distance séparant les objets. Lorsque cette distance est nulle, les objets sont en collision. A ce stade, nous devons déterminer exactement le point de contact, c'est-à-dire parcourir les hiérarchies de volumes englobants. Dans les paragraphes suivants, nous présentons la méthode utilisée pour le calcul de distance ainsi que les résultats obtenus. Puis, nous détaillons le test d'interférences mis en œuvre pour définir les couples de triangles des objets en contact. Nous donnerons également les résultats obtenus.

2.4.3 Le calcul de distances entre les objets de la scène

Dans le cas d'arbres de sphères, le calcul de la distance entre les objets consiste à réaliser le test suivant :

Variables internes :

- S_1 la première sphère englobante définie par son centre C_1 de coordonnées x_1 , y_1 et z_1 , et par son rayon R_1
- S_2 la deuxième sphère englobante définie par son centre C_2 de coordonnées x_2 , y_2 et z_2 , et par son rayon R_2

$$|C_1C_2| > R_1 + R_2$$

Ou sous une autre forme, nous obtenons :

$$(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 - (R_1 + R_2)^2 > 0 \Leftrightarrow D > 0$$

La Figure IV.88 montre l'exemple du calcul de la distance entre deux sphères maillées identiques.

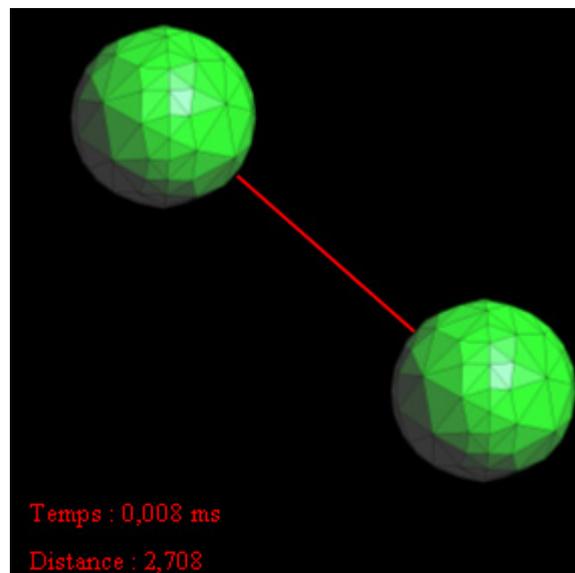


Figure IV.88 : Calcul de distance entre deux sphères

Chaque sphère est constituée de 180 nœuds, 200 triangles, 1137 tétraèdres

Les tests effectués avec d'autres formes d'objets nous ont donné les mêmes résultats en temps de calcul. Ce temps est constant et égal à 0,008 ms car lors de la phase de construction des hiérarchies de sphères effectuée en pré-calcul, le centre et le rayon de chaque sphère englobante sont stockés dans un tableau. Le calcul de distance est donc immédiat.

Nous décrivons dans le paragraphe suivant l'algorithme développé pour détecter les interférences entre les objets de la scène.

2.4.4 La détection d'interférences

Dans le cas de hiérarchies de volumes englobants, il existe principalement deux cas de figure :

- si la collision est détectée entre les premiers niveaux des hiérarchies, nous sommes dans le cas de contact entre polygones convexes. Nous utilisons les méthodes de détection exacte décrites dans le paragraphe IV.2.1.
- si la collision est détectée au niveau des feuilles, nous comparons les triangles associés à chaque sphère englobante avec les algorithmes de Quinlan [Quinlan 94] et de Möller [Moller 97].

La Figure IV.89 présente le test d'interférences entre les arbres binaires d'un objet *a* et d'un objet *b*. Dans cet exemple, le contact s'effectue au troisième niveau de l'arbre, entre *b* et *d*. Pour détecter les collisions entre les deux objets, nous calculons la distance entre les deux sphères racines de l'objet *a* et de l'objet *b*. S'il y a collision, nous descendons d'un niveau dans l'un des arbres et nous répétons l'opération jusqu'à obtenir les triangles en collision (Cf. Figure IV.89).

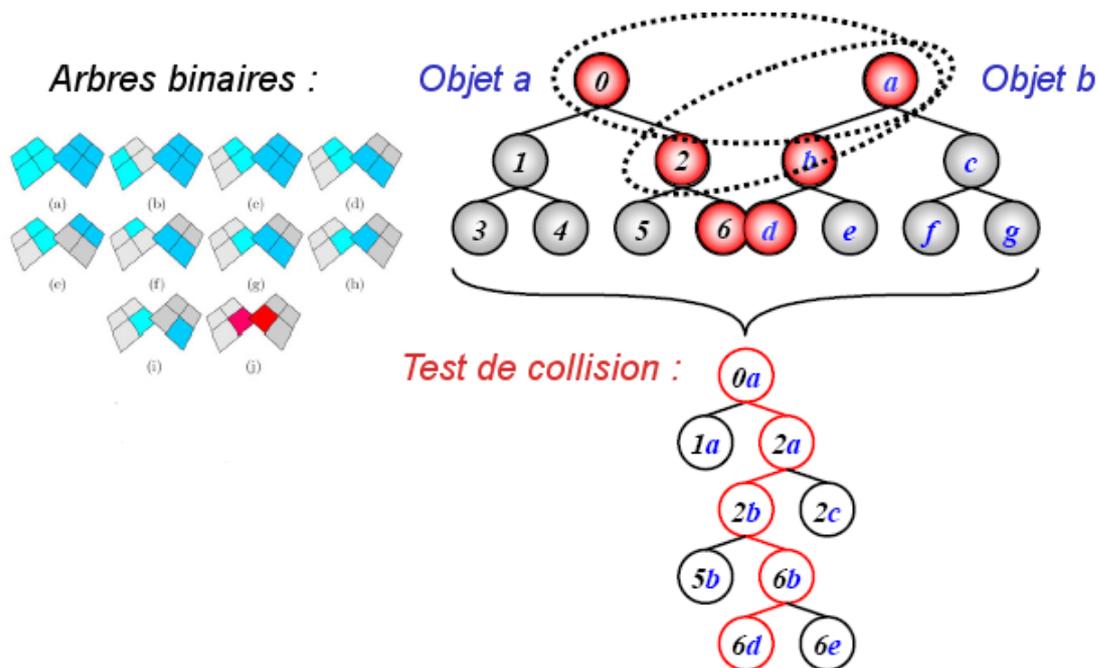


Figure IV.89 : Exemple de descente d'arbres lors d'une collision

L'algorithme utilisé pour la détection d'interférences entre les arbres binaires de deux objets A et B est le suivant :

Variables internes :

- Arbo est un arbre de sphères,
- N est le niveau courant de parcours de l'arbre,
- S est une sphère englobante de l'arbre,
- $S(N, \text{Droit})$ et $S(N, \text{Gauche})$ sont les sphères-filles d'une sphère $S(N)$ de l'arbre Arbo,
- F est une feuille d'une sphère englobante S : $F \in S$,
- t est un triangle : $t \in F$,
- H est la hauteur d'une hiérarchie de sphères, i.e. le nombre de niveaux,
- D est la distance séparant deux sphères englobantes.

Algorithmes utilisés :

- `sphereSphere()` retournant la distance entre éléments,
- `triangleTriangles()` retournant le résultat du test de triangles pour la collision (paire des triangles en contact)

```

Début
Mesure de  $H_A$ 
Mesure de  $H_B$ 

Pour  $N_A = 0, 0 \leq N_A \leq H_A$ 
  Pour  $N_B = 0, 0 \leq N_B \leq H_B$ 
     $D = \text{sphereSphere}(S_A(N_A), S_B(N_B))$ 

    Si  $D > 0$  : PAS DE COLLISION
       $\rightarrow$  on sort et on teste d'autres objets
    Sinon  $D \leq 0$  : COLLISION

      Si  $H_A < H_B$ 
         $\rightarrow$  on permute l'objet A avec l'objet B
        (le but est, dans la comparaison entre 2 objets, de tester
        l'arborescence la plus grande contre la plus petite).
      Fin si

      (1) Si  $N_B = H_B$ 
         $\rightarrow S_B = F_B$  et  $t_B \in F_B$  (i.e. on a atteint le niveau feuille de
        l'arborescence de B)
         $\rightarrow \text{triangleTriangles}(t_B, t_A \in S_A(N_A))$ 
      Sinon  $N_B = N_B + 1$ 
         $D = \text{sphereSphere}(S_A(N_A), S_B(N_B, \text{Gauche}))$ 

        (2) Si  $D \leq 0$  :  $N_A = N_A + 1$ 
           $D = \text{sphereSphere}(S_A(N_A, \text{Gauche}), S_B(N_B, \text{Gauche}))$ 

          Si  $D \leq 0$  : COLLISION
             $\rightarrow$  reprendre à (1)

          Sinon :
             $D = \text{sphereSphere}(S_A(N_A, \text{Droit}), S_B(N_B, \text{Gauche}))$ 
            Si  $D \leq 0$  : COLLISION
               $\rightarrow$  reprendre à (1)
            Fin si
          Fin si

          Sinon  $D = \text{sphereSphere}(S_A(N_A), S_B(N_B, \text{Droit}))$ 
            Si  $D \leq 0$  : COLLISION
               $\rightarrow$  reprendre à (2)
            Fin si
          Fin si
        Fin si
      Fin si
    Fin Pour
  Fin Pour
Fin

```

Nous proposons d'évaluer du point de vue du temps de calcul la pertinence du critère de descente dans l'arbre et la méthode de détection de collision précédemment décrits.

Nous effectuons des tests de collision entre deux objets ayant le même maillage. En l'occurrence, nous réalisons quatre tests différents avec des sphères possédant un nombre de nœuds croissant :

- deux sphères de 70 nœuds et 406 éléments,
- deux sphères de 108 nœuds et 779 éléments,
- deux sphères de 274 nœuds et 1954 éléments,
- deux sphères de 861 nœuds et 6131 éléments.

Nous représentons les sphères par des hiérarchies de volumes englobants pour la détection de collision. Le choix des sphères a été fait en rapport avec la forme des coussinets déformables des doigts de la main virtuelle.

La Figure IV.90 ci-dessous présente le résultat d'un test d'interférence. Dans ce cas de figure, nous avons détecté douze triangles en contact, soit 6 paires.

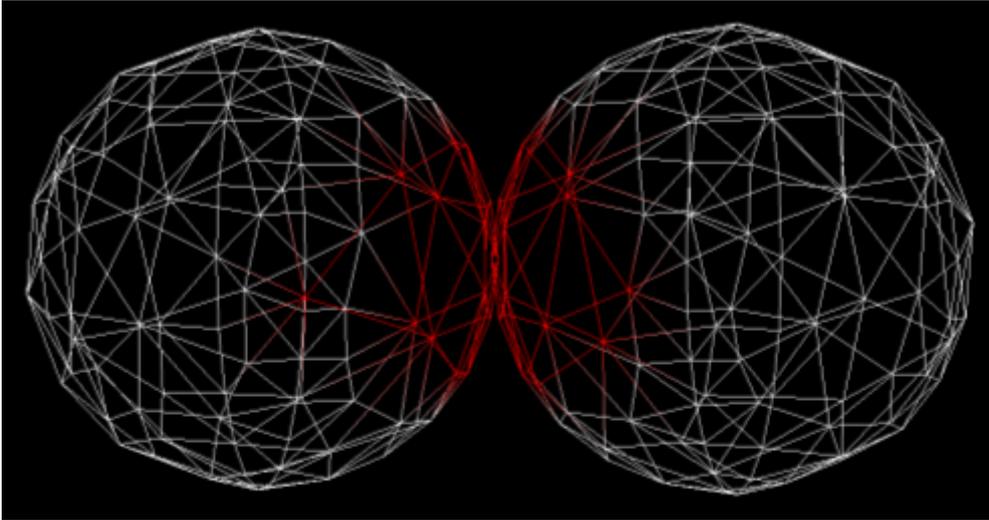


Figure IV.90 : Test de triangles entre deux sphères maillées identiques

Les sphères sont constituées de 108 nœuds et 779 éléments

La Figure IV.91 ci-dessous nous montrent que le temps de calcul augmente de façon linéaire avec le nombre de nœuds et d'éléments. Ceci est dû au nombre plus important de primitives à tester lors du parcours des hiérarchies des sphères englobantes.

Temps de calcul pour le test des triangles

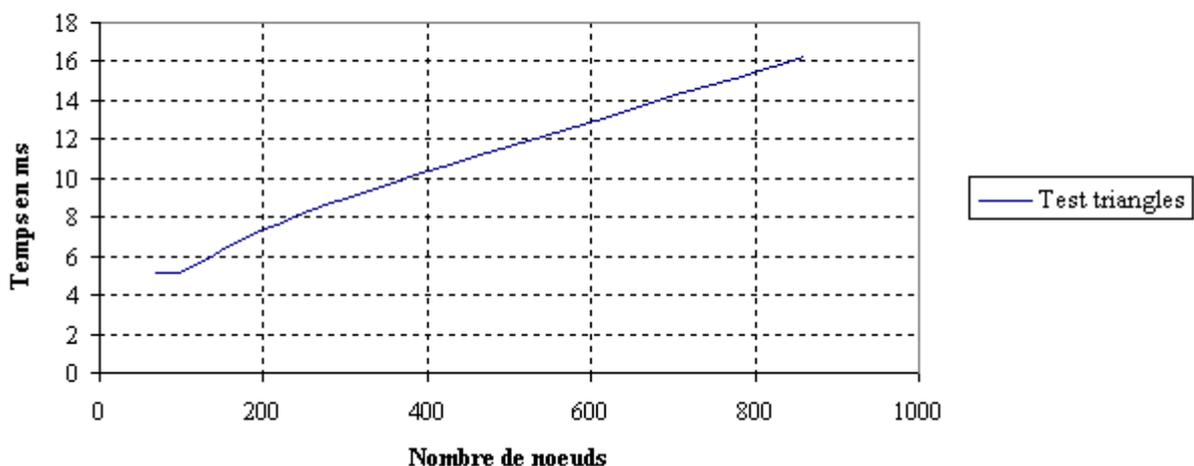


Figure IV.91 : Temps de calcul de la détection de collision en fonction des noeuds

Nous remarquons sur la courbe ci-dessus que le temps d'exécution du test des triangles a pour valeur minimale 5ms et cela même pour des objets comportant peu d'éléments. Ceci est dû au

manque de compacité des sphères englobantes qui entraîne un nombre important de tests d'interférence.

A la fin de cette étape, nous obtenons une liste de paires de primitives en contact. Celle-ci est utilisée pour la gestion de la collision qui est décrite dans le paragraphe 3.

2.4.5 Conclusion sur la méthode programmée

Pour détecter les collisions entre les coussinets déformables de la main virtuelle et les objets rigides de la scène, nous avons utilisé des hiérarchies de sphères englobantes. Ces dernières ont pour objectif de minimiser le nombre de tests d'interférence entre les coussinets et les objets de la scène en éliminant les éléments qui ne sont pas en collision.

Ensuite, elles permettent d'identifier les niveaux des arbres de sphères qui sont en contact et elles retournent la liste des couples de sphères englobantes susceptibles d'être en collision. Chaque sphère enveloppe un (ou plusieurs) triangle(s) du maillage suivant leur niveau dans la hiérarchie. Les tests d'intersection sont donc réalisés entre triangles. Notre méthode permet ainsi d'identifier les éléments caractéristiques en contact (sommet, face ou côté) et de déterminer la distance de pénétration et la normale à chaque point de contact.

Les résultats obtenus montrent que les tests d'interférence entre objets complexes deviennent rapidement coûteux en temps de calcul. Ceci est dû en grande partie à l'utilisation de sphères englobantes qui nécessitent de nombreux tests pendant la descente de l'arbre binaire en raison de leur manque de compacité. Pour optimiser le temps de calcul pour cette méthode de détection de collision et donc répondre aux spécifications temps réel de nos applications, il est nécessaire d'utiliser d'autres volumes englobants tels que les AABB.

Après avoir identifié la zone de la collision entre les mains et les objets, il faut calculer les réactions aux collisions. Cela revient à déterminer les forces à appliquer aux coussinets déformables pour empêcher l'interpénétration, ainsi qu'à calculer les accélérations contraintes ou les vitesses post impact. Cette phase est appelée « gestion de la collision » ou « traitement du contact ». Dans le paragraphe suivant, nous détaillons les méthodes existantes ainsi que celle développée pour notre application.

3 Le traitement du contact

Dans notre simulation, les objets rigides et les mains déformables sont en mouvement dans l'environnement virtuel. Ils peuvent donc subir deux types d'actions : les actions dues aux forces extérieures telle que la gravité et les actions de contact résultant de la collision entre les objets rigides et les mains déformables virtuelles. Dans ce paragraphe, nous nous intéressons plus particulièrement à la réponse à la collision. Ainsi, nous cherchons à déterminer les forces de contact afin de calculer la cinématique et la dynamique des objets virtuels et celle des mains. Ces forces permettent également de mettre à jour les déformations des mains virtuelles lors de la simulation des tâches manuelles.

Le traitement du contact se révèle être complexe dans notre contexte. En effet, la simulation de tâches de préhension avec une main virtuelle déformable est un cas multi-contacts car plusieurs zones de la main (i.e. plusieurs doigts et/ou la paume) se trouvent en collision avec l'objet saisi. Par conséquent, il est difficile d'appliquer les algorithmes de réponse traditionnels par couples d'objets en contact : une approche globale est nécessaire. De plus, afin d'obtenir une simulation réaliste, il est important de tenir compte des caractéristiques physiques de la main comme le frottement. Or, le calcul du frottement au niveau des contacts

ne peut être traité a posteriori car il conditionne la réponse à la collision. L'objectif est donc de définir une méthode qui gère plusieurs contacts simultanés, avec frottement, entre des objets rigides et des mains déformables tout en respectant la contrainte de temps-réel.

Après avoir donné des définitions sur les différentes contraintes possibles entre objets, nous décrirons les différentes méthodes existantes qui permettent de traiter les contacts entre objets rigides et déformables [Lin 96], [Meseure 03], [Teschner 05]. Puis, nous présenterons la méthode que nous utilisons pour résoudre les multiples collisions simultanées entre notre main virtuelle déformable et les objets rigides dans notre environnement virtuel dynamique.

3.1 Définitions

Il existe différentes interactions entre des objets dans un environnement réel ou virtuel : les contacts (appelés également « chocs » ou « liaisons unilatérales ») ou les liaisons entre des objets appelées « liaisons bilatérales ». Ces différentes interactions seront appelées « contraintes » dans la suite du manuscrit. On note que ces contraintes limitent les mouvements des objets dans l'espace et diminuent par conséquent l'espace des configurations possibles.

Les définitions qui suivent sont données pour un corps rigide mais elles restent vraies pour un corps déformable.

Soit q_t la configuration d'un corps solide à l'instant t . Cette configuration est déterminée au maximum par six degrés de liberté : trois translations et trois rotations.

Soit $f(q_t)$ la fonction contrainte qui est une fonction de l'espace des configurations possibles des corps rigides à \mathbb{R}^d , avec d le nombre de degrés de liberté que la contrainte supprime du système.

- Les contraintes holonomes traduisent le fait que la contrainte sur la vitesse peut être obtenue en dérivant la contrainte de position. Si les contraintes ne respectent pas cette définition, elles sont non-holonomes.
- Une contrainte est dite « unilatérale » lorsque le mouvement d'un corps rigide est empêché dans une seule direction comme par exemple lors d'une collision (Cf. Figure IV.92) [Tubiana 80]. Elle se traduit par une inégalité à respecter de la forme $f(q_t) \leq 0$ ou $f(q_t) \geq 0$.

Par exemple, dans le cas d'un contact entre deux objets, on a $\delta_n \geq 0$, avec δ_n la distance entre les deux objets portée par la normale au contact. Cette relation signifie que :

- soit les deux objets sont au contact : $\delta_n = 0$,
- soit les deux objets ne sont pas au contact : $\delta_n > 0$.

Il faut aussi distinguer les contraintes unilatérales avec contact permanent ou contact non permanent, ainsi que les contacts sans frottement et avec frottement. Différents types de contacts sont présentés dans l'Annexe A.2.

- Une contrainte est dite « bilatérale » si deux solides liés peuvent se déplacer l'un par rapport à l'autre sans rompre le contact comme par exemple dans le cas d'une liaison (Cf. Figure IV.93) [Tubiana 80]. Elle s'exprime par une égalité appelée « équation de liaison » qui est de la forme $f(q_1, q_2, \dots, q_n) = 0$. Par conséquent, une liaison entre

deux solides consiste à supprimer un ou plusieurs degrés de liberté, ou à rendre dépendant un certain nombre de degrés de liberté.

Dans notre cas de figure, la main est modélisée par un système de corps rigides qui sont reliés entre eux par des liaisons pivot. De plus, il existe des actions de contact entre la main et les objets de l'environnement virtuel qu'elle manipule. Il est donc nécessaire de gérer en temps réel des contraintes bilatérales et unilatérales avec frottement.



Figure IV.92 : Exemple de liaison unilatérale



Figure IV.93 : Exemple de liaison bilatérale

Dans le paragraphe suivant, nous présentons les principaux algorithmes existants qui gèrent des systèmes contraints en environnement dynamique.

3.2 Les méthodes de gestion de contraintes existantes

Simuler des environnements de manière réaliste exige la modélisation physique des interactions entre les objets de la scène : c'est-à-dire les actions de liaison et les forces de contact. De nombreuses recherches sur le traitement du contact ont été réalisées et sont réalisées dans les domaines de l'animation graphique, de la robotique et de la mécanique. Nous présentons les principales méthodes ci-après [Meseure 03].

3.2.1 La méthode des pénalités

La méthode à pénalité revient en pratique à modéliser chaque contrainte par un ressort. Lorsque la contrainte n'est pas respectée, le ressort exerce une force tendant à diminuer l'erreur et à ramener le système à l'état cohérent [Moore 88]. Il est nécessaire d'utiliser des pas de temps aussi petits que possible et une grande constante de rigidité pour le ressort. Par exemple, dans le cas de ressort visqueux, on obtient l'équation suivante :

$$f = k(l - l_0) + b(\dot{x}_1 - \dot{x}_2) \quad (\text{Eq. IV-96})$$

avec :

- k : le coefficient de raideur,
- b : le coefficient d'amortissement ou de viscosité,
- l_0 : la longueur à vide du ressort
- l : la longueur courante du ressort
- \dot{x}_1 et \dot{x}_2 : la vitesse des extrémités du ressort

Cette méthode a été utilisée pour la simulation d'objets rigides par Mirtich [Mirtich 00] et d'objets déformables par Terzopoulos [Terzopoulos 87] et [Terzopoulos 88]. Mais, elle présente deux inconvénients principaux. Tout d'abord, il est difficile de régler la constante de raideur du ressort. Quand on l'augmente, on ajoute une autre fréquence au système d'où d'éventuels problèmes de convergence des équations. Si on la diminue, les chocs sont lents, peu énergétiques, et la contrainte n'est pas respectée pendant un temps long. Le second

inconvéniéent apparaît sous la forme d'instabilité et d'oscillations lorsque la contrainte est quasiment respectée.

3.2.2 La méthode à base de contraintes

Un simulateur, appelé **Coriolis**, a été proposé par Baraff [Baraff 94]. Il permet de calculer directement les forces de contact, sans ou avec frottement, ainsi que les impulsions à appliquer au niveau des points de contact pour empêcher les corps rigides de s'interpénétrer. Les méthodes analytiques mises en oeuvre permettent de résoudre plusieurs contraintes unilatérales simultanées.

Nous expliquons ce modèle de contact.

Soit un contact i entre deux solides rigides A et B. Pour chaque point de contact c_i , on note a_i le scalaire représentant l'accélération relative entre les corps. Elle est portée par la normale à la surface de contact en c_i . On a donc les différentes configurations suivantes :

- soit $a_i < 0$: les deux corps sont en collision,
- soit $a_i = 0$: les deux corps restent en contact,
- soit $a_i > 0$: les deux corps rompent le contact.

Pour éviter l'interpénétration, on doit donc avoir $a_i \geq 0$.

La force agissant entre les corps A et B est notée \mathbf{F} et son intensité est le scalaire F_i . Elle est normale à la surface au point de contact c_i dans le cas sans frottement.

Le système à résoudre est :

$$\begin{cases} \mathbf{a} = \mathbf{A}\mathbf{F} + \mathbf{b} \\ a_i \geq 0 \\ F_i \geq 0 \\ F_i \cdot a_i = 0 \end{cases} \quad (\text{Eq. IV-97})$$

avec :

- $\mathbf{A} \in \mathbb{R}^{n \times n}$: une matrice symétrique et semi-définie positive (PSD). Elle représente les masses des corps et les géométries de contact,
- $\mathbf{b} \in \mathbb{R}^n$: les forces d'inertie et les forces extérieures du système.

Résoudre ce système revient à résoudre un problème de complémentarité linéaire, dit LCP pour *Linear Complementarity Problem*. Dans le cas sans frottement, la méthode proposée par Baraff est semblable à celle du pivot de Gauss [Baraff 94].

Dans les cas avec frottement statique ou dynamique, le problème peut également s'écrire sous la forme d'un LCP. Mais dans ces cas de figure, il est possible qu'aucune solution n'existe ou au contraire, que plusieurs solutions soient obtenues. De nombreux articles traitent de ces problèmes et des méthodes ont été proposées par Baraff [Baraff 94], Pang et Trinkle [Pang 96], Sauer et Schömer [Sauer 98] ou encore Mukundakrishnan [Mukundakrishnan 04].

L'avantage de cette méthode est que tous les contacts sont traités en même temps, dans un seul et même système. Le calcul est donc beaucoup plus rapide. L'inconvénient est que la détection des contacts doit se faire par une autre méthode car cette méthode ne traite que les contacts.

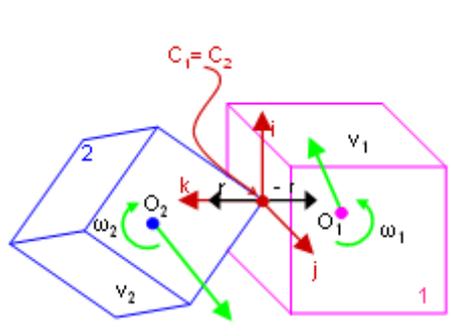
3.2.3 La méthode des impulsions

Un autre simulateur, appelé **Impulse**, a été proposé par Mirtich et Canny [Mirtich 94]. Les interactions entre les objets sont modélisées par des micro-collisions et des impulsions sont

appliquées pour symboliser les contacts au repos. Cette simplification donne un comportement réaliste aux objets et elle ne pénalise pas le rendu physique des simulations. Ce simulateur permet également de gérer les contacts avec frottement (roulement, glissement ou adhérence).

Une impulsion peut être vue comme une très grande force agissant pendant un temps infinitésimal. Il existe deux méthodes pour calculer une impulsion :

- la méthode analytique : soient deux solides 1 et 2 en contact. Les vitesses linéaires et angulaires aux centres de gravité O_1 et O_2 sont respectivement \mathbf{v}_1 et \mathbf{v}_2 , $\boldsymbol{\omega}_1$ et $\boldsymbol{\omega}_2$. Au point de contact, on a c_1 et c_2 qui sont confondus. L'impulsion \mathbf{r} est obtenue en résolvant le système suivant (Cf. Figure IV.94) [Meseure 03] :



$$\left\{ \begin{array}{l} m_1 \mathbf{v}_{1_{\text{corr}}} = m_1 \mathbf{v}_1 + \mathbf{r} \\ m_2 \mathbf{v}_{2_{\text{corr}}} = m_2 \mathbf{v}_2 + \mathbf{r} \\ I_1 \boldsymbol{\omega}_{1_{\text{corr}}} = I_1 \boldsymbol{\omega}_1 + O_1 C_1 \times \mathbf{r} \\ I_2 \boldsymbol{\omega}_{2_{\text{corr}}} = I_2 \boldsymbol{\omega}_2 - O_2 C_2 \times \mathbf{r} \\ \mathbf{r} \cdot \mathbf{i} = 0 \\ \mathbf{r} \cdot \mathbf{j} = 0 \\ (\mathbf{v}_{2_{\text{corr}}} + \boldsymbol{\omega}_{2_{\text{corr}}} \times O_2 C_2 - (\mathbf{v}_{1_{\text{corr}}} + \boldsymbol{\omega}_{1_{\text{corr}}} \times O_1 C_1)) \cdot \mathbf{k} = 0 \end{array} \right.$$

Figure IV.94 : Méthode analytique pour les impulsions

- l'intégration numérique : la collision est décomposée en une phase de compression et une phase de restitution. La limite entre ces deux phases s'appelle le point de compression maximum. Pour déterminer ce point, on exprime la dérivée de la vitesse de pénétration $\dot{\mathbf{u}}$ en fonction du coefficient de frottement μ et de la vitesse de pénétration \mathbf{u} . On obtient alors l'équation différentielle suivante :

$$\begin{bmatrix} \dot{u}_x \\ \dot{u}_y \\ \dot{u}_z \end{bmatrix} = M \begin{bmatrix} -\mu \frac{u_x}{\sqrt{u_x^2 + u_y^2}} \\ -\mu \frac{u_y}{\sqrt{u_x^2 + u_y^2}} \\ 1 \end{bmatrix} \quad (\text{Eq. IV-98})$$

Il est possible de suivre l'évolution de \mathbf{u} pendant la collision grâce à une intégration numérique. Lorsque \mathbf{u} est nulle, le problème est résolu.

Les avantages de cette méthode sont la simplicité de mise en œuvre et la robustesse. Contrairement à la méthode décrite précédemment, il existe toujours une solution dans les contacts sans et avec frottement. Les inconvénients sont le traitement unitaire des collisions et le cas des solides en contact sans interpénétration.

3.2.4 La méthode pour des chaînes articulées issue de la Robotique

Afin de généraliser les simulateurs précédents qui ne considèrent que les contacts entre des corps rigides, Ruspini et Khatib [Ruspini 97] ont proposé un simulateur, appelé **Simpact**, qui considère des systèmes articulés. Les contraintes sont résolues dans l'espace des contacts en

s'inspirant des méthodes utilisées en Robotique. Ainsi, on ne calcule que les impulsions externes et les forces de contact sur le système articulé. L'utilisation des coordonnées globales permet d'ignorer les forces internes au système pendant la simulation.

3.2.5 La méthode utilisée en Mécanique pour des milieux granulaires

Il existe une méthode de simulation numérique qui résout les multiples contacts simultanés avec frottement. Cette méthode, appelée « dynamique des contacts », est utilisée pour répondre à la problématique des milieux granulaires (i.e. des ensembles de particules de tailles et de formes diverses). Ainsi, lorsqu'une particule vient heurter un empilement de particules, elle crée une impulsion incidente qui se propage à travers le réseau de contacts sous forme d'onde. Les effets ne sont pas localisés et ils doivent être traités de manière collective. Il est donc nécessaire de formuler des lois de contacts de manière plus générale afin de traiter de la même façon les contacts maintenus et les collisions ainsi que de tenir compte de l'ensemble des particules en contact (i.e. tout le réseau). Cette méthode a été mise au point par Moreau et Jean [Moreau 88] [Jean 99] [Jean 01] [Moreau 03]. Elle permet de traiter les collisions multiples en mettant en œuvre des lois de contact généralisées.

La prise d'objets engendre également des contacts frottants multiples entre la main et les objets saisis. Bien que les milieux granulaires étudiés soient faits de particules rigides, cette méthode est apparue pertinente pour la résolution de contacts entre objets rigides et déformables. Une adaptation de la dynamique des contacts en milieu granulaire a d'ailleurs été réalisée pour les environnements virtuels par Duriez [Duriez 04].

3.2.6 Conclusion

Les méthodes par impulsions cherchent à respecter précisément le contact entre les objets. Elles permettent de calculer les forces de contact exactes et les impulsions nécessaires pour garantir la non-pénétration des objets. L'inconvénient est que ces méthodes sont peu robustes dans le cas de mouvements complexes ou de collisions multiples dans des environnements complexes (sur contraintes).

Les méthodes par contraintes sont efficaces dans le cas de solides rigides ou de systèmes articulés. Mais, elles ne sont pas adaptées à la collision entre des corps rigides et déformables. Comme il est difficile de prévoir les déformations des corps déformables lors de l'impact, il est difficile, voire impossible, de fixer les contraintes en position des points de contact.

Les méthodes de pénalités sont simples à mettre en œuvre et robustes. Elles utilisent des forces de restauration proportionnelles à la distance de pénétration pour le mouvement des objets. Elles permettent de définir plusieurs contraintes et de toujours obtenir une solution. L'inconvénient de ces méthodes est d'ajouter des hautes fréquences dans le système et donc d'exiger des pas de temps très petits lors de la collision pour obtenir des résultats cohérents. Mais dans le cas d'objets déformables, l'énergie créée au cours de la pénétration des corps est utile pour déformer ces derniers : la forme des objets après contact dépend donc de l'absorption de l'énergie du choc par chacun.

La dynamique des contacts est complexe à mettre en œuvre mais elle permet de gérer plusieurs contacts simultanés avec frottement. C'est cette méthode que nous avons retenue.

3.3 Le modèle de contact proposé

Dans notre cas de figure, nous avons choisi de mettre en œuvre une méthode globale pour traiter les collisions entre les doigts déformables et les objets rigides de la simulation. Plus exactement, nous avons utilisé et adapté à notre cas de figure la méthode présentée par Duriez [Duriez 2004].

Nous commencerons par expliquer les notations. Puis, nous expliquerons la construction de l'espace des contacts dans lequel notre système d'équations peut se résoudre sous la forme d'un problème de complémentarité linéaire. Ensuite, nous introduirons le comportement de la chaîne arborescente – i.e. le squelette – ainsi que les déformations des coussinets. Après la résolution des contacts normaux, nous nous intéresserons aux contacts avec frottement. Et nous terminerons par l'algorithme utilisé pour résoudre les contacts multiples et pour calculer les efforts exercés sur les objets.

3.3.1 Définition de la configuration initiale

La configuration d'un objet, notée \mathbf{q} , représente sa translation et sa rotation à un instant t . Sa dérivée première, notée $\dot{\mathbf{q}}$, désigne la vitesse et sa dérivée seconde, notée $\ddot{\mathbf{q}}$, l'accélération. Les équations de la dynamique donnent :

$$\mathbf{M} \ddot{\mathbf{q}} = \mathbf{F}_C + \mathbf{F}_{\text{ext}} \quad (\text{Eq. IV-99})$$

avec :

- \mathbf{M} : la matrice de masse du système comprenant les termes d'inertie,
- \mathbf{F}_C : les forces de contact,
- \mathbf{F}_{ext} : les forces extérieures.

On décrit de la même manière les mouvements de chaque objet de l'environnement virtuel : on obtient un vecteur de configuration global $\mathbf{q} = (q_1^T \ q_2^T \ \dots \ q_n^T)$ avec $1 \leq i \leq n$ le nombre d'objets présents dans la simulation. Les équations peuvent être résolues sous la forme d'un système d'équations.

On appelle « mouvement libre » le mouvement des objets qui ne sont soumis qu'aux forces extérieures, comme par exemple la gravité ou les forces de Coriolis. Dans ce cas, les objets ne sont pas en contact et la contribution des efforts de contact est nulle. La configuration calculée correspond à la position dans laquelle est réalisée la détection de collision (Cf. §2.4) : on la note $\mathbf{q}^{\text{libre}}$.

Quand une collision est détectée entre les objets et les mains de l'environnement virtuel, certaines configurations deviennent dépendantes. Dans ce cas, il est nécessaire d'appliquer des forces de contraintes pour empêcher les interpénétrations. Pour cela, il faut construire l'espace des contacts afin de calculer le mouvement résultant du contact que l'on appelle « mouvement contraint ». On obtient alors une relation entre la configuration libre, les forces de contact \mathbf{F}_C et la configuration contrainte notée $\mathbf{q}^{\text{contraint}}$:

$$\mathbf{q}^{\text{contraint}} = \mathbf{C} \cdot \mathbf{F}_C + \mathbf{q}^{\text{libre}} \quad (\text{Eq. IV-100})$$

avec \mathbf{C} une matrice homogène à une compliance.

Nous allons définir les relations entre les forces de contact et les déplacements dans l'espace de contact. Pour cela, nous commençons par expliquer la construction de l'espace de contact.

3.3.2 Définition de l'espace de contact

Considérons deux objets A et B suffisamment proches pour que l'on détecte une collision en deux points C_A et C_B , appartenant respectivement à A et B. Notons I, le point commun à ces deux objets, on a donc au contact : $I=C_A=C_B$.

On définit alors une normale au contact, notée \mathbf{n} , perpendiculaire à la surface de contact, ainsi que deux vecteurs tangents aux deux objets. On obtient ainsi un repère orthonormé local appelé $\mathcal{R}_C = (I, \mathbf{t}_1, \mathbf{t}_2, \mathbf{n})$. La Figure IV.95 présente les paramètres pour le modèle de contact [Meseure 03] :

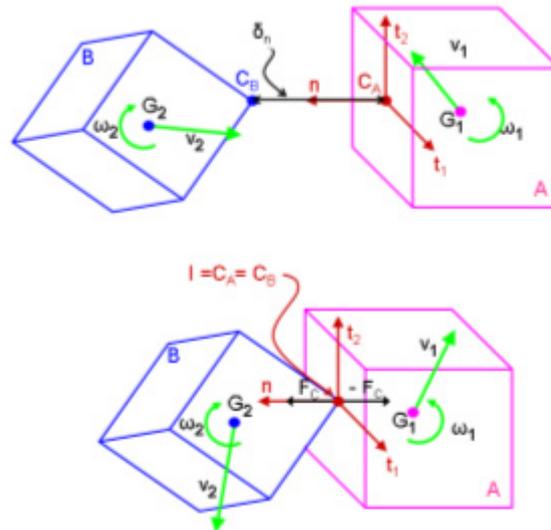


Figure IV.95 : Contact entre deux objets A et B

Dans ce repère local, les forces de contact, notées \mathbf{F}_C , et les vitesses peuvent être décomposées en une partie normale portée par \mathbf{n} ainsi qu'en des parties tangentielles portées par \mathbf{t}_1 et \mathbf{t}_2 :

$$\mathbf{F}_C = F_{C_n} \mathbf{n} + F_{C_{t_1}} \mathbf{t}_1 + F_{C_{t_2}} \mathbf{t}_2 \quad (\text{Eq. IV-101})$$

Pour quantifier le contact, on définit la distance entre objets notée δ_n portée par la normale au contact \mathbf{n} . On a donc en un point de contact I la relation suivante :

$$\delta_n = \mathbf{n}^T \cdot (\mathbf{X}_A - \mathbf{X}_B) \quad (\text{Eq. IV-102})$$

avec :

- $\mathbf{n}^T = [n_x \ n_y \ n_z]$,
- $\mathbf{X}_A = [x_{C_A} \ y_{C_A} \ z_{C_A}]^T$,
- $\mathbf{X}_B = [x_{C_B} \ y_{C_B} \ z_{C_B}]^T$

On a les deux cas de figure suivants :

- si δ_n est positif, les objets ne sont pas en contact, donc $F_{C_n} = 0$,
- si δ_n est nul, les objets sont en contact, donc $F_{C_n} \geq 0$. On note que la vitesse et l'accélération ne sont pas nulles dans ce cas de figure.

Par conséquent, on définit le contact sans frottement par la condition suivante :

$$\delta_n \geq 0, F_{C_n} \geq 0, \delta_n \cdot F_{C_n} = 0 \quad (\text{Eq. IV-103})$$

Cette condition est appelée « condition de Signorini ».

Dans notre cas de figure, nous utilisons la condition de Signorini en vitesse comme le proposent Moreau [Moreau 88] [Moreau 03a], Jean [Jean 99] et Renouf [Renouf 04]. Comme nous le verrons ultérieurement, cette formulation nous facilitera la résolution des contacts frottants.

Ainsi, on écrit l'équation IV-103 en utilisant la vitesse normale relative entre les objets A et B au point de contact I :

$$\begin{aligned} \dot{\delta}_n &= v_n \\ &= \mathbf{n}^T \cdot (\mathbf{v}_{C_A} - \mathbf{v}_{C_B}) \end{aligned} \quad (\text{Eq. IV-104})$$

avec :

- $\mathbf{n}^T = [n_x \ n_y \ n_z]$,
- $\mathbf{v}_{C_A} = \begin{bmatrix} v_{x_{C_A}} & v_{y_{C_A}} & v_{z_{C_A}} \end{bmatrix}^T$,
- $\mathbf{v}_{C_B} = \begin{bmatrix} v_{x_{C_B}} & v_{y_{C_B}} & v_{z_{C_B}} \end{bmatrix}^T$.

On obtient alors une relation équivalente à l'équation IV-104 :

$$v_n \geq 0, F_{C_n} \geq 0, v_n \cdot F_{C_n} = 0 \quad (\text{Eq. IV-105})$$

C'est également un système de conditions de complémentarité.

En conclusion, on définit un repère local et les conditions de complémentarité pour chaque contact détecté entre deux objets. Cela revient à construire un problème de complémentarité linéaire pour l'ensemble des contacts détectés. Ce dernier est appelé « LCP » pour *Linear Complementarity Problem* (Cf. 3.2.2). Pour résoudre ce genre de problème, il est nécessaire d'effectuer un changement de repère : les équations exprimées dans le repère global doivent être réécrites dans le repère local, i.e. dans l'espace des contacts.

Considérons \mathbf{H} l'application linéaire qui permet de passer du niveau local au niveau global. \mathbf{H}^T est l'application transposée de \mathbf{H} . On obtient dans le repère global les forces de contact \mathbf{F}_A et \mathbf{F}_B exercées sur les objets A et B par :

$$\mathbf{F}_A = \mathbf{H}^T \cdot \mathbf{F}_{C_n} \quad (\text{Eq. IV-106})$$

$$\mathbf{F}_B = -\mathbf{H}^T \mathbf{F}_{C_n} \quad (\text{Eq. IV-107})$$

En écrivant l'équation IV-100 pour chaque objet dans le repère global, on obtient :

$$\mathbf{q}_A^{\text{contraint}} = \mathbf{C}_A \cdot \mathbf{F}_A + \mathbf{q}_A^{\text{libre}} \quad (\text{Eq. IV-108})$$

$$\mathbf{q}_B^{\text{contraint}} = \mathbf{C}_B \cdot \mathbf{F}_B + \mathbf{q}_B^{\text{libre}} \quad (\text{Eq. IV-109})$$

Si l'on ne considère que les déplacements suivant la normale, on obtient les équations suivantes à partir des vecteurs position :

$$\mathbf{X}_A^{\text{contraint}} = \mathbf{C}_A \cdot \mathbf{F}_A + \mathbf{X}_A^{\text{libre}} \quad (\text{Eq. IV-110})$$

$$\mathbf{X}_B^{\text{contraint}} = \mathbf{C}_B \cdot \mathbf{F}_B + \mathbf{X}_B^{\text{libre}} \quad (\text{Eq. IV-111})$$

D'où la distance $\delta_n^{\text{contraint}}$ au point de contact :

$$\begin{aligned} \delta_n^{\text{contraint}} &= \mathbf{H} \cdot \mathbf{X}_{C_A}^{\text{contraint}} - \mathbf{H} \cdot \mathbf{X}_{C_B}^{\text{contraint}} \\ &= \mathbf{H} \cdot (\mathbf{C}_A \cdot \mathbf{F}_A + \mathbf{X}_A^{\text{libre}}) - \mathbf{H} \cdot (\mathbf{C}_B \cdot \mathbf{F}_B + \mathbf{X}_B^{\text{libre}}) \\ &= \mathbf{H} \cdot \mathbf{C}_A \cdot \mathbf{H}^T \cdot \mathbf{F}_{C_n} + \mathbf{H} \cdot \mathbf{X}_A^{\text{libre}} + \mathbf{H} \cdot \mathbf{C}_B \cdot \mathbf{H}^T \cdot \mathbf{F}_{C_n} - \mathbf{H} \cdot \mathbf{X}_B^{\text{libre}} \quad (\text{Eq. IV-112}) \\ &= (\mathbf{H} \cdot \mathbf{C}_A \cdot \mathbf{H}^T + \mathbf{H} \cdot \mathbf{C}_B \cdot \mathbf{H}^T) \mathbf{F}_{C_n} + \mathbf{H} \cdot \mathbf{X}_A^{\text{libre}} - \mathbf{H} \cdot \mathbf{X}_B^{\text{libre}} \\ &= (\mathbf{H} \cdot \mathbf{C}_A \cdot \mathbf{H}^T + \mathbf{H} \cdot \mathbf{C}_B \cdot \mathbf{H}^T) \mathbf{F}_{C_n} + \delta_n^{\text{libre}} \end{aligned}$$

Ainsi, si l'on a plusieurs contacts, on résout le LCP suivant :

$$\begin{cases} \mathbf{F}_{C_n} \geq 0 \\ \delta_n^{\text{contraint}} \geq 0 \\ \mathbf{F}_{C_n} \perp \delta_n^{\text{contraint}} \end{cases} \Leftrightarrow \begin{cases} \mathbf{F}_{C_n} \geq 0 \\ \left(\sum_{i=1}^n [\mathbf{H} \cdot \mathbf{C}_i \cdot \mathbf{H}^T] \right) \mathbf{F}_{C_n} + \delta_{n_i}^{\text{libre}} \geq 0 \\ \mathbf{F}_{C_n} \perp \delta_n^{\text{contraint}} \end{cases} \quad (\text{Eq. IV-113})$$

avec i le nombre de contacts donnés par la méthode de détection de collision.

La matrice $[\mathbf{H} \cdot \mathbf{C}_i \cdot \mathbf{H}^T]$ porte le nom de l'opérateur de Delassus et elle contient toute l'information locale : c'est-à-dire l'orientation et les couplages entre contacts. On la note \mathbf{W}_i .

Cette approche dite locale revient à condenser les équations de la dynamique au niveau des contacts. La résolution se fera en plusieurs étapes. Dans un premier temps, on résoudra les contacts, puis une fois que toutes les réactions aux contacts auront été calculées, on vérifiera ces relations au niveau global, i.e. au niveau des objets.

Dans notre cas de figure, il faut résoudre les contacts multiples mais il est également nécessaire d'intégrer le mouvement rigide de la main. En effet, le squelette de la main est une chaîne polyarticulée rigide. Afin de tenir compte des mouvements du squelette, nous utilisons la méthode proposée par Ruspini et Khatib [Ruspini 97] pour les prendre en compte dans notre système de conditions de complémentarité.

3.3.3 Intégration du mouvement rigide d'une chaîne arborescente

L'équation de la dynamique en coordonnées généralisées est donnée par :

$$\boldsymbol{\tau} = \mathbf{A}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) \quad (\text{Eq. IV-114})$$

avec :

- $\boldsymbol{\tau}$: le couple généralisé appliqué à l'objet,
- $\mathbf{A}(\mathbf{q})$: la matrice d'énergie cinétique,
- $\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}})$: les forces centrifuges et les forces de Coriolis,
- $\mathbf{g}(\mathbf{q})$: la force de gravité.

On obtient ainsi l'accélération de l'objet dans l'espace libre des mouvements :

$$\ddot{\mathbf{q}} = \mathbf{A}^{-1}(\mathbf{q}) [\boldsymbol{\tau} - \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{g}(\mathbf{q})] \quad (\text{Eq. IV-115})$$

Une première intégration nous donne la vitesse, et une deuxième intégration nous donne la configuration sous l'action des forces extérieures.

La mécanique des solides rigides nous donne également l'expression de la vitesse d'un objet A (idem pour un objet B) dans l'espace :

$$\mathbf{v}_{C_A} = \mathbf{v}_A + \boldsymbol{\omega}_A \times \mathbf{r}_A \quad (\text{Eq. IV-116})$$

avec :

- \mathbf{v}_A et $\boldsymbol{\omega}_A$ les vitesses linéaire et angulaire de l'objet A,
- \mathbf{r}_A : le vecteur déplacement entre le point de contact C_A et l'origine O_A du repère de l'objet A.

Notons \mathbf{J}_A la matrice Jacobienne de l'objet A. Elle permet de passer de l'espace des mouvements à l'espace des contacts. On a donc :

$$\begin{bmatrix} \mathbf{v}_A \\ \boldsymbol{\omega}_A \end{bmatrix} = \mathbf{J}_A \cdot \dot{\mathbf{q}} \quad (\text{Eq. IV-117})$$

Au point de contact $C_A=C_B=I$, on obtient alors :

$$\mathbf{v}_{C_A} = \mathbf{J}_{C_A} \cdot \dot{\mathbf{q}} \quad (\text{Eq. IV-118})$$

ou
$$\mathbf{v}_{C_B} = \mathbf{J}_{C_B} \cdot \dot{\mathbf{q}} \quad (\text{Eq. IV-119})$$

ou
$$\mathbf{v}_I = \mathbf{J}_I \cdot \dot{\mathbf{q}} \quad (\text{Eq. IV-120})$$

A chaque point de contact I, on peut donc calculer la matrice Jacobienne \mathbf{J}_C qui décrit les vitesses relatives des objets à chaque point de contact, portées par les normales au contact. On peut alors écrire la vitesse dans l'espace des contacts sous la forme :

$$\mathbf{v}_n = \mathbf{J}_C \cdot \dot{\mathbf{q}} \quad (\text{Eq. IV-121})$$

Avec cette définition de la vitesse relative au contact, on exprime la matrice d'inertie dans l'espace des contacts sous la forme :

$$\mathbf{M}^{-1} = \mathbf{J}_C \cdot \mathbf{A}^{-1} \cdot \mathbf{J}_C^T \quad (\text{Eq. IV-122})$$

avec \mathbf{M} , la matrice de masse aux points de contact. La diagonale correspond aux contacts suivant la normale et les autres termes traduisent les couplages entre les deux objets. On a alors $\mathbf{M}^{-1} = \mathbf{C}$

Pour résoudre les forces de contact dans ce cas de figure, on exprime l'accélération au contact à partir de la vitesse relative. On obtient :

$$\begin{aligned} \mathbf{a}_n &= \dot{\mathbf{v}}_n \\ &= \mathbf{J}_C \cdot \ddot{\mathbf{q}} + \dot{\mathbf{J}}_C \cdot \dot{\mathbf{q}} \end{aligned} \quad (\text{Eq. IV-123})$$

En comparant avec l'équation de la dynamique, on peut exprimer l'accélération dans l'espace des contacts sous la forme d'une relation linéaire entre les forces de contact et une accélération issue du mouvement libre (i.e. mouvement sans forces de contact) [Baraff 89] [Ruspini 97] :

$$\mathbf{a}_{\text{rigide}}^{\text{contraint}} = \left[\mathbf{J}_C \cdot \mathbf{A}^{-1} \cdot \mathbf{J}_C^T \right] \cdot \mathbf{F}_{C_n} + \mathbf{a}_{\text{rigide}}^{\text{libre}} \quad (\text{Eq. IV-124})$$

avec $\mathbf{a}_{\text{rigide}}^{\text{contraint}} = \ddot{\delta}^{\text{contraint}}$

En intégrant deux fois l'accélération, on obtient donc le LCP suivant à résoudre :

$$\left\{ \begin{array}{l} \mathbf{F}_{C_n} \geq 0 \\ \delta_n^{\text{contraint}} \geq 0 \\ \mathbf{F}_{C_n} \perp \delta_n^{\text{contraint}} \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{l} \mathbf{F}_{C_n} \geq 0 \\ \left(\sum_{i=1}^n [\mathbf{H} \cdot \mathbf{C}_i \cdot \mathbf{H}^T] + \left[\mathbf{J}_C \cdot \left(\frac{\mathbf{A}}{\Delta t^2} \right)^{-1} \cdot \mathbf{J}_C \right] \right) \mathbf{F}_{C_n} + \delta_{n_i}^{\text{libre}} \geq 0 \\ \mathbf{F}_{C_n} \perp \delta_n^{\text{contraint}} \end{array} \right\} \quad (\text{Eq. IV-125})$$

La modélisation physique d'une tâche de préhension nécessite du frottement. En animation graphique et en simulation physique, le frottement est souvent modélisé par les lois de frottement de Coulomb. Nous allons donc présenter la méthode utilisée pour introduire des contacts frottants dans notre modèle de contact.

3.3.4 Résolution des contacts frottants

De nombreuses études sont menées au sujet de la résolution de contacts frottants. Nous citons entre autres Moreau [Moreau 88] [Moreau 03b], Jean [Jean 01], Glocker et Pfeiffer [Glocker 95], Pang et Trinkle [Pang 96], Song [Song 02], Renouf [Renouf 04] ou Duriez [Duriez 04]. Nous avons fait le choix d'utiliser les résultats de recherche de Duriez [Duriez 04] car il s'est orienté vers les problèmes

de contact pour les applications en temps-réel.

Dans ce paragraphe, nous exposons rapidement le problème des contacts multiples simultanés ainsi que la méthode utilisée dans notre cas de figure.

La formulation des lois de Coulomb dans l'espace et dans le cadre de contacts multiples n'est pas immédiate. Dans ce cas, ces lois conduisent à d'importantes non-linéarités.

Soient \mathbf{t}_1 et \mathbf{t}_2 les tangentes du repère local \mathcal{R}_c .

Soient \mathbf{F}_{t_1} et \mathbf{F}_{t_2} les forces tangentes définies positives. Notons λ le déplacement tangent d'un objet dans l'espace. La loi de Coulomb (Cf. Annexe A.2) s'écrit dans le cas 3D :

$$\sqrt{(\mathbf{F}_{t_1})^2 + (\mathbf{F}_{t_2})^2} \leq \mu F_n \quad (\text{Eq. IV-126})$$

avec μ le coefficient de frottement.

Comme la direction de la force tangente n'est pas connue à l'avance, on impose qu'elle s'oppose à la direction du glissement. On doit donc respecter la relation suivante :

$$\mathbf{F}_{t_1} + \mathbf{F}_{t_2} = -\mu \cdot \mathbf{F}_n \times \frac{\delta_{t_1} + \delta_{t_2}}{\|\delta_{t_1} + \delta_{t_2}\|} \quad (\text{Eq. IV-127})$$

On obtient donc pour l'ensemble des contacts un système d'équations à résoudre avec une formulation linéaire avec les conditions de complémentarité entre le déplacement et les efforts normaux, entre les déplacements et les efforts tangents, et la condition d'adhérence :

$$\left\{ \begin{array}{l} 0 \leq \delta_n \perp F_n \geq 0 \\ 0 \leq \lambda + \delta_{t_1} \perp F_{t_1} \geq 0 \\ 0 \leq \lambda + \delta_{t_2} \perp F_{t_2} \geq 0 \\ \vdots \\ 0 \leq \lambda + \delta_{t_p} \perp F_{t_p} \geq 0 \\ 0 \leq \mu F_n - \sum_{k=1}^p F_{t_k} \perp \lambda \geq 0 \end{array} \right. \quad (\text{Eq. IV-128})$$

Des travaux précédents ont fait une approximation du cône de frottement de Coulomb par une pyramide pour résoudre ce système d'équations : c'est la méthode de Lemke qui a été programmée par Dubois [Dubois 04] ainsi que par Lloyd [Lloyd 05]. Mais il existe une autre solution sans aucune approximation du cône de frottement qui utilise un algorithme de type Gauss-Seidel. Comme la formulation linéaire utilisée pour résoudre les contacts multiples avec frottement ne permet pas d'obtenir une solution directe, les inconnues seront donc déterminées de façon implicite en utilisant une méthode itérative. Pour cela, on utilise un algorithme de type Gauss Seidel non linéaire [Jean 01] [Moreau 03b] [Dubois 04], également appelé méthode « GSNL » ou « NSCD » pour *Non Smooth Contact Dynamics*. Cette méthode consiste à passer en revue cycliquement les contacts détectés sur le pas de temps. A chacun de ces passages, on résout un contact unique ; c'est-à-dire que l'on calcule localement la réaction

au contact en résolvant un problème unilatéral frottant pour le contact considéré. Les réactions des autres contacts sont alors considérées comme connues. Ce procédé itératif doit être initialisé par un choix de valeurs des réactions pour chaque contact. Afin d'accélérer la convergence à chaque pas de temps, on choisit comme valeurs d'initialisation les valeurs calculées au pas de temps précédent pour les contacts maintenus, sinon, on utilise les valeurs initiales qui ne sont pas forcément nulles.

L'algorithme de type Gauss Seidel non linéaire utilise un calcul de type implicite qui permet d'assurer la stabilité numérique et qui est beaucoup plus précis que la facettisation des cônes de frottement. En effet, cet algorithme est plus efficace en temps de calcul quand le nombre de contacts simultanés augmente [Duriez 04].

Dans notre cas de figure, nous avons pour objectif la simulation de tâches manuelles en environnement virtuel. Or, la saisie et la manipulation d'objets entraînent plusieurs points de contact frottants simultanés entre l'objet et la main. C'est pourquoi, nous avons choisi de résoudre les contacts par un algorithme de type Gauss Seidel pour notre problème de préhension d'objets par une main virtuelle.

3.3.5 Evaluation expérimentale

Dans ce paragraphe, nous présentons les résultats obtenus pour la résolution des contacts avec notre code.

3.3.5.1 Les conditions expérimentales

La plate-forme d'exécution est un processeur Pentium M 1,60 GHz avec 512 Mo de RAM. Les algorithmes présentés dans ce paragraphe font partie de la librairie **Collision**. Le diagramme de cette classe est présenté en Annexe A.5.

3.3.5.2 Principe de l'algorithme

Dans une simulation, les objets sont soumis à différentes forces classées en deux catégories :

- les forces dont la valeur est connue comme par exemple la gravité : elles sont dites « forces explicites ».
- les forces inconnues appliquées à l'objet comme par exemple les forces de contact : elles sont dites « implicites ».

Ainsi, au début de chaque pas de temps, nous réalisons une première intégration du mouvement à partir des forces explicites. Ce mouvement est qualifié de « mouvement libre » (Cf. § 3.3.1). Il permet d'obtenir les positions sur lesquelles est effectuée la détection de collision.

La méthode mise en oeuvre pour détecter les collisions entre les objets rigides de la scène et la main virtuelle déformable a été détaillée dans le paragraphe 2.4. Grâce aux algorithmes développés, nous pouvons identifier le moment et le lieu exacts de la collision entre la main et un objet ainsi que les normales au contact à la surface des objets. A partir de ces données, nous construisons alors l'espace des contacts (Cf. § 3.3.2). Nous obtenons alors le problème de Signorini à résoudre (problème de complémentarité linéaire – *LCP*) (Cf. eq. IV-113 et eq. IV-125). Ces systèmes d'équations sont résolus par un algorithme de type Gauss Seidel. Nous obtenons alors les forces de contact.

A ce niveau, nous effectuons une nouvelle intégration du mouvement pour le pas de temps avec l'ensemble des forces explicites et implicites. Ce mouvement est appelé « mouvement contraint ». Nous pouvons donc résumer ces étapes par l'algorithme suivant :

Début

```

Pour chaque pas de temps t
  Calcul du mouvement libre
  Détection de collision
  Construction du problème de complémentarité linéaire (LCP)
  Résolution du LCP par un algorithme de type Gauss Seidel
  Calcul du mouvement contraint
  Mise à jour de la cinématique et de la dynamique

```

Fin

Nous proposons d'évaluer les performances de cet algorithme dans le cadre de nos applications.

3.3.5.3 Exemple d'application

Pour obtenir des résultats, nous prenons l'exemple d'une sphère soumise à la gravité qui entre en contact avec un plan rigide fixe (Cf. Figure IV.96). Dans ce cas de figure, la sphère se déforme donc sous l'effet de son propre poids.

La sphère utilisée est un modèle élastique linéaire isotrope. Pour cet exemple, nous avons intégré les caractéristiques mécaniques d'un matériau de type caoutchouc avec $E = 0,01$ GPa et $\nu = 0,49$. Cet exemple simple s'apparente à la déformation de la pulpe d'un doigt au contact d'un plan rigide.

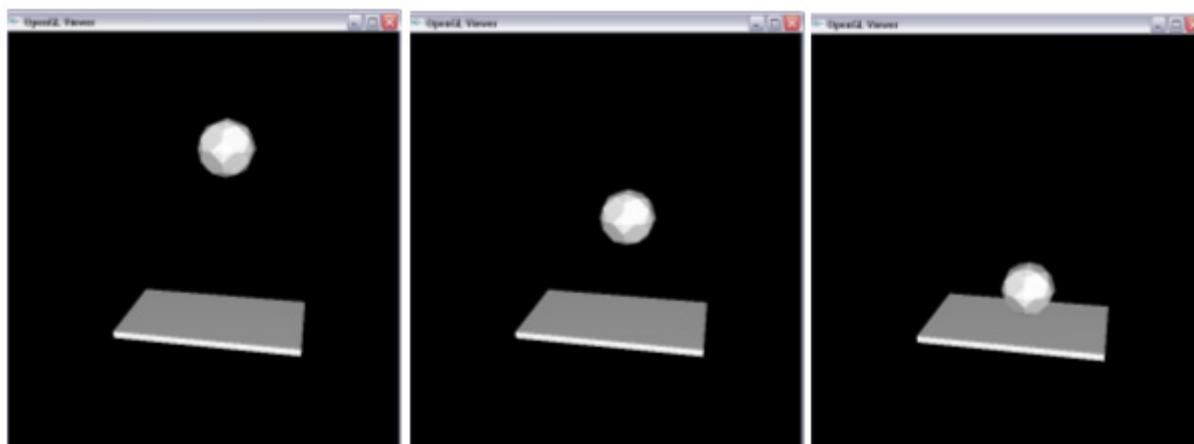


Figure IV.96 : Exemple utilisé pour évaluer le modèle de contact

Le Tableau IV.6 ci-dessous nous donne les résultats pour la construction de l'espace des contacts et la résolution du LCP par un algorithme de type Gauss Seidel en fonction du nombre de contacts et dans les cas de figure avec frottement ($\mu=0,4$) et sans frottement ($\mu=0$) :

	Nombre contacts	Construction espace des contacts (ms)	Résolution Gauss Seidel (ms)
MU=0	4	3,609	12,0555
	6	9,3104	23,8752
	8	14,6936	26,768
	14	14,15175	27,222
MU=0,4	4	4,5408333	17,278333
	6	9,31166667	23,10416667
	8	19,01416667	32,623333
	14	15,76416667	99,20916667

Tableau IV.6 : Les temps d'exécution pour la résolution des contacts

Nous remarquons que les temps de calcul augmentent rapidement avec le nombre de contacts pour se stabiliser vers une quinzaine de contacts (Cf. Figure IV.97 et Figure IV.98). Nous observons également que la résolution des contacts varie avec la présence ou non de frottement (Cf. Figure IV.97 et Figure IV.98). Elle est beaucoup plus rapide dans le cas de contacts normaux.

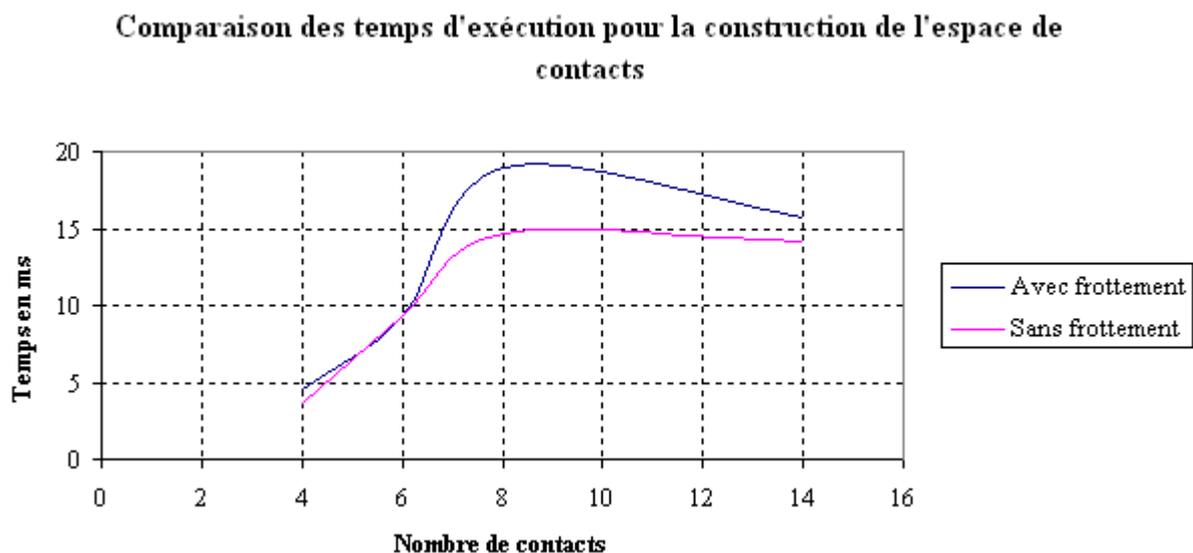


Figure IV.97 : Temps de construction de l'espace des contacts

Comparaison des temps d'exécution pour la résolution du LCP par un algorithme de type Gauss Seidel

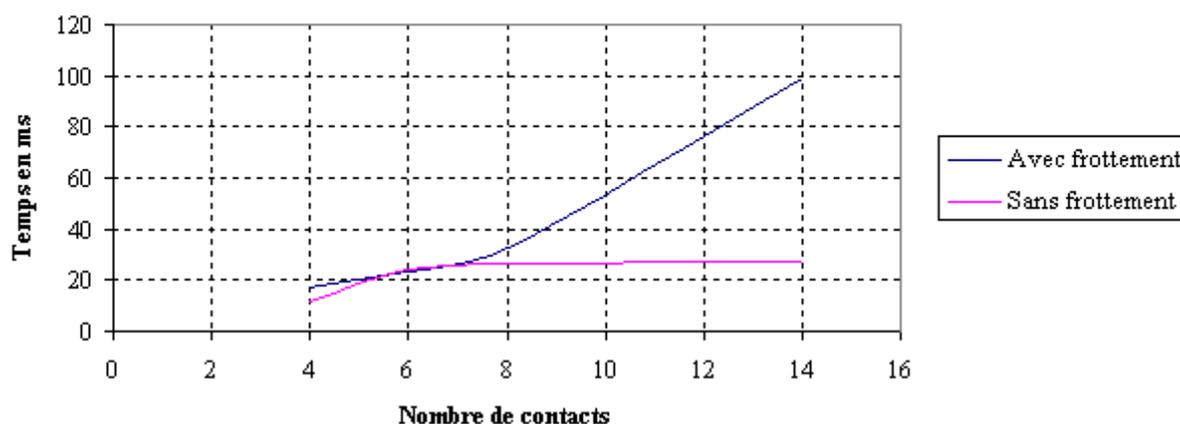


Figure IV.98 : Temps d'exécution pour la résolution du LCP

Sur ces courbes, nous constatons que le temps d'exécution minimum pour résoudre un LCP avec un algorithme de Gauss Seidel est de 12ms. De plus, les temps de calcul augmentent rapidement en fonction du nombre de contacts. Pour pouvoir répondre aux spécifications temps réel de nos applications, il est nécessaire d'optimiser notre programmation afin d'améliorer la convergence de l'algorithme de résolution des contacts. Ceci est d'autant plus important que le nombre de contacts simultanés générés lors d'une préhension d'objet est très grand.

3.3.6 Conclusion sur le modèle de contact mis en oeuvre

Le cas de la préhension d'objets en temps réel est une problématique complexe à résoudre. En situation réelle, une tâche manuelle entraîne plusieurs points de contact frottants simultanés afin de maintenir l'objet dans la main. Pour résoudre ces contraintes en animation graphique et satisfaire la contrainte de temps-réel, nous avons intégré un algorithme de type Gauss Seidel qui a donné des résultats encourageants pour des applications en Réalité Virtuelle (Cf. [Duriez 04] pour plus de détails et pour des résultats plus optimisés pour la résolution du LCP).

Les résultats obtenus ne permettent pas de respecter la contrainte de temps réel que nous nous sommes imposée. Les temps longs d'exécution enregistrés sont dus au manque d'optimisation du calcul de la matrice de Delassus (Cf. équation IV-113), dont la complexité de calcul est quadratique par rapport au nombre de contacts. Cette difficulté entraîne de fait la convergence faible de notre algorithme de type Gauss Seidel ; d'où un nombre important d'itérations pour approcher une solution.

Il serait donc important de revoir la programmation de la résolution du LCP par l'algorithme de type Gauss Seidel afin de pouvoir le mettre en oeuvre dans des applications en Réalité Virtuelle.

Après avoir expliqué la méthode utilisée pour la détermination des forces de contact entre les objets rigides et les mains déformables dans un environnement virtuel, nous présentons dans le paragraphe suivant les méthodes développées pour mettre à jour les déformations de la main et sa cinématique.

4 Le modèle de déformations

Dans notre simulation, les coussinets des mains virtuelles se déforment lorsqu'ils entrent en contact avec les objets rigides de l'environnement virtuel. Dans le paragraphe précédent, nous avons vu comment sont calculées les forces de réaction entre les objets et les mains lorsqu'ils interagissent. Dans ce paragraphe, nous expliquerons le modèle utilisé pour évaluer et mettre à jour les déformations des coussinets des mains virtuelles lors de la simulation des tâches manuelles.

La modélisation d'une main déformable pour simuler la préhension se révèle être une tâche difficile. En effet, comme nous l'avons vu dans les chapitres et paragraphes précédents, la main est un organe complexe constitué de plusieurs couches. Chacune d'entre elles possède des propriétés mécaniques spécifiques comme par exemple la loi de comportement ou encore le modèle de frottement.

Ainsi, le modèle de déformation que nous proposons est spécifique aux tâches manuelles en Réalité Virtuelle. Il doit prendre en compte les caractéristiques physiologiques de la main humaine tout en respectant la contrainte temps-réel. En tenant compte de ces contraintes, nous avons fait le choix d'utiliser la méthode des éléments finis afin d'obtenir un réalisme physique. Enfin, comme nous simulons la préhension d'objets virtuels dans le cas statique, nous avons mis en œuvre un modèle élastique linéaire isotrope en petites déformations pour les coussinets déformables de notre main virtuelle. En effet, lorsque les mains ne subissent pas de vibrations, le modèle linéaire est une bonne approximation du comportement des doigts [Asada 02]. De plus, ce modèle permet d'optimiser le temps de calcul en conduisant à un système d'équations linéaires à résoudre.

Nous commencerons par rappeler les principes de base de la méthode des éléments finis. Les explications données sont issues des ouvrages écrits par Bathe [Bathe 96], Garrigues [Garrigues 02] et Salençon [Salençon 05]. Puis, nous présenterons le problème à résoudre pour les coussinets déformables. Ensuite, nous expliquerons la méthode mise en œuvre pour la résolution des équations. Enfin, nous décrirons les résultats obtenus avec ce modèle dans les cas 2D et 3D.

4.1 Le principe des éléments finis

Un milieu continu est subdivisé en un nombre fini d'éléments discrets dont le comportement est représenté par un nombre fini de paramètres. La mise en œuvre de la méthode des éléments finis permet de résoudre un problème global continu par assemblage de ces éléments. On obtient ainsi les valeurs numériques de la solution approchée aux nœuds du maillage. Les différentes étapes de la méthode des éléments finis pour résoudre un problème sont les suivantes :

- 1) On pose un problème physique sous la forme d'une équation différentielle ou aux dérivées partielles à satisfaire sur un domaine Ω avec des conditions aux limites sur le bord $\partial\Omega$;
- 2) On construit une formulation intégrale du système différentiel à résoudre et de ses conditions aux limites. On l'appelle la formulation variationnelle du problème ;
- 3) On divise Ω en sous domaines : c'est le maillage. Les sous-domaines sont appelés « mailles ». Il faut faire un compromis entre la finesse de la représentation géométrique et le coût de calcul ;
- 4) On choisit la position des nœuds dans les sous-domaines et les fonctions qui définissent le champ local en fonction des valeurs aux nœuds. On les appelle « fonctions d'interpolation » et elles sont souvent des polynômes. Ces informations attachées à la maille donnent un « élément » ;

- 5) On définit une solution approchée à partir des valeurs aux nœuds des éléments : c'est la discrétisation ;
- 6) On résout alors le problème discret et on construit la solution approchée à partir des valeurs trouvées aux nœuds.

Maintenant, nous allons décrire notre problème à résoudre pour les coussinets déformables de notre main virtuelle.

4.2 Coussinets déformables en petites déformations

Dans notre cas de figure, nous posons l'hypothèse d'un modèle élastique linéaire isotrope pour un coussinet déformable de la main. Les hypothèses sont celles des petites déformations et des petits déplacements ; c'est-à-dire que l'on a un problème statique.

Nous nous intéressons à la formulation en déplacement pour les corps élastiques. Pour cela, nous recherchons les déplacements des points de l'objet qui permettent d'obtenir l'équilibre entre les forces extérieures, les forces de contact, les conditions initiales et les caractéristiques mécaniques des matériaux. Après avoir donné quelques définitions, nous présentons le problème global à résoudre, puis, nous expliquons les choix faits pour la discrétisation. Ensuite, nous résolvons le problème sous sa forme discrète.

4.2.1 Définitions et notations

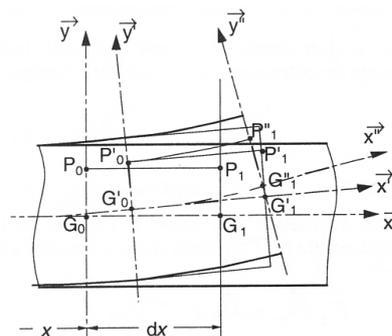
Soit un domaine Ω avec des conditions aux limites sur le bord $\partial\Omega$. On suppose que le domaine Ω est un volume V de frontière S et de normale extérieure \mathbf{n} . On définit sur ce domaine les paramètres suivants :

- On note \mathbf{U} le champ de déplacement. Chaque composante est appelée « degré de liberté ». Il y a donc trois degrés de liberté en translation (en mètre) – U_x , U_y et U_z – et trois degrés de liberté en rotation (en radian) – θ_x , θ_y et θ_z .
- On note $\boldsymbol{\varepsilon}$ le tenseur des déformations. Une déformation est le rapport de l'allongement à la longueur initiale (Cf. Figure IV.99). Il existe les déformations angulaires et les déformations linéaires. Une déformation est sans dimension.

Figure IV.99 : Définition de la déformation

L'élément de poutre dx se déplace et se déforme.

Le vecteur associé à la variation de la distance P_0P_1 est représenté par le vecteur $\mathbf{P}'_1\mathbf{P}''_1$, différence entre le déplacement de l'élément réel déformé et le déplacement de l'élément virtuel déplacé mais non déformé.



- On note $\boldsymbol{\sigma}$ le tenseur des contraintes. Une contrainte représente les efforts internes qui s'appliquent dans la structure (Cf. Figure IV.100). Il existe les contraintes normales – notées σ_{ii} – et les contraintes de cisaillement – notées σ_{ij} , avec i l'indice de définition de l'orientation de la facette et j l'indice de projection de la contrainte. Une contrainte est homogène à une pression ($\text{N.m}^{-2} \equiv \text{Pa}$).

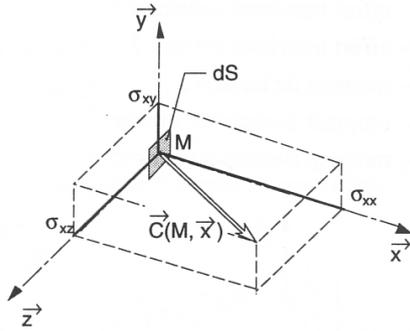


Figure IV.100 : Définition de la contrainte

- La contrainte normale au point M pour une normale orientée par \mathbf{x} est définie par $\sigma_{xx} = \mathbf{C}(\mathbf{M}, \mathbf{x}) \cdot \mathbf{x}$
- La contrainte tangentielle suivant \mathbf{y} au point M pour une normale orientée par \mathbf{x} est définie par $\sigma_{xy} = \mathbf{C}(\mathbf{M}, \mathbf{x}) \cdot \mathbf{y}$
- La contrainte tangentielle suivant \mathbf{z} au point M pour une normale orientée par \mathbf{x} est définie par $\sigma_{xz} = \mathbf{C}(\mathbf{M}, \mathbf{x}) \cdot \mathbf{z}$

- On note \mathbf{F} les efforts exercés avec \mathbf{F}_s les efforts surfaciques et \mathbf{F}_v les efforts volumiques.
- Soit dS un élément de surface sur une partie d'un milieu continu. Le vecteur $d\mathbf{F}_s$ qui caractérise les forces de contact exercées sur dS est défini à partir de la contrainte exercée sur cet élément :

$$\begin{aligned} d\mathbf{F}_s &= -\boldsymbol{\sigma} \cdot \mathbf{n} \cdot dS \\ &= -\boldsymbol{\sigma}_s \cdot dS \end{aligned} \quad (\text{Eq. IV-129})$$

avec \mathbf{n} la normale à la surface de l'élément.

4.2.2 Formulation variationnelle du problème

En élasticité linéaire et dans le cas statique, les forces volumiques exercées sur l'objet doivent être en équilibre avec l'état de contrainte au niveau de l'élément [Bathe 96] [Cotin 99] [Brown 03] [Hyncik 02]. Notre problème revient à trouver le champ de déplacement tel que :

$$\text{div} \boldsymbol{\sigma} + \mathbf{F} \cdot \mathbf{v} = 0 \quad (\text{Eq. IV-130})$$

avec $\boldsymbol{\sigma}$ la contrainte à l'intérieur de l'objet.

Les forces exercées sur les frontières et les déplacements imposés sur les frontières n interviennent que dans les conditions limites. On a donc sur la frontière S :

- les déplacements \mathbf{U}_0 imposés :

$$\mathbf{U}(\mathbf{P}) - \mathbf{U}_0(\mathbf{P}) = \mathbf{0} \quad \forall \mathbf{P} \in S_d \quad (\text{Eq. IV-131})$$

- les contraintes $\boldsymbol{\sigma}_s$ imposées :

$$\boldsymbol{\sigma}(\mathbf{P}) \otimes \overline{\mathbf{n}}(\mathbf{P}) = \mathbf{q}_0(\mathbf{P}) \quad \forall \mathbf{P} \in S_f \quad (\text{Eq. IV-132})$$

avec S_d pour les déplacements et S_f pour les forces exercées sur la surface S .

L'équation IV-130 est une équation vectorielle qui peut se mettre sous la forme variationnelle suivante :

$$\int_V \boldsymbol{\psi} \otimes (\text{div} \boldsymbol{\sigma} + \mathbf{F}) dV = 0 \quad \forall \boldsymbol{\psi}(\mathbf{P}) \quad (\text{Eq. IV-133})$$

avec $\boldsymbol{\psi}(\mathbf{P})$ un champ vectoriel dans Ω .

On obtient le problème équivalent suivant :

$$-\int_V \overline{\overline{\boldsymbol{\sigma}}} \otimes \overline{\overline{\mathbf{grad} \boldsymbol{\psi}}} dV + \int_S \overline{\boldsymbol{\psi}} \otimes \overline{\boldsymbol{\sigma}} \otimes \overline{\mathbf{n}} dS + \int_V \overline{\boldsymbol{\psi}} \otimes \overline{\mathbf{F}} dV = 0 \quad \forall \boldsymbol{\psi}(\mathbf{M}) \quad (\text{Eq. IV-134})$$

On obtient ainsi les contraintes sur la surface des objets. Cette intégrale de surface, appelée « intégrale de bord » permet de tenir compte des conditions aux limites (IV-131) et (IV-132). Ces contraintes aux limites peuvent être nulles.

On obtient à la surface :

$$\int_S \overline{\boldsymbol{\psi}} \otimes \overline{\boldsymbol{\sigma}} \otimes \overline{\mathbf{n}} dS = \int_{S_d} \overline{\boldsymbol{\psi}} \otimes \overline{\boldsymbol{\sigma}} \otimes \overline{\mathbf{n}} dS + \int_{S_f} \overline{\boldsymbol{\psi}} \otimes \overline{\boldsymbol{\sigma}} \otimes \overline{\mathbf{n}} dS \quad (\text{Eq. IV-135})$$

On peut alors éliminer la condition aux limites de contraintes imposées en la reportant dans l'intégrale :

$$\int_{S_f} \overline{\boldsymbol{\psi}} \otimes \overline{\boldsymbol{\sigma}} \otimes \overline{\mathbf{n}} dS = \int_{S_f} \overline{\boldsymbol{\psi}} \otimes \overline{\mathbf{q}_0} dS \quad (\text{Eq. IV-136})$$

Le problème devient donc :

$$-\int_V \overline{\overline{\boldsymbol{\sigma}}} \otimes \overline{\overline{\mathbf{grad} \boldsymbol{\psi}}} dV + \int_V \overline{\boldsymbol{\psi}} \otimes \overline{\mathbf{F}} dV + \int_{S_f} \overline{\boldsymbol{\psi}} \otimes \overline{\mathbf{q}_0} dS + \int_{S_d} \overline{\boldsymbol{\psi}} \otimes \overline{\boldsymbol{\sigma}} \otimes \overline{\mathbf{n}} dS = 0 \quad \forall \boldsymbol{\psi}(\mathbf{M}) \quad (\text{Eq. IV-137})$$

Si l'on considère que le champ $\boldsymbol{\psi}$ est un champ de vitesse virtuelle, on obtient l'expression du théorème des puissances virtuelles. Ainsi, cette équation signifie que la puissance virtuelle des efforts intérieurs et extérieurs est nulle dans tout champ de vitesse virtuelle appliqué à la solution.

Comme la solution exacte du problème sous forme variationnelle est inaccessible $\forall \boldsymbol{\psi}(\mathbf{M})$, on va chercher à satisfaire cette équation que pour certains $\boldsymbol{\psi}(\mathbf{M})$: ce qui est une condition moins contraignante. Ainsi, il faut déterminer les fonctions qui respectent l'équation (IV-137) et les conditions limites. Pour construire une solution approchée du problème variationnel, on utilise la discrétisation.

4.2.3 Discrétisation des coussinets

Tout d'abord, il faut diviser le domaine continu Ω en sous-domaines [Bathe 96] [Guarrigues 02]. Pour ce faire, nous devons définir un maillage du continuum en éléments finis. Il existe différentes formes d'éléments finis comme le triangle ou le quadrilatère en 2D ou encore le tétraèdre ou l'hexaèdre en 3D.

Chaque élément fini possède un certain nombre de nœuds – trois pour un triangle et quatre pour un tétraèdre. Les nœuds forment l'ensemble discret des points du système en lesquels les propriétés physiques sont calculées.

Le principe de la méthode des éléments finis étant d'interpoler les champs physiques en tout point du continuum en fonction de leur valeur aux nœuds, diverses fonctions d'interpolation peuvent être choisies. La plus simple est une fonction d'interpolation linéaire : seuls les nœuds aux sommets sont utilisés comme éléments finis.

Dans notre cas de figure, nous considérons deux maillages :

- un maillage surfacique avec le triangle comme maille,
- un maillage volumique avec le tétraèdre comme maille. Le tétraèdre est la figure tridimensionnelle la plus simple.

Nous avons fait le choix d'utiliser des mailles linéaires car les fonctions d'interpolation prennent la même valeur au nœud commun. Les éléments linéaires assurent la continuité C_0 ce qui permet de respecter les théorèmes de convergence de la méthode des éléments finis. De plus, une interpolation linéaire sur les éléments finis limite au maximum le temps de calcul.

Nous devons donc obtenir l'expression générale du champ des déplacements à l'intérieur d'un triangle (resp. un tétraèdre) en fonction du déplacement de ses trois sommets (resp. quatre sommets), en faisant l'hypothèse que ce champ est linéaire. Pour cela, nous devons exprimer les coordonnées d'un point quelconque à l'intérieur du triangle (resp. tétraèdre) en fonction des positions des sommets (Cf. Annexe A.4 pour le détail des calculs).

Soit un point P quelconque à l'intérieur de l'élément fini. Le déplacement de P est noté $\mathbf{U}(\mathbf{P})$ et les déplacements des nœuds sont notés $\mathbf{U}(\mathbf{v}_i)$, avec v_i , avec $i=0, \dots, 2$. Les déplacements des nœuds sont les inconnues du système d'équations.

Nous introduisons les fonction g_i , appelées « fonctions test » ou « fonctions d'interpolation », qui définissent les déplacements de chaque point à l'intérieur d'un élément à partir des déplacements des nœuds de la maille considérée.

- Pour un triangle, nous obtenons alors la relation suivante [Cotin 99] [Hyncik 02] (Cf. Figure IV.101) :

$$\begin{aligned} \mathbf{U}(\mathbf{P}) &= g_1 \mathbf{U}(\mathbf{v}_1) + g_2 \mathbf{U}(\mathbf{v}_2) + g_3 \mathbf{U}(\mathbf{v}_3) \\ &= \frac{\text{Aire}_1}{\text{Aire totale}} \mathbf{U}(\mathbf{v}_1) + \frac{\text{Aire}_2}{\text{Aire totale}} \mathbf{U}(\mathbf{v}_2) + \frac{\text{Aire}_3}{\text{Aire totale}} \mathbf{U}(\mathbf{v}_3) \end{aligned} \quad (\text{Eq. IV-138})$$

avec v_i , avec $i=0, \dots, 2$, ses trois sommets.

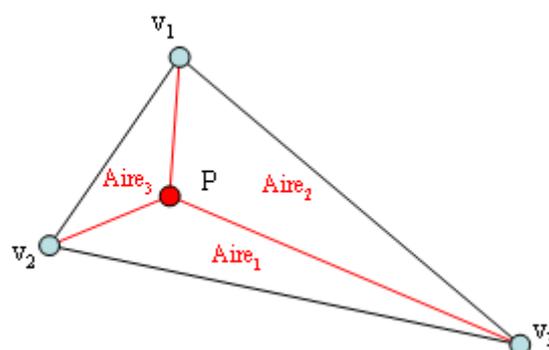


Figure IV.101 : Fonctions d'interpolation pour un triangle

- Pour un tétraèdre, nous obtenons la relation suivante [Cotin 99] [Delingette 04] (Cf. Figure IV.102) :

$$\mathbf{U}(\mathbf{P}) = -\sum_{i=0}^3 \frac{\mathbf{A}_i \cdot (\mathbf{P} - \mathbf{p}_{i+1})}{6V(T_e)} \mathbf{U}(\mathbf{p}_i) = \mathbf{G} \cdot \begin{bmatrix} \mathbf{U}(\mathbf{p}_0) \\ \mathbf{U}(\mathbf{p}_1) \\ \mathbf{U}(\mathbf{p}_2) \\ \mathbf{U}(\mathbf{p}_3) \end{bmatrix} \quad (\text{Eq. IV-139})$$

avec :

- \mathbf{p}_i les quatre sommets, avec $i=0, \dots, 3$, tels que $\mathbf{p}_i = (p_{ix}, p_{iy}, p_{iz})^T$;
- $\mathbf{U}(\mathbf{p}_i)$ les vecteurs déplacements des sommets ;
- $V(T_e)$ le volume du tétraèdre ;
- \mathbf{A}_i le vecteur normal de la surface opposée au sommet \mathbf{p}_i considéré ;
- \mathbf{G} la matrice des fonctions d'interpolations g_i

Les fonctions d'interpolation $g_i(M)$ correspondent à l'interpolation linéaire à l'intérieur du tétraèdre. Ces fonctions définissent les coordonnées barycentriques du point P à l'intérieur du tétraèdre en fonction des coordonnées des sommets du tétraèdre.

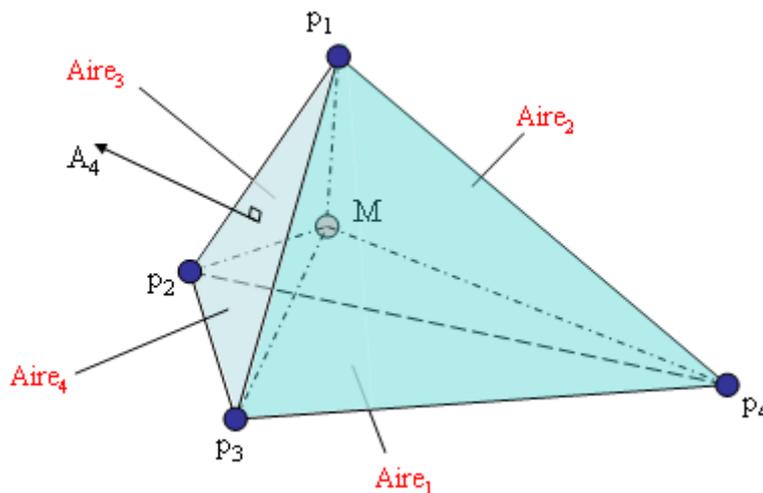


Figure IV.102 : Fonctions d'interpolation pour un tétraèdre

Nous obtenons donc un système linéaire en fonction du déplacement \mathbf{U} . La construction de ce nouveau système d'équations s'appelle « l'assemblage ».

Une fois le déplacement défini par les fonctions d'interpolation, nous calculons la contrainte à l'intérieur de chaque élément. Tout d'abord, nous calculons la déformation qui résulte de l'action de la force de réaction à travers chaque élément. Elle est notée $\boldsymbol{\varepsilon}$ et elle est liée au déplacement \mathbf{U} par le tenseur non linéaire de Green – Lagrange :

$$\varepsilon_{ij}(\mathbf{U}) = \frac{1}{2} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} + \sum_{k=1}^3 \frac{\partial U_k}{\partial x_i} \frac{\partial U_k}{\partial x_j} \right) \quad (\text{Eq. IV-140})$$

avec $1 \leq i, j \leq 3$.

Il est invariant en rotation et en translation. Les termes diagonaux du tenseur de déformation

représentent la déformation dans la direction normale, alors que les autres termes représentent le cisaillement.

Dans notre cas de figure, nous avons fait l'hypothèse de petits déplacements [Bro Nielsen 98] [Cotin 99] [Brown 03] [Delingette 04] : les termes d'ordre 2 peuvent donc être négligés. On linéarise le tenseur de Green – Lagrange et on obtient le tenseur de déformation de Cauchy :

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \quad (\text{Eq. IV-141})$$

avec $1 \leq i, j \leq 3$ et \mathbf{U} un vecteur déplacement défini par les composantes $u(x, y, z)$, $v(x, y, z)$ et $w(x, y, z)$.

Nous obtenons donc le tenseur de déformation $\boldsymbol{\varepsilon}$ en fonction du déplacement par la relation suivante (Cf. Annexe A.4 pour le détail des calculs) :

$$\boldsymbol{\varepsilon} = \mathbf{B} \cdot \mathbf{U} \quad (\text{Eq. IV-142})$$

Nous exprimons le champ de déplacement d'un point de l'élément discret en fonction des déplacements des nœuds :

$$\mathbf{U}(\mathbf{M}) = g_0 \mathbf{U}(\mathbf{p}_0) + g_1 \mathbf{U}(\mathbf{p}_1) + g_2 \mathbf{U}(\mathbf{p}_2) + g_3 \mathbf{U}(\mathbf{p}_3) \quad (\text{Eq. IV-143})$$

Or, pour tous les points à l'intérieur du tétraèdre, la déformation est supposée constante. Comme nous utilisons des fonctions d'interpolation linéaires, nous avons donc une relation linéaire entre le tenseur des déformations $\boldsymbol{\varepsilon}$ et les déplacements \mathbf{U}_i des nœuds d'un élément. Ainsi, la matrice \mathbf{B} de l'équation (IV-146) est une matrice constante (Cf. Annexe A.4 pour le détail des calculs).

Comme il existe également une relation entre l'état de déformation et l'état de contrainte, nous calculons par conséquent le tenseur des contraintes $\boldsymbol{\sigma}$ exercées dans l'élément. Physiquement, ces deux états sont reliés par une loi de comportement du matériau. On note \mathbf{C} la matrice d'élasticité de notre objet. D'où la relation contrainte/déformation de la forme :

$$\boldsymbol{\sigma} = \mathbf{C} \cdot \boldsymbol{\varepsilon} \quad (\text{Eq. IV-144})$$

Dans le paragraphe qui suit, nous explicitons la loi de comportement utilisée.

4.2.4 Calcul de la contrainte à partir de la loi de comportement

Dans notre cas de figure, nous avons fait l'hypothèse d'une déformation élastique, linéaire et isotrope afin de réduire la complexité du matériau biologique réel [Bro Nielsen 98] [Cotin 99] [Brown 03] [Delingette 04]. Ainsi, la relation entre la contrainte et la déformation est linéaire et la matrice \mathbf{C} est une matrice de constantes. Cette relation s'appelle « la loi de comportement de Hooke ». On a donc l'équation suivante entre la contrainte et la déformation :

$$\boldsymbol{\sigma} = 2\mu \boldsymbol{\varepsilon} + \lambda \text{tr}(\boldsymbol{\varepsilon}) \mathbf{I}_d \quad (\text{Eq. IV-145})$$

avec :

- \mathbf{I}_d la matrice identité 3x3,
- $\boldsymbol{\varepsilon}$ le tenseur de déformation,

- μ et λ les constantes de Lamé qui sont directement liées au module d'Young E et au coefficient de Poisson ν du matériau :

$$\mu = \frac{E}{2(1+\nu)} \text{ et } \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$$

On obtient donc en écriture matricielle :

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{bmatrix} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ & 1 & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ & & 1 & 0 & 0 & 0 \\ & & & \frac{1-2\nu}{2(1-\nu)} & 0 & 0 \\ \text{Sym} & & & & \frac{1-2\nu}{2(1-\nu)} & 0 \\ & & & & & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{bmatrix} \quad (\text{Eq. IV-146})$$

En remplaçant ε par la relation (IV-142), nous obtenons alors la relation matricielle simplifiée suivante entre la contrainte et le déplacement des noeuds :

$$\boldsymbol{\sigma} = \mathbf{C} \cdot \mathbf{B} \cdot \mathbf{U} \quad (\text{Eq. IV-147})$$

Ainsi, en élasticité linéaire, à partir d'une solution approchée du champ des déplacements, nous déduisons le champ des tenseurs de déformation, puis le champ des tenseurs de contrainte. Il ne reste alors plus qu'à calculer la force exercée sur notre objet. Pour cela, résolvons le problème sous une forme approchée.

4.2.5 Résolution du problème discrétisé

Dans les paragraphes précédents, nous avons discrétisé notre domaine (i.e. le coussinet) en éléments, nous avons défini les fonctions d'interpolation entre un point quelconque d'un élément et ses nœuds, et nous avons choisi notre loi de comportement. Nous avons obtenu une relation linéaire entre les déplacements des nœuds d'un élément et la contrainte exercée dans cet élément.

Ainsi, l'équation de l'équilibre (IV-134), qui est exprimée sous sa forme variationnelle par l'équation (IV-141), nous permet d'établir une relation entre la contrainte et la force exercées sur un élément [Bathe 96] [Garrigues 02]. On la note :

$$\begin{aligned} \mathbf{F}_e &= \mathbf{A} \cdot \boldsymbol{\sigma}_e \\ &= \mathbf{A} \cdot \mathbf{C} \cdot \mathbf{B} \cdot \mathbf{U}_e \end{aligned} \quad \text{ou encore} \quad \mathbf{F}_e = \mathbf{K}_e \cdot \mathbf{U}_e \quad (\text{Eq. IV-148})$$

avec :

- \mathbf{F}_e , la force exercée sur l'élément.
- \mathbf{K}_e , la matrice de raideur de l'élément,
- \mathbf{U}_e , les déplacements des nœuds,

- **A**, la relation force/contrainte,
- **C**, la matrice d'élasticité (i.e. la relation contrainte/déformation),
- **B**, la transformation des déplacements des nœuds en déformations.

Comme ces matrices sont constantes, trouver **F** revient donc à résoudre le système linéaire (IV-146).

Pour calculer la matrice **K_e** pour un élément, on utilise le principe des déplacements virtuels. On écrit que le travail des forces extérieures **F_e** appliquées à l'élément pour un déplacement virtuel **δU_e** est égal au travail interne des contraintes **δε_e** pour ce même déplacement :

$$\int_V \delta \boldsymbol{\varepsilon}_e^T \cdot \boldsymbol{\sigma}_e \, dV = \int_V \delta \mathbf{U}_e^T \cdot \mathbf{F}_{V_e} \, dV + \int_S \delta \mathbf{U}_e^T \cdot \mathbf{F}_{S_e} \, dS \quad (\text{Eq. IV-149})$$

D'après l'équation de l'équilibre, il existe deux types d'efforts surfaciques : les efforts liés aux déplacements et ceux liés aux contraintes. Nous obtenons alors :

$$\int_V \delta \boldsymbol{\varepsilon}_e^T \cdot \boldsymbol{\sigma}_e \, dV = \int_V \delta \mathbf{U}_e^T \cdot \mathbf{F}_{V_e} \, dV + \int_{S_f} \delta \mathbf{U}_e^T \cdot \mathbf{q}_{0_e} \, dS + \int_{S_d} \delta \mathbf{U}_e^T \cdot \boldsymbol{\sigma}_e \cdot \mathbf{n} \, dS \quad (\text{Eq. IV-150})$$

En remplaçant **ε_e** et **σ_e** à partir des relations (IV-142) et (IV-147), nous avons :

$$\begin{aligned} \int_V \delta \boldsymbol{\varepsilon}_e^T \cdot \boldsymbol{\sigma}_e \, dV &= \int_V (\mathbf{B} \cdot \delta \mathbf{U}_e)^T (\mathbf{C} \cdot \mathbf{B} \cdot \delta \mathbf{U}_e) \, dV \\ &= \int_V \delta \mathbf{U}_e^T \cdot \mathbf{B}^T \cdot \mathbf{C} \cdot \mathbf{B} \cdot \delta \mathbf{U}_e \, dV \\ &= \delta \mathbf{U}_e^T \left(\int_V \mathbf{B}^T \cdot \mathbf{C} \cdot \mathbf{B} \, dV \right) \delta \mathbf{U}_e \end{aligned} \quad (\text{Eq. IV-151})$$

Ainsi, le principe des déplacements virtuels permet de résoudre de manière approchée le problème variationnel (IV-137) par le biais du système IV-152 suivant :

$$\begin{aligned} \delta \mathbf{U}_e^T \left(\int_V \mathbf{B}^T \cdot \mathbf{C} \cdot \mathbf{B} \, dV \right) \delta \mathbf{U}_e &= \int_V \delta \mathbf{U}_e^T \cdot \mathbf{F}_{V_e} \, dV + \int_{S_f} \delta \mathbf{U}_e^T \cdot \mathbf{q}_{0_e} \, dS + \int_{S_d} \delta \mathbf{U}_e^T \cdot \boldsymbol{\sigma}_e \cdot \mathbf{n} \, dS \\ \Leftrightarrow \delta \mathbf{U}_e^T \left(\int_V \mathbf{B}^T \cdot \mathbf{C} \cdot \mathbf{B} \, dV \right) \delta \mathbf{U}_e &= \delta \mathbf{U}_e^T \cdot \left(\int_V \mathbf{F}_{V_e} \, dV + \int_{S_f} \mathbf{q}_{0_e} \, dS + \int_{S_d} \boldsymbol{\sigma}_e \cdot \mathbf{n} \, dS \right) \\ \Leftrightarrow \left(\int_V \mathbf{B}^T \cdot \mathbf{C} \cdot \mathbf{B} \, dV \right) \delta \mathbf{U}_e &= \left(\int_V \mathbf{F}_{V_e} \, dV + \int_{S_f} \mathbf{q}_{0_e} \, dS + \int_{S_d} \boldsymbol{\sigma}_e \cdot \mathbf{n} \, dS \right) \end{aligned}$$

Ce système peut se mettre sous la forme **F_e = K_e · U_e** et on obtient une valeur approchée de **K_e** pour un élément :

$$\mathbf{K}_e = \int_V \mathbf{B}^T \cdot \mathbf{C} \cdot \mathbf{B} \, dV \quad (\text{Eq. IV-153})$$

La matrice **K_e** est symétrique.

On construit la matrice de raideur de l'objet discrétisé par l'addition des matrices de raideur des éléments. On la nomme « matrice d'assemblage » et on la note \mathbf{K}_{ass} :

$$\mathbf{K}_{\text{ass}} = \sum_m \left(\int_v \mathbf{B}^T \cdot \mathbf{C} \cdot \mathbf{B} \, dV \right) \quad (\text{Eq. IV-154})$$

avec m le nombre d'éléments.

Il faut également connaître les forces et les contraintes globales exercées sur l'objet discrétisé. Pour ce faire, on utilise les fonctions d'interpolation g_i définies dans le paragraphe 4.2.3. Par exemple, pour les forces de volume, on obtient la relation suivante :

$$\mathbf{F}_v = \sum_m \left(\int_v \mathbf{G}^T \cdot \mathbf{F}_v \, dV \right) \quad (\text{Eq. IV-155})$$

Ainsi, dans le cas statique, on résout un problème variationnel en suivant les étapes suivantes :

- 1) On discrétise l'objet en choisissant un maillage adapté (type de la maille et précision) ;
- 2) On raisonne sur un élément et on calcule la matrice de raideur \mathbf{K}_e de chaque élément ;
- 3) On assemble toutes les matrices \mathbf{K}_e de notre objet pour construire la matrice globale \mathbf{K}_{ass} ;
- 4) On construit le vecteur de chargement \mathbf{F} qui modélise les forces exercées sur l'objet : les forces extérieures et les forces de contact ;
- 5) On prend en compte les conditions limites sur les déplacements. Ainsi, certains nœuds de la structure sont considérés comme fixes et donc avec des déplacements nuls. Comme leurs ligne et colonne sont nulles, on les supprime pour ne pas réaliser de calculs inutiles. On obtient alors une matrice $\tilde{\mathbf{K}}$. Ce qui donne la relation suivante :
 $\mathbf{F} = \tilde{\mathbf{K}} \cdot \mathbf{U}$
- 6) On calcule le déplacement \mathbf{U} en résolvant l'équation $\mathbf{U} = \tilde{\mathbf{K}}^{-1} \cdot \mathbf{F}$. Il faut donc inverser la matrice $\tilde{\mathbf{K}}$.
- 7) Puis l'on obtient la déformation et la contrainte appliquées sur l'objet à partir des relations (IV-142) et (IV-144).

La construction de la matrice \mathbf{K}_{ass} à partir des déplacements s'appelle la « méthode directe ». Elle peut devenir très coûteuse en temps de calcul. Certains objets sont maillés finement : ce qui entraîne un nombre important d'éléments, et qui requiert un temps de calcul élevé pour l'évaluation des matrices de raideur \mathbf{K}_e .

4.3 Principe des algorithmes développés

Les opérations les plus coûteuses en temps de calcul sont la construction du maillage et le calcul des matrices de raideur.

Le maillage est construit dans une phase préliminaire de calculs. Comme nous l'avons vu dans le paragraphe 2.4. Toutes les données relatives au maillage sont stockées dans des tableaux afin d'optimiser la détection de collision. De plus, l'utilisation d'éléments linéaires tétraédriques et l'intégration d'une loi de comportement de matériau linéaire élastique et isotrope permettent d'améliorer les temps de calcul.

Quant aux matrices de raideur, elles sont pré-calculées et stockées en mémoire mais il est nécessaire de recalculer en ligne certaines matrices car la forme des objets change lorsqu'ils

entrent en contact. Ces équations matricielles sont longues à résoudre et elles peuvent pénaliser la simulation au niveau du temps CPU.

Nous présentons dans la suite les tests développés afin de valider notre modèle éléments finis.

4.4 Les conditions expérimentales

La plate-forme d'exécution est un processeur Pentium M 1,60 GHz avec 512 Mo de RAM.

En première approche, nous avons programmé la méthode des éléments finis sous Matlab © afin de prototyper la bibliothèque finale. Ce travail a consisté à modéliser le doigt humain et à simuler différents mouvements dans le cas plan. La description des tests et des résultats est donnée dans le paragraphe 4.5.

Ensuite, la méthode des éléments finis a été programmée en langage C/C++ et elle a été intégrée dans la bibliothèque **Collision** précédemment décrite. La description de la classe pour un objet maillé est donnée en Annexe A.5. Les résultats obtenus pour le modèle de déformation sont décrits dans le paragraphe 4.6.

Pour ces exemples, nous avons intégré les caractéristiques mécaniques d'un matériau de type caoutchouc avec $E = 0,01$ GPa et $\nu = 0,49$.

4.5 Le modèle dans le cas 2D

Nous avons développé un modèle éléments finis dans le cas plan afin de tester et valider les modèles de contact et de déformation proposés dans les paragraphes précédents. Le logiciel utilisé est Matlab ©.

Comme ce modèle a été la maquette de la bibliothèque **Collision**, nous retrouvons dans ce paragraphe certains tests et résultats présentés dans les paragraphes 2.4 et 3.3.

Tout d'abord, nous présentons le modèle ainsi que les différentes simulations réalisées.

4.5.1 Description du modèle

Chaque doigt est constitué d'une partie rigide, qui représente le squelette, et d'une partie déformable, qui modélise les coussinets adipeux sous cutanés palmaires. La Figure IV.103 ci-dessous présente le modèle de doigt utilisé ainsi que les paramètres de la simulation :

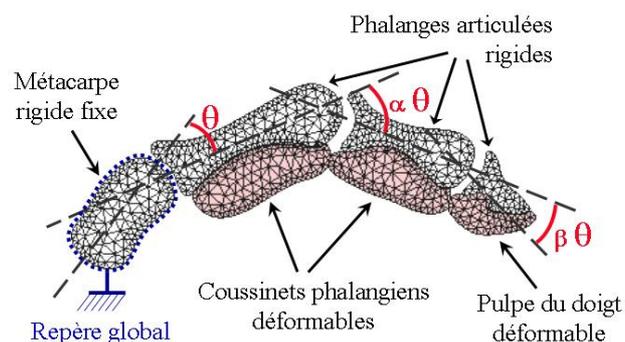


Figure IV.103 : Description du modèle de doigt utilisé

Les coussinets et les os sont maillés par des triangles et des tétraèdres à l'aide du mailleur GID [wGid]. Comme le squelette est considéré comme rigide, les nœuds des os sont fixes. Les nœuds des coussinets se déplacent selon les efforts exercés. Nous considérons comme fixes les nœuds qui sont à proximité du squelette rigide.

Le mouvement de chaque phalange est piloté par un angle θ qui varie dans un intervalle défini par l'utilisateur. Les mouvements des deuxième et troisième phalanges respectent les relations de dépendance décrites dans le Chapitre I [Lansmeer 55] [Rijkema 91] par la mise en place des

coefficients α et β . Ainsi, d'après la description de la main humaine faite dans le Chapitre I, nous obtenons les valeurs suivantes pour ces coefficients (Cf. § 4.2 du Chapitre I) :

$$\alpha = 2 \text{ et } \beta = 4/3 \quad (\text{Eq. IV-156})$$

Nous utilisons le modèle de contact frottant et le modèle de déformation par éléments finis décrits respectivement dans les paragraphes 3.3 et 4.2. Les contraintes sont dues aux actions de contact sur les coussinets.

Les résultats obtenus avec le modèle 2D ont été publiés dans [Pouliquen 05a].

4.5.2 Simulations et résultats

Nous avons réalisé trois simulations pour étudier le comportement de nos coussinets déformables modélisés par des éléments finis :

- la flexion simple ou « mouvement libre » du doigt,
- la saisie d'un objet par les doigts entiers ou « prise en force »,
- la saisie d'un objet par les extrémités des doigts ou « prise de précision ».

4.5.2.1 Flexion d'un doigt

La première simulation consiste à contrôler en position le doigt. Nous imposons un angle θ aux phalanges du doigt. Au cours de la flexion du doigt, nous constatons que les coussinets phalangiens et la pulpe entrent en contact et se déforiment (Cf. Figure IV.104).

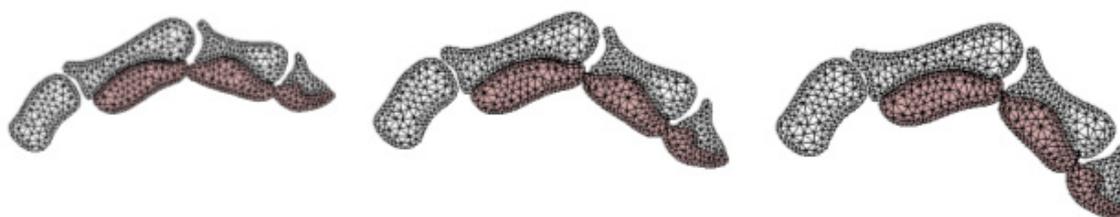


Figure IV.104 : Pilotage d'un doigt déformable et contraintes internes

4.5.2.2 Préhension d'un cylindre rigide dans le cas 2D sans frottement

La deuxième simulation permet de visualiser la préhension d'un cylindre rigide par deux doigts déformables : le pouce et l'index. Au fur et à mesure que les doigts se referment sur le cylindre, nous observons la déformation des coussinets de chacun des doigts. Dans ce cas de figure, nous obtenons une prise stable lorsque le cylindre est maintenu en position par quatre coussinets (Cf. Figure IV.105).

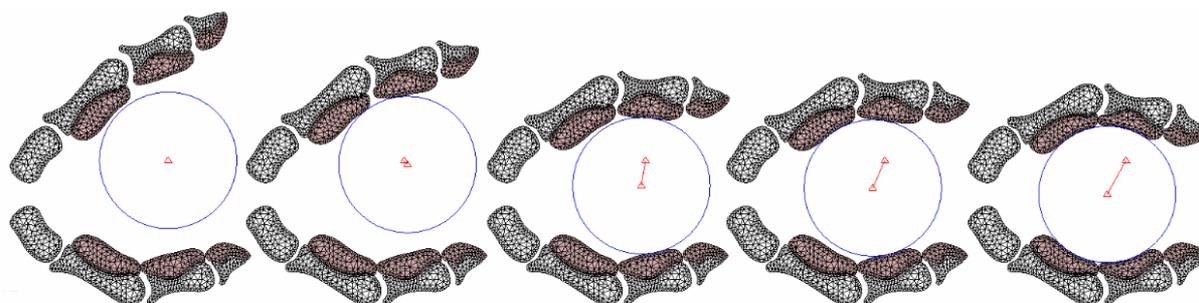


Figure IV.105 : Saisie d'un cylindre avec deux doigts

4.5.2.3 Préhension d'un cylindre rigide dans le cas 2D avec frottement

La dernière simulation est semblable à la précédente. Mais, au lieu de saisir le cylindre par la totalité des doigts, i.e. une prise en force, nous utilisons les extrémités des doigts pour une prise de précision (Cf. Figure IV.106). Cette simulation souligne l'importance des modèles de déformation et de frottement intégrés dans nos coussinets.

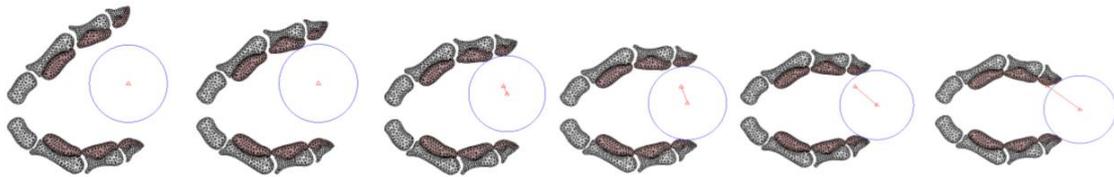


Figure IV.106 : Saisie d'un cylindre par le bout des doigts

En effet, s'il n'y a pas de frottement, le cylindre glisse et s'échappe des doigts. Il est donc nécessaire d'augmenter le coefficient de frottement pour bloquer le cylindre en position. Dans notre cas de figure, i.e. avec un comportement du matériau identique à celui du caoutchouc, nous avons obtenu une valeur seuil $\mu=0,4$ pour maintenir le cylindre rigide entre les extrémités des doigts.

4.5.2.4 Résultats des tests

Nous avons tout d'abord estimé les temps de calcul obtenus avec ce modèle. Nous obtenons la courbe de la Figure IV.107 :

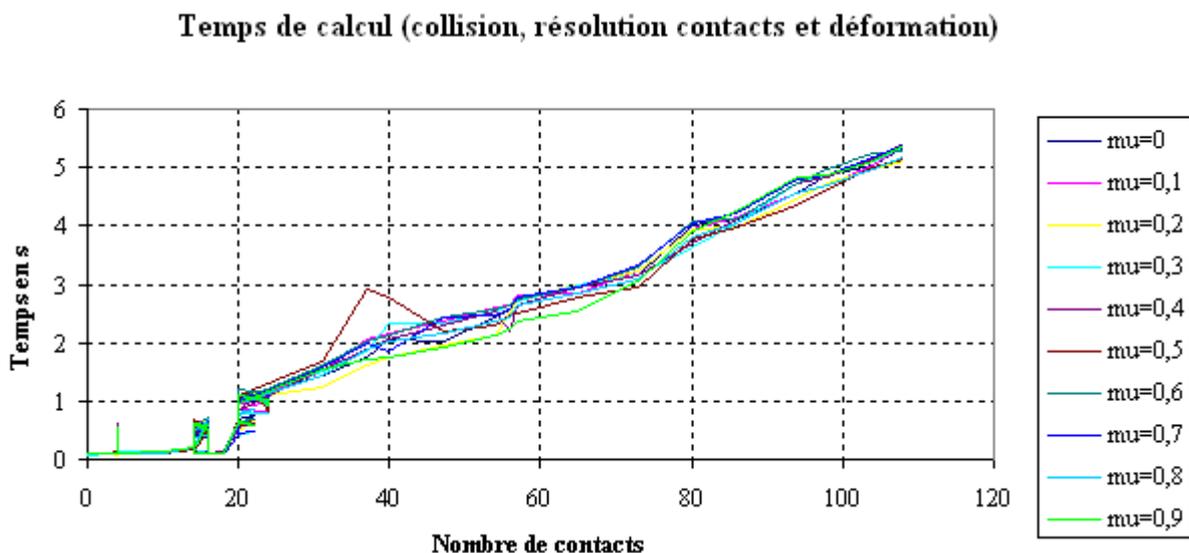


Figure IV.107 : Temps de calcul en fonction du coefficient de frottement

Nous observons que le temps de calcul varie quasi linéairement avec le coefficient de frottement. Les quelques distorsions de la courbe observées correspondent au phénomène de « broutage » décrit dans l'Annexe A.2. Pour ces valeurs de frottement et le nombre de contacts correspondant, le coefficient d'adhérence est plus grand que le coefficient de frottement. Notons que la valeur du coefficient de frottement a une influence moindre sur le temps d'exécution de l'algorithme que le nombre de contacts.

Nous constatons également que le temps de calcul est inférieur à 0,8 s dans le cas d'une

vingtaine de contacts et qu'il est au maximum égal à 5 s pour une centaine de contacts simultanés - frottants ou pas. Ces résultats ne nous permettent pas de respecter la contrainte de temps réel imposée pour nos applications.

Ensuite, nous avons souhaité vérifier le comportement de nos coussinets lorsqu'ils se déforment. Pour ce faire, nous avons modifié le module d'Young E de nos coussinets et nous avons simulé la prise d'un cylindre par les doigts (prise en force).

Nous constatons sur la Figure IV.108 que la force augmente linéairement avec le déplacement imposé quelle que soit la valeur de E :

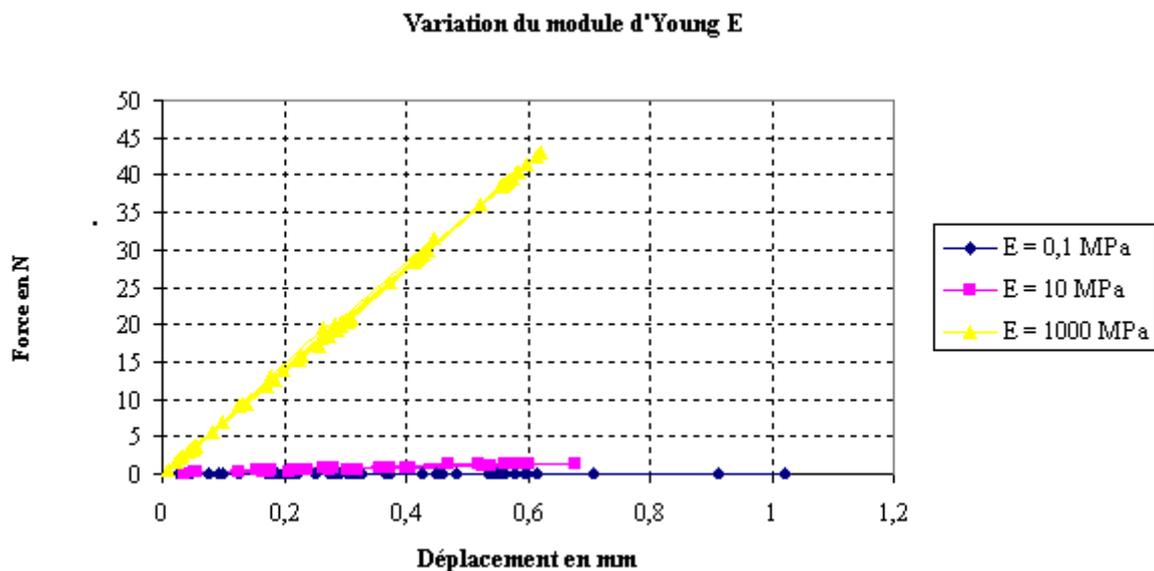


Figure IV.108 : Force en fonction de déplacement et du module d'Young

Ces résultats correspondent bien au comportement attendu pour une loi mécanique élastique linéaire. Notons le fait que plus le module d'Young est important, plus le matériau devient rigide.

Ainsi, le Tableau IV.7 confirme que plus le module d'Young est important, plus la force de contact est grande et plus le déplacement des nœuds est faible. Il montre également que l'on obtient un comportement conforme à un matériau élastique linéaire quelle que soit la valeur de E .

Module d'Young E	Force moyenne (N)	Force max (N)	Déplacement moyen (mm)	Déplacement max (mm)
0,01	0,001006846	0,00387354	0,514718455	1,89775
0,1	0,00994256	0,0386511	0,484887273	1,89813
1	0,097359347	0,377141	0,460912756	1,90169
10	0,806239444	2,54183	0,385705895	1,0175
100	3,777230268	14,2177	0,282014095	0,864565
1000	9,013631952	42,9858	0,236817147	0,622476
10000	13,49591321	67,5501	0,313969487	0,697051

Tableau IV.7 : Influence du module d'Young E sur les forces et les déplacements

Nous avons également étudié l'impact du module d'Young E sur les temps de calcul et par conséquent les performances de notre algorithme.

Nous constatons par le Tableau IV.8 que plus le coefficient E diminue, i.e. plus le matériau devient déformable, plus le nombre moyen de contacts augmente. Ce qui entraîne une légère augmentation du temps de calcul dû à l'augmentation de l'espace de contacts et à la construction d'un opérateur de Delassus de plus grande dimension (Cf. § 3.3).

Module d'Young E (Mpa)	Nombre moyen de contacts	Temps pour la collision (s)	Temps pour l'algorithme de Gauss Seidel (s)	Temps de calcul total (s)
0,01	153	0,016129032	0,729225806	1,635
0,1	153	0,016064516	0,815483871	1,723225806
1	147	0,01483871	0,787290323	1,633451613
10	87	0,015548387	0,27716129	0,632612903
100	45	0,004548387	0,232064516	0,421612903
1000	34	0,005483871	0,851225806	1,034677419
10000	31	0,006774194	0,849935484	1,030483871

Tableau IV.8 : Les temps d'exécution moyens en fonction du module d'Young E

4.5.3 Conclusion dans le cas 2D

Ces simulations ont permis de tester et valider les différents modèles de collision, de résolution de contact et de déformation décrits dans ce chapitre afin de prototyper la bibliothèque finale en langage C/C++ appelée **Collision**.

Les résultats obtenus dans le cadre des simulations du mouvement libre du doigt et de la préhension d'un objet rigide ont mis en évidence un comportement des coussinets déformables conforme aux attentes.

Ainsi, les performances en temps de calcul décrites dans ce paragraphe ne sont pas compatibles avec la construction d'applications en temps réel malgré la simplicité de programmation matricielle sous Matlab © qui facilite les calculs pour la méthode des éléments finis.

Mais, notre objectif est d'obtenir une bibliothèque en langage C/C++ et non un code en langage « interprété » comme Matlab ©.

Nous avons constaté préalablement que le code de la bibliothèque **Collision** pouvait être optimisé pour la détection de collisions et la résolution des contacts (Cf. § 2.4.5 et § 3.3.6) afin d'améliorer les temps d'exécution. Dans le paragraphe qui suit, nous vérifions plus particulièrement le comportement de nos objets modélisés dans **Collision**.

4.6 Le modèle dans le cas 3D

Le modèle 3D reprend et généralise les équations utilisées dans le cas 2D pour la détection de collision et les modèles de contact et de déformation. Les algorithmes ont été codés en langage C/C++.

L'objectif est de simuler une main virtuelle avec des coussinets déformables. Nous modélisons les doigts rigides et des coussinets déformables par des primitives simples (cylindres et sphères).

4.6.1 Simulations et résultats

Nous proposons d'évaluer les performances de cet algorithme avec la même d'application que celle mise en oeuvre pour la méthode de traitement de contacts. Nous disposons donc d'une sphère élastique linéaire qui entre en contact avec un plan rigide fixe (Cf. Figure IV.96).

Pour cet exemple, nous avons intégré les caractéristiques mécaniques d'un matériau de type caoutchouc avec $E = 0,01$ GPa et $\nu = 0,49$. Cet exemple simple s'apparente à la déformation de la pulpe d'un doigt au contact d'un plan rigide.

La simulation consiste à représenter un coussinet déformable en compression. Sur la Figure IV.109, nous constatons que la force augmente quasi linéairement avec le déplacement imposé : ce qui correspond bien au comportement attendu pour une loi mécanique élastique linéaire. Une légère convexité de la courbe vers le bas peut être décelée : elle correspond à la relaxation du système.

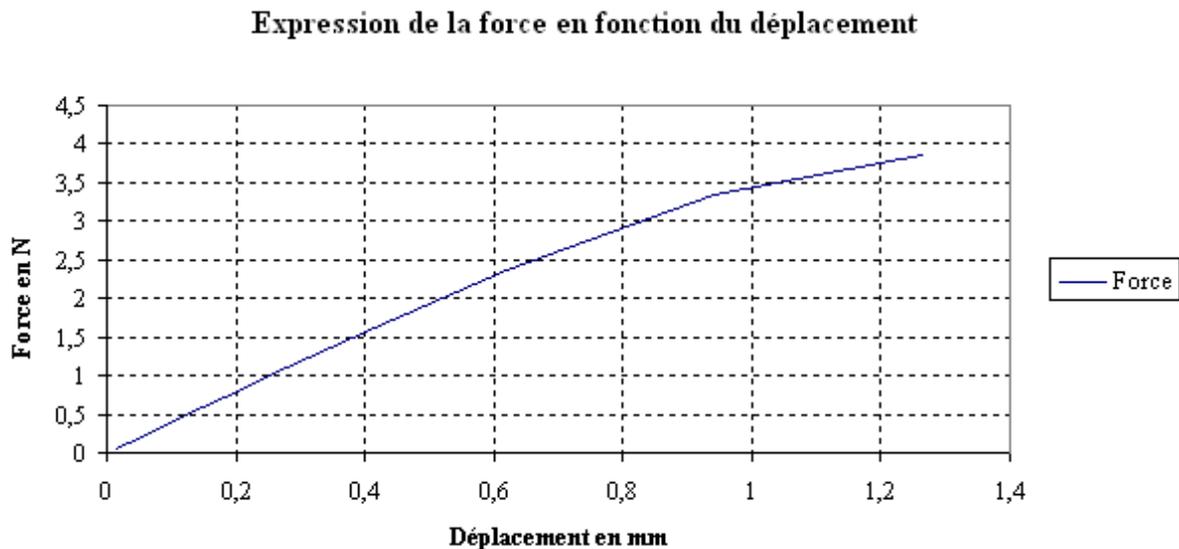


Figure IV.109 : Expérience simulée de compression d'une sphère élastique linéaire sur un plan

Nous remarquons également que l'origine de la courbe est légèrement décalée par rapport au zéro. Ceci est dû au fait que le point de collision est recherché lors de la première itération alors que les forces ne sont calculées qu'à partir de la deuxième.

4.6.2 Conclusion dans le cas 3D

Suite aux tests réalisés sous Matlab © et à leurs résultats prometteurs, nous avons programmé le modèle éléments finis en 3D en langage C/C++. Dans ce cas de figure, le comportement des coussinets obtenus est conforme aux attentes : ce qui est cohérent avec les résultats obtenus sous Matlab ©. Le modèle de coussinets est donc réaliste du point de vue physique. Mais, nous n'avons programmé et testé que des modèles unitaires comme par exemple le contact d'une sphère sur un plan (i.e. la pulpe d'un doigt qui se déforme) ou encore le contact de deux sphères de chaque côté d'un même plan (i.e. la préhension d'un objet simple).

Afin d'évaluer les performances réelles de notre modèle, il est essentiel de parvenir à modéliser une main complète avec des pulpes déformables en 3D. Nous pourrions alors estimer le temps d'exécution pour notre modèle éléments finis.

4.7 Conclusion pour le modèle déformable développé

Nous avons fait le choix de mettre en oeuvre la méthode des éléments finis pour les doigts déformables. Cette méthode basée sur un modèle continu de la matière permet :

- D'être indépendante de la topologie d'un maillage ou d'une forme particulière d'objet,

- D'envisager des extensions vers un modèle viscoélastique par exemple grâce à une mise en œuvre flexible.

Les simulations obtenues avec cette méthode sont par conséquent plus réalistes du point de vue physique que les systèmes de type masses/ressorts.

Nous modélisons donc les coussinets phalangiens et la pulpe des doigts par des sphères discrétisées par des éléments tétraédriques. Nous intégrons ensuite une loi de comportement élastique linéaire pour le matériau des coussinets déformables et nous nous plaçons dans le cas des petites déformations. Ces hypothèses ont été adoptées pour nos simulations avec l'objectif d'optimiser le temps de calcul afin de répondre aux spécifications temps réel de nos applications.

Dans un premier temps, nous avons montré que les tests réalisés sous Matlab © dans le cas de la préhension 2D sont conformes aux attentes. En effet, en se référant aux mesures expérimentales réalisées sur des doigts humains pour définir les modèles de pulpe existants, nous constatons que la force est inférieure à 5 N pour un déplacement maximal de 2,5 mm lors des expériences de compression de la pulpe [Dandekar 97] [Pawluk 97] [Serina 98] [Wu 04] (Cf. paragraphe 3 du Chapitre II). Pour être conforme à cette réalité, nous devons donc choisir dans notre cas de figure, une valeur de module d'Young E comprise entre 0,001 MPa et 20 MPa pour les coussinets. Au regard des résultats obtenus au cours de nos tests, il s'avère que le module d'Young du caoutchouc de 10MPa utilisé satisfait à cette contrainte et permet d'obtenir les temps de calcul les plus intéressants. De plus, nous avons observé que le comportement des coussinets en contact est élastique linéaire et respecte bien la loi de comportement intégrée. Mais, les résultats obtenus montrent que les temps de calcul ne sont pas assez rapides pour respecter la contrainte de temps réel car ils requièrent en moyenne une minute pour les simulations.

Dans un second temps, nous avons codé notre modèle en langage C/C++. Comme pour la détection de collision et pour le traitement des contacts, la méthode des éléments finis s'est révélée être coûteuse en temps de calcul pour nos applications de préhension d'objets. Ceci est dû à la taille des matrices de compliance, et par conséquent des matrices de raideur, qui dépendent du nombre de contacts simultanés. Il apparaît donc nécessaire de trouver un compromis entre le nombre de contacts maximal pour une itération et la qualité des rendus visuel et physique des simulations de préhension d'objets pour respecter la contrainte de temps réel.

5 Conclusion

Dans ce chapitre nous avons présenté un modèle de préhension pour des applications en Réalité Virtuelle. Ce modèle est constitué de :

- une méthode de détection de collision qui met en œuvre des hiérarchies de sphères pour englober les objets de la scène. La détection d'interférences se fait au niveau des triangles du maillage surfacique. Elle retourne la liste des points en contact avec les distances de pénétration, les normales et les tangentes ;
- un modèle de contact qui utilise une méthode issue de la mécanique des milieux granulaires. La mise en place d'un algorithme de type Gauss Seidel qui permet le calcul des forces de contact à partir des informations obtenues par la détection de collision dans le cas de contacts frottants simultanés ;
- un modèle de déformation qui utilise la méthode des éléments finis pour modéliser les coussinets déformables des doigts. Nous mettons en œuvre un modèle élastique

linéaire isotrope pour calculer le déplacement des nœuds à partir des forces de réaction calculées. Ainsi les coussinets se déforment en résolvant un problème quasi-statique.

Ces trois modèles sont intégrés dans la bibliothèque **Collision** qui permet de simuler des interactions physiques entre objets virtuels rigides et/ou déformables (Cf. Annexe A.5). Ainsi, au sein de cette même bibliothèque, nous pouvons répondre à un grand nombre de requêtes différentes comme par exemple le calcul de distances, la détection de collision, le calcul des forces de contacts entre objets convexes ou concaves, ou encore le calcul des déformations dues à un contact.

Une version de ces programmes existe également sous Matlab © dans le cas 2D. Celle-ci a été utilisée pour tester et valider les modèles de la bibliothèque **Collision** du point de vue du réalisme physique mais pas de celui des performances en temps d'exécution.

En premier lieu, nous avons constaté que le temps de calcul pour la détection de collision augmente rapidement avec la complexité des objets. Ceci est dû à l'utilisation de sphères englobantes qui conduisent à exécuter de nombreux tests de collision lors du parcours des arbres binaires représentant les objets. Pour optimiser le temps de calcul de cette étape, il faudrait mettre en œuvre des boîtes englobantes de type AABB en plus des sphères englobantes. Nous avons déjà prévu la structure de cette nouvelle classe dans notre code mais nous n'avons pas encore intégré les méthodes de construction et de mise à jour des hiérarchies dans ce cas de figure.

Ensuite, les tests ont mis en évidence la faible convergence de notre algorithme de traitement de contacts. Les temps de calcul pour la construction de l'espace des contacts et plus particulièrement pour le calcul des forces de contact augmentent quasi exponentiellement avec le nombre de contacts frottants. Or dans le cas de simulations de préhension d'objets, plusieurs contacts frottants existent simultanément entre la main et l'objet saisi. En fonction du type de prise – prise en force ou prise de précision - il est important de prêter une attention toute particulière à l'obtention de l'opérateur de Delassus. En effet, la taille des matrices de compliance dépend du nombre de contacts et de leur localisation : si les contacts sont répartis sur les doigts, comme c'est le cas pour la simulation décrite dans le paragraphe 4.5.2.2, la multiplication matricielle pose un problème. Dans le cadre de nos applications de préhension d'objets, il est donc nécessaire de mettre en place des stratégies pour réduire le temps de calcul de l'opérateur de Delassus et respecter la contrainte de temps réel.

Enfin, les tests réalisés ont mis en évidence le réalisme physique de notre modèle de déformation. En effet, nous avons intégré les caractéristiques mécaniques d'un matériau élastique linéaire isotrope dans nos coussinets déformables éléments finis et nous avons observé le comportement linéaire de la force en fonction du déplacement : ce qui est conforme à nos attentes. De plus, les valeurs obtenues pour les forces de contact en fonction du déplacement des nœuds des maillages sont cohérentes avec celles données par les modèles des pulpes des doigts existants (Cf. paragraphe 3 du Chapitre II).

En conclusion, les tests réalisés ont mis en évidence le non respect de la contrainte de temps réel pour chacun des modules (détection de collision, traitement des contacts et modèle de déformation). Il serait donc nécessaire de revoir la programmation de certaines parties afin d'optimiser les temps d'exécution observés lors de nos tests. Il serait également possible de faire appel à des processeurs plus performants qui possèdent plus de ressources que celui actuellement utilisé pour réaliser ces calculs complexes. Mais, il est important de noter que les résultats obtenus avec notre programme codé sous Matlab © et avec notre bibliothèque **Collision** sont physiquement réalistes.

CHAPITRE V : Pour une meilleure intégration de l'Homme à la conception d'équipements de travail

Dans les chapitres précédents, nous avons décrit le modèle de main que nous avons développé pour interagir avec un environnement virtuel. Pour rappel, cette main virtuelle est constituée :

- d'un modèle squelettique rigide qui permet de déplacer la main et d'en animer les doigts dans l'environnement virtuel,
- de coussinets déformables qui permettent d'assurer la stabilité des prises manuelles et d'améliorer les réalismes physique et visuel lors des tâches de préhension.

1 Introduction

Dans ce chapitre, nous présentons les apports d'un tel modèle dans le cadre de l'étude d'humains en situation de travail et de la prévention des risques professionnels.

Afin de mettre en évidence l'intérêt de notre modèle, nous avons réalisé plusieurs applications pour simuler les interactions Homme-machine en environnement virtuel. Pour ce faire, nous avons utilisé différentes interfaces comme la capture de mouvements ou un bras à retour d'effort. Ces démonstrateurs ont été définis conjointement avec l'INRS (Institut National de Recherche et de Sécurité) avec les objectifs suivants :

- simuler de façon réaliste les interactions avec un environnement virtuel : l'utilisateur doit être capable de toucher, saisir et manipuler des objets ou des outils virtuels de manière stable et en temps réel ;
- modéliser de façon réaliste un équipement de travail et une situation de travail ;
- intégrer la prévention des risques au plus tôt dans le cycle de conception en simulant des opérations sur un équipement de travail virtuel.

Le premier objectif est celui que nous avons cherché à atteindre pendant la thèse. Les résultats sont présentés dans ce manuscrit. Le modèle de main est un élément majeur car le degré du risque d'accidents de travail dépend principalement des interactions entre l'opérateur et son environnement.

En outre, les deux autres objectifs font l'objet de recherches au sein de l'INRS depuis de nombreuses années. Un modèle de situation de travail a notamment été développé par Hasan [Hasan 02]. Il permet de prendre en compte l'ensemble des données d'une situation de travail et leurs interdépendances, et ce plus particulièrement au travers de la notion de risque. En couplant ce modèle à un environnement de travail virtuel, il est possible d'afficher en temps-réel dans la scène un indice de risque de la situation simulée [Marsot 05]. Cela permet d'introduire la prévention des risques au plus tôt dans le cycle de conception en utilisant les techniques de la Réalité Virtuelle.

Dans un premier temps, nous présenterons l'équipement de travail avec lequel l'opérateur interagit en utilisant les mains virtuelles. Puis, nous décrirons la plateforme de Réalité Virtuelle dans laquelle évolue l'utilisateur réel ainsi que l'environnement virtuel dans lequel se déplace l'opérateur virtuel. Ensuite, nous exposerons comment le modèle de situation de travail permet de mieux intégrer l'utilisateur dans la simulation. Nous présenterons également

l'indice de risque qui facilite l'estimation du risque pour une configuration de travail donnée. Enfin, le dernier paragraphe illustrera les différentes applications développées en partenariat avec l'INRS afin d'intégrer et valider le modèle de main.

2 La simulation d'une presse plieuse

Pour tester le modèle de main et pour explorer le champ des apports possibles de la Réalité Virtuelle pour la prévention des risques d'accidents de travail, nous avons retenu le cas exemple de la presse plieuse. En effet, cette machine-outil nécessite l'intervention d'un opérateur pour plier des tôles métalliques. Celui-ci utilise ses mains pour maintenir ou manipuler la tôle lors de ces opérations. C'est également un équipement industriel pour lequel les accidents professionnels sont gravement mutilants. De plus, l'expérience montre qu'il est relativement difficile de sécuriser la presse plieuse de manière générale du fait de la diversité des cas d'utilisation (taille de la tôle, proximité des plis, nombre d'opérateurs...).

Dans les paragraphes suivants, nous allons décrire le principe de fonctionnement de cette machine-outil ainsi que les simulateurs qui ont été développés par l'INRS.

2.1 Principe de fonctionnement

La presse plieuse est une machine qui sert à plier des métaux en feuille en les pinçant entre deux outils : la matrice et le poinçon. Sur la presse de notre cas exemple, la matrice est fixe par rapport au bâti, et le poinçon est fixé sur un tablier supérieur mobile. La montée et la descente de ce tablier mobile sont commandées par une pression soit sur une pédale, soit sur des boutons poussoirs d'une commande bimanuelle (Cf. Figure V.110).

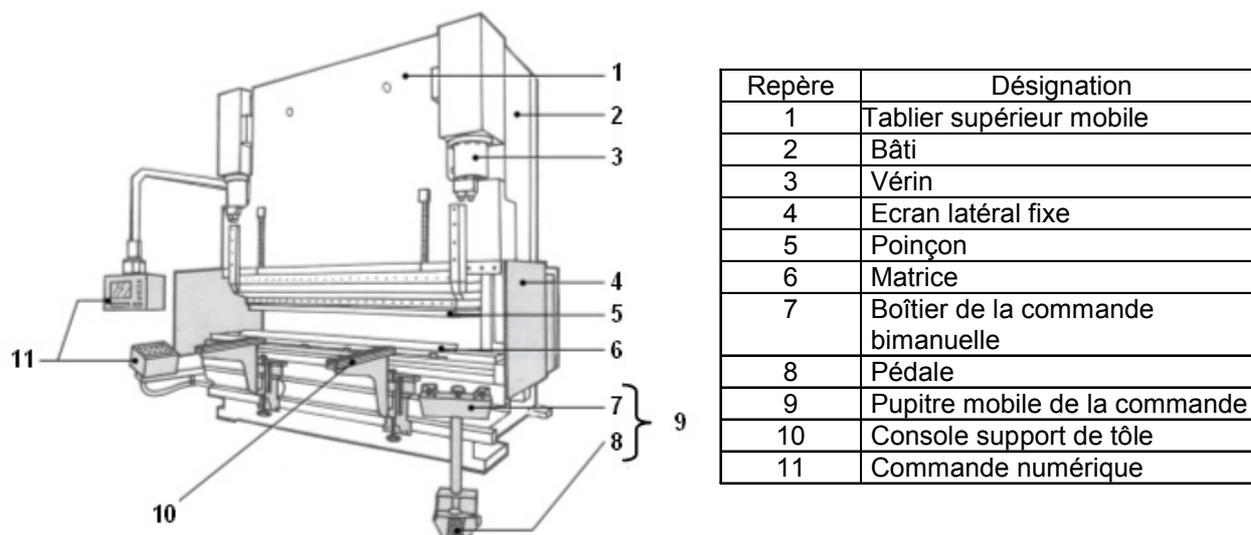


Figure V.110 : Vue schématique d'une presse plieuse

La presse est programmée pour fonctionner suivant des cycles avec un nombre de plis et des angles de pliage prédéfinis. Des supports de tôle aident l'opérateur à maintenir la tôle au niveau de la matrice. Des butées motorisées réglables permettent de plier la tôle à une distance précise. Après chaque pli, celles-ci se positionnent pour que l'opérateur n'ait plus qu'à placer la tôle contre, et commander le pli suivant. Lorsque le cycle de pliage est terminé, le poinçon remonte complètement : l'opérateur peut alors enlever la tôle sans gêne.

Dans le paragraphe suivant, nous présentons le résultat de la modélisation de la presse.

2.2 La presse plieuse virtuelle

Une pré-faisabilité de simulation d'une presse plieuse a été réalisée en novembre 2001 par le CEA List sur spécifications et pour le compte de l'INRS. Bien que la simulation de la presse plieuse actuelle repose sur cette première maquette, elle a été complètement reconçue pour les besoins de la plateforme de l'INRS [Pinczon 04].

Nous simulons une presse plieuse complète en modélisant toute la partie opérative (poinçon, supports, tôle métallique, etc.) nécessaire à la réalisation des plis de tôles métalliques. Les dispositifs de sécurité sont également modélisés comme par exemple les faisceaux laser, le rideau lumineux ou encore les protections latérales. La Figure V.111 illustre une configuration possible de la presse plieuse :

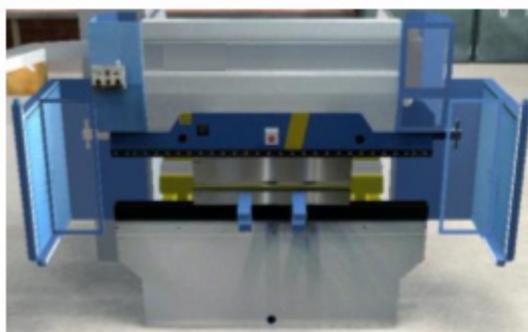


Figure V.111 : La presse virtuelle

Nous avons également modélisé l'atelier afin d'améliorer le sentiment d'immersion de l'opérateur quand il se déplace dans cet environnement virtuel (Cf. Figure V.112).



Figure V.112 : L'atelier virtuel

Nous avons donc représenté une presse plieuse et un atelier de travail dans leur totalité. Ainsi, l'opérateur réel se déplacera sur la plateforme de Réalité Virtuelle alors que ses mains virtuelles interagiront avec la presse virtuelle et évolueront dans cet atelier virtuel.

Nous allons maintenant décrire la plateforme de Réalité Virtuelle de l'INRS ainsi que le matériel utilisé pour interagir avec l'environnement virtuel et simuler les opérations de pliage de tôles.

3 La plateforme de Réalité Virtuelle de l'INRS

Le premier démonstrateur ayant permis de valider la faisabilité technique du simulateur de presse en 2001, le projet **EVICS** (Environnement **VI**rtuel pour la **C**onception de **S**ystèmes sûrs) a été lancé. Le but de ce projet est d'obtenir une méthodologie de l'utilisation de la Réalité Virtuelle pour la prévention des risques professionnels, ainsi que des cas exemples servant de démonstrateurs pour les entreprises [Ciccotelli 02a].

C'est dans le cadre de ce projet que le sujet de la présente thèse a été défini avec l'objectif de développer des mains virtuelles pour mieux intégrer l'opérateur dans le simulateur et lui permettre un nouveau mode d'interaction avec l'environnement virtuel.

Pour mener à bien ce projet, l'INRS s'est doté d'une plateforme multi-modale de Réalité Virtuelle. Cette dernière comprend :

- un calculateur physique temps réel qui utilise le logiciel **VORTEX**™ (PC bi-processeur), fourni avec une licence pour l'API Vortex [wCMLabs], pour la simulation physique des objets virtuels. C'est une librairie en langage C regroupant un ensemble de fonctions permettant de simuler un comportement physique d'un objet dans un environnement virtuel en utilisant un moteur de collision et un moteur dynamique. Nous l'utilisons pour modéliser la presse plieuse, l'atelier et les autres objets de la scène. Ceci permet de donner une consistance physique aux objets. Il est alors possible de détecter les collisions entre la presse et la tôle virtuelles lors de l'opération de pliage ou d'imposer des déplacements comme par exemple pour le poinçon ou les butées motorisées.
- une station d'affichage 3D temps réel qui utilise le logiciel **VIRTOOLS**™ [wVirtools] pour développer des applications multimédias interactives de façon rapide. Son utilisation sur la plateforme de réalité virtuelle de l'INRS permet de décharger le PC physique de l'aspect graphique de la simulation comme la visualisation ou le rendu des objets virtuels.
- un système de projection stéréoscopique passif composé d'un écran rigide de 2.5m de base par 2m de haut et de deux projecteurs LCD fabriqués par Barco [wBarco]. Dans ce cas, l'affichage se fait en retro-projection pour éviter toute ombre sur l'écran. C'est ce mode d'affichage que nous utilisons pour notre simulation.
- un bras à retour d'effort **Virtuose 6D** à six degrés de liberté, développé par le CEA, et commercialisé par la société **Haption** [wHaption] (Cf. Figure V.113). Il permet de contrôler un objet virtuel en position et en orientation. Pour commander ce bras, on utilise la **Virtuose API**. C'est une librairie qui régit d'une part la communication entre le simulateur et l'interface haptique **Virtuose 6D** et qui effectue d'autre part les calculs d'asservissement de haut niveau par le contrôleur. Ainsi, les fonctions existantes permettent l'initialisation du bras, le couplage aux objets virtuels et elles renvoient la position et l'orientation du bras afin d'effectuer l'asservissement des objets virtuels par une consigne en position et en orientation. Ce bras haptique a un retour maximal d'effort en translation de 35 N avec une résolution de 0,1 N et en rotation de 3,1 Nm. L'espace de travail est un cube de 450 mm de côté.

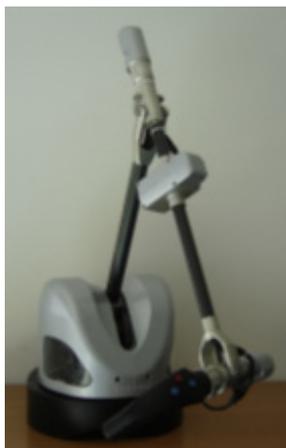


Figure V.113 : Le bras à retour d'effort Virtuose 6D

Dans notre cas de figure, une plaque de Plexiglas a été fixée sur le manche du Virtuose 6D afin que l'opérateur puisse manier la tôle virtuelle comme il le fait en situation réelle (Cf. Figure V.114). Les efforts qu'il exerce sur la plaque ou la presse virtuelle sont mesurés par l'interface haptique et utilisés pour la simulation.



Figure V.114 : Simulation dynamique d'un pliage de tôle avec le bras à retour d'effort

- Un système de capture de mouvements fourni par la société Vicon [wVicon]. Il s'agit d'un système optique constitué de caméras et de marqueurs, dont la position est calculée en temps-réel. Afin de reconnaître facilement les marqueurs dans l'image filmée par chaque caméra, un filtre est utilisé pour ne laisser passer qu'une certaine bande de fréquences proche de l'infrarouge car celle-ci est bien réfléchiée par les marqueurs. Chaque caméra est complétée d'un émetteur dans cette même bande de fréquence. Ainsi, les seuls objets qui apparaissent sur l'image acquise par chaque caméra sont les marqueurs présents dans la scène. Avant toute utilisation, un calibrage est nécessaire pour déterminer la position des caméras les unes par rapport aux autres. Ces informations sont importantes pour déterminer les positions des marqueurs. Ces dernières peuvent être reconstruites en trois dimensions si le marqueur est visible par deux caméras au minimum.

La plateforme de l'INRS est dotée de quatre caméras. Elles sont disposées en face de l'utilisateur afin de suivre les mouvements de ses deux mains. Pour déterminer l'orientation d'une main, nous utilisons quatre marqueurs – qui forment des corps rigides ou *rigidbody* (Cf. Figure V.115) [Pinczon 04]. Cela nous permet également de distinguer la main droite de la main gauche et par conséquent de résoudre les problèmes d'occultation et de superposition inhérents aux opérations bimanuelles.

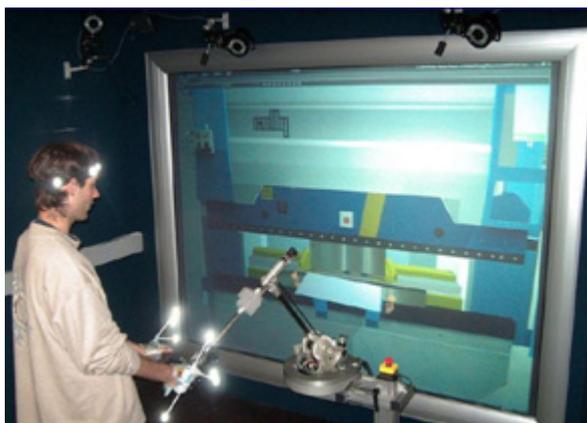


Figure V.115 : Les marqueurs utilisés pour le suivi des mouvements de l'opérateur

- Un système de son stéréo pour créer l'environnement sonore.
- Un gant de données CyberGlove avec 22 capteurs vendu par Immersion Corporation [wImmersion]. Il possède trois capteurs par doigt pour la flexion, trois capteurs pour l'abduction des doigts, un capteur pour l'arc de la paume et trois capteurs pour le pouce (flexion et abduction). Les capteurs ont une résolution de $0,5^\circ$. Nous utilisons ce gant de données pour suivre les mouvements de flexion et d'abduction des doigts de l'opérateur. Il permet de mesurer les angles des phalanges en temps réel. En utilisant les contrôleurs présentés dans le Chapitre III, notre main virtuelle peut reproduire les mouvements de la main de l'opérateur interactivement comme le montre la Figure V.116 et la Figure V.117.



Figure V.116 : Utilisation du CyberGlove pour animer la main virtuelle

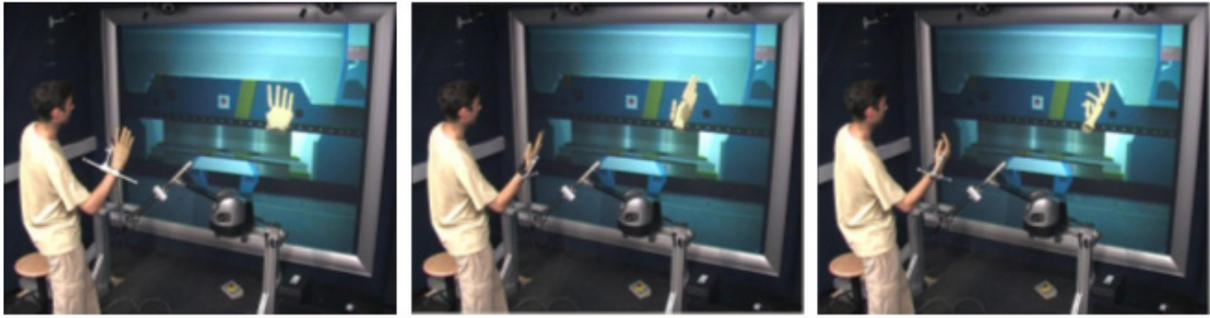


Figure V.117 : Utilisation du CyberGlove dans l'environnement de la presse plieuse

En résumé, dans notre cas de figure, nous utilisons :

- le système de capture de mouvements (i.e. les traqueurs optiques et le gant de données) qui permet de situer l'opérateur dans la scène, de suivre les déplacements de ses mains dans la zone de démonstration et également de suivre les mouvements de ses doigts ;
- le bras à retour d'effort qui permet d'interagir avec la tôle virtuelle ;
- et la pédale qui permet de lancer le scénario des pliages (Cf. § 2.1).

La Figure V.118 ci-dessous présente la disposition de ces périphériques sur la plateforme de Réalité Virtuelle :

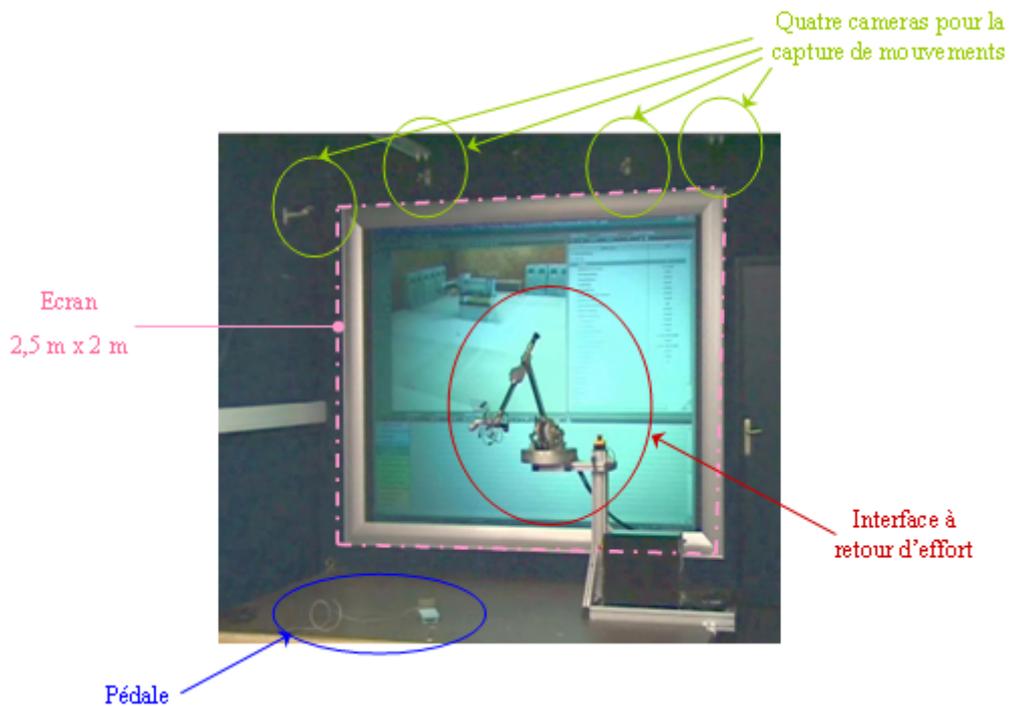


Figure V.118 : La plateforme de RV de l'INRS

Ainsi, la simulation de la presse plieuse communique :

- avec le système de capture de mouvements pour connaître les déplacements et les configurations des mains de l'opérateur afin de les appliquer aux mains virtuelles en utilisant les contrôleurs décrits dans le Chapitre III ;
- avec le bras à retour d'efforts afin de connaître les actions effectuées par l'utilisateur pour lui envoyer les forces de réaction appliquées par l'environnement sur la tôle en utilisant la Virtuose API ;
- avec le PC graphique en lui transmettant la position de tous les objets pour leur

affichage dans la scène virtuelle : c'est la simulation graphique ;

- avec le PC physique en lui communiquant toutes les données physiques telles que la configuration, la vitesse ou encore l'accélération pour connaître les déplacements des objets virtuels et simuler de manière réaliste leur comportement dans la scène virtuelle : c'est la simulation physique.

Les échanges entre les PC et les périphériques se font à l'aide d'une transmission basée sur du VRPN [Pinczon 04]. L'ensemble nous permet de simuler la préhension de la tôle et son pliage par la presse virtuelle. Nous obtenons alors l'architecture globale de la plateforme suivante (Cf. Figure V.119) :

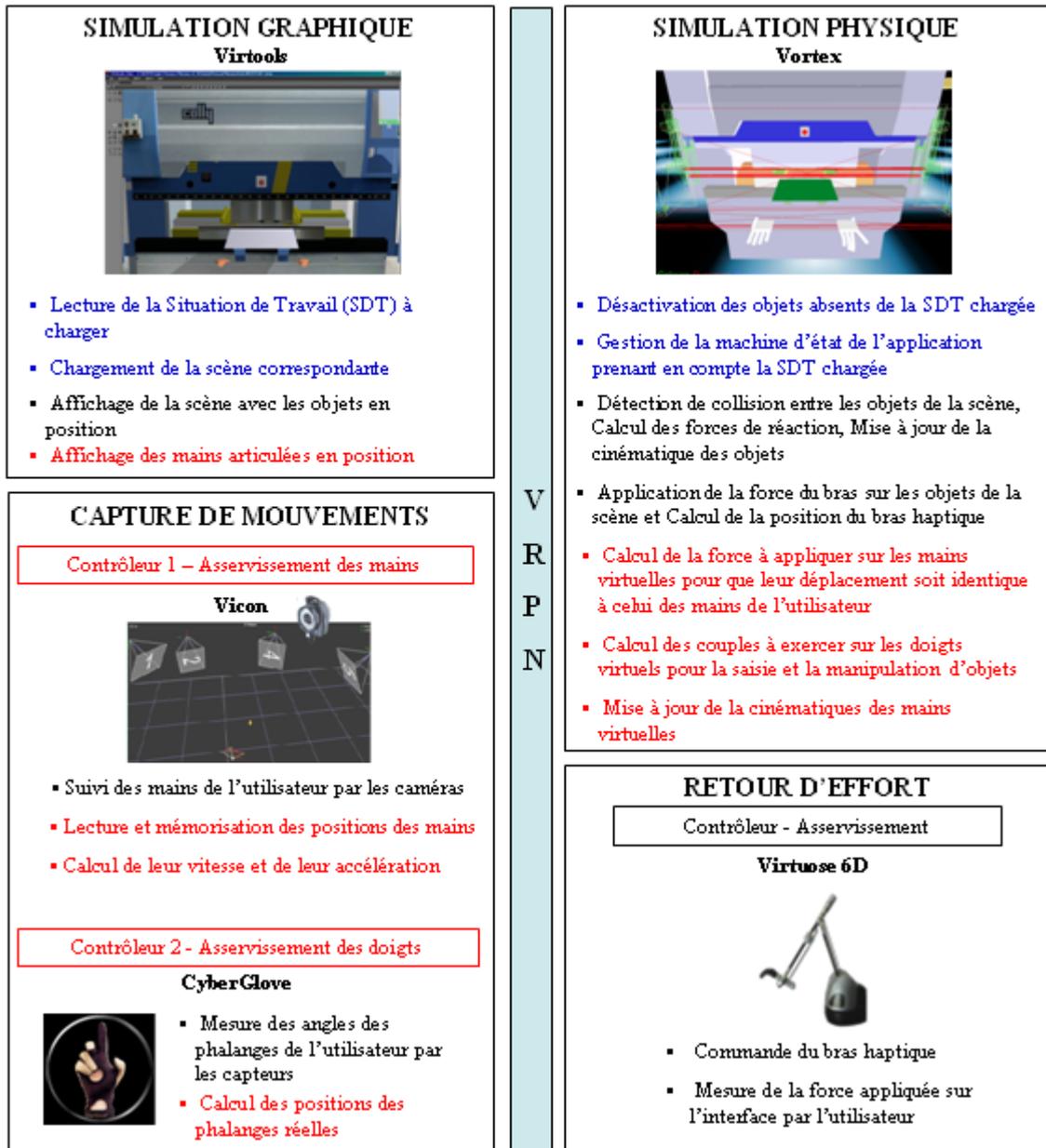


Figure V.119 : Architecture de la simulation de la presse plieuse

En noir : les fonctionnalités des logiciels et périphériques utilisés,

En bleu : les développements réalisés pour le simulateur de presse plieuses par l'INRS,

En rouge : les développements que nous avons réalisés pour les mains virtuelles.

Après avoir détaillé les logiciels et les matériels utilisés pour la simulation de pliage de tôle,

nous présentons les outils développés par l'INRS pour intégrer la prévention des risques d'accidents dans la simulation.

Les recherches pour la prévention des risques professionnels ont mis en évidence le fait que l'interaction quotidienne Homme-machine est loin d'être un processus sûr et sans erreur. En effet, le risque d'accidents est toujours présent. Et cela s'explique notamment par la différence entre les situations de travail définies par le concepteur et les conditions de travail réelles. Afin de prendre en compte les questions de sécurité très en amont dans le cycle de conception, l'INRS a choisi de mettre en œuvre les techniques de la Réalité Virtuelle.

Dans le paragraphe suivant, nous nous intéressons plus particulièrement aux outils développés pour estimer le degré de risques d'une situation de travail. Après une présentation brève des précédents travaux mettant en œuvre les techniques de la Réalité Virtuelle pour la prise en compte des facteurs humains, nous décrivons le modèle de situation de travail et l'indice de risque dynamique intégrés dans le simulateur de la presse plieuse.

4 L'intégration de la prévention des risques

Concevoir des systèmes dits «sûrs» est une condition nécessaire mais insuffisante pour éviter les accidents. En effet, on constate que malgré toutes les précautions prises – normes de sécurité et protection de l'utilisateur - il existe toujours des situations de travail susceptibles de devenir critiques, voire dangereuses. Pour diminuer ces risques, de nouveaux modèles de représentation de situations de travail et d'avatars de l'opérateur peuvent être mis en œuvre. Ainsi, l'INRS propose d'amener le concepteur à mieux appréhender ces situations. Il s'agit pour cela de le doter de moyens de prototypage et d'évaluation d'équipements semi-virtuels [Ciccotelli 02a]. Cela consiste à étudier et à développer des principes permettant d'utiliser la Réalité Virtuelle pour vérifier que le système conçu permettra d'assurer sa fonction, d'être réglé et d'être entretenu sans mettre en danger les opérateurs qui l'utilisent.

4.1 Utilisation de la réalité virtuelle pour la prévention des risques professionnels

L'objectif de tout concepteur d'équipements de travail est d'obtenir le bon fonctionnement du système conçu tout en respectant les contraintes imposées par le cahier des charges fonctionnel. Au cours du processus de conception, des décisions sont prises sans connaissance du système final ce qui génère des facteurs inconnus susceptibles de conduire à des situations à risques lors du fonctionnement ou lors de la maintenance de ce système. Ainsi, des déficiences dans le processus de conception ont déjà causé des pannes et des accidents qui auraient pu être évités en adoptant des approches de prévention des risques dans le cycle de conception. Il est donc essentiel que la sécurité d'un système soit systématiquement évaluée à chaque étape du cycle de conception : cette notion est appelée « prévention intégrée »⁶. Elle consiste à appliquer au plus tôt des principes de conception sûre à un futur équipement de travail [Ciccotelli 97].

Différents outils et méthodes ont ainsi été développés pour traiter des questions de sécurité dans les phases de conception et de production des produits [Honk Kong 99] [FAA 00] [Lund 04]. Par exemple il est possible de réaliser des analyses de tâches qui sont des études de cas effectuées

⁶ Les principes de conception sûre ont été introduits dans la loi du 6 décembre 1976 et les décrets du 15 juillet 1980 (INRS, 1983) et inscrits au niveau européen par la directive codifiée 98/37/CE.

par le biais d'observations et de questions. Il est également possible de mettre en œuvre des phases de formation et/ou d'entraînement professionnels pour appréhender un nouvel équipement de travail ou des situations dangereuses.

Dans les années 90, le rapide développement des ordinateurs et des logiciels a permis d'étudier des systèmes dans des environnements virtuels. L'utilisation de ces nouvelles technologies était relativement rare en raison du manque de connaissances et de l'absence de méthodes pour les appliquer à l'analyse de risque.

Les premières expériences d'analyse du degré de risque en environnement virtuel ayant obtenu des résultats probants ont été réalisées par Dotson et al. à l'Institut National pour la sécurité occupationnelle et la Santé (NIOSH⁷) [Dotson 97] et par Laitinen à l'Institut Finnois de la santé au travail (FIOH⁸) [Laitinen 97]. Elles concernaient l'étude de la manutention et du travail en hauteur. Puis, Bell et Scott Fogler [Bell 00] ont réalisé plusieurs simulations pour l'analyse des risques chimiques. Ils ont également mis en œuvre des accidents virtuels pour que leurs étudiants puissent expérimenter des situations dangereuses sans se mettre en danger [Bell 01]. Cette expérience leur a permis de mieux intégrer les consignes de sécurité. De la même manière que Bell et Scott Fogler, Nasios a montré dans sa thèse que la Réalité Virtuelle permettait d'éduquer et d'entraîner le personnel d'une usine chimique grâce à différents scénarii [Nasios 01]. Ses expériences ont démontré que le fait de confronter le personnel à des situations dangereuses en environnement virtuel permettait de les sensibiliser aux risques électriques et aux risques chimiques afin de diminuer le nombre d'accidents du travail dans l'usine chimique. Plus récemment, Määttä a mis en évidence dans sa thèse que les techniques de la Réalité Virtuelle ont amélioré l'analyse des risques dans différentes situations comme une usine de transformation de l'acier ou un convoyeur de charbon [Määttä 03]. Ces nouveaux moyens de visualisation et de modélisation mis en œuvre dans ses travaux de recherche ont permis d'appliquer des mesures de prévention sur ces installations industrielles.

Ainsi, la Réalité Virtuelle sensibilise les utilisateurs à la sécurité en leur permettant d'analyser leur environnement de travail et les tâches à effectuer en environnement virtuel ainsi que d'expérimenter des situations dangereuses virtuelles en toute sécurité [Weiss 98].

Différentes recherches en matière d'ergonomie ont également été menées tant au niveau de l'analyse des situations de travail [Sagot 03] [Shaikh 04] [Zülch 05] que de l'étude du comportement sensorimoteur et cognitif de l'Homme [Burkhardt 03].

Ces différents exemples mettent en évidence l'intérêt de la Réalité Virtuelle et de la formation pour l'amélioration du confort des postes de travail et la prévention des risques professionnels. Mais, ces nouvelles technologies connaissent des limitations et des effets secondaires. L'utilisation de systèmes de Réalité Virtuelle conduit à des problèmes physiques, physiologiques et/ou psychologiques. En effet, suivant les interfaces utilisées et les situations simulées – i.e. en fonction du niveau d'immersion - l'utilisateur est susceptible de souffrir de la maladie de l'espace, appelée *cybersickness* en anglais, de difficultés d'orientation ou encore de désordres oculaires [Costello 97] [Cobb 99] [Nichols 02]. La plupart de ces symptômes disparaîtront avec les progrès techniques des ordinateurs qui offriront une plus grande performance de calcul et donc autoriseront des simulations complexes en temps-réel. Les symptômes restants continueront de se manifester chez une partie de la population et ils devront être mieux étudiés afin de connaître leurs effets sur l'Homme à long terme.

Malgré les effets secondaires décrits précédemment, les technologies de la Réalité Virtuelle permettent de développer des simulations dynamiques qui peuvent être exploitées à des fins d'études ergonomiques des postes de travail et de prévention des risques professionnels.

⁷ NIOSH : National Institute for Occupational Safety and Health (Etats Unis)

⁸ FIOH : Finnish Institute of Occupational Health (Finlande)

C'est ainsi que l'Institut National de Recherche et de Sécurité (INRS) étudie les possibilités qu'offrent les techniques de la Réalité Virtuelle afin d'obtenir une meilleure intégration de la prévention des risques dans le cycle de conception des systèmes/produits [Ciccotelli 02b]. D'une part, l'objectif est de démontrer que si le concepteur possède des ressources virtuelles opérationnelles, il est alors capable de vérifier que les procédures, les instructions ou les autres opérations liées à l'équipement n'induisent pas de risques spécifiques ou ne dégradent pas le niveau de sécurité du système. D'autre part, ce même environnement virtuel peut être utilisé pour entraîner les opérateurs aux tâches d'assemblage, de maintenance ou autres tâches opérationnelles afin d'améliorer leurs capacités et de les sensibiliser à la sécurité [Ciccotelli 05].

Afin d'évaluer l'intérêt d'exploiter des environnements virtuels dans l'objectif de mieux prévenir les risques professionnels, l'INRS a développé une maquette numérique à l'échelle 1 d'une presse plieuse (Cf. § 2) [Marsot 04]. C'est une machine-outil qui permet de créer une large gamme de pièces allant des tôles de quelques centimètres pour lesquelles les doigts sont très proches du poinçon aux tôles de plusieurs mètres de long qui exigent l'intervention de deux opérateurs pour réaliser les pliages. Afin de reproduire dans les meilleures conditions la diversité des opérations de pliage, différentes situations de travail ont été créées. Elles modélisent les configurations de la presse plieuse avec différents dispositifs de sécurité ou avec différents modes opératoires. Et elles sont introduites dans la simulation de la presse plieuse sous la forme d'un scénario à charger. Nous présentons dans ce paragraphe le modèle de situation de travail utilisé.

4.2 Les modèles de situation de travail

Des études récentes sur la conception de systèmes complexes et leur modélisation ont permis de proposer un outil de données appelé « MOSTRA » qui est basé sur le modèle de situation de travail décrit par Guillevic [Guillevic 91]. Cette application permet au concepteur de prendre en compte simultanément les aspects techniques et humains lors de la conception d'un système ainsi que d'analyser leurs interdépendances à travers la notion de risque (voir [Hasan 02] pour plus de précisions). Un premier démonstrateur de ce modèle a été développé sous AccessTM. En le couplant avec le simulateur de la presse plieuse, nous pouvons :

- configurer l'environnement virtuel correspondant à une situation de travail requise : il est alors possible de choisir les dispositifs de protection ou encore les modes opératoires ;
- avoir accès à différents attributs et liens de la situation de travail en les sélectionnant dans l'environnement virtuel ;
- afficher dans la scène de façon dynamique l'indice de risque de la situation simulée en temps réel.

Par conséquent, ces modèles de situation de travail permettent de simuler différentes configurations de la presse à partir d'un modèle de base en modifiant les protections et/ou les modes opératoires. Dans notre cas de figure, nous pouvons utiliser différents dispositifs de sécurité comme par exemple les protections latérales, le barrage immatériel qui protège une large zone devant l'opérateur ou encore les systèmes laser qui protègent localement les doigts dans une zone restreinte proche du poinçon. Ces derniers seront choisis en adéquation avec le scénario que l'on souhaite jouer [Inrs 04a].

Dans la Figure V.120, les positions des mains se trouvent dans une zone dangereuse. Elles sont alors détectées par les faisceaux laser qui stoppent la descente du poinçon et qui déclenchent l'arrêt de la machine.

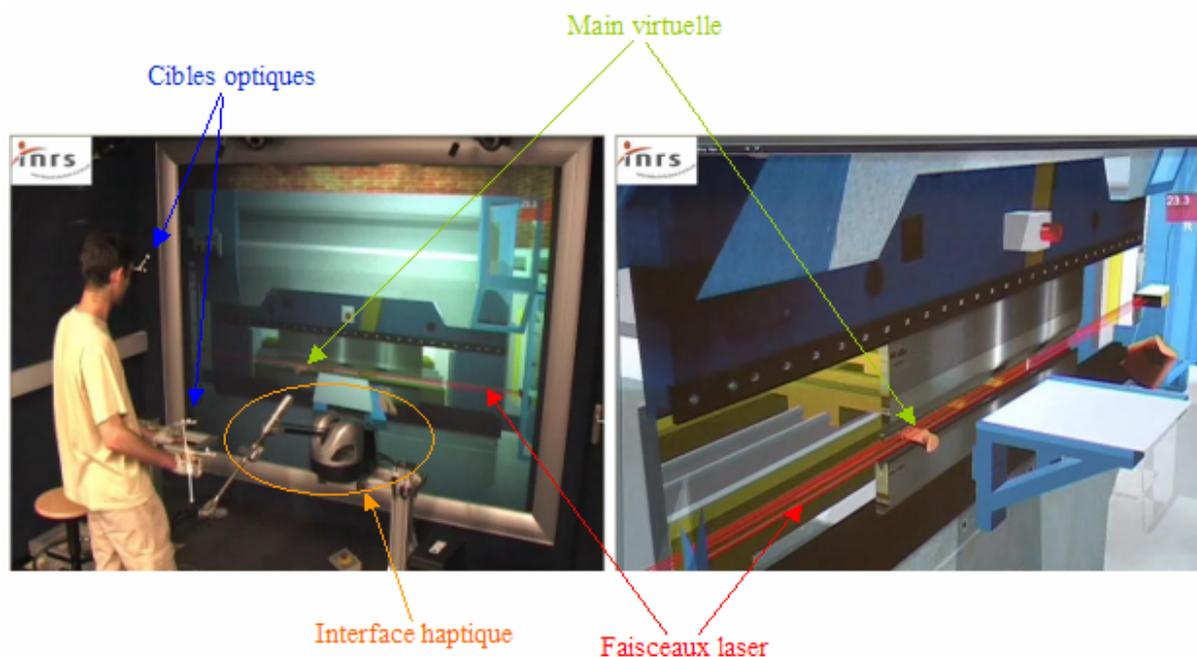


Figure V.120 : Utilisation des faisceaux laser pour protéger les mains de l'opérateur

Après avoir configuré la presse, il est important que l'utilisateur soit informé du risque qu'il encourt en réalisant son opération de pliage. C'est le rôle de l'indice de risque. Dans le paragraphe suivant nous présentons le principe de cet indice.

4.3 L'indice de risque dynamique

L'objectif de cet outil dynamique est d'évaluer le niveau de sécurité de la situation simulée sur la presse plieuse virtuelle. L'indice de risque a donc été paramétré en fonction de la méthodologie décrite par AISS [AISS 98] afin de vérifier, en environnement virtuel, que les choix techniques, les procédures ou encore les instructions de travail mises en œuvre ne dégradent pas le niveau de sécurité de la future situation de travail en regard de l'équipement utilisé. Nous avons donc un indice global de la situation de travail que l'on note R. Il est calculé à partir de trois facteurs :

- les risques liés à la machine : l'indice M ;
- les risques dus à l'environnement de travail : l'indice E ;
- et, le niveau de compétences de l'opérateur, i.e. l'habilité à contrôler ces différents risques : il est noté P.

Chacun de ces facteurs dépend de différents paramètres. Certains de ces paramètres sont statiques car ils sont intrinsèques à la situation de travail. C'est le cas du niveau d'énergie, du niveau de sécurité des équipements de protection, de la qualification de l'opérateur, etc. Ils peuvent néanmoins être modifiés durant la simulation, et ensuite être mis à jour dans la base de données MOSTRA.

D'autres paramètres sont dynamiques. Ils ne sont pas lus dans la base mais ils sont calculés au cours de la simulation. C'est par exemple le cas de paramètres relatifs à la tâche et à l'opérateur comme la durée de la tâche, la posture de l'opérateur ou la fréquence d'exposition dans la zone dangereuse.

Tous ces facteurs dépendent de l'application simulée. Par conséquent, nous pouvons estimer le niveau global de risque de la simulation et le comparer avec des indices de risques prédéfinis. Ce coefficient R et les trois facteurs M, E et P sont affichés à l'écran en haut à droite par des jauges de couleurs (Cf. Figure V.121) :

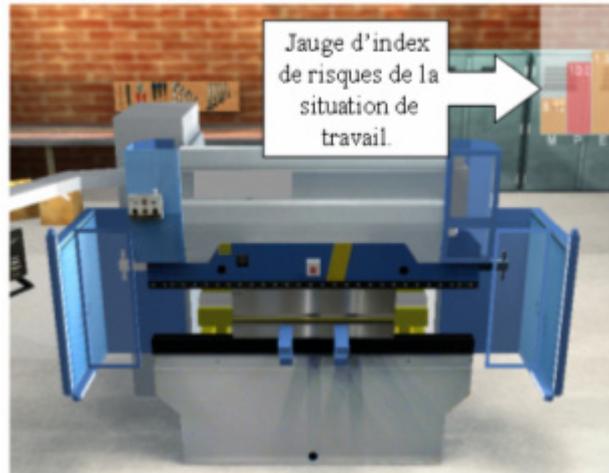


Figure V.121 : Estimation du risque pour une configuration donnée

La couleur est définie par des valeurs seuils (Cf. Figure V.122). Ainsi, l'indice de risque varie en fonction des actions de l'opérateur pendant qu'il manipule la tôle métallique virtuelle. A chaque pas de temps, il permet à l'opérateur de se rendre compte de l'amélioration ou de la dégradation des conditions de sécurité de l'opération de pliage en cours [Marsot 05].

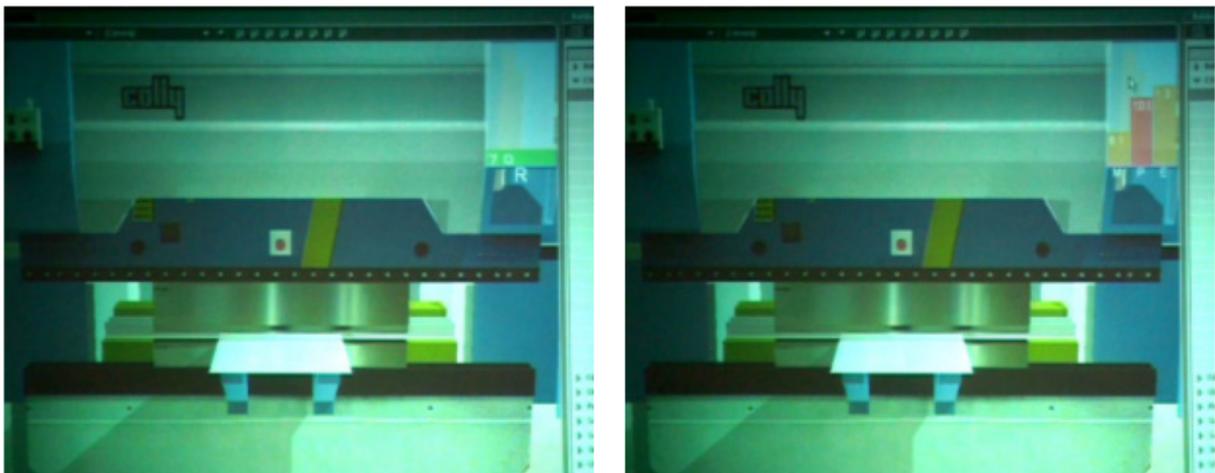


Figure V.122 : Affichage des facteurs de risque

L'intégration des mains virtuelles dans la simulation de la presse plieuse a permis d'affiner la mesure de la fréquence et celle de la durée d'exposition directement liées à la distance séparant l'extrémité des doigts de l'opérateur de la zone dangereuse (ligne poinçon/matrice). Ce paramètre agit de façon proportionnelle sur l'indice de risque « M » et donc sur l'indice de risque global « R ». La Figure V.123 illustre le calcul dynamique de l'indice de risque : plus la main se rapproche de la presse, plus le risque global de la simulation augmente.



Figure V.123 : Illustration de l'indice de risque dynamique avec la main

Quant à la Figure V.124, elle illustre l'activation des dispositifs de sécurité lorsque la main est proche du poinçon :

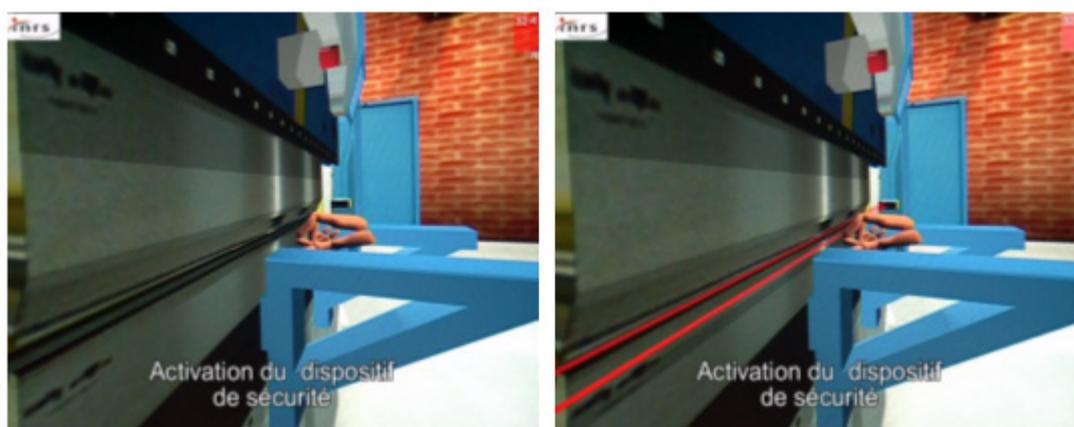


Figure V.124 : Activation du dispositif de sécurité avec la main

Nous avons donc à notre disposition des outils qui nous permettent d'estimer en temps réel le degré de risques d'accidents encouru par l'opérateur.

Dans le paragraphe suivant, nous montrons les résultats obtenus avec la simulation de la presse plieuse et les mains virtuelles. Nous avons développé et testé différentes applications sur la plateforme de Réalité Virtuelle de l'INRS. Ceci nous a permis de mettre en évidence l'intérêt du modèle de main pour la simulation des interactions Homme-machine ainsi que celui des outils de prévention pour l'estimation des dangers encourus par l'opérateur sur son poste de travail.

5 Les applications développées

Nous avons développé conjointement avec l'INRS trois applications qui vont nous aider à juger si les objectifs fixés en début de ce chapitre ont été atteints ; c'est-à-dire l'interaction en temps réel avec un environnement virtuel, la simulation d'une situation de travail et l'intégration de la prévention des risques. Tout d'abord, nous présenterons une opération de pliage de tôle. Puis, nous montrerons comment l'utilisateur interagit avec l'environnement virtuel en utilisant la capture de mouvements.

5.1 La simulation d'une opération de pliage de tôle avec du retour d'efforts

5.1.1 Description

Dans cette application, l'opérateur utilise le bras à retour d'efforts pour déplacer la tôle métallique. Cette application permet de mettre en œuvre les modèles décrits au début du Chapitre V pour simuler un cas réel avec un équipement de travail virtuel. Elle nous permet également de savoir si les outils développés améliorent l'intégration de l'utilisateur dans la simulation et s'ils nous aident à estimer le risque d'une opération de pliage de tôle.

5.1.2 Objectif(s)

L'utilisateur doit plier la tôle métallique virtuelle à l'aide d'un bras à retour d'effort en suivant un scénario prédéfini.

5.1.3 Résultats

Dans ce cas de figure, nous définissons tout d'abord les modes opératoires, les dispositifs de sécurité et les plis à effectuer sur la pièce en choisissant une configuration de la presse plieuse ainsi qu'un scénario de pliage de tôle (Cf. Figure V.125). En l'occurrence, nous avons choisi comme moyens de protection des systèmes laser.

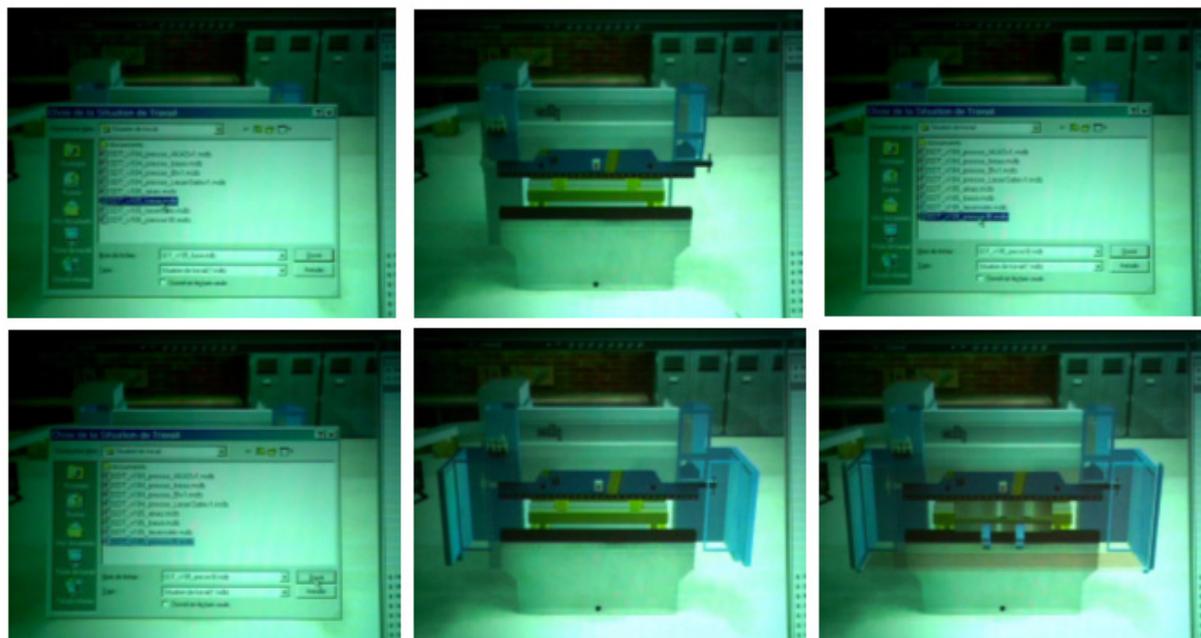


Figure V.125 : Chargement de la configuration de la presse plieuse

Une fois la situation de travail chargée, l'utilisateur est prêt pour réaliser une opération de pliage de tôle. Pour cela, il lance la simulation en appuyant sur la pédale (Cf. § 2.1). La plaque de plastique attachée au manche de l'interface haptique lui permet de contrôler le mouvement de la tôle (Cf. Figure V.126) et donc de simuler une opération de pliage.

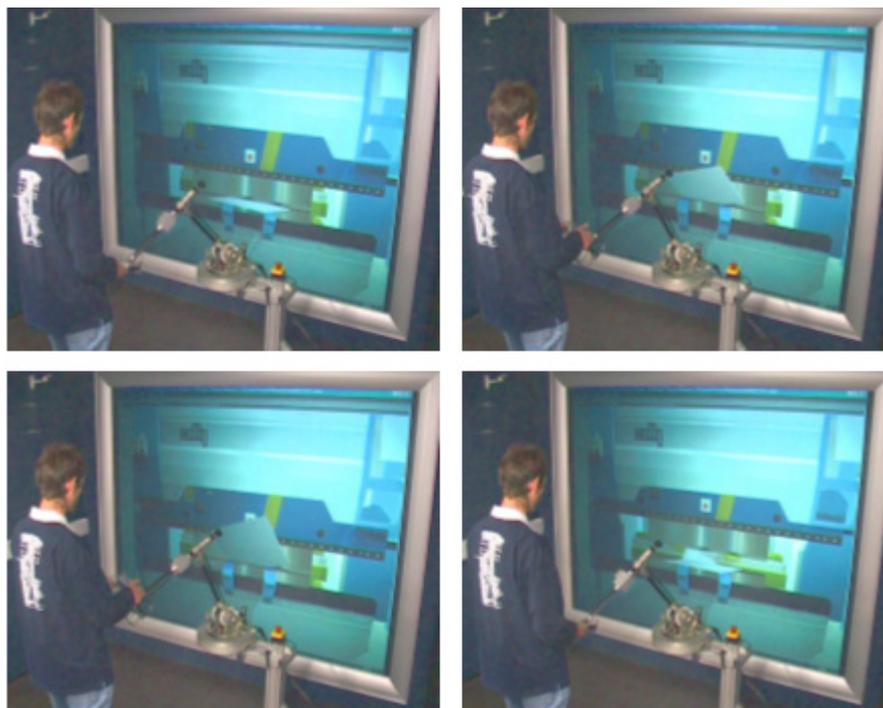


Figure V.126 : Simulation d'un pliage de tôle

De plus, l'opérateur est informé à chaque instant du risque qu'il encourt pendant la simulation par l'indice affiché en haut de l'écran (Cf. § 4.3).

Ainsi, cette simulation montre qu'il est possible de réaliser une opération de pliage de tôle en utilisant un bras à retour d'effort pour commander la tôle virtuelle. De plus, le modèle de situation de travail permet de configurer la presse virtuelle afin de simuler de manière réaliste un pliage de tôle ; ce qui aide l'utilisateur à se sentir immergé dans la simulation. Il est également possible d'estimer le danger encouru par l'utilisateur au cours de l'opération via un indice de risque dynamique.

Dans le paragraphe suivant, nous simulons la saisie et la manipulation d'objets virtuels en utilisant des données issues de la capture de mouvements.

5.2 L'interaction avec l'environnement virtuel

5.2.1 Description

Dans cette application, les mains de l'opérateur sont traquées par des capteurs optiques et les doigts par des gants de données. Nous simulons la préhension de la tôle en n'utilisant que les données issues de la capture de mouvements afin de tester les algorithmes présentés dans le Chapitre III.

Les résultats de cette application ont fait l'objet d'un article présenté à Virtual Concept [Pouliquen 05b]. La simulation a également été récompensée par le deuxième prix de la compétition Immersion & Interactive Design de cette conférence.

5.2.2 Objectif(s)

L'opérateur doit saisir et manipuler la tôle avec les mains virtuelles en utilisant uniquement

les informations issues de la capture de mouvements.

5.2.3 Résultats

Dans ce cas de figure, l'opérateur interagit avec la presse virtuelle par le biais du système de capture de mouvements et les gants de données. Cela signifie qu'il doit se positionner dans la zone de démonstration pour se trouver proche de l'objet à saisir et qu'il doit refermer ses doigts pour bloquer l'objet en position dans la main virtuelle.

En premier lieu, nous n'avons utilisé que le système Vicon. Cela revient à interagir avec des mains non articulées. Nous avons réussi à abaisser et à soulever la tôle virtuelle (Cf. Figure V.127).



Figure V.127 : Manipulation avec les deux mains

Puis, nous avons couplé un gant de données au système Vicon. Nous avons ainsi piloté la main virtuelle articulée, présentée dans les Chapitres III et IV, ce qui nous a permis de simuler des tâches de saisie et manipulation d'objets virtuels comme avec la tôle virtuelle sur la Figure V.128.

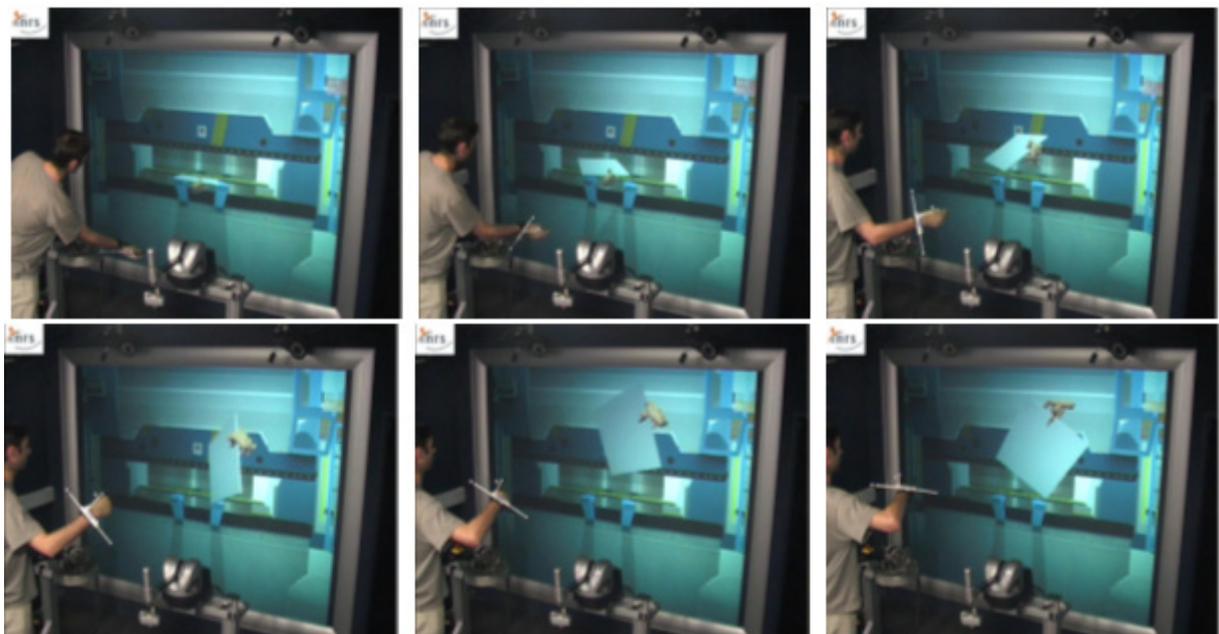


Figure V.128 : Saisie de la tôle avec une main virtuelle articulée

La Figure V.129 montre que l'asservissement permet de maintenir la tôle dans la main

virtuelle en toute stabilité :

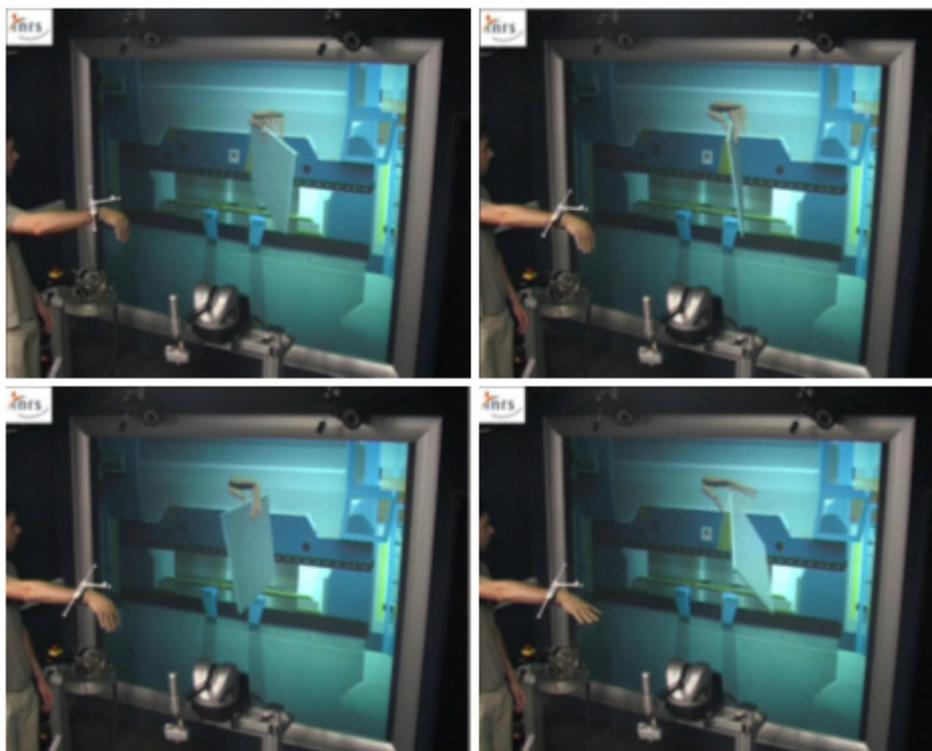


Figure V.129 : Maintien de la tôle dans la main et lâcher

Ainsi, le couplage virtuel de type ressort-amortisseur réalisé entre les mains de l'utilisateur et les mains virtuelles (Cf. Chapitre III) permet de reproduire les mouvements des mains de l'opérateur. Il permet également d'interagir en temps réel avec la presse virtuelle ou la tôle métallique. Le système que nous avons développé permet donc de simuler une tâche manuelle qui est le principal mode d'interaction avec l'environnement.

5.3 Le pliage avec du retour d'effort et de la capture de mouvements

5.3.1 Description

Dans cette application, les mains de l'opérateur sont traquées par la capture de mouvements – le système Vicon et le gant de données – se déplacent dans l'environnement virtuel et permettent de situer l'opérateur. La tôle est manipulée par le biais du bras à retour d'efforts.

5.3.2 Objectif(s)

L'utilisateur doit plier la tôle métallique virtuelle à l'aide d'un bras à retour d'effort en suivant un scénario prédéfini. De plus, il évolue dans la scène virtuelle par le biais du système de capture de mouvements.

5.3.3 Résultats

L'opération de pliage de tôle s'effectue dans la même configuration que celle du paragraphe 5.1. Le bras à retour d'effort permet de commander la tôle virtuelle pendant que les cibles optiques et les gants de données suivent les mouvements des mains de l'utilisateur (Cf. Figure

V.130) :



Figure V.130 : Pliage de tôle simulé avec les mains virtuelles et le retour d'effort

Il est également possible de suivre les mouvements des mains virtuelles au cours de l'opération de pliage de tôle (Cf. Figure V.131) :

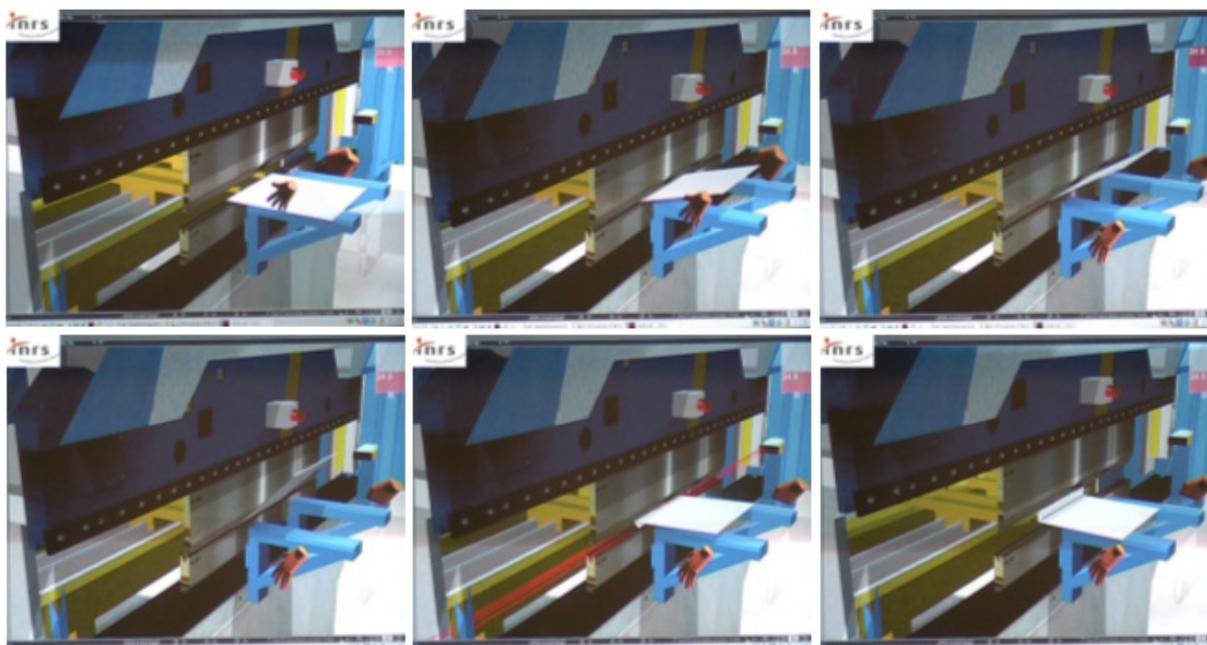


Figure V.131 : Suivi des mains de l'utilisateur dans la simulation

Les mains permettent de pousser la tôle jusqu'aux butées afin qu'elle se trouve sous le poinçon. Au moment du pliage, l'utilisateur n'interagit pas avec la tôle : il la récupère une fois le pli effectué pour la mettre en position pour le pli suivant.

Mais au cours de cette simulation, nous avons rencontré des difficultés pour coupler les asservissements des mains décrits dans le Chapitre III avec le contrôleur du bras à retour d'efforts. Pour remédier à ces instabilités et intégrer les mains dans l'opération de pliage de tôle, nous ne mettons en œuvre qu'une consigne en position pour piloter les mains virtuelles. Cette solution conduit à un décalage entre les déplacements et les configurations des mains de l'utilisateur et celles des mains virtuelles, ainsi qu'à une perte de la consistance physique des mains virtuelles. Cela signifie que des interpénétrations entre les mains et la tôle virtuelles peuvent avoir lieu au cours de la simulation.

Ce problème peut être résolu en modifiant les gains des contrôleurs utilisés pour la capture de

mouvements et pour le bras à retour d'efforts. En effet, nous avons paramétré les contrôleurs pour la paume et pour les doigts sans utiliser le retour d'effort et les valeurs ont été définies à la limite de l'instabilité uniquement avec ces interfaces. De la même manière, le bras à retour d'effort a été configuré dans le cas d'une utilisation sans capture de mouvements. Malheureusement, ces modifications n'ont pu être effectuées pour nos applications car nous n'avions pas la possibilité de modifier les gains du contrôleur du bras à retour d'efforts.

5.4 Conclusions des applications

Ainsi, les applications ont mis en évidence l'intérêt des mains virtuelles pour interagir avec l'environnement virtuel et intégrer l'utilisateur dans les simulations. De plus, les résultats obtenus tendent à mettre en évidence l'intérêt des techniques de la Réalité Virtuelle pour intégrer la prévention des risques dans la conception d'équipements de travail. Mais, pour s'en assurer, il est important d'effectuer des tests de validation afin de déterminer les apports et les limites d'un tel simulateur.

Dans le paragraphe suivant, nous présentons le processus de validation mis en œuvre par l'INRS pour le simulateur de presse plieuse.

6 La validation du simulateur de presse plieuse

Une équipe de psycho-ergonomes et d'ingénieurs de l'INRS a été chargée de la validation globale du simulateur ainsi que de celle de la plateforme de Réalité Virtuelle. Pour cela, Marc et al. ont suivi deux approches [Marc 06] :

- la première approche concerne le pouvoir suggestif des simulations produites en RV. Ainsi, une simulation en Réalité Virtuelle est considérée comme valide si elle permet à des opérateurs de se projeter dans une activité future probable avec plus de facilité ou plus de précision en comparaison avec d'autres outils de représentation externe comme un plan ou une simulation statique. C'est ce qu'on désigne par la « simulation cognitive » ;
- la seconde approche renvoie à la reproduction de l'activité d'un opérateur en simulation. Elle se focalise sur la possibilité de reproduire une activité de référence à travers la simulation. Deux types de validité sont alors à considérer pour la simulation :
 - la validité interne qui s'appuie sur la capacité du simulateur à reproduire l'ensemble des fonctionnalités de l'équipement de travail simulé, et également à reproduire une tâche qui est considérée comme la succession d'opérations simples ;
 - la validité externe qui permet de vérifier si des réalisations et des activités comparables à celles observées en situation réelle peuvent être reproduites en simulation. Les situations servant pour la comparaison sont à la fois des situations de laboratoire et des situations d'entreprise. Dans ces cas de figure, les situations de laboratoire permettent de tester des productions de pièces complexes en situation contrôlée. Alors que les situations de terrain permettent de prendre comme référence des situations de production de pièces plus simples mais avec des contraintes plus importantes comme par exemple les contraintes de cadences.

Les premiers résultats obtenus à l'issue de ces expériences ont montré une bonne validité interne du simulateur de presse plieuse. Cela signifie que le comportement de l'opérateur dans l'environnement virtuel est conforme à celui observé en situation réelle. De même, les difficultés de pliages rencontrées en situation réelle ont été retrouvées en simulation. Cependant, la situation simulée ne permet pas de reproduire les conditions réelles de l'opération de pliage. En effet, en raison du bras à retour d'effort, l'opérateur n'est pas dans une position « naturelle » pour réaliser l'opération de pliage de tôle. Pour remédier à ce problème, une co-localisation parfaite entre l'opérateur et la machine virtuelle est nécessaire ; ce qui est impossible à réaliser à l'heure actuelle en raison de l'interface haptique [Marc 06].

Les premières évaluations ont montré que la Réalité virtuelle permet de simuler les futures fonctions d'usage d'un équipement de travail (validité interne) et des dispositifs de sécurité associés ainsi que de reproduire une tâche en amont du processus de conception.

D'autres expériences sont en cours afin de vérifier s'il est possible de reproduire en simulation des réalisations et des activités comparables à celles observées en situation réelle (validité externe). Mais pour que les prototypes virtuels trouvent pleinement leur place dans le cycle de conception, il est nécessaire que les interfaces utilisées, et plus particulièrement les interfaces haptiques, soient plus performantes en terme d'amplitude de mouvements et d'accessibilité.

7 Conclusion

Dans ce chapitre, nous avons présenté trois applications utilisant la plateforme et les interfaces de Réalité Virtuelle pour interagir avec un environnement virtuel complexe. Ainsi, en mettant en œuvre les modèles présentés dans les Chapitres III et IV, nous avons intégré les mains virtuelles dans le simulateur de la presse plieuse. L'utilisateur est alors capable de simuler de manière stable une opération de pliage de tôle ou encore une tâche de préhension d'objets. Ainsi, les techniques de la Réalité Virtuelle offrent la possibilité de faire interagir les opérateurs réels avec le système qui n'est alors décrit qu'informatiquement. Cette interaction donne aux concepteurs la possibilité de découvrir certains problèmes et de rétro-agir avant même la réalisation d'un premier prototype physique.

Nous avons également intégré dans la simulation les outils développés pour la prévention des risques comme les modèles de situations de travail et l'indice de risque dynamique. Ces derniers permettent d'estimer à chaque instant le risque encouru par l'utilisateur au cours de ces différentes opérations et de l'en informer en temps réel. Ils permettent donc de prendre en compte les questions de sécurité très en amont dans le cycle de conception. Par conséquent, ils rendent possible l'estimation du risque d'accidents sur un système à chaque étape de la conception de ce dernier. Ceci est essentiel à la protection du futur opérateur car les décisions prises dans la conception ont souvent un impact sur la sécurité du produit ou de la production. En conclusion, les techniques de Réalité Virtuelle et les mains virtuelles se révèlent être utiles au concepteur car elles offrent la possibilité de simuler les interactions Homme-machine très en amont dans le cycle de conception. De plus, les outils de prévention des risques développés permettent de mieux intégrer l'utilisateur dans la simulation et conduisent ainsi à une conception plus sûre d'un poste de travail.

CONCLUSION

La modélisation de la main humaine fait l'objet de nombreuses recherches dans des domaines aussi divers que la médecine, la robotique ou encore l'animation graphique. Dans ce mémoire, nous avons plus particulièrement étudié la main en tant que principal organe d'interaction de l'Homme avec son environnement et nous avons abordé le problème de sa modélisation pour des applications en environnement virtuel.

Après avoir présenté la main humaine, nous avons étudié les modèles de mains existants. Cette étude nous a conduit à la définition d'un modèle multicouches de la main ainsi qu'à l'identification des caractéristiques fondamentales à respecter pour simuler des tâches de préhension.

Ainsi, la base de notre modèle de main est constituée d'un squelette rigide qui respecte la cinématique de la main humaine et qui tient compte des contraintes biomécaniques. Il est constitué de seize primitives simples (sphères, cylindres et parallélépipède) articulées par vingt degrés de liberté pour les doigts et six pour le poignet. Pour animer ce modèle géométrique de la main virtuelle, nous utilisons les données issues de la capture de mouvements. Nous avons donc développé des contrôleurs de type proportionnel-dérivé pour la main et les doigts virtuels. Ils consistent à calculer les efforts – couples et/ou forces – à appliquer aux objets virtuels afin de réduire les erreurs en position et en vitesse qui existent entre la main réelle et la main virtuelle au cours des mouvements et/ou des tâches de saisie et manipulation. Ces asservissements confèrent à la main un réalisme physique qui permet à l'utilisateur d'interagir de manière stable avec l'environnement virtuel et plus spécifiquement de manipuler des objets de la scène virtuelle en temps réel. L'ensemble de ces développements constituent une bibliothèque en langage C/C++ appelée VirtualHand.lib.

Afin de pouvoir simuler les déformations de la main humaine dues aux contacts lors de la saisie et de la manipulation d'objets virtuels, nous nous sommes ensuite intéressés à la modélisation des tissus adipeux et des muscles. Nous proposons donc de modéliser cette couche intermédiaire de notre main virtuelle par des coussinets déformables positionnés sur le squelette rigide décrit précédemment. La définition de ces coussinets nous a conduit à développer un modèle de préhension dédié à la Réalité Virtuelle en intégrant une méthode de détection de collision, une méthode de résolution de contacts et un modèle de déformation. Ainsi, chaque objet de la scène est représenté par une hiérarchie de volumes englobants.

La méthode de détection de collision développée permet d'identifier les couples de triangles en contact. Ces informations nous permettent de modéliser le contact sous forme de relations entre les résultantes des efforts appliqués et le déplacement relatif du doigt : nous construisons ainsi un problème de complémentarité linéaire. Nous résolvons ce système d'équations par un algorithme de type Gauss Seidel. Nous obtenons ainsi les efforts à appliquer aux objets.

Ces derniers sont modélisés par des éléments finis. Nous avons intégré un modèle élastique linéaire afin de simuler le comportement des coussinets et de la pulpe des doigts humains. L'hypothèse des petites déformations et des petits déplacements peut être jugée restrictive, mais dans notre cas de figure nous réactualisons la géométrie de nos coussinets à chaque itération et/ou à chaque contact pour s'affranchir au mieux de ce modèle simplifié. Nous réalisons ainsi une simulation quasi statique qui se révèle être tout à fait réaliste pour nos applications en Réalité Virtuelle. Nous sommes donc parvenus à simuler des déformations des

coussinets sous l'action des forces de contact qui sont réalistes du point de vue physique mais qui ne permettent pas de respecter la contrainte de temps réel. Il est donc possible de saisir et manipuler des objets rigides avec des mains virtuelles déformables mais dans des environnements peu contraints.

Les différents algorithmes développés pour construire notre modèle de main forment deux bibliothèques de calcul : VirtualHand.lib et Collision.lib. Ils ont été testés dans le cadre de différentes applications en environnement virtuel.

Tout d'abord nous avons testé le modèle géométrique proposé et réglé les asservissements en position utilisés à l'aide d'une application de manipulation d'objets puis d'une simulation de pliage de tôle. Dans ces deux cas de figure, les mains virtuelles ont permis d'interagir avec la presse plieuse ou l'environnement virtuel de manière stable.

Puis, nous avons couplé notre main virtuelle à un indice de risque dynamique ce qui nous a permis d'estimer à chaque instant le risque encouru par l'utilisateur et de l'en informer en temps réel.

Bien que ces développements aient été réalisés dans le cadre d'une application de pliage de tôle, le modèle et les asservissements sont suffisamment génériques pour être utilisés dans d'autres applications en environnement virtuel.

Nous avons constaté certaines limitations à l'utilisation de nos modèles dans les différentes applications. Bien que notre modèle squelettique nous permette d'interagir avec l'environnement virtuel en temps réel de manière stable, nous avons observé des décalages au cours de mouvements rapides de rotation. Ceux-ci peuvent devenir gênants pour l'utilisateur car ils ont un impact sur le réalisme de la simulation et ils diminuent son sentiment d'immersion. De plus, les performances du modèle de préhension, i.e. la bibliothèque Collision.lib, se sont révélées être insuffisantes pour des simulations temps réel dans des environnements virtuels complexes. Nous nous sommes donc focalisés sur la validation du modèle géométrique et sur son intérêt dans la prévention des risques d'accidents.

En conclusion, les techniques de la Réalité Virtuelle et les mains virtuelles permettent de simuler les interactions Homme-machine en amont du cycle de conception. En intégrant des outils de prévention des risques, cette interaction donne aux concepteurs la possibilité de découvrir certains problèmes et de rétroagir avant même la réalisation d'un premier prototype physique. Il devient donc possible de mieux intégrer l'utilisateur dans la simulation et de prendre en compte les questions de sécurité au plus tôt dans le cycle de conception. L'ensemble, mains virtuelles et outils de prévention des risques, offre la possibilité de concevoir un poste de travail plus sûr.

Les prolongements possibles de ces travaux de recherche sont nombreux et variés. Tout d'abord, nous intégrerons de nouvelles lois de commande de type Proportionnel Intégral et Dérivé dans la bibliothèque VirtualHand.lib afin de mieux contrôler les mouvements de l'utilisateur en rotation dans l'environnement virtuel. Il sera donc nécessaire de faire une nouvelle étude de la stabilité numérique avec ces nouveaux contrôleurs.

Puis, nous devons optimiser les algorithmes de la bibliothèque Collision.lib afin d'améliorer les performances. Cela consistera à intégrer des nouveaux types de volumes englobants pour accélérer la détection de collision et à optimiser la programmation de l'algorithme de Gauss Seidel pour qu'il converge plus vite vers une solution.

Un autre défi consistera à coupler notre main virtuelle à une interface à retour d'effort à doigts adaptée afin de manipuler des objets virtuels. Il sera alors nécessaire d'identifier les nouvelles lois de commande pour cette interface. Dans ce cas de figure, il ne sera plus nécessaire de suivre les mouvements des doigts : il suffira de connaître la position et l'orientation de la main de l'utilisateur dans l'environnement.

Enfin, il serait souhaitable de réaliser des mesures expérimentales précises de la pulpe et des coussinets phalangiens *in vivo* afin de mieux définir le coefficient d'adhérence de la peau ainsi que la zone de contact lors de différentes tâches de préhension. Ceci nous permettrait d'intégrer des caractéristiques mécaniques et physiologiques proches de celles de la main humaine, de prendre en compte les surfaces de contact ou encore d'optimiser les formes géométriques des coussinets déformables. Nous obtiendrons ainsi un modèle physique qui sera plus proche de la réalité. Le modèle proposé tient compte de cet objectif puisqu'il offre la possibilité d'intégrer d'autres lois de comportement grâce à une modélisation locale des propriétés mécaniques.

REFERENCES BIBLIOGRAPHIQUES

- [AISS 98] AISS, "Calculez vous-même vos risques accidents – Appréciation du risque mécanique au poste de travail", AISS Prévention, séries n°2034 (F), ISBN 92-843-2130-1, DTE 175, 32 p., 1998
- [Albrecht 03] I. Albrecht, J. Haber, and H-P. Seidel, "Construction and Animation of anatomically-based human hand models", Dans les actes de Eurographics / ACM SIGGRAPH Symposium on Computer Animation, San Diego, USA, 2003
- [Asada 02] H.H. Asada et S. Mascaro, « Fingernail sensors for measurement of fingertip forces and finger posture – Part 2 : Filter design and Calibration », Rapport Technique Phase 3, Home Automation and Healthcare Consortium, Laboratory for Information Systems and Technology, MIT, 2002
- [Baraff 89] D. Baraff, "Analytical methods for dynamic simulation of non-penetrating rigid bodies", *Computer Graphics*, 23 : 223-232, 1989
- [Baraff 90] D. Baraff, "Curved surfaces and coherence for non-penetrating rigid body simulation", Dans les Actes de SIGGRAPH'90, *Computer Graphics*, 24(4): 19-28, 1990
- [Baraff 94] D. Baraff, "Fast contact force computation for non-penetrating rigid bodies", Dans Andrew Glassner editor, dans les Actes de ACM SIGGRAPH '94, pp. 23-34, Orlando, USA, 1994
- [Baraff 98] D. Baraff et A. Witkin, « Large steps in cloth simulation », dans les Actes de SIGGRAPH'98, *Computer Graphics*, Orlando, USA, juillet 98
- [Bathe 96] K-J. Bathe, *Finite Element Procedures*, Editions Prentice-Hall, Inc., ISBN 0-13-301458-4, 1996
- [Bell 00] J.T. Bell et H. Scott Fogler, "A Virtual Reality Safety and Hazard Analysis Simulation", dans les Actes de American Society for Eng. Education Annual Conf., St Louis, USA, American Society Engineering Education, 2000
- [Bell 01] J.T. Bell et H. Scott Fogler, "Virtual Reality Laboratory Accidents", dans les Actes de American Society for Engineering Education Annual Conference & Exposition, ASEE'01, 2001
- [Bernier 97] S. Bernier, "Modèle théorique de la rigidité des doigts humains", Mémoire de Master, Département de Génier Mécanique, Faculté des Sciences et de Génie, Université de Laval, Canada, avril 1997
- [Bloomenthal 91] J. Bloomenthal et K. Shoemake, "Convolution Surfaces", dans les Actes de SIGGRAPH'91, pp.251-256, 1991
- [Bloomenthal 93] J. Bloomenthal, "Hand Crafted", SIGGRAPH Course Notes 25, 1993
- [Borst 05] C.W. Borst et A.P. Indugula, "Realistic Virtual Grasping", dans les Actes de IEEE Virtual Reality Conference 2005 (VR'05), pp. 91-98, 320, Bonn, Allemagne, 2005.
- [Bouisset 02] Simon Bouisset, *Biomécanique et physiologie du mouvement*, Collection Abrégés de Masson, ISBN 2-225-83157-2, 2002
- [Boulic 96] R. Boulic, S. Rezzonico et D. Thalmann, « Multi-finger manipulation of virtual objects », dans les Actes de ACM Symposium on Virtual Reality Software and Technology, VRST'96, pp. 67-74, Hong Kong, 1996
- [Boyer 05] E. Boyer, "Représentation des courbes et des surfaces", Cours RICM2 – Synthèse d'images, Institut d'Informatique et de Mathématiques Appliquées de Grenoble, Université Joseph Fourier, 2005
- [Brand 81] P.W.Brand, R.B. Beach et D.E. Thompson, « Relative tension and potential excursion of muscles in the forearm and hand », *The Journal of Hand Surgery*, Vol. 6, n°3, pp.209-219, mai 1981
- [Bro Nielsen 96] M. Bro-Nielsen et S. Cotin, "Real-time volumetric deformable models for surgery simulation using finite elements and condensation", *Computer Graphics Forum (Eurographics'96)*, vol. 15("), pp. 57-66, 1996
- [Bro Nielsen 98] M. Bro Nielsen, "Finite Element Modeling in Surgery simulation", dans les Actes de IEEE, Invited Paper, 86: 3, pp. 490–503, 1998

- [Brown 02] J. Brown, S. Sorkin, J-C. Latombe, K. Montgomery et M.Stephanides, "Algorithmic tools for real-time microsurgery simulation", *Medical Image Analysis*, Elsevier, 6(3) : 289-300, 2002
- [Brown 03] J. Brown, "Real time soft tissue and suture simulation", Thèse de Doctorat, Département d'Animation Graphique, Université de Stanford, 2003
- [Burdea 93] G. Burdea et Ph. Coiffet, *La Réalité Virtuelle*, Editions Hermès, ISBN : 2-86601-386-7, 402 pages, 1993
- [Burkhardt 03] J.-M. Burkhardt, « Réalité virtuelle et ergonomie : quelques apports réciproques », *Le travail humain*, Vol. 66, No.1, Janvier-Mars 2003
- [Cameron 97] S. Cameron, « Enhancing GJK : Computing minimum penetration distances between convex polyhedra », Dans les Actes de IEEE International Conference on Robotics and Automation, 4 Aut., pp. 3112-3117, 1997
- [Capell 02] S. Capell, T. Duchamp, and Z. Popovic, "Interactive skeleton-driven dynamic deformations", dans les actes de ACM SIGGRAPH'02, San Antonio, USA, 2002
- [Castaing 99a] J. Brizon et J. Castaing, *Les feuillets d'anatomie – Fascicule 1 : ostéologie du membre supérieur*, Maloine, ISBN 2 224 000588, 1999
- [Castaing 99b] J. Brizon et J. Castaing, *Les feuillets d'anatomie – Fascicule 4 : Muscles du membre supérieur*, Maloine, ISBN 2 224 000847, 1999
- [Chèze 05] L. Chèze, "Modèle mécanique du muscle squelettique", Cours de Biomécanique, Mécanique et Structure, Section de Génie Mécanique, Faculté des Sciences et Techniques de l'Ingénieur, Ecole Polytechnique Fédérale de Lausanne, Année Universitaire 2004/2005
- [Ciccotelli 97] J. Ciccotelli, « Vers des machines et des systèmes plus sûrs – Quelques perspectives de recherche et de développement », *CND Hygiène et Sécurité du Travail* 166, pp. 189-199, 1997
- [Ciccotelli 02a] J. Ciccotelli, J. Marsot et F. Gardeux, « Environnement Virtuel pour la Conception de Systèmes sûrs (EVICS) », Rapport d'étude d'instruction du projet, IET – S / 02DT – 064 - JCI, octobre 2002
- [Ciccotelli 02b] J. Ciccotelli, « Réalité virtuelle, une aide à la décision pour la conception de systèmes sûrs », Dans les Actes de European Conf. On System Dependability and Safety Decision Making and Risk Management, Lyon, France, 2002
- [Ciccotelli 05] J. Ciccotelli et J. Marsot, « Réalité Virtuelle et Prévention, Apports et tendances », *HST-CND*, n° 199, pp. 99-111, ISSN 0007-9952, 2ème trimestre 2005
- [Cobb 99] S. Cobb, S. Nichols, A.R. Ramsey et J.R. Wilson, "Virtual Reality Induced Symptoms and Effects", *Presence : Teleoperators and Virtual Environments*, 8(2), 1999
- [Cohen 95] J. D. Cohen, M. C. Lin, D. Manocha, M. K. Ponamgi, « I-COLLIDE: An Interactive and Exact Collision Detection System for Large-Scale Environments », Dans les Actes du Symposium on Interactive 3D Graphics, ACM SIGGRAPH'95, pp. 189-196, Los Angeles, USA, 1995
- [Coquillart 90] S. Coquillart, "Extended Free Form Deformation : a sculpting tools for 3D geometric modelling", Dans les Actes de SIGGRAPH'90, pp.187-196, 1990
- [Costello 97] P.J. Costello, « Health and Safety issues associated with Virtual Reality – A review of current literature », Rapport Technique 37, 23p., Advisory Group on Computer Graphics (AGOCG), Université de Loughborough, Royaume Uni, 1997
- [Cotin 99] S. Cotin, H. Delingette et N. Ayache, « Real Time elastic deformations of soft tissues for surgery simulation », *IEEE Transactions on Visualization and Computer Graphics*, 5: 62-73, 1999
- [Dandekar 97] K. Dandekar et M. A. Srinivasan, "Role of Mechanics in Tactile Sensing of Shape", *Touch Lab Report 2*, RLE TR-604, MIT, Cambridge, 1997

- [Dandekar 03] K. Dandekar, B.I. Raju et M.A. Srinivasan, "3-D Finite-Element Models of Human and Monkey Fingertips to Investigate the Mechanics of Tactile Sense", *Journal of Biomechanical Engineering*, Vol. 125, pp. 682-691, ASME Press, 2003
- [Davis 94] J. Davis et M. Shah, « Determining 3-D hand motion », dans les Actes de 28th Asimolar Conference on Signals, Systems and Computer, 1994
- [Debunne 00] G. Debunne, "Animation multirésolution d'objets déformables en temps-réel", Thèse de Doctorat, Institut National Polytechnique de Grenoble, 2000
- [Delingette 98] H. Delingette, "Towards Realistic Soft Tissue Modeling in Medical Simulation", Numéro spécial IEEE Surgery Simulation, pp. 512-523, 1998
- [Delingette 04] H. Delingette et N. Ayache, "Soft Tissue Modeling for Surgery Simulation", dans *Computational Models for the Human Body*, N. Ayache, editor, *Handbook of Numerical Analysis* (Ed : Ph. Ciarlet), Elsevier, pp. 453-550, 2004
- [Dewaele 04] G. Dewaele, F. Devernay et R. Horaud, "Hand Motion for 3D point trajectories and a smooth surface model", Poster I, *Lecture Notes in Computer Science, Computer Vision - ECCV 2004: 8th European Conference on Computer Vision, Volume 3021 / 2004*, pp. 495-507, Prague, République Tchèque, 2004, ISBN: 3-540-21984-6, Publisher : Springer Berlin / Heidelberg, 2004
- [Dobkin 90] D. Dobkin et D. Kirkpatrick, « Determining the separation of preprocessed polyhedra – A unified approach », *Lectures notes in Computer Science*, volume 443, Springer Verlag, 1990
- [Dotson 97] B. Dotson, H. Hsiao, S.S. Chiou et J.J. Dipasquale, "Occupational Traumatic Injury Prevention Using Virtual Reality", *National Occupational Injury Research Symposium, NOIRS Symposium, Morgantown, USA, 1997*
- [Dubois 04] F. Dubois, « Dynamique non régulière des systèmes de corps rigides ou peu déformables », Cours n°10 de DEA, Université des Sciences et Techniques du Languedoc, Montpellier, 2002
- [Duidam 01] V. Duidam et S. Stramigioli, "Variable Spatial Springs for Robot Control Applications", dans les Actes de IEEE/RSJ International Conference on Intelligent Robots and Systems, Maui, USA, 2001
- [Duriez 04] C. Duriez, « Contact frottant entre objets déformables dans des simulations temps réel avec retour haptique », Thèse de doctorat de l'Université d'Evry, 2004
- [Eberly 98] David Eberly, "Dynamic Collision Detection using Oriented Bounding Boxes", <http://www.cs.uofs.edu/~emg5/coldet/doc3.pdf>.
- [ElKoura 03] G. ElKoura et K. Singh, "Handrix : Animating the Human Hand", dans les Actes de Eurographics / ACM SIGGRAPH Symposium on Computer Animation, San Diego, USA, 2003
- [FAA 00] FAA, "Chapter 12: Facilities Safety", *FAA System Safety Handbook*, décembre 2000
- [Fasse 98a] E.D.Fasse et P.C.Breedveld, "Modeling of elastically Coupled bodies : Part 1 – General Theory and Geometric Potential Function Method", dans *Transactions of the ASME*, Vol. 120, pp. 496-506, 1998
- [Fasse 98b] E.D.Fasse et P.C.Breedveld, "Modeling of elastically Coupled bodies : Part 2 – Exponential and Generalized coordinate methods", dans *Transactions of the ASME*, Vol. 120, pp. 496-506, 1998
- [Fasse 99] E.D. Fasse et J.F. Broenink, « A spatial Impedance Controller for Robotic Manipulation », *IEEE Transactions on Robotics and Automation*, Vol 13, no 4, pp. 546-556, ISSN 1042-296X
- [Fisher 01] S. Fisher et M.C. Lin, "Fast penetration depth estimation for elastic bodies using deformed distance fields", dans les Actes d'IROS'01, *International Conference on Intelligent Robots and Systems*, vol.1, pp. 330-336, 2001

- [Forsey 91] D.R. Forsey, "Character animation using hierarchical B-splines", dans les Actes du 2nd EUROGRAPHICS Workshop on Animation and Simulation, pp. 155-170, 1991
- [Fuchs 01] P. Fuchs, G. Moreau et J.P. Papin, Le traité de la réalité virtuelle, première édition, les Presses de l'Ecole des Mines de Paris, 530 p., novembre 2001
- [Fuhrmann 03] A. Fuhrmann, G. Sobotka et C. Gross, "Distance fields for rapid detection in physically-based modeling", Dans les Actes de Graphicon'03, pp. 58-65, Moscou, Russie, 2003
- [Fung 93] Y.C. Fung, « Skeletal muscle », Biomechanics : Mechanical Properties of living tissues, Berlin, Springer-Verlag, 1993
- [Garrigues 02] J. Garrigues, « Initiation à la méthode des éléments finis », Cours de l'EGIM, 2002, <http://jgarrigues.perso.egim-mrs.fr/index.html>
- [Gartner 99] B. Gärtner, "Fast and Robust Smallest Enclosing Balls.", dans les Actes de 7th Annual European Symposium on Algorithms, Lecture Notes in Computer Science 1643, Springer, pp. 325-338, ISBN 3-540-66251-0, 1999
- [Gibson 97] S.F.F. Gibson et B. Mirtich, "A survey of deformable modelling in Computer Graphics", TR-97-19, MERL, novembre 1997
- [Gilbert 88] E. G. Gilbert, D. W. Johnson et S. S. Keerthi, "A Fast Procedure for Computing the Distance between Objects in three-Dimensional Space", dans les Actes de IEEE International Conference on Robotics and Automation, 4, pp.193-203, 1988
- [Glocker 95] C. Glocker et F. Pfeiffer, « Multiple impacts with friction in rigid multibody systems », Non linear Dynamics, 7 : 471-497, Kluwer Academic Publishers, 1995
- [Gottschalk 96] S. Gottschalk, M. Lin et D. Manocha, "OBB-Tree: A Hierarchical Structure for Rapid Interference Detection", Dans les Actes de ACM SIGGRAPH, pp. 171-180, 1996.
- [Gourret 89] J-P. Gourret, N. Magnenat Thalmann et D. Thalmann, « Simulation of Object and Human skin deformations in a grasping task », dans les Actes de SIGGRAPH'89, Computer Graphics, vol. 23, n°3, pp. 21-30, 1989
- [Granjon 01] Y. Granjon, Automatique – Systèmes linéaires, non linéaires, à temps continu, à temps discret, representation d'état, Editions Dunod, 381p., ISBN 2 10 005399 X, 2001
- [Guigue 03] P. Guigue, O. Devillers, "Fast and Robust Triangle-Triangle Overlap Test using Orientation Predicates", Journal of Graphics Tools, 8 (1), 2003
- [Guillevic 91] Ch. Guillevic, Psychologie du travail, Éditions Nathan, collection Fac Psychologie, Paris, 1991, 225p
- [Gulati 97] R.J. Gulati et M.A. Srinivasan, "Determination of Mechanical Properties of the Human Fingerpad, In Vivo, Using a Tactile Stimulator", Touch Lab Report 3, RLE TR-605, MIT, Cambridge, 1997
- [Hasan 02] R. Hasan, "Contribution à l'amélioration des performances des systèmes complexes par la prise en compte des aspects socio-techniques dès la conception : proposition d'un modèle original de situation de travail pour une nouvelle approche de conception », Thèse de Doctorat, Proposition Automatisée, Université Henri Poincaré, Nancy I, NST n° 222, 227p., 2002
- [Hill 38] A.V.Hill, "The heat of shortening and the dynamic constants of muscle", pp.136-195, Dans les Actes de the Royal Society of London, B. 126, 1938
- [Hoffman 01] J.D. Hoffman, Numerical methods for engineers and scientists, 2nde Edition, Marcel Dekker, 823p., ISBN 0-8247-0443-6, 2001
- [Hong Kong 99] "A Guide to Safety Management", Occupational Safety and Health Branch, Labour Department, Honk Kong, 1999, <http://www.info.gov.hk/labour>
- [Hsu 92] W. Hsu, J.F. Hugues, H. Kaufman, "Direct manipulation of Free Form Deformations", dans les Actes de SIGGRAPH'92, pp.177-184, 1992
- [Huang 96] Z. Huang, "Motion control for Human animation", Thèse de Doctorat n° 1601, 109p., Laboratoire d'Informatique Graphique, Département d'informatique, EPFL, 1996

- [Hubbard 96] P. M. Hubbard, "Approximating Polyhedra with Spheres for Time-Critical Collision Detection", *ACM Transactions on Graphics*, 15 (3), pp. 179-210, 1996
- [Hui 02] K.C. Hui, et N.N. Wong, "Hands on a virtually elastic object", *The Visual Computer*, Vol. , pp. 150-163, 2002
- [Huxley 57] A. F. Huxley, "Muscle structure and theories of contraction", *Prog. Biophysics and Biophysical Chemistry* 7: 255-318, 1957
- [Hyncik 02] L. Hyncik, "Biomechanical Model of Abdominal Organs and Tissues for Car Crash Tests Purposes", Thèse de Doctorat, Faculté des Sciences Appliquées, Université de West Bohemia à Pilsen, 2002
- [Inrs 04a] INRS, "Presses plieuses hydrauliques pour le travail à froid des métaux. Amélioration de la sécurité des machines en service dans le cadre de leur rénovation. Spécifications techniques à l'usage des utilisateurs, des préventeurs et des rénovateurs", ED 927, 64 p, ISBN 2-7389-1229-X, juillet 2004
- [Inrs 04b] « La main et la machine », Editions INRS ED940, 1ère édition décembre 2004, ISBN 2-7389-1267-26
- [James 01] D.L. James et D.K. Pai, « A unified treatment of elastostatic contact simulation for real-time Haptics », *Haptics-e, The Electronic Journal of Haptics Research*, Vol. 2, n°1, septembre 2001
- [James 02] D.L. James et D.K. Pai, "Real-time simulation of multizone elastokinematic models", dans les Actes de IEEE International Conference on Robotics and Automation, Washington DC, USA, 2002
- [Jean 01] M. Jean, V. Acary et Y. Monerie, « Non Smooth contact dynamics approach of cohesive materials », *Transactions of the Royal Society of London, A* (2001) 359, pp. 2497-2518, 2001
- [Jean 99] M. Jean, "The Non Smooth Contact Dynamics method" dans *Computational Modeling of Contact and Friction*, un numéro special de *Computer Meth. in Appl. Mech. and Engng.*, 177, pp. 235-257, J.A.C. Martins & A. Klarbring, eds., 1999
- [Jindrich 03] D.L. Jindrich, Y. Zhou, T. Becker, J. Tigh Dennerlein, "Non-linear viscoelastic models predict fingertip pulp force-displacement characteristics during voluntary tapping", *Journal of Biomechanics* 36, pp. 497-503, 2003
- [Joukhadar 96] A. Joukhadar, A. Wabbin C. Laugier. « Fast Contact Localisation between Deformable Polyhedra in Motion », Dans les Actes de CGS Conference on Computer Animation, p. 126, 1996
- [Kapandji 05] A.I.Kapandji, *Physiologie articulaire*, Tome 1, 6ème édition, Editions Maloine, ISBN 2-224-02647-1, 2005
- [Khale 94] W. Khale, H. Leonhardt et W. Platzer, *Anatomie 1 : Appareil locomoteur*, Edition française rédigée par C. Cabrol, Flammarion Médecine-Sciences, 1994
- [Kry 02] P.G. Kry, D.L. James et D.K. Pai, « EigenSkin : Real Time large deformation character skinning in hardware », dans les Actes de ACM SIGGRAPH Symposium on Computer Animation, pp. 153-159, San Antonio, USA, 2002
- [Kuch 95] J.J. Kuch et T.S. Huang, "Vision based hand modelling and tracking for virtual teleconferencing and telecollaboration", dans les Actes de IEEE International Conference on Computer Vision, pp. 666-671, Cambridge, USA, 1995
- [Laitinen 97] H. Laitinen, "Usability through Product Design: New Design Methods and Industrial Applications", Research project, Dept. of Occupational Safety, FIOH, Vantaa, Finland, 1997
- [Landsmeer 55] J.M.F.Landsmeer, "Anatomical and functional investigations on the articulations of the human fingers", *Acta anatomica*, suppl. 25, pp. 1-69, 1955

- [Larsson 01] T. Larsson et A. Möller, « Collision detection for continuously deforming bodies », Dans les Actes de EUROGRAPHICS'01 Conference, Computer Graphics Forum, volume 20(3), Manchester, 2001
- [Lewis 00] J.P. Lewis, M. Cornder et N. Fong, « Pose Space Deformation : A unified approach to shape interpolation and skeleton-driven deformation », dans les Actes de SIGGRAPH'00, ACM Computer Graphics, pp. 165-172, New Orleans, USA, 2000
- [Lin 91] M. C. Lin et J. F. Canny, "A Fast Algorithm for Incremental Distance Calculation", Dans les Actes de IEEE International Conference on Robotics and Automation, vol. 2, pp. 1008-1014, ISBN: 0-8186-2163-X, Sacramento, USA, 1991
- [Lin 96] M.C. Lin, D. Manocha, J. Cohen et S.Gottschalk, « Collision Detection : algorithms and applications », dans Algorithms for Robot Motion and Manipulation, pp. 129–142, eds. J-P. Laumond, M. Overmars et A.K. Peters (Invited Submission), 1996
- [Lin 00] J. Lin, Y. Wu et T.S. Huang, "Modeling the constraints of human hand motion", dans les Actes du Workshop on Human Motion, pp. 121-126, 2000
- [Lloyd 05] J.E. Lloyd, « Fast implementation of Lemke's algorithm for rigid body contact simulation », dans les Actes de IEEE International Conference on Robotics and Automation, ICRA'05, Barcelone, Espagne, 2005
- [Longchamp 06] R. Longchamp, Commande numérique de systèmes dynamiques – Cours d'Automatique, 2^{de} Edition, Presses Polytechniques et Universitaires Romandes, 765p., ISBN 2-88074-642-6, 2006
- [Lund 04] J. Lund et L.E. Aarø, "Accident prevention. Presentation of a model placing emphasis on human, structural and cultural factors", Safety Science 42 (2004) 271–324, 2004
- [Määttä 03] T. Määttä, "Virtual environments in machinery safety analysis", Thèse de Doctorat, VTT Publications 516, 2003
- [Magenat-Thalmann 88] N. Magneat-Thalmann, R. Laperrière et D. Thalmann, « Joint-dependent Local Deformation for hand animation and object grasping », dans les Actes de Graphics Interface, pp.26-33, Edmonton, Canada, 1988
- [Magenat-Thalmann 90] N. Magneat Thalmann et D. Thalmann, "Human body deformations using joint-dependent local operators and Finite Element theory", dans Making them move, (N. Badler, B.A. Barsky, D. Zelter, eds), Morgan Kaufmann, San Mateo, USA, pp. 243-262, 1990
- [Marc 06] J. Marc, N. Belkacem et J. Marsot, "VR, a design tool to better take usability into account - Elements of validation", dans les Actes de 9th international symposium of the ISSA Research Section "Design process and human factors integration: Optimising company performance", Nice, France, 1-3 March 2006
- [Marie 05] F. Marie, "Préhension d'objets virtuels", Rapport de stage, IMERIR, 2005
- [Marsot 04] J. Marsot, J. Ciccotelli et F. Gardeux, "Virtual Machine for Risk Prevention", dans les Actes du 35th Int. Symp. On Robotics, ISR'2004, Paris, France, 2004
- [Marsot 05] J. Marsot et J. Ciccotelli, "Dynamical indexation of risk in a virtual work environment", dans les Actes de Virtual Concept 2005, Biarritz, France, novembre 2005
- [Mas Sanso 94] R. Mas Sanso, "A hand control and automatic grasping system for synthetic actors", dans les Actes de ACM EUROGRAPHICS'94, pp. 167-178, Oslo, Norvege, 1994
- [Mc Donal 01] J. McDonald, J. Toro, K. Alkoby, A. Berthiaume, R. Carter, P. Chomwong, J. Christopher, M.J. Davidson, J. Furst, B. Konie, G. Lancaster, L. Roychoudhuri, E. Segdwick, N. Tomuro et R. Wolfe, "An improved articulated model of the human hand", The Visual Computer, 17: 158-166, Springer Verlag, 2001

- [Meseure 03] Ph. Meseure, A. Kheddar et F. Faure, « Détection des collisions et Calcul de la réponse », Rapport de l'Action Spécifique CNRS n°90, 17 décembre 2003
- [Mirtich 94] Brian Mirtich et John Canny, "Impulse-based dynamic simulation", dans les actes du Symposium on Interactive 3D Graphics, 1994, eecs.berkeley.edu
- [Mirtich 98] B. Mirtich, "V-Clip: Fast and Robust Polyhedral Collision Detection", ACM Transactions on Graphics, 17 (3), pp. 177-208, 1998
- [Mirtich 00] B. Mirtich, "Timewarp rigid body simulation", dans les actes de ACM SIGGRAPH'00, Computer Graphics, pp. 193-200, New Orleans, USA, 2000
- [Moccozet 96] L. Moccozet, « Hands Modeling and Animation for Virtual Humans », Thèse de Doctorat, Université de Genève, 1996
- [Moccozet 97] L. Moccozet et N. Magnenat Thalmann, "Dirichlet Free Form Deformation and their application to hand simulation", Computer Animation '97, p. 93, 1997
- [Mohammadi 03] B. Mohammadi et J-H. Saïac, Pratique de la simulation numérique, Editions Dunod, Industries et Technologies, 420p., ISBN 2-10-006407-X
- [Möller 97] T. Möller, "A Fast Triangle-Triangle Intersection Test", Journal of Graphics Tools, 2 (2), pp. 25-30, 1997
- [Moore 88] M. Moore et J. Wilhelms, "Collision detection and response for computer animation", dans les Actes de ACM SIGGRAPH'88, Computer Graphics, 22(4), pp. 289-298, Atlanta, USA, 1988
- [Moreau 88] J-J. Moreau, "Unilateral contact and dry friction in finite freedom dynamics", Nonsmooth Mechanics and Applications, CISM Courses and Lectures, Vol. 302, pp. 1-82, Springer-Verlag, New York, J-J. Moreau & P. D. Panagiotopoulos, eds., 1988
- [Moreau 03a] J-J; Moreau, "Modélisation et simulation de matériaux granulaires », dans les Actes du 35ème congrès national d'analyse numérique, B. Mohammadi, ed., juin 2003
- [Moreau 03b] J-J. Moreau, "Indétermination liée au frottement sec dans le calcul des granulats", dans les Actes du 6ème Colloque National en Calcul des structures, vol.3, pp. 465-472, ISBN 2-7302-1034-2, M. Potier-Ferry, M. Bonnet, A. Bignonnet, coordonnateurs, 2003
- [Mukundakrishnan 04] B. Mukundakrishnan, "Simulation of multi-rigid body Dynamics with frictional contact", Rapport technique, Université de Pennsylvanie, Philadelphie, mai 2004
- [Nasios 01] Konstantinos Nasios, "Improving Chemical Plant Safety Training using Virtual Reality", Thèse de Doctorat, School of Chemical, Environmental, and Mining Engineering, Université de Nottingham, 2001
- [Nedel 00] L. Porcher Nedel et D. Thalmann, "Anatomic modelling of deformable human bodies", The Visual Computer (2000) 16: 306-321, Springer-Verlag, 2000
- [Nichols 02] S. Nichols et H. Patel, "Health and safety implications of Virtual Reality : a review of empirical evidence", Applied Ergonomics, vol. 33, pp. 251-271, 2002
- [Nirei 96] K. Nirei, H. Saito, M. Mochimaru et S. Ozawa, « Human hand tracking from binocular image sequences », IEEE IECON –PROCEEDINGS, NUMBER 1, pp. 297-302, 1996
- [Nölker 99] C. Nölker et H. Ritter, "GREFIT : Visual recognition of hand postures", dans les Actes de International Gesture Workshop, GW'99, Springer Verlag, LNAI 1739, pp.61-72, Gif-sur-Yvette, France, 1999
- [Ouhaddi 98] H.Ouhaddi et P. Horain, "Conception et ajustement d'un modèle 3D articulé de la main", 6ème GT RV, mars 1998
- [Pang 96] J-S. Pang et J.C. Trinkle, "Complementarity formulations and existence of solutions of dynamic multi-rigid contact problems with Coulomb friction", Mathematical Programming, cs.unc.edu, 1996

- [Pavlovic 97] V.I. Pavlovic, R. Sharma et T.S. Huang, “Visual interpretation of hand gestures for Human-computer interaction : A review”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, n°7, Juillet 1997
- [Pawluk 97] D.T.V. Pawluk, “A viscoelastic model of the human fingerpad and a Holistic model of Human touch”, Thèse de Doctorat, Harvard University, UMI number : 9733376, Mai 1997
- [Pawluk 99a] D.T.V. Pawluk et R.D. Howe, “Dynamic lumped element response of the human Zingerpad”, Journal of Biomechanical Engineering—Transactions of the ASME 121, 178–183, 1999
- [Pawluk 99b] D.T.V. Pawluk et R.D. Howe, “Dynamic contact of the human fingerpad against a flat surface”, Journal of Biomechanical Engineering, Transactions of the ASME, vol. 121, pp. 605-611, 1999
- [Philipps 81] J.R. Philipps et K.O. Johnson, “Tactile spatial resolution III : a continuum mechanics model of skin predicting mechanoreceptor responses to bars, edges and gratings”, Journal of Neurophysiology, 46(6) : 1204-1225, 1981.
- [Pinczon 04] Y. Pinczon du Sel, “La réalité virtuelle pour la conception sûre d'équipements », Mémoire de fin d'études, Rapport n° IET - S / 04DT – 084 – YP
- [Pollard 05] N.S. Pollard et V.B. Zordan, “Physically-based Grasping Control from example”, dans les Actes de Eurographics / ACM SIGGRAPH Symposium on Computer Animation, Los Angeles, USA, 2005
- [Ponamgi 95] M.K. Ponamgi, D. Manocha et M.C. Lin, “Incremental algorithms for collision detection between solid models”, ACM/SIGGRAPH Symposium on Solid Modelling, pp. 293-304, 1995
- [Pouliquen 05a] M. Pouliquen, C. Duriez, C. Andriot, A. Bernard, L. Chodorge et F. Gosselin, « Real Time Finite Element Finger Pinch Grasp Simulation », dans les Actes de IEEE World Haptics Conference, WHC'05, pp. 323-328, Pise, Italie, 18 – 20 mars 2005
- [Pouliquen 05b] M. Pouliquen, Y. Pinczon du Sel, A. Bernard, J. Marsot et L. Chodorge, “Human-Computer Interactions with Motion Capture for Virtual Reality Applications », dans les Actes de Virtual Concept 2005, Biarritz, France, novembre 2005
- [Provot 95] X. Provot, “Deformation Constraints in a Mass-Spring Model to describe rigid cloth behaviour”, Dans les actes de Graphics Interface'95, pp. 147-154, Québec, Canada, 1995
- [Provot 97] X. Provot, “Collision and self-collision handling in cloth model dedicated to design garments”, Dans les Actes de Graphics Interface'97 Conference, pp. 177-189, Kelowna, Canada, 1997.
- [Quinlan 94] S. Quinlan, “Efficient distance computation between non-convex objects”, dans les Actes de IEEE International Conference on Robotics and Automation, pp. 3324-3329, San Diego, USA, 1994
- [Raghupathi 03] L. Raghupathi, V. Cantin, F. Faure et M-P. Cani, “Real-time simulation of self-collisions for virtual intestinal surgery”, Dans les Actes de International Symposium on Surgery Simulation and Soft Tissue Modelling, Volume 2673 / 2003, pp. 15-26, Juan-les-Pins, France, ISBN: 3-540-40439-2, 2003
- [Ravani 02] B. Ravani, “Lie Groups and Computer Graphics”, dans les actes du Workshop on Geometric Modelling and Animation, Fondation on Computational Mathematics, FOCM'02, Minneapolis, USA, 2002
- [Redon 01] S. Redon, A. Kheddar et S. Coquillart, “Contact : In-between motions for collision detection”, IEEE International Workshop on Robot and Human Interactive Communication (ROMAN), Bordeaux, France, 2001
- [Redon 04] S. Redon, « Continuous Collision Detection for Rigid and Articulated Bodies », ACM SIGGRAPH Courses Notes, SIGGRAPH'04, Los Angeles, USA, 2004

- [Rehg 93] J.M. Rehg et T. Kanade, « DigitEyes : Vision-based human hand tracking », Rapport Technique CMU-CS-93-220, School of Computer Sciences, Université de Carnegie Mellon, USA, 1993
- [Renouf 04] M. Renouf, « Optimisation numérique et calcul parallèle pour l'étude des milieux divisés bi- et tri-dimensionnels », Thèse de Doctorat de l'Université des Sciences et Techniques du Languedoc, Montpellier II, 2004
- [Rezzonico 95] S. Rezzonico, R. Boulic, Z. Huang, N. Magnenat Thalmann et D. Thalmann, "Consistent Grasping interactions with virtual actors based on the multi-sensor hand model", dans les Actes du Second EUROGRAPHICS Workshop on Virtual Environments, Monte-Carlo, 1995
- [Rijkema 91] H. Rijkema et M. Girard, "Computer animation of knowledge-based human grasping", Computer Graphics, Volume 25, n°4, pp.339 – 348, Juillet 1991
- [Rodriguez 03] R. Rodriguez, « Détection de collision et localisation de contact des polyèdres déformables en temps réel dans un environnement virtuel », Rapport de DEA, équipe SHARP, GRAVIR, INRIA Rhône Alpes, 2003
- [Rouvière 02] H. Rouvière et A. Delmas, Anatomie humaine descriptive, topographique et fonctionnelle, Tome 3 : Membres, Système nerveux central, 15ème édition, Editions MASSON, ISBN : 2294003934, 2002
- [Ruspini 97] D.C. Ruspini et O. Khatib, « Collision/Contact Models for the dynamic simulation of complex environments », dans les Actes de IEEE IROS'97, International Conference on Intelligent Robots and Systems, Grenoble, France, 1997
- [Sagot 03] J-C. Sagot, V. Gouin et S. Gomes, « Ergonomics in product design : safety factor », Safety Science, vol. 41, pp. 137-154, 2003
- [Salençon 05] J. Salençon, Mécanique des milieux continus : Tome 1, Concepts généraux, 370 pages, ISBN : 2730212450, 2005
- [Sauer 98] J. Sauer et E. Schömer, "A constraint based approach to rigid body dynamics for virtual reality applications", dans les Actes de ACM Symposium on Virtual Reality Software and Technology, pp. 153–161, Taipei, Taiwan, 1998
- [Savescu 04] A-V. Savescu, L. Cheze, X. Wang, G. Beurier et J-P. Verriest, "A 25 Degrees of Freedom Hand Geometrical Model for better hand attitude simulation", 2004-01-2196, SAE International, 2004
- [Sederberg 86] T.W. Sederberg, and S. R.Parry, "Free-Form Deformation of solid geometric models", Computer Graphics, volume 20 (4), pp. 151-160, 1986
- [Serina 98] E. Serina, E. Mockensturm, C.J. Mote, D. Rempel, "A structural model of the forced compression of the fingertip pulp", Journal of Biomechanics, 31: 639–46, 1998
- [Shaikh 04] I. Shaikh, U. Jayaram, S. Jayaram et C. Palmer, "Participatory Ergonomics using VR integrated with Analysis Tools", dans les Actes de Winter Simulation Conference, pp. 1746-1754, 2004
- [Sibille 02] L. Sibille, M. Teschner, S. Srivastav, et J-C. Latombe, "Interactive Simulation of the human Hand", dans les Actes de 6ème conférence annuelle de Int. Society for Computer Aided Surgery, 16th Int. Congress of Computer Aided Radiology and Surgery (CARS), Paris, France, 2002
- [Singh 97] U. Singh, "Temporal Characteristics of the Human Finger", Rapport de Master, Dept. of EE&CS, mai 1997
- [Singh 00] K. Singh et E. Kokkevis, "Skinning Characters using Surface-Oriented Free-Form Deformations", dans les Actes de Graphics Interface 2000, pp. 35-42, 2000
- [Song 02] P. Song, "Modeling, Analysis and Simulation of Multibody Systems with Contact and Friction", Thèse de Doctorat de l'Université de Pennsylvanie, 2002

- [Srinivasan 89] M.A. Srinivasan, "Surface deflection of primate fingertip under line load", *Journal of Biomechanics*, vol. 22, n° 4, pp. 343-349, 1989
- [Stramigioli 01a] S. Stramigioli et H. Bruyninckx, "Geometry and Screw Theory for Robotics", Tutorial T9, dans les Actes de IEEE International Conference on Robotics and Automation, ICRA'01, Séoul, Corée, 2001
- [Stramigioli 01b] S. Stramigioli et H. Bruyninckx, « Modeling and Control of rigid mechanical systems », Cours, Dutch Institute of Systems and Control (DISC), 2001
- [Stroeve 97] S. Stroeve, "Impedance characteristics of a neuromusculoskeletal model of the human arm", Dans *Posture control, Man-Machine Systems and Control group*, Department of Mechanical Engineering, Delft, University of Technology, The Netherlands, octobre 1997
- [Terzopoulos 87] D. Terzopoulos, J.C. Platt, et A.H. Barr, « Elastically deformable models », dans les Actes de ACM SIGGRAPH'87, *Computer Graphics*, 21, pp. 205-214, Anaheim, USA, 1987
- [Terzopoulos 88] D. Terzopoulos et K. Fleisher, « Deformable models », *Visual Computer*, 4, pp. 306-331, 1988
- [Terzopoulos 90] D. Terzopoulos et K. Waters, « Physically-based facial modeling, analysis and animation », *Journal of Visualization and Computer Animation*, 1 : 73-80, 1990
- [Teschner 03] M. Teschner, B. Heidelberger, M. Mueller, D. Pomeranets et M. Gross, « Optimized spatial hashing for collision detection of deformable objects », Dans les Actes de *Vision, Modeling, Visualization Conference, VMV'03*, pp. 47-54, Munich, Allemagne, 2003
- [Teschner 04] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrmann, M-P. Cani, F. Faure, N. Magnenat Thalmann, W. Strasser et P. Volino, « Collision detection for deformable objets », *Etat de l'art, Eurographics'04*, Grenoble, France, 2004
- [Teschner 05] M. Teschner, B. Heidelberger, D. Manocha, N. Govindaraju, G. Zachmann, S. Kimmerle, J. Mezger et A. Fuhrmann, "Collision Handling in Dynamic Simulation Environments", *Eurographics Tutorial, Eurographics'05*, Dublin, Ireland, 2005
- [Tubiana 80] R. Tubiana, *Traité de la chirurgie de la main – Tome 1 : Anatomie et physiologie*, Editions Masson, ISBN 2-225-63998-1, 789 p., 1980
- [Tubiana 90] R. Tubiana et J-M. Thomine, *La main : Anatomie fonctionnelle et examen clinique, Abrégés Masson*, ISBN 2-225-81962-9, 210p., 1990
- [Van den Bergen 97] G. Van den Bergen, "Efficient Collision Detection of Complex Deformable Models using AABB Trees", *Journal of Graphics Tools*, 2(4): 1-13, 1997
- [Van den Bergen 99] G. Van den Bergen, "A Fast and Robust GJK Implementation for Collision Detection of Convex Objects", *Journal of Graphic Tools*, Volume 4, Issue 2, pp. 7 – 25, ISSN: 1086-7651, 1999
- [Vardy 98] A. Vardy, "Articulated human hand model with inter-joint dependency constraints", *Rapport Technique, Computer Science 6752*, décembre 1998
- [Volino 00] P. Volino et N. Magnenat Thalmann, "Accurate Collision Response on polygonal meshes", *Computer Animation Conference*, Philadelphie, USA, 2000
- [Wan 04] H. Wan, S. Gao, Q. Peng, G. Dai et F. Zhang, "MIVAS : A multi-modal immersive virtual assembly system", dans les Actes de *DETC'04, ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, DETC2004-57660*, Salt Lake City, USA, octobre 2004
- [Weiss 98] P. Weiss et A.S. Jessel, "Virtual reality applications to Work", *Work*, Vol. 11(3), pp. 277-293, 1998
- [Winter 90] D.A. Winter, *Muscle Mechanics*, in *Biomechanics and Motor Control of the Human Movements*, 2nd edition, New York : Wiley, 1990
- [Wu 01] Y. Wu, "Vision and learning for intelligent Human-Computer interaction", *Thèse de Doctorat, University of Illinois, Urbana-Champaign*, 2001

- [Wu 02] J.Z. Wu, R.G. Dong, S. Rakheja, A.W. Schopper et W.P. Smutz, “Effects of static compression on the vibration modes of a fingertip”, *Journal of Low Frequency Noise, Vibration and Active control*, Vol. 21, n°4, pp. 229-243, 2002
- [Wu 04] J.Z. Wu, R.G. Dong, S. Rakheja, A.W. Schopper et W.P. Smutz, “A structural fingertip model for simulating of the biomechanics of tactile sensation”, *Medical Engineering & Physics* 26, pp. 165–175, 2004
- [Younblut 96] C. Younblut, R.E. Johnson, S.H. Nash, R.A. Wienclaw et C.A. Will, “Review of Virtual Environment Interface Technology”, IDA Paper P-3186, Institute for Defense Analyses, 249p., 1996
- [Zachmann 00] G. Zachmann, “Virtual Reality in assembly simulation : Collision detection, Simulation algorithms and Interaction techniques”, Thèse de Doctorat, Université de Bonn, 2000
- [Zajac 89] F.E. Zajac, “Muscle and tendon : properties, models scaling and applications to biomechanics and motor control”, *CRC Critic. Rev. in Biomed. Engineering*, 17 : 359-411, 1989
- [Zülch 05] G. Zülch et T. Grieger, “Modelling of Occupational health and safety aspects in the Digital Factory”, *Computers in Industry*, vol. 56, pp. 384-392, 2005
- [w3DCafé] www.3dcafe.com
- [w3ds] usa.autodesk.com (www.discreet.com)
- [w5dt] www.5dt.com
- [wAlias] www.alias.com
- [wArt] www.ar-tracking.de
- [wAscension] www.ascension-tech.com
- [wBarco] www.barco.com
- [wBlender] www.blender.org
- [wCharnwood] www.codamotion.com
- [wCMLabs] www.cm-labs.com
- [wGid] gid.cimne.upc.es
- [wHaption] www.haption.com
- [wImmersion] www.immersion.com
- [wIntersense] www.isense.com
- [wNorthern] www.ndigital.com
- [wOrigin] orin.com
- [wP5] www.videogamealliance.com/VGA/video_game/P5.php
- [wPolhemus] www.polhemus.com
- [wQualisys] www.qualisys.com
- [wRhino] www.fr.rhino3D.com
- [wVicon] www.vicon.com
- [wStandard] www.viconstandard.org
- [wVirtools] www.virtools.com

PUBLICATIONS PERSONNELLES

Les publications suivantes ont été produites dans le cadre de ce travail :

Publications dans des revues d'audience internationale à comité de lecture ou articles sélectionnés dans des ouvrages de synthèse

- Mahmoud Shahrokhi, Mamy Pouliquen et Alain Bernard, “Human Modeling in Industrial Design”, publié dans *Advances in Design*, Springer Series in Advanced Manufacturing, ElMaraghy, Hoda A.; ElMaraghy, Waguih H. (Eds.), 2006, 595 p., Hardcover, ISBN: 1-84628-004-4 ;
- Mamy Pouliquen, Alain Bernard, Jacques Marsot et Laurent Chodorge, “Virtual Hands and Virtual Reality Multimodal Platform to design safer industrial systems”, acceptée pour publication dans *Computers in Industry*, Elsevier ➔ sous presse

Communications à des congrès internationaux à comité de sélection et actes publiés

- Mamy Pouliquen, Christian Duriez, Claude Andriot, Alain Bernard, Laurent Chodorge et Florian Gosselin, « Real Time Finite Element Finger Pinch Grasp Simulation », IEEE World Haptics Conference, Pise, Italie, 18 – 20 mars 2005, pp. 323-328 ;
- Mamy Pouliquen, Alain Bernard, Joseph Ciccotelli et Laurent Chodorge, “An application of Virtual Reality in the Design Process for Accident Prevention”, CIRP Design Seminar, Shanghai, Chine, 22 – 25 mai 2005, pp. 426-432 ;
- Mamy Pouliquen, Yann Pinczon du Sel, Alain Bernard, Jacques Marsot et Laurent Chodorge, “Human-Computer Interactions with Motion Capture for Virtual Reality Applications”, Virtual Concept, Biarritz, France, 8 – 10 novembre 2005 ;
- Mamy Pouliquen, Alain Bernard et Jacques Marsot, “Virtual Hands for Risk Prevention integration in Human-Computer Interactions”, IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems, La Corogne, Espagne, 10 – 12 juillet 2006.

ANNEXES

A.1. Compléments sur les caractéristiques de la main

Dans cette annexe, nous apportons quelques informations complémentaires sur la description de la main humaine faite dans le Chapitre I.

1. Le squelette de la main

Ce squelette est une structure arborescente composée de vingt-sept os possédant vingt-sept degrés de liberté avec six degrés de liberté pour le poignet et vingt-et-un degrés de liberté pour les articulations des doigts (Cf. Figure A1.132) (d'après *Biomécanique et physiologie du mouvement*, de S. Bouisset, 2002).

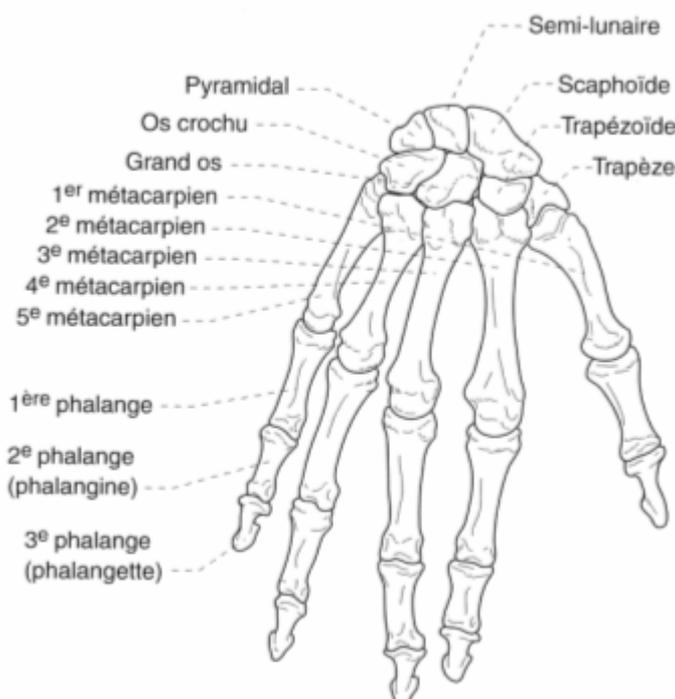


Figure A1.132 : Les os de la main

Nous décrivons tout d'abord le poignet puis les doigts de la main.

1.1 Le poignet

Le poignet comporte huit os courts et les deux extrémités (épiphyses) de l'ulna (ou cubitus) et du radius. Les os sont disposés en deux rangées horizontales superposées de quatre os chacune (Cf. Figure A1.133) (d'après *Les feuillets d'anatomie*, de J. Brizon et J. Castaing, 1999) :



Figure A1.133 : Les os du poignet

La première rangée, appelée rangée proximale ou rangée antibrachiale, s'articule au squelette de l'avant-bras. Elle est formée du scaphoïde, du semi-lunaire, du pyramidal et du pisiforme. La deuxième rangée, appelée rangée distale ou rangée métacarpienne, s'articule aux métacarpiens. Elle est formée par le trapèze, le trapézoïde, le grand os et l'os crochu.

Le poignet a six degrés de liberté. Les mouvements de flexion/extension et inclinaison latérale s'effectuent au niveau des articulations radio-carpiennes, medio-carpiennes et carpo-métacarpiennes (Cf. Figure A1.134) (d'après *Anatomie 1 : Appareil locomoteur*, de W. Khale, H. Leonhardt et W. Platzer, 1994) :

- la radio-carpienne est l'articulation de la flexion : $80^\circ - 90^\circ$.
- la medio-carpienne est l'articulation de l'extension : 50° .
- Les inclinaisons latérales sont les suivantes :
 - inclinaison cubitale : 25° ,
 - inclinaison radiale : 25° .

Les mouvements de rotation, autrement appelés mouvements de prono-supination de la main, sont réalisés par les deux os de l'avant-bras.

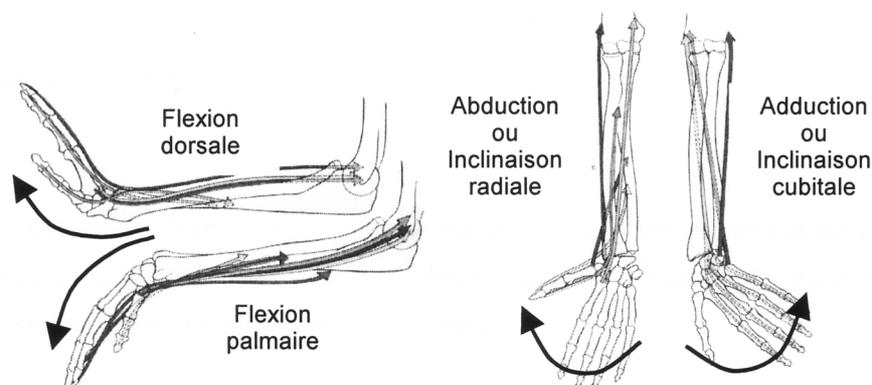


Figure A1.134 : Les mouvements du poignet

1.2 Les doigts

La main est modélisée par cinq chaînes poly-articulaires. Chaque doigt, à l'exception du pouce, est composé de quatre os reliés entre eux par quatre liaisons pivots.

L'articulation de chaque doigt sur son métacarpe est appelée « métacarpo-phalangienne » (MCP). Les articulations entre phalanges sont appelées « interphalangiennes proximales » (IPP) et « interphalangiennes distales » (IPD) pour les quatre doigts. En ce qui concerne le pouce, elles sont appelées « interphalangiennes » (IP). L'articulation située à la base du pouce est dite « carpométacarpienne » (CMC).

1.3 Le pouce

Le pouce, quant à lui, n'est composé que de trois os : un métacarpe et deux phalanges. Il n'a pas de phalange intermédiaire.

Il est plus actif que les autres doigts de la main. L'articulation carpo-métacarpienne (CMC) qui le relie au trapèze autorise deux axes de rotation. La souplesse de cette liaison autorise aussi un mouvement de rotation de faible amplitude qui permet au pouce d'être en opposition par rapport aux autres doigts. Cette dernière est due à des os du carpe et à celle de la base du métacarpe, mais aussi, au relâchement des ligaments les reliant (Cf. Figure A1.135) (d'après, *La main : Anatomie fonctionnelle et examen clinique*, de R. Tubiana et J-M. Thomine, 1990).

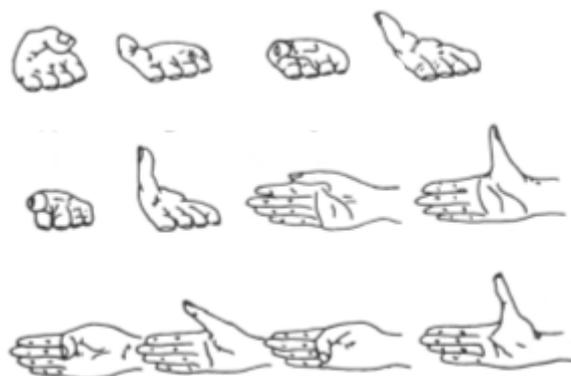


Figure A1.135 : Les mouvements libres du pouce

De même que pour les quatre autres doigts, l'articulation métacarpo-phalangienne (MCP) du pouce a deux degrés de liberté (flexion/extension et très faible amplitude en abduction/adduction), et l'articulation interphalangienne (IPD) n'a qu'un degré de liberté (flexion/extension).

2. Les muscles et les tendons

Le réseau des muscles et tendons qui animent le squelette de la main est un système complexe. Il se décompose en trois groupes :

- les muscles extrinsèques,
- les muscles intrinsèques,
- les muscles du poignet.

La Figure A1.136 présente les muscles et les tendons de la main (d'après *Les feuillets d'anatomie*, de J. Brizon et J. Castaing, 1999). A gauche, sont illustrées les insertions musculaires sur la face dorsale de la main et à droite, ce sont les tendons :

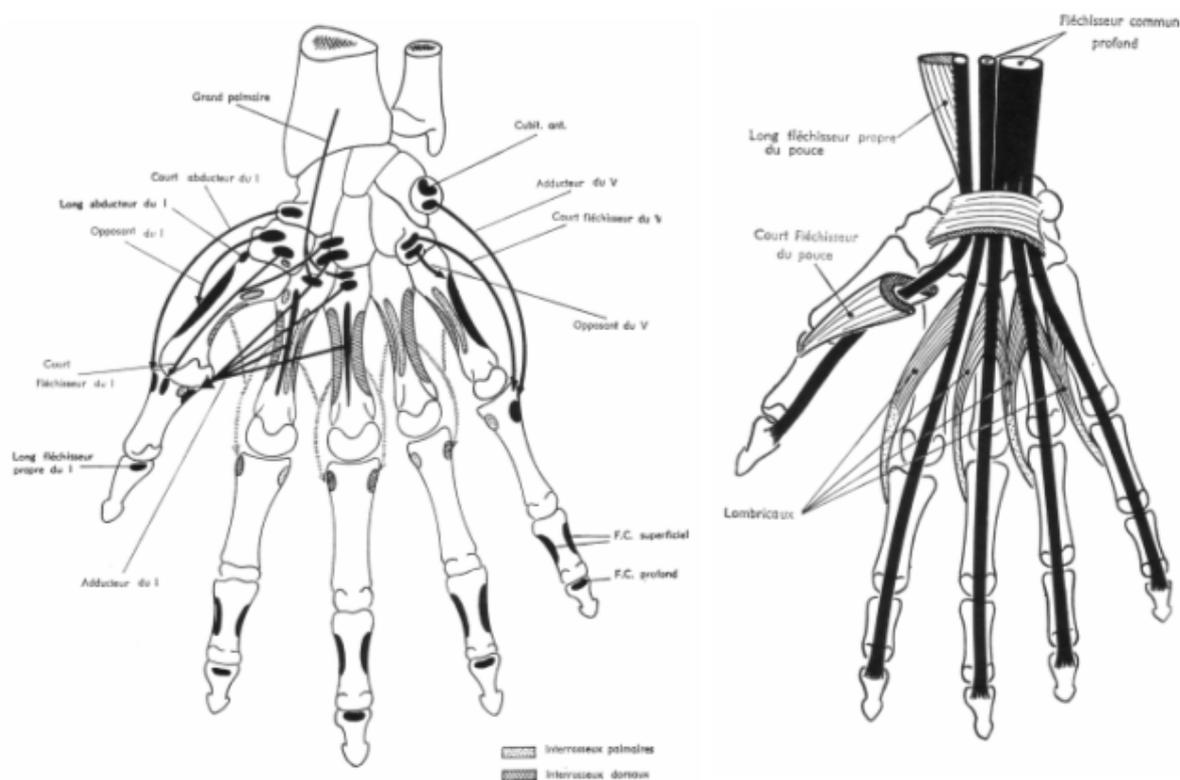


Figure A1.136 : Les muscles et tendons de la main

Nous décrivons les différents groupes musculaires.

2.1 Les muscles intrinsèques

Ces muscles ont à la fois leurs origines et leurs insertions terminales au niveau de la main.

Les muscles intrinsèques qui agissent sur les doigts (sauf le pouce) sont :

- les interosseux dorsaux (D.I.) : quatre muscles,
- les interosseux palmaires (P.I.) : quatre muscles,
- les lombricaux (Lumbrical) : quatre muscles,
- les muscles de l'éminence hypothénar qui agissent sur l'auriculaire : quatre muscles.
- le palmaire cutané,
- l'adducteur (A.D.Q.),
- le court fléchisseur (F.D.Q.),
- et l'opposant (O.D.Q.).

Les muscles du pouce sont appelés muscles de l'éminence thénar. Ils sont au nombre de quatre :

- court abducteur du pouce (A.P.B),
- opposant du pouce (O.P.),

- court fléchisseur du pouce (F.P.B),
- adducteur du pouce (A.P.).

2.2 Les muscles extrinsèques

Ces muscles ont leurs origines dans l'avant-bras et qui s'insèrent dans les phalanges des doigts. Les muscles extrinsèques qui agissent sur les doigts (sauf le pouce) sont :

- l'extenseur commun des doigts (E.D.C.) : quatre muscles,
- l'extenseur propre de l'auriculaire (E.D.Q.) : un muscle,
- le fléchisseur commun superficiel (F.D.S.) : quatre muscles,
- le fléchisseur commun profond (F.D.P.) : quatre muscles,
- l'extenseur propre de l'index (E.I.P.) : un muscle.

Les muscles du pouce sont au nombre de quatre :

- long fléchisseur propre du pouce (F.P.L),
- long abducteur du pouce (A.P.L),
- court extenseur du pouce (E.P.B.),
- long extenseur du pouce (E.P.L.).

Ces muscles interviennent dans le geste en terme de force.

2.3 Les muscles du poignet

Ces muscles permettent d'effectuer les mouvements de rotation, de flexion et d'extension du poignet. Ils sont au nombre de six :

- le petit palmaire (P.L.),
- le grand palmaire (F.C.R.),
- le cubital antérieur (F.C.U.),
- le cubital postérieur (E.C.U.),
- le premier radial (E.C.R.L.),
- le deuxième (E.C.R.B.).

Les muscles du poignet jouent un rôle important dans la saisie d'un objet.

Il est à noter que les muscles extrinsèques du pouce ont une action sur le poignet quand le poing est serré.

Le Tableau TA1.9 présente l'ensemble des muscles de la main ainsi que leur longueur moyenne :

Nom	Abréviation	Longueur fibre neutre (mm)
flexor carpi radialis	FCR	52
palmaris longus	PL	50
flexor digitorum superficialis index	FDS index	72
flexor digitorum superficialis majeur	FDS majeur	70
flexor digitorum superficialis annulaire	FDS annulaire	73
flexor digitorum superficialis auriculaire	FDS auriculaire	70
flexor carpi ulnaris	FCU	42
flexor digitorum profundus index	FDP index	66
flexor digitorum profundus majeur	FDP majeur	66
flexor digitorum profundus annulaire	FDP annulaire	68
flexor digitorum profundus auriculaire	FDP auriculaire	62
flexor pollicis longus	FPL	59
extensor carpi radialis longus	ECRL	93
extensor carpi radialis brevis	ECRB	61
extensor digitorum index	EDC index	55
extensor digitorum majeur	EDC majeur	60
extensor digitorum annulaire	EDC annulaire	58
extensor digitorum auriculaire	EDC auriculaire	59
extensor digiti minimi	EDQ	59
extensor carpi ulnaris	ECU	45
extensor pollicis longus	EPL	57
extensor indicis proprius	EIP	55
abductor pollicis longus	APL	46
extensor pollicis brevis	EPB	43
abductor digiti minimi	ADQ	40
flexor digiti minimi brevis	FDQ	34
opponens digiti minimi	ODQ	15
abductor pollicis brevis	APB	37
flexor pollicis brevis	FPB	36
opponens pollicis	OP	24
adductor pollicis	AP	36
lumbrical index		55
lumbrical majeur		66
lumbrical annulaire		60
lumbrical auriculaire		49
Palmar interosseus I	PI index	15
Palmar interosseus II	PI majeur	17
Palmar interosseus III	PI annulaire	15
Dorsal interosseus I	DI index	25
Dorsal interosseus II	DI majeur	14
Dorsal interosseus III	DI annulaire	15
Dorsal interosseus IV	DI auriculaire	15

Tableau TA1.9 : Liste des muscles de la main

La main est donc un organe complexe modélisé par cinq chaînes poly-articulaires arborescentes et actionnée par une cinquantaine de muscles. Le nombre élevé de degrés de liberté offre plusieurs configurations possibles pour réaliser une même tâche. C'est dans le but de déterminer les muscles à utiliser pour un geste qu'intervient la commande nerveuse. En effet, la coordination des muscles répond à des consignes envoyées par le système sensori-moteur.

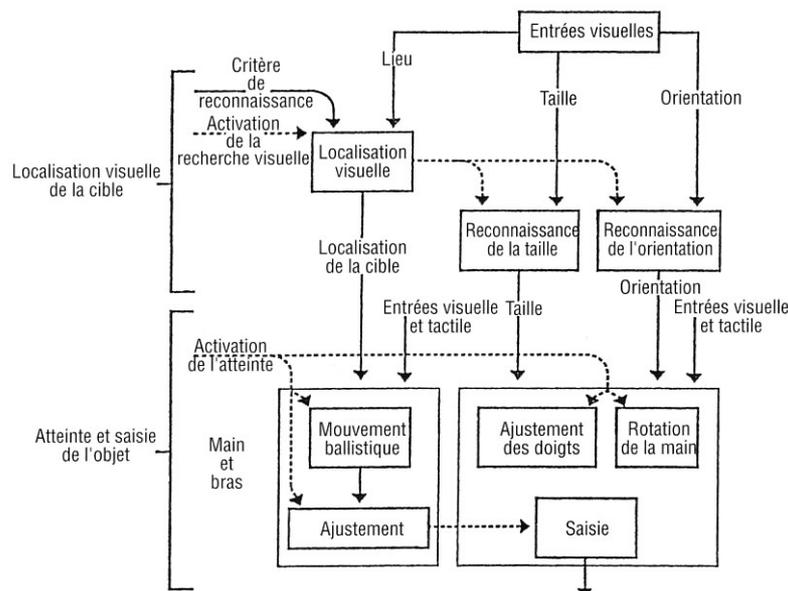
Dans le paragraphe qui suit, nous apportons quelques informations sur le contrôle-commande de tâches manuelles mais sans entrer dans le détail des structures nerveuses.

3. Commande nerveuse du geste

Le système nerveux est un système de commande complexe et de nombreuses études sont réalisées afin d'en mieux comprendre le fonctionnement et d'essayer de le modéliser. On distingue trois fonctions principales dans l'organisation de toute activité gestuelle :

- une entrée qui capte et transmet l'information aux centres nerveux,
- un système de commande
- une sortie qui correspond à la réponse motrice et à la réalisation de la tâche.

Le processus de l'élaboration de la commande consiste en des stimulations électriques qui sont des phénomènes produits aux niveaux les plus élevés du système nerveux central. Ces dernières permettent de produire une succession de contractions musculaires qui commandent le squelette par des couples articulaires. La Figure A1.137 propose un exemple de schéma de commande pour une tâche de préhension (d'après *Biomécanique et physiologie du mouvement*, de S. Bouisset, 2002) :



Légende : en pointillés : signaux d'activation – en traits pleins : transfert de données

Figure A1.137 : Programme de commande d'une tâche de saisie

Ainsi, la tâche de préhension se divise en deux étapes principales. Tout d'abord, le cerveau décompose l'objet en trois caractéristiques visuelles : localisation, taille et orientation. Puis, la commande de saisie entraîne un transfert de la main au voisinage de l'objet, un ajustement de

la position des doigts à la taille de l'objet et une rotation de la pince selon l'orientation de l'objet. L'objet est alors maintenu en position dans la main.

Il est également important de prendre en compte la diversité des mains humaines dans notre modélisation. Les dimensions des différentes parties du corps humain sont obtenues par l'anthropométrie.

Après avoir donné les principales définitions relatives aux études anthropométriques, nous présentons les données existantes pour les mains humaines (Homme, Femme, enfants).

4. Les données anthropométriques

L'anthropométrie est l'étude comparative des dimensions du corps humain. Elle permet d'obtenir des données sur les caractéristiques physiques à partir de mesures scientifiques du corps humain. Celles-ci sont généralement utilisées pour la conception et pour l'évaluation ergonomique des systèmes mécaniques, des équipements de travail, des produits manufacturés ou encore des environnements de travail.

4.1 Définitions

Les mesures de base du corps humain pour la conception technologique sont définies par la norme européenne NF EN ISO 7250 (novembre 1997). Ces mesures répondent à un cahier des charges précis et elles sont réalisées en respectant un protocole prédéfini.

Nous donnons l'exemple de la mesure de la stature afin d'illustrer les différentes définitions.

La Figure A1.138 présente la méthode de Morand utilisée pour définir la hauteur du sommet de la tête au-dessus du sol. Le sujet est debout, le dos en contact avec le plan de référence et la tête orientée dans le plan perpendiculaire au plan de référence (d'après *Enquête anthropométrique sur les conducteurs français – 1981/1982*, de R. Rebiffé, J. Guillien et P. Pasquet, Laboratoire de Physiologie et de Biomécanique de l'association Peugeot - Renault, 1982).

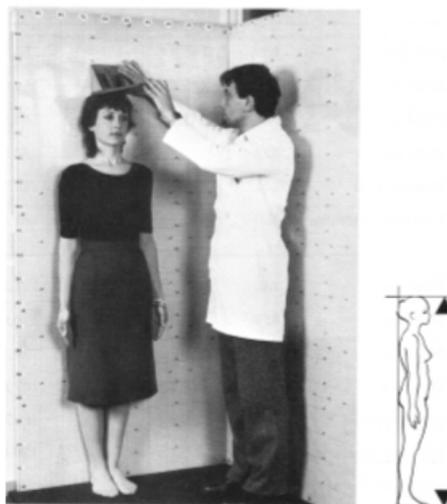


Figure A1.138 : Protocole de mesure de la stature

Les valeurs des statures obtenues par cette mesure pour les populations françaises féminines et masculines sont données par la Figure A1.139 :

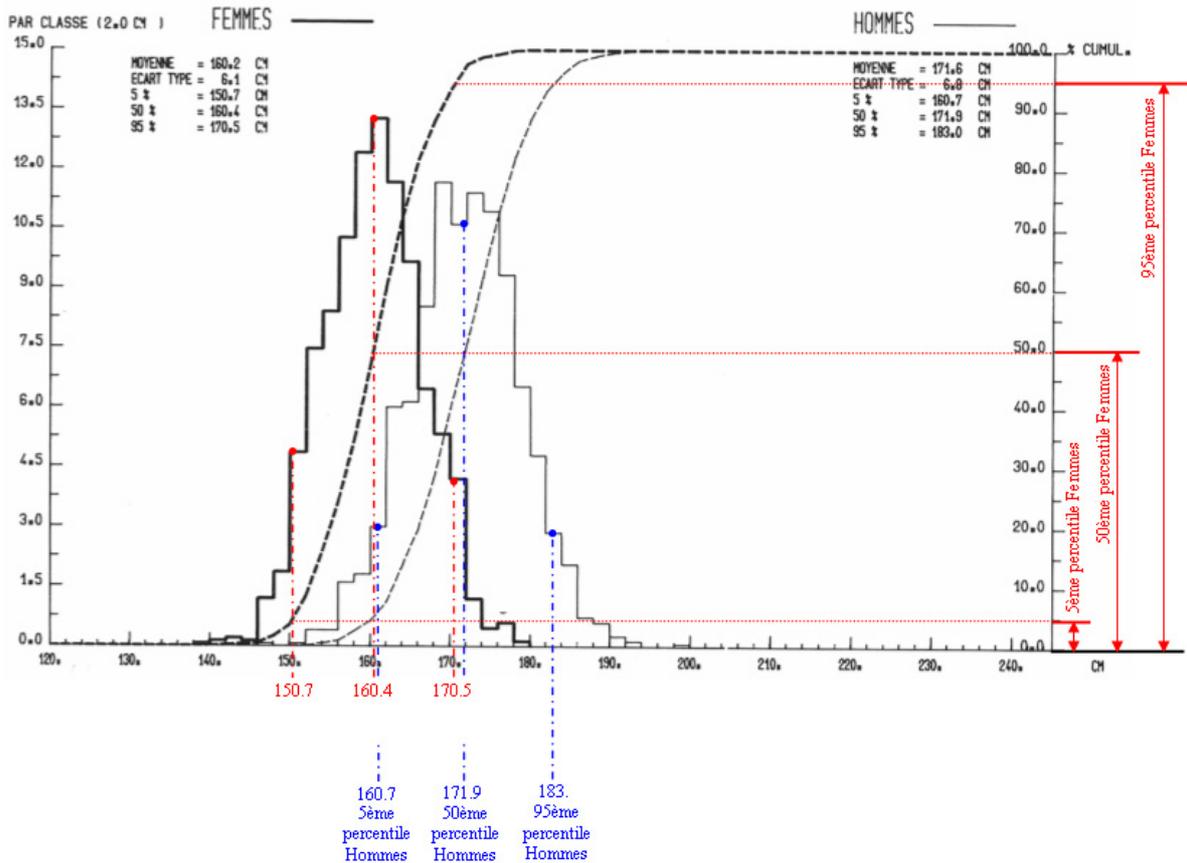


Figure A1.139 : Résultats des mesures de la stature pour les femmes et les hommes

- 5^{ème} percentile (5%) : 5% de la population
- 50^{ème} percentile (50%) : 50% de la population
- 95^{ème} percentile (95%) : 95% de la population

On obtient ainsi la distribution des statures des Femmes et des Hommes sous forme de courbes gaussiennes. Ces dernières révèlent une grande diversité dans les mesures des dimensions du corps humain. Afin de prendre en compte cette variabilité dans la conception de dispositifs techniques ou dans l'évaluation ergonomique des postes de travail, différents mannequins ont été définis. Les trois principaux modèles utilisés – 5%, 50% et 95% - sont représentés sur la Figure A1.140 :

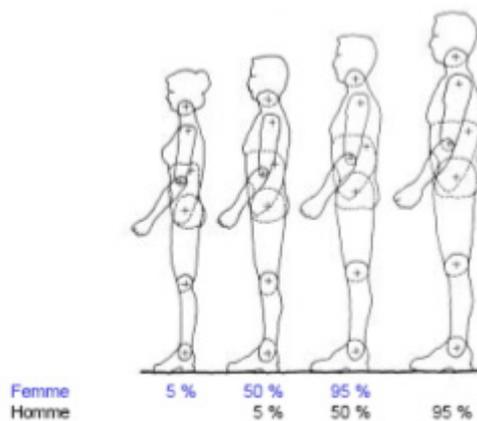


Figure A1.140 : Variabilité des dimensions du corps humain

Après avoir décrit les objectifs de l'anthropométrie ainsi que les définitions générales, nous nous intéressons plus particulièrement aux données relatives aux mains dans le paragraphe suivant.

4.2 Dimensions et contraintes anthropométriques de la main

Comme la main est notre principal organe d'interaction avec l'environnement, des points de repère anatomiques et des mesures anthropométriques de base ont été définis.

Nous présentons les mesures utilisées pour notre modèle ainsi que différentes mesures anthropométriques obtenues pour la main humaine.

4.2.1 Définitions des mesures sur la main

Les mesures anthropométriques de base permettent de définir et de mesurer les dimensions utiles de la main comme par exemple (Cf. Figure A1.141) :

- La longueur de la main (cote 57) : distance styloïde radiale-dactylien,
- La longueur de la paume (cote 58),
- La largeur de la main à la tête des métacarpiens (cote 59),
- Le périmètre maximal de la paume (cote 60),
- L'épaisseur maximale de la main (cote 61).

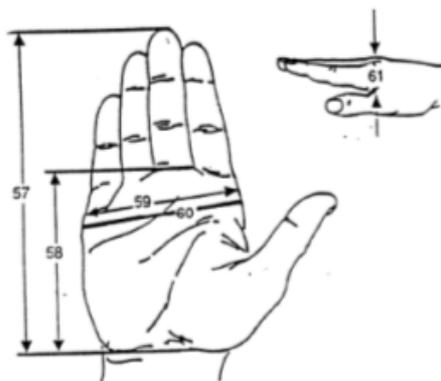


Figure A1.141 : Mesures anthropométriques de base de la main

Toutes ces mesures sont relevées la main tendue dans le prolongement de l'avant-bras, face palmaire posée à plat sur un plan dur (à l'exception de l'épaisseur) à l'aide d'un pied à coulisse ou d'un ruban métrique.

4.2.2 Résultats

De nombreuses études anthropométriques ont été réalisées pour connaître les dimensions des mains. Le Tableau TA1.10 ci-dessous présente les résultats obtenus par différentes études anthropométriques :

Longueur en mm (cote 57)	Références						
	FAA	MIL			NASA	DINBelg	D. Chaffin
Hommes							
1%	173					176	
5%	179	175	181	178	179	182	
50%	193				193	197	
95%	211	211	209	205	206	212	
99%	219					218	
Femmes							
1%	159					160	
5%	165	165	172	176	158	166	
50%	180				172	181	
95%	197	201	195	204	187	196	
99%	205					202	
Largeur en mm (cote 59)	Références						
	FAA	MIL			NASA	DINBelg	D. Chaffin
Hommes							
1%	81					78	
5%	84	82	84	82	82	81	
50%	90				89	88	87
95%	98	98	97	96	96	95	95
99%	100					98	
Femmes							
1%	71					69	
5%	73	69	75	71	69	71	
50%	79				78	77	76
95%	86	86	86	83	86	83	84
99%	89					85	
Périmètre en mm (cote 60)	Références						
	FAA	MIL			NASA	DINBelg	D. Chaffin
Hommes							
1%	192						
5%	199	199	199	200	203		
50%	213				218		215
95%	230	235	228	231	234		232
99%	237						
Femmes							
1%	167						
5%	173	168	176	172	165		
50%	186				179		183
95%	200	200	201	199	193		198
99%	207						

Epaisseur en mm (cote 61)	Références					
	FAA	MIL	NASA	DINBelg	D. Chaffin	
Hommes						
1%					22	
5%					24	
50%					29	30
95%					34	33
99%					36	
Femmes						
1%					19	
5%					21	
50%					26	25
95%					31	29
99%					33	

Tableau TA1.10 : Résultats des mesures anthropométriques de la main

Avec :

- FAA : “Chapter 14 : Anthropometry and biomechanics”, Human Factors Design Standard (HFDS), Federal Aviation Administration (FAA), 2003
- MIL : “MIL-HDBK-759C”, Notice 1, Department of Defense Handbook, Human Engineering Design Guidelines, 1997
- NASA : “Anthropometric Source Book : Volume 1”, NASA RP 1024, 7-78./ NASA-STD-3000, Anthropometry for Designers Anthropology Staff/Webb Associates, NASA, 2000
- DINBelg : “DINBelg 2005”, R. Motmans et E. Ceriez, Groupe d'Ergonomie, Katholieke Hogeschool Limburg, Genk, 2005
- D. Chaffin : « Occupational Ergonomics », D. Chaffin et G. Andersson, John Wiley & Sons, Brisbane, 366 p., 1984

Non seulement il est possible de connaître les dimensions de la main mais il est également possible de définir des contraintes anthropométriques entre les longueurs mesurées.

A partir des données issues des études anthropométriques, des relations entre les mesures réalisées et des intervalles de valeurs possibles peuvent être formulés et intégrés dans un modèle sous forme de contraintes. Dans le cas de la main, nous obtenons des relations entre les longueurs des doigts et de la main de la forme suivante :

- (Largeur paume) / (Longueur main) = $0,44 \pm 0,007$;
- (Longueur index) / (Longueur main) = $0,449 \pm 0,003$;
- (Longueur 1^{ère} phalange index) / (Longueur main) = $0,245 \pm 0,001$;
- (Longueur 1^{ère} phalange majeur) / (Longueur main) = $0,266 \pm 0,003$.

Ces contraintes anthropométriques permettent de spécifier les longueurs de la main (paume et phalanges) conformément aux mesures réelles.

4.3 Conclusion

L'anthropométrie permet de connaître les dimensions corporelles d'une population afin de réaliser des études ergonomiques de postes de travail ou de modéliser et paramétrer des systèmes technologiques.

Dans le cadre de la simulation de tâches de préhension en environnement virtuel, ces mesures nous permettent de dimensionner notre main virtuelle afin qu'elle soit la plus réaliste possible. Les contraintes anthropométriques nous permettent également de modifier les longueurs en fonction de l'utilisateur en représentant une main d'adulte (Homme ou Femme) ou une main d'enfant. Avec un modèle conforme à la main de l'utilisateur, il nous est alors possible d'évaluer en environnement virtuel l'accessibilité d'un opérateur, l'encombrement dû à la main ou à un outil au cours d'une tâche de maintenance ou d'assemblage, ou encore l'ergonomie d'une tâche manuelle.

A.2. Les modèles de contact et de distribution de pression

On appelle « action de contact » l'action exercée par deux objets en contact au niveau de leur surface commune. Lorsque deux objets entrent en contact, une pression de contact s'exerce entre eux. Celle-ci engendre des contraintes et des déformations dans les objets au voisinage de leur surface commune. Afin de déterminer les actions de contact, il faut connaître en règle générale :

- le coefficient de frottement,
- la géométrie des surfaces en contact,
- la loi de répartition de la pression.

Dans cette annexe, on décrit les différentes actions de contact (avec ou sans frottement) ainsi que les pressions de contact qui peuvent exister entre deux objets supposés indéformables, déformables, ou encore l'un déformable et l'autre infiniment rigide.

Les explications données sont issues de Systèmes Mécaniques : Théorie et Dimensionnement, de M. Aublin, R. Boncompain, M. Boulaton, D. Caron, E. Jeay, B. Lacage et J. Réa, Editions Dunod, 1992, ISBN : 2 10 004162 2.

1. Actions de contact sans frottement

On distingue deux types de contact :

- le contact surfacique : dans le cas où le contact se fait suivant une surface nominale importante,
- le contact dit linéique ou ponctuel : dans le cas où le contact se fait selon une petite surface nominale.

1.1 Contact surfacique

Dans le cas où le contact est sans frottement, l'action élémentaire d'un solide sur un autre est portée par la normale au plan de contact (Cf. Figure A1.132) :

$$p_{2/1}(M)dS = -p(M)n_1(M)dS \quad (\text{Eq. A2-157})$$

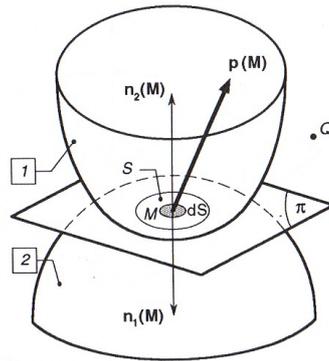


Figure A2.142 : Contact entre deux solides

1.2 Contact linéique et ponctuel

On suppose les solides indéformables :

- dans le cas du contact linéique, la zone commune de contact des solides est une ligne. L'action s'exerce donc en n'importe quel point de cette ligne.
- dans le cas du contact ponctuel, les deux solides sont uniquement en contact en un point I.

Ces modèles sont souvent employés dans les études statiques. Mais, de manière générale, on les utilise dans tous les cas où la surface de contact est peu étendue.

2. Actions de contact avec frottement

2.1 Modèle d'étude

La Figure A2.143 présente deux solides 1 et 2 en contact suivant une surface commune S. Supposons qu'ils puissent glisser l'un par rapport à l'autre. L'action de contact exercée par 2 sur 1 est définie à partir d'une répartition ou densité de charge $\vec{p}(M)$, déterminée en chaque point M de la surface S.

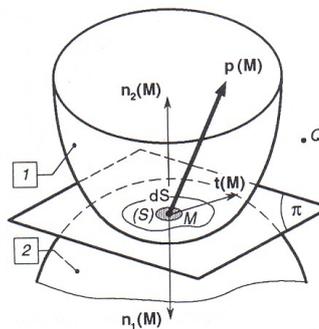


Figure A2.143 : Action de contact avec frottement

En présence de frottement, on obtient :

$$\vec{p}(M) = -p_n(M)\vec{n}_1(M) + p_t\vec{t}(M) \quad (\text{Eq. A2-158})$$

Dans cette relation, on a :

p_n représente la répartition surfacique normale,

p_t est la répartition surfacique tangentielle,

$\vec{n}_1(M)$ est la normale en M au solide 1,

$\vec{t}(M)$ est la tangente en M au solide 1, sa direction sera définie ultérieurement.

Pour étudier le frottement de deux solides, on utilise les lois expérimentales de Coulomb.

2.2 Lois de Coulomb

Ces lois permettent de distinguer deux cas suivant que la vitesse de glissement d'un solide par rapport à l'autre est nulle ou pas.

2.2.1 Premier cas

On fait l'hypothèse que la vitesse de glissement n'est pas nulle :

$$\vec{V}(M,1/2) \neq 0 \quad (\text{Eq. A2-159})$$

On a donc les résultats suivants :

$$\begin{cases} \vec{t}(M) \wedge \vec{V}(M,1/2) = \vec{0} \\ \vec{t}(M) \cdot \vec{V}(M,1/2) = 0 \\ \frac{p_t(M)}{p_n(M)} = \tan\varphi = f \end{cases} \quad (\text{Eq. A2-160})$$

avec φ l'angle de frottement et $\tan\varphi = f$ le coefficient de frottement de glissement.

Dans ce cas, la pression de contact au point M se trouve sur un cône d'angle au sommet φ , appelé cône de frottement (Cf. Figure A2.144).

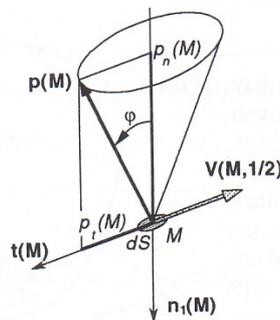


Figure A2.144 : Contact entre deux solides avec frottement, vitesse de glissement non nulle

2.2.2 Deuxième cas

On fait l'hypothèse que la vitesse de glissement est nulle :

$$\vec{V}(M, 1/2) = \vec{0} \quad (\text{Eq. A2-161})$$

Les solides sont immobiles l'un par rapport à l'autre, ou bien ils roulent ou pivotent sans glisser. On obtient l'équation suivante :

$$\frac{p_t(M)}{p_n(M)} = \tan \alpha \quad (\text{Eq. A2-162})$$

avec $0 \leq \alpha \leq \varphi'$

φ' : est l'angle d'adhérence,

$\alpha = \varphi'$ à la limite de l'équilibre (équilibre strict),

$\vec{t}(M)$ appartient au plan tangent en M à 1 et 2 ; sa direction est a priori inconnue.

Dans ce cas, l'action de contact au point M se trouve à l'intérieur d'un cône d'angle au sommet φ' appelé cône d'adhérence (Cf. Figure A2.145).

En général, on a $\varphi' \geq \varphi$.

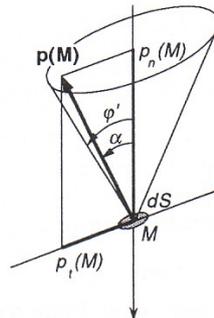


Figure A2.145 : Contact entre deux solides avec frottement, vitesse de glissement nulle

2.3 Le phénomène de « broutage »

Ce phénomène, appelé *stick/slip* en anglais, résulte du fait que le coefficient d'adhérence est plus grand que le coefficient de frottement. Par conséquent, l'effort nécessaire pour mettre en mouvement un corps préalablement au repos est plus important que celui qui sert à entretenir le mouvement. L'effort appliqué n'est pas maintenu.

L'équation A2-2 donne :

$$\begin{aligned} \vec{p}(M) &= -p_n(M)\vec{n}_1(M) + p_t\vec{t}(M) \\ &= \vec{F}_n(M) + \vec{F}_t(M) \end{aligned} \quad (\text{Eq. A2-163})$$

avec \vec{F}_n la force normale et \vec{F}_t la force tangente en M.

Dans ce cas de figure, on exprime les lois de Coulomb par une complémentarité entre la force tangente, le mouvement $\vec{V}(M, 1/2)$, la force normale et un coefficient de frottement $f = \tan \varphi$ qui dépend du matériau. On obtient alors la relation suivante :

$$\|\vec{F}_t(M)\| \leq f \|\vec{F}_n(M)\| \quad (\text{Eq. A2-164})$$

La phase de glissement se traduit par :

$$\text{si } \|\vec{V}(M, 1/2)\| > V_{\min} \quad \vec{F}_t(M) = f \|\vec{F}_n(M)\| \cdot \frac{\vec{V}(M, 1/2)}{\|\vec{V}(M, 1/2)\|} \quad (\text{Eq. A2-165})$$

La phase d'adhérence s'exprime par la relation suivante :

$$\text{si } \|\vec{V}(M, 1/2)\| < V_{\min} \quad \|\vec{F}_t(M)\| < f \|\vec{F}_n(M)\| \Rightarrow \vec{V}(M, 1/2) = \vec{0} \quad (\text{Eq. A2-166})$$

La formulation de Coulomb a l'avantage d'être très simple. Mais parfois, ce modèle est loin de la réalité car il ne prend en compte que la nature des matériaux en contact. Avec les équations de Coulomb, il est possible de connaître les actions appliquées au niveau des contacts mais on n'a aucune information sur la distribution du champ de pression.

Dans certains cas, il est important de considérer des paramètres tels que la distribution de la pression de contact ou encore la vitesse relative.

3. Pressions de contact entre solides

Le problème de la détermination des pressions de contact met en jeu la géométrie des surfaces de contact et la nature de la déformation des matériaux. Plusieurs modèles sont couramment utilisés pour définir le champ de pression :

- dans le cas de contacts ponctuel ou linéique avec une aire de contact petite, on utilise la théorie de Hertz,
- dans le cas de contacts larges et étendus, on utilise trois modèles comme la pression uniforme ou encore la pression fonction de la déformation.

Dans ce paragraphe, nous décrivons ces différentes méthodes.

3.1 La théorie de Hertz

Quand deux surfaces d'objets se touchent, elles sont initialement en contact en un unique point. Au fur et à mesure qu'elles se rapprochent et que la charge entre elles augmente, se crée une déformation au niveau de ce point de contact. Par conséquent, la zone de contact augmente ainsi que la distribution et la magnitude des forces de pression qui s'exercent entre les deux surfaces (Cf. Figure A2.146).

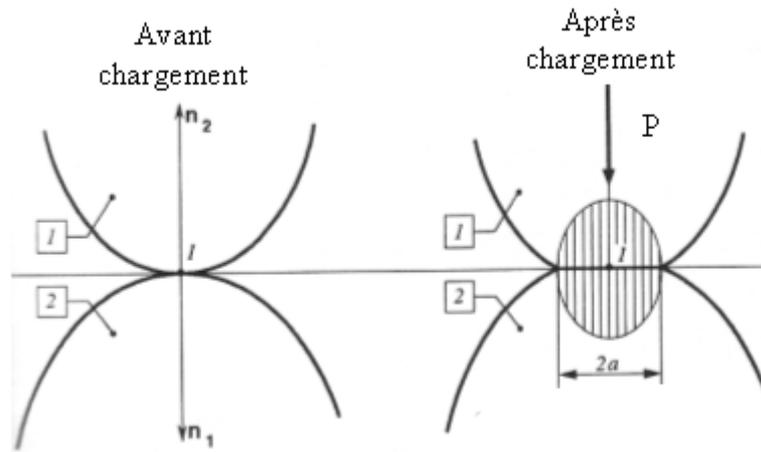


Figure A2.146 : Modélisation de la déformation

La théorie de Hertz permet de déterminer les dimensions de la surface de contact, la pression de contact maximale et les contraintes engendrées en surface et en profondeur lorsque deux surfaces entrent en contact.

3.1.1 Définitions

Les objets en contact sont des corps élastiques, homogènes et isotropes. Chaque matériau est caractérisé par un module d'élasticité E et un coefficient de Poisson ν . On a donc le module d'élasticité E_1 et le coefficient de Poisson ν_1 pour le corps 1, et respectivement E_2 et ν_2 pour le corps 2.

On note également : $k_i = \frac{1-\nu_i^2}{\pi E_i}$

Les rayons de courbure principaux des corps 1 et 2 sont donnés par $C_1 = \frac{1}{R_1}$ et $C_2 = \frac{1}{R_2}$.

Ainsi, le rayon de courbure relatif entre les deux corps en contact est donné par :

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2}$$

Dans le cas d'un contact ponctuel, la surface de contact est supposée elliptique. Elle sera déterminée par son grand axe $2a$ et son petit axe $2b$.

Dans ce cas de figure, la déformation est modélisée selon un demi-ellipsoïde. La répartition de pression se fait de la manière suivante (Cf. Figure A2.147) :

- au point de contact I, centre de l'ellipsoïde, la déformation et la pression de contact sont maximales,
- à la frontière, la pression de contact est nulle.

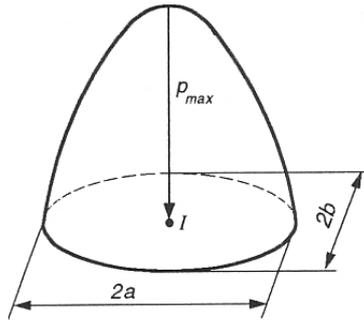


Figure A2.147 : Modélisation de la pression de contact

Connaissant la charge appliquée P , on obtient respectivement la pression de contact maximale notée p_{max} et la pression moyenne p_{moy} par :

$$p_{max} = \frac{3}{2} \frac{P}{\pi ab} \quad (\text{Eq. A2-167})$$

$$p_{moy} = \frac{P}{\pi ab} \quad (\text{Eq. A2-168})$$

Si la déformation est connue, la théorie de l'élasticité permet de déduire le champ de contraintes en profondeur. Ainsi lorsque l'on descend dans la matière, l'évolution des contraintes principales le long de la perpendiculaire en I au plan de l'ellipse d'appui n'est pas parallèle. Les tri-cercles de Mohr associés à ces contraintes principales permettent de définir les contraintes de cisaillement à différentes profondeurs.

Il est également possible de définir le rapprochement des deux solides avec cette méthode. Celui-ci est défini par la différence de distance prise perpendiculaire au plan π de deux points situés loin des zones de déformation entre la position initiale et la position finale :

$$\delta = r \frac{3\pi}{4a} (k_1 + k_2) P \quad (\text{Eq. A2-169})$$

avec r la distance radiale au point de contact I .

3.1.2 Cas particulier

Dans le cas particulier d'un contact ponctuel entre des objets cylindriques, on obtient les équations décrites dans le paragraphe ci-après.

Quand les deux objets cylindriques entrent en contact, et que s'exerce une force de pression connue notée P , on définit le rayon a de la zone de contact par :

$$a = \left(\frac{3\pi}{4} PR(k_1 + k_2) \right)^{1/3} \quad (\text{Eq. A2-170})$$

avec :

$k_1 = \frac{1 - \nu_1^2}{\pi E_1}$ et $k_2 = \frac{1 - \nu_2^2}{\pi E_2}$ où E_1 est le module d'élasticité de l'objet 1 et E_2 est celui de l'objet 2, et ν_1 est le coefficient de Poisson de l'objet 1, et ν_2 celui de l'objet.

En faisant l'hypothèse que la zone de contact est circulaire, définie par un rayon a et un chargement P , alors la pression de contact maximale notée p_{\max} est donnée par :

$$p_{\max} = \left(\frac{3P}{2a^2\pi} \right) \quad (\text{Eq. A2-171})$$

La distribution de pression qui en résulte est de la forme :

$$p = p_{\max} \sqrt{1 - \left(\frac{r}{a} \right)^2} \quad (\text{Eq. A2-172})$$

avec r la distance radiale à partir du point de contact tel que $r < a$.

Dans le cas de la simulation de préhension d'objets avec une main virtuelle, il est possible d'utiliser la théorie de Hertz pour prédire la distribution de pression à la surface des extrémités des doigts quand elles entrent en contact avec des objets cylindriques. On note alors R_1 le rayon de la pulpe du doigt et R_2 le rayon de l'objet en contact. De plus, on définit le coefficient de Poisson du doigt ν_1 égal à 0,48 et si l'objet est incompressible, on pose alors $\nu_2 = 0,5$.

En utilisant les équations A2-10 à A2-12, on est alors capable d'estimer le champ de pression au moment de la saisie de l'objet ainsi que de calculer la déformation entre un doigt et l'objet cylindrique.

Les équations précédentes permettent de résoudre des contacts ponctuels. Lorsque l'étendue de la surface commune devient importante, le contact entre les objets se fait de manière aléatoire. On utilise dans ce cas de figure plusieurs modèles simples.

3.2 La pression uniforme

Ce modèle suppose que la répartition de pression de contact est uniforme sur toute la surface de contact :

$$p(M) = p_0 = \text{cste} \quad (\text{Eq. A2-173})$$

Ce modèle est utilisé dans le cas de contact entre solides indéformables avec une géométrie parfaite des surfaces en contact.

3.3 La pression proportionnelle à la déformation

Ce modèle suppose que l'un ou les deux objets en contact se déforment. On fait l'hypothèse d'une relation entre la déformation $\delta(M)$ en un point de la surface de contact et la pression $p(M)$ en ce même point. On obtient la relation :

$$p(M) = K \delta(M)^\alpha \quad (\text{Eq. A2-174})$$

avec K coefficient lié à la rigidité des matériaux en contact, et $\alpha \geq 1$ traduit le comportement des matériaux en contact.

A.3. L'Analyse en Composantes Principales (ACP)

L'ACP est une méthode classique pour la réduction de dimensions : elle permet d'extraire les directions principales d'un nuage de points.

Les explications données dans cette annexe sont issues de The Symmetric Eigenvalue Problem, de G. H. Golub, C. F. Van Loan, dans *Matrix Computations*, 2nd ed. Baltimore, MD: Johns Hopkins University Press, 1989.

1. Définitions

Pour définir les directions principales, on fait l'approximation que ces N_p points forment un ellipsoïde. On calcule alors :

- les axes de cet ellipsoïde qui forment la base ACP,
- la longueur de chacun d'eux.

Ces valeurs sont nécessaires pour identifier la direction principale de l'ellipsoïde (Cf. Figure A3.148) et pour quantifier sa taille et son volume.

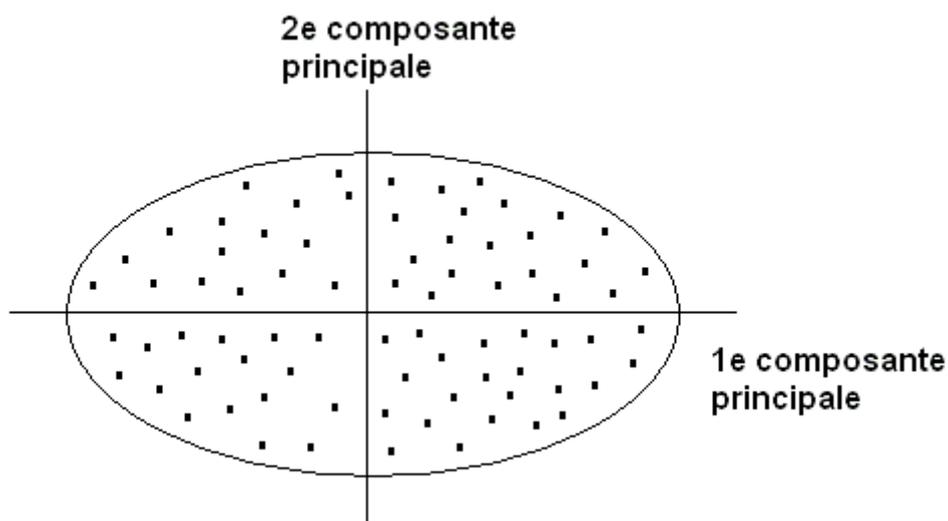


Figure A3.148 : Composantes principales d'un nuage de points

On note $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ la base dans laquelle s'expriment les coordonnées de ces N_p points.

Soit $(\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3)$ la base ACP.

Pour connaître les coordonnées des N_p points dans la base $(\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3)$, nous calculons la matrice de passage de la base $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ à la base $(\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3)$.

2. Calcul de la matrice de covariance

Calculer la matrice de covariance revient à faire l'approximation que les N_p points font partie d'un ellipsoïde à trois dimensions.

Le terme général de la matrice de covariance s'écrit :

$$\text{cov}(x, y) = \sum_{i=1}^p (x_i - \bar{x})(y_i - \bar{y}) \quad (\text{Eq. A3-175})$$

avec \bar{x} la moyenne arithmétique de x telle que $\bar{x} = \frac{1}{p} \sum_{i=1}^p x_i$

Sachant que les N_p points du nuage sont dans un espace à trois dimensions, la matrice \mathbf{C} est donc une matrice de dimension (3×3) .

Soient X, Y, Z respectivement l'ensemble des coordonnées x, y, z des N_p points.

La matrice de covariance \mathbf{C} s'écrit de la façon suivante :

$$\mathbf{C} = \begin{bmatrix} \text{cov}(X, X) & \text{cov}(X, Y) & \text{cov}(X, Z) \\ \text{cov}(Y, X) & \text{cov}(Y, Y) & \text{cov}(Y, Z) \\ \text{cov}(Z, X) & \text{cov}(Z, Y) & \text{cov}(Z, Z) \end{bmatrix} \quad (\text{Eq. A3-176})$$

C'est une matrice symétrique et par conséquent, ses vecteurs propres forment une base orthonormée.

En diagonalisant la matrice de covariance \mathbf{C} , on obtient :

- les vecteurs propres de \mathbf{C} qui forment les axes de l'ellipsoïde,
- les valeurs propres correspondantes qui sont les longueurs des axes.

Ainsi, le vecteur propre correspondant à la plus grande valeur propre donne la direction de plus grande dispersion du nuage de points.

3. Calcul de la matrice de passage

La matrice de passage s'obtient par diagonalisation de la matrice de covariance \mathbf{C} associée aux N_p points.

Soit \mathbf{A} une matrice $(n \times n)$ et un vecteur $\mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq 0$. \mathbf{A} est la matrice de passage si et seulement si :

$$\mathbf{A} \cdot \mathbf{x} = \lambda \cdot \mathbf{x} \Leftrightarrow (\mathbf{A} - \lambda \cdot \mathbf{I}) \cdot \mathbf{x} = 0 \quad (\text{Eq. A3-177})$$

avec λ un scalaire quelconque et \mathbf{I} la matrice identité. .

λ est appelé « valeur propre » de \mathbf{A} , et \mathbf{x} est le « vecteur propre » correspondant.

La partie gauche de l'équation A3-3 est connue comme étant l'équation caractéristique de \mathbf{A} .

La partie droite est le polynôme caractéristique de la matrice \mathbf{A} .

En écrivant cette relation sous forme matricielle, on obtient :

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} = \lambda \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} \quad (\text{Eq. A3-178})$$

Ce qui est équivalent au système homogène suivant :

$$\begin{bmatrix} a_{11} - \lambda & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} - \lambda & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} - \lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (\text{Eq. A3-179})$$

avec \mathbf{I} la matrice identité.

On remarque que plus la dimension de la matrice est grande, plus le système à résoudre devient complexe. Différents algorithmes ont été mis en place afin de résoudre les valeurs propres et les vecteurs propres :

- La décomposition QR,
- La décomposition SVD (Singular Value Decomposition),
- La méthode de Jacobi.

A.4. Compléments sur les Eléments Finis

1. Fonction d'interpolation pour le triangle

Soit T_r un triangle. Il est défini par ses trois sommets notés v_i , avec $i=0, \dots, 2$. La position initiale d'un point P du triangle est donnée par ses coordonnées x et y . Dans le cas 2D, un vecteur déplacement \mathbf{U} a deux composantes : $u(x, y)$ et $v(x, y)$. Ainsi, pour le point P , on écrit le vecteur déplacement $\mathbf{U}(\mathbf{P})$ à partir des vecteurs déplacement des sommets [Cotin 99] [Hyncik 02] :

$$\mathbf{U}(\mathbf{P}) = \sum_{i=0}^2 g_i(\mathbf{P}) \mathbf{U}(\mathbf{v}_i) \quad (\text{Eq. A4-180})$$

Avec $\psi_i(\mathbf{P})$ les fonctions d'interpolation qui correspondent à l'interpolation linéaire à l'intérieur du triangle et $\mathbf{U}(\mathbf{v}_i)$ les déplacements des sommets v_1 , v_2 et v_3 du triangle (Cf. Figure A4.149). On a alors :

$$\begin{aligned} \mathbf{U}(\mathbf{P}) &= g_1 \mathbf{U}(\mathbf{v}_1) + g_2 \mathbf{U}(\mathbf{v}_2) + g_3 \mathbf{U}(\mathbf{v}_3) \\ &= \frac{\text{Aire}_1}{\text{Aire totale}} \mathbf{U}(\mathbf{v}_1) + \frac{\text{Aire}_2}{\text{Aire totale}} \mathbf{U}(\mathbf{v}_2) + \frac{\text{Aire}_3}{\text{Aire totale}} \mathbf{U}(\mathbf{v}_3) \quad (\text{Eq. A4-181}) \end{aligned}$$

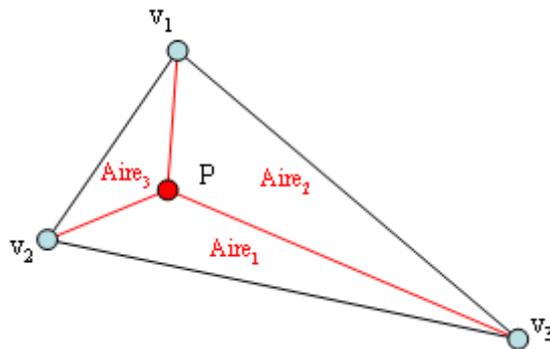


Figure A4.149 : Fonctions d'interpolation pour un triangle

2. Fonction d'interpolation pour le tétraèdre

Soit T_e un tétraèdre. Il est défini par ses quatre sommets notés p_i , avec $i=0, \dots, 3$. La position initiale d'un point M du tétraèdre est donnée par ses coordonnées x, y et z . Dans ce cas, un vecteur déplacement $\mathbf{U}(\mathbf{M})$ est défini par les composantes $u(x, y, z)$, $v(x, y, z)$ et $w(x, y, z)$.

On a l'expression du vecteur déplacement $\mathbf{U}(\mathbf{M})$ à partir des vecteurs déplacements des sommets $\mathbf{U}(\mathbf{p}_i)$ et des fonctions d'interpolation [Cotin 99] [Delingette 04] :

$$\mathbf{U}(\mathbf{M}) = \sum_{i=0}^3 g_i(\mathbf{M}) \mathbf{U}(\mathbf{p}_i) \quad (\text{Eq. A4-182})$$

Les fonctions d'interpolation $g_i(\mathbf{M})$ correspondent à l'interpolation linéaire à l'intérieur du tétraèdre (Cf. Figure A4.150). Ces fonctions définissent les coordonnées barycentriques du point M à l'intérieur du tétraèdre en fonction des coordonnées des sommets du tétraèdre. On obtient l'expression de ces fonctions par la relation linéaire suivante :

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} p_{0x} & p_{1x} & p_{2x} & p_{3x} \\ p_{0y} & p_{1y} & p_{2y} & p_{3y} \\ p_{0z} & p_{1z} & p_{2z} & p_{3z} \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \mathbf{P} \cdot \mathbf{G} \quad (\text{Eq. A4-183})$$

Avec $\mathbf{p}_i = (p_{ix}, p_{iy}, p_{iz})^T$ les coordonnées de chaque sommet du tétraèdre, \mathbf{P} la matrice des coordonnées des sommets et \mathbf{G} la matrice des fonctions d'interpolations g_i .

La matrice \mathbf{P} enveloppe la forme du tétraèdre au repos. On obtient le déterminant de \mathbf{P} :

$$\det(\mathbf{P}) = 6 \times V(T_e) \quad (\text{Eq. A4-184})$$

avec $V(T_e)$ le volume du tétraèdre.

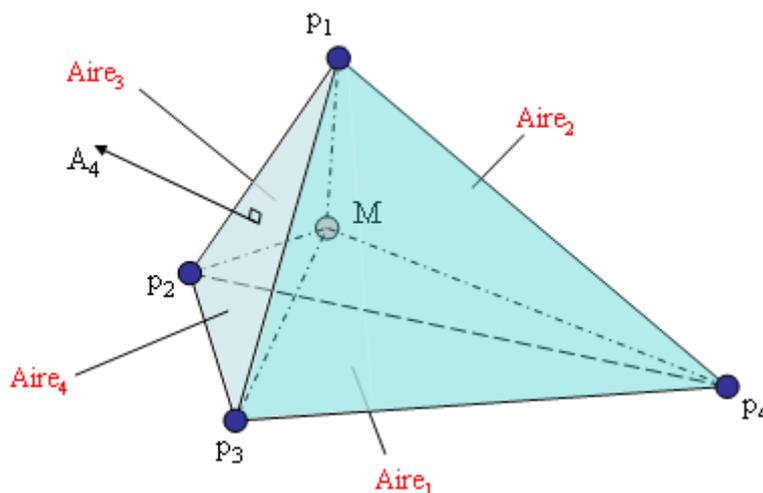


Figure A4.150 : Fonctions d'interpolation pour un tétraèdre

On peut donc inverser la matrice \mathbf{P} et calculer la matrice \mathbf{G} pour déterminer les fonctions $g_i(\mathbf{M})$. On obtient :

$$\mathbf{P}^{-1} = \frac{-1}{6V(T_e)} \begin{bmatrix} A_{0x} & A_{0y} & A_{0z} & -V_0 \\ A_{1x} & A_{1y} & A_{1z} & -V_1 \\ A_{2x} & A_{2y} & A_{2z} & -V_2 \\ A_{3x} & A_{3y} & A_{3z} & -V_3 \end{bmatrix} \quad (\text{Eq. A4-185})$$

Les fonctions d'interpolation sont données par la relation suivante :

$$g_i(\mathbf{M}) = -\frac{\mathbf{A}_i \cdot \mathbf{M} - V_i}{6V(T_e)} \quad (\text{Eq. A4-186})$$

Avec \mathbf{A}_i le vecteur normal de la surface opposée au sommet p_i considéré, et V_i est six fois le volume du tétraèdre obtenu en considérant l'origine et les sommets p_i .

En tenant compte des fonctions d'interpolation, on peut exprimer le vecteur déplacement par :

$$\mathbf{U}(\mathbf{M}) = -\sum_{i=0}^3 \frac{\mathbf{A}_i \cdot (\mathbf{M} - \mathbf{p}_{i+1})}{6V(T_e)} \mathbf{U}(\mathbf{p}_i) = \mathbf{G} \cdot \begin{bmatrix} \mathbf{U}(\mathbf{p}_0) \\ \mathbf{U}(\mathbf{p}_1) \\ \mathbf{U}(\mathbf{p}_2) \\ \mathbf{U}(\mathbf{p}_3) \end{bmatrix} \quad (\text{Eq. A4-187})$$

Avec :

$$\mathbf{G} = \begin{bmatrix} g_0 & 0 & 0 & g_1 & 0 & 0 & g_2 & 0 & 0 & g_3 & 0 & 0 \\ 0 & g_0 & 0 & 0 & g_1 & 0 & 0 & g_2 & 0 & 0 & g_3 & 0 \\ 0 & 0 & g_0 & 0 & 0 & g_1 & 0 & 0 & g_2 & 0 & 0 & g_3 \end{bmatrix} \quad (\text{Eq. A4-188})$$

On obtient donc un système linéaire en fonction du déplacement \mathbf{U} . La construction de ce nouveau système d'équations s'appelle « l'assemblage ».

3. Calcul des déformations à partir des déplacements :

Une fois le déplacement défini par les fonctions d'interpolation, on calcule la contrainte à l'intérieur de chaque élément. Tout d'abord, on calcule la déformation qui résulte de l'action de la force de réaction à travers chaque élément. Elle est notée $\boldsymbol{\varepsilon}$ et elle est liée au déplacement \mathbf{U} par le tenseur non linéaire de Green – Lagrange :

$$\varepsilon_{ij}(\mathbf{U}) = \frac{1}{2} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} + \sum_{k=1}^3 \frac{\partial U_k}{\partial x_i} \frac{\partial U_k}{\partial x_j} \right) \quad (\text{Eq. A4-189})$$

avec $1 \leq i, j \leq 3$.

Il est invariant en rotation et en translation. Les termes diagonaux du tenseur de déformation représentent la déformation dans la direction normale, alors que les autres termes représentent

le cisaillement.

Dans notre cas de figure, nous avons fait l'hypothèse de petits déplacements [Bro Nielsen 98] [Cotin 99] [Brown 03] [Delingette 04] : les termes d'ordre 2 peuvent donc être négligés. On linéarise le tenseur de Green – Lagrange et on obtient le tenseur de déformation de Cauchy :

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \quad (\text{Eq. A4-190})$$

avec $1 \leq i, j \leq 3$ et \mathbf{U} un vecteur déplacement défini par les composantes $u(x, y, z)$, $v(x, y, z)$ et $w(x, y, z)$.

Dans le cas 3D, on obtient le tenseur de déformation $\boldsymbol{\varepsilon}$ suivant :

$$\boldsymbol{\varepsilon}^T = \left[\varepsilon_{xx} \quad \varepsilon_{yy} \quad \varepsilon_{zz} \quad \varepsilon_{xy} \quad \varepsilon_{yz} \quad \varepsilon_{zx} \right] \quad (\text{Eq. A4-191})$$

avec :

$$\begin{aligned} \varepsilon_{xx} &= \frac{\partial u}{\partial x} & \varepsilon_{yy} &= \frac{\partial v}{\partial y} & \varepsilon_{zz} &= \frac{\partial w}{\partial z} \\ \gamma_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} & \gamma_{yz} &= \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} & \gamma_{zx} &= \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \end{aligned}$$

Sous forme matricielle, on a :

$$\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{bmatrix} = \begin{bmatrix} \partial/\partial x & 0 & 0 \\ 0 & \partial/\partial y & 0 \\ 0 & 0 & \partial/\partial z \\ \partial/\partial y & \partial/\partial x & 0 \\ 0 & \partial/\partial z & \partial/\partial y \\ \partial/\partial z & 0 & \partial/\partial x \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (\text{Eq. A4-192})$$

que l'on note :

$$\boldsymbol{\varepsilon} = \mathbf{B} \cdot \mathbf{U} \quad (\text{Eq. A4-193})$$

Nous exprimons le champ de déplacement d'un point de l'élément discret en fonction des déplacements des nœuds :

$$\mathbf{U}(\mathbf{M}) = g_0 \mathbf{U}(\mathbf{p}_0) + g_1 \mathbf{U}(\mathbf{p}_1) + g_2 \mathbf{U}(\mathbf{p}_2) + g_3 \mathbf{U}(\mathbf{p}_3) \quad (\text{Eq. A4-194})$$

Soit en écriture matricielle :

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} g_0 & 0 & 0 & g_1 & 0 & 0 & g_2 & 0 & 0 & g_3 & 0 & 0 \\ 0 & g_0 & 0 & 0 & g_1 & 0 & 0 & g_2 & 0 & 0 & g_3 & 0 \\ 0 & 0 & g_0 & 0 & 0 & g_1 & 0 & 0 & g_2 & 0 & 0 & g_3 \end{bmatrix} \begin{bmatrix} u_0 \\ v_0 \\ w_0 \\ u_1 \\ v_1 \\ w_1 \\ u_2 \\ v_2 \\ w_2 \\ u_3 \\ v_3 \\ w_3 \end{bmatrix} \quad (\text{Eq. A4-195})$$

En remplaçant dans l'équation (A4-192), on obtient l'équation (A4-196) suivante :

$$\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{bmatrix} = \begin{bmatrix} \partial/\partial x & 0 & 0 \\ 0 & \partial/\partial y & 0 \\ 0 & 0 & \partial/\partial z \\ \partial/\partial y & \partial/\partial x & 0 \\ 0 & \partial/\partial z & \partial/\partial y \\ \partial/\partial z & 0 & \partial/\partial x \end{bmatrix} \begin{bmatrix} g_0 & 0 & 0 & g_1 & 0 & 0 & g_2 & 0 & 0 & g_3 & 0 & 0 \\ 0 & g_0 & 0 & 0 & g_1 & 0 & 0 & g_2 & 0 & 0 & g_3 & 0 \\ 0 & 0 & g_0 & 0 & 0 & g_1 & 0 & 0 & g_2 & 0 & 0 & g_3 \end{bmatrix} \begin{bmatrix} u_0 \\ v_0 \\ w_0 \\ u_1 \\ v_1 \\ w_1 \\ u_2 \\ v_2 \\ w_2 \\ u_3 \\ v_3 \\ w_3 \end{bmatrix}$$

Pour tous les points à l'intérieur du tétraèdre, la déformation est supposée constante. Comme on utilise des fonctions d'interpolation linéaires, on a une relation linéaire entre le tenseur des déformations ε et les déplacements U_i des nœuds d'un élément. Ainsi, la matrice \mathbf{B} de l'équation (A4-193) est une matrice constante.

A.5. Diagrammes des Classes de la Bibliothèque « Collision »

1. Diagramme de classe de hiérarchies de volumes englobants

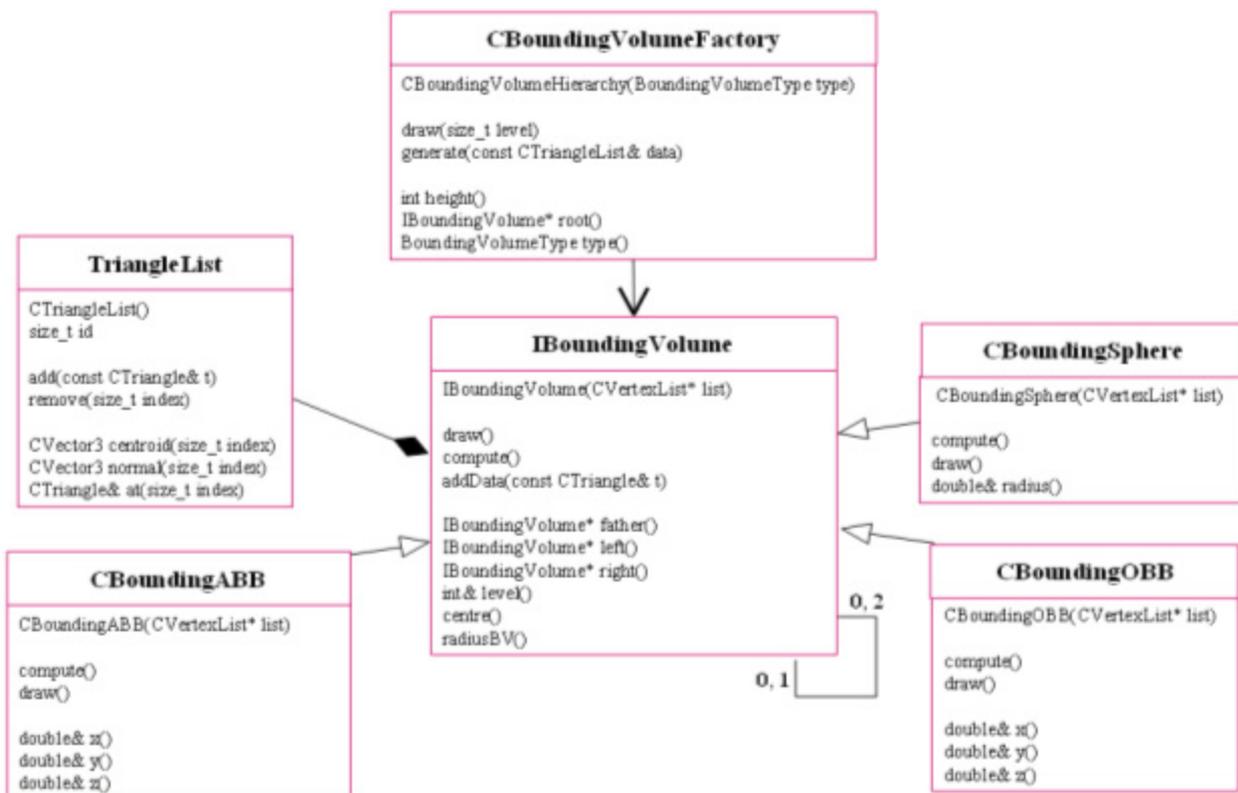


Figure A5.151 : Diagramme des classes pour les BVH

2. Diagramme de classe d'un objet maillé

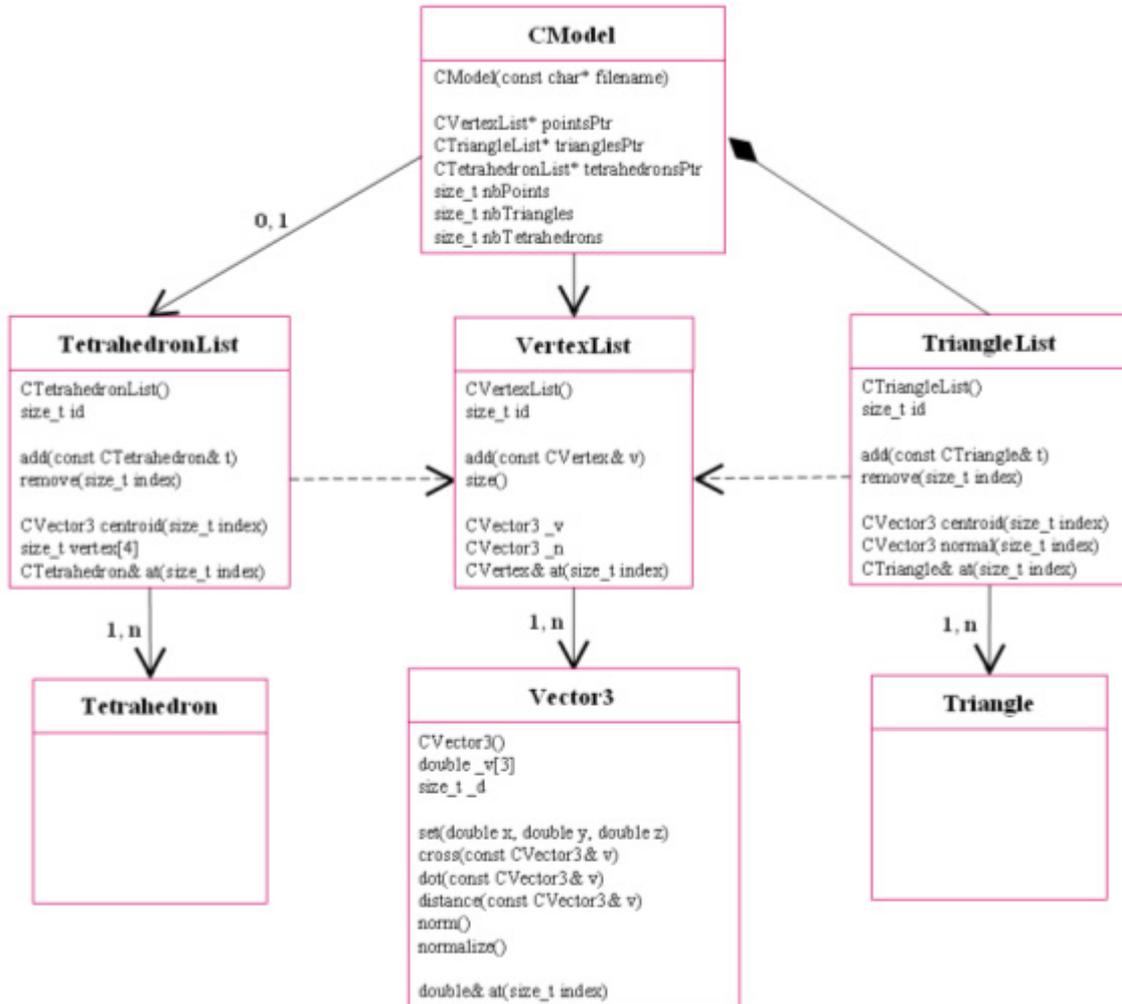


Figure A5.152 : Diagramme des classes d'un modèle

3. Diagramme des classes de la détection de collision

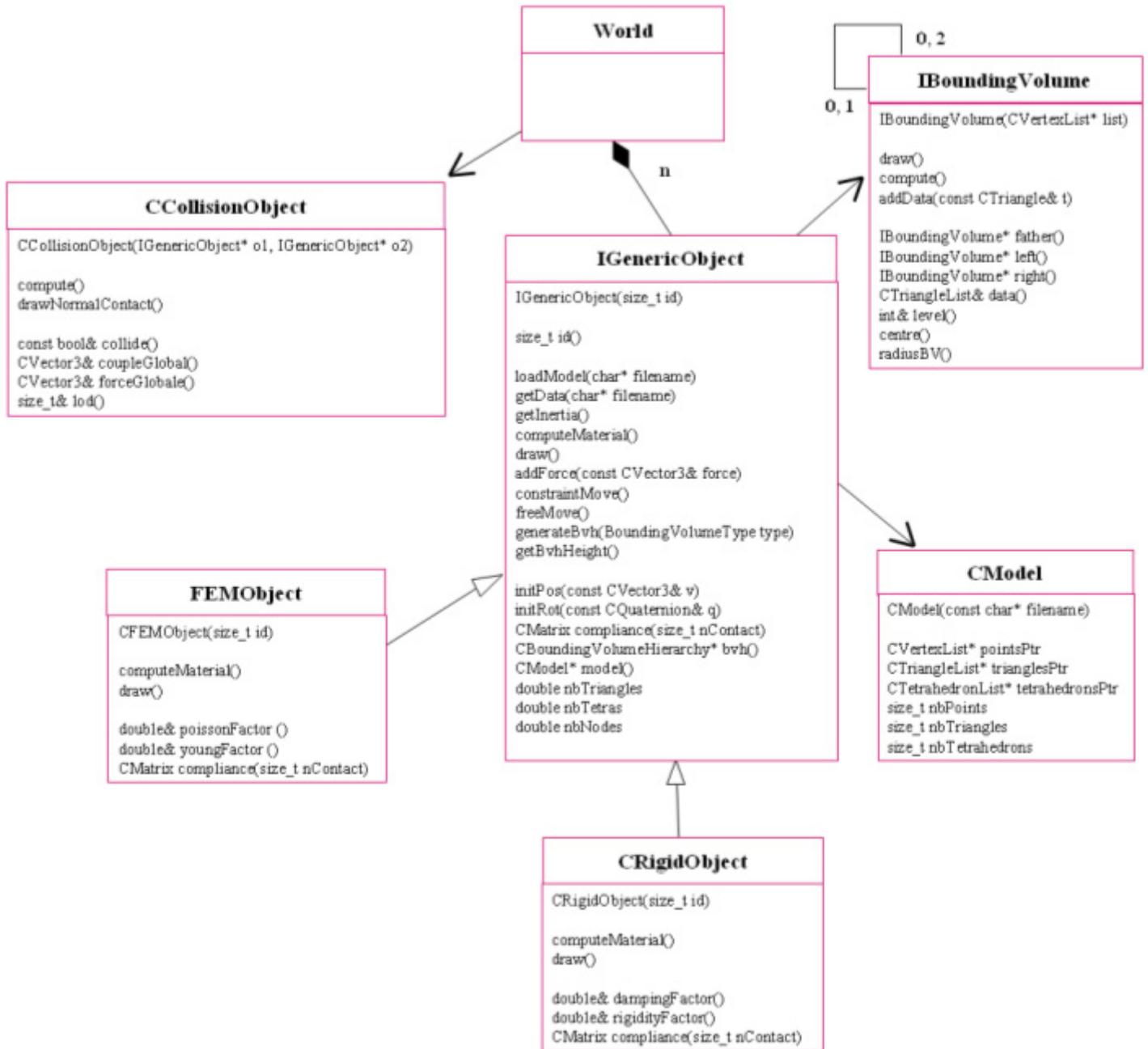


Figure A5.153 : Diagramme des classes de la détection de collision

PROPOSITION D'UN MODELE DE MAIN POUR LA SIMULATION DES INTERACTIONS HOMME-MACHINE EN ENVIRONNEMENT VIRTUEL. APPLICATION A LA PREVENTION DES RISQUES D'ACCIDENTS.

Résumé :

Les techniques de la Réalité Virtuelle permettent d'interagir avec des modèles et des environnements proches de la réalité en toute sécurité. Dans ce travail de recherche, nous proposons un modèle de main afin de simuler les interactions Homme-machine en environnement virtuel.

Cette main virtuelle est un modèle multicouches constitué d'un squelette rigide, d'une chair et d'une peau déformables. Le squelette est articulé par 26 degrés de liberté. Il est piloté par les données issues de la capture de mouvements à l'aide d'asservissements en position et en vitesse de type proportionnel-dérivé. Cela équivaut à un couplage virtuel de type ressort-amortisseur entre la main de l'utilisateur et la main virtuelle.

La chair volumique est modélisée par des coussinets sphériques déformables fixés au squelette rigide. Ils sont discrétisés en éléments finis tétraédriques. Nous imposons l'hypothèse des petites déformations et nous intégrons un modèle élastique linéaire pour simuler les déformations des doigts en contact. La peau est représentée par un maillage surfacique.

Nous avons également couplé ce modèle de main à un simulateur de presse plieuse réalisé par l'Institut National de Recherche et de Sécurité (INRS). Grâce aux mains virtuelles l'utilisateur est intégré à la scène virtuelle. Ainsi, il peut manipuler des objets et réaliser le pliage de tôles métalliques dans la plateforme de Réalité Virtuelle en temps réel. De plus, grâce à l'intégration du modèle de situation de travail et de l'indice de risque dynamique développés par l'INRS, il est possible d'évaluer le degré de dangerosité de la situation de travail et d'en informer l'utilisateur en temps réel.

Les résultats présentés dans ce travail ont donc conduit à trois contributions principales. La première est de permettre à l'utilisateur de saisir et de manipuler avec une main rigide des objets virtuels en temps réel et de manière stable dans des scènes complexes. La deuxième contribution est de proposer un modèle déformable éléments finis dédié à des tâches de préhension en environnement virtuel qui assure des déformations des doigts physiquement et visuellement réalistes. Enfin, la dernière contribution consiste à informer l'utilisateur du risque encouru lors de la simulation de tâches manuelles en environnement virtuel. Cette main virtuelle offre donc la possibilité de simuler et de mieux appréhender les futures interactions Homme-machine tout en tenant compte des aspects de sécurité. Au final, elle se révèle être une interface entre les métiers de la conception et de la prévention des risques.

L'ensemble des modèles proposés dans le cadre de ces travaux de recherche offre des perspectives intéressantes de développements futurs en matière de simulation d'interactions Homme-machine en environnement virtuel.

Mots-clés :

Réalité Virtuelle, Main virtuelle, Préhension, Modélisation basée sur la physique, Interaction Homme-machine, Capture de mouvements, Asservissement, Méthode des Eléments Finis, Prévention des Risques, Sûreté, Equipements de travail.

A HUMAN HAND MODEL TO SIMULATE HUMAN-MACHINE INTERACTIONS IN VIRTUAL ENVIRONMENTS. APPLICATION TO RISK PREVENTION

Abstract :

Virtual Reality techniques allow us to interact with life-like models and environments in safety. In this work, we propose a Human hand model in order to simulate Human-machine interactions in virtual environments.

Our virtual hand is a multilayer model made up of a rigid skeleton with simple primitives, a deformable layer for the soft tissues and a deformable mesh for the skin. It is articulated by 26 DOFs. The skeleton is controlled by motion capture data thanks to a virtual coupling that acts like a spring and damper between the user's hand and the virtual hand.

The soft tissues are modelled by spherical deformable pads fixed on the underlying skeleton. They are discretized by finite elements. We assume that we simulate finite deformations and we integrate a linear elastic model for the behaviour law. The displacements of the nodes of the mesh are given by the behaviour law and linear interpolation functions. Moreover; skin is modelled by a surface mesh.

We have coupled our virtual hand with a virtual press-brake developed by the French Institute of Research and Security (INRS). It allows the user to interact in real time with the virtual complex environment. Objects manipulation or bending tasks are simulated in the Virtual Reality platform of the INRS. Thanks to the integration of the work situation model and the dynamic risk index implemented by the INRS, we are able to estimate the degree of hazard of a given situation and by the way to inform the user about the risk in real time.

Results present three main contributions. The first one allows the user to grasp and manipulate virtual objects in real time thanks to a rigid virtual hand and motion capture data. The second one proposes a deformable model dedicated to grasping tasks in virtual environment. Deformable pads are modelled by linear finite elements. They simulate physically-based deformations of the fingers under contact while grasping. Finally, the last contribution consists in informing in real time the user of the degree of hazard of a given situation. Our virtual hand offers new possibilities to estimate the future Human-machine interactions while respecting safety issues. As a conclusion, our Human-hand model can be considered as an interface between design and risk prevention.

The whole of the developed models in this research offers interesting future issues to address in the case of Human-machine interactions in virtual environments.

Keywords :

Virtual Reality, Virtual Hand, Grasping, Physically-based modelling, Human-machine Interactions, Motion Capture, Control, Finite Element Method, Risk Prevention, Safety, Work facilities

Discipline : Sciences de l'Ingénieur