HAL
archives-ouvertes.fr

Esquisse virtuelle en conception mécanique

Christophe Bascoul

▶ **To cite this version:**

Christophe Bascoul. Esquisse virtuelle en conception mécanique. Mécanique [physics.med-ph]. Université Blaise Pascal - Clermont-Ferrand II, 2007. Français. <NNT : 2007CLF21775>. <tel-00718284>

HAL Id: tel-00718284

https://tel.archives-ouvertes.fr/tel-00718284

Submitted on 16 Jul 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : D.U : 1775
EDSPIC : 304

L'UNIVERSITE BLAISE PASCAL – CLERMONT II

Ecole Doctorale

Sciences pour l'Ingénieur de Clermont Ferrand

Thèse

Présentée par

Christophe BASCOUL

pour obtenir le grade de

Docteur d'Université

Spécialité : Génie Mécanique

L'esquisse virtuelle en conception mécanique

Soutenue publiquement le 28 octobre 2007 devant le jury :

Mme COQUILLART Sabine	Présidente
Mr DUMONT Georges	Rapporteur
Mr LEON Jean-Claude	Rapporteur
Mr GOGU Grigore	Examinateur
Mr FISCHER Xavier	Examinateur
Mr RAY Pascal	Directeur de thèse

Laboratoire de Mécanique et Ingénieries,
Institut Français de Mécanique Avancée et Université Blaise Pascal

Remerciements

Le travail que j'ai réalisé dans le cadre de cette thèse est avant tout un travail solitaire mais qui n'aurait pu aboutir sans l'aide et le soutien de tous.
Mes deux premières années de thèse se sont déroulées au sein du LIREST de Cachan sous la direction de Jean-Louis MARTINAND, mon sujet de thèse ayant au départ un lien important avec les effets de l'introduction de nouvelles technologies dans la didactique de la conception mécanique. Je dois remercier Jan-Louis MARTINAND pour son accueil chaleureux au sein du LIREST et Didier MARQUIS qui est à l'origine de mon inscription dans ce laboratoire et aussi de la possible décharge à mi-temps pour la réalisation de ma thèse. Didier est aussi la personne qui a su comprendre que le moment était venu pour moi de me confronter à la recherche, c'est avec lui que j'ai élaboré le sujet de ma thèse.
J'ai réintégré le LAMI pour mes deux dernières années de thèse, grâce à Pascal RAY qui a accepté d'être mon directeur de thèse. Pascal a eu le courage de me pousser au bout de ce travail et c'est à mon sens, le plus difficile à réaliser : écriture du rapport, composition du jury, préparation de la soutenance, etc. J'insisterais juste sur la composition du jury, l'honneur qui m'est fait de soutenir ma thèse devant un jury d'une telle qualité est une des conséquences de l'énorme travail réalisé par Pascal à l'IFMA, au LAMI, dans l'AIP-PRIMECA, etc. Encore merci pour cette expérience (même si j'ai encore beaucoup de progrès à faire).
Mon intégration au LAMI, s'est faite dans l'équipe MMS (Machines, Mécaniques et Système) actuellement orchestrée par Gregore GOGU. Nous avons mis en place tous les deux, une unité de valeur, Méthodes d'innovation et de Conception. J'ai aussi eu la chance de participer au développement de son cours à distance sur l'innovation. Gregore m'a toujours ouvert sa porte en cas de questions sur la physique ou sur la métaphysique, qu'il en soit cordialement remercié.
Jean-Claude LEON et Georges DUMONT, les deux rapporteurs de ma thèse ont eu la patience de faire une lecture approfondie de mon rapport. J'espère avoir pris en compte la majeure partie de leurs remarques, toutes très pertinentes, lors des modifications.
Sabine COQUILLART doit être particulièrement remerciée, en premier lieu parce qu'une partie de mon travail sur la dynamique sous contrainte a pour origine un travail de thèse qu'elle a encadré. Ensuite et plus généralement pour le travail qu'elle effectue pour le développement de la réalité virtuelle en France.
Xavier FISCHER m'a aussi aidé à plusieurs titres, en conception par exemple, les travaux sur la conception interactive à l'ESTIA ou à l'ENSAM de Bordeaux sont totalement en phase avec mes travaux. Ensuite il me paraît important de le remercier une nouvelle fois pour l'énorme travail qu'il fournit dans le cadre de la mise en place de la conférence internationale Virtual Concept et du journal IJIDeM.
Encore un grand merci à l'ensemble des membres de mon jury pour l'intérêt porté à mon travail.
Bien sur, je remercie l'ensemble de mes collègues IFMA ou LAMI, principalement ceux qui ont eu à supporter ma demi-décharge, tous ceux qui prennent régulièrement le café au cercle, les collègues de langue qui m'ont donné les coups de pouce bien utiles pour l'oral et l'écrit. Emmanuel Mesnard, collègue de l'ISIMA, qui m'a permis de profiter du travail en projet de ses étudiants, futurs ingénieurs en informatique, sans eux les threads seraient encore un simple concept pour moi.
Les étudiants que j'ai suivis en projet sur l'utilisation de la réalité virtuelle en mécanique, m'ont aussi permis d'avancer en confrontant mes visions théoriques à un pratique sans faille. Ils sont pour une bonne part à l'origine de mon sujet de thèse.
Ces quatre années n'ont cependant pas toujours été très faciles à gérer, les amis, la famille sont là, toujours près à vous encourager et c'est pourquoi, je dois du

fond du coeur remercier mes amis et ma famille proche, surtout mes parents.
Ma fille, Aurélia, doit recevoir toutes les excuses d'un père qui n'a pas toujours pris le temps dans une période où elle non plus n'en avait pas. Je n'oublie pas son compagon, Philippe, qui m'a prouvé que le golf n'était pas que de la détente (on reste dans la dynamique avec choc).
Je termine ces remerciements par ma femme, Annie, artiste plasticienne, qui m'a donné l'occasion de découvrir un monde très éloigné de la mécanique. Cette découverte explique en partie mon goût pour l'esquisse en tant que processus de création. Les travaux que j'ai réalisés pour elle ces dernières années touchent à des domaines très variés : vidéo, image de synthèse, installation d'oeuvres, etc. Le monde scientifique semble parfois faire preuve de rigueur, dans la qualité de rédaction d'un rapport par exemple, mais c'est en travaillant avec une artiste que j'ai compris ce que le mot rigueur voulait dire.

Table des matières

Introduction..... 13
1 Etat de l'art..... 17
1.1 L'esquisse..... 18
1.1.1 L'esquisse et la conception..... 19
1.1.2 Les différentes utilisations de l'esquisse..... 21
1.1.3 L'esquisse et les nouvelles technologies..... 22
1.1.4 L'esquisse et la cognition..... 25
1.2 La conception..... 26
1.2.1 Point de vue descriptif..... 26
1.2.2 Le point de vue prescriptif..... 29
1.3 La réalité virtuelle..... 35
1.3.1 Matériel..... 35
1.3.2 Logiciel..... 42
1.3.3 Les problèmes cognitifs..... 57
1.4 Les outils d'esquisse existants..... 58
2 Dynamique dans l'esquisse virtuelle..... 63
2.1 Dynamique sous contraintes..... 63
2.1.1 Rappel du principe de moindres contraintes de Gauss..... 64
2.1.2 Formulation d'Udwadia et Kalaba..... 65
2.1.2.1 Contraintes idéales..... 65
2.1.2.2 Contraintes non idéales..... 67
2.1.3 Éléments de calcul..... 67
2.1.3.1 Détermination des efforts aux points de contact..... 67
2.1.3.2 Calcul de la pseudo-inverse..... 68
2.1.3.3 Calcul de la racine de la matrice d'inertie..... 69
2.1.4 Essais de la dynamique sous contraintes en 2D..... 70
2.1.4.1 Ecriture des contraintes..... 71
2.1.4.2 Détermination des accélérations..... 72
2.1.4.3 Résultats du test en 2D..... 72
2.2 Les contraintes dans l'esquisse virtuelle..... 74
2.2.1 Obtention des contraintes..... 75
2.2.2 Géométries des contacts..... 76
2.2.2.1 Différents types de géométries..... 76
2.2.2.2 Différents types de contact..... 77
2.2.2.3 Contacts inutiles..... 77
2.2.2.4 Nombre de points de contact..... 78
2.2.3 Contraintes de glissement..... 78
2.2.3.1 Contraintes de glissement en translation..... 79
2.2.3.2 Contraintes de glissement en rotation..... 81
2.2.4 Contraintes de choc..... 82
2.2.4.1 Contraintes de choc en translation..... 82
2.2.4.2 Contraintes de choc en rotation..... 83
2.2.5 Contraintes géométriques..... 83
2.2.5.1 Contraintes géométriques de distance..... 84
2.2.5.2 Contraintes géométriques angulaires..... 84
2.2.6 Contraintes de type liaison..... 85
2.2.6.1 Contraintes en rotation dans une liaison..... 86
2.2.6.2 Contraintes en translation dans une liaison..... 87
2.2.7 Combinaison de contraintes..... 88
2.3 Conclusion sur la dynamique..... 89
3 Gestion des changements d'états..... 91
3.1 Détection de changement d'état..... 91

3.1.1 Recherche du temps de changement d'état....92
3.1.1.1 Définition de la forme de la trajectoire....93
3.1.1.2 Recherche du temps de contact....95
3.1.2 Déplacement jusqu'au contact....100
3.1.3 Recherche de changement d'état au contact....100
3.1.3.1 Nouveau choc....100
3.1.3.2 Modification d'un glissement....103
3.2 Application des nouveaux états....104
3.2.1 Gestion des nouveaux chocs....104
3.2.2 Gestion des glissements....105
3.3 Conclusion sur la gestion des changements d'états....105
4 Développements des outils d'esquisse virtuelle....107
4.1 Première version de l'outil....107
4.1.1 Les interactions avec l'utilisateur....108
4.1.2 Les périphériques....109
4.1.2.1 Main gauche....109
4.1.2.2 Main droite....109
4.1.2.3 Les yeux....110
4.1.3 Le logiciel....110
4.1.3.1 La gestion des périphériques....110
4.1.3.2 La gestion de l'affichage....112
4.1.3.3 La gestion du retour d'effort....113
4.1.3.4 La gestion de la dynamique....113
4.1.3.5 La gestion des tâches....114
4.1.4 Essais et conclusions....115
4.1.4.1 Mécanisme défini à main levée avec retour d'effort....115
4.1.4.2 Monte-charge avec ou sans jeu....116
4.1.4.3 Mécanisme plus complexe....117
4.1.4.4 Conclusion sur le premier outil....118
4.2 Deuxième version de l'outil....119
4.2.1 La structure matérielle....119
4.2.2 La structure logicielle....120
4.2.2.1 Nouvelle architecture logicielle....120
4.2.2.2 La tâche haptique....121
4.2.2.3 La tâche dynamique....122
4.2.2.4 Gestion des fichiers et interfaçage....125
4.2.3 Essais....125
4.2.3.1 Système bielle manivelle....126
4.2.3.2 Manipulation haptique d'un système parallèle....132
4.2.3.3 Exemple de conception avec l'esquisse virtuelle....136
Conclusion générale et perspectives....147

Table des matières des annexes

A.1 Définition des géométries....153
A.2 Hiérarchie de sphères....156
A.3 Arithmétique des intervalles....159
A.3.1 Fonctions de base....159
A.3.2 Fonctions spéciales....159
A.3.3 Hiérarchie d'intervalles....160
A.4 Format de fichier....162
A.4.1 mecanisme.dtd....162
A.4.2 BielleManivelle.xml....164
A.5 Courbes des tests....166
A.5.1 Descente du piston en fonction du jeu....166
A.5.2 Descente du piston avec des liaisons....166
A.5.3 Vitesse de rotation de la manivelle....167
A.6 Exemple de détermination des accélérations....169
A.6.1 Système bielle manivelle avec liaisons....169
A.6.2 Système bielle manivelle avec liaisons et choc....171

Index des illustrations

Figure 1.1 : Plan de la maison de la cascade....20
Figure 1.2 : Photo de la maison de la cascade....20
Figure 1.3 : Transporteur vertical dit Transcal....22
Figure 1.4 : Maquette du Transcal....22
Figure 1.5 : Simulation du fonctionnement du Transcal....22
Figure 1.6 : Du croquis à l'élément 3D en bibliothèque avec Sketch-VR [DO03]. 23
Figure 1.7 : Schéma animé par la physique qualitative après reconnaissance de forme [DA01]....24
Figure 1.8 : Moteur de recherche 3D avec entrée en 3D avec Teddy [FU01]....24
Figure 1.9 : Illustration tome1 Encyclopédie Diderot D'alembert.... 29
Figure 1.10 : Illustration de l'atlas des Eléments des machines de Bach.... 29
Figure 1.11 : Illustration du tome 3 du Manuel de l'ingénieur de Hütte.... 29
Figure 1.12 : Formalisation d'Ullman [UL02]....31
Figure 1.13 : Formalisation de Benami et Jin [BE02]....31
Figure 1.14 : Formalisation de Roy et Bharadwaj [RO02]....31
Figure 1.15 : Environnement fonction/comportement/structure situé [GE01].... 32
Figure 1.16 : Le modèle User/Product/Environment de Horvath et Van der Vegte 33
Figure 1.17 : Design For Assembly de Boothroyd Dewhurst, Inc....34
Figure 1.18 : Design For Manufacture de Boothroyd Dewhurst, Inc.... 34
Figure 1.19 : Utilisateur manipulant un objet....35
Figure 1.20 : Technologie amBX de Philips pours les jeux.... 36
Figure 1.21 : Vue fonctionnelle de l'utilisateur.... 36
Figure 1.22 : Boucle de contrôle sur les déplacements....36
Figure 1.23 : Projection dans le plan visuel....37
Figure 1.24 : Boucle de contrôle sur les efforts....37
Figure 1.25 : Projection sphérique de la société Elumens....38
Figure 1.26 : Projection conique sur miroir [BI01].... 38
Figure 1.27 : Ecran autostéréocopique avec son circuit intégré....39
Figure 1.28 : Contrôleur WOWvx OpenGL....39
Figure 1.29 : Lien entre les applications OpenGL et les écrans holographiques...40
Figure 1.30 : Phantom 3.0.... 41
Figure 1.31 : HapticMaster....41
Figure 1.32 : Spidar....41
Figure 1.33 : Structure générale d'un programme de réalité virtuelle.... 42
Figure 1.34 : Décomposition d'un système mécanique.... 42
Figure 1.35: Démonstration de démolition de château par Intel....46
Figure 1.36 : Système multi-corps de type chaîne [NA01].... 47
Figure 1.37 : Comparaison entre la formulation hamiltonième et la formulation de Lagrange en terme d'erreur [NA01]....47
Figure 1.38: Cube écrasé par une plaque [DR02]....48
Figure 1.39 : Approche corotationnelle globale développée par Duriez [DU02].. 48
Figure 1.40 : SOFA : Simulation Open Framework Architecture [AL03].....49
Figure 1.41: Obtention de forme avec les EDP volumétriques implicites [DU04] 49
Figure 1.42 : Téléphone portable modélisé avec MagicalSketch, visualisé avec un rendu non réaliste [KA02]....50
Figure 1.43 : Système de retournement de pièces....51
Figure 1.44 : Contraintes 3D sur un ensemble de pièces.... 51
Figure 1.45 : Simulation de l'utilisation d'une poubelle à papier modélisée par son squelette [HO01]....51
Figure 1.46 : Deux styles de cafetières. Source www.alessi.it....52
Figure 1.47 : Modification du profil du capot de la cafetière [DU01]....52

Figure 1.48 : Objet déplacé [PE02]..... 52
Figure 1.49 : Volume enveloppe généré [PE02]..... 52
Figure 1.50 : Comparaison entre deux modèles générés soit avec des B-Splines soit avec des triangles [PE02] 52
Figure 1.51 : Création d'une mêche de cheveu à partir d'un tracé à main levée [KI01]..... 53
Figure 1.52 : Génération de squelette hiérarchique à différents niveaux [CO01] .53
Figure 1.53 : Interface graphique du logiciel Synthetica [SU02]..... 54
Figure 1.54 : Projet lg3d..... 54
Figure 1.55 : Projet Wonderland..... 54
Figure 1.56 : Principe du WFS (Wave Field Synthesis)..... 55
Figure 1.57 : Panneaux MAP développés à l'Université de Delft..... 55
Figure 1.58 : Fonctionnement de SOFA sans CUDA [AL03]..... 56
Figure 1.59 : Fonctionnement de SOFA avec CUDA [AL03]..... 56
Figure 1.60 : Simulateur d'aide de première urgence développé par Guppyworks avec Ogre3D..... 57
Figure 1.61 : Application de déminage développée par Applied Research Associates, Inc. avec Delta3D..... 57
Figure 1.62 : Utilisation du système Reachin [SH01]..... 58
Figure 1.63 : Défaut du tracé haptique [SC04]..... 58
Figure 1.64 : Schéma d'un support de tasse 1 [DI03]..... 59
Figure 1.65 : Schéma d'un support de tasse 2[DI03]..... 59
Figure 1.66 : Proposition de travail en 3D dans le cadre du projet SmartSketches [BA02]..... 59
Figure 1.67 : Analyse des gestes des sculpteurs [BO01]..... 59
Figure 1.68 : Prototype du projet Touch and Design [BO01]..... 60
Figure 1.69 : Système de Yoshida et al.[YO03]..... 60
Figure 1.70 : Résultat brut à partir du sketcher [YO03]..... 60
Figure 1.71 : Sketch avant traitement [YO03]..... 60
Figure 1.72 : Sketch après traitement [YO03]..... 60
Figure 2.1 : Un point dans un cercle..... 71
Figure 2.2 : Glissement en translation..... 79
Figure 2.3 : Contrainte de glissement en rotation..... 81
Figure 2.4 : Contrainte de choc en rotation..... 83
Figure 2.5 : Paramétrage d'une liaison pivot glissant..... 86
Figure 2.6 : Glissement avec choc..... 88
Figure 2.7 : Double choc..... 88
Figure 2.8 : Choc avec perte de glissement..... 89
Figure 3.1 : Recherche du temps de contact de 0 vers dt..... 92
Figure 3.2 : Recherche de la perte de contact de dt vers 0..... 92
Figure 3.3 : Recherche de perte de contact par décollement..... 92
Figure 3.4 : Hiérarchie d'intervalles..... 95
Figure 3.5 : Hiérarchie d'intervalles sur un segment..... 96
Figure 3.6 : Déplacement de deux segments sur un intervalle de temps..... 96
Figure 3.7 : Projection des intervalles sur x et y..... 97
Figure 3.8 : Projection des intervalles sur z..... 97
Figure 3.9 : Navigation dans la hiérarchie d'intervalles..... 99
Figure 3.10 : Les 9 distances sur les intervalles de départ..... 99
Figure 3.11 : Sélec-tion des points les plus proches..... 99
Figure 3.12 : Division par 2 de la longueur des intervalles..... 99
Figure 3.13 : Nouvelle recherche..... 99
Figure 3.14 : 2 contacts sur 2 entités..... 101
Figure 3.15 : 2 contacts sur une entité..... 101
Figure 3.16 : Recherche géométrique du point de contact..... 101
Figure 3.17 : Un choc entre deux cercles..... 102
Figure 3.18 : Un premier point de choc trouvé..... 102

Figure 3.19 : Un deuxième point de choc trouvé..... 102
Figure 3.20 : Détermination du troisième point.....103
Figure 3.21 : Recherche de la nouvelle position du point de glissement..... 103
Figure 3.22 : Problème des chocs successifs.....105
Figure 4.1: Esquisse en 3D avec retour d'effort et orientation à la SpaceMouse.108
Figure 4.2 : Interactions utilsateur/sketcher.....108
Figure 4.3 : Périphériques Main gauche.....109
Figure 4.4 : Périphériques Main droite..... 110
Figure 4.5 : Périphériques occulaires..... 110
Figure 4.6 : Tracé dans le plan de clipping..... 111
Figure 4.7 : Animation des tracés.....111
Figure 4.8 : Gestion de l'affichage.....112
Figure 4.9 : Gestion du retour d'effort.....113
Figure 4.10 : Gestion de la dynamique..... 114
Figure 4.11 : Gestion des tâches..... 115
Figure 4.12: Création d'une boucle à l'aide du retour d'effort.....116
Figure 4.13 : Dessin à main levée avec retour d'effort.....116
Figure 4.14 : Monte-charge sans jeu..... 117
Figure 4.15 : Monte-charge avec jeu.....117
Figure 4.16 : Vue de côté en position de départ.....117
Figure 4.17 : Vue en perspective en position stabilisée..... 117
Figure 4.18 : Vérification de trajectoire verticale.....118
Figure 4.19 : Relaxation due au jeu.....118
Figure 4.20 : Poste de travail ergonomique.....119
Figure 4.21 : Nouvelle structure des tâches.....121
Figure 4.22 : Echantillonage de la force..... 121
Figure 4.23 : Choix périphérique haptique..... 122
Figure 4.24 : Mouse Spectre.....122
Figure 4.25 : Boucle dynamique.....122
Figure 4.26 : Complexité de la détection de collision continue.....123
Figure 4.27 : Détection de changement d'état.....124
Figure 4.28 : Recherche de la trajectoire..... 124
Figure 4.29 : Calcul des accélérations..... 124
Figure 4.30 : Définition des fichiers xml.....125
Figure 4.31 : Système bielle-manivelle.....126
Figure 4.32 : Position à atteindre après un double choc simultané..... 127
Figure 4.33 : Nouvelle recherche de trajectoire..... 127
Figure 4.34 : Allure du déplacement du piston sur un temps court.....128
La qualité de la simulation peut être montrée en laissant la simulation se dérouler et en vérifiant par exemple que les points extrêmes sont toujours atteints. Nous vous présentons ce type de courbe identique à celle ci-dessus pour seulement deux cycles (Fig. 4.36). La figure 4.35 n'est en fait qu'un très gros zoom du point de départ de la courbe (point en haut à gauche) de la figure 4.36..... 129
Figure 4.36 : Allure du déplacement du piston sur deux cycles.....129
Figure 4.37 : Déroulement des calculs matriciels.....130
Figure 4.38: Recherche de l'origine de l'hyperstatisme..... 131
Figure 4.39 : Système bielle manivelle avec choc.....131
Figure 4.40 : Evolution de la vitesse de rotation de la manivelle..... 131
Figure 4.41 : Système parallèle.....132
Figure 4.42 : Système en vue de face au départ.....133
Figure 4.43 : Système den vue de face an fin de mouvement.....133
Figure 4.44 : Système en vue de dessus au départ.....134
Figure 4.45 : Système en vue de dessus en fin de mouvement.....134
Figure 4.46 : Déplacement sur x de l'effecteur avec une force sur celui-ci..... 134
Figure 4.47 : Déplacement sur x de l'effecteur avec une force sur la pièce 1.....134
Figure 4.48 : Déplacement sur x de la pièce 1 avec une force sur la pièce 1.....134

Figure 4.49 : Système parallèle avant déplacement....135
Figure 4.50 : Position intermédiaire du système parallèle....135
Figure 4.51 : Position recherchée du système parallèle....135
Figure 4.52 : Premier essai d'interfaçage avec Catia....135
Figure 4.53 : Squelette seul....136
Figure 4.54 : Squelette avec surfaces....136
Figure 4.55 : Machine parallèle chinoise Links-Exe700 sous licence Exechon... 137
Figure 4.56 : Squelette de l'Exechon....137
Figure 4.57 : Agencement des pièces et des liaisons....137
Figure 4.58 : Boucle des deux bras symétriques de l'Exechon....138
Figure 4.59 : Débattement du système parallèle soumis à une force sur z.... 139
Figure 4.60 : Allure de la vitesse en translation sur z de la plateforme.... 139
Figure 4.61 : Allure de la vitesse en translation sur x de la plateforme....139
Figure 4.62 : Débattement du système parallèle soumis à un moment sur x....140
Figure 4.63 : Allure de la vitesse de rotation sur x de la plateforme....140
Figure 4.64 : Allure de la vitesse en translation sur x de la plateforme....140
Figure 4.65 : Manipulateur cylindrique....142
Figure 4.66 : Début de définition de la plateforme dans la CAO....143
Figure 4.67 : Visualisation avec VRML....143
Figure 4.68 : Volume enveloppe de la plateforme, vue 1....144
Figure 4.69 : Volume enveloppe de la plateforme, vue 2....144
Figure 4.70 : Volume enveloppe dans l'outil d'esquisse virtuelle....144
Figure 4.71 : Vue 3D de l'intérieur de la structure de l'Exechon.... 144
Figure A.1 : Paramétrage du Point2D....153
Figure A.2 : Paramétrage du Segment2D....153
Figure A.3 : Paramétrage du Cercle2D....153
Figure A.4 : Paramétrage du Point3D....154
Figure A.5 : Paramétrage du Segment3D....154
Figure A.6 : Paramétrage du Cercle3D....154
Figure A.7 : Paramétrage du Plan3D....155
Figure A.8 : Paramétrage du Cercle3D....155
Figure A.9 : Paramétrage de la Sphere3D....155
Figure A.10 : Paramétrage du Cone3D....155
Figure A.11 : Recherche de la plus courte distance....156
Figure A.12 : Premier niveau de la hiérarchie de sphères....156
Figure A.13 : Deuxième niveau de la hiérarchie de sphères....157
Figure A.14 : Troisième niveau de la hiérarchie de sphères....157
Figure A.15 : Quatrième niveau de la hiérarchie de sphères....157
Figure A.16 : Dernier niveau de la hiérarchie de sphères....158
Figure A.17 : Arbre binaire....158
Figure A.18 : Déplacement du piston avec des liaisons sur un temps court.... 167
Figure A.19 : Déplacement du piston avec liaisons sur deux cycles.... 167
Figure A.20 : Vitesse de rotation de la manivelle sans choc....167

Index des tables

Tableau 1 : Comparaison de stratégies entre un concepteur novice et un concepteur expert [PO01]..... 28
Tableau 2 : Classification par Blundell et Schwartz [BL02].....38
Tableau 3 : Classification des systèmes à retour d'effort utilisables pour l'esquisse virtuelle.....41
Tableau 4 : Evolution des chocs en fonction du nombre de cycles..... 73
Tableau 5 : Evolution des chocs en fonction du nombre de cycles avec un contact ressort/amortisseur.....73
Tableau 6 : Evolution des chocs et glissements en fonction du coefficient d'absorption.....74
Tableau 7 : Différents types de contact..... 77
Tableau 8 : Différents types de contact utiles.....77
Tableau 9 : Nombre de points de contact..... 78
Tableau 10 : Liaisons entre deux pièces.....85
Tableau 11 : Les trois types de changement d'état pour un point de contact.....92
Tableau 12 : Evolution des deux segments au cours du temps.....97
Tableau 13 : Recherche du minimum de distance.....99
Tableau 14 : Mauvaise gestion du double choc simultané..... 127
Tableau 15 : Allure du déplacement du piston en fonction du jeu..... 128
Tableau 16 : Evolution du système bielle manivelle avec jeu au cours du temps 128
Tableau 17 : Evolution de la vitesse de rotation de la manivelle en fonction du coefficient d'absorption.....132
Tableau 18 : Variation de l'écartement des bras du système parallèle..... 141
Tableau 19 : Comparaison de l'allure des courbes de la vitesse de rotation de la plateforme 141
Tableau 20 : Evolution du déplacement du piston en fonction du jeu..... 166
Tableau 21 : Evolution de la vitesse de rotation de la manivelle en fonction du coefficient d'absorption..... 168

Introduction

L'esquisse virtuelle est un outil de conception de systèmes mécaniques qui doit permettre de modéliser et de simuler plus rapidement des objets contenant des pièces en mouvement. Le terme esquisse sous entend que la création et la manipulation doivent être aussi simples que l'utilisation d'un crayon et d'une feuille de papier pour la création de schémas. Nous devons par contre apporter par rapport à cette version de l'esquisse, démocratisée sous la Renaissance, la visualisation et la manipulation de l'objet en trois dimensions, la simulation dynamique permettant de vérifier, en temps réel, le fonctionnement de l'objet.

Motivations

Le travail réalisé au cours de cette thèse est l'aboutissement d'une réflexion personnelle [BA08] qui dure depuis plusieurs dizaines d'années sur la conception de systèmes mécaniques et le manque d'outil support pour celle-ci. Une réflexion sur le pourquoi de cette inertie doit être envisagée et des réponses en terme de démarches et d'outils doivent être proposées. Notre travail s'inscrit dans ce cadre.

Un des éléments de réponse concernant l'inertie pour la production de nouveaux outils est la complexité des outils actuels qui demandent l'intervention d'un nombre important de collaborateurs de plus en plus spécialisés. Le spécialiste de simulation dynamique imposera l'utilisation de la méthode de Lagrange, l'informaticien trouvera naturel d'utiliser un modèle géométrique polyédrique, l'ergonome cogniticien définira l'interface graphique, etc. Le résultat est que le concepteur de systèmes mécaniques se retrouve face à un outil qui affiche de belles images mais qui ne correspond pas à ses besoins. Une de nos motivations est d'éviter cette impasse en réalisant un outil très simple dans les fonctionnalités tout en gardant à l'esprit l'utilisation finale de celui-ci.

Nous répondons à ces différents points en fournissant un état de l'art que nous voulons large sur l'esquisse virtuelle et en essayant de présenter en profondeur les méthodes utilisées avec de nombreux exemples. Certains choix effectués lors de ce travail, nous semblent mal connus et peu utilisés en conception, nous souhaitons que la lecture de ce document questionne et apporte, si possible, des réponses.

Démarches et méthodes

Notre réflexion sur l'esquisse virtuelle datant de plus d'une dizaine d'années, les objectifs à atteindre sont clairs et précis mais pas le chemin à parcourir.

Deux grandes phases ont séquencé notre travail, toutes les deux donnant lieu à un prototype correspondant à un des objectifs majeurs à atteindre :

- Dans une première phase, notre objectif est de montrer la nécessité d'utiliser le retour d'effort pour dessiner des objets dans l'espace et pour pouvoir ensuite les animer. Le premier prototype a validé cette hypothèse et nous a permis de nous confronter au développement pour la réalité virtuelle. Afin de limiter celui-ci, nous avons fait des choix qui bien qu'opérationnels, nous ont fait ressortir certains problèmes pour une utilisation efficace du système.
- La deuxième phase a pour but de faire cette fois les bons choix sur tous les points critiques. L'état de l'art et les quelques problèmes détectés dans le premier outil, nous ont aidé à effectuer une analyse plus fine de chaque point, d'où de nouveaux développements importants dans des domaines de la dynamique et de la détection de collision. Chaque nouveau développement a été testé séparément avant d'être intégré au deuxième prototype.

L'évolution rapide des technologies est un élément qui freine l'évolution des outils de conception. Cette idée peut paraître antinomique mais la complexité des outils de développement est telle qu'il est possible de passer plus de temps sur l'apprentissage de ceux-ci que sur la mise en pratique des prototypes développés pour valider nos essais. Nous avons fait le choix de suivre l'évolution des outils de développement tout au long de la durée de la thèse, dans le cas contraire, nous aurions couru le risque d'une incompatibilité de l'outil développé avec d'autres librairies ou logiciels. Un exemple actuel est l'utilisation du formalisme XML pour le format des fichiers de conception.

Un dernier point qui a conduit nos choix est le fait d'effectuer des développements réutilisables et ne nécessitant pas de librairies propriétaires. Nous estimons qu'il est primordial que notre travail puisse être réutilisé, critiqué et complété. Nous avons nous-mêmes bénéficié du travail d'autres chercheurs en réutilisant leurs librairies (OpenDE : www.ode.org, Haptik Library : www.haptiklibrary.org, Jthread : freshmeat.net/projects/jthread/, etc.) et nous avons aussi évité tout développement spécifique ainsi que l'usage exclusif de périphériques spécifiques, chers et encombrants.

Objectifs

Dans ce travail, nous cherchons à montrer comment l'esquisse virtuelle peut être une aide pendant les phases avancées de conception de systèmes mécaniques. Le

titre même de la thèse mérite quelques précisions :

- le terme esquisse est à rapprocher du terme disegno [CI01] qui est né pendant la renaissance en Italie. C'est l'arrivée du papier qui a permis aux peintres d'utiliser plus systématiquement l'esquisse comme outil de création et de préparation du travail. Ciaravino indique que « *cet instrument qui entend cerner l'idée dès sa première manifestation (Alberti écrit que le dessin est un tracé uniforme conçu dans l'esprit, « prescriptio concepta animo ») et s'enrichit d'un nouveau potentiel intellectuel, sert bien les intentions des artistes : expliquer le rapport de la main et de la pensée et faire valoir une nouvelle réflexion théorique derrière chaque construction matérielle. Voilà comment le dessin devient disegnio.* ». Le terme d'Alberti : « prescriptio concepta animo » nous semble très bien refléter ce qu'est un outil d'esquisse virtuelle ;
- La conception de systèmes mécaniques implique la prise en compte du mouvement des pièces. Ce point est celui qui différencie le plus notre travail des autres recherches sur l'esquisse virtuelle. Il faut alors chercher à simuler les déplacements, les efforts, les collisions entre les pièces, bref le comportement réel d'un mécanisme ;
- Les phases avancées de conception (conceptual design) sont les phases les plus critiques puisque dans ces phases la quasi totalité du coût du produit est définie. Par contre, les données du problème à traiter sont floues voir inexistantes. Il faut donc que les modèles choisis pour représenter l'esquisse soient eux aussi les plus simples possibles. Toute définition sur la géométrie contraint les autres champs de la conception intégrée (usinage, assemblage, ...). La tendance qui consiste à vouloir rapidement obtenir une belle géométrie de la pièce doit être évitée mais il faut cependant que la représentation soit compréhensible ;
- Aider le concepteur dans son travail, c'est essayer de savoir comment travaille celui-ci. De nombreux travaux permettent de nous éclairer. La notion de charge cognitive est très souvent mise en avant dans la compréhension des schémas par exemple. Un homme n'a en effet qu'une très faible capacité de traitement, toute surcharge cognitive doit donc être éliminée. Reste à savoir ce qui est considéré comme surcharge cognitive dans une tâche donnée. Des essais d'utilisation d'écran autostéréoscopique auprès de concepteur de chez Renault, ont prouvé, par exemple, un gain cognitif pour les tâches de résolution propres à la conception [CA01].

Plan de la thèse

Nous présentons au chapitre 1 un état de l'art que nous avons souhaité large et précis. Nous avons choisi de présenter trois champs principaux formant les piliers de l'esquisse virtuelle : l'esquisse, la conception et la réalité virtuelle. Les interconnections entre ces trois champs seront elles aussi présentées. Nous concluons cet état de l'art par un panorama des systèmes existants d'esquisse virtuelle.

Le chapitre 2 présente la dynamique sous contraintes que nous avons mis en oeuvre ainsi que le mode d'obtention de ces contraintes pour les chocs, les glissements et les liaisons. Nous montrerons pourquoi cette dynamique sous contraintes unifie la dynamique qui nous permet toutes les manipulations d'objet et la gestion des contraintes.

La dynamique sous contraintes nous permet d'obtenir les accélérations de toutes les pièces du mécanisme crées ou manipulées, il faut à partir de celles-ci déterminer les trajectoires de ces pièces et en déduire si de nouveaux contacts apparaissent ou si certains disparaissent. Nous avons pour cela mis au point un système de gestion des changements d'états du mécanisme présenté au chapitre 3 qui détermine à chaque pas de temps les contacts entre les pièces, leur position et leur type (choc ou glissement).

Deux outils d'esquisse virtuelle ont été développés, le premier nous a permis de vérifier la nécessité du retour d'effort pour le dessin dans l'espace et de montrer les limites des modèles choisis. Un deuxième a été mis au point et nous l'avons testé sur plusieurs cas de conception ou de manipulation. Les détails sur le développement de ces deux outils sont donnés au chapitre 4.

Une conclusion générale sur le résultat de ces années de travail sera proposée ainsi que les perspectives multiples et très motivantes de l'esquisse virtuelle.

1 Etat de l'art

L'état de l'art se décomposera en quatre parties :

- l'esquisse,
- la conception,
- la réalité virtuelle,
- les outils d'esquisse virtuelle existants.

Nous avons cherché dans cet état de l'art à définir les différents termes inclus dans le titre de la thèse : l'esquisse virtuelle en conception mécanique. Les trois champs qui nous serviront de base que sont l'esquisse, la conception et la réalité virtuelle débordent largement notre sujet d'étude. Nous avons cependant trouvé plus utile de déborder du cadre strict pour déterminer les limites de celles-ci et pour en déduire des pistes possibles d'extension. Les champs, esquisse et conception, sont particulièrement développés pour une raison historique, l'esquisse et la conception sont pratiquées depuis des siècles. La réalité virtuelle est en constante évolution, nous avons essayé de faire un état de l'art qui ne soit pas obsolète trop rapidement en séparant le matériel du logiciel. Cette décomposition peut parfois être aberrante, une carte graphique par exemple, peut être classée dans le matériel mais aussi dans le logiciel. Des liens sont effectués entre les différents champs : esquisse/conception, esquisse/réalité virtuelle, etc. Des liens avec les aspects cognitifs sont aussi présents pour chaque champ.

Nous concluons cet état de l'art par une analyse des différents systèmes d'esquisse virtuelle existants.

1.1 L'esquisse

Historique

Il y a peu de traces anciennes d'esquisses. Elles n'avaient pas vocation à être conservées. Il faudra attendre Léonard de Vinci pour obtenir un nombre suffisant d'esquisses qui soient de vrais croquis de recherche et non des illustrations placées dans des livres. Comme l'indique Ciaravino [CI01] dans son livre sur le disegno, l'apparition de l'esquisse est liée à celle du papier d'où son utilisation massive à partir de la renaissance italienne.

L'histoire du graphisme technique a été étudiée mais les investigations sérieuses ne peuvent commencer qu'au XVIème siècle. Dans ces études historiques, on cherche à montrer les évolutions du dessin technique et plus récemment, quelle a été l'influence des nouvelles technologies (CAO en particulier) sur l'usage des différents graphismes disponibles.

Champs d'utilisation

L'utilisation de l'esquisse est sans limite c'est pourquoi nous allons nous restreindre à son utilisation dans le domaine de la création. Pour nous, une esquisse se situe dans un processus cognitif permettant la génération d'idées en vue d'une conception. Nous verrons que la majorité des travaux de recherche sont effectués dans le domaine de l'architecture et du design. Les ingénieurs de la renaissance connaissaient autant l'architecture que les arts graphiques, la conception de machine, l'anatomie, etc. Les champs de connaissance sont aujourd'hui beaucoup plus dissociés mais peut-on en conclure que l'utilisation de l'esquisse est différente entre un architecte et un ingénieur ? Quand on cherche à comprendre comment les architectes utilisent les esquisses, on ne peut que voir des similitudes avec celles réalisées par les ingénieurs. De plus, les ingénieurs sont de plus en plus des architectes dans des secteurs comme l'automobile et l'aéronautique et les architectes utilisent de plus en plus des formes complexes, de pièces en mouvement ce qui les rapprochent des ingénieurs. Nous élargirons donc notre champ d'étude à toute esquisse conduisant à une conception.

L'esquisse et les nouvelles technologies

A l'ère du numérique, le statut de l'esquisse est ambigu, on comprend mal qu'il soit encore nécessaire de travailler avec du papier et un crayon. Cette interrogation peut devenir une source importante d'innovation en conception. Nous allons faire une présentation sommaire de ces travaux en indiquant à chaque fois les points qui nous semblent importants par rapport à notre recherche propre. De nouveaux périphériques permettent de rendre plus naturelle la création d'esquisses numériques que ce soit en 2D ou en 3D. Nous essayerons de montrer le lien entre ces périphériques et l'esquisse.

1.1.1 L'esquisse et la conception

La question centrale récurrente est : l'esquisse est-elle un outil d'aide à la conception ?

Il est important de noter que le terme conception s'entend ici comme la recherche de principes de solutions, nous reviendrons ultérieurement sur le sens précis de ce terme .

Bilda, Gero et Purcell [BI04] montrent qu'un architecte expérimenté réalise des conceptions de même qualité en travaillant les yeux fermés ou en utilisant des esquisses. Ils concluent leur étude en indiquant : « Cette étude montre que l'utilisation d'esquisse n'est pas une activité essentielle pour les architectes expérimentés pendant les phases avancées de conception ». Cette conclusion est en contradiction avec certaines parties de leur article : « L'externalisation est nécessaire pour décharger la mémoire de travail visuo-spatiale et pour la même raison, les dessins et les diagrammes jouent un rôle important pour concevoir ». On peut aussi critiquer ces résultats puisqu'ils n'ont été effectués qu'avec trois architectes. Les éléments à concevoir étaient situés dans un plan ce qui limite la conception à des problèmes de placement de zones en 2D, les problèmes complexes d'architecture ont été occultés : création de volumes, prise en compte de la lumière et des effets de perspective. Pour évaluer les conceptions, à l'aveugle ou avec esquisse, tous les participants rendent une esquisse de leur projet, il y a donc en final à chaque fois utilisation du tracé pour obtenir le résultat.

Avant d'expliquer pourquoi l'esquisse est très importante en conception mécanique et pour d'autres domaines, nous citons une anecdote qui est à l'origine de recherches identiques à celle présentée ci-dessus. Cette anecdote décrit comment l'architecte Frank Lloyd Wright [MC03] a conçu la maison de la cascade (Fig. 1.2) pour Edgar Kaufmann. Wright s'était engagé à réaliser cette maison d'été mais n'avait pas le temps pour la réalisation des plans. Kaufmann lui a téléphoné pendant plus de 15 mois pour connaître la progression du travail et Wright lui indiquait à chaque fois que le travail avançait normalement mais aucun dessin n'était réalisé. Enervé, Kaufmann téléphona une dernière fois et indiqua qu'il allait arriver 1h30 plus tard dans les bureaux de Wright. Celui-ci dessina les plans très proches du résultat final en une heure. Wright écrivit en 1928 sur son processus de conception :

"*Concevoir le bâtiment dans l'imagination complètement, pas sur un papier mais dans l'esprit, avant de toucher au papier....Laisser le vivre là pour qu'il prenne graduellement des formes plus définitives avant de le soumettre à la planche à dessin. Quand les choses vivent en vous, commencez à les dessiner avec des ustensiles. Pas avant. Dessiner pendant la conception ou esquisser, comme nous disons, en expérimentant des ajustements pratiques à évaluer, n'est intéressant que si la conception est assez claire pour être sérieusement développée....Mais si le concept original est perdu pendant que le dessin est réalisé, il faut tout jeter et recommencer*".

Plusieurs remarques peuvent être faites sur cette vision de Wright :

- la maison de la cascade est un bâtiment assez complexe, multi-niveaux, mais les formes sont très cubiques, ce qui facilite une représentation plane (Fig. 1.1). On ne peut pas considérer que cette conception soit complexe pour la partie représentation de l'espace ;
- Wright était déjà à l'époque un architecte confirmé, on sait aujourd'hui que des zones graphiques du cerveau peuvent être utilisées pour des problèmes non graphiques après un apprentissage important. Les concepteurs novices

ont besoin d'une mémoire externe pour créer un bâtiment complet ;

– Les problèmes traités restent statiques sans prise en compte de mouvements possibles et encore moins d'effets liés à de la dynamique. Seule la dimension visuelle est gérée comme dans des problèmes de mémorisation de trajectoires à l'intérieur de bâtiment.

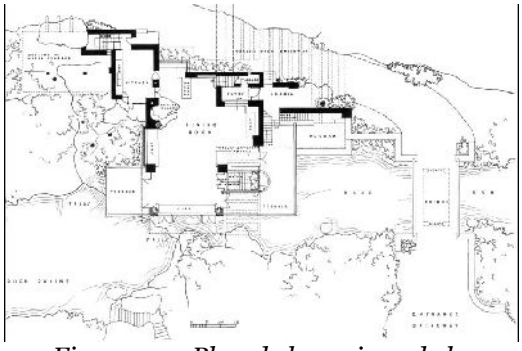


Figure 1.1 : Plan de la maison de la cascade

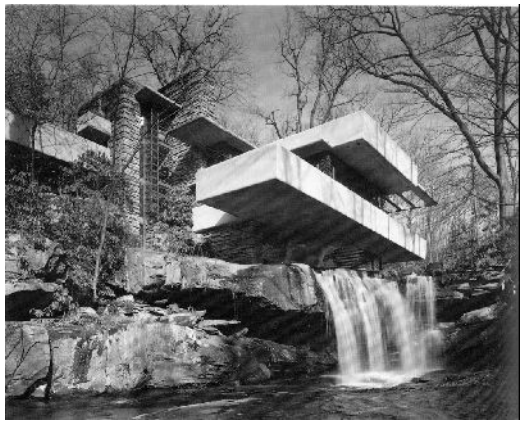


Figure 1.2 : Photo de la maison de la cascade

A partir de nombreuses autres études, nous pouvons affirmer que l'utilisation de la mémoire externe est indispensable pendant les phases avancées de conception et que l'esquisse est l'outil préféré des concepteurs comme support de la mémoire externe. Purcell et de Gero [PU01] proposent dans la revue Design Studies des états de l'art sur l'intérêt des schémas lors de la conception. Ils font le point sur les implications des sciences cognitives pour la compréhension des différents phénomènes liés à l'utilisation des schémas en conception [PU01]. Un point très important est fait sur les nouvelles connaissances sur la mémoire à court terme, la mémoire de travail, etc. qui permettent d'expliquer comment les schémas sont utilisés en tant que mémoire externe.

Pour essayer de démontrer plus formellement les relations entre l'esquisse et la conception, on peut aussi établir des liens entre la production de sketchs et la qualité du résultat de la conception [YA01].

Ullman, Wood et Craig [UL01] ont essayé de prouver l'intérêt de l'esquisse pour les concepteurs. Ils ont cherché à valider les 5 hypothèses suivantes :

– Hypothèse 1. Le dessin est la méthode préférée par les concepteurs de systèmes mécaniques pour la représentation externe des données ;

– Hypothèse 2. L'esquisse est une forme importante de représentation graphique comblant des besoins non remplis par le dessin ;

– Hypothèse 3. Le dessin est un langage imagé en extension nécessaire en conception mécanique. C'est une extension nécessaire des capacités cognitives du concepteur en dehors des représentations les plus triviales, de la propagation de contrainte et de la simulation mentale ;

– Hypothèse 4. Les dessins demandent des transformations de la mémoire du concepteur au support de la mémoire étendue. La nature de la transformation dépend des caractéristiques du médium ;

– Hypothèse 5. Les dessins utilisent et déterminent les unités cognitives (caractéristiques de conception) utilisées dans la formulation de l'image mentale. L'organisation de l'information cognitive du concepteur est ainsi

interdépendante des caractéristiques des dessins.

Cross [CR03] qui a beaucoup interviewé les concepteurs chevronnés a écrit sur le rôle du croquis dans la conception, nous allons citer plusieurs de ses commentaires. Le premier comme pour répondre à Wright, cite l'architecte Richard MacCormac : « *J'utilise le dessin comme processus de critique et de découverte* ». Cross commente ainsi cette phrase : « *Les concepts ébauchés dans les croquis le sont pour être critiqués, et non admirés; et ils font partie du travail de découverte et d'exploration qui constitue l'activité de conception* ».

Un autre extrait très important pour décrire l'utilisation à différentes échelles de l'esquisse : « *Le concepteur pense tout à la fois au concept global et aux détails de la mise en oeuvre du concept ; mais, bien sûr, pas à tous les détails, sinon il pourrait élaborer directement l'ensemble des dessins définitifs. Il utilise donc le croquis pour identifier et pour réfléchir sur les détails critiques* ».

On terminera les réflexions de Cross avec un extrait sur les liens qu'il établit avec les théories de la conception : « *Enfin, comme l'ont indiqué plusieurs théoriciens de la conception, les croquis facilitent la reconnaissance des aspects et des propriétés émergentes du concept de la solution. Ils aident à faire ce que Goel appelait des « transformations latérales » dans l'espace de la solution : le déplacement créatif vers de nouvelles alternatives. Ils facilitent ce que Goldschmidt appelait la « dialectique du croquis », le dialogue entre le « voir que » et le « voir comme », où le premier représente la critique réfléchie et le dernier renvoie au raisonnement analogique et à la réinterprétation du croquis qui, à nouveau, stimule la créativité. Et les croquis aident le concepteur à découvrir les conséquences inattendues, les surprises qui maintiennent la dynamique de l'exploration dans ce que Schön & Wiggins appelaient la « conversation réflexive avec la situation » qui caractérise la pensée du concepteur.* »

Les schémas peuvent aussi être utilisés pour suivre l'évolution de la conception pendant les différentes phases de celle-ci [RO01] d'où la nécessité de systèmes de gestion des versions pour les entrées à la main de type esquisse.

L'esquisse est donc un outil d'aide en conception de systèmes mécaniques et bien sur dans d'autres champs d'application.

1.1.2 Les différentes utilisations de l'esquisse

L'esquisse est utilisée dans de nombreux domaines : mécanique, architecture, mathématiques, arts graphiques, etc. 70 % des recherches effectuées sur l'esquisse, le sont dans le domaine de l'architecture [DO02][BI02]. Durant les phases avancées de conception, il existe de nombreuses similarités entre la conception architecturale et la conception de système mécanique. Les constructeurs automobiles parlent d'architecte moteur pour les ingénieurs responsables de l'agencement des différents éléments autour du moteur.

Les architectes utilisent encore souvent des vues 2D pour l'agencement des différentes pièces mais l'apparition de structures plus complexes et de nouveaux matériaux, permet de créer et de gérer des volumes en 3D. Cédric Lalemant, ancien élève de l'Ecole d'Architecture de Clermont-Ferrand, a développé un transporteur vertical nommé Transcal (Fig. 1.3). Les mouvements du Transcal et des utilisateurs sont tellement complexes qu'il est impossible de comprendre le

principe de fonctionnement sur un croquis en 2D. Cédric Lalemant a donc réalisé une animation en image de synthèse avec personnages (Fig. 1.5) et nous avons de notre côté réalisé une maquette du Transcal (Fig. 1.4).

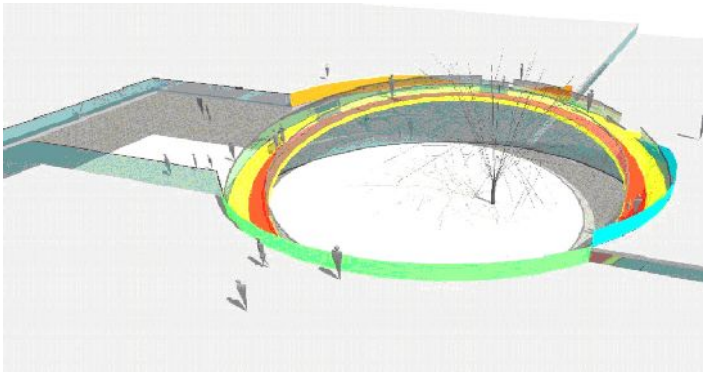


Figure 1.3 : Transporteur vertical dit Transcal

Figure 1.4 : Maquette du Transcal

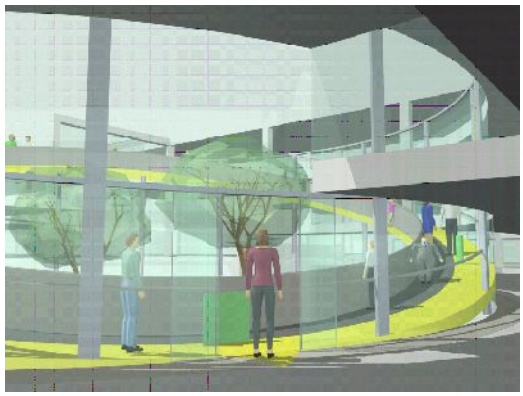


Figure 1.5 : Simulation du fonctionnement du Transcal

Cet exemple permet de comprendre pourquoi nous avons pris en compte les travaux effectués en architecture dans cet état de l'art.

La création en 2D des esquisses est donc plus liée au matériel utilisé qu'à la fonction finale de celles-ci. En dehors, des esquisses proches de l'écriture (schéma électrique, équation mathématique, etc.), ces graphismes devraient être directement créés en 3D mais il n'existe que peu de systèmes permettant de créer directement en 3D.

L'esquisse est donc un outil universel quelque soit le domaine d'application, il est malheureusement encore difficile de déterminer l'évolution de celle-ci, après sa migration vers la 3D.

1.1.3 L'esquisse et les nouvelles technologies

Le lien entre l'esquisse et l'informatique passe par des périphériques d'entrée principalement en 2D, souris, tablette à digitaliser, etc. Les points sont stockés ainsi que l'instant de stockage permettant ultérieurement une interprétation dynamique du mouvement du tracé. Une fois enregistrée l'esquisse peut être

réutilisée, analysée, transmise pour échange, etc.

Des essais ont permis de montrer que la possibilité de saisir directement les croquis [HO03] sur une tablette par exemple, améliore encore la puissance de ce moyen d'expression. Après une très petite phase d'adaptation (passage d'un mode relatif à un mode absolu), l'utilisation d'une tablette graphique devient beaucoup plus naturelle pour un graphiste que la simple manipulation d'une souris. Une des raisons supplémentaires est que ces tablettes sont sensibles à la pression exercée sur celles-ci. La dynamique d'un trait effectué avec une tablette permet de donner des effets de volume ou de déplacement sur un dessin statique en 2D. Il est important de remarquer qu'une tablette est un système à retour d'effort passif.

La reconnaissance de gestes lors de la création des croquis ou à la lecture de ceux-ci est importante. Une étude comparative a été effectuée dans le cadre du projet SmartSketches par Company et al. [CO03] sur 32 méthodes de reconstruction à partir d'une seule vue et 24 méthodes multi-vues. Tous les travaux se basent sur des tracés 2D [LI03][OH01][VA02] mais peuvent être étendus à la 3D [BI03][PE01][AL02]. En 2D, l'analyse des mouvements pendant le tracé permet de comprendre quels sont les détails importants dans un schéma: renforcement des lignes, ombrage, etc. [LI04]. Cet aspect dynamique de la reconnaissance n'est pas utilisé dans les méthodes de reconnaissance qui utilisent une grille de pixels. Dans ce cas le processus classique consiste à effectuer un pré-traitement de l'image, à réaliser une analyse polaire permettant de reconnaître les symboles quelque soit leur orientation puis à appliquer une série de classificateurs (distance de Hausdorff, distance de Hausdorff modifiée, coefficient de Tanimoto, coefficient de Yule, etc.) [BU01]. Plusieurs techniques sont utilisées pour transformer les points en courbes de différents types comme pour les polices de caractères, les courbes de Bézier [HE01] sont les plus utilisées. De nombreux travaux viennent de la reconnaissance d'écriture, La Viola a réalisé MathPad² qui relie l'écriture d'équation à la main avec de petites animations [LA02]. La reconnaissance de gestes est donc spécifique à chaque type d'utilisation mais il faut remarquer que plus les objets à reconnaître sont simples plus il est facile d'effectuer cette tâche. En architecture, des éléments comme les chaises sont représentés par de simples carrés en 2D, il est donc possible de placer une chaise en 3D sur le sol en lieu et place d'un carré, c'est ce qui est fait dans Sketch-VR [DO03] par exemple (Fig. 1.6).

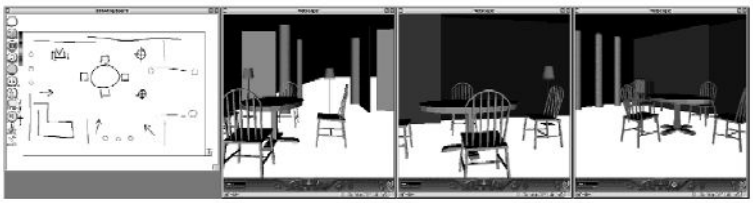


Figure 1.6 : Du croquis à l'élément 3D en bibliothèque avec Sketch-VR [DO03]

Il est logiquement plus simple de trouver un segment dans un tracé plutôt qu'une courbe de type NURBS que ce soit en 2D ou en 3D. Il faut donc rechercher les éléments juste nécessaires pour l'application visée.

Une fois la reconnaissance de forme réalisée, de nombreuses manipulations peuvent être effectuées sur les schémas, nous citons ici quelques applications possibles :

– Des couplages sont aussi mis en place avec des outils d'intelligence artificielle soit pour comprendre les intentions du concepteur [DO01], soit pour l'aider en lui proposant des alternatives, des améliorations, etc ;

– L'animation de croquis en utilisant la dynamique est rare, le MIT AI Lab travaille depuis plusieurs années sur un système de sketch en 2D [DA01][AL01] qui utilise de la physique qualitative pour l'animation. Dans la figure 1.7, la fenêtre de gauche correspond à l'entrée à main levée du schéma, le bouton Run permet de lancer l'animation qui apparait dans la fenêtre de droite ;

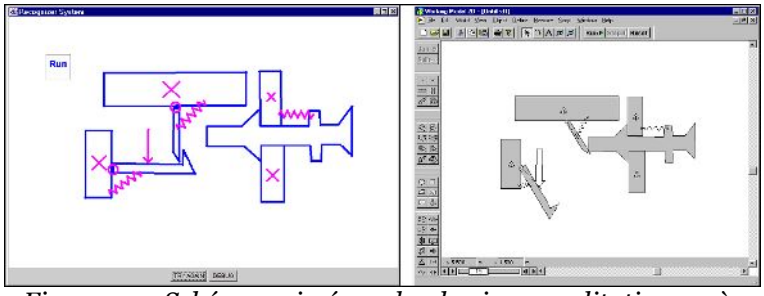


Figure 1.7 : Schéma animé par la physique qualitative après reconnaissance de forme [DA01]

– Les moteurs de recherche 3D utilisent souvent une entrée à main levée en 2D ou en 3D [FU01]. La figure 1.8 montre le résultat d'une recherche d'objets 3D à partir d'une entrée effectuée avec l'outil de sketch Teddy. Des bases de données sur les mécanismes anciens commencent à voir le jour, elles permettent d'obtenir des informations sur les cas d'utilisation, les avantages et inconvénients des différentes configurations. La digitalisation d'anciens documents est en cours dans des projets comme DMG-Lib [DO04] ;

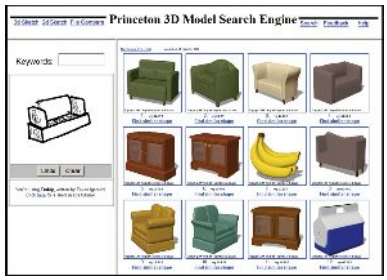


Figure 1.8 : Moteur de recherche 3D avec entrée en 3D avec Teddy [FU01]

– Le schéma comme moyen de communication lors de la conception en groupe [VA01] doit être souligné.

L'informatisation du tracé et de la visualisation de l'esquisse pose cependant encore des problèmes: certaines études tentent de répondre à ceux-ci en donnant des indications sur les fonctionnalités d'un système de sketch informatisé [VE01] ou en proposant, pour améliorer la visualisation d'un tracé dans l'espace, de projeter l'ombre sur le sol [CO04].

Les nouvelles technologies commencent dès à présent à faciliter le dialogue homme-machine en passant par l'esquisse qui peut ainsi être vue comme un système de tracking très fin qui dans notre cas devient dynamique et avec retour d'effort. L'intelligence du geste peut ainsi être captée.

1.1.4 L'esquisse et la cognition

L'esquisse est un formidable sujet d'étude pour les sciences cognitives. De multiples sens sont activés et par conséquent de nombreuses zones du cerveau. L'évolution rapide des sciences cognitives grâce, entre autres, à de nouveaux instruments d'investigation comme l'IRM, permettent de mieux entrevoir la complexité du raisonnement humain et le fonctionnement des différents sens.

Les cogniticiens nous aident à comprendre les difficultés lors de la création des esquisses ou lors de la visualisation de celles-ci: des études de systèmes mécaniques montrent les difficultés de compréhension des schémas [YO01]. Ils nous donnent aussi des éléments sur l'intérêt réel ou non de l'esquisse dans les phases de conception avancée.

Les études réalisées sur les esquisses ont certaines caractéristiques communes : étude d'esquisse en 2D, comparaison d'esquisse de novice et d'expert, lien de l'esquisse avec les autres supports :

- 95 % des études sur les esquisses sont effectuées sur les esquisses en 2D que ce soit sur papier ou sur outil informatique. L'objectif est ici de comprendre pourquoi un concepteur a besoin de faire des croquis [MC02] ? Est-il possible d'améliorer l'utilisation de ces croquis en les informatisant ? Comment peut-on faciliter la compréhension des schémas chez les étudiants ? Etc ;
- Une partie importante de ces études [ME03] consiste à comparer les croquis d'étudiants avec ceux de concepteurs confirmés. Jonson [JO01] essaye d'expliquer pourquoi les étudiants utilisent peu les croquis dans leurs projets. Pable [PA01] note que : « Il apparait, cependant, que le manque de capacité pour l'esquisse est toujours la première cause d'échec à l'examen de compétence d'architecture d'intérieur du NCIDQ ». Des études confirment que l'utilisation de graphique est bien meilleure que la simple utilisation de texte dans la phase de génération de concepts [MC01] ;
- Il est aussi possible d'étudier le lien entre l'utilisation d'esquisses et d'autres supports comme des documents sur d'anciens projets, des représentations CAO, etc. Ces études [GO02] permettent de comparer l'importance des différents supports les uns par rapport aux autres afin de comprendre quelles sont les informations à fournir au concepteur et à quel moment.

Nous pouvons remarquer que peu d'études existent cependant sur la génération ou la manipulation de l'esquisse directement dans l'espace. Les chercheurs qui travaillent sur l'esquisse virtuelle essaient de réduire la charge cognitive en supprimant la projection de l'espace sur un support en 2D. Nous souhaitons pour notre part après la mise au point de l'outil d'esquisse virtuelle, pouvoir vérifier certains de ces points grâce à des tests en situation en liaison avec une équipe d'ergonomes cogniticiens.

1.2 La conception

Les recherches sur la conception peuvent être rangées en deux grandes directions, une descriptive qui cherche à analyser ce qui passe réellement lors de la conception, l'autre prescriptive qui cherche à définir les méthodes de conception. Cette seconde direction a conduit à de nombreuses méthodes de conception, Pahl and Beitz[PA08], Rodenacker[RO03], Ullman[UL02], Dieter[DI04], Cross[CR04]. L'évolution historique de ces méthodes est des plus intéressantes, elle nous permet de comprendre les points importants à prendre en compte dans ces méthodes. On peut ainsi remarquer le poids de plus en plus fort des clients, la prise en compte des DFX (Design For X : Design For Assembly, Design For Manufacturing, Design For Cost, etc.), etc. Poussée à l'extrême, cette direction aboutit à la théorie de la conception (Design Theory [NE05]) qui peut apparaître comme un pur travail de laboratoire mais qui permet, en fait, de proposer de nouvelles formalisations.

La vision descriptive de la conception cherche à mieux comprendre comment le concepteur travaille, seul ou en groupe. Un ensemble de techniques est développé pour permettre l'analyse des traces laissées par le concepteur, sketchs, enregistrements sonores, vidéos, etc.

Nous allons faire une brève présentation des travaux actuels sur la conception en commençant par le point de vue descriptif ce qui nous permettra de justifier les éléments prescriptifs qui sont à retenir dans notre travail sur la conception de systèmes mécaniques.

1.2.1 Point de vue descriptif

Une partie importante des travaux cherchant à décrire ce qui se passe chez les concepteurs consiste à comparer les conceptions effectuées par des concepteurs confirmés et par des novices.

Les méthodes d'analyse

L'analyse du processus de conception est facilitée par l'utilisation du journal de conception, remis à la mode par Sobek [SO01] à l'Université du Montana. Des techniques plus lourdes sont mises en place, décomposant les activités cognitives [SU01], permettant ainsi une meilleure compréhension du processus complet. Des représentations graphiques des activités [CH02] pendant la conception permettent d'identifier certains problèmes.

L'analyse in-situ

L'ergonomie cognitive apporte une nouvelle vision du processus de conception

[VI02] qui se base sur le travail réel des concepteurs. Visser dans ce rapport de recherche critique la vision de la conception de Simon : S.I.P. (Symbolic Information-Processing), la compare avec une approche situationniste : S.I.T. (Situativity) pour aboutir à sa proposition sur les problèmes de conception. Sa critique de la conception vue par Simon doit être citée pour cerner le cadre de son étude. Cette critique se décompose en deux parties. Les points qui sont sous-estimés et ceux qui sont surestimés dans la vision de Simon :

- sous-estimation de la spécificité des problèmes mal définis et de leur résolution, du rôle de la construction de la représentation du problème, des liens indéterminés ;
- surestimation du rôle de la décomposition de problème, des analyses moyens-fins, du rôle de la recherche.

Ce qui permet à Visser de définir certaines caractéristiques des problèmes de conception : mal défini, ambigu, complexe, contraint, satisfaisant. La mauvaise définition des problèmes de conception est due au fait que les attentes des clients sont mal définies et évoluent au cours du temps mais le but à atteindre évolue aussi au fur et à mesure de la conception. L'ambiguité vient de l'impossibilité de comprendre le problème de conception de la même manière pour deux personnes différentes. La complexité est croissante par exemple dans le domaine automobile, ce qui rend la décomposition de problème de plus en plus difficile et implique de rechercher une indépendance des sous-problèmes. Les contraintes en conception ne sont pas toutes connues au départ puisque les informations sont mal définies. Aux contraintes classiques spatiales utilisées dans les outils de CAO, s'ajoutent des contraintes temporelles qui s'appliquent en particulier sur les actions mises en oeuvre lors de la résolution du problème de conception. La notion d'optimum en conception est illusoire et doit être remplacée par la notion de satisfaction plus tolérante et réaliste.

Pour l'élaboration des solutions conceptuelles lors de la génération de solutions, Darses, Détienne et Visser [DA04] indiquent que : « Selon les domaines, les outils CAO sont utilisés avec plus ou moins de succès pour ces phases. En électronique, où la conception (et la fabrication) de circuits intégrés et de cartes imprimées se fait principalement à partir de composants et de structures d'assemblage préexistants, ces outils peuvent être utiles dès les phases amont. Mais dès que l'on sort de la conception répétitive, la CAO n'est pas encore capable d'offrir une assistance effective à l'élaboration des solutions conceptuelles et n'est utilisée qu'une fois cette activité terminée. Cela vaut également dans les domaines de la mécanique, de l'architecture et du bâtiment. Les outils actuellement sur le marché permettant le travail d'esquisse ne fournissent pas d'assistance à l'élaboration de solutions conceptuelles, activité pourtant essentielle pour la conception. Seuls certains prototypes de recherche tentent d'implémenter des systèmes assistant les phases amont de la conception qui prennent en compte les spécifications que les études cognitives ont conduit à formuler et qu'on peut résumer ainsi :

- faciliter une entrée de données par mouvements de la main ;
- permettre l'entrée de données imprécises ;
- autoriser le passage entre niveaux et entre types de représentation ;
- aider à la comparaison entre différents concepts de solution ;
- faire des suggestions ;
- aider à évaluer des choix par un feedback au concepteur (par exemple, par des systèmes critiques). »

Ce travail d'ergonome cogniticien va nous permettre de critiquer les modèles prescriptifs et nous essayerons d'utiliser ces remarques lors de la réalisation de notre outil d'esquisse virtuelle.

L'analyse in-situ permet aussi de montrer comment de nouveaux outils comme SAP [CA03] peuvent réduire la créativité.

La comparaison entre un concepteur expérimenté et un novice

L'étude in-situ du travail de bons concepteurs est souvent présentée par Cross [CR01] mais est-il possible d'en déduire un modèle générique de conception ? On peut cependant conclure de ces études que les experts se focalisent sur les solutions plutôt que sur les problèmes, contrairement aux novices [CR02], qu'ils produisent des solutions de meilleure qualité, qu'ils passent plus de temps à la résolution de problèmes considérant plus d'alternatives et faisant plus de transitions entre les étapes de la conception [AT01].

L'expertise en conception permet de comprendre les différences dans les processus de conception utilisés par les novices ou par les concepteurs confirmés. Popovic a mis au point un modèle de connexion des connaissances pour illustrer ces différences [PO01]. Il distingue :

- les stratégies générales (GS: General strategies) ;
- les stratégies à objectif limité (GLS: Goal-limited strategies) ;
- les connaissances spécifiques à un domaine (DSK: Domain-Specific Knowledge) ;
- les connaissances expérimentales (EK: Experiential Knowledge) ;
- les hypothèses (ASS: Assumption).

et synthétise dans le tableau 1 les différences entre les novices et les experts.

Concepteur novice	***Concepteur expert***
Contenu faible des stratégies générales (GLS)	Contenu riche des stratégies générales (GLS)
Petits 'morceaux'	Très gros 'morceaux'
Peu de connaissances spécifiques à un domaine (DSK)	Possession de connaissances spécifiques à un domaine (DSK)
Beaucoup d'hypothèses (ASS)	Hypothèses très faibles (ASS)
Connaissances expérimentales limitées (EK)	Possession de connaissances expérimentales (EK)
Peu de stratégies générales (GS)	Stratégies générales (GS) bien développées

Tableau 1 : Comparaison de stratégies entre un concepteur novice et un concepteur expert [PO01]

On peut déduire de ces études qu'il est inutile de décomposer un problème de conception très finement et que les bons concepteurs font énormément référence à leurs connaissances. Ce qui nous amènera dans le futur à rendre notre outil d'esquisse plus réactif en proposant automatiquement des alternatives par exemple. L'utilisation des moteurs de recherche 3D paut être une première étape.

La validation de l'outil d'esquisse virtuelle s'appuiera sur les travaux précédents en particulier pour analyser les modifications de comportement chez les concepteurs utilisant celui-ci.

Les préconisations des ergonomes cogniticiens pour la réalisation de systèmes d'aide à la conception sont en totale harmonie avec nos objectifs ce qui est rassurant mais ne nous épargnera pas un long travail de tests in-situ.

1.2.2 Le point de vue prescriptif

L'évolution des méthodes prescriptives de conception proposée par Cross [CR03] est la suivante :

- prescription d'un processus idéal de conception (1962-67 : la période de la « conception systématique » proposée par les défenseurs du « mouvement des méthodes de conception ») ;
- description de la nature intrinsèque de l'activité de conception (1966-76 : on découvrit que les problèmes de conception n'étaient pas si facilement systématisables, on comprit la complexité de ces problèmes et le fait qu'ils soient mal structurés) ;
- observation de la réalité de l'activité de conception (après 1970 : enregistrement méthodique de données sur l'activité de conception réelle) ;
- réflexion sur les concepts fondamentaux de la conception (1972-82 : émergence d'une approche plus fondamentale et philosophique des méthodes de conception).

Ces méthodes prescriptives malgré des défauts évidents aujourd'hui, ont permis à l'époque une réflexion sur le processus de conception alors que seuls étaient disponibles des ouvrages proches des catalogues de solutions ou des encyclopédies. L'encyclopédie Diderot D'Alembert (Fig. 1.9) par exemple qui a été très longtemps recopiée et améliorée. Au début du siècle, les ouvrages disponibles en France étaient souvent des traductions d'ouvrages allemands (Eléments des machines, leur calcul et leur construction par C. Bach (Fig. 1.10) traduit de la 7° édition allemande par L. Desmarest et édité en 1901 par la librairie polytechnique Ch. Béranger, Le manuel de l'ingénieur de la société Hütte (Fig. 1.11) traduit à partir de la 24° édition allemande et édité en 1926 par la librairie polytechnique Ch. Béranger).

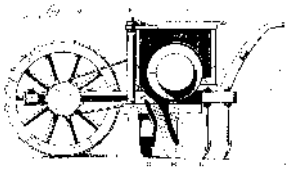


Figure 1.9 : Illustration tome1 Encyclopédie Diderot D'alembert

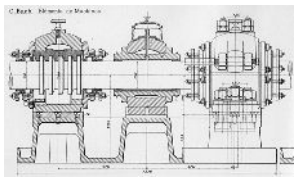


Figure 1.10 : Illustration de l'atlas des Eléments des machines de Bach

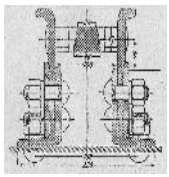


Figure 1.11 : Illustration du tome 3 du Manuel de l'ingénieur de Hütte

Les premières propositions de méthodes sont venues d'Allemagne comme celle de Rodenacker dans son livre édité en 1970 : Methodisches Konstruiren, mais celle

qui a le plus marqué cette période fut celle de Pahl and Beitz éditée en 1977 [PA07] en allemand puis en 1984 [PA08] en anglais grâce à la traduction de Ken Wallace.

Nous pouvons remarquer qu'il est difficile de classer, dans cette chronologie, l'analyse de la valeur mise en place aux Etats Unis dans les années 40 par Miles de la compagnie General Electric. Elle n'est pas une méthode de conception mais une méthode de réduction de coût utilisée dans la phase de définition du produit. Depuis les années 70 en France, l'analyse de la valeur est à peu près la seule méthode enseignée. Un des premiers ouvrages pour l'enseignement de cette méthode en France est le livre : Dessin de construction de l'analyse à la définition [CH10] de Chevalier et Lecrinier édité en 1972 par Hachette. Dans ce livre largement basé sur l'analyse de la valeur, les auteurs ne parlent pas de conception mais d'analyse et de définition de produit et indiquent dans la préface : « La formulation des fonctions est indispensable à la connaissance du matériel et c'est en même temps un exercice d'expression française ». Cette phrase est discutable, de nombreuses personnes utilisent des produits et même en conçoivent sans connaître les fonctions (au sens analyse fonctionnelle) de ceux-ci. Et plus loin : « La troisième partie indique les éléments géométriques et physico-chimiques qui définissent un composant mécanique ». On retrouve ici l'influence des méthodes allemandes avec la réflexion sur les effets physico-chimiques mis en jeu.

Le Japon a ensuite apporté sa série de méthodes souvent liée à la recherche d'amélioration de la qualité comme la méthode QFD (Quality Function Deployment) [AK01] développée par le professeur Noriaki Kano dans les années 80. Taguchi a quant à lui posé les bases de la conception robuste qui cherche à réduire les effets des bruits en entrée sur les résultats de la conception.

Les méthodes plus récentes (Engineering Design Methods, Cross 1989 [CR04] ; The mechanical Design Process, Ullman 1992 [UL02] ; etc.) cherchent à faire un panachage de l'ensemble des techniques proposées et sont donc intrinsèquement plus flexibles et modulaires. L'utilisation des DFX est devenue courante avec volonté marquée de pouvoir être appliquée dès les phases amont de la conception. La prise en compte du facteur temps lié à la réduction du cycle de vie des produits tend à automatiser au maximum les tâches routinières grâce à la capitalisation de connaissance et un bon usage de nouveaux outils de KBE (Knowledge Based-Engineering) directement liés aux outils de CAO. Ces outils imposent une formalisation importante et, malgré tout, flexible de la conception et du produit. Les solutions proposées pour le KBE sont souvent un panachage de techniques venant de l'intelligence artificielle (Système à base de règles, algorithme génétique, raisonnement à base de cas, recuit simulé, etc.) et de systèmes de gestion de contraintes.

Nous allons présenter plusieurs formalisations possibles puis quelques techniques de KBE utilisées pour la conception de systèmes mécaniques.

Formalisation de la conception et/ou du produit

La méthode d'analyse de la qualité du produit est basée sur l'analyse du flux d'effort qui est aussi préconisée en DFA [GR01]. Un autre concept important est la notion de frontières de conception délimitant les zones dans lesquelles se situent les divers éléments. Des propositions [CS01][VA04] sont faites pour formaliser ces zones.

Dans sa méthode, Ullman propose d'utiliser le triangle forme/matière/production avec au centre de celui-ci les fonctions (Fig. 1.12).

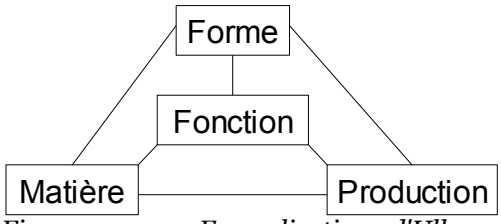


Figure 1.12 : Formalisation d'Ullman [UL02]

Cette formalisation nommée, concurrent design, par Ullman sert de base chez lui pour effectuer le passage du conceptual design au detail design en utilisant une modélisation basée sur les squelettes de pièces. L'avantage de cette formulation est sa simplicité mais contrairement aux suivantes, elle est statique et n'inclut pas de notion de comportement permettant de décrire l'évolution du produit au cours du temps.

D'autres relations sont proposées [HI05] : Fonction/Forme/Comportement [BE02] avec la synthèse pour passer de la fonction à la forme, la modélisation pour passer de la forme au comportement et enfin l'analyse pour passer du comportement à la fonction [PA02] (Fig. 1.13).

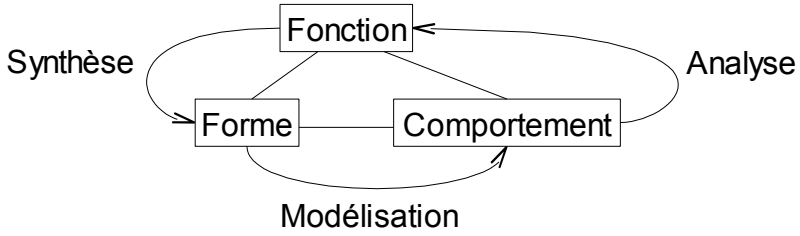


Figure 1.13 : Formalisation de Benami et Jin [BE02]

Cette formulation donne une vision très théorique de la conception qui a pour avantage de clarifier les liens entre les différents champs et les différentes tâches. Par contre contrairement à Ullman aucune référence n'est établie avec la production ou la matière.

Roy et Bharadwaj propose un modèle de fonctions (Fig. 1.14) pour les pièces (PFM: Part Function Model) [RO02] qui relient les entres/sorties du système par sa fonction celle-ci étant associée au comportement qui s'appuie lui-même sur la géométrie et les lois physiques.

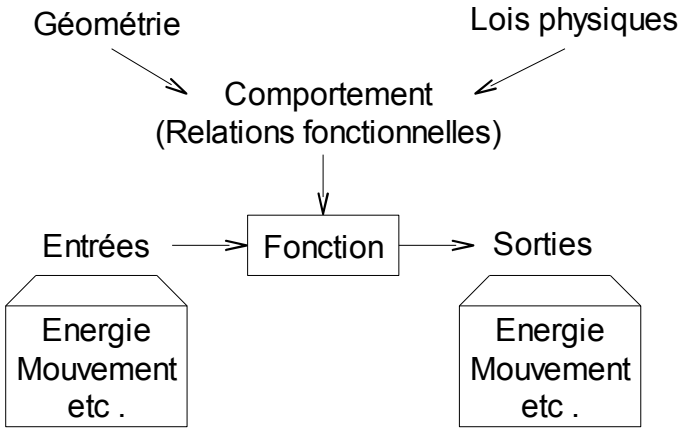


Figure 1.14 : Formalisation de Roy et Bharadwaj [RO02]

Cette formulation est une extension de celle de Benami et Jin, l'ajout d'entrées/ sorties pour définir les fonctions étant souvent implicite.

Zhang et al. ont proposé un modèle fonctionnel B-FES (behavior-driven function-environment-structure) manipulé par un raisonnement comportemental (CBR: Causal Behavioral Reasoning) [ZH01]. Starling a mis au point une grammaire parallèle utilisant un modèle Fonction/Comportement/Structure [ST03]. Ces formalisations permettent de décrire les connaissances acquises durant les anciennes conceptions, la liaison fonction/composant peut par exemple être représentée sous forme matricielle [VA05]. Le lien avec des représentations simplifiées [VI04] permet ensuite de comparer et de choisir la solution finale tout en commençant à gérer les dimensions et les positions des différents éléments.

Gero propose une version située du modèle Fonction/Comportement/Structure [GE01] permettant de prendre en compte le concepteur lui-même et les liens avec les milieux extérieurs. Pour cela il définit trois mondes : un monde externe, un monde interprété et un monde souhaité, ces trois mondes étant inclus les uns dans les autres, Gero aboutit à la figure 1.15 qui permet de nombreuses manipulations (formulation, synthèse, analyse, documentation, reformulation).

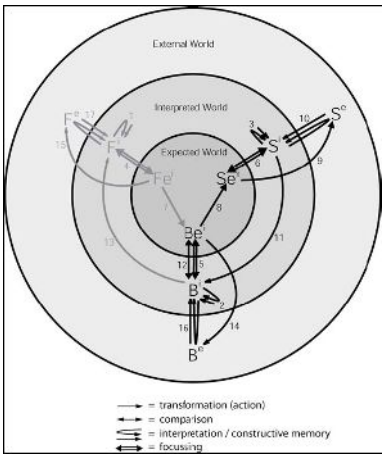


Figure 1.15 : Environnement fonction/comportement/structure situé [GE01]

La formulation de Gero a pour principal intérêt l'ajout de la prise en compte de la situation ou si l'on préfère du point de vue du problème de conception. La prise en compte de la situation lors de la résolution de problème prend son origine dans la cognition située [LA06]. Les trois mondes qui en découlent peuvent facilement trouver des équivalents en réalité virtuelle. Dans ce modèle le monde extérieur peut représenter l'environnement gravitant autour de la conception comme les utilisateurs du produit ou la production par exemple. Le lien peut donc être réalisé avec le modèle d'Ullman.

Il est aussi possible, avec ces formulations, de représenter les mécanismes avec des graphes acycliques directs (DAG) et d'utiliser sur ceux-ci des techniques de l'intelligence artificielle (algorithme génétique par exemple [HA03]) pour rechercher un nouveau mécanisme.

Un formalisme original est utilisé en architecture pour décrire les associations d'idées pendant les phases de conception avancées, il s'agit du formalisme ICF (Issue-Concept-Form). Lai et Chang [LA05] donnent les définitions suivantes :

– Issue : Les questions de conception sont reliées aux tâches de conception qui sont étudiées par le concepteur. C'est un terme pratique pour identifier les points particuliers dans les problèmes de conception.

– Concept : Le concept est la formulation d'une opinion en relation avec une question ou une sous-question de conception. C'est une forme abstraite de réflexion reliée à la tâche de conception.

– Form : La forme est l'élément de conception spécifique qui matérialise le principe de solution. Il est important de noter que c'est un élément de la conception totale du bâtiment qui est directement relié à la question ou la sous-question de conception.

Ils construisent ensuite trois graphes (Issue map, Concept map, Form map) à l'aide de 3 types d'association : similarity, contrast ou contiguity. La sauvegarde de la connaissance générée pendant la conception est ainsi réalisée en dessinant les graphes.

Cette formulation possède la particularité de ne pas faire intervenir la notion de fonction, celle d'Ullman met en avant les notions de matière et de production. L'utilisateur est absent dans tous ces formalismes ce qui conduit à l'absence de gestion de l'interaction avec celui-ci, nous devrons rechercher d'autres formalismes pour aboutir à des processus de conception participative. Nous pouvons signaler des recherches sur ce type de conception [DE05][VI01] qui imposent une confrontation directe entre le client et les croquis des solutions. Seul le modèle de Gero permet d'intégrer l'interaction du concepteur avec sa propre conception.

Le modèle de Horvath et Van der Vegte [HO02] repose sur un modèle User/Product/Environment (Fig. 1.16) qui permet d'obtenir un squelette du produit basé sur les Nucleus.

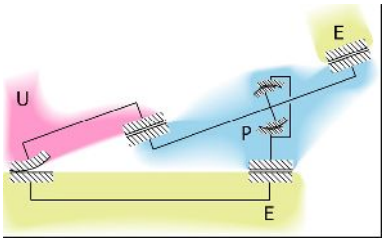


Figure 1.16 : Le modèle User/Product/Environment de Horvath et Van der Vegte

Ce dernier modèle ne prend pas en compte les problèmes de manufacturing même s'il est proche de celui-d'Ullman.

Nous avons retenu des toutes ces formulations, une modélisation du produit basée sur le squelette avec des surfaces de contact très simples (Nucleus ou Wirk elements). La notion de comportement est dans notre outil intrinsèquement intégrée ainsi que l'interaction avec l'utilisateur. L'intérêt de la notion de fonction en conception reste posée et nous n'avons pas pour l'instant eu besoin de l'intégrer.

KBE (Knowledge Based Engineering)

La conception est souvent présentée comme un problème de résolution sous contraintes. Le nombre de contraintes augmentant au cours du processus de conception, on réduit ainsi les plages de solutions possibles [RE04]. Pour formaliser les modèles par contraintes [VE03], il est utile d'utiliser des outils classiques comme l'organigramme technique et le bloc diagramme fonctionnel

avec quelques aménagements. La qualification des modèles étant effectuée à l'aide de quatre critères : la parcimonie, l'exactitude, la précision et la spécialisation avec les définitions suivantes :

- la parcimonie est évaluée au travers de l'inverse du nombre de variables et de relations d'un modèle.
- L'exactitude est une mesure de la distance entre l'espace des solutions du modèle et un comportement de référence.
- la précision est estimée au travers de l'étendue du domaine des variables critères.
- la spécialisation est estimée au travers des principales hypothèses restreignant son domaine d'application et du plus bas niveau systématique utilisé pour sa formulation.

L'intelligence artificielle est souvent utilisée dans les phases avancées de conception, un exemple récent couple l'utilisation d'algorithme génétique dans la phase de recherche de concepts avec un réseau de neurones pour le choix de concepts [HU03]. Les modules de KBE disponibles dans les outils de CAO utilisent avec efficacité ces algorithmes.

Le KBE représente pour nous une solution pour aider le concepteur sans se limiter à une liste de propositions de solutions. La difficulté sur ce point est l'implantation de ces techniques, l'appel de fonctions sur les modules KBE des outils de CAO ne permettrait pas à notre avis d'atteindre la souplesse nécessaire.

DFX (Design for X)

La conception pour la X (qualité, assemblage, usinage, coût, etc.) est une technique modulaire permettant de vérifier la conception en amont. Elle doit pouvoir être utilisée sans que toutes les formes ne soient définies donnant ainsi des indications sur les choix. Les coûts peuvent par exemple être estimés à partir du squelette de la pièce, les choix de machine d'usinage effectués en cohérence avec les spécifications à atteindre [SI02] et les formes modifiées pour faciliter l'assemblage. Nous présentons figure 1.17 et 1.18, des copies d'écran d'un logiciel de DFA et d'un autre de DFM de la société Boothroyd Dewhurst, Inc.

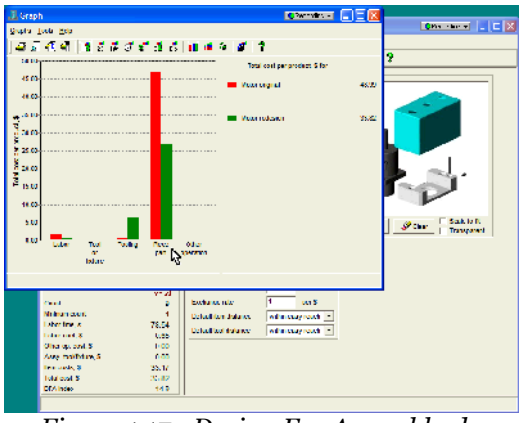


Figure 1.17 : Design For Assembly de Boothroyd Dewhurst, Inc.

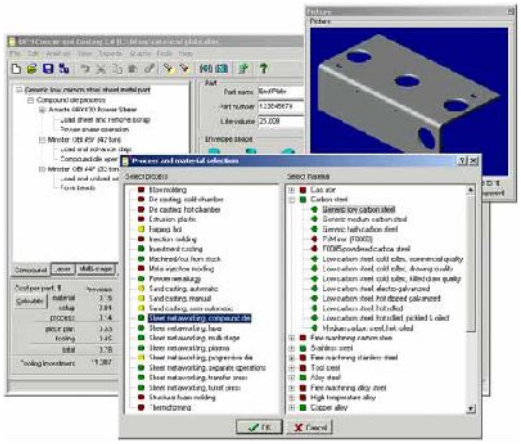


Figure 1.18 : Design For Manufacture de Boothroyd Dewhurst, Inc.

Le manque d'outils informatiques et de recherches supportant les premières

phases de conception est souvent souligné [WA01]. L'usage de maquette physique reste fréquent afin de vérifier l'assemblage d'un mécanisme, par exemple. Les imprimantes 3D permettent d'obtenir très rapidement une représentation physique des pièces ou du mécanisme.

Les recherches en conception permettent de remettre en question certains choix de modèles tant au niveau du processus de conception lui-même qu'au niveau de la structuration du produit. L'analyse des travaux de recherche sur la conception nous a permis de valider une partie de nos modèles de structure de produit (modélisation basée sur les squelettes, surfaces de contact simples) et de processus de conception (modèle de Ullman avec ajout de comportement).

1.3 La réalité virtuelle

Dans le cadre du projet européen SmartSketches [ST01], un état de l'art sur le matériel et les logiciels utiles pour la réalisation de schéma a été réalisé. Des états de l'art plus généraux sur la réalité virtuelle sont disponibles en nombre [LE03]. *Le traité de la réalité virtuelle* sous la direction de Philippe Fuchs [FU02] est la référence la plus importante en France avec *La réalité virtuelle* de Burdéa et Coiffet [BU02]. Nous présenterons ici les éléments matériels et logiciels qui peuvent intervenir dans un outil d'esquisse virtuelle, nous ne nous sommes pas restreints aux seuls domaines qui seront utilisés dans nos outils d'esquisse. La déformation a ainsi été traitée, il nous a semblé important d'effectuer cet état de l'art large pour faire des choix qui puissent être facilement extensibles.

1.3.1 Matériel

L'interface homme-machine en réalité virtuelle utilise de nombreux sens, il est important de classer les différents types de matériel en fonction de ces sens. Nous utiliserons pour cela la figure ci-dessous (Fig. 1.19) représentant un utilisateur manipulant un objet avec sa main gauche et le déformant avec sa main droite.

Figure 1.19 : Utilisateur manipulant un objet

Les sens qui peuvent être pris en compte pour notre application sont donc : la vue, le toucher, l'ouïe et la voix. Nous n'avons pas pris en compte dans notre état de l'art des périphériques utilisés dans le domaine du jeu vidéo même si nous n'excluons pas cette possibilité. Philips vient de proposer par exemple de nouveaux périphériques pour les jeux simulant le vent, des vibrations transmises aux poignées, des lumières d'ambiance en fonction de celles apparaissant à l'écran, etc. Cette technologie amBX est présentée dans la figure 1.20.

Figure 1.20 : Technologie amBX de Philips pours les jeux

On peut aussi représenter l'être humain comme une boîte noire avec des entrées et des sorties, ce qui permet de découpler par exemple la sensation du toucher de l'effort que l'on peut exercer sur un objet avec les doigts.

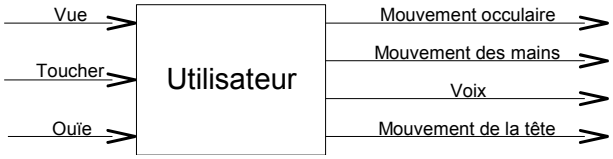


Figure 1.21 : Vue fonctionnelle de l'utilisateur

Les deux sens les plus développés en réalité virtuelle sont la vue et le toucher, nous ne présenterons ici que des exemples proches de l'esquisse virtuelle, une littérature abondante existe aujourd'hui pour une description plus globale des dispositifs utilisés en réalité virtuelle.

Un critère important pour la qualité de perception d'un schéma dans l'espace est la mise en correspondance de la visualisation de l'esquisse avec l'extrémité de l'outil effectuant le tracé, il s'agit de la corrélation visualisation/mouvement de la main droite. Cette corrélation peut être vue comme une boucle de contrôle entre la vue et les mouvements de la main (Fig. 1.22).

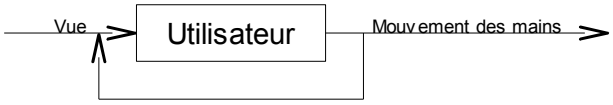


Figure 1.22 : Boucle de contrôle sur les déplacements

Pour évaluer ce critère, il faut définir un repère global et mesurer la position du point extrémité de l'outil et le point correspondant dans le système de vision. Si la visualisation s'effectue dans un espace à trois dimensions, un simple calcul de distance entre ces deux points permet de qualifier le critère. Si la visualisation s'effectue dans un espace à deux dimensions, on est obligé de projeter le point extrémité de l'outil dans cet espace pour déterminer la distance entre les deux

points dans le plan. Il est facile de visualiser cette projection sur la figure suivante (Fig. 1.23). L'extrémité du doigt de la main droite représentant l'extrémité de l'outil, un repère est attaché à celui-ci. L'image représentant la vision de l'utilisateur, il est facile de placer un repère sur cette image. Classiquement ce repère est x de gauche à droite et y de haut en bas avec le point 0 en haut à gauche.

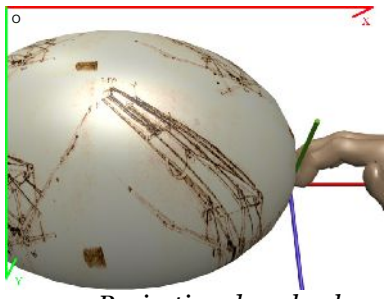


Figure 1.23 : Projection dans le plan visuel

Tous les systèmes ne permettent pas de maintenir cette correspondance. Si l'on utilise une souris par exemple, il est clair qu'il existe entre le plan de déplacement de la souris et le plan visuel, une transformation correspondant à une rotation d'axe x et une translation. L'utilisateur intègre cette transformation après un temps d'apprentissage mais il n'en demeure pas moins que celle-ci n'est pas naturelle. Pour que la corrélation visualisation/mouvement de la main droite soit acceptable, il faut donc que la transformation précédente soit la plus simple possible.

Un deuxième critère dont nous essayons de montrer l'importance dans le cadre de ce travail est la correspondance entre l'effort réel en cas de contact avec le mécanisme virtuel et l'effort virtuel généré par un système à retour d'effort dans la main droite. Ce deuxième critère n'est donc valable qu'en cas d'utilisation de l'haptique. Du point de vue de l'utilisateur, c'est la corrélation entre la sensation d'effort et le mouvement de sa main droite qui est importante. Cette corrélation peut être vue comme une boucle de contrôle entre le toucher et les mouvements de la main droite (Fig. 1.24).

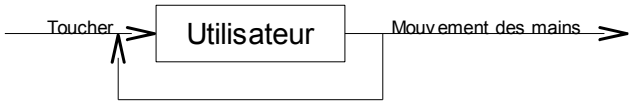


Figure 1.24 : Boucle de contrôle sur les efforts

D'autres critères de mise en correspondance peuvent être établis pour les autres sens (ou entrée/sortie).

Système de visualisation

De nombreux classements existent pour les systèmes de vision en réalité virtuelle, nous ne présenterons que la classification de Blundell et Schwartz [BL02]. Cette classification décompose les systèmes de visualisation en 3 grandes familles : les systèmes monoculaires, stéréoscopiques ou auto-stéréoscopiques (Tableau).

	Monoculaire	Stéréoscopique	Autostéréoscopique
Techniques générales	Ecran plat traditionnel	Anaglyphe Frame séquentiel	Systèmes holographiques Réalité virtuelle immersive Réalité augmentée Ecrans AutoQ Feuille lenticulaire
Commentaires sur la mise au point en profondeur	Manque d'accommodation, de convergence et de parallaxe binoculaire	Ajout de mise au point en profondeur par parallaxe binoculaire. Découplage des mises au point de l'accommodation et de la convergence	La parallaxe binoculaire est présente. Les mises au point de l'accommodation et de la convergence peuvent être obtenues ou peuvent être découplées. L'occlusion de la vue peut être ou ne pas être présente.
	Visualisation directe Une seule direction de visualisation	Utilisation de lunettes de visualisation Une seule direction de visualisation	Visualisation directe mais il peut être nécessaire d'utiliser un équipement de visualisation. Liberté pour la direction de visualisation, cependant ce degrès de liberté varie.

Tableau 2 : Classification par Blundell et Schwartz [BL02]

Il faut remarquer dans cette classification qu'il est important de faire le choix du système de visualisation en fonction de l'utilisation supposée du système global. Dans notre cas, la manipulation de l'objet visualisé pose des problèmes d'occlusion qui sont plus ou moins difficiles à gérer en fonction de la solution retenue. L'occlusion peut être partielle si l'on utilise des lunettes stéréoscopiques et que la main droite occulte la vision de l'oeil droit ce qui conduit à la perte de la stéréoscopie.

La visualisation d'un objet numérique s'effectue en projetant sur un support, une source lumineuse. Le support peut être en 2D ou en 3D (hologramme). Les supports en 2D peuvent être de formes diverses (plane, sphérique [FE03] Fig. 1.25, conique [BI01] Fig. 1.26, etc.) et/ou constitués d'un ensemble de ces formes (CAVE, MOOVE) [VA03].

Figure 1.25 : Projection sphérique de la société Elumens

Figure 1.26 : Projection conique sur miroir [BI01]

Le choix est souvent conditionné par l'angle de vue recherché et donc par la taille et l'échelle de l'objet à visualiser. Les murs de projection permettent par exemple de visualiser un véhicule à l'échelle 1. Le sens de la projection a son importance et conditionne le type de matériau utilisé (toile spéciale, verre, etc.). Si des manipulations doivent être effectuées, on préférera la rétroprojection afin de ne

pas occulter la projection surtout dans les systèmes de type CAVE. Les systèmes autostéréscopiques sont utilisables en rétroprojection [BA03] sur une vitre simulant ainsi un workbench.

La stéréoscopie est obtenue naturellement grâce aux systèmes encore rares qui utilisent des supports en 3D, si le support est en 2D, il faut projeter l'image pour l'oeil droit et l'image pour l'oeil gauche sur cet unique support. Plusieurs techniques sont utilisées, les plus courantes étant l'affichage oeil droit/oeil gauche alternativement à une fréquence double de la fréquence d'affichage classique des moniteurs. La différence d'angle de vue entre l'oeil droit et l'oeil gauche est utilisée avec les réseaux lenticulaires. On affiche dans ce cas en même temps les deux images mais alternativement sur chaque colonne de pixels.

La réalité augmentée peut être utilisée avantageusement pour contourner les problèmes d'occultation. Nous verrons au paragraphe 1.4 que la plupart des systèmes d'esquisse virtuelle utilisent une vitre semi-réfléchissante (Fig. 1.62) ce qui évite de couper le champ de vision avec la main. On peut, grâce à ce système, simuler l'implantation de l'esquisse directement dans un système existant. L'utilisation de technique venant de la réalité augmentée [BI01] sont à essayer avec le sketcheur, il serait ainsi possible de concevoir un système mécanique in-situ sans reconstruction de l'environnement.

On peut projeter le schéma réalisé directement sur la table de travail [SA01]. La représentation en 3D des objets en temps réel n'est pas simple, plusieurs techniques sont proposées : hologramme, PCI (Planar Contour Imaging) [ST02].

Les deux technologies les plus en adéquation avec un outil d'esquisse virtuelle sont l'autostéréoscopie et l'holographie. La première va se généraliser à l'horizon 2010 pour le grand public (jeu, film en relief) avec des écrans plats qui pourront passer de la vision classique en 2D à la 3D grâce à des réseaux lenticulaires. Philips propose des écrans de ce type (Fig. 1.27) avec un logiciel qui permet à toute application OpenGL d'être utilisée directement sans réécriture de code (Fig. 1.28).

Figure 1.27 : Ecran autostéréocopique avec son circuit intégré

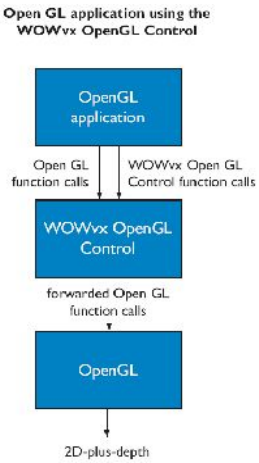


Figure 1.28 : Contrôleur WOWvx OpenGL

L'holographie est, elle aussi, déjà sur le marché avec la proposition par exemple de la société Holographica qui comme pour l'autostéréoscopie propose des solutions logicielles permettant d'utiliser sans modification les applications OpenGL (Fig. 1.29).

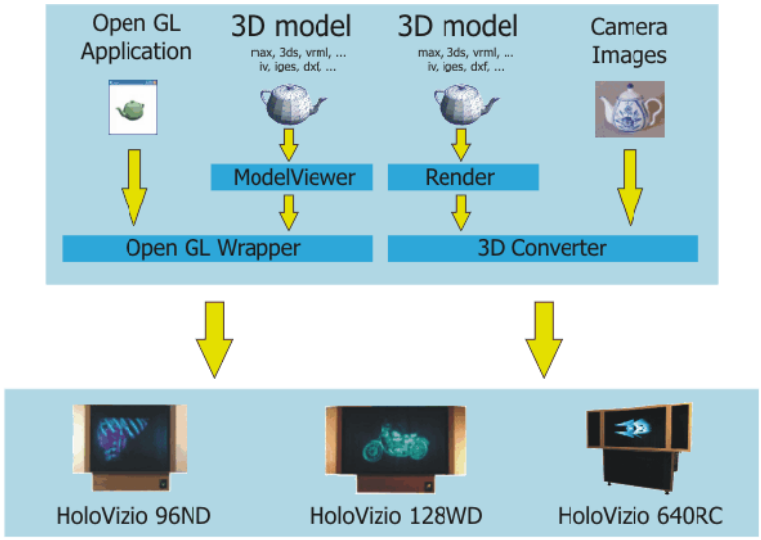


Figure 1.29 : Lien entre les applications OpenGL et les écrans holographiques

Système haptique

Les systèmes à retour d'effort utiles pour le tracé dans l'esquisse virtuelle doivent posséder les caractéristiques suivantes :

- grand débattement (comme pour les tablettes graphiques des problèmes de précision surviennent avec les surfaces de traçé trop petites)
- faible inertie (la dynamique du traçé ne peut être obtenue que sil'on l'impression de déplacer un élément d'une masse aussi faible que celle d'un stylo)
- vrai retour d'effort et non simple vibration (les systèmes à base de vibration ne peuvent qu'indiquer la présence de contact mais pas l'orientation de celui-ci)
- occultation minimale du champ de vision (si le choix est fait de maintenir une correspondance entre le point visuel et le point haptique, il est évident que l'occultation peut gêner lors du tracé)
- retour d'effort indispensable et retour de couple optionnel (pendant la phas de tracé, nous simulons l'utilisation d'un stylo dans l'espace, le retour de couple n'est pas utile)

Pour la manipulation des systèmes mécaniques créés, le retour en couple peut devenir indispensable. De petits périphériques [SN01][VE02] avec un seul degré de liberté peuvent être utilisés, l'entrée des mécanismes pour l'utilisateur ne possède souvent qu'un seul mouvement, translation ou rotation.

La transmission d'effort dans ces systèmes s'effectue soit par des structures sérielles (Fig. 1.30) soit par des structures parallèles (voir tableau 3). Ces dernières peuvent aussi se décomposer en structures à base de jambes mécaniques (Fig. 1.31) ou en structure à base de câbles (Fig. 1.32).

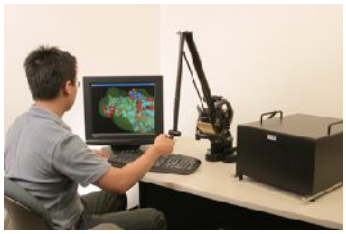
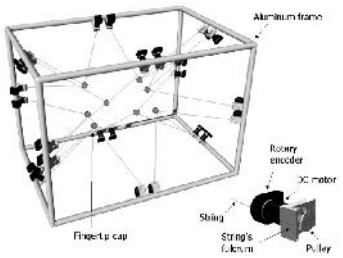
Système sériel	Système parallèle	
	Avec jambes mécaniques	Avec câbles
 *Figure 1.30 : Phantom 3.0*	*Figure 1.31 : HapticMaster*	 *Figure 1.32 : Spidar*

Tableau 3 : Classification des systèmes à retour d'effort utilisables pour l'esquisse virtuelle

Les systèmes à structure parallèle existants ont des débattements réduits et leur commande est complexe, de nouvelles propositions sont faites dans notre laboratoire pour obtenir des débattements plus importants et surtout des structures isotropes. Le déplacement en translation d'un moteur correspondant exactement au déplacement de la partie terminale d'où une commande simplifiée. Une structure parallèle de ce type est présentée en exemple au chapitre 4.2.

Les systèmes parallèles [CH01] permettent de répondre à des tâches plus précises. Pour les grands déplacements avec peu d'inertie, les systèmes à base de fils [TA01][RA01] sont une très bonne solution pour un sketcheur. On peut coupler les systèmes haptiques avec d'autres périphériques [WA02] pour des tâches spécifiques.

L'utilisation des systèmes à retour d'effort en 3D peut être essayée pour la réalisation et la manipulation de schéma en 2D [RA02].

Les interfaces haptiques sont souvent utilisées dans le développement rapide de produit [CH06].

Autres dispositifs d'entrée

Le tracking de main et de tête [DO06] est courant avec les workbench ou les Cave permettant la saisie de grands déplacements. De nombreuses techniques ont été explorées pour effectuer ce tracking, tracking mécanique, tracking par ultrasons, tracking par modification d'un champ électromagnétique, tracking par caméra (infrarouge ou non), tracking à base de gyroscopes, etc. Ces systèmes permettent de capter les gestes effectués par l'opérateur et ainsi de simuler un travail à deux mains par exemple. Les travaux qui vont dans ce sens ne sont pas concluants pour le travail en esquisse, ils permettent juste de générer des surfaces très approximatives puisque aucune sensation de toucher n'est présente. Même au niveau artistique qui accepte la réalisation de formes plus floues, la plupart des essais réalisés dans des systèmes immersifs avec tracking de main, ce sont voués à des echecs.

1.3.2 Logiciel

Le logiciel doit gérer les différents périphériques et permettre la création, la modification et la manipulation du système mécanique. Nous allons ici aussi faire une présentation générale de l'état de l'art sans se retreindre à l'esquisse virtuelle. Nous commencerons par la définition de la structure générale d'un programme de réalité virtuelle (Fig. 1.33). Nous limiterons notre état de l'art en ne prenant en compte pour l'utilisateur que l'ouïe et le toucher en sortie du programme et le mouvements des mains en entrée.

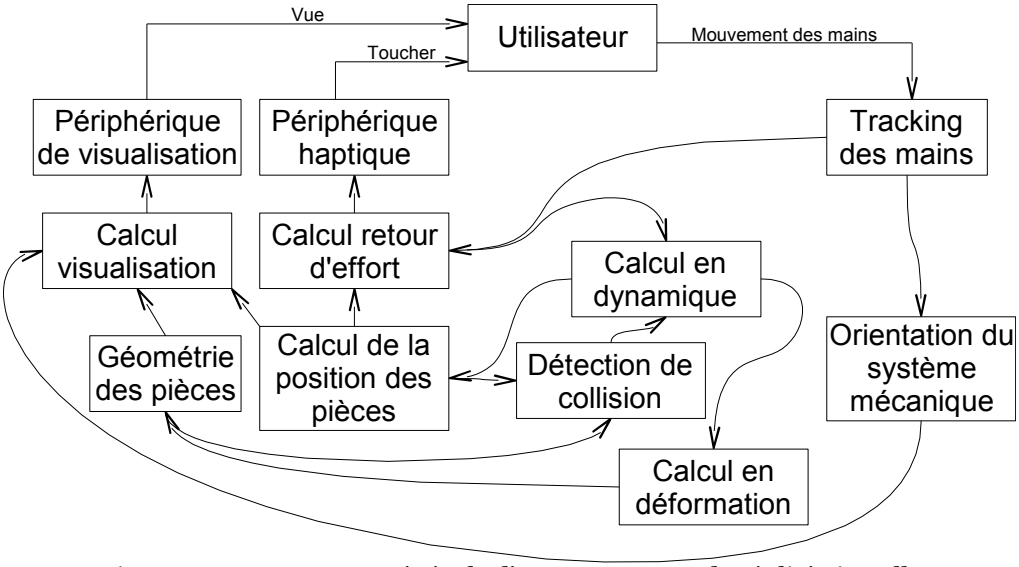


Figure 1.33 : Structure générale d'un programme de réalité virtuelle

Cette structure n'est représentée que pour structurer cet état de l'art sur la partie logiciel, elle est constituée d'un ensemble de blocs de calcul et de modélisation. En réalité tous ces blocs sont reliés entre eux à différents niveaux, nous montrerons que le choix d'un modèle géométrique influe par exemple sur la visualisation. Une autre décomposition (Fig. 1.34) doit être effectuée afin de décrire à quoi correspond un système mécanique. Afin de rester le plus général possible, nous considèrerons qu'un système mécanique est un ensemble de pièces reliées entre elles par des contraintes (liaisons, positionnements, etc.) et ayant la possibilité de rentrer en collision.

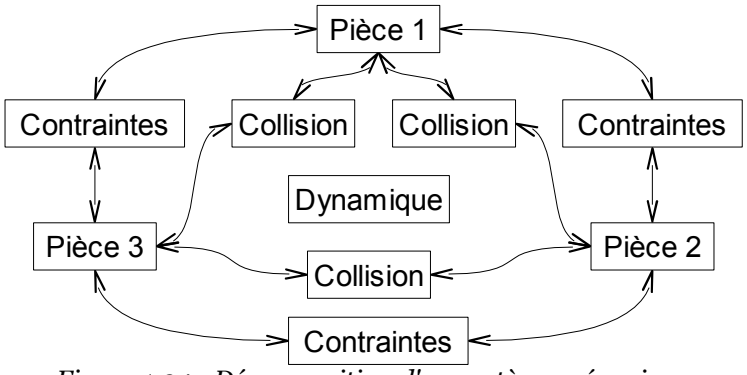


Figure 1.34 : Décomposition d'un système mécanique

Les pièces étant elles-mêmes constituées d'un ensemble d'entités géométriques reliées entre elles par des contraintes géométriques par exemple. On peut aussi utiliser le concept de squelette de pièce pour les modéliser, ce qui nous conduit à utiliser le concept d'entités géométriques intra-contact et celui d'entités géométriques inter-contact.

Nous allons maintenant présenter les différents composants logiciels.

Haptique

Historiquement, la téléopération est une des premières applications qui a eu besoin d'un retour d'effort [TU01]. La difficulté majeure du développement d'application pour l'haptique est la vitesse à laquelle les calculs doivent être effectués. La fréquence la plus couramment utilisée est 1 KHz mais il faudrait être capable d'atteindre 4 KHz pour simuler un contact sur des matériaux durs. Les constructeurs de matériel haptique fournissent leurs systèmes avec des librairies de développement qui ne sont pas toujours très efficaces et qui sont trop liées à un matériel donné. Les recherches en haptique essaient donc de proposer des méthodes de calcul rapides et le plus souvent spécifiques à une application donnée.

Le lien du calcul de retour d'effort avec la détection de collision est évidente et peut être vu comme un complément de celle-ci. Pour le calcul de l'effort, il faut connaître les zones de contact, la pénétration, les matériaux en contact, etc. Le retour d'effort nécessite donc des connaissances mécaniques sur les contacts.

Du point de vue plus informatique, l'haptique nécessite de calculer l'effort en retour, de gérer cette tâche de calcul à la fréquence voulue, de piloter le système à retour d'effort.

Le calcul de l'effort en retour s'effectue en utilisant des lois de mécanique, fonction des comportements des matériaux en contact et de la géométrie des contacts. En fonction du niveau de détail, les calculs peuvent devenir très compliqués comme pour la simulation d'objets fortement déformables et rugueux. La notion de niveau de détail (LOD) est aussi présente en haptique [KO01] afin de sentir des imperfections sur une pièce par exemple. Le LOD peut être géré par des ondelettes [PA04].

La gestion de la fréquence de calcul impose de travailler en multi-tâche (multi-thread) et donc d'utiliser des librairies C++ comme Pthred, Jthread, etc.

Le pilotage du système à retour d'effort doit être le plus transparent possible d'où l'utilisation de librairies [TS01] permettant de piloter plusieurs périphériques comme Chai3D, Haptik library, H3D, etc.

Une autre direction de recherche très en vogue en haptique est la sculpture d'objets (modification des formes, peinture en 3D). Le retour d'effort pour la sculpture d'objet est un thème de recherche important [BL01][JA01] qui a déjà conduit à des produits commerciaux (FreeFrom). La validation du retour d'effort pour la sculpture d'objets n'est pas toujours obtenue en fonction du type d'objets à réaliser [EV01]. L'ajout ou l'enlèvement de matière avec des systèmes à retour d'effort impose des modélisations appropriées de type continu ou discontinu. Les modélisations discontinues sont couplées avec un lissage de l'effort afin de supprimer les effets dynamiques de mise en résonance du système à retour d'effort. Le modèle le plus classique qui nous vient des technologies utilisées dans les cartes graphiques est la représentation de surfaces à base de triangles. Comme

pour la détection de collision, ce modèle présente de nombreux inconvénients ce qui amène à rechercher d'autres modèles suivant deux directions possibles : soit des modèles continus qui peuvent contenir plus d'informations, soit des modèles discontinus encore plus simples ou permettant de gérer des informations sur le volume et non simplement sur la surface de l'objet à modéliser.

Un des modèles discontinus les plus étudiés actuellement est le modèle basé sur les points (Point-based Models). La visualisation de ce type de modèle s'effectue à l'aide de Surfels. L'utilisation de ce type de modèle en chirurgie dentaire [YA03] montre qu'il est plus facile de modifier en temps réel, un nuage de point plutôt qu'une surface triangularisée.

Les modèles continus peuvent être dérivées de modèles connus sur les efforts de coupe [BO03] permettant ainsi de déterminer les efforts lors de la sculpture par enlèvement de matière.

Des propositions sont faites pour peindre des modèles 3D avec des systèmes haptiques [GR02].

Détection de collision

On peut citer quelques présentations générales du problème de la détection de collision [ME01]. Les premiers travaux sur la détection de collision ont eu lieu en robotique [RU01], les objets à manipuler étaient souvent simples et les robots étaient représentés par des chaînes ouvertes en série.

La détection de collision se décompose en deux grandes familles la détection de collision discontinue ou continue (par rapport au temps). La détection discontinue n'effectue qu'un test à chaque pas de temps pour savoir si les objets se touchent ou non. Pour les objets modélisés par des géométries fines comme des lignes, ce type de détection est inappropriée, il faut alors passer à de la détection continue [RE01] qui cherche à quel moment exact s'effectue le contact. Nous donnons plus de détail sur la mise en place de ce type de détection de collision que nous avons utilisé dans le deuxième prototype, au chapitre 3.

La géométrie des pièces est aussi très critique pour la qualité des résultats de la détection, cette modélisation est souvent différente de celle utilisée pour la visualisation ou pour l'haptique. Il faut aussi choisir la modélisation qui réduira au maximum les temps de calculs tout en représentant un maximum des géométries utilisées dans l'application cible.

Un autre critère de choix est la présence ou non de déformation dans les pièces. L'utilisation de corps rigides permet de simplifier les calculs de distance. En cas de déformation, des hypothèses sur les déplacements (petits déplacements par exemple) doivent être prises en compte. La détection de collision entre objets déformables est traitée dans plusieurs travaux [GA02][BR04][LA04] mais peu de librairies actuelles gèrent correctement ce problème.

Au point de vue informatique, la détection de collision s'effectue au minimum en deux étapes : recherche rapide des possibilités de contact à partir d'une hiérarchie de géométries englobantes puis recherche des couples de géométries en contact.

La première étape fait appel à de nombreux types de hiérarchies, Octree, Sphere-Tree [BR02][OS01][ZA01]. La détection de collision utilisée dans le premier prototype est du type Sphere-Tree, nous donnons des détails supplémentaires sur cette méthode en Annexe 2. Une difficulté est l'évolution de la hiérarchie en cas de déformation, elle doit être modifiée en continu [KA03]. James et Pai proposent une version des Sphere-Tree mieux adaptée à la déformation : le BD-

Tree [JA03].

La seconde étape dépend beaucoup du modèle de géométrie choisi pour le calcul de distance. La majorité des librairies de détection de collision utilisent des surfaces triangularisées, nous commençons à trouver des librairies spécialisées pour le retour d'effort comme H-Collide [GR03]. Dans les systèmes mécaniques un nombre très important de surfaces sont des quadriques pourtant peu de librairies de détection de collision sont spécialisées pour ce type de surface. On trouve ce type de librairies dans les travaux sur l'assemblage [SC05]. Les surfaces paramétriques, les Nurbs peuvent être utilisées en courbe ou en surface [TH01][DA02][ST04][LI01][NE01] mais la recherche de distance est plus complexe.

Il peut être préférable d'utiliser des librairies de recherche de proximité qui peuvent parfois gérer des tolérances [LA01] lors de l'assemblage de pièces. Le calcul de distance entre entités simples [LE01][BA07][CH09] : points, droites, cercles, quadriques permet d'effectuer des calculs plus précis et plus rapides. Ces calculs ont été implantés dans la librairie SiLVIA [LE02] qui utilise elle-même la librairie Exacus, c'est à priori une des rares librairies qui effectue des calculs sans discrétisation des formes. La rapidité des calculs devient très relative dans le cas de deux ellipsoïdes soumis à des mouvements rationnels [CH04]. L'utilisation de l'axe central est aussi très courante afin de diminuer les calculs sur la collision entre polyèdres [KI03]. Des techniques très spécialisées peuvent être utilisées pour les splines, comme les cônes tangents [JO04]. Il est prouvé que la distance entre deux cercles dans l'espace ne peut pas être écrite avec des radicaux [NE02].

Certaines recherches tentent d'unifier la recherche de distance entre objets de différents types [JO03].

Cinématique

Un lien entre le schéma et la théorie des mécanismes doit être établi ce qui permettrait de déduire rapidement les mouvements possibles. Dans ce cas, le sketch doit être contraint et les liaisons supposées exactes, il faut ensuite trouver les degrés de liberté du mécanisme [ZH02]. Nous montrerons que l'on peut obtenir les degrés de liberté des mécanismes dans notre deuxième prototype à chaque pas de temps. Ce résultat est automatiquement obtenu lors de l'inversion de la pseudo-inverse de la matrice des contraintes.

La cinématique permet de générer les séquences d'assemblage et de désassemblage de mécanismes. La prise en compte des tolérances dans ce cas devient primordiale. Une technique pour gérer les tolérances est l'utilisation des espaces de configuration [JO05] mais les calculs mis en jeu restent encore trop lourds pour le temps réel en haptique.

Dynamique

Lors du Siggraph 94, Baraff [BA05] présentait une méthode de calcul rapide des forces de contact entre des corps rigides sans pénétration. Cette méthode basée sur l'algorithme de Dantzig pour la résolution du LCP (Linear Complementarity Problem) a été souvent réutilisée dans les librairies de dynamique comme OpenDE [RU02] par exemple.

De nombreuses autres formulations ont été proposées et comparées [RE06]. Il est

malheureusement très difficile de tirer des conclusions claires sur les avantages et les inconvénients de ces formulations. Les exemples traités sont parfois très éloignés des systèmes mécaniques et plus proche de l'animation en temps réel. La destruction de murs de briques est un grand classique repris par Intel pour une démonstration au GDC 2006 de sa nouvelle librairie permettant la gestion des systèmes multi-coeurs. Dans cette démonstration, de nombreux contacts sont présents lors de la démolition et c'est pourtant OpenDE qui est utilisé pour les calculs de dynamique.

Figure 1.35: Démonstration de démolition de château par Intel

Hasegawa et Sato [HA02] propose une méthode de pénalité pour de la dynamique avec retour d'effort en 6-dof qu'ils comparent avec d'autres méthodes. Redon et al. ont fait plusieurs propositions pour améliorer le couplage dynamique/détection de collision : l'utilisation du principe de Gauss [RE02][RE05][BR05][CH08], la détection de collision continue avec une arithmétique des intervalles [RE03], etc. Une des critiques formulées [EG01] sur la thèse de Redon est l'emploi de la méthode de recherche du point le plus proche (Nearest-Point Problem) de Wilhelmsen.

Kalaba et Udwadia ont proposé une formulation de la dynamique sous contraintes à partir du principe de Gauss [KA05] donnant directement l'accélération. Cette formulation est plus générale que celle de Lagrange [UD01]. Elle implique l'utilisation de matrice pseudo-inverse. Différentes méthodes sont disponibles pour l'obtention de la pseudo-inverse [DE07][CO08]. La dynamique sous contraintes permet d'étudier l'effet des jeux dans les mécanismes [VI03].

A partir de la dynamique sous contraintes, on peut établir une dynamique dans l'espace des tâches [DE06] et montrer la symétrie entre tâche et contrainte. Ce type de dynamique peut être efficacement utilisée en contrôle de systèmes parallèles.

Une approche hamiltonienne récursive est présentée par Naudet [NA01]. Celui-ci justifie l'utilisation de l'approche hamiltonienne par le fait que l'élimination de multiplicateurs de Lagrange ne nécessite que deux dérivations contre trois dans la formulation classique utilisée dans les algorithmes basés sur les accélérations. Nous pouvons remarquer que dans la formulation d'Udwadia et Kalaba, l'élimination des multiplicateurs de Lagrange n'existe pas. La stabilité et la qualité du résultat (Fig. 1.37) est dans ce cas en nette amélioration. La récursivité de la formulation de Naudet lui permet principalement de traiter des systèmes multi-corps de type chaîne (Fig. 1.36) qui possèdent des topologies très répétitives.

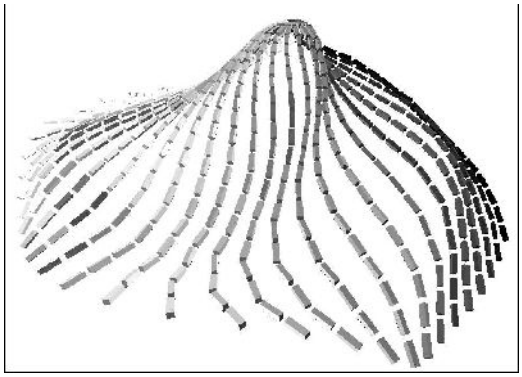


Figure 1.36 : Système multi-corps de type chaîne [NA01]

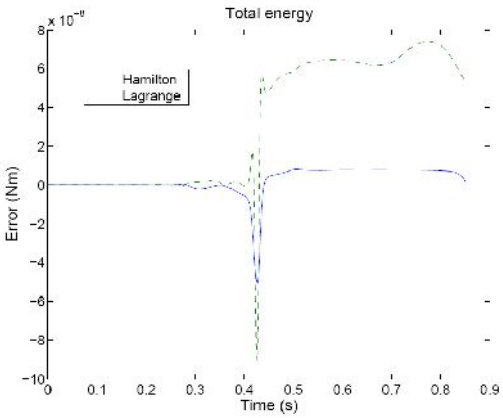


Figure 1.37 : Comparaison entre la formulation hamiltoniène et la formulation de Lagrange en terme d'erreur [NA01]

Une large littérature existe sur l'utilisation de la dynamique en animation. Erleben [ER01] a proposé à la suite de sa thèse la librairie : OpenTissue (www.opentissue.org).

Le frottement est bien géré dans la formulation de Kalaba et Udwadia [KA08] , de nombreux artifices sont proposés pour obtenir des simulations cohérentes et stables [KA07].

Notre état de l'art sur les méthodes utilisées en dynamique pour le temps-réel, nous a amené à choisir la formulation de Kalaba et Udwadia et à l'adapter à notre cas particulier de contact entre éléments géométriques simples.

Déformation

Des bilans existent sur la déformation en animation [HA01][BA04][NE03] mais ils sont plus rares quand les calculs doivent être plus rapides (dynamique, haptique). La plupart des travaux en déformation touchent à la déformation d'organes pour la médecine [CO05][CO06][PI01][CH07], il est intéressant de suivre l'évolution des méthodes utilisées dans ce domaine. Dans le cadre du projet DIJA [DA03], l'utilisation de déformation de formes libres (FFD) a été implémenté en Java3D dans le but de rendre plus intuitif le dialogue Homme – Machine en CAO. Plusieurs librairies [HI02] ont été développées qui utilisent le retour d'effort. Les modèles par éléments finis sont trop lourds pour le temps-réel et nécessitent des pré-calculs [BA01], inadaptés aux systèmes interactifs. Druesne et al. utilisent la transformation de Karhunen-Loève [DR01] pour réduire le modèle et arrivent ainsi à des grandes déformations en temps-réel. Des modèles hiérarchiques en temps et en espace [DE01] permettent des performances utilisables avec l'haptique. Le modèle masse-tenseur, proposé par Schwartz, est un bon compromis entre le modèle masse-ressort et la FEM [SC03]. Les éléments de frontières (BEM) sont souvent utilisés en chirurgie [WA04] dès qu'il y a de la découpe d'organe, il n'est pas nécessaire de propager les modifications de la surface du maillage à l'intérieur des organes. La modélisation d'organes multimatériaux n'est par contre que difficilement possible. La déformation de tôle métallique peut être obtenue en utilisant des fonctions implicites de type F-Rep [SO02]. Des déformations plus locales peuvent être simulées par des systèmes masses-ressorts pour le plissement de la peau par exemple [LA03]. Un

système assez similaire basé sur l'utilisation de volumes liés par des potentiels est mis au point dans ParSys [PI02]. Dans ces modèles les solides sont souvent discrétisés par des sphères [JA04], Janson tente d'intégrer ce travail dans FeniCS [JA05] (www.fenics.org). Les éléments frontières sont aussi utilisés [JA02]. Druon et al. [DR02] utilisent les automates cellulaires pour réaliser la déformation visco-plastique en temps-réel d'un cube écrasé par une plaque (Fig. 1.38).

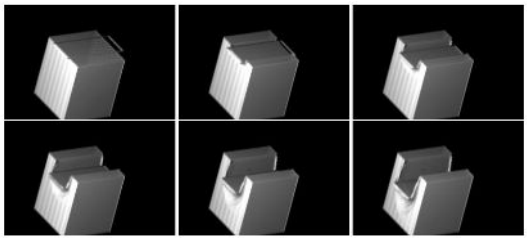


Figure 1.38: Cube écrasé par une plaque [DR02]

Pour déformer des pièces dans un mécanisme, l'utilisation de modèle corotationnel [DU02] est à envisager. On suppose dans ce cas que les déformations sont faibles et les calculs s'effectuent en pseudo statique linéaire dans un repère ayant subi un mouvement rigide en dynamique (Fig. 1.39).

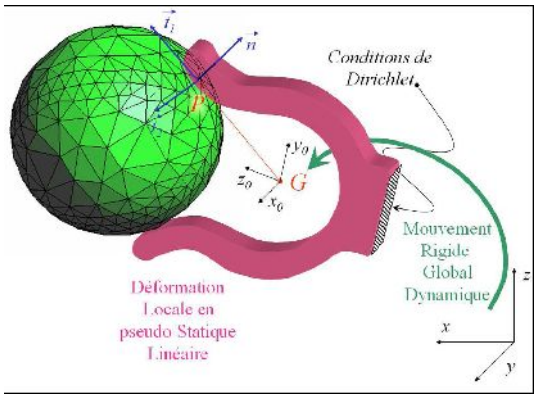


Figure 1.39 : Approche corotationnelle globale développée par Duriez [DU02]

La déformation de cordes est présentée dans les travaux pour le médical de Brown pour la couture des plaies [BR03]. La déformation de câbles peut être simulée en découpant celui-ci en succession de corps rigides [LO01].

Un effort d'intégration a été réalisé dans GiPSi qui est une plateforme de développement orientée simulation chirurgicale [GO01]. Un projet international de proposition de plateforme ouverte poursuit cet effort d'intégration. SOFA pour Simulation Open Framework Architecture [AL03] propose une décomposition en modèle de visualisation, modèle de collision, modèle haptique et modèle de comportement, partie gauche de la figure 1.40. Dans la partie droite de la même figure, on peut voir un exemple d'organe représenté par un modèle par élément fini pour le comportement, un modèle lissé pour la visualisation et un modèle avec hiérarchie de sphères pour la détection de collision.

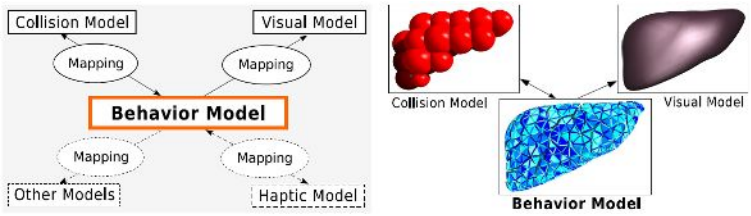


Figure 1.40 : SOFA : Simulation Open Framework Architecture [AL03]

La décomposition proposée dans SOFA est proche que nous avons utilisée dans la deuxième version de notre outil d'esquisse virtuelle.

Le couplage dynamique/déformation n'est pas actuellement envisageable en temps réel, les modèles proposés le sont plus pour l'animation ou la simulation [JA06].

L'utilisation de modèles simplifiés doit être recherchée pour accélérer les calculs, Yang et al. [YA02] proposent pour la simulation de brosses à dents multi-matériaux, un modèle simplifié de poutres en torsion permettant de tester rapidement différentes configurations.

On peut remarquer qu'une grande variété de modèles existe pour la déformation mais qu'il n'existe pas de modèle simple à utiliser pour le temps réel de type haptique. Nous avons donc choisi pour l'instant de ne pas gérer la déformation des objets en dehors des astuces classiques basées sur des discrétisations masse/ressort.

Modélisation géométrique

L'utilisation des fonctions implicites [PA05] est de plus en plus courante pour la génération de formes complexes dans le domaine de l'animation. On peut aussi employer ces fonctions pour générer des formes à partir d'entrée à main levée en 2D [SC02]. Les EDP (équations différentielles partielles) volumétriques implicites [DU04] permettent d'obtenir à partir de quelques tracés dans l'espace, des surfaces lissées modifiables par variation d'iso-valeur. La figure 1.41 montre (a) les trois courbes en 3D, (b) et (c) représentent les surfaces pour deux iso-valeurs et (d) une coupe de la forme implicite.

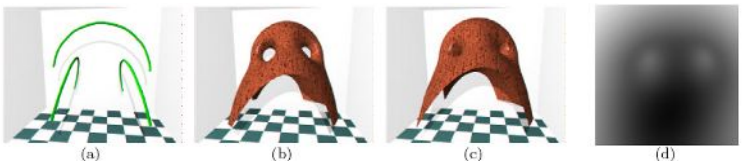


Figure 1.41: Obtention de forme avec les EDP volumétriques implicites [DU04]

Plusieurs méthodes sont mises au point pour obtenir un tracé en 3D à partir d'une entrée en 2D. Des systèmes projectifs originaux [TO01][KA01] sont utilisés et des générations de surfaces [HI01] suivant l'axe de la vue. Des techniques de blobbing [BO02][KA02] sont couramment utilisées, elles permettent d'utiliser les tracés simulant un ombrage ou une accentuation de la forme (Fig. 1.42). La sculpture de ces formes peut être accélérée [FE02][RA03] en utilisant des décompositions de type octree. Des fonctions complexes comme les RBF (Radial Basis Functions) [CA02][OH02], les MPU (Multi-level Partition of Unit) [OH03]

permettent de faire un lien avec les objets numérisés.

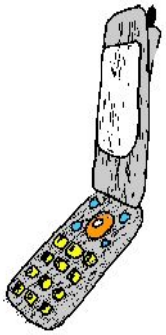


Figure 1.42 : Téléphone portable modélisé avec MagicalSketch, visualisé avec un rendu non réaliste [KA02]

Pour représenter des traits plus proches des coups de crayon, nous pouvons utiliser des textures sur des rectangles allongés qui sont modifiés au cours de la rotation du point de vue [SC01]. Il est possible aussi d'essayer d'imiter les tracés de coups de pinceau par diffusion de l'encre [HU01].

La modélisation d'objets peut être obtenue grâce à des paires de courbes axiales [HI03], facilitant la modélisation de formes longilignes.

Une aide à la génération de nouvelles formes peut être apportée grâce aux réseaux de neurones [HS01], au raisonnement à base de cas [GI02], aux algorithmes évolutionnistes [LI02] et aux grammaires de formes [PA06].

L'utilisation de la géométrie algébrique [HI06] est une voie à explorer. Hitzer a développé Kamiwaai [HI04], un système interactif 3D écrit en Java permettant de montrer les possibilités offertes par la géométrie algébrique pour la représentation d'objets simples (point, segments, cercles, sphères, etc.).

Un nombre très important de formes dans les pièces mécaniques peut être représenté par des quadriques. Les calculs peuvent dans certains cas être accélérés, comme c'est le cas pour la recherche d'intersection [DU03].

Gestion des contraintes

Il existe peu de systèmes de gestion de contraintes géométriques dans l'espace. Afin de simplifier la résolution des équations, Frontier [SI01][OU01] cherche les clusters (regroupement de variables ou d'équations) dans le graphe des contraintes. Une présentation très formelle du système est décrite dans [HO05][HO06]. Une autre technique utilise la décomposition en arbre de connectivité [GA01]. La notion de degrés de rigidité est développée par Jermann [JE01] en parallèle avec une résolution par intervalles. Messine [ME02] utilise des algorithmes de type Branch and Bound par intervalles pour la résolution exacte de problèmes d'optimisation globale, continus ou mixtes, et assujettis ou non à des contraintes. La propagation des contraintes sur les intervalles peut aussi être utilisée pour l'étalonnage des robots [BA06].

La librairie commerciale la plus utilisée pour les sketchers est D-Cube (UGS).

Cette librairie gère aussi bien les géométries exactes que celles facétisées et les contraintes sont en 3D (Fig. 1.44). Il n'y a pas de liaison directe avec la dynamique celle-ci est gérée par d'autres logiciels, comme dans l'exemple du système de retournement de pièces ci-dessous (Fig. 1.43).

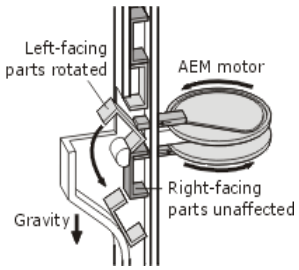


Figure 1.43 : Système de retournement de pièces

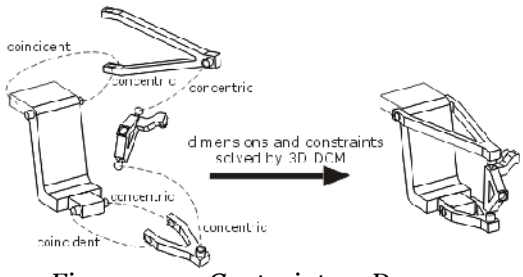


Figure 1.44 : Contraintes 3D sur un ensemble de pièces

La gestion des contraintes de contact est rarement traitée, ce sujet se trouve être à la frontière entre la gestion des contraintes et la détection de collision. Une analyse [XI01] en 2D sur des segments montre déjà les difficultés si l'on recherche tous les cas possibles de contact. Baraff [BA05] définit lui aussi une technique permettant d'obtenir ce type de contrainte et son intégration dans une méthode analytique pour la simulation dynamique.

Nelson et Cohen [NE04] proposent une gestion simultanée des contraintes géométriques et de la dynamique en effectuant une résolution symbolique des équations dues aux contraintes. Ils parlent ainsi de variation interactive de la conception. Par contre, toutes les équations doivent être précalculées ce qui n'est pas utilisables dans un sketcheur.

Modélisation à partir de squelette de pièces

Un des rares formalismes a été proposé par Horvath et van der Vegte [HO01][HO02] mais n'a pas été réutilisé dans d'autres travaux. Une ontologie [HO04] a été proposée pour supporter ce formalisme. La figure 1.45 montre la simulation d'un dépôt d'objet dans une poubelle à papier, le squelette des pièces est défini à l'aide de Nucleus.

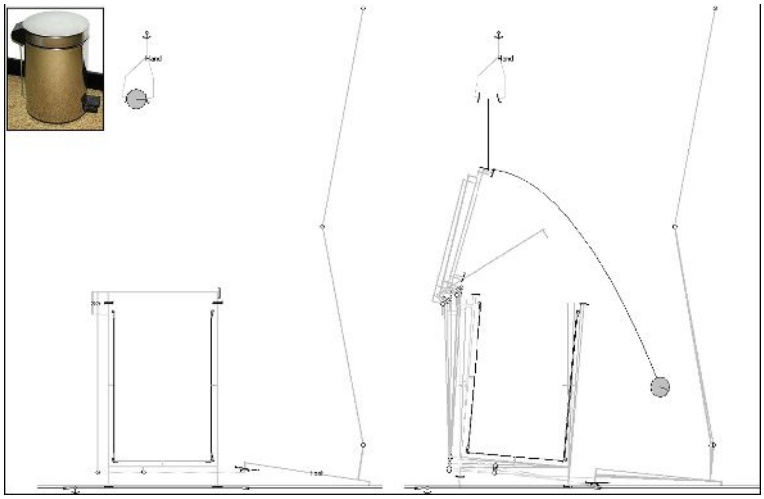


Figure 1.45 : Simulation de l'utilisation d'une poubelle à papier modélisée par son squelette [HO01]

La possibilité d'obtenir des formes à partir du squelette [FR01][CO02][AN01][YO02] est souvent utilisée en animation pour la création de personnages ou d'éléments organiques [DE02] mais aussi en conception de produit [DU01]. On peut illustrer la conception basée sur le squelette sur l'exemple du capot d'une cafetière (Fig. 1.46). Dans ce cas seul le profil est modifié, le squelette reste le même (Fig. 1.47). La déformation de personnage est facilitée par la simple modification des points clés du squelette [KH01]. Il est possible de décrire les objets de révolution à l'aide de langages spécialisés (PSL: Prescribed Sketching Language) [FA01] permettant ainsi un transfert plus léger des schémas sur Internet ou entre téléphones portables.

Figure 1.46 : Deux styles de cafetières. Source www.alessi.it

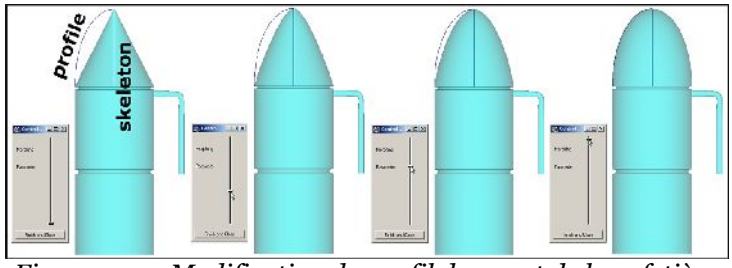


Figure 1.47 : Modification du profil du capot de la cafetière [DU01]

La génération de volume enveloppe (Swept Volumes) peut être considérée comme une géométrie générée à partir d'un squelette qui est la trajectoire de la pièce. Dans les systèmes mécaniques et d'autant plus dans mon sketcheur, générer les enveloppes des pièces en mouvement les unes par rapport aux autres peut être d'une grande utilité comme en simulation d'usinage [MA02]. Mais la génération d'enveloppe est complexe [AB01]. Un exemple développé dans [PE02] montre qu'avec un objet à déplacer relativement simple (Fig. 1.48), on obtient rapidement un volume enveloppe complexe (Fig. 1.49). Plusieurs méthodes de lissage sont proposées pour la surface de l'enveloppe, par triangles, par B-Spline ou par points avec des surfels. La figure 1.50 montre une comparaison entre un lissage par triangles et un lissage par B-Spline.

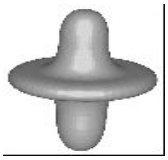


Figure 1.48 : Objet déplacé [PE02]

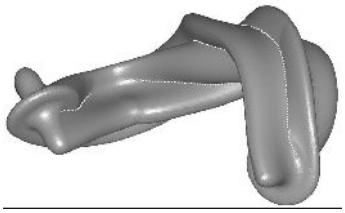


Figure 1.49 : Volume enveloppe généré [PE02]

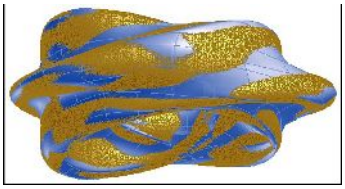


Figure 1.50 : Comparaison entre deux modèles générés soit avec des B-Splines soit avec des triangles [PE02]

Les cylindres généralisés sont une autre forme de modèle utilisant le squelette, ils sont très utilisés pour générer des plantes ou les touffes de cheveux [KI01][MA01] (Fig. 1.51). Leur utilisation peut aussi être envisagée pour des formes de type tuyau [CH05].

La recherche de squelette à partir de surfaces polygonalisées permet de détecter les zones saillantes [HI03] utiles pour la détection de collision. On peut en déduire un squelette filaire ou surfacique [DE04]. Ils peuvent être hiérarchiques [CO01] ce qui permet de définir le niveau de détail (Fig. 1.52). Son utilisation pour la génération d'arbre de sphères [BR02] en détection de collision sera utile dans mon travail.

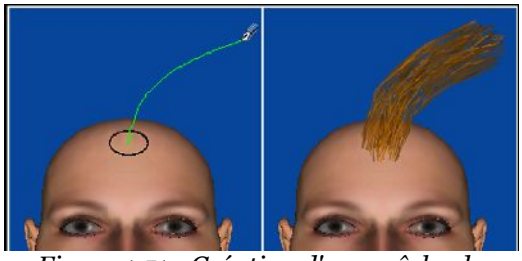


Figure 1.51 : Création d'une mêche de cheveu à partir d'un tracé à main levée [KI01]

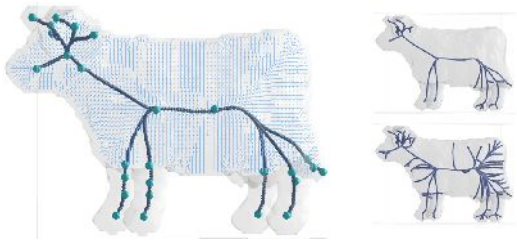


Figure 1.52 : Génération de squelette hiérarchique à différents niveaux [CO01]

La création de formes avec squelette à partir d'un nuage de points [FE01] permet aussi de faire un lien avec les systèmes précédents.

Le squelette d'objets articulés peut être à la base de la capture de mouvement à partir de plusieurs caméras vidéo [BR01]. La déformation de la peau d'un objet articulé, un avatar par exemple, utilise des techniques de skinning plus ou moins évoluées [KA04]. Le couplage squelette/sphères est aussi envisagé pour obtenir une déformation interactive de corps pour le médical [CO07].

L'utilisation du squelette de pièce en phase de conception avancée fait l'objet de quelques recherches [TH02].

Dans nos outils d'esquisse, la modélisation par squelette est intrinsèquement utilisée ce qui correspond de plus à notre choix de suivre au plus près la méthode de conception d'Ullman.

Interfaces graphiques

Les interfaces graphiques en réalité virtuelle utilisent peu le clavier et il est souvent nécessaire de revoir l'activation des fonctions [AP01]. La gestion des différents périphériques peut s'effectuer à l'aide de bibliothèques spécialisées [HU02].

Les recherches effectuées sur les interfaces 2D sont bien entendu applicables aux interfaces 3D. On essaye de placer l'information importante au centre de l'écran et de faciliter ensuite les zooms pour affiner la recherche [KI04].

Le développement d'interfaces graphiques complexes nécessite d'effectuer des choix de langage et de librairies judicieux. La portabilité est aussi un point important comme la possibilité de faire fonctionner ces interfaces via Internet. Ces différents points sont particulièrement bien gérés grâce au langage Java comme on peut le constater dans l'exemple suivant : Synthetica [SU02] (Fig. 1.53).

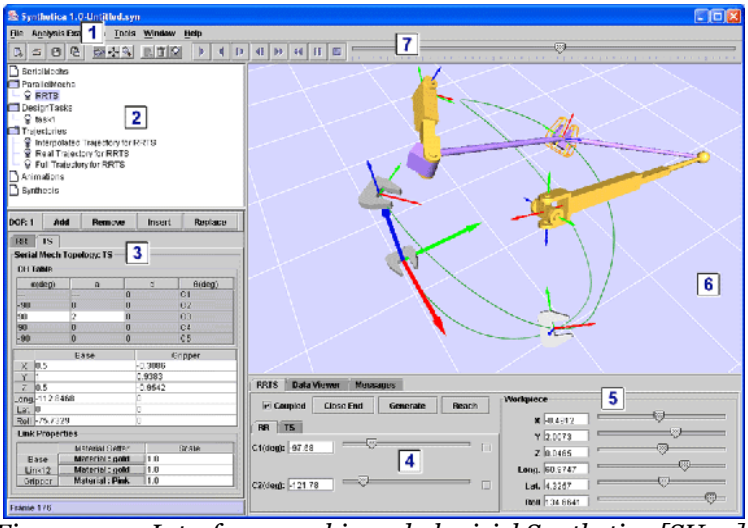


Figure 1.53 : Interface graphique du logiciel Synthetica [SU02]

L'utilisation de visualisation 3D impose l'utilisation d'interface graphique directement manipulable dans l'espace sans utilisation du clavier. De nouvelles interfaces répondent à cette exigence, le nouveau système d'exploitation de Microsoft intègre l'utilisation de la 3D pour la manipulation des fenêtres. Sun développe le projet LG3d en surcouche à Java3D (Fig. 1.54). Ce projet sert lui-même de base pour le projet Wonderland dans lequel l'utilisateur est représenté par un avatar et effectue toutes les interactions par son intermédiaire (Fig. 1.55).

Figure 1.54 : Projet lg3d

Figure 1.55 : Projet Wonderland

Gestion du son

Lors de la simulation de mécanisme comme la pascaline (machine à calculer de Blaise Pascal), nous nous sommes rendu compte de l'importance du son lors de l'animation en 3D. Plus la scène est complexe au niveau visuel, plus le son devient un atout. Il est possible d'entendre un choc qui a eu lieu en dehors du champ de vision.

Le son 3D pose le problème de la localisation de la source sonore par rapport à sa perception. Si l'utilisateur ne bouge pas, il est possible d'utiliser des artifices donnant des impressions de son 3D même avec un simple casque stéréophonique. Cette technique est celle du son binaurale. La méthode classique est celle utilisant 4 ou 6 haut-parleurs situés autour de l'utilisateur ou un ensemble plus importants de haut-parleurs situés dans un caisson devant celui-ci. Les deux librairies les plus utilisées pour la programmation du son 3D sont

DirectSound3D de Microsoft et OpenAL qui est intégré dans de nombreux outils de développements de réalité virtuelle.

Si l'on veut que le son dépende de la position et de l'orientation de l'utilisateur, il faut repérer la position de celui-ci avec des techniques classiques de tracking ou utiliser l'équivalent de l'holographie pour le son : la synthèse de champ sonore holophonique (Fig. 1.56). Cette technique utilise soit une multitude de haut-parleurs soit des panneaux MAP (Multi Actuators Panel, voir figure 1.57).

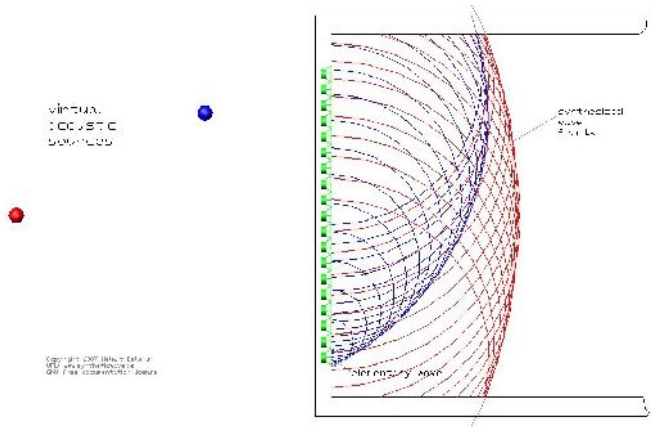


Figure 1.56 : Principe du WFS (Wave Field Synthesis)

Figure 1.57 : Panneaux MAP développés à l'Université de Delft

L'utilisation du son, couplé avec l'haptique peut permettre d'améliorer l'utilisation des programmes [RA04].

Outils de développement propres à la réalité virtuelle

La modélisation de scène pour la réalité virtuelle possède certaines spécificités qui ont conduit les développeurs à créer des librairies dédiées. Elles sont toutes des surcouches à des librairies de bas niveau, les deux les plus utilisées étant OpenGL et DirectX. La recherche de performance lors de la navigation a conduit à une structure arborescente de la scène avec la possibilité de rendre inactives certaines branches. Les librairies de réalité virtuelle sont par exemple en C++ : OpenSG, OpenSceneGraph, OpenMask et en Java : Xith3D, Jogl, Java3D. Ces librairies de moyen niveau sont très souvent intégrées dans des outils de développement incluant d'autres aspects de la réalité virtuelle, gestion des périphériques, gestion du son, gestion des avatars, etc.

Les outils de développement dépendent beaucoup du langage de programmation utilisé, dans le monde Java, ces outils sont tous gratuits, simples d'utilisation, dans le monde C++, le langage par lui même est un frein à une programmation de qualité, sans erreur, sans débordement de mémoire, etc. Les problèmes de coût et de changement de version sont aussi une limitation dans l'utilisation de ces outils. La conséquence est que les environnements de développement en C++ cherchent aussi à rendre plus facile la programmation, l'exemple qui montre bien cette tendance est XVR (www.vrmedia.it) qui utilise son propre langage de script et ne nécessite pas d'outil de développement en C++. La difficulté majeure lors du choix d'un de ces outils de développement est la pérennité de la solution, les outils commerciaux ne sont pas plus stables à cet égard que ceux libres de droit. Même si les standards manquent en réalité virtuelle, il est donc très important d'essayer d'utiliser ceux qui existent.

Dans la panoplie très large d'outils de développement actuellement disponibles,

nous allons présenter rapidement ceux qui nous semblent être d'un intérêt certain. En Java c'est simple deux outils gratuits sont utilisés, Netbeans (www.netbeans.org) plus orienté pour la réalisation d'interface graphique et Eclipse (www.eclipse.org). Ces deux outils possèdent la possibilité de réaliser des analyses de performance des programmes développés (Profiling).

En C++, les performances graphiques des outils de développement sont meilleures par une intégration plus rapide des nouvelles fonctions des cartes graphiques. Le manque de norme pousse cependant à la prudence, certains développements ne fonctionnent par exemple que sur certaines cartes. L'utilisation des GPU pour effectuer des calculs scientifiques est certe élégante mais risquée, SOFA pour Simulation Open Framework Architecture [AL03] fait appel à la technologie CUDA (Compute Unified Device Architecture) de Nvidia. CUDA permet d'effectuer de nombreux calculs scientifiques (calculs matriciels, FFT, ondelettes, etc.) directement dans la GPU accélérant ceux-ci surtout s'ils sont directement liés au rendu graphique. La figure 1.58 montre SOFA sans CUDA tournant à 5 fps et la figure 1.59 avec CUDA à 30 fps (copie d'écran sur le site www.sofa-framework.org).

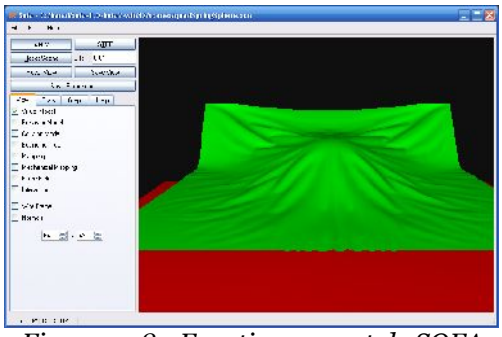


Figure 1.58 : Fonctionnement de SOFA sans CUDA [AL03]

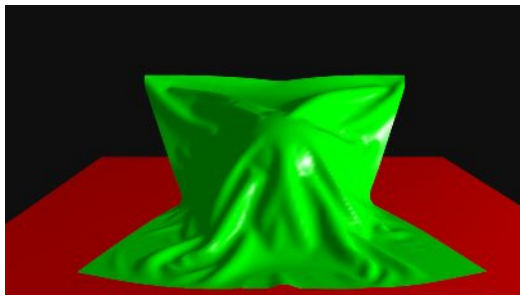


Figure 1.59 : Fonctionnement de SOFA avec CUDA [AL03]

Nous présentons deux outils de développement qui sont tous les deux des surcouches à OpenSceneGraph : Ogre3D (www.ogre3d.org) et Delta3D (www.delta3d.org). Elles intègrent toutes les deux OpenAL pour le son et OpenDE pour la dynamique. Toutes les deux gèrent l'animation des personnages, Delta3D utilise la librairie Cal3D pour cela.

Les figures ci-dessous présentent deux applications de formation réalisées avec ces outils de développement, une pour la formation aux premiers gestes en cas d'accident (Fig. 1.60) et l'autre pour la formation au déminage (Fig. 1.61). On peut remarquer la qualité graphique obtenue en temps réel avec de nombreux calculs de dynamique pour les particules qui permettent de simuler le feu, la fumée, les nuages, etc.

Delta3D est principalement utilisé comme outil de simulation par l'armée américaine avec des liaisons en cours avec la norme SCORM (www.adlnet.org) pour la formation et avec la norme S1000D (www.s1000d.org) pour la documention de maintenance.

Figure 1.60 : Simulateur d'aide de première urgence développé par Guppyworks avec Ogre3D

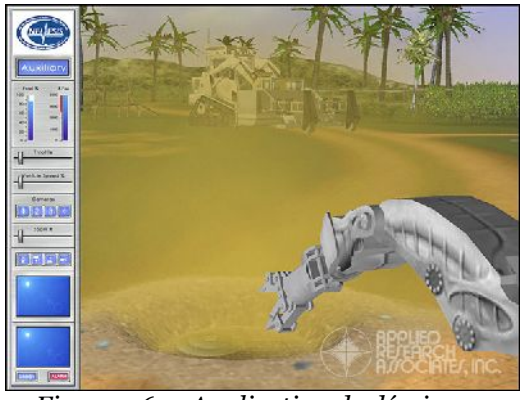


Figure 1.61 : Application de déminage développée par Applied Research Associates, Inc. avec Delta3D

1.3.3 Les problèmes cognitifs

La réalité virtuelle utilisant plusieurs sens pose des problèmes cognitifs qu'il ne faut pas négliger. Dominjon et al. ont fait des essais essayant de vérifier si la perception du poids d'un objet était fiable avec un retour d'effort [DO05]. L'utilisation de vision d'objet en 3D n'est pas non plus évidente, il serait bon de tester les capacités des utilisateurs mais aucun consensus n'existe sur le test à utiliser pour qualifier les capacités de vision spatiale. Les capacités de manipulations en 3D ne sont pas corrélées avec les capacités en mathématiques, elles semblent par contre être corrélées avec les capacités de manipulation en 2D [HO07].

Des incohérences cognitives peuvent intervenir entre la vision et l'haptique [AT02][WA03][WO01][JO02][KI02]. Les aveugles [RI01] et les enfants, nous apprennent beaucoup sur le couplage vision/haptique. Des modèles de perception [MI01] peuvent nous aider à prévoir ces difficultés.

Chercher à résoudre ces problèmes cognitifs en conception implique quelques connaissances en psychologie cognitive et surtout en ergonomie cognitive. La théorie de l'activité [BE01] est une des théories qui permet de décrire ce qui se passe chez le concepteur. Des théories sur les images mentales [PY01] sont aussi très discutées.

Il est possible à contrario d'utiliser certaines difficultés cognitives pour accélérer les simulations, nous pouvons par exemple tolérer des erreurs dans les collisions qui ne se situent pas dans la zone de visée de l'utilisateur [OS02].

Un résultat récent du projet ECOVIA [LE04] permet d'indiquer qu'il est parfois préférable de délocaliser la source haptique de la source visuelle, en fonction de la qualité de la source haptique. Nous retenons la délocalisation dans nos deux prototypes, nous pensons qu'un apprentissage a eu lieu chez les utilisateurs d'ordinateurs avec la souris et chez les graphistes avec la tablette à digitaliser. La délocalisation est ainsi devenue naturelle et ne demande plus de surcharge cognitive.

La prise en compte des problèmes cognitifs dans la réalisation de nos prototypes a été recherchée même s'il est important de rester modeste et accepter les difficultés pour des mécaniciens des déterminer les éléments pertinents à prendre

en compte. Ignorer ce point serait une erreur et rendrait totalement impossible la mise en place et l'interprétation de tests par la suite.

1.4 Les outils d'esquisse existants

Plusieurs outils existent en laboratoire, de nombreux points communs sont repérables comme l'utilisation de la vision stéréoscopique, le travail à deux mains, etc. Il paraît important de noter les différences de point de vue quand à la finalité de ces outils et les choix de solutions effectués.

Le Tacitus Project [SH01] réalisé en commun avec l'Edinburgh College of Art et l'Edinburgh Virtual Environment Centre permet de réaliser le design d'objet à l'aide d'un système à retour d'effort en traçant des segments dans l'espace. Seuls des objets statiques sont créés, par contre de nombreux tests de comparaisons ont été réalisés sur l'intérêt de l'haptique et de la stéréoscopie [SC04]. On peut remarquer (Fig. 1.62) l'utilisation d'un système Reachin (vitre semi-réfléchissante avec lunettes stéréoscopiques), d'un Phantom Desktop et d'une SpaceMouse. La difficulté de dessiner des traits dans l'espace a été montrée dans ce projet comme on peut le voir sur la figure 1.63.

Figure 1.62 : Utilisation du système Reachin [SH01]

Figure 1.63 : Défaut du tracé haptique [SC04]

Le 3D-Sketcher proposé par Martin Pache et Udo Lindemann est encore plus proche de ma proposition mais il ne représente que des systèmes mécaniques sans mouvement. Les mouvements sont simplement indiqués par la création de flèche [DI03], les figures suivantes (Fig. 1.64 et 1.65) montrent deux exemples de conception de support de tasse :

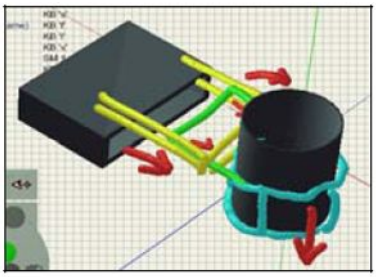


Figure 1.64 : Schéma d'un support de tasse 1 [DI03]

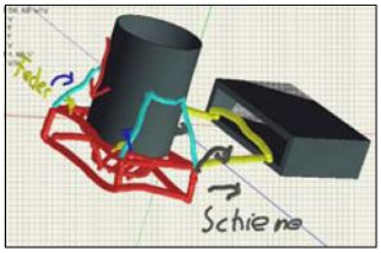


Figure 1.65 : Schéma d'un support de tasse 2[DI03]

Les réflexions de Lindeman sur ce sujet sont très intéressantes, celui-ci a même organisé en mars 2003 le colloque Human Behavior in Design [LI05]. Un processus de concrétisation [DI02] a été proposé qui permet à partir des segments irréguliers tracés dans l'espace d'obtenir une forme par apprentissage ou par sélection. Les derniers travaux sur le 3D-Sketcher cherchent à faire un lien avec la CAO en utilisant OpenCascade [DI01], ce choix ne semble pas très judicieux mais mérite d'être testé.

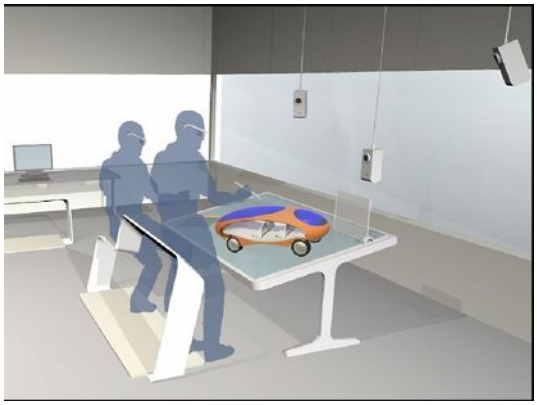


Figure 1.66 : Proposition de travail en 3D dans le cadre du projet SmartSketches [BA02]

Le SmartSketches project qui a fait une analyse détaillée de l'utilisation des croquis dans les processus de conception. Une des dernières présentations [BA02] de ce projet propose de travailler directement en 3D comme le montre la figure 1.66.

Le projet Touch and Design [BO01] qui cherche à reproduire la réalisation de prototype physique pour les designers. Les gestes des sculpteurs (Fig. 1.67) ont été analysés et des outils haptiques ont été mis au point en fonction.

Figure 1.67 : Analyse des gestes des sculpteurs [BO01]

Le prototype mis au point utilise deux systèmes haptiques en parallèle pour permettre le travail à deux mains.

Figure 1.68 : Prototype du projet Touch and Design [BO01]

Yoshida et al. [YO03] proposent un outil pour le style automobile. La projection stéréo s'effectue sur un mur en stéréo ce qui est logique pour ce type d'application. La manipulation est à deux mains mais il n'y a pas d'utilisation du retour d'effort, on peut douter de la possibilité de créer le tableau de bord présenté en figure 1.69 simplement avec une vision stéréoscopique.

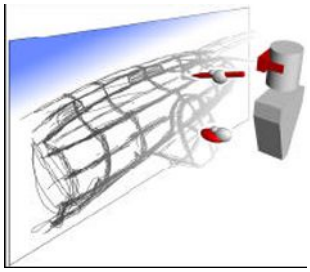


Figure 1.69 : Système de Yoshida et al.[YO03]

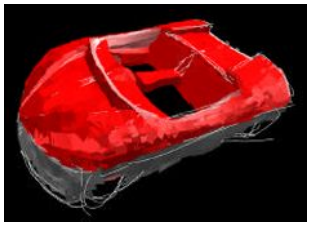


Figure 1.70 : Résultat brut à partir du sketcher [YO03]

Les quelques croquis présentés sont d'assez mauvaise qualité (Fig. 1.70). Pour corriger le résultat, un post traitement est effectué en utilisant des NURBS.

Figure 1.71 : Sketch avant traitement [YO03]

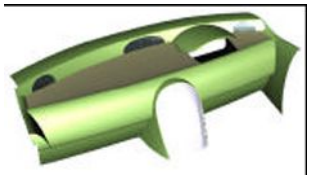


Figure 1.72 : Sketch après traitement [YO03]

Ye et al. [YE01] font une proposition très proche de celle de Lindemann mais ils choisissent dès le départ de réaliser un nouveau système de CAO ce qui ne correspond pas au titre de leur article : « An investigation into the implementation of virtual reality technologies in support of conceptual design ». Les objets manipulés sont des NURBS et leur interface ressemble beaucoup à une interface de système de CAO. Des tests ont été réalisés mais l'on ne connait pas l'origine des 16 testeurs. 5 points ont été évalués en terme de degrés d'utilité :

- opération à deux mains 3.9375
- vue stéréoscopique 3.8125
- interaction haptique 3.375

- retour sonore 3.0625
- souris/clavier 2.90625

Nous pouvons remarquer que les systèmes présentés utilisent tous le travail à deux mains et que l'haptique et la stéréoscopie sont nécessaires. On remarque aussi que les systèmes utilisant le retour d'effort sont tous architecturés de la même manière. Nous retiendrons cette solution mais en essayant de réfléchir à l'ergonomie du poste de travail comme nous le montrerons au chapitre 4.2.1.

Tous les outils d'esquisse présentés ne permettent que le tracé en 3D d'objets statiques. Nous souhaitons créer des systèmes mécaniques qui par définition sont dynamiques. Le chapitre 2 va nous permettre de montrer comment nous prenons en compte le mouvement des pièces avec ou sans retour d'effort.

2 Dynamique dans l'esquisse virtuelle

L'esquisse virtuelle a pour but dans notre cas de concevoir des systèmes mécaniques qui sont par définition des systèmes multi-corps régis par les lois de la dynamique. Nous avons dans notre premier outil, utilisé une librairie de dynamique OpenDE et une génération d'effort entre pièces utilisant des ressorts et des amortisseurs. Ce modèle ayant montré ses limites, nous avons remplacé les ressorts et les amortisseurs par des chocs et des glissements ce qui nous a conduit à utiliser une détection de collision continue et une dynamique sous contraintes. Nous présentons dans ce chapitre, cette méthode ainsi que le mode d'obtention des contraintes et de leurs résolutions.

2.1 Dynamique sous contraintes

La dynamique sous contraintes que nous utilisons est obtenue à partir du principe de Gauss de moindres contraintes, ce principe est utilisé dans la librairie Contact développée par Redon[RE05]. Nous n'avons pas retenu par contre la méthode de résolution utilisée par Redon qui aboutit à un problème équivalent au calcul de point le plus proche (Nearest-point problem) et qu'il résout avec l'algorithme de Wilhelmsen. Egan [EG01] critique ce choix en indiquant qu'il existe peu de littérature sur les méthodes de résolution des problèmes de calcul de point le plus proche.

Après quelques recherches sur les différentes utilisations du principe de Gauss des moindres contraintes, nous avons trouvé une formulation qui ne nécessite pas de calcul de minimisation. En effet, Udwadia et Kalaba [UD01] proposent une formulation du principe de Gauss qui possède de nombreux avantages.

Nous commencerons par faire un rappel du principe de Gauss de moindres contraintes, nous présenterons ensuite la formulation de Udwadia et Kalaba. Cette formulation nécessite d'effectuer des calculs particuliers comme la racine d'une matrice, une matrice pseudo inverse, etc. Nous indiquerons les choix que nous avons effectués pour simplifier ou accélérer les calculs de ces éléments.

2.1.1 Rappel du principe de moindres contraintes de Gauss

Gauss a énoncé son principe de moindres contraintes [GA03] comme suit :

Étant données les masses des points m, m', m'' , a, a', a'' les positions de ces points au temps t ; b, b', b'' les positions que ces points prendraient s'ils étaient totalement libres après un petit pas de temps dt sous les forces agissant pendant ce temps sur ceux-ci et à partir des vitesses et des directions atteintes au temps t , prendraient. Les vraies positions c, c', c'' seront alors celles qui, respectant les contraintes du système, minimisent :

$$m(bc)^2+m'(b'c')^2+m''(b''c'')^2$$

Dans son article Gauss propose une démonstration de ce principe mais celle-ci est mise en défaut dans plusieurs articles [DE08][GE02], en effet la condition du minimum ne serait valable que dans certains cas. Le principe de Gauss des moindres contraintes s'écrit généralement dans l'espace des accélérations mais il n'est jamais fait référence à des conditions d'application.

Nous allons maintenant montrer comment l'écriture originale du principe peut conduire à sa forme dans l'espace des accélérations. La somme à minimiser dans le principe de Gauss peut être mise sous la forme :

$$\sum m_i(\overrightarrow{B_i(dt)C_i(dt)})^2 \tag{2.1}$$

Où m_i est la masse du ième point, $\overrightarrow{C_i(dt)}$ est la nouvelle position du point quand celui-ci est soumis aux contraintes et $\overrightarrow{B_i(dt)}$ serait la position du point dans un mouvement sans contraintes.
Pour passer des positions aux accélérations, on intègre deux fois l'accélération sans contraintes $\vec{a}_i$ et l'accélération sous contraintes $\vec{\ddot{x}}_i$, ce qui donne :

$$\begin{aligned}\overrightarrow{OC}_i(t)&=\frac{1}{2}\vec{\ddot{x}}_i t^2+\vec{v}_{A_i}t+\overrightarrow{OA}_i\\ \overrightarrow{OB}_i(t)&=\frac{1}{2}\vec{a}_i t^2+\vec{v}_{A_i}t+\overrightarrow{OA}_i\end{aligned} \tag{2.2}$$

On trouve alors :

$$\overrightarrow{B_i}(dt)\overrightarrow{C_i}(dt)=\frac{1}{2}(\vec{\ddot{x}}_i-\vec{a}_i)dt^2 \tag{2.3}$$

La somme à minimiser devient donc :

$$\sum m_i(\vec{\ddot{x}}_i-\vec{a}_i)^2 \tag{2.4}$$

Le principe des moindres contraintes de Gauss s'écrit donc : pour un ensemble de points matériels de masse m_i , d'accélération sans contraintes $\vec{a}_i$, après un petit pas de temps dt , les nouvelles accélérations $\vec{\ddot{x}}_i$ respectant les contraintes du système minimisent la somme de l'équation 2.4.
L'extension de ce principe à la dynamique de corps rigides est évidente.
Malgré les quelques limitations mentionnées ci-dessus, le principe de Gauss de moindres contraintes est aujourd'hui couramment utilisé. Ce principe permet de passer outre les limitations des méthodes classiquement utilisées en dynamique. Plusieurs recherches récentes [CH08] prouvent que la dynamique sous contraintes non holonomes ne doit pas ou ne peut pas être résolue à l'aide des multiplicateurs de Lagrange. Tatarinov [TA02] écrit en parlant de la dynamique des systèmes non-holonomes : « *Les équations avec les multiplicateurs de Lagrange sont hors de considération dans ce qui suit, car elles ne sont pas totalement différentielles au départ* ».

2.1.2 Formulation d'Udwadia et Kalaba

Une des difficultés lors de l'utilisation du principe de Gauss est le fait que l'on recherche un minimum. Redon a transformé la recherche de minimum en recherche du point le plus proche mais la méthode qu'il utilise pour cette recherche manque de référence. Nous avons préféré, l'utilisation de la formulation d'Udwadia et de Kalaba qui va nous permettre d'obtenir une relation directe entre les accélérations non-contraintes et contraintes. De plus l'extension aux contraintes non idéales est elle aussi directe.

Comme le souligne Kneller, dans la dynamique des systèmes moléculaires [KN01], cette formulation n'est pas nouvelle. Il reconnait cependant que l'utilisation de matrice pseudo-inverse ouvre une perspective numérique intéressante. Nous pouvons aussi trouver d'autres formulations en utilisant la notion de force contrainte généralisée [OR01].

2.1.2.1 Contraintes idéales

L'obtention de la formulation d'Udwadia et de Kalaba est simple à obtenir dans ce cas. On peut écrire le terme à minimiser de la ligne 2.4, sous la forme :

$$G=\sum_{i=1}^{p} m_i(\ddot{x}_i-a_i)^T(\ddot{x}_i-a_i) \tag{2.5}$$

pour p particules. Où sous forme matricielle :

$$G=(\ddot{\boldsymbol{x}}-\boldsymbol{a})^T\boldsymbol{M}(\ddot{\boldsymbol{x}}-\boldsymbol{a}) \tag{2.6}$$

Dans laquelle les différents termes peuvent être définis ci-dessous :

$$\ddot{\boldsymbol{x}}=\begin{bmatrix}\ddot{x}_1\\ \ddot{x}_2\\ .\\ .\\ \ddot{x}_{3p}\end{bmatrix} \tag{2.7}$$

$$\boldsymbol{a} = \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ a_{3p} \end{bmatrix} \tag{2.8}$$

$$M = \begin{bmatrix} m_1 & 0 & 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 \\ 0 & m_1 & 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 \\ 0 & 0 & m_1 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 \\ 0 & 0 & 0 & m_2 & 0 & 0 & . & . & . & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_2 & 0 & . & . & . & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & m_2 & . & . & . & 0 & 0 & 0 \\ . & . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & 0 & 0 & . & . & . & m_p & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & . & . & . & 0 & m_p & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & m_p \end{bmatrix} \tag{2.9}$$

Les contraintes sont exprimées par des égalités qui sous formes matricielles donnent l'équation suivante:

$$\boldsymbol{A}\ddot{\boldsymbol{x}} = \boldsymbol{b} \tag{2.10}$$

On pose alors :

$$\boldsymbol{y} = \boldsymbol{M}^{1/2}(\ddot{\boldsymbol{x}} - \boldsymbol{a}) \tag{2.11}$$

D'où :

$$\ddot{\boldsymbol{x}} = \boldsymbol{M}^{-1/2}\boldsymbol{y} + \boldsymbol{a} \tag{2.12}$$

A partir des équations 2.6 et 2.10, on peut écrire :

$$G = \boldsymbol{y}^T \boldsymbol{y} \tag{2.13}$$

Et :

$$\boldsymbol{A}\boldsymbol{M}^{-1/2}\boldsymbol{y} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{a} \tag{2.14}$$

On multiplie les deux membres de l'équation 2.14 par la pseudo-inverse de $\boldsymbol{A}\boldsymbol{M}^{-1/2}$: $(\boldsymbol{A}\boldsymbol{M}^{-1/2})^+$ ce qui conduit à la solution :

$$\boldsymbol{y} = (\boldsymbol{A}\boldsymbol{M}^{-1/2})^+(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{a}) \tag{2.15}$$

Ce qui permet de trouver les accélérations sous contraintes :

$$\ddot{\boldsymbol{x}} = \boldsymbol{a} + \boldsymbol{M}^{-1/2}(\boldsymbol{A}\boldsymbol{M}^{-1/2})^+(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{a}) \tag{2.16}$$

Udwadia et Kalaba démontrent [UD01] que le terme

$M^{-1/2}(AM^{-1/2})^{+}(b-Aa)$ ne fournit aucun travail quelque soit le déplacement virtuel choisi d'où le terme de contrainte idéale. Dans ce cas le principe de D'Alembert est vérifié, ce qui ne sera plus le cas avec l'ajout de contraintes non idéales.

2.1.2.2 *Contraintes non idéales*

Udwadia et Kalaba propose une formulation plus complète permettant d'ajouter des efforts conduisant à une perte de travail. Dans l'équation 2.16, le terme :

$$M^{-1/2}(AM^{-1/2})^{+}(b-Aa) \tag{2.17}$$

correspond uniquement aux efforts normaux dans les contacts. On peut donc montrer que ceux-ci ne fournissent aucun travail. L'équation générale prend la forme :

$$\ddot{x}=a+M^{-1/2}(AM^{-1/2})^{+}(b-Aa)+M^{-1/2}(I-(AM^{-1/2})^{+}(AM^{-1/2}))z \tag{2.18}$$

Udwadia et Kalaba propose pour expliciter les différents termes, de multiplier l'équation 2.18 par M ce qui donne :

$$M\ddot{x}=Q+Q_i^c+Q_{ni}^c \tag{2.19}$$

Q représente les forces d'inertie, Q_i^c les forces idéales dues aux contraintes et Q_{ni}^c les forces non idéales dues aux contraintes. Une force est dite idéale si elle ne créée pas de travail lors d'un déplacement virtuel respectant les contraintes.

Quelques points importants de cette formulation générale doivent être donnés :

- Cette forme générale est applicable à tous les systèmes contraints idéals ou non, qu'ils vérifient ou non le principe de d'Alembert. Cette formulation est obtenue sans faire appel au principe des travaux virtuels.
- Q_{ni}^c doit être donné par le mécanicien après une compréhension du système mécanique à partir de la spécification du vecteur $z(x,\dot{x},t)$.

2.1.3 Éléments de calcul

2.1.3.1 *Détermination des efforts aux points de contact*

Nous avons besoin de connaître dans de nombreux cas, les efforts transmis au niveau des points de contact ou au minimum le sens de cet effort. Cette information permettant de détecter une perte de contact lors d'un glissement ou de déterminer l'effort tangentiel dû au frottement par exemple.

Dans la formulation d'Udwadia et Kalaba, on peut montrer à quoi correspondent les différents termes d'un point de vue effort en multipliant l'équation 2.16 par la matrice d'inertie :

$$M\ddot{x}=Ma+M(M^{-1/2}(AM^{-1/2})^{+}(b-Aa)) \tag{2.20}$$

Ce qui se traduit par le fait que la somme des forces appliquées sur un solide

$M\ddot{x}$ est égale à la somme des forces dues aux accélérations non contraintes Ma et des forces dues aux accélérations contraintes $M(M^{-1/2}(AM^{-1/2})^{+}(b-Aa))$. Si l'on écrit :

$$c=(b-Aa) \tag{2.21}$$

Ce vecteur c est de dimension égale au nombre de contraintes imposées au système. Si l'on cherche à déterminer l'effort dans un contact dû à la ième contrainte, il suffit d'effectuer le calcul suivant :

$$F_i=M(M^{-1/2}(AM^{-1/2})^{+}.\begin{bmatrix}0\\ \dots\\ c_i\\ \dots\\ 0\end{bmatrix}) \tag{2.22}$$

Ces calculs s'effectuent au cours de la détermination de l'accélération et ne nécessitent donc pas d'énormes calculs supplémentaires.

2.1.3.2 *Calcul de la pseudo-inverse*

Nous utilisons la méthode développée par Courrieu [CO08] pour la pseudo-inverse de Moore-Penrose. Cette méthode est rapide mais permet surtout d'obtenir un résultat même si la matrice à inverser est dégénérée ce qui peut rapidement être le cas dans un mécanisme. Dans l'exemple du segment qui tombe à l'intérieur de deux cercles, la méthode de Gréville est en défaut, pas celle de Courrieu.

Si l'on nomme G la matrice $m\,x\,n$ à inverser, nous trouvons la matrice carrée $n\,x\,n$ symétrique positive $G'G$ de rang $r \leqslant n$ où G' est la transposée de G. On peut écrire $G'G$ sous la forme $S'S$ avec S une matrice triangulaire supérieure avec exactement $n-r$ colonnes vides. En enlevant les colonnes vides de S, on peut construire une matrice $r\,x\,n$, nommée L' de rang r. On peut alors écrire :

$$G'G=S'S=LL' \tag{2.23}$$

Courrieu démontre alors le théorème suivant :
Avec G et L comme définis ci-dessus, on a

$$G^{+}=L(L'L)^{-1}(L'L)^{-1}L'G' \tag{2.24}$$

Une formulation plus classique est proposée par Kneller [KN01] pour la dynamique des systèmes moléculaires mais celle-ci nécessite de trouver une décomposition de la matrice G du type :

$$G=AB \tag{2.25}$$

Où A est une matrice $m\ x\ r$ et B, une matrice $r\ x\ n$. Dans ce cas :

$$G^{+}=B(BB')^{-1}(A'A)^{-1}A \tag{2.26}$$

Cette méthode nécessite deux inversions et cinq multiplications matricielles contre une inversion et cinq multiplications pour la méthode de Courrieu. Il faut

en plus trouver $\boldsymbol{A}$ et $\boldsymbol{B}$ à partir de l'équation 2.25.

2.1.3.3 *Calcul de la racine de la matrice d'inertie*

Pour un solide en 3d, la matrice d'inertie est une matrice 6x6. La définition de cette matrice par rapport aux axes principaux d'inertie est :

$$\boldsymbol{M}=\begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x & 0 & 0 \\ 0 & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & 0 & 0 & I_z \end{bmatrix}=\begin{bmatrix} \boldsymbol{M_m}=m\,\boldsymbol{I} & 0 \\ 0 & \boldsymbol{M_I} \end{bmatrix} \tag{2.27}$$

Il est inutile de garder cette matrice sous cette forme car elle contient beaucoup de zéros même après rotation. Comme :

$$\boldsymbol{M^{-1/2}}=\begin{bmatrix} \boldsymbol{M_m^{-1/2}}=m^{-1/2}\,\boldsymbol{I} & 0 \\ 0 & \boldsymbol{M_I^{-1/2}} \end{bmatrix} \tag{2.28}$$

Il est simplement nécessaire d'évaluer $\boldsymbol{M_I^{-1/2}}$ à chaque pas de temps. Si les axes principaux d'inertie ont subi une rotation définie par la matrice $\boldsymbol{Q}$. La nouvelle matrice d'inertie est :

$$\boldsymbol{M_{IQ}}=\boldsymbol{Q^t}\,\boldsymbol{M_I}\,\boldsymbol{Q} \tag{2.29}$$

Si on pose :

$$\boldsymbol{R}\,\boldsymbol{R^t}=\boldsymbol{M_{IQ}^{-1}}=\boldsymbol{Q}\,\boldsymbol{M_I^{-1}}\,\boldsymbol{Q^t} \tag{2.30}$$

Où :

$$\boldsymbol{M_I^{-1}}=\begin{bmatrix} \frac{1}{I_x} & 0 & 0 \\ 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & \frac{1}{I_z} \end{bmatrix} \tag{2.31}$$

On cherche donc :

$$\boldsymbol{R}=\boldsymbol{M_{IQ}^{-1/2}} \tag{2.32}$$

Le calcul de la racine carrée d'une matrice est complexe mais nous sommes ici dans un cas particulier qui nous permet de poser :

$$\boldsymbol{B} = \begin{bmatrix} \frac{1}{I_x^{1/2}} & 0 & 0 \\ 0 & \frac{1}{I_y^{1/2}} & 0 \\ 0 & 0 & \frac{1}{I_z^{1/2}} \end{bmatrix} \tag{2.33}$$

Et donc :

$$\boldsymbol{B}\,\boldsymbol{B}^t = \boldsymbol{M}_I^{-1} \tag{2.34}$$

En remplaçant dans l'équation 2.30 :

$$\boldsymbol{R}\,\boldsymbol{R}^t = \boldsymbol{Q}\,\boldsymbol{B}\,\boldsymbol{B}^t\,\boldsymbol{Q}^t \tag{2.35}$$

On obtient le résultat simple à calculer :

$$\boldsymbol{R} = \boldsymbol{Q}\,\boldsymbol{B} \tag{2.36}$$

L'évaluation de $\boldsymbol{Q}$ à chaque pas de temps est la multiplication de la matrice de rotation du repère de la pièce avec la matrice de rotation du repère des axes principaux d'inertie par rapport à celui de la pièce. Cette dernière matrice est constante. Si l'on écrit :

$$\boldsymbol{Q} = \boldsymbol{R}_O\,\boldsymbol{R}_I \tag{2.37}$$

L'équation 2.36 devient :

$$\boldsymbol{R} = \boldsymbol{R}_O\,\boldsymbol{R}_I\,\boldsymbol{B} \tag{2.38}$$

On peut donc précalculer $\boldsymbol{R}_I\,\boldsymbol{B}$ pour chaque pièce.

2.1.4 Essais de la dynamique sous contraintes en 2D

Avant de passer à l'application en 3D de la dynamique sous contraintes respectant la formulation d'Udwadia et de Kalaba, nous avons effectué des tests en 2D sur un point rebondissant à l'intérieur d'un cercle et un point rebondissant à l'intérieur d'un carré. L'avantage de ces deux cas simples est qu'il est possible d'obtenir de manière littérale l'expression des accélérations.
Nous avons développé ces applets en Java afin d'accélérer les temps de développement et pour profiter des outils d'analyse disponibles de type Profiler disponibles directement sous Netbeans ou Eclipse. L'analyse des temps de calcul sur des problèmes en 2D, nous a permis d'effectuer une pré-optimisation du code pour la 3D.
Nous ne présenterons que les résultats obtenus dans le cas d'un point dans un cercle.

2.1.4.1 Ecriture des contraintes

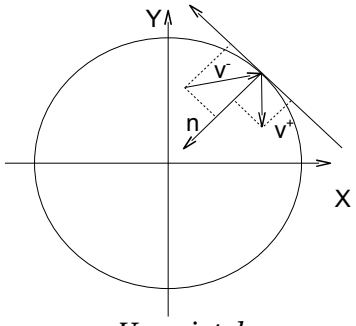


Figure 2.1 : Un point dans un cercle

Contrainte de choc

On peut utiliser la relation suivante donnant le rapport entre la vitesse avant et après le choc :

$$v^{+}.n=-e.v^{-}.n \tag{2.39}$$

Où e est le coefficient de restitution. On supposant que le temps écoulé pour passer de la vitesse avant et après le choc est dt , on peut écrire :

$$\ddot{x}=\frac{(v^{+}-v^{-})}{dt} \tag{2.40}$$

En multipliant par n, on obtient :

$$\ddot{x}.n=\frac{(v^{+}.n-v^{-}.n)}{dt}=-\frac{(1+e)}{dt}v^{-}.n \tag{2.41}$$

On peut écrire à un instant t :

$$\ddot{x}=\begin{bmatrix}\ddot{x}_1\\ \ddot{x}_2\end{bmatrix} \tag{2.42}$$

$$n=-\frac{1}{\sqrt{(x_1^2+x_2^2)}}\begin{bmatrix}x_1\\ x_2\end{bmatrix} \tag{2.43}$$

$$v^{-}=\begin{bmatrix}v_1\\ v_2\end{bmatrix} \tag{2.44}$$

$$\ddot{x}_1.x_1+\ddot{x}_2.x_2=-\frac{(1+e)}{dt}.(v_1.x_1+v_2.x_2) \tag{2.45}$$

On trouve donc A et b pour la contrainte d'un choc :

$$\boldsymbol{A}=[x_1\ x_2]\quad et\quad b=-\frac{(1+e)}{dt}.(v_1.x_1+v_2.x_2) \tag{2.46}$$

Contrainte de glissement

La contrainte de glissement impose que le point reste sur le cercle donc :

$$x_1^2+x_2^2=R^2 \tag{2.47}$$

On dérive deux fois cette équation par rapport à t , d'où :

$$\begin{aligned} v_1 . x_1 + v_2 . x_2 = 0 \\ v_1^2 + v_2^2 + \ddot{x}_1 . x_1 + \ddot{x}_2 . x_2 = 0 \end{aligned} \tag{2.48}$$

La matrice A est identique au cas du choc mais avec :

$$b = -(v_1^2 + v_2^2) \tag{2.49}$$

2.1.4.2 *Détermination des accélérations*

Dans notre cas :

$$M = \begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \tag{2.50}$$

$$M^{-1/2} = \begin{bmatrix} m^{-1/2} & 0 \\ 0 & m^{-1/2} \end{bmatrix} \tag{2.51}$$

$$\boldsymbol{AM^{-1/2}} = [\frac{x_1}{\sqrt{m}} \quad \frac{x_2}{\sqrt{m}}] \tag{2.52}$$

$$(\boldsymbol{AM^{-1/2}})^+ = \frac{m}{(x_1^2 + x_2^2)} \begin{bmatrix} \frac{x_1}{\sqrt{m}} \\ \frac{x_2}{\sqrt{m}} \end{bmatrix} \tag{2.53}$$

$$\boldsymbol{A}a = [x_1 \quad x_2] \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = x_1 a_1 + x_2 a_2 \tag{2.54}$$

Dans le cas du choc, on a donc :

$$\begin{aligned} \ddot{x}_1 &= a_1 - \frac{\frac{(1+e)}{dt} . (v_1 . x_1 + v_2 . x_2) + x_1 . a_1 + x_2 . a_2}{(x_1^2 + x_2^2)} . x_1 \\ \ddot{x}_2 &= a_2 - \frac{\frac{(1+e)}{dt} . (v_1 . x_1 + v_2 . x_2) + x_1 . a_1 + x_2 . a_2}{(x_1^2 + x_2^2)} . x_2 \end{aligned} \tag{2.55}$$

Et en cas de glissement :

$$\begin{aligned} \ddot{x}_1 &= a_1 - \frac{(v_1^2 + v_2^2) + x_1 . a_1 + x_2 . a_2}{(x_1^2 + x_2^2)} . x_1 \\ \ddot{x}_2 &= a_2 - \frac{(v_1^2 + v_2^2) + x_1 . a_1 + x_2 . a_2}{(x_1^2 + x_2^2)} . x_2 \end{aligned} \tag{2.56}$$

2.1.4.3 *Résultats du test en 2D*

Les premiers résultats permettent de tester la robustesse du modèle choisi pour les contraintes de choc avec un choc parfait ($e = 1.0$).

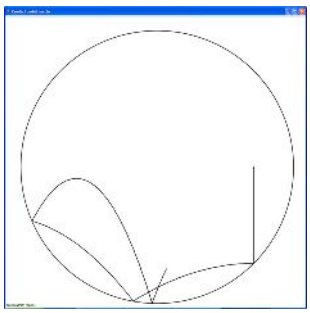
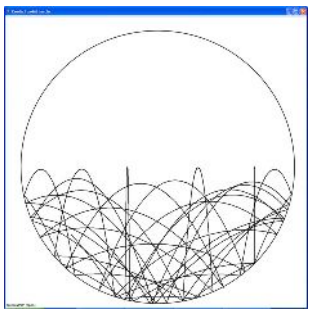
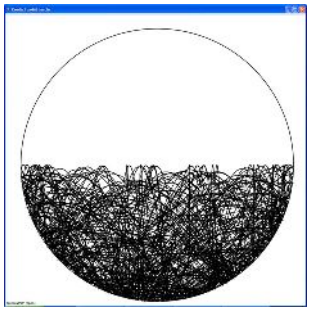
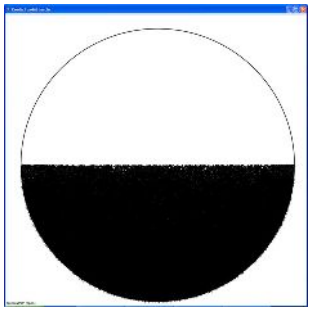
500		5 000	
50 000		500 000	

Tableau 4 : Evolution des chocs en fonction du nombre de cycles

Les 4 simulations ci-dessus permettent de voir que le point remonte correctement à l'horizontale quelque soit le nombre de cycles de calcul (500, 5 000, 50 000, 500 000).
Nous avons essayé de comparer l'utilisation de la dynamique sous contraintes à l'utilisation d'une dynamique plus classique avec ressort et amortisseur pour représenter le contact.

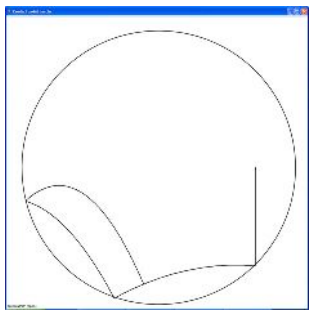
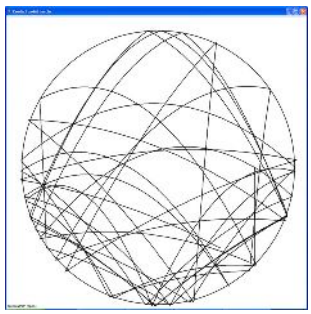
500		5 000	

Tableau 5 : Evolution des chocs en fonction du nombre de cycles avec un contact ressort/amortisseur

Comme pour le premier outil, nous essayons d'éviter que le point ne sorte du cercle en faisant un bon choix de la raideur et de l'amortissement. Ce choix qui peut difficilement être optimal conduit dans notre cas à un comportement divergent. Le gros avantage de la dynamique sous contraintes est donc de supprimer ce type de problème.
Le dernier essai consiste à montrer l'influence du coefficient d'amortissement et de montrer que le passage du choc au glissement du point sur le cercle est correctement géré.

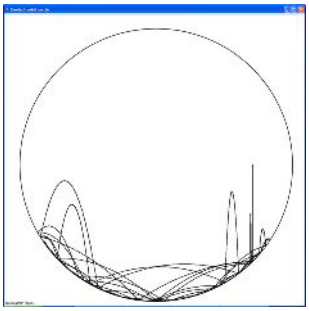
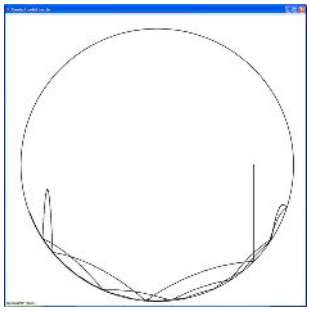
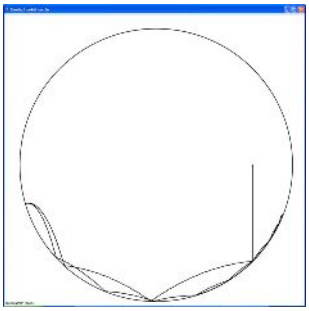
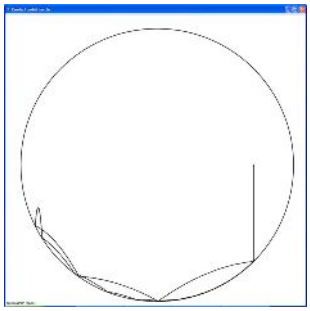
0.95		0.9	
0.85		0.8	

Tableau 6 : Evolution des chocs et glissements en fonction du coefficient d'absorption

Toutes les simulations ci-dessus ont été réalisées en 5 000 cycles ce qui nous permet de passer en glissement dès que le coefficient d'absorption devient inférieur à 0.9. Une cinquième simulation a déjà été présentée dans le cas du choc parfait. Dans les deux dernières figures le point entre en glissement et il n'est plus possible de le détecter à l'écran ce qui prouve que celui-ci reste bien sur le cercle.
Ces quelques tests bien que logiquement plus précis puisque calculés littéralement, nous ont assurés de la validité de la formulation d'Udwadia et de Kalaba pour des contraintes plus complexes que celles utilisées dans les modèles géométriques discrets.

2.2 Les contraintes dans l'esquisse virtuelle

L'utilisation d'une dynamique sous contraintes impose une définition claire des différents types de contraintes utilisées dans l'outil d'esquisse virtuelle. Dans l'état actuel de ces recherches, quatre types de contraintes sont utilisés :

- les contraintes de glissement
- les contraintes de choc
- les contraintes géométriques
- les contraintes de liaison

Nous allons essayer de présenter le mode d'obtention des contraintes dans notre système de manière générale pour présenter ensuite les contraintes de glissement, de choc, géométriques et de liaison.

2.2.1 Obtention des contraintes

Dans notre dynamique, les contraintes s'appliquent dans l'espace des accélérations et non dans l'espace des positions. On peut écrire une contrainte générale sous la forme :

$$f(p_{1,}p_{2,}..,\vec{V}_{1,}\vec{V}_{2,}...,[M_1],[M_2],...)=0 \tag{2.57}$$

ou

$$f(p_i,\vec{V}_j,[M_k])=0 \tag{2.58}$$

Les p_i sont des paramètres dépendants ou non du temps. Par exemple le module de restitution pour les chocs.
Les $\vec{V}_j$ sont des vecteurs qui dépendent ou non du temps. Par exemple le vecteur $\overrightarrow{OO_1}$ qui définit la position de la pièce 1 par rapport au repère fixe.
Les $[M_k]$ sont des matrices dépendantes ou non du temps. Par exemple la matrice $[R_{O_1}]$ correspond à la matrice de rotation de la pièce 1 par rapport au repère fixe.
Si cette équation possède un élément dans l'espace des positions, il faut dériver deux fois celle-ci par rapport au temps. Si elle ne possède pas d'élément dans l'espace des positions mais au moins un élément dans l'espace des vitesses, il faut la dériver une fois par rapport au temps.
Cas classique d'élément dans l'espace des positions :

- l'orientation des pièces définie par la matrice $[R_{O_i}]$ qui s'utilise le plus souvent avec une multiplication de vecteur :

  $$\vec{A}=[R_{O_i}]\vec{B} \tag{2.59}$$

  Dans ce cas particulier, la première dérivation donne :

  $$\frac{d\vec{A}}{dt}=[R_{O_i}]\frac{d\vec{B}}{dt}+\vec{\omega}_i \Lambda [R_{O_i}]\vec{B} \tag{2.60}$$

  si $\vec{B}$ dépend du temps sinon $\frac{d\vec{A}}{dt}=\vec{\omega}_i \Lambda [R_{O_i}]\vec{B}$.

  La seconde dérivation conduit à la relation suivante :

  $$\frac{d^2\vec{A}}{dt^2}=[R_{O_i}]\frac{d^2\vec{B}}{dt^2}+2\vec{\omega}_i \Lambda [R_{O_i}]\frac{d\vec{B}}{dt}+\vec{\omega}_i \Lambda [R_{O_i}]\vec{B}+\vec{\omega}_i \Lambda (\vec{\omega}_i \Lambda [R_{O_i}]\vec{B}) \tag{2.61}$$

- la position d'un point d'une pièce i situé sur une courbe :

  $$\overrightarrow{OP}=\overrightarrow{OO_i}+[R_{O_i}]\overrightarrow{O_iP(u_i)} \tag{2.62}$$

  $\overrightarrow{O_iP}$ Représente l'équation de la courbe dans le repère de la pièce i. Celle-ci dépend d'un paramètre u_i qui dépend lui-même du temps. Ce qui conduit à la première dérivée :

  $$\frac{d\overrightarrow{OP}}{dt}=\vec{V}_{iO_i}+[R_{O_i}]\frac{d\overrightarrow{O_iP}}{dt}+\vec{\omega}_i \Lambda [R_{O_i}]\overrightarrow{O_iP} \tag{2.63}$$

  Et en dérivant une deuxième fois :

$$\frac{d^2\overrightarrow{OP}}{dt^2}=\overrightarrow{\gamma_{iO_i}}+[R_{O_i}]\frac{d^2\overrightarrow{O_iP}}{dt^2}+2\vec{\omega}_i\Lambda[R_{O_i}]\frac{d\overrightarrow{O_iP}}{dt}+\vec{\omega}_i\Lambda[R_{O_i}]\overrightarrow{O_iP}+\vec{\omega}_i\Lambda(\vec{\omega}_i\Lambda[R_{O_i}]\overrightarrow{O_iP}) \quad (2.64)$$

Les dérivées de $\overrightarrow{O_iP}$ par rapport au temps peuvent être exprimées ainsi :

$$\frac{d\overrightarrow{O_iP(u_i)}}{dt}=\frac{du_i}{dt}\frac{d\overrightarrow{O_iP(u_i)}}{du_i}=\frac{du_i}{dt}\vec{t}_{u_i} \quad (2.65)$$

Et

$$\frac{d^2\overrightarrow{O_iP(u_i)}}{dt^2}=\frac{d^2u_i}{dt^2}\frac{d\overrightarrow{O_iP(u_i)}}{du_i}+\left(\frac{du_i}{dt}\right)^2\frac{d^2\overrightarrow{O_iP(u_i)}}{d^2u_i} \quad (2.66)$$

Ces dérivations sont réutilisées pour obtenir les contraintes dans l'espace des accélérations.
Les contraintes utilisées peuvent donc être très variées mais il faut cependant que nous obtenions dans l'espace des accélérations, une équation qui soit une combinaison linéaire des accélérations :

$$a_0\gamma_{x1}+a_1\gamma_{y1}+a_2\gamma_{z1}+a_3\dot{\omega}_{x1}+a_4\dot{\omega}_{y1}+a_5\dot{\omega}_{z1}+a_6\gamma_{x2}+a_7\gamma_{y2}+....=b \quad (2.67)$$

Cette limitation étant une conséquence de l'utilisation d'une méthode matricielle (Equ. 2.10) dans la dynamique sous contraintes. Nous allons voir que les contraintes de glissement et de choc donnent des contraintes respectant cette condition.

2.2.2 Géométries des contacts

Avant de définir les contraintes de glissement et de choc qui ont lieu au niveau du point, de la courbe ou de la surface de contact. Les géométries utilisées dans notre outil d'esquisse virtuelle sont des géométries simples mais nous verrons qu'une extension à des surfaces plus complexes est possible.

2.2.2.1 Différents types de géométries

La recherche de contact entre entités impose de prévoir tous les cas de contact possible et donc toutes les géométries correspondantes. Par exemple si l'on lance un segment sur un plan, celui-ci va rebondir sur une de ses extrémités. Il est donc indispensable d'ajouter aux extrémités des points et d'effectuer une recherche de contact entre ces points et le plan. On aboutit ainsi à la liste suivante de géométries (le paramétrage des géométries est donné en Annexe 1) :

- le point
- les courbes (le segment, le cercle)
- les surfaces (le plan, le cylindre, la sphère, le cône)

Afin de limiter des zones sur les surfaces, on peut utiliser des géométries dans le plan des paramètres u et v de celles-ci. Ces géométries ne sont que des courbes simples : le segment et le cercle

2.2.2.2 *Différents types de contact*

En effectuant toutes les combinaisons possibles de géométries [BA11] on obtient tous les différents types de contact possibles dans le tableau ci-dessous :

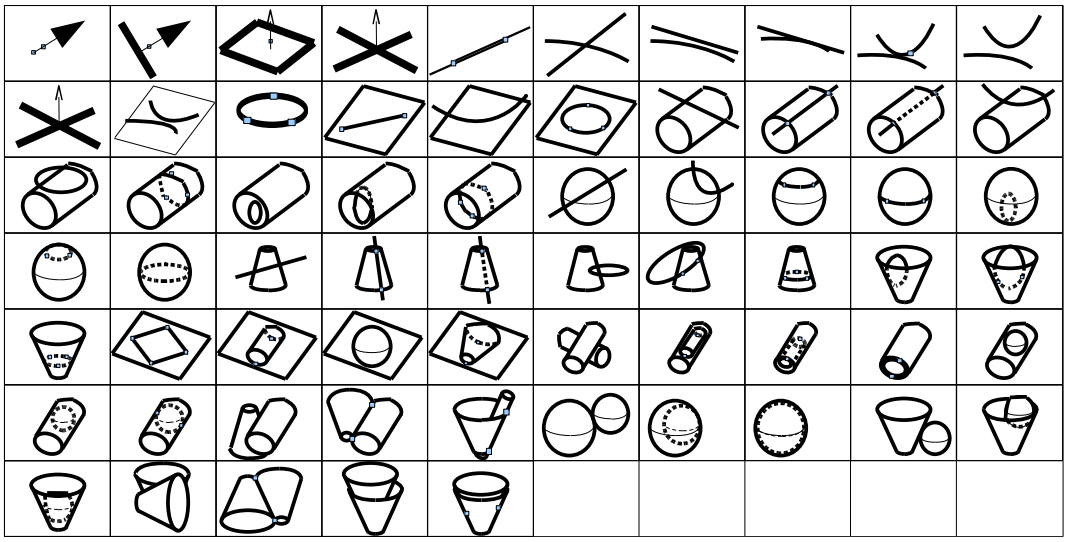


Tableau 7 : Différents types de contact

On peut en conclure qu'il existe une très grande variété de contacts, en fait une synthèse de ce tableau fait apparaître une classification par le nombre minimum de points de contact nécessaire pour définir le type de contact. Le nombre maximum de points utiles pour cette identification est de trois. Nous donnerons dans une prochaine section le nombre de points de chaque contact.
Dans cet ensemble de contact, certains sont inutiles, par exemple le segment sur un plan ne sera jamais traité puisqu'il est équivalent à deux contacts du type point sur plan. La section suivante fait l'inventaire des contacts inutiles.

2.2.2.3 *Contacts inutiles*

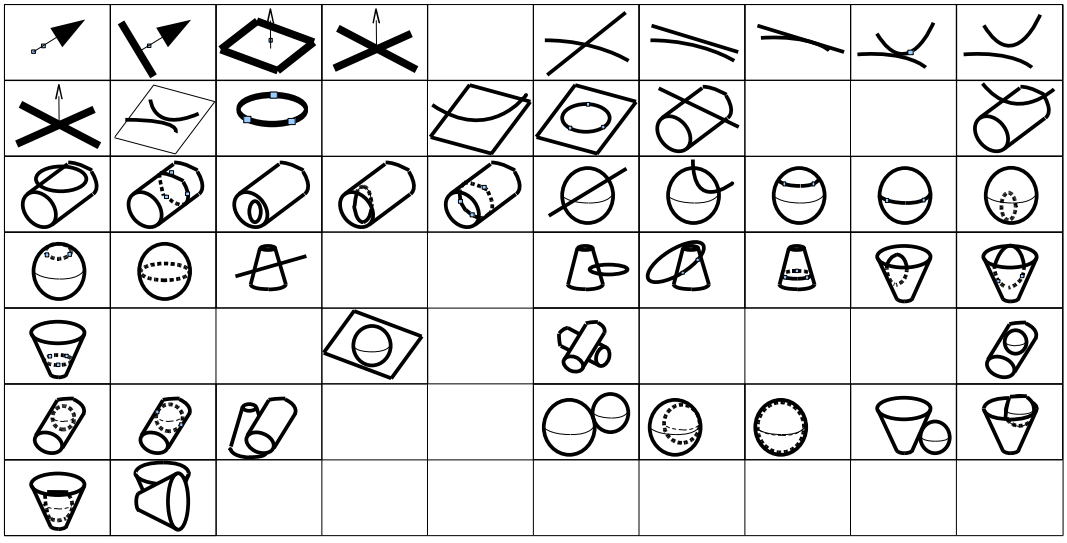


Tableau 8 : Différents types de contact utiles

Tous les contacts inutiles ont été supprimés dans le Tableau 8. La réduction du nombre de cas simplifie l'implémentation ultérieure. Nous pouvons maintenant définir le nombre de points de contact pour chaque cas.

2.2.2.4 Nombre de points de contact

1	1	1	1		1	1	2	1	1
2	1	3		1	2	1			1
2	3	1	2	3	1	1	2	2	1
2	2	1			1	2	3	1	2
3			1		1				1
1	2	1			1	1	2	1	1
2	1								

Tableau 9 : Nombre de points de contact

Le nombre de points de contact entre deux géométries sera utilisé dans la phase de détection de collision pour déterminer le type de contact.

2.2.3 Contraintes de glissement

La formulation des contraintes de glissement dans l'espace des positions est simple puisqu'il s'agit de vérifier que le point de contact est le même pour la pièce 1 et pour la pièce 2. Il existe cependant quelques difficultés dans la projection de cette égalité vectorielle en fonction du type de géométrie en contact et aussi pour l'écriture des conditions de glissement en rotation. Par exemple, un cercle qui roule sur un plan doit avoir une tangente au point de contact qui reste dans le plan.

Nous allons séparer les contraintes de glissement en deux familles, une pour la translation et l'autre pour la rotation.

2.2.3.1 Contraintes de glissement en translation

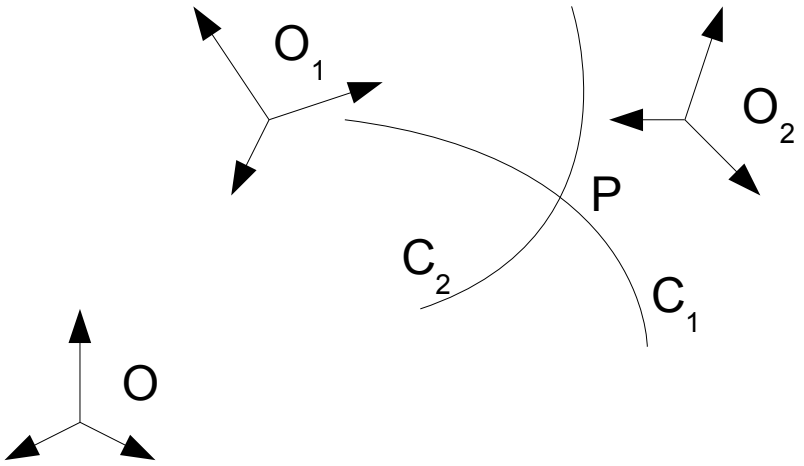


Figure 2.2 : Glissement en translation

La contrainte de glissement s'écrit :

$$\overrightarrow{OP}=\overrightarrow{OO_1}+[R_{O_1}]\overrightarrow{O_1P}=\overrightarrow{OO_2}+[R_{O_2}]\overrightarrow{O_2P} \tag{2.68}$$

D'où en dérivant une première fois :

$$\begin{gathered}\overrightarrow{v_P}=\overrightarrow{v_{1O_1}}+\overrightarrow{\omega_1}\Lambda[R_{O_1}]\overrightarrow{O_1P}+[R_{O_1}]\frac{d\overrightarrow{O_1P}}{dt}= \\ \overrightarrow{v_{2O_2}}+\overrightarrow{\omega_2}\Lambda[R_{O_2}]\overrightarrow{O_2P}+[R_{O_2}]\frac{d\overrightarrow{O_2P}}{dt}\end{gathered} \tag{2.69}$$

Puis une seconde fois :

$$\begin{gathered}\overrightarrow{\gamma_P}=\overrightarrow{\gamma_{1O_1}}+\dot{\overrightarrow{\omega}}_1\Lambda[R_{O_1}]\overrightarrow{O_1P}+\overrightarrow{\omega_1}\Lambda(\overrightarrow{\omega_1}\Lambda[R_{O_1}]\overrightarrow{O_1P})+ \\ [R_{O_1}]\frac{d^2\overrightarrow{O_1P}}{dt^2}+2\overrightarrow{\omega_1}\Lambda[R_{O_1}]\frac{d\overrightarrow{O_1P}}{dt}= \\ \overrightarrow{\gamma_{2O_2}}+\dot{\overrightarrow{\omega}}_2\Lambda[R_{O_2}]\overrightarrow{O_2P}+\overrightarrow{\omega_2}\Lambda(\overrightarrow{\omega_2}\Lambda[R_{O_2}]\overrightarrow{O_2P})+ \\ [R_{O_2}]\frac{d^2\overrightarrow{O_2P}}{dt^2}+2\overrightarrow{\omega_2}\Lambda[R_{O_2}]\frac{d\overrightarrow{O_2P}}{dt}\end{gathered} \tag{2.70}$$

La contrainte de glissement en translation est finalement :

$$\begin{gathered}\overrightarrow{\gamma_{1O_1}}+\dot{\overrightarrow{\omega}}_1\Lambda[R_{O_1}]\overrightarrow{O_1P}-\overrightarrow{\gamma_{2O_2}}-\dot{\overrightarrow{\omega}}_2\Lambda[R_{O_2}]\overrightarrow{O_2P}= \\ -\overrightarrow{\omega_1}\Lambda(\overrightarrow{\omega_1}\Lambda[R_{O_1}]\overrightarrow{O_1P})-[R_{O_1}]\frac{d^2\overrightarrow{O_1P}}{dt^2}-2\overrightarrow{\omega_1}\Lambda[R_{O_1}]\frac{d\overrightarrow{O_1P}}{dt} \\ +\overrightarrow{\omega_2}\Lambda(\overrightarrow{\omega_2}\Lambda[R_{O_2}]\overrightarrow{O_2P})+[R_{O_2}]\frac{d^2\overrightarrow{O_2P}}{dt^2}+2\overrightarrow{\omega_2}\Lambda[R_{O_2}]\frac{d\overrightarrow{O_2P}}{dt}\end{gathered} \tag{2.71}$$

Cette contrainte valable pour tous les types de géométries doit maintenant être projetée. Par exemple dans l'exemple du dessin 2.2 , nous devons projeter l'équation 2.71 sur la normale définie par le produit vectoriel des deux tangentes au point P.

Trois cas sont à envisager pour une seule contrainte de glissement :

- 3 équations : 3 projections sur x, y, z, c'est le cas très particulier du glissement d'un point sur un autre point.
- 2 équations : 2 projections dans un plan, ce cas est possible quand un point

glisse sur une courbe, le plan de projection est perpendiculaire à la courbe.

– 1 équation : 1 projection suivant une normale, c'est le cas du dessin 2.2.

Si l'on développe ce dernier cas, à partir de l'équation 2.71 multipliée par la normale, on obtient :

$$\begin{aligned}
&(\overrightarrow{\gamma_{1O_1}}+\overrightarrow{\omega_1}\Lambda[R_{O_1}]\overrightarrow{O_1P}-\overrightarrow{\gamma_{2O_2}}-\overrightarrow{\omega_2}\Lambda[R_{O_2}]\overrightarrow{O_2P})\vec{n}=\\
&(-\overrightarrow{\omega_1}\Lambda(\overrightarrow{\omega_1}\Lambda[R_{O_1}]\overrightarrow{O_1P})+\overrightarrow{\omega_2}\Lambda(\overrightarrow{\omega_2}\Lambda[R_{O_2}]\overrightarrow{O_2P}))\vec{n}\\
&(-[R_{O_1}](\frac{d^2u_i}{dt^2}\frac{d\overrightarrow{O_1P(u_i)}}{du_i}+(\frac{du_i}{dt})^2\frac{d^2\overrightarrow{O_1P(u_i)}}{d^2u_i}))\vec{n}\\
&+([R_{O_2}](\frac{d^2u_i}{dt^2}\frac{d\overrightarrow{O_2P(u_i)}}{du_i}+(\frac{du_i}{dt})^2\frac{d^2\overrightarrow{O_2P(u_i)}}{d^2u_i}))\vec{n}\\
&+(-2\overrightarrow{\omega_1}\Lambda[R_{O_1}](\frac{du_i}{dt}\frac{d\overrightarrow{O_1P(u_i)}}{du_i})+2\overrightarrow{\omega_2}\Lambda[R_{O_2}](\frac{du_i}{dt}\frac{d\overrightarrow{O_2P(u_i)}}{du_i}))\vec{n}
\end{aligned} \tag{2.72}$$

comme,

$$\vec{n}=([R_{O_1}]\frac{d\overrightarrow{O_1P}}{du_1})\Lambda([R_{O_2}]\frac{d\overrightarrow{O_2P}}{du_2}) \tag{2.73}$$

Les termes de la partie droite en $[R_{O_1}]\frac{d\overrightarrow{O_1P}}{du_1}$ ou en $[R_{O_2}]\frac{d\overrightarrow{O_2P}}{du_2}$ disparaissent, on obtient finalement :

$$\begin{aligned}
&(\overrightarrow{\gamma_{1O_1}}+\overrightarrow{\omega_1}\Lambda[R_{O_1}]\overrightarrow{O_1P}-\overrightarrow{\gamma_{2O_2}}-\overrightarrow{\omega_2}\Lambda[R_{O_2}]\overrightarrow{O_2P})\vec{n}=\\
&(-\overrightarrow{\omega_1}\Lambda(\overrightarrow{\omega_1}\Lambda[R_{O_1}]\overrightarrow{O_1P})+\overrightarrow{\omega_2}\Lambda(\overrightarrow{\omega_2}\Lambda[R_{O_2}]\overrightarrow{O_2P}))\vec{n}\\
&(-[R_{O_1}]((\frac{du_i}{dt})^2\frac{d^2\overrightarrow{O_1P(u_i)}}{d^2u_i})+[R_{O_2}]((\frac{du_i}{dt})^2\frac{d^2\overrightarrow{O_2P(u_i)}}{d^2u_i}))\vec{n}\\
&+(-2\overrightarrow{\omega_1}\Lambda[R_{O_1}](\frac{du_i}{dt}\frac{d\overrightarrow{O_1P(u_i)}}{du_i})+2\overrightarrow{\omega_2}\Lambda[R_{O_2}](\frac{du_i}{dt}\frac{d\overrightarrow{O_2P(u_i)}}{du_i}))\vec{n}
\end{aligned} \tag{2.74}$$

Les termes en $\frac{d^2u_1}{dt^2}$ et $\frac{d^2u_2}{dt^2}$ ayant disparu, il ne reste plus qu'à déterminer les termes $\frac{du_1}{dt}$ et $\frac{du_2}{dt}$.

Cette détermination peut s'effectuer de différentes manières, une première technique a été testée qui consiste à stocker les valeurs des u_1 et des u_2 au cours du temps. Cette technique pose des problèmes lors de la phase d'entrée en glissement, en effet, il faut utiliser les valeurs des paramètres u_i lors des derniers chocs ce qui manque de précision. Nous allons voir qu'une autre solution nous permet d'obtenir directement les termes $\frac{du_i}{dt}$.

A partir de l'équation 2.69, on peut écrire :

$$\overrightarrow{v_{1O_1}} - \overrightarrow{v_{2O_2}} + \overrightarrow{\omega_1} \Lambda [R_{O_1}] \overrightarrow{O_1 P} - \overrightarrow{\omega_2} \Lambda [R_{O_2}] \overrightarrow{O_2 P} = -[R_{O_1}] \frac{d \overrightarrow{O_1 P}}{dt} + [R_{O_2}] \frac{d \overrightarrow{O_2 P}}{dt} \quad (2.75)$$

On peut en utilisant le vecteur $\vec{n}$ précédent définir un vecteur $\vec{s_1}$:

$$\vec{s_1} = [R_{O_1}] \frac{d \overrightarrow{O_1 P}}{du_1} \Lambda \vec{n} \quad (2.76)$$

En multipliant 2.75 par $\vec{s_1}$, il reste :

$$(\overrightarrow{v_{1O_1}} - \overrightarrow{v_{2O_2}} + \overrightarrow{\omega_1} \Lambda [R_{O_1}] \overrightarrow{O_1 P} - \overrightarrow{\omega_2} \Lambda [R_{O_2}] \overrightarrow{O_2 P}) . \vec{s_1} = [R_{O_2}] \frac{du_2}{dt} \frac{d \overrightarrow{O_2 P}}{du_2} . \vec{s_1} \quad (2.77)$$

Et finalement :

$$\frac{du_2}{dt} = \frac{(\overrightarrow{v_{1O_1}} - \overrightarrow{v_{2O_2}} + \overrightarrow{\omega_1} \Lambda [R_{O_1}] \overrightarrow{O_1 P} - \overrightarrow{\omega_2} \Lambda [R_{O_2}] \overrightarrow{O_2 P}) . \vec{s_1}}{[R_{O_2}] \frac{d \overrightarrow{O_2 P}}{du_2} . \vec{s_1}} \quad (2.78)$$

La même technique peut être utilisée pour obtenir $\frac{du_1}{dt}$.

Les termes $\frac{du_1}{dt}$ et $\frac{du_2}{dt}$ étant déterminés, on peut obtenir les contraintes de glissement en translation à l'instant t à partir de l'équation 2.74 sans stocker les valeurs précédentes des paramètres u_1 et u_2 . Ce résultat est très important, avant d'être en glissement, les objets ne sont par définition pas en contact. Il est alors impossible de connaître les paramètres u_1 et u_2 juste avant l'entrée en glissement.

2.2.3.2 *Contraintes de glissement en rotation*

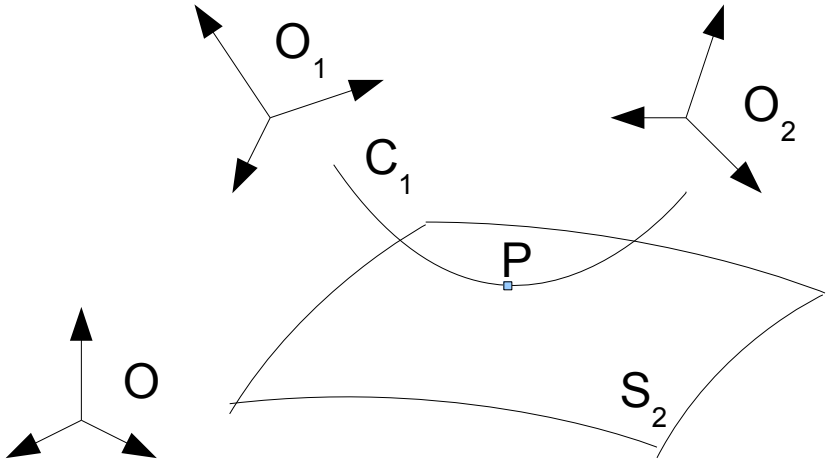


Figure 2.3 : Contrainte de glissement en rotation

La figure 2.3 illustre une contrainte de glissement en rotation entre une courbe et une surface. La condition est que la tangente à la courbe en P reste dans le plan tangent à la surface en P. Ce qui nous conduit à la contrainte suivante :

$$[R_{O_1}]\frac{d\overrightarrow{O_1P}}{du_1}.[R_{O_2}](\frac{d\overrightarrow{O_2P}}{du_2}\Lambda\frac{d\overrightarrow{O_2P}}{dv_2})=0 \quad (2.79)$$

Ou :

$$[R_{O_1}]\vec{t}_{u_1}.[R_{O_2}]\vec{n}_2=0 \quad (2.80)$$

Dans cette dernière formulation, $\vec{t}_{u_1}$ est la tangente suivant le paramètre u_1 et $\vec{n}_2$ est la normale à la surface en P.
On dérive une première fois cette équation :

$$(\vec{\omega}_1\Lambda[R_{O_1}]\vec{t}_{u_1}).[R_{O_2}]\vec{n}_2+[R_{O_1}]\vec{t}_{u_1}.(\vec{\omega}_2\Lambda[R_{O_2}]\vec{n}_2)=0 \quad (2.81)$$

Après la deuxième dérivation la contrainte prend la forme :

$$\begin{gathered}(\vec{\omega}_1\Lambda[R_{O_1}]\vec{t}_{u_1}).[R_{O_2}]\vec{n}_2+[R_{O_1}]\vec{t}_{u_1}.(\vec{\omega}_2\Lambda[R_{O_2}]\vec{n}_2)= \\ -2(\vec{\omega}_1\Lambda[R_{O_1}]\vec{t}_{u_1}).(\vec{\omega}_2\Lambda[R_{O_2}]\vec{n}_2)-(\vec{\omega}_1\Lambda(\vec{\omega}_1\Lambda[R_{O_1}]\vec{t}_{u_1})).[R_{O_2}]\vec{n}_2 \\ -[R_{O_1}]\vec{t}_{u_1}.(\vec{\omega}_2\Lambda(\vec{\omega}_2\Lambda[R_{O_2}]\vec{n}_2))\end{gathered} \quad (2.82)$$

Lors du contact surface/surface, deux équations de ce type doivent être utilisées, une avec $\vec{t}_{u_1}$ et l'autre avec $\vec{t}_{u_2}$.

2.2.4 Contraintes de choc

Comme pour les contraintes de glissement, les contraintes de choc sont aussi décomposées en deux types :

- contrainte de choc en translation
- contrainte de choc en rotation

2.2.4.1 Contraintes de choc en translation

La contrainte de choc en translation définit la réduction de la vitesse après le choc d'après la loi suivante :

$$(\vec{v}_{2P}^{+}-\vec{v}_{1P}^{+}).\vec{n}=-e((\vec{v}_{2P}^{-}-\vec{v}_{1P}^{-}).\vec{n}) \quad (2.83)$$

Dans cette équation, $\vec{v}_{iP}^{+}$ représente la vitesse après le choc du point P appartenant à la pièce i, $\vec{v}_{iP}^{-}$ est la même vitesse mais avant le choc. e est le coefficient d'absorption. Le vecteur $\vec{n}$ n'est pas toujours une normale au point de contact, par exemple même si ce choc est assez improbable, dans le cas du contact entre deux points, la notion de normale n'a plus de sens. Il faut alors utiliser le vecteur passant par les deux points juste avant le choc.
Comme pour les contraintes de glissement, nous aboutissons à 3 cas en fonction des géométries en contact. Il faut définir comment $\vec{n}$ dans chacun de ces cas :

- Contact point/point : comme nous l'avons dit précédemment $\vec{n}$ est le vecteur définit par les deux points juste avant le contact.
- Contact point/courbe : $\vec{n}$ est défini par la normale à la courbe qui passe par le point juste avant le contact.

– Tous les autres types de contact utilisent $\vec{n}$ comme la normale au point de contact.

L'équation 2.83 peut être exprimée dans l'espace des accélérations sans dérivation en effet :

$$\begin{aligned}\overrightarrow{v_{1P}}^{+}&=\overrightarrow{v_{1O_1}}^{+}+\overrightarrow{\omega_1}^{+}\Lambda[R_{O_1}]\overrightarrow{O_1P}\quad et\quad \overrightarrow{v_{2P}}^{+}=\overrightarrow{v_{2O_2}}^{+}+\overrightarrow{\omega_2}^{+}\Lambda[R_{O_2}]\overrightarrow{O_2P}\\ \overrightarrow{v_{1P}}^{-}&=\overrightarrow{v_{1O_1}}^{-}+\overrightarrow{\omega_1}^{-}\Lambda[R_{O_1}]\overrightarrow{O_1P}\quad et\quad \overrightarrow{v_{2P}}^{-}=\overrightarrow{v_{2O_2}}^{-}+\overrightarrow{\omega_2}^{-}\Lambda[R_{O_2}]\overrightarrow{O_2P}\end{aligned} \tag{2.84}$$

Et

$$\overrightarrow{\gamma_{1O_1}}=\frac{\overrightarrow{v_{1O_1}}^{+}-\overrightarrow{v_{1O_1}}^{-}}{dt}\quad \overrightarrow{\gamma_{2O_2}}=\frac{\overrightarrow{v_{2O_2}}^{+}-\overrightarrow{v_{2O_2}}^{-}}{dt}\quad \vec{\omega}_1=\frac{\overrightarrow{\omega_1}^{+}-\overrightarrow{\omega_1}^{-}}{dt}\quad \vec{\omega}_2=\frac{\overrightarrow{\omega_2}^{+}-\overrightarrow{\omega_2}^{-}}{dt} \tag{2.85}$$

D'où :

$$\begin{aligned}&(\overrightarrow{\gamma_{2O_2}}+\vec{\omega}_2\Lambda[R_{O_2}]\overrightarrow{O_2P}-\overrightarrow{\gamma_{1O_1}}-\vec{\omega}_1\Lambda[R_{O_1}]\overrightarrow{O_1P}).\vec{n}=\\ &-\frac{e+1}{dt}((\overrightarrow{v_{2O_2}}^{-}+\overrightarrow{\omega_2}^{-}\Lambda[R_{O_2}]\overrightarrow{O_2P}-\overrightarrow{v_{1O_1}}^{-}-\overrightarrow{\omega_1}^{-}\Lambda[R_{O_1}]\overrightarrow{O_1P}).\vec{n})\end{aligned} \tag{2.86}$$

On peut remarquer que la partie gauche de cette équation est très semblable à celle obtenue pour la contrainte de glissement.

2.2.4.2 *Contraintes de choc en rotation*

Les contraintes de choc en rotation sont peu courantes, elles apparaissent par exemple pour un segment tombant sur un plan. Cette contrainte est de la forme :

$$(\overrightarrow{\omega_2}^{+}-\overrightarrow{\omega_1}^{+}).([R_{O_1}]\vec{y}_1\Lambda[R_{O_2}]\vec{n}_2)=-e((\overrightarrow{\omega_2}^{-}-\overrightarrow{\omega_1}^{-}).([R_{O_1}]\vec{y}_1\Lambda[R_{O_2}]\vec{n}_2)) \tag{2.87}$$

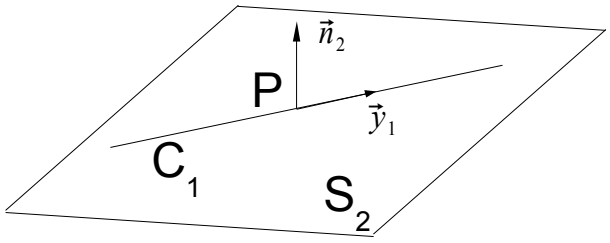


Figure 2.4 : Contrainte de choc en rotation

La projection sur l'axe $(\vec{y}_1\Lambda\vec{n}_2)$ correspond au fait que lors du choc, le segment ne peut tourner autour de cet axe. Le coefficient d'absorption est le même que pour le choc en translation. L'obtention de l'équation est elle aussi identique, d'où :

$$(\vec{\omega}_2-\vec{\omega}_1).([R_{O_1}]\vec{y}_1\Lambda[R_{O_2}]\vec{n}_2)=-\frac{e+1}{dt}((\overrightarrow{\omega_2}^{-}-\overrightarrow{\omega_1}^{-}).([R_{O_1}]\vec{y}_1\Lambda[R_{O_2}]\vec{n}_2)) \tag{2.88}$$

2.2.5 Contraintes géométriques

Les contraintes géométriques interviennent en phase de modification d'un mécanisme. En effet, pour modifier une entité dans l'outil d'esquisse virtuelle, on considère que cette entité est une nouvelle pièce avec ses caractéristiques

inertielles propres. Cette nouvelle pièce a son mouvement contraint par des relations géométriques du type : parallélisme, passe par, distant de, etc. Nous allons décrire comment nous obtenons quelques unes de ces contraintes et montrer que ces contraintes sont équivalentes à des contraintes de glissement. Ce résultat très intéressant va nous permettre de réaliser un outil d'esquisse dans lequel sont unifiées les contraintes géométriques et les contraintes liées à la dynamique.

2.2.5.1 Contraintes géométriques de distance

Cette contrainte permet de maintenir une certaine distance entre deux géométries, par exemple la distance entre deux points. Cette contrainte est équivalente à une contrainte de glissement maintenant un des deux points sur une sphère centrée sur le deuxième point avec la distance recherchée comme rayon.

Il existe quelques différences entre ces deux cas, en effet quand nous cherchons à maintenir un point à une certaine distance d'un autre, l'effort qui va nous obliger à rester à cette distance est bidirectionnelle. Dans les contraintes de glissement, nous considérons que les pièces ont de la matière. Cette modification est très simple à prendre en compte. La deuxième différence est qu'il parait plus simple et plus général de considérer des géométries dont certaines dimensions sont infinies. Par exemple si nous voulons imposer une distance entre un point et un segment, nous utilisons comme équivalent, une contrainte de glissement d'un point sur un cylindre. En considérant que ce cylindre a une dimension infinie, nous maintenons la contrainte de distance même si le point n'est plus projeté sur le segment.

Exemples de conversion, contraintes géométriques de distance en contrainte de glissement :

- distance point/point : glissement point/sphère
- distance point/segment : glissement point/cylindre
- distance point/plan : glissement point/plan
- distance segment/segment : glissement segment/cylindre

On peut généraliser en donnant comme règle qu'il faut faire un offset de la géométrie de plus grande dimension. Cette généralisation n'est pas possible actuellement pour le cercle qui par offset donne un tore.

2.2.5.2 Contraintes géométriques angulaires

Cette contrainte va permettre de définir des parallélismes, des perpendicularités ou des angles quelconques entre les géométries. L'angle est défini entre deux directions caractéristiques de deux géométries. Comme pour le glissement en rotation, c'est le produit vectoriel entre les vecteurs support des deux directions qui va définir la contrainte dans l'espace des positions :

$$[R_{O_1}]\vec{v}_1 . [R_{O_2}]\vec{v}_2 = b \tag{2.89}$$

Par exemple la perpendicularité entre deux segments va s'écrire :

$$[R_{O_1}]\vec{y}_1.[R_{O_2}]\vec{y}_2=0 \quad (2.90)$$

Et le parallélisme entre deux segments :

$$[R_{O_1}]\vec{y}_1.[R_{O_2}]\vec{y}_2=1 \quad (2.91)$$

Le parallélisme d'un segment par rapport à un plan :

$$[R_{O_1}]\vec{y}_1.[R_{O_2}]\vec{n}_2=0 \quad (2.92)$$

Où $\vec{n}_2=\vec{x}_2 \Lambda \vec{y}_2$.

On peut donc remarquer qu'il y a de grosses similitudes avec la contrainte de glissement en rotation. L'existence d'une constante du côté droit de l'équation disparait à la première dérivation. On peut donc unifier les contraintes géométriques angulaires et les contraintes de glissement en rotation :

$$\begin{aligned}(\vec{\omega}_1 \Lambda [R_{O_1}]\vec{v}_1).[R_{O_2}]\vec{v}_2+[R_{O_1}]\vec{v}_1.(\vec{\omega}_2 \Lambda [R_{O_2}]\vec{v}_2)= \\ -2(\vec{\omega}_1 \Lambda [R_{O_1}]\vec{v}_1).(\vec{\omega}_2 \Lambda [R_{O_2}]\vec{v}_2)-(\vec{\omega}_1 \Lambda (\vec{\omega}_1 \Lambda [R_{O_1}]\vec{v}_1)).[R_{O_2}]\vec{v}_2 \\ -[R_{O_1}]\vec{v}_1.(\vec{\omega}_2 \Lambda (\vec{\omega}_2 \Lambda [R_{O_2}]\vec{v}_2))\end{aligned} \quad (2.93)$$

Des contraintes plus complexes seront définies par combinaison des contraintes précédentes.

2.2.6 Contraintes de type liaison

Nous allons maintenant montrer comment les liaisons entre pièces peuvent être converties en contraintes dans l'espace des accélérations. Le tableau suivant permet de lister les liaisons les plus courantes utilisées dans les mécanismes. Ce tableau classe les liaisons an fonction de deux paramètres : le nombre de degrès de liberté en rotation de 0 à 3 et le nombre de degrès de liberté en translation de 0 à 2.

Rotation Translation	0	1	2	3
0		Liaison Pivot		Liaison Rotule
1	Liaison Glissière	Liaison Pivot Glissant		Liaison Linéaire Annulaire
2		Liaison Appui Plan	Liaison Linéaire Rectiligne	Liaison Contact Ponctuel

Tableau 10 : Liaisons entre deux pièces

Par mesure de simplification, nous ne prendrons pas en compte les liaisons Linéaire Rectiligne et Contact Ponctuel. Ces liaisons peuvent être réalisées par des contraintes de glissement. Avec cette simplification, les contraintes en rotation sont très simples à exprimer, en effet une contrainte en rotation correspond à la suppression d'un degré de liberté en rotation. La difficulté réside donc uniquement dans la possibilité d'exprimer ce blocage sur les axes

correspondants. Pour chaque liaison, doit être défini un repère fixé à une des pièces définissant la liaison, nous ne montrerons ce paramétrage que dans le cas de la liaison pivot glissant d'axe $\vec{y}$ (Fig. 2.5) :

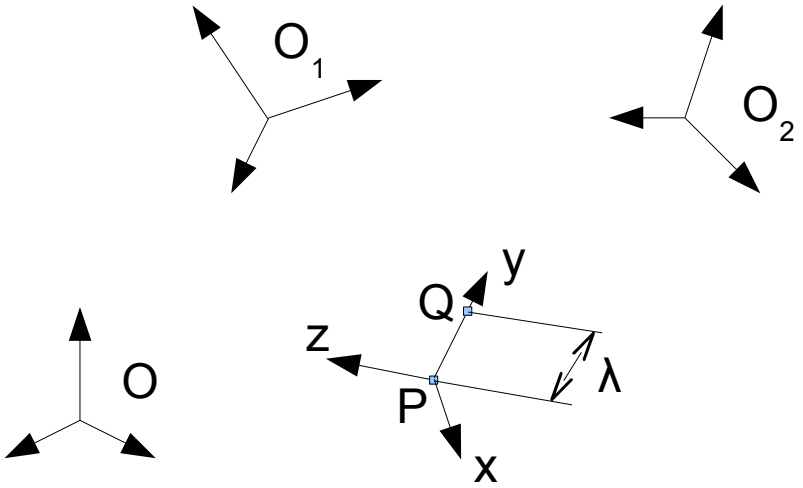


Figure 2.5 : Paramétrage d'une liaison pivot glissant

Nous définissons un repère, définit par la position $\overrightarrow{O_1P}$ et l'orientation R_l de la liaison, fixe par rapport à la pièce 1 . A $t=0$, Le point P est équivalent au point Q qui lui est fixe par rapport à la pièce 2, puis au cours du temps Q se déplace le long de la droite (P , $\vec{y}$). La distance entre P et Q est notée λ .

2.2.6.1 *Contraintes en rotation dans une liaison*

La contrainte de rotation entre la pièce 1 et la pièce 2 est déterminée par le fait que l'axe $\vec{y}$ reste identique au cours du temps dans le cas de la liaison pivot, ce qui s'écrit :

$$[R_{O_1}][R_l]\vec{y}=[R_{O_2}][R_l]\vec{y} \tag{2.94}$$

En dérivant une première fois, on obtient :

$$\overrightarrow{\omega_1}\Lambda[R_{O_1}][R_l]\vec{y}=\overrightarrow{\omega_2}\Lambda[R_{O_2}][R_l]\vec{y} \tag{2.95}$$

En dérivant une fois par rapport au temps, on obtient :

$$\begin{aligned}\dot{\overrightarrow{\omega_1}}\Lambda[R_{O_1}][R_l]\vec{y}+\overrightarrow{\omega_1}\Lambda(\overrightarrow{\omega_1}\Lambda[R_{O_1}][R_l]\vec{y})=\\ \dot{\overrightarrow{\omega_2}}\Lambda[R_{O_2}][R_l]\vec{y}+\overrightarrow{\omega_2}\Lambda(\overrightarrow{\omega_2}\Lambda[R_{O_2}][R_l]\vec{y})\end{aligned} \tag{2.96}$$

Et donc :

$$\begin{aligned}\dot{\overrightarrow{\omega_1}}\Lambda[R_{O_1}][R_l]\vec{y}-\dot{\overrightarrow{\omega_2}}\Lambda[R_{O_2}][R_l]\vec{y}=\\ -\overrightarrow{\omega_1}\Lambda(\overrightarrow{\omega_1}\Lambda[R_{O_1}][R_l]\vec{y})+\overrightarrow{\omega_2}\Lambda(\overrightarrow{\omega_2}\Lambda[R_{O_2}][R_l]\vec{y})\end{aligned} \tag{2.97}$$

En projetant sur $\vec{x}$ et sur $\vec{z}$, on obtient les deux équations de contraintes suivantes :

$$
\begin{array}{c}
(\vec{\omega_1} \Lambda [R_{O_1}][R_l]\vec{y} - \vec{\omega_2} \Lambda [R_{O_2}][R_l]\vec{y})[R_{O_1}][R_l]\vec{x} = \\
(-\vec{\omega_1} \Lambda (\vec{\omega_1} \Lambda [R_{O_1}][R_l]\vec{y}) + \vec{\omega_2} \Lambda (\vec{\omega_2} \Lambda [R_{O_2}][R_l]\vec{y}))[R_{O_1}][R_l]\vec{x} \\
(\vec{\omega_1} \Lambda [R_{O_1}][R_l]\vec{y} - \vec{\omega_2} \Lambda [R_{O_2}][R_l]\vec{y})[R_{O_1}][R_l]\vec{z} = \\
(-\vec{\omega_1} \Lambda (\vec{\omega_1} \Lambda [R_{O_1}][R_l]\vec{y}) + \vec{\omega_2} \Lambda (\vec{\omega_2} \Lambda [R_{O_2}][R_l]\vec{y}))[R_{O_1}][R_l]\vec{z}
\end{array}
\qquad (2.98)
$$

2.2.6.2 Contraintes en translation dans une liaison

Comme pour les contraintes de glissement, nous partons de l'égalité vectorielle suivante :

$$
\overrightarrow{OQ} = \overrightarrow{OO_1} + [R_{O_1}](\overrightarrow{O_1P} + [R_l]\lambda\vec{y}) = \overrightarrow{OO_2} + [R_{O_2}](\overrightarrow{O_2Q}) \qquad (2.99)
$$

En dérivant une première fois par rapport au temps, on obtient :

$$
\begin{array}{c}
\overrightarrow{V_{1O_1}} + \vec{\omega_1} \Lambda [R_{O_1}](\overrightarrow{O_1P} + [R_l]\lambda\vec{y}) + [R_{O_1}][R_l]\frac{d\lambda}{dt}\vec{y} = \\
\overrightarrow{V_{2O_2}} + \vec{\omega_2} \Lambda [R_{O_2}](\overrightarrow{O_2Q})
\end{array}
\qquad (2.100)
$$

La deuxième dérivation nous permet d'aboutir à :

$$
\begin{array}{c}
\overrightarrow{\gamma_{1O_1}} + \vec{\omega_1} \Lambda [R_{O_1}](\overrightarrow{O_1P} + [R_l]\lambda\vec{y}) - \overrightarrow{\gamma_{2O_2}} - \vec{\omega_2} \Lambda [R_{O_2}]\overrightarrow{O_2Q} = \\
-\vec{\omega_1} \Lambda (\vec{\omega_1} \Lambda [R_{O_1}](\overrightarrow{O_1P} + [R_l]\lambda\vec{y})) - [R_{O_1}][R_l]\frac{d^2\lambda}{dt^2}\vec{y} \\
-2\vec{\omega_1} \Lambda [R_{O_1}][R_l]\frac{d\lambda}{dt}\vec{y} + \vec{\omega_2} \Lambda (\vec{\omega_2} \Lambda [R_{O_2}]\overrightarrow{O_2Q})
\end{array}
\qquad (2.101)
$$

Puis en projetant sur $\vec{x}$ et sur $\vec{z}$, on obtient les deux contraintes suivantes :

$$
\begin{array}{c}
(\overrightarrow{\gamma_{1O_1}} + \vec{\omega_1} \Lambda [R_{O_1}](\overrightarrow{O_1P} + [R_l]\lambda\vec{y}) - \overrightarrow{\gamma_{2O_2}} - \vec{\omega_2} \Lambda [R_{O_2}]\overrightarrow{O_2Q})[R_{O_1}][R_l]\vec{x} = \\
(-\vec{\omega_1} \Lambda (\vec{\omega_1} \Lambda [R_{O_1}](\overrightarrow{O_1P} + [R_l]\lambda\vec{y})) - [R_{O_1}][R_l]\frac{d^2\lambda}{dt^2}\vec{y})[R_{O_1}][R_l]\vec{x} \\
+(-2\vec{\omega_1} \Lambda [R_{O_1}][R_l]\frac{d\lambda}{dt}\vec{y} + \vec{\omega_2} \Lambda (\vec{\omega_2} \Lambda [R_{O_2}]\overrightarrow{O_2Q}))[R_{O_1}][R_l]\vec{x}
\end{array}
\qquad (2.102)
$$

Et

$$
\begin{array}{c}
(\overrightarrow{\gamma_{1O_1}} + \vec{\omega_1} \Lambda [R_{O_1}](\overrightarrow{O_1P} + [R_l]\lambda\vec{y}) - \overrightarrow{\gamma_{2O_2}} - \vec{\omega_2} \Lambda [R_{O_2}]\overrightarrow{O_2Q})[R_{O_1}][R_l]\vec{z} = \\
(-\vec{\omega_1} \Lambda (\vec{\omega_1} \Lambda [R_{O_1}](\overrightarrow{O_1P} + [R_l]\lambda\vec{y})) - [R_{O_1}][R_l]\frac{d^2\lambda}{dt^2}\vec{y})[R_{O_1}][R_l]\vec{z} \\
+(-2\vec{\omega_1} \Lambda [R_{O_1}][R_l]\frac{d\lambda}{dt}\vec{y} + \vec{\omega_2} \Lambda (\vec{\omega_2} \Lambda [R_{O_2}]\overrightarrow{O_2Q}))[R_{O_1}][R_l]\vec{z}
\end{array}
\qquad (2.103)
$$

Les contraintes dans les autres liaisons peuvent facilement être déterminées de la même manière.

2.2.7 Combinaison de contraintes

On obtient une combinaison de contraintes dès qu'il existe plusieurs contacts entre deux pièces simultanément. L'utilisation de la pseudo-inverse permet de gérer automatiquement les cas de multiples glissements, par contre la gestion des chocs multiples ou des combinaisons glissements avec choc ne peut être réalisée simplement. L'exemple (Fig. 2.6) suivant permet d'expliquer l'incompatibilité entre glissement et choc :

Figure 2.6 : Glissement avec choc

La contrainte de glissement impose que le point reste horizontal mais quand le point entre en choc avec le segment incliné, la vitesse après choc devrait se situer en dessous de l'axe horizontal. Si l'on applique les deux contraintes en même temps, le résultat du mouvement est totalement incohérent. Pour trouver une solution à ce problème, il faut revenir à la physique du problème pour comprendre que la contrainte de glissement se transforme en contrainte de choc lors de l'impact (phénomène que l'on retrouve dans la théorie du billard). On peut vérifier la validité de ce choix sur le cas simple précédent, en écrivant les équations de contraintes de choc à partir du schéma suivant (Fig. 2.7) au moment du choc :

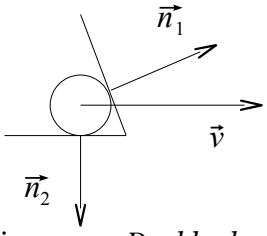


Figure 2.7 : Double choc

On peut écrire les vecteurs $\vec{n}_1$, $\vec{n}_2$ et $\vec{v}$ sous la forme simplifiée :

$$\vec{n}_1 = \begin{bmatrix} \cos(\alpha) \\ \sin(\alpha) \end{bmatrix} \tag{2.104}$$

$$\vec{n}_2 = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \tag{2.105}$$

$$\vec{v} = \begin{bmatrix} v \\ 0 \end{bmatrix} \tag{2.106}$$

En utilisant simplifiée de l'équation 2.86 :

$$\vec{\gamma}.\vec{n} = -\frac{e+1}{dt}(\vec{v}.\vec{n}) \tag{2.107}$$

On obtient les deux équations suivantes :

$$\begin{gathered} \gamma_x.\cos(\alpha) + \gamma_y.\sin(\alpha) = -\frac{e+1}{dt}.\cos(\alpha).v \\ -\gamma_y = 0 \end{gathered} \tag{2.108}$$

Ce qui conduit à :

$$\begin{gathered} \gamma_x = -\frac{e+1}{dt}.v \\ \gamma_x = 0 \end{gathered} \tag{2.109}$$

Ce résultat est cohérent au détail près qu'un seul coefficient d'absorption n'apparait ce qui signifie que quelque soit l'absorption avec l'axe horizontal, le point reviendra à la même vitesse, horizontalement vers la gauche. Par contre, il est tout à fait satisfaisant que l'angle n'intervienne pas dans ce résultat.
Un autre exemple (Fig. 2.8) similaire à celui de la figure 2.6 mais avec un angle d'ouverture plus important montre que dans certains cas, la contrainte de glissement doit simplement disparaître :

Figure 2.8 : Choc avec perte de glissement

Dans ce cas quand le point va entrer en choc, il va simplement rebondir sur le segment incliné et perdre le contact avec le segment horizontal. Nous avons indiqué au paragraphe 2.1.3.1 comment retrouver l'effort au point de contact ce qui va nous permettre de déterminer si un glissement est remplacé par un choc ou s'il est simplement supprimé.

2.3 Conclusion sur la dynamique

Nous avons décrit comment un lien pouvait être établi entre la formulation de la dynamique sous contraintes d'Udwadia et de Kalaba et les contraintes de choc, de glissement, géométriques et de liaison. Ce lien ouvre de nouvelles possibilités puisque tout est dynamique et contrainte, il n'y a plus aucune difficulté pour imposer des contraintes supplémentaires. Nous montrerons dans les exemples d'utilisation comment nous manipulerons ce lien pour augmenter par exemple la distance entre deux liaisons.

Nous pouvons donc trouver les accélérations contraintes à tout instant, il nous faut maintenant déterminer quand apparait un nouveau choc, une perte de contact etc. Nous regroupons toutes ces modification sous le terme gestion des changements d'états que nous détaillerons dans le chapitre 3 .

3 Gestion des changements d'états

La dynamique sous contraintes nous a permis de déterminer les trajectoires des pièces, nous allons montrer dans ce chapitre comment à partir de la connaissance de ces mouvements, nous allons déterminer les changements d'état : nouveau contact, perte de contact, changement de type de contact (de choc en glissement par exemple).

3.1 Détection de changement d'état

La détection de changement d'état ne se limite pas à la détection de collision, il est en effet tout aussi important de détecter à quel moment deux géométries ne se touchent plus que de savoir quand elles rentrent en contact.

Dans la dynamique sous contraintes, nous pouvons obtenir les efforts dans le contact après application des contraintes à un instant t . On ne peut pas inversement savoir à quel instant t , un effort peut prendre une valeur précise. La perte d'un glissement est donc déterminée après application des contraintes quand on vérifie que l'effort dans le contact a changé de signe. Ceci n'est pas très gênant car la perte de glissement ne se fait pas brusquement.

Il existe donc 3 types de changement d'état pour un point de contact (Tableau 11):

- Nouveau point de contact, détecté en partant de $t=0$ vers $t=dt$ (Fig. 3.1).
- Perte de contact géométrique, détectée en partant de $t=dt$ vers $t=0$ (Fig. 3.2).
- Perte de contact par décollement (Fig. 3.3).

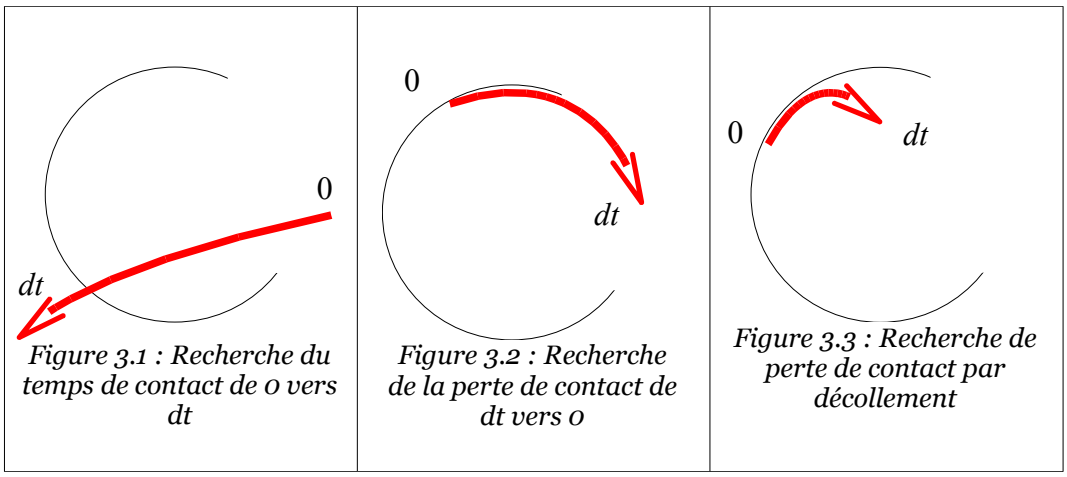


Figure 3.1 : Recherche du temps de contact de 0 vers dt

Figure 3.2 : Recherche de la perte de contact de dt vers 0

Figure 3.3 : Recherche de perte de contact par décollement

Tableau 11 : Les trois types de changement d'état pour un point de contact

Nous ne traiterons dans ce chapitre que les deux premiers changements d'états, le troisième, ne posant pas de problème, est traité complètement à part dans notre programme.

Nous avons vu dans la définition des géométries que 3 points au maximum permettaient de déterminer le type de contact, il existe donc plus de 3 types de changement d'état pour un contact. Par exemple, dans le cas d'un contact entre un segment et un cercle, on peut avoir deux points de contact avec les changements d'états suivants :

- Deux nouveaux points de contact, ceci peut apparaître si l'on a deux chocs entre $t=0$ et $t=dt$.
- Deux pertes de contact géométrique, le cercle peut aussi être un arc de cercle et dans ce cas, il est possible que le segment en glissement sur cet arc, perde le contact aux deux extrémités dans l'intervalle de temps $(0,dt)$.
- Un nouveau point de contact et une perte de contact géométrique quasi-simultanément dans l'intervalle de temps $(0,dt)$.

La gestion du changement d'état implique premièrement de déterminer à quel instant il y a changement de configuration (nouveau contact, perte de contact). Ensuite, on laisse tous les objets se déplacer jusqu'à l'instant du changement et finalement, on cherche les changements d'état à cet instant. Cette solution nous permet de gérer tous les cas possibles de prise ou de perte de contact, qu'ils soient multiples ou uniques.

3.1.1 Recherche du temps de changement d'état

Afin de déterminer si un changement intervient dan l'intervalle $(0,dt)$, il faut commencer par définir les trajectoires des différentes pièces et ensuite rechercher l'instant exact du changement s'il existe.

3.1.1.1 *Définition de la forme de la trajectoire*

Nous avons dans un premier temps utilisé la technique proposée par Redon, utilisant l'axe central mais cette technique n'est utilisable que si les éléments constituant la géométrie sont des segments ou des plans et si les vitesses de translation et de rotation sont faibles. Ces limitations ne sont pas acceptables dans notre cas puisque nous souhaitons travailler avec des géométries plus complexes et que l'animation de mécanisme conduit souvent à des vitesses importantes, pendant la chute d'un solide par exemple.

Nous avons effectué une première modification après avoir constaté qu'il était impossible de faire glisser un segment sur un cercle si l'on supposait que la vitesse était constant de 0 à dt. En effet si le segment est sur le cercle et qu'il glisse à l'intérieur de celui-ci à dt, le segment est obligatoirement à l'extérieur du cercle.

La dynamique utilisée nous donnant la valeur de l'accélération à instant donné, il nous est paru plus logique d' « intégrer » les accélérations pour obtenir les déplacements des objets. Nous obtenons ainsi par exemple pour la translation sur x :

$$x(t)=\frac{1}{2}c_x t^2+v_x(0)t \tag{3.1}$$

Au lieu de :

$$x(t)=v_x(0)t \tag{3.2}$$

dans le travail de Redon.
Après plusieurs tests, nous avons remarqué que la solution de l'équation 3.1 ne donnait des bons résultats que sur des segments entre eux. Lors du glissement d'un segment sur un cercle, celui-ci sortait plus ou moins rapidement du cercle en fonction du coefficient d'absorption, de la taille du rayon, etc. La raison est assez simple, nous supposons qu'entre 0 et dt, l'accélération est constante et en fait celle-ci varie énormément d'un point à un autre.
En analysant le cas d'un point, soumis à la gravité, glissant sur un cercle en 2D, nous voyons que la forme de l'accélération en fonction du temps est proche d'une sinusoïde. Nous avons donc réécrit l'équation 3.1 pour qu'après dérivation, nous obtenons les premiers termes du développement des fonctions sinus et cosinus. Ce qui donne :

$$x(t)=\frac{1}{24}a_x t^4+\frac{1}{6}b_x t^3+\frac{1}{2}c_x t^2+d_x t \tag{3.3}$$

On obtient l'accélération en translation sur x :

$$\gamma_x(t)=\frac{1}{2}a_x t^2+b_x t+c_x \tag{3.4}$$

La détermination des paramètres a_x, b_x et c_x est obtenue à partir des accélération à $t=0$: γ_x^0, $t=-dt_1$: γ_x^- et $t=-(dt_1+dt_2)$: γ_x^{--}, c'est-à-dire avec l'accélération obtenue par la dynamique et avec les deux accélérations précédentes qu'il faut stocker pour chaque pièce en mouvement. Après calcul, on a :

$$a_x = 2\frac{\gamma_x^0 . dt_2 - \gamma_x^- . (dt_1 + dt_2) + \gamma_x^{--} . dt_1}{dt_1 . dt_2 . (dt_1 + dt_2)} \tag{3.5}$$

Et,

$$b_x = \frac{\gamma_x^0 . dt_2 . (2.\mathrm{dt}_1 + dt_2) - \gamma_x^- . (dt_1 + dt_2)^2 + \gamma_x^{--} . dt_1^2}{dt_1 . dt_2 . (dt_1 + dt_2)} \tag{3.6}$$

Si la pièce ne glisse pas, elle n'est soumise qu'à la gravité, dans ce cas les paramètres a_x et b_x sont nuls.
Nous avons donc maintenant une définition de la trajectoire suffisamment générale et que nous allons utiliser pour déterminer le moment du choc.

Les éléments ci-dessus nous permettent d'obtenir la définition de la trajectoire en translation mais il n'est pas possible d'utiliser directement les angles obtenus. L'orientation des solides doit être définie par une matrice de rotation, nous avons choisi d'utiliser les quaternions pour passer des angles à la matrice de rotation.
On utilise une matrice colonne $\boldsymbol{\theta}$ définie par les trois paramètres θ_x , θ_y et θ_z :

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} \tag{3.7}$$

On peut en déduire l'angle de rotation total θ :

$$\theta = \|\boldsymbol{\theta}\| \tag{3.8}$$

Si l'on calcule les variables suivantes :

$$\begin{aligned} q_w &= \cos\left(\frac{\theta(t)}{2}\right) \\ q_x &= \frac{\theta_x(t)}{\theta(t)} \sin\left(\frac{\theta(t)}{2}\right) \\ q_y &= \frac{\theta_y(t)}{\theta(t)} \sin\left(\frac{\theta(t)}{2}\right) \\ q_z &= \frac{\theta_z(t)}{\theta(t)} \sin\left(\frac{\theta(t)}{2}\right) \end{aligned} \tag{3.9}$$

Le quaternion correspondant à la rotation s'écrit :

$$q_w + i\,q_x + j\,q_y + k\,q_z \tag{3.10}$$

Et la matrice de rotation :

$$\begin{bmatrix} 1-2\,q_y^2-2\,q_z^2 & 2\,q_x q_y - 2\,q_z q_w & 2\,q_x q_z + 2\,q_y q_w \\ 2\,q_x q_y + 2\,q_z q_w & 1-2\,q_x^2-2\,q_z^2 & 2\,q_y q_z - 2\,q_x q_w \\ 2\,q_x q_z - 2\,q_y q_w & 2\,q_y q_z + 2\,q_x q_w & 1-2\,q_x^2-2\,q_y^2 \end{bmatrix} \tag{3.11}$$

La trajectoire de chaque pièce est donc définie par six variables, trois translations et trois angles, qui sont déterminés par des équations du même type que

l'équation 3.3 . Les trois translations définissent le vecteur translation et les trois rotations, la matrice de rotation à l'aide de l'équation 3.11.

3.1.1.2 Recherche du temps de contact

Cette recherche s'effectue en 3 temps :
- recherche des intervalles de contact
- recherche du minimum de distance
- vérification de la direction du contact

L'utilisation de ces trois techniques doit nous permettre de trouver le meilleur compromis pour minimiser le temps de recherche tout en gardant une bonne qualité du résultat. Nous utilisons l'arithmétique des intervalles pour faire une première recherche tout en balayant l'ensemble de l'espace de recherche ce qui nous permet de ne pas oublier de contact. Il est facile de déterminer la précision avec laquelle on obtient le temps jusqu'au contact puis que l'intervalle initial est $[0 \;\; 0.0033]$ (33 ms correspondant approximativement à une fréquence de 300 Hz) et que l'on divise l'intervalle par 2 à chaque descente dans l'arborescence. Pour un niveau N de la hiérarchie, la taille finale de l'intervalle est $\frac{0.0033}{2^N}$.

Une fois les intervalles de la zone de contact trouvés, une méthode nous permet d'affiner la recherche en utilisant cette fois une notion de distance entre les entités. La bonne localisation du point de contact va nous permettre de vérifier que les éléments géométriques sont bien en phase d'approche, en effet juste après un choc, nous pouvons détecter un contact avec un temps très court qui correspond au point en phase d'éloignement.

Recherche des intervalles de contact

L'objectif de cette méthode est de trouver dans les intervalles : $[0 \;\; dt]$ pour le temps, $[0 \;\; 1]$ pour u_1 et $[0 \;\; 1]$ pour u_2 (pour le contact entre deux courbes), les sous-intervalles correspondent à la zone de contact s'il y en a une.
Les sous-intervalles découlent d'une hiérarchie dichotomique d'intervalles (Fig 3.4).

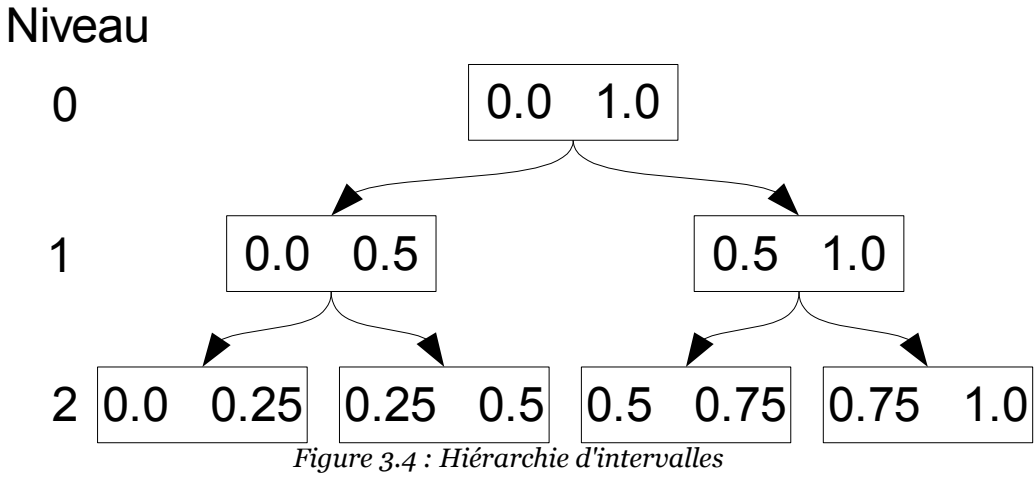


Figure 3.4 : Hiérarchie d'intervalles

La génération des hiérarchies d'intervalles sur les géométries peut être réalisée avant le lancement de la simulation. Si l'on plaque la hiérarchie précédente sur un segment, on obtient par exemple (Fig. 3.5):

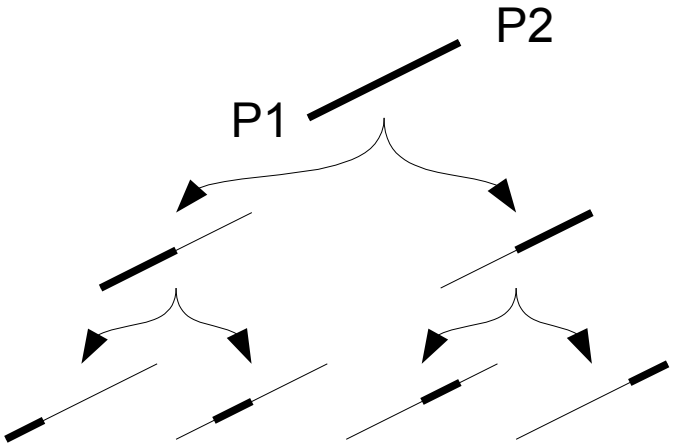


Figure 3.5 : Hiérarchie d'intervalles sur un segment

Nous avons fait le choix d'utiliser le même niveau de hiérarchie pour le temps et pour les paramètres des éléments géométriques (u_i , v_i) ce qui nous permet lors de la recherche de rester au même niveau dans les hiérarchies lors de la recherche du contact.
Pour vérifier la possibilité de contact entre deux entités, nous calculons le vecteur intervalle $\widetilde{P_1P_2}$ où P_1 est un point sur l'entité 1 et P_2 est un point sur l'entité 2 .
La figure 3.6 montre les déplacements pendant un intervalle de temps d'un point P_1 situé sur un intervalle d'un segment vertical en rotation autour de l'axe y et d'un point P_2 situé sur un intervalle d'un segment dirigé suivant l'axe x et en translation rectiligne dans le plan (y, z) .

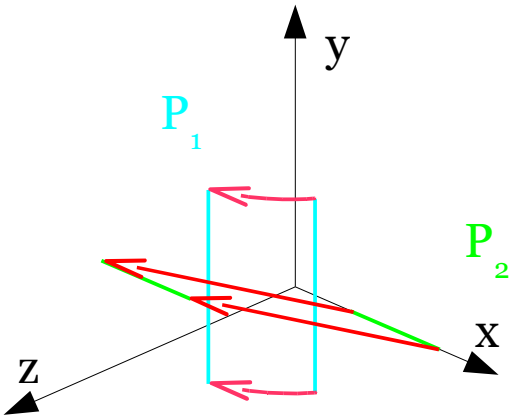


Figure 3.6 : Déplacement de deux segments sur un intervalle de temps

Le tableau 12 montre l'évolution des deux segments au cours du temps, les intervalles représentés sont de gauche à droite $[0\ \frac{dt}{8}]$, $[0\ \frac{dt}{4}]$, $[0\ \frac{dt}{2}]$ et $[0\ dt]$. Nous pouvons ainsi voir comment progressent les deux surfaces correspondant aux positions possibles des points P_1 et P_2 .

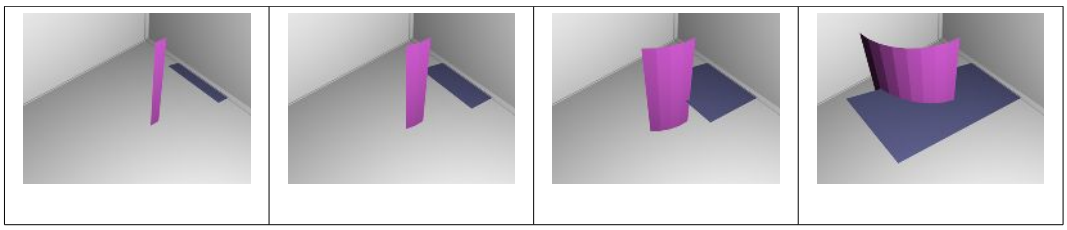


Tableau 12 : Evolution des deux segments au cours du temps

Pour déterminer s'il y a une possibilité de contact, nous vérifions si les différentes projections du vecteur intervalle $\widetilde{P_1P_2}$ possèdent un recouvrement. Cette technique augmente la possibilité de contact mais permet d'accélérer les calculs, nous pouvons en effet commencer à vérifier s'il y a recouvrement suivant l'axe x. Si ce n'est pas le cas, on sait qu'il n'y aura pas contact et donc arrêter la recherche sinon, nous vérifions le recouvrement sur l'axe y puis sur l'axe z. Dans notre exemple, on peut remarquer qu'il y a recouvrement sur x et y (Fig. 3.7) et aussi sur z (Fig. 3.8).

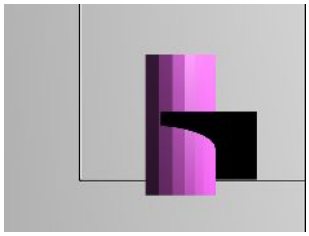


Figure 3.7 : Projection des intervalles sur x et y

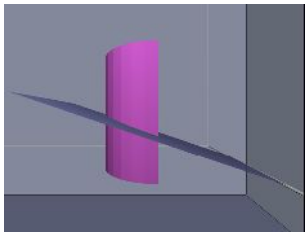


Figure 3.8 : Projection des intervalles sur z

On peut constater dans ce cas que toutes les projections ont un recouvrement, il y a donc possibilité de contact et il faut continuer à descendre dans la hiérarchie d'intervalles. Le tableau 12 nous donne un exemple de descente dans cette hiérarchie si l'on regarde les images de droite à gauche, nous pouvons remarquer dans ce cas que cette descente est inutile dès la troisième image puisque les projections sur z ne se recouvrent pas.

La navigation dans la hiérarchie d'intervalles doit permettre de trouver la zone de contact le plus rapidement possible, en effet à chaque descente dans la hiérarchie, nous effectuons 8 nouveaux tests (pour le contact entre deux courbes) puisque nous coupons en deux l'intervalle du temps et les deux intervalles des paramètres u_i des géométries.

Pour illustrer (Fig. 3.9) le fonctionnement de la recherche, nous pouvons choisir un cas plus simple dans lequel nous présentons le problème de la recherche de contact entre un cercle et un segment en projection dans le plan du cercle et le segment étant perpendiculaire à ce plan. Le schéma supérieur correspond au niveau le plus élevé de la hiérarchie, point de départ de la recherche. On remarque que le cercle ne se déplace pas dans le plan, celui-ci est donc encadré par un carré. Le segment qui donne un point en projection est soumis à une translation rectiligne. Le point part près du centre du cercle pour se trouver finalement à l'extérieur, en bas et à droite du cercle.

A la première descente dans la hiérarchie, nous retrouvons 4 possibilités à tester dans l'ordre : haut du cercle/haut de la trajectoire, haut du cercle/bas de la trajectoire, bas du cercle/haut de la trajectoire, bas du cercle/bas de la trajectoire. Les deux premiers essais sont sans recouvrement sur les deux axes. Le troisième essai a bien un recouvrement sur les deux axes, nous effectuons donc une nouvelle descente. Ainsi de suite jusqu'au dernier niveau de la hiérarchie.

Le calcul du vecteur intervalle $\widetilde{P_1P_2}$ doit aussi être rapide et s'effectue à l'aide

de l'arithmétique des intervalles. Les trajectoires des solides de 0 à dt sont définies à partir des accélérations calculées par la dynamique et des relations du paragraphe 3.1.1.1.
L'arithmétique des intervalles a été codée en C++ dans un souci de rapidité. Des approximations ont été faites sur le calcul des fonctions les plus coûteuses. Quand les angles deviennent petits, nous utilisons directement l'angle plutôt que son sinus. Lors du calcul du quaternion si l'angle est petit, on peut remplacer les équations 3.9 par :

$$\begin{aligned} q_w &\sim 1 - \left(\frac{\theta(t)}{2}\right)^2 = 1 - \left(\frac{\theta_x(t)^2 + \theta_y(t)^2 + \theta_z(t)^2}{4}\right) \\ q_x &\sim \frac{\theta_x(t)}{2} \\ q_y &\sim \frac{\theta_y(t)}{2} \\ q_z &\sim \frac{\theta_z(t)}{2} \end{aligned} \qquad (3.12)$$

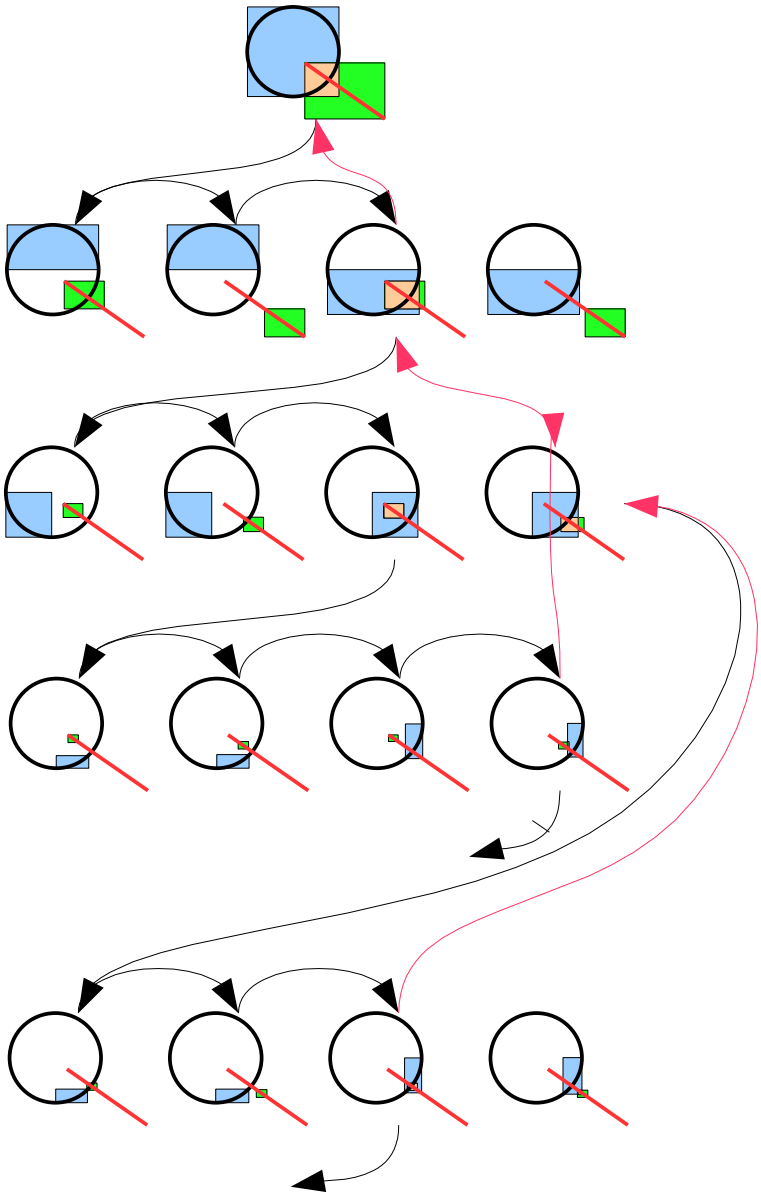


Figure 3.9 : Navigation dans la hiérarchie d'intervalles

Recherche du minimum de distance

A partir des intervalles précédents et de la valeur du temps de contact, nous calculons les 9 distances (Fig. 3.10) entre 3 points (les deux extrémités et le centre) sur l'intervalle de l'entité géométrique 1 et les 3 autres points sur l'intervalle de l'entité géométrique 2 .

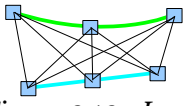
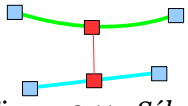
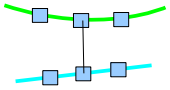
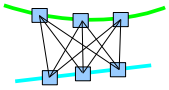
 *Figure 3.10 : Les 9 distances sur les intervalles de départ*	 *Figure 3.11 : Sélection des points les plus proches*	 *Figure 3.12 : Division par 2 de la longueur des intervalles*	 *Figure 3.13 : Nouvelle recherche*

Tableau 13 : Recherche du minimum de distance

On sélectionne ensuite les points les plus proches par simple recherche de la distance minimum (Fig. 3.11). La taille des intervalles est ensuite divisée par deux et quatre nouveaux points sont déterminés (Fig. 3.12). Une nouvelle recherche peut être lancée (Fig. 3.13) et ce processus est reproduit jusqu'à obtention de la précision désirée sur les paramètres u_i des entités géométriques.

Vérification de la direction du contact

Il faut savoir si les entités géométriques sont en phase d'approche ou de répulsion au niveau du contact. En effet, juste après un choc, le contact est à nouveau détecté même si physiquement il n'existe pas. Cette vérification nécessite de déterminer la position physique du contact avec suffisamment de précision d'où l'étape précédente de recherche du minimum de distance.
La vitesse relative est calculée au point de contact puis elle est projetée sur la normale au point de contact. Une simple observation des changements de sens de la projection permet de savoir si on est en phase d'approche ou l'inverse.

3.1.2 Déplacement jusqu'au contact

Deux cas peuvent se produire :

- il n'y a pas eu de contact détecté dans l'intervalle $[0\ 0.0033]$ et dans ce cas, il n' y a pas de changement d'état.
- un temps de contact a été calculé et il faut déterminer le changement d'état correspondant. Dans les deux cas, nous devons déplacer les pièces soit de $0{,}0033\,s$, soit du temps jusqu'au contact.

Le mouvement des pièces est déterminé par les équations du mouvement décrites au paragraphe 3.1.1.1. Les paramètres de ces équations sont identiques à ceux utilisés pendant la recherche du temps afin d'obtenir le contact souhaité.

3.1.3 Recherche de changement d'état au contact

Dans cette étape, nous savons qu'il existe au moins un point de contact mais celui-ci peut être en glissement ou en choc. D'un point de vue informatique des listes sont utilisées pour mémoriser l'évolution des contacts. Nous supposons ici que nous sommes capables de détecter le cas dans lequel nous nous trouvons.

3.1.3.1 Nouveau choc

Nous réutilisons les mêmes types de calcul que pour la recherche du temps de contact mais en fixant cette fois l'intervalle de temps au plus petit intervalle de temps proche de 0. Puis nous cherchons tous les contacts possibles à cet instant, il n'y a donc pas de restriction quant au nombre de contact entre deux pièces.

Dans les exemples que nous traitons, nous avons un segment qui tombe à l'intérieur de deux cercles de même diamètre (Fig. 3.14). Le système détecte bien les deux contacts qui ont lieu exactement au même instant, ce qui donne lieu à

deux contraintes de choc, utilisées ensuite dans la dynamique sous contraintes.

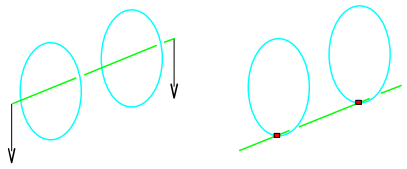


Figure 3.14 : 2 contacts sur 2 entités

Il est par contre plus délicat de détecter plusieurs contacts sur une même entité géométrique comme dans le cas d'un segment qui tombe à plat sur un cylindre à l'horizontale (Fig. 3.15).

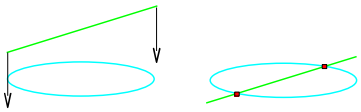


Figure 3.15 : 2 contacts sur une entité

En pratique, l'éventualité de chocs simultanés est quasi nulle mais la non-prise en compte de ces cas particuliers conduira à des pertes de contacts et donc à des simulations erronées. D'autre part, ces cas particuliers peuvent devenir beaucoup plus courants en glissement, il est donc nécessaire de traiter dès à présent tous les chocs simultanés possibles.

Nous allons commencer par l'explication de la recherche d'un seul choc puis nous reviendrons sur la méthode employée pour trouver les autres chocs éventuels.

Recherche d'un seul choc

Cette recherche s'effectue en 3 temps :
- recherche des intervalles de contact
- recherche du minimum de distance
- recherche géométrique du point de contact

Les deux premières méthodes sont identiques à celles du paragraphe 3.1.1.2. Comme nous l'avons indiqué précédemment, cette fois il n'y a pas de descente dans la hiérarchie des intervalles du temps. La connaissance précise du lieu du contact est importante pour assurer la qualité de la simulation, nous avons ajouté un dernier calcul simple pour affiner les valeurs des paramètres u_i .

Recherche géométrique du point de contact

Après la recherche du minimum de distance, nous avons des valeurs assez précises des paramètres u_i , il est alors possible d'approximer les courbes par des segments tangents puis de chercher la distance minimum entre les deux segments.

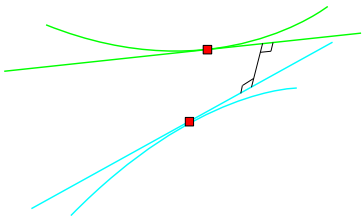


Figure 3.16 : Recherche géométrique du point de contact

Recherche de plusieurs chocs sur une même entité géométrique

Le nombre maximum de points de contact est 3 comme nous l'avons montré au paragraphe 2.2.2.4 cependant nous n'avons besoin de chercher que deux points de contact, le troisième point est défini par les deux premiers et nous ne faisons que vérifier si ce point est en choc. Nous allons illustrer ce point sur le contact entre deux cercles.

Recherche d'un choc entre deux cercles

Dans la méthode de descente proposée au paragraphe 3.1.1.2, les paramètres u_i sont parcourus en croissant de 0 à 1. S'il n'y a qu'un point de choc, nous trouvons obligatoirement ce point (Fig. 3.17).

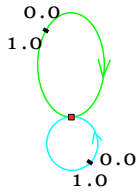


Figure 3.17 : Un choc entre deux cercles

Recherche de deux chocs entre deux cercles

Si l'on n'applique que la méthode précédente, on ne trouve qu'un des deux points de contact (Fig. 3.18).

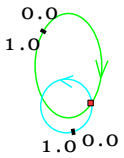


Figure 3.18 : Un premier point de choc trouvé

La solution pour trouver le deuxième point (Fig. 3.19) est simple, il suffit de reparcourir l'arborescence en sens inverse (les paramètres u_i décroissants).

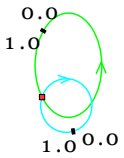


Figure 3.19 : Un deuxième point de choc trouvé

Recherche de trois chocs entre deux cercles

Nous ne sommes pas réellement en présence d'un troisième point de choc mais s'il y a un troisième point en contact où qu'il soit, alors les deux cercles se touchent complètement lors du choc. A partir des deux premiers points de choc, on détermine la position du troisième point. Dans la figure ci-dessous (Fig. 3.20), on obtient par exemple $u_1=0.25$ pour le premier point et $u_1=0.75$ pour le deuxième point. On peut alors vérifier s'il existe un contact au point équidistant

en u : $u_1 = \frac{(0.25+0.75)}{2} = 0.5$. Si ce contact existe, les deux cercles sont les uns sur les autres et il faut gérer le choc correspondant, sinon c'est qu'il n'existe que deux points de choc.

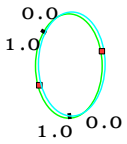


Figure 3.20 : Détermination du troisième point

3.1.3.2 Modification d'un glissement

Si le point n'est pas en choc, c'est qu'il est en glissement. Dans ce cas, il faut mettre à jour les coordonnées, les vitesses, etc. du point de glissement. Nous sommes partis du principe qu'un point de glissement change progressivement de position et qu'il était donc plus rapide de trouver le nouveau point de glissement à partir du précédent. La hiérarchie d'intervalles n'est donc pas utilisée, nous utilisons par contre comme pour la recherche du choc, la recherche du minimum de distance suivie de la recherche géométrique.

La recherche du minimum de distance commence cette fois par la recherche des meilleurs intervalles pour les paramètres u_i . Dans la figure ci-dessous (Fig. 3.21), nous remarquons que nous passons du point de départ au nouveau point de glissement par une série de bonds de même amplitude, correspondant à l'intervalle le plus petit sur les u_i . Un fois les intervalles correspondant à la distance minimum trouvée, nous utilisons la même méthode de recherche de minimum qu'au paragraphe 3.1.1.2. Comme nous l'avions vu dans ce paragraphe, l'amplitude des bonds est divisée par deux à chaque nouvelle recherche.

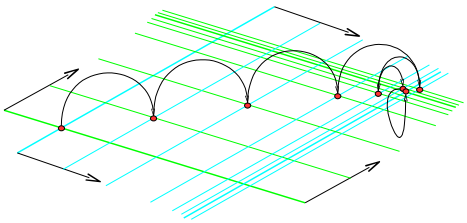


Figure 3.21 : Recherche de la nouvelle position du point de glissement

Plusieurs techniques sont envisageables pour accélérer la recherche du nouveau point. On sait que le nouveau point se retrouvera plus ou moins loin en fonction de la vitesse de déplacement du point de départ, cette information peut être utilisée pour estimer la position du point d'arrivée. Cette méthode serait terminée par la même recherche de distance minimum. Nous n'avons pas cherché actuellement à utiliser ce type de technique car nous savons que la perte de temps est mineure.

3.2 Application des nouveaux états

Nous avons maintenant une vue d'ensemble des nouveaux états du système, le nombre de chocs et de glissements, la position des points de contact, etc. Il faut maintenant appliquer les contraintes correspondantes et en déduire les nouvelles accélérations.

Nous devons commencer par gérer les nouveaux chocs qui vont modifier les vitesses des différentes pièces et ensuite déterminer les nouvelles accélérations sous l'effet des contraintes de glissement.

3.2.1 Gestion des nouveaux chocs

Cette étape va permettre de faire rebondir les pièces en appliquant les contraintes de choc mais elle va tout d'abord effectuer un test permettant de déterminer si au lieu d'un choc, nous ne sommes pas au début d'un glissement.

Test d'entrée en glissement

Ce test est très simple puisqu'il se limite au calcul de l'angle au point de choc entre la normale et la vitesse juste avant le choc. Si cet angle est proche de l'angle droit, nous entrons en glissement sinon c'est simplement un choc. Actuellement le choix de cet angle limite s'effectue de manière totalement empirique, certains paramètres influants sur ce choix sont assez faciles à identifier comme la courbure au point de contact par exemple, mais nous aimerions si possible que ce test reste le plus simple et le plus général possible.

Ce calcul, après essai, a été légèrement modifié, en effet, le calcul de l'angle est inutile, nous pouvons par contre utiliser :

$$projection = \frac{\overrightarrow{V_{12}^{P}} . \vec{n}}{\| \overrightarrow{V_{12}^{P}} \| . \| \vec{n} \|} \tag{3.13}$$

ce qui correspond au sinus de l'angle.
Si nous prenons le cas d'un segment tombant verticalement à partir du centre d'un cercle, le point de choc reste théoriquement toujours le même (le point le plus bas du cercle). Dans ce cas, après plusieurs rebonds, la vitesse qui précède le choc, diminue en intensité mais sa direction reste celle de $\vec{n}$ d'où une valeur de la projection égale à 1.
Physiquement cette situation est absurde, ce point devrait après un certain temps être considéré en glissement sur le cercle. Nous avons donc choisi de modifier l'équation 3.13 pour que la projection s'annule quand la vitesse tend vers 0.
D'où le nouveau calcul de la projection :

$$projection = \frac{\overrightarrow{V_{12}^{P}} . \vec{n}}{\| \vec{n} \|} \tag{3.14}$$

Le vecteur $\vec{n}$ n'est pas normalisé, il correspond en fait au produit vectoriel des deux tangentes.

Le choix de la valeur limite de la projection se fait actuellement de manière empirique mais nous pouvons assez rapidement trouver une bonne valeur, en effet, il faut choisir cette valeur pour qu'il n'y ait jamais de passage direct d'un choc à un autre. La figure de gauche (Fig. 3.22) représente un minimum, c'est-à-dire au moins un point libre entre deux chocs. Dans celle de droite, la simulation finira par perdre le contact. En effet dans le premier cas, le temps minimum entre deux contacts est de 0,0033 s puisqu'il y a au minimum un pas de temps durant lequel on ne trouve pas de contact. Dans le second cas, en fonction du contact trouvé, le pas de temps entre deux contacts peut tendre vers 0. L'élimination du cas 2 permet aussi de supprimer les effets de rebond qui peuvent se produire en fin de mouvement quand l'objet est bloqué dans un coin.

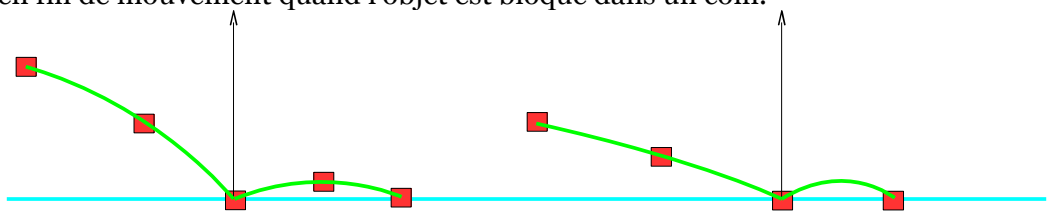


Figure 3.22 : Problème des chocs successifs

Détermination des vitesses après choc

Nous nous contentons ici d'appliquer les contraintes de choc avec les coefficients d'absorption correspondants. En cas l'entrée en glissement, il suffit d'utiliser un coefficient d'absorption nul. Ceci aura comme effet très intéressant de remettre la vitesse au point de contact totalement tangente aux entités géométriques ce qui est indispensable pour une application des contraintes de glissement.

3.2.2 Gestion des glissements

Nous appliquons cette fois les contraintes de glissement afin de déterminer les accélérations contraintes. Cette étape est déterminante pour la qualité des calculs effectués dans les étapes précédentes. Un bonne connaissance des paramètres u_i et de leur évolution au cours du temps est indispensable. Comme pour la résolution d'équations différentielles, une bonne partie du résultat dépend des conditions aux limites.

Il est très important de comprendre qu'il n'est pas question de corriger la trajectoire des solides en vérifiant la pénétration des pièces. L'application des contraintes définit à elle seule la trajectoire.

3.3 Conclusion sur la gestion des changements d'états

La gestion des changements d'états est un véritable travail d'horlogerie, l'objectif à atteindre est un objectif de précision suffisant pour assurer la robustesse de la simulation. Tous les cas possibles doivent être prévus et correctement gérés. Une fois ce travail réalisé, tout devient simple et presque évident. Les difficultés

classiques en haptique comme la gestion des contacts multiples ou l'utilisation d'un retour en 6D, sont directement prises en compte.

Cette gestion se décompose en série d'étapes exécutées séquentiellement, toute erreur dans les premières étapes rend les suivantes inopérantes. De nombreuses méthodes ont été mises en place (arithmétique des intervalles, recherche de distance dichotomique, etc.) qu'il faut tester rigoureusement avant de passer à l'étape suivante. Nous montrerons sur des exemples que nous avons réussi à implémenter dans notre deuxième outil une version certes perfectible mais fonctionnelle de cette gestion.

4 Développements des outils d'esquisse virtuelle

Un premier outil d'esquisse a été développé [BA09], permettant de vérifier les possibilités d'utilisation du retour d'effort lors de la réalisation d'esquisse en 3D. Ce premier outil a aussi mis en évidence les inconvénients de certains modèles tant géométriques que dynamiques ce qui nous a en particulier conduit à définir notre propre dynamique sous contraintes.

La première version de l'outil nous a cependant permis de définir l'interface graphique et de valider le choix des périphériques. Et ainsi, nous prouvons qu'il était indispensable d'avoir un retour d'effort pour dessiner un mécanisme dans l'espace. Nous aurions pu simplement améliorer cette version mais plusieurs points nous sont apparus très importants après essais : la possibilité de simuler des mécanismes avec jeu, une détection de collision continue sur des géométries elles aussi continues, l'utilisation détournée de la dynamique sous contraintes pour la modification de l'esquisse virtuelle.

Dans la présentation du premier outil, nous nous intéresserons plus particulièrement à l'interface, aux périphériques utilisables ainsi qu'aux avantages et aux inconvénients de cet outil lors des essais. La présentation du deuxième outil permettra de montrer les avancées obtenues à partir d'exemples couvrant plusieurs types d'utilisations de l'esquisse virtuelle.

4.1 Première version de l'outil

L'objectif de cette première version est de monter qu'il est effectivement possible de dessiner dans l'espace à l'aide du retour d'effort et ensuite d'animer un mécanisme. Cette première étape est validée. Nous avons, en effet, mis au point

un outil permettant de créer un mécanisme avec des pièces sous forme filaire, la preuve est faite que le retour d'effort est indispensable au dessin dans l'espace (voir Fig. 4.1). Nous avons en effet essayé de reproduire les mêmes systèmes en filaire avec la souris et le résultat n'est pas exploitable.

Figure 4.1: Esquisse en 3D avec retour d'effort et orientation à la SpaceMouse

Nous allons maintenant présenter la structure matérielle et logicielle de l'outil.

4.1.1 Les interactions avec l'utilisateur

La figure 4.2 représente les différentes possibilités d'interaction avec l'outil d'esquisse virtuelle. La seule interaction que nous n'avons pas testé pour l'instant est la voix de l'utilisateur vers le système par de la reconnaissance vocale. Le choix a été fait d'une manipulation à deux mains : la main gauche pour l'orientation de l'objet et la droite pour le tracé. La visualisation doit normalement s'effectuer en stéréoscopie. Le son se limite actuellement à la synthèse vocale mais il serait intéressant de chercher à générer des sons liés aux contacts entre les pièces en mouvement.

La main droite est fortement sollicitée, c'est le seul organe qui fonctionne dans les deux sens, c'est elle qui impose la position du stylet mais c'est elle aussi qui est soumise au retour d'effort. Le retour d'effort peut être envisagé dans la main gauche. Dans l'utilisation du système avec une souris, c'est la main droite qui doit servir pour effectuer l'orientation, ce qui n'est pas très ergonomique. La sélection dans les menus est aussi réalisée par la main droite quel que soit le périphérique utilisé.

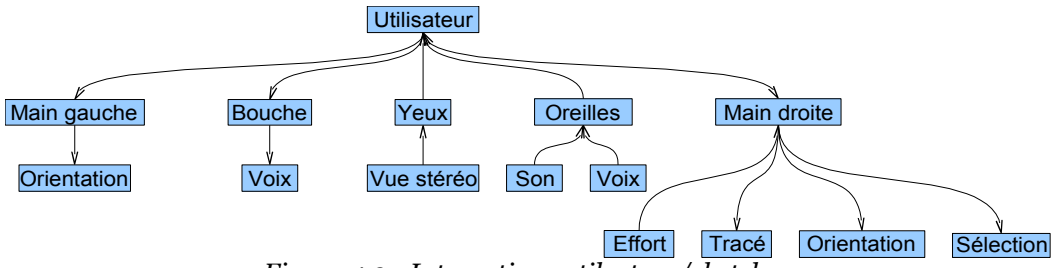


Figure 4.2 : Interactions utilsateur/sketcher

4.1.2 Les périphériques

Comme nous l'avons vu dans l'état de l'art, il existe aujourd'hui de nombreux périphériques disponibles en réalité virtuelle, nous avons choisi de nous limiter à certains périphériques en fonction de critères simples comme la facilité d'utilisation et le prix (SpaceMouse, souris, système à retour d'effort, écran avec ou sans stéréoscopie, etc.). Ce choix est lié à l'application développée qui ne nécessite pas à priori une trop grande qualité des différents rendus et aussi au fait que nous partons du principe que le développement d'un prototype doit permettre de démontrer la robustesse de la solution. Il ne nous paraît donc pas utile de travailler avec un matériel et même avec des logiciels très performants, c'est le résultat global qui doit être validé. Nous ne présenterons que les périphériques en liaison avec les deux mains et les yeux.

4.1.2.1 Main gauche

La main gauche servant à changer la position et l'orientation de l'objet, il est important d'un point de vue ergonomique de ne pas laisser la main en l'air pendant toute la durée de la manipulation. Certains capteurs comme ceux utilisés pour le tracking de main utilisent plutôt la capture du geste et captent donc des mouvements amples. Les nouveaux périphériques utilisés dans les consoles de jeu en sont des exemples. A l'opposé un périphérique comme la SpaceMouse permet d'effectuer tous les mouvements nécessaires avec des déplacements très faibles en entrée.

Les périphériques pour l'orientation (Fig. 4.3) qui sont implantés dans le système sont la SpaceMouse et la souris. Le tracking de main peut être effectué par de nombreuses solutions technologiques, nous n'en avons envisagé que deux autres, le tracking par système magnétique ou par vision. Le Flock of Bird peut facilement être intégré et possède des possibilités de déplacements suffisantes pour notre application. La vision avec la librairie ARToolkit et une WebCam est une solution peu coûteuse.

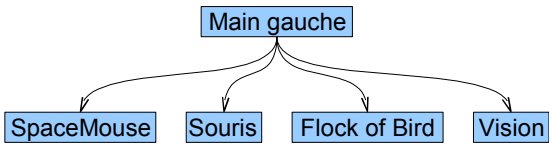


Figure 4.3 : Périphériques Main gauche

4.1.2.2 Main droite

La main droite (Fig. 4.4) est celle qui permet de créer l'objet, il faut donc que les déplacements possibles soient suffisamment importants pour que l'entrée soit naturelle. Par contre plus les mouvements sont amples en 3D, plus la fatigue se fait sentir dans le bras, il est nécessaire de trouver des positions agréables et de choisir des dispositifs nécessitant des efforts minima lors du déplacement.

Un seul périphérique à retour d'effort a été intégré, le Premium 3.0 de la société Sensable fonctionnant avec la librairie Ghost. La souris n'est utilisée que pour des raisons de test et de possiblités de programmation sans périphériques haptiques. La vision est envisagée comme pour l'orientation et finalement l'utilisation d'une tablette graphique orientée avec un Flock of Bird est prévue.

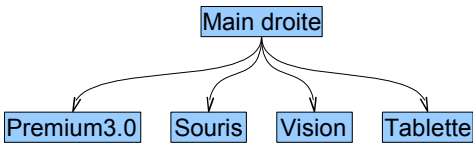


Figure 4.4 : Périphériques Main droite

4.1.2.3 *Les yeux*

De nombreuses solutions existent pour la vision mais elles ne sont pas toutes adaptées au travail à deux mains sur des objets filaires et à une durée d'utilisation dépassant les 5 minutes (casques de vision). Nous souhaitions principalement pouvoir tester la différence entre la vision mono et stéréo. D'autres techniques sont envisageables pour améliorer l'effet de profondeur comme l'ajout de brouillard par exemple.

Seules deux techniques ont été essayées avec la première version du sketcher, la stéréoscopie avec des lunettes et un écran, la monoscopie avec un simple écran. Des essais avec plaque semi-réfléchissante sont prévus. Nous possédons un écran autostréréoscopique mais qui ne peut pas évoluer pour notre application, ce système n'est plus maintenu et les liens avec Java3D sont inexistants..

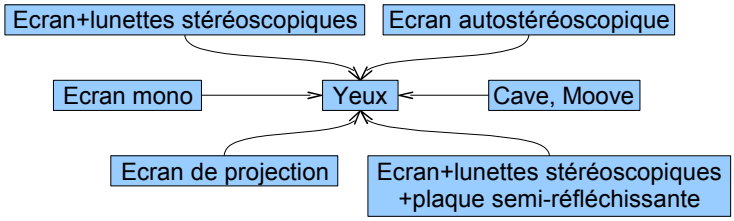


Figure 4.5 : Périphériques occulaires

4.1.3 Le logiciel

La présentation de la partie logicielle de l'outil d'esquisse virtuelle va suivre en partie la décomposition matérielle précédente. La flexibilité dans la gestion des périphériques a simplement été gérée par changement de paramètres dans le programme. Nous retrouverons en liaison avec les yeux : la gestion de l'affichage et en liaison avec la main droite : la gestion du retour d'effort. L'animation du mécanisme est pilotée par la gestion de la dynamique. Ce programme faisant fonctionner plusieurs processus, une gestion des tâches permet de séquencer ceux-ci aux vitesses désirées et permet d'assurer un échange fiable d'informations.

Les développements de cette première version ont été effectués pour l'interface en Java, Java3D sous Kawa et la partie haptique/dynamique en C++ avec les librairies OpenDE pour la dynamique, Ghost pour l'haptique sous Visual Studio 6.0.

4.1.3.1 *La gestion des périphériques*

Tous les périphériques précédemment cités n'ont pas été intégrés à l'application, notre objectif était surtout de faciliter la mise au point de celui-ci et de tester différentes combinaisons : avec stéréo/sans stéréo, avec retour d'effort/sans

retour d'effort.

Les différents paramètres sont directement modifiés dans le code en Java :

```
int peripheriqueDeplacement = 0; //déplacement du plan de travail à la souris
//int peripheriqueDeplacement = 1; //déplacement du plan de travail à la
SpaceMouse
//int peripheriqueDeplacement = 2; //déplacement du plan de travail avec un Flock
of Bird

int peripheriqueDessin = 0; //dessin à la souris
//int peripheriqueDessin = 1; //dessin à la tablette
//int peripheriqueDessin = 2; //dessin avec retour d'effort

//int peripheriqueVisualisation = 0; //plein écran avec stéréo
int peripheriqueVisualisation = 1; //plein écran sans stéréo
//int peripheriqueVisualisation = 2; //visualtisation dans une fenêtre en mono

//int peripheriqueHaptique = 0; //Retour d'effort avec le Phantom 3.0
int peripheriqueHaptique = 1; //Retour d'effort avec le Phantom Omni


int syntheseVocale = 0; // pas de synthèse vocale
//int syntheseVocale = 1; // utilisation de la synthèse vocale

double echelleZoom = 1.0;
```

La configuration ci-dessus est la configuration minimale, l'orientation de la scène s'effectue à la souris (peripheriqueDeplacement = 0) ainsi que le dessin des pièces (peripheriqueDessin = 0), la visualisation est en plein écran sans stéréo (peripheriqueVisualisation = 1) et il n'y a pas de synthèse vocale (syntheseVocale = 0).

Le choix des différents paramètres modifie la gestion des périphériques et modifie l'environnement de travail. Il est par exemple impossible de dessiner en 3D avec une souris, nous avons donc décidé de forcer le tracé dans le plan de clipping. Cette méthode nous permet de mieux maîtriser la position des autres tracés par rapport à ce plan de clipping même quand on utilise la stéréo.

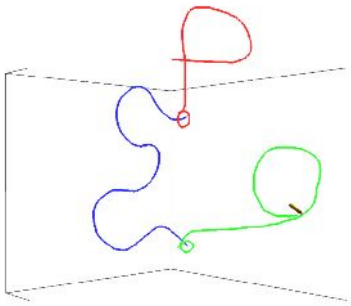


Figure 4.6 : Tracé dans le plan de clipping

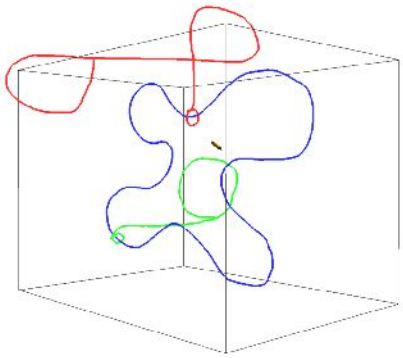


Figure 4.7 : Animation des tracés

On remarque dans la figure 4.6 qu'un clipping est activé, le tracé de la pièce sur la droite se situe donc dans ce plan (z = 0). Le tracé de cette pièce a commencé par une petite boucle autour d'un point de la pièce fixe ce qui permet de simuler une liaison linéaire annulaire. Une fois l'animation lancée (Fig. 4.7), le clipping disparaît et l'on revoit la scène entièrement. Lors de l'utilisation du système haptique le clipping disparaît même lors du dessin de l'esquisse en 3D (voir Fig. 4.1).

4.1.3.2 *La gestion de l'affichage*

Toute la partie interface graphique est effectuée avec le langage Java3D. La gestion des périphériques de visualisation est immédiate, le chargement des fichiers VRML pour les liens avec la CAO, la gestion de la SpaceMouse existe. Comme toutes les librairies de réalité virtuelle, Java3D utilise une arborescence pour créer la scène. La scène actuelle est représentée ci-dessous:

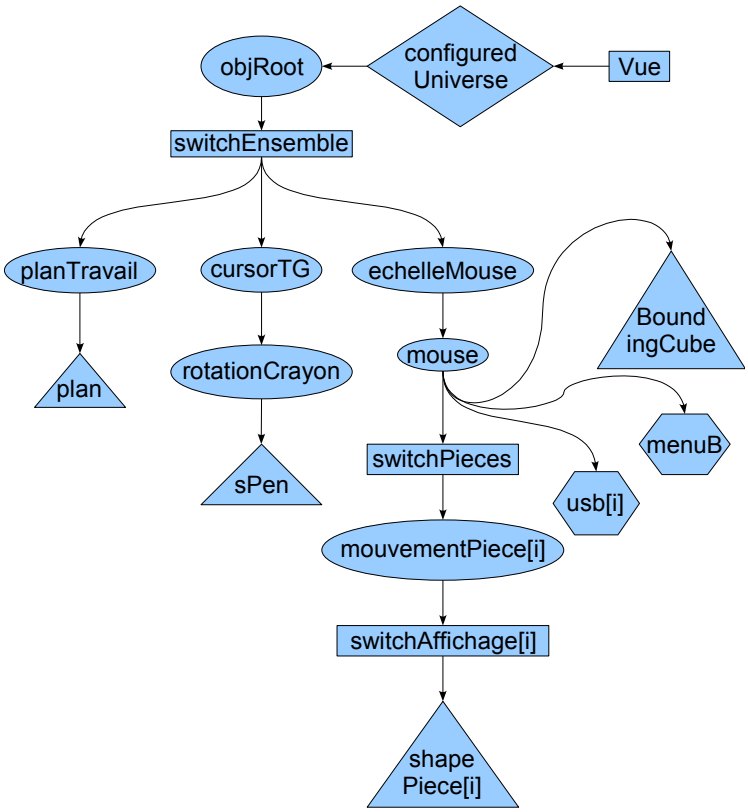


Figure 4.8 : Gestion de l'affichage

Légende de la figure 4.8 :

- shapePiece[i] est le tracé qui est constitué d'un ensemble de segments.
- sPen est un fichier VRML représentant le stylo.
- plan est le plan contenant la texture du menu.
- usb[i] correspond au comportement qui permet la modification des coordonnées des segments constituant le tracé.
- menuB est le comportement attaché à l'utilisation du menu.
- boundingCube est une boite qui délimite le mouvement possible des bras à retour d'effort.
- switchs permettent de choisir les éléments visibles dans la scène, on peut par exemple afficher ou masquer le plan contenant l'image du menu simplement en changeant un bit dans swithEnsemble.

La plupart des ovales sont des TransformGroup qui contiennent une matrice homogène 4x4 permettant de gérer les déplacements des objets.

- cursorTG est relié au déplacement du stylet du système à retour d'effort, les valeurs du déplacement proviennent de la librairie en C++ livrée avec celui-ci.

- mouvementPiece[i] est quand à lui piloté par la librairie de dynamique (OpenDE dans cette version) et permet ainsi de visualiser le mouvement des pièces lors de l'animation.
- mouse permet de faire le lien avec la main gauche et donc avec la rotation du point de vue.
- configuredUniverse permet de gérer les différents systèmes de vision.

4.1.3.3 *La gestion du retour d'effort*

La boucle haptique doit être rafraîchie à 1KHz et sa structure est imposée par les librairies fournies avec les systèmes haptiques. Il faut s'assurer que les calculs effectués dans cette boucle soient rapides sinon la simulation est arrêtée. Voici la structure de notre boucle haptique :

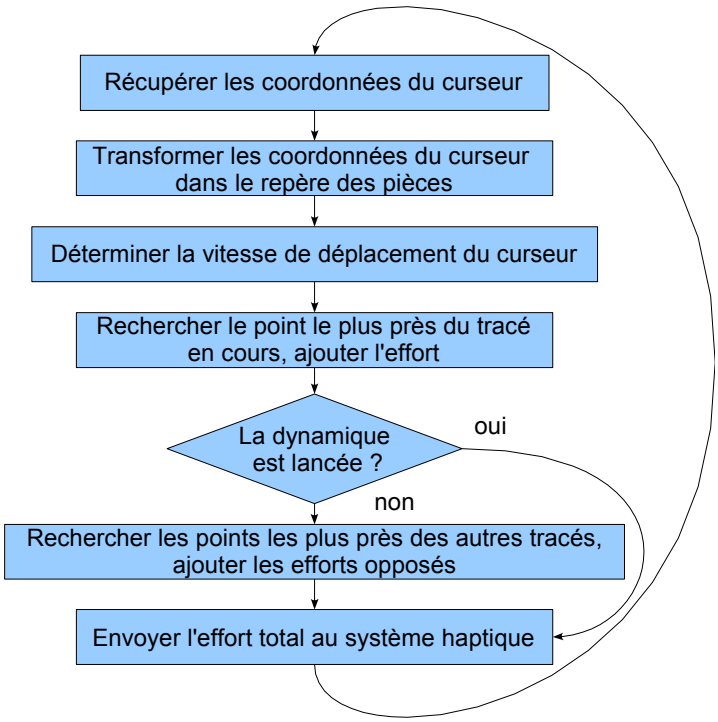


Figure 4.9 : Gestion du retour d'effort

La recherche du point le plus proche est accélérée grâce à une hiérarchie de sphères, cette technique est présentée en Annexe 2. Cette technique a pour avantage de ne pas être modifiée par la rotation du tracé lui-même. Il est simplement nécessaire de recalculer la nouvelle position des centres des sphères de l'arborescence à chaque pas de temps. Les méthodes de type octree, KD-Tree etc. sont mal adaptées aux objets en mouvements. Les hiérarchies de sphères peuvent de plus être utilisées avec des objets soumis à de petites déformations [JA03].

4.1.3.4 *La gestion de la dynamique*

La librairie de dynamique utilisée dans cette première version est OpenDE qui est la librairie « open » la plus utilisée. Une première phase d'initialisation définit les

corps utilisés dans la simulation avec la position du centre de gravité, la masse et les inerties. Ces éléments sont déterminés à partir d'une section et d'une masse volumique prédéfinies pour l'instant. Ensuite la boucle dynamique est lancée, voici une description simplifiée de celle-ci :

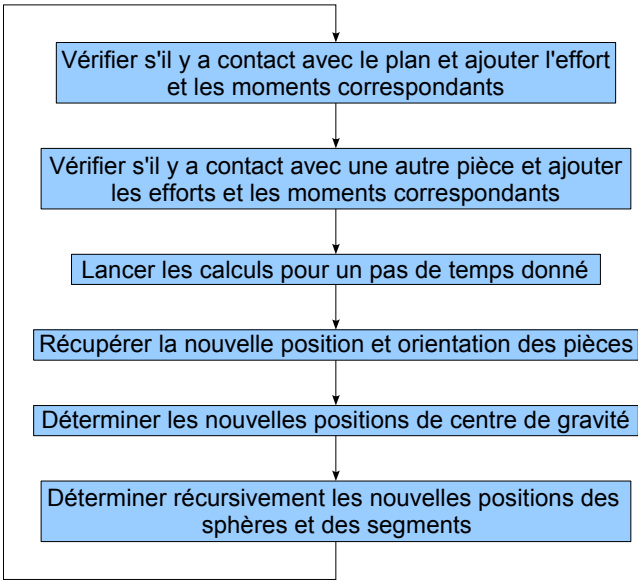


Figure 4.10 : Gestion de la dynamique

La librairie OpenDE permet aussi d'ajouter des liaisons, des ressorts, etc. Nous avons principalement testé la possibilité d'utiliser des liaisons sans jeu en combinaison avec des liaisons avec jeu. La sortie récente de la puce PhysX d'Ageia devrait nous conduire à essayer cette solution qui utilise une librairie de dynamique offrant des possibilités plus vastes.

4.1.3.5 La gestion des tâches

Les trois éléments précédents doivent fonctionner comme des tâches séparées mais dans l'état actuel, cette gestion n'est pas très rigoureuse. Les 3 tâches correspondant à la boucle graphique, la boucle de dynamique et la boucle haptique sont liées entre elles par des appels de fonction non maîtrisés venant de Java3D. Les appels dans la boucle haptique sont correctement effectués, la librairie utilisée bloquant les appels extérieurs. La boucle graphique est elle aussi à la bonne fréquence puisque c'est la librairie graphique qui séquence l'affichage. Nous pensons que le découpage Java3D/C++/librairie haptique créé intrinsèquement une parallélisation des différentes tâches, ce qui expliquerait le bon fonctionnement de l'ensemble.

Voici une vue simplifiée (Fig. 4.11) de ces tâches en liaison avec l'utilisateur :

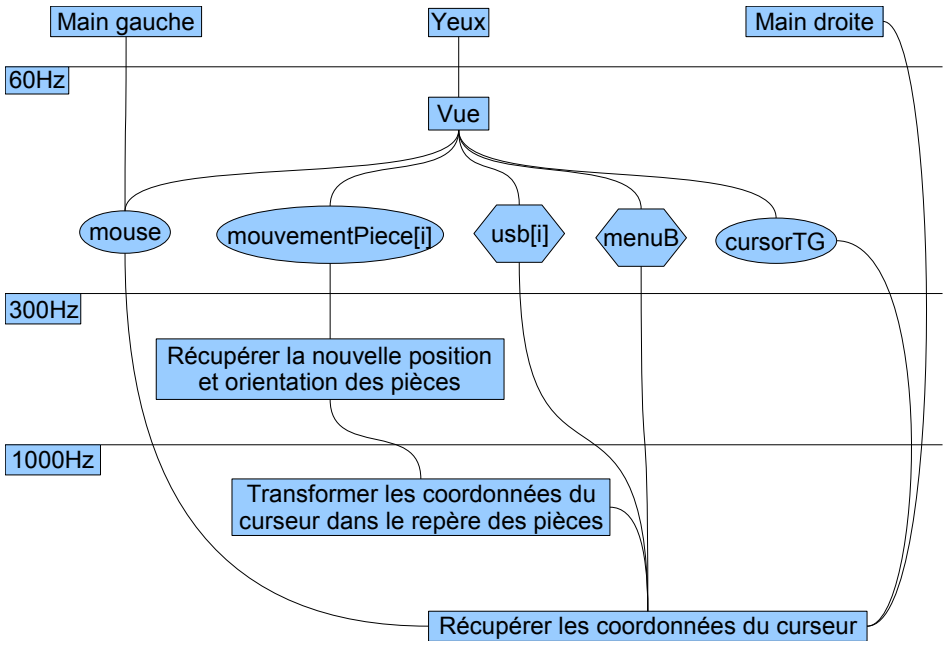


Figure 4.11 : Gestion des tâches

Dans cette première version, rien ne permet d'assurer les fréquences de chaque tâche. Les simulations sont correctes même avec le retour d'effort quand le nombre d'élements reste faible. Malgré ce défaut, nous n'avons pas constaté de pertes d'informations entre les tâches. Par souci de clarté du code, d'efficacité et d'une possible parallélisation des tâches, nous verrons que la gestion des tâches sera complètement revue dans le deuxième outil.

4.1.4 Essais et conclusions

De nombreux essais ont été réalisés lors de la mise au point de ce premier outil, nous avons, en premier lieu, cherché à montrer la possibilité de dessiner dans l'espace puis nous avons ajouté la dynamique. Nous ne montrerons ici que trois exemples significatifs : un mécanisme filaire défini à main levée avec le système à retour d'effort, un monte-charge avec ou sans jeu et un mécanisme plus complexe montrant les limites du système. Ces deux derniers systèmes sont définis à partir de fichier texte, un format ayant été défini pour faciliter la création de mécanisme et aussi pour échanger avec d'autres logiciels.

4.1.4.1 Mécanisme défini à main levée avec retour d'effort

Les mécanismes réalisés directement à partir du retour d'effort ressemblent souvent à des jeux réalisés avec du fil de fer. L'attraction du stylet sur les tracés déjà réalisés a amélioré le graphisme mais la réalisation d'un cercle dans l'espace reste difficile. Pour réaliser une liaison linéaire annulaire entre les pièces 0 et 1, on doit créer un cercle dans la pièce 1 autour du tracé existant de la pièce 0. On montre dans la figure 4.12, le processus de création de cette boucle :

- a Le stylet s'approche sans aucun effort et commence à subir à un contre-effort en frôlant le tracé d'une autre pièce

- b Le dessinateur est aidé dans son mouvement circulaire par l'équilibre qui se créé entre l'effort sur le stylet et le contre-effort. On arrive ainsi à effectuer une boucle sans apprentissage particulier.
- c La boucle étant achevée, on retombe sur le tracé de la pièce et un nouvel effort d'attraction est généré.

Figure 4.12: Création d'une boucle à l'aide du retour d'effort

Nous avons simplement inversé l'effort d'attraction précédent en un effort qui repousse le stylet, ce qui permet de tourner autour du tracé sans jamais le toucher. Avec un peu d'habitude, il devient assez naturel de créer des boucles qui nous permettent de générer différents types de liaison.

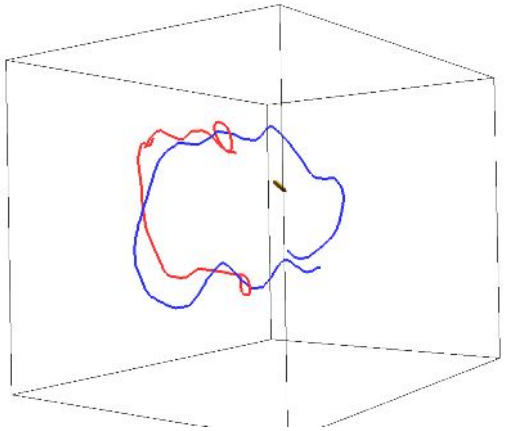


Figure 4.13 : Dessin à main levée avec retour d'effort

La figure 4.13 : une pièce fixe (0) qui forme une boucle et une pièce mobile (1) rattachée à la première par deux boucles (deux liaisons linéaires annulaires). L'animation peut ensuite être lancée avec ou sans retour d'effort. Les limitations de ce système sont évidentes mais à notre grande surprise, la prise en main et l'intérêt pour les utilisateurs sont encourageants. La facilité d'utilisation et le retour d'effort sont pour nous les raisons de ce bon résultat.

Dans ce premier outil, on peut donc simuler par exemple les casses-tête à base de fil mais notre objectif ne se limitant à ce type de système, nous essayerons de choisir une définition moins approximative de la géométrie dans le deuxième outil.

4.1.4.2 *Monte-charge avec ou sans jeu*

Ce monte-charge (Fig. 4.14) est inspiré de dessins de Léonard de Vinci. Le dessin est ici défini dans un fichier texte, il est plus géométrique et contient aussi beaucoup moins de segments. Nous avons réalisé plusieurs machines anciennes de ce type afin de pouvoir les manipuler avec le retour d'effort.

Nous avons rajouté la possibilité d'insérer des liaisons entre les pièces dans le fichier. L'avantage des liaisons est qu'elles diminuent la quantité de calcul

principalement en détection de collision. L'inconvénient est que l'on ne simule plus les jeux qui sont omniprésents dans les anciens mécanismes.

La simulation de mécanisme avec jeu et retour d'effort pose par contre de gros problèmes d'instabilité liés à l'utilisation de systèmes ressort-amortisseur pour repousser les pièces. Cette instabilité peut être analysée mais conduit à une relation entre masse raideur et coefficient d'amortissement qui n'est pas gérable dans notre cas, nous ne connaissons pas les masses des pièces à priori.

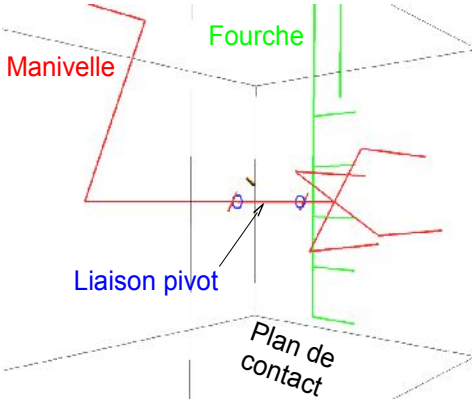


Figure 4.14 : Monte-charge sans jeu

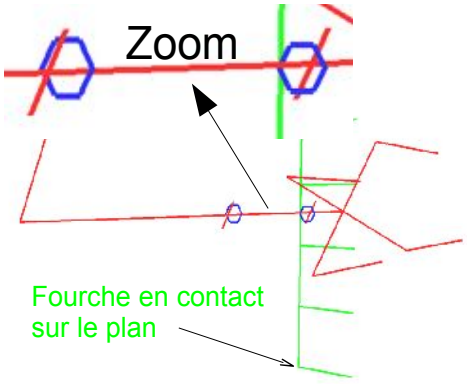


Figure 4.15 : Monte-charge avec jeu

La figure 4.14 représente le monte-charge avec liaisons, la fourche est en phase de descente et il est possible avec le retour d'effort d'actionner la manivelle sur la gauche pour remonter cette fourche. Dans la figure 4.15, on peut voir la même simulation mais avec du jeu (visible dans le zoom) dans toutes les liaisons. La position verticale de la fourche est limitée par un plan, elle rebondit sur celui-ci, imposant un mouvement inverse de la manivelle, etc.

4.1.4.3 *Mécanisme plus complexe*

Le mécanisme le plus complexe (Fig. 4.16) que nous ayons simulé avec cette première version de l'outil est un mécanisme de Peaucellier. La particularité de ce mécanisme est qu'il n'est constitué que de liaisons pivot et qu'un des axes supportant ces liaisons décrit une trajectoire rectiligne. Dans notre configuration, c'est l'axe le plus à droite qui doit décrire une trajectoire rectiligne verticale.

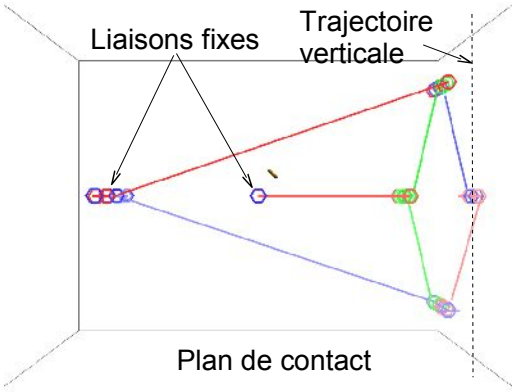


Figure 4.16 : Vue de côté en position de départ

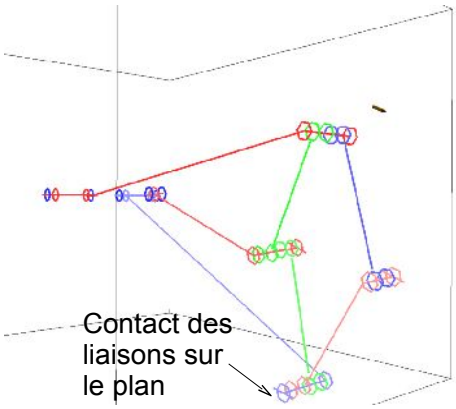


Figure 4.17 : Vue en perspective en position stabilisée

L'influence des jeux dans le fonctionnement de ce type de mécanisme est difficile à connaître, nous pouvons avec notre outil visualiser le fonctionnement du mécanisme avec les petits déplacements dus aux jeux.

Dans notre exemple, on peut vérifier que la trajectoire verticale de l'axe le plus à gauche est respectée pendant la phase de descente (Fig. 4.18). Quand l'axe du bas rentre en contact avec le plan horizontal, l'axe de gauche poursuit sa trajectoire vers la gauche en suivant approximativement un arc de cercle (Fig. 4.19). Cette vision plane reste simple, la compréhension des petits mouvements liés à la fois à la dynamique et aux jeux dans l'espace ne peut être obtenue qu'en perspective (Fig. 4.17) et en redéroulant l'animation.

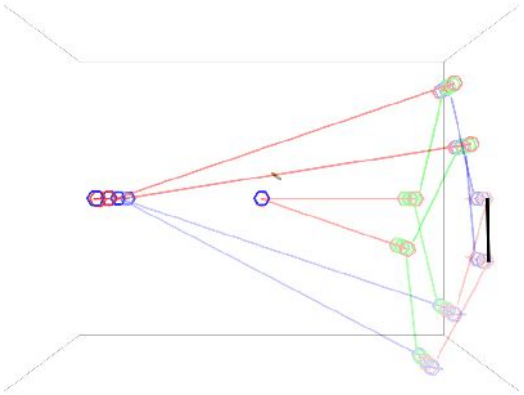


Figure 4.18 : Vérification de trajectoire verticale

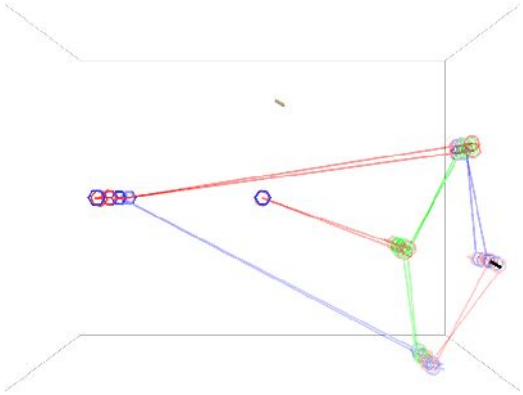


Figure 4.19 : Relaxation due au jeu

4.1.4.4 Conclusion sur le premier outil

Ce premier outil bien qu'opérationnel ne nous satisfait pas complètement mais nous a permis de découvrir les limites des choix effectués :

- instabilité de la dynamique
- collision effectuée sur des géométries discrétisées
- code mal structuré
- retour haptique géré uniquement pour les systèmes de la société Sensable
- fichiers d'entrée/sortie mal formatés
- etc.

Nous savons maintenant que la détection de collision continue avec des objets constitués d'éléments géométriques sans épaisseur est obligatoire. De même l'utilisation de ressort/amortisseur pour repousser les pièces entre elles est une source de dysfonctionnement, la notion d'intérieur ou d'extérieur n'est pas toujours connue, la dynamique devient rapidement instable. Ces deux points principaux nous ont conduit à relancer notre recherche sur la dynamique sous contraintes, la détection de collision continue devenant dans ce cas implicite.

Les autres améliorations seront présentées dans le chapitre suivant.

4.2 Deuxième version de l'outil

Ce deuxième outil va effectuer l'animation des pièces à l'aide d'une dynamique sous contraintes. Les aspects théoriques de ce deuxième outil sont largement développés dans les chapitres précédents, nous ne reviendrons ici que sur la structuration plus pratique de celui-ci ainsi que sur les essais mis en place pour valider nos choix.

4.2.1 La structure matérielle

Plusieurs points sont venus modifier notre vision de la structure matérielle de l'outil. Premièrement, la possibilité d'utiliser de nouveaux périphériques que ce soit du point de vue retour d'effort ou vision. Nous avons cherché à faciliter l'intégration matérielle et logicielle d'un nombre de périphériques plus variés.

La sortie sur le marché de l'Omni de la société Sensable à un prix nettement plus bas que les systèmes existants, nous a rassuré sur les possibilités d'utilisation de l'esquisse virtuelle par un plus grand nombre de concepteurs mais ces outils ont un espace de travail plus restreint ce qui nous a conduit à réfléchir sur l'ergonomie complète du poste de travail. Nous sommes repartis d'analyses de travail à deux mains sur des petits objets (poterie, peinture, assemblage, etc.) pour définir les positions et l'orientation des mains dans l'espace. Notre constat est clair, les postes de travail existants proposent une position trop évidente de la SpaceMouse et du retour d'effort, ces deux éléments sont juste posés à l'horizontale sur une table. On peut voir sur la figure 4.20, une vue du poste de travail dans une configuration particulière, tous les éléments (SpaceMouse et retour d'effort) peuvent s'orienter, se repositionner et nous avons choisi de ne pas occulter la vision de l'opérateur, nous laissant ainsi la possibilité d'utiliser ce poste de travail en réalité augmentée.

Figure 4.20 : Poste de travail ergonomique

4.2.2 La structure logicielle

Seule l'interface graphique en Java3D a été conservée, la structure du programme a été totalement revue avec une gestion des tâches basée sur des technologies provenant du multi-tâches [DE03], pour faciliter l'intégration de nouveaux périphériques haptiques sans avoir à gérer les API de développement propriétaires. Le passage à la détection de collision continue nous a conduit à redéfinir une librairie d'arithmétique par intervalles (Annexe 3) limitée aux fonctions utiles en dynamique mais comprenant certaines optimisations. La dynamique sous contraintes comme l'aspect théorique a pu le montrer, emploie des opérateurs peu classiques en C++ et souvent peu performants en Matlab par exemple, en particulier le calcul de la pseudo-inverse de Moore-Penrose. Nous avons aussi modifié le format de fichier de définition des mécanismes en nous appuyant sur le formalisme XML pour des raisons de fiabilité, de lisibilité et pour une meilleure réinterprétation des données dans d'autres logiciels (modélisation géométrique, calcul de mécanismes, calcul de déformation, etc.).

L'évolution rapide des outils de développement informatique nous a contraint à abandonner les outils utilisés pour le premier outil, pour la partie Java, Java3D, nous utilisons Eclipse 3.2, pour la partie C++, nous programmons avec Visual Studio 2005 et finalement la gestion des fichiers XML et la DTD correspondante s'effectue avec Cooktop 2.5.

Nous présentons ici les éléments les plus originaux des développements de cette nouvelle version.

4.2.2.1 Nouvelle architecture logicielle

Pour repartir sur de bonnes bases, nous avons développé un programme minimum qui n'avait pour objectif que de découper correctement les différentes tâches. Nous avons rajouté les contraintes suivantes : faire le lien avec une interface graphique en Java3D qui fonctionne comme une tâche indépendante et utiliser la librairie Haptik Library afin d'être indépendant du périphérique haptique. La solution technologique retenue est l'utilisation de la librairie JThread qui utilise pthread sur les plateformes de type Unix et les threads Win32 natifs sous Windows.

La librairie JThread ne contient en fait que deux classes : JThread et JMutex. JThread permet de définir la tâche elle-même exactement comme dans le langage Java. La classe JMutex (Mutex : Mutual Exclusive Object) permet de synchroniser les tâches pour assurer en particulier l'échange d'informations entre tâches. Nous pouvons représenter les différentes tâches et leurs relations ainsi :

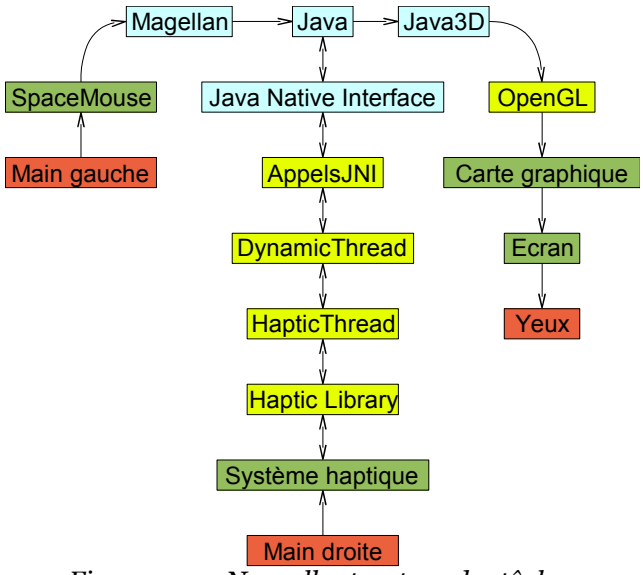


Figure 4.21 : Nouvelle structure des tâches

Le contrôle des fréquences se pose uniquement pour la tâche dynamique (DynamicThread) et pour la tâche haptique (HapticThread). La tâche haptique n'effectue qu'un lissage de l'effort, en effet celui-ci est transmis par la tâche dynamique à 300 Hz et doit ensuite être transmis à la librairie Haptik Library à 1KHz. On peut donc effectuer un simple échantillonage de la force (Fig. 4.22).

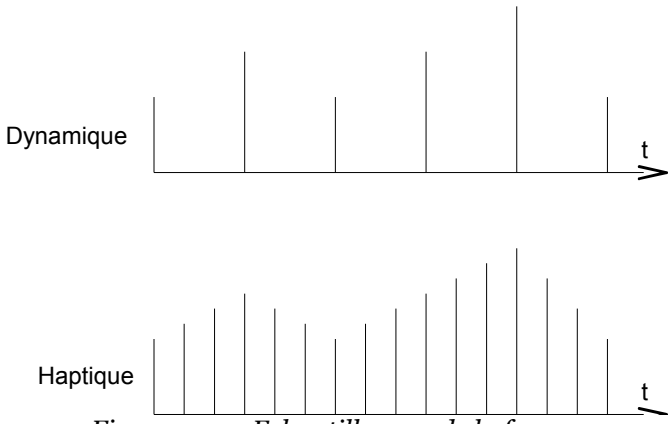


Figure 4.22 : Echantillonage de la force

La tâche dynamique concentre par contre tous les calculs de détection de collision et de dynamique et le temps de calcul à chaque pas est variable et inconnu à priori. Pour obtenir la fréquence désirée de 300 Hz pour cette tâche, la technique utilisée dans les threads consiste à endormir la tâche un court instant, cette méthode est très approximative et seule l'utilisation du temps réel permettrait d'obtenir des résultats mieux cadencés. Nous avons cependant avec cette nouvelle structure, amélioré la stabilité (par l'utilisation des Mutex) et la lisibilité du programme.

4.2.2.2 *La tâche haptique*

Dans le premier outil, deux périphériques de la même société (Sensable), ont été utilisés avec un changement de librairie. Les incompatibilités entre ces deux versions étaient telles que deux versions ont dû fonctionner en parallèle. Face à

cette difficulté, nous avons recherché des librairies haptiques de plus haut niveau et nous avons choisi Haptik Library [BA10] développé par Maurizio de Pascale à l'Université de Sienne. En effet cette librairie est la seule qui se focalise sur la possibilité d'encapsuler les différents périphériques haptiques, les autres librairies du même type gèrent aussi l'affichage, le son. On remarquera que la plus connue d'entre elles : H3D (www.h3dapi.org) intègre Haptik Library.

Figure 4.23 : Choix périphérique haptique

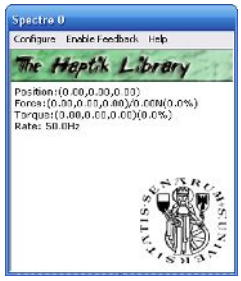


Figure 4.24 : Mouse Spectre

Une boîte de dialogue (Fig. 4.23) permettant de choisir le périphérique utilisé s'affiche au chargement du programme. Un des intérêts d'Haptik Library est la possibilité d'utiliser une simple souris pour émuler un périphérique haptique : le Mouse Spectre. Si l'on choisit le Mouse Spectre comme périphérique, le déplacement du curseur s'effectue avec la souris et une séquence de touches pour la profondeur. Si dans la position atteinte par le curseur, le système détermine qu'un effort doit être retourné, celui-ci sera simplement affiché dans la boîte de dialogue de la Mouse Spectre (Fig. 4.24). Cette possibilité est importante pour des raisons de coût mais plus particulièrement pour détecter les erreurs dans un programme utilisant un périphérique haptique.

Le deuxième outil peut donc grâce à Haptic Library fonctionner sans aucune modification avec les périphériques suivants : Phantom (Sensable), Omni (Sensable), Delta (ForceDimension), Virtuose (Haption), etc.

4.2.2.3 *La tâche dynamique*

Une vision synthétique (Fig. 4.25) de cette tâche décomposerait celle-ci en deux sous-tâches :

- détection de changement d'état
- recherche de la trajectoire

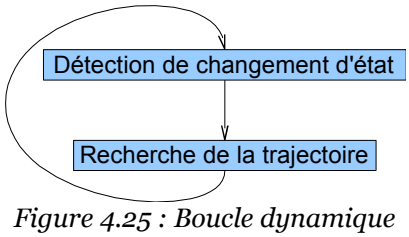


Figure 4.25 : Boucle dynamique

Une présentation plus théorique a déjà été faite au chapitre 3.1, nous nous limiterons à quelques aspects plus pratiques. La complexité de la mise en place d'une détection de collision continue peut être illustrée par un extrait de la documentation générée automatiquement par Doxygen :

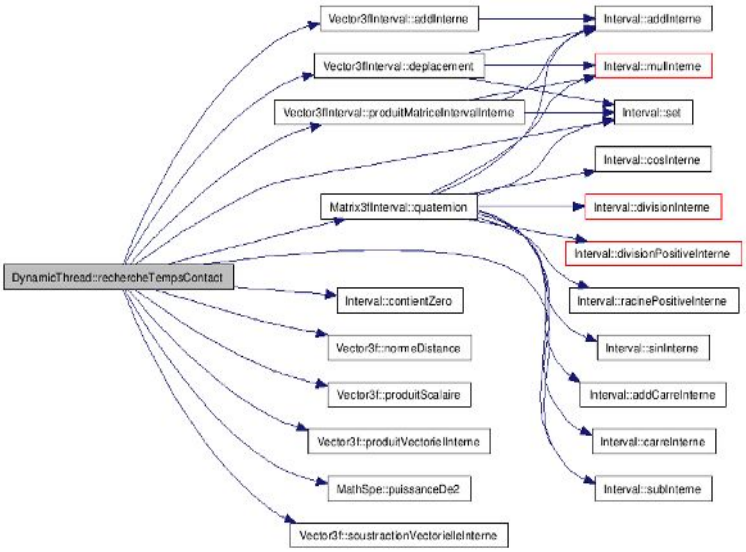


Figure 4.26 : Complexité de la détection de collision continue

Ce graphisme permet d'illustrer la complexité d'une seule des fonctions utilisées dans cette phase : la fonction rechercheTempsContact. Des classes très importantes sont appelées par la classe DynamicThread, puisque nous utilisons l'arithmétique des intervalles pour rechercher le temps et le lieu du contact, on retrouve très souvent la classe Interval. Cette classe contient toute les fonctions mathématiques de base (addition, multiplication, etc.) mais aussi les fonctions trigonométriques qui ont été optimisées en faisant des approximations quand les angles sont petits. En effet la recherche de contact s'effectue sur l'intervalle de temps [0 0.0033] (correspondant à 300 Hz), les angles parcourrus sont souvent faibles et par contre les fonctions trigonométriques sont très couteuses en CPU.

Ce type d'optimisation permet de justifier ce développement spécifique, l'utilisation d'une librairie d'arithmétique des intervalles permettrait d'atteindre une meilleure fiabilité mais au détriment des performances. De plus d'autres classes beaucoup plus complexes ont dû être créées, un vecteur d'intervalles : Vector3fInterval et une matrice 3x3 d'intervalles : Matrix3fInterval. Une des méthodes les plus complexes mise en place est par exemple le calcul de la matrice de rotation à l'aide des quaternions en arithmétique des intervalles.

Nous indiquons en annexe 3, quelques éléments de base de l'arithmétique des intervalles ainsi que certains problèmes liés à son utilisation, principalement dans le cas des hiérarchies d'intervalles.

La recherche du temps de contact est suivie d'un déplacement de cette durée permettant de se trouver dans une configuration géométrique où les pièces se trouvent en contact s'il y a choc. Dans cette position, tous les points de choc sont recherchés afin d'en déduire les contraintes à imposer à la dynamique.

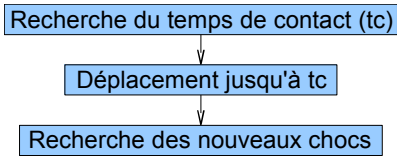


Figure 4.27 : Détection de changement d'état

Recherche de la trajectoire

La dynamique sous contraintes intervient dans cette sous-tâche et se décompose ainsi (Fig. 4.28) :

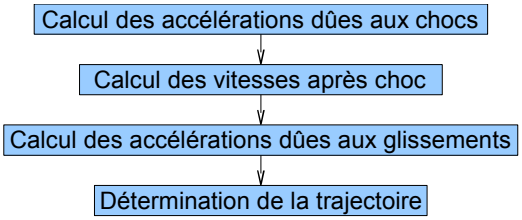


Figure 4.28 : Recherche de la trajectoire

On peut remarquer qu'il est fait appel deux fois à la dynamique sous contraintes pour la recherche des accélérations, une première fois en cas de choc pour en déduire les vitesses après choc puis une deuxième fois en cas de glissement. La détermination de la trajectoire est faite à partir des accélérations et des vitesses.

Les deux tâches : Calcul des accélérations, peuvent être aussi décomposées de manière synthétique (Fig. 4.29) :

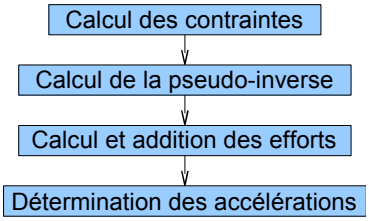


Figure 4.29 : Calcul des accélérations

Les éléments théoriques sur ces différents calculs ont été donnés dans les chapitres précédents, nous allons détailler certains éléments de ces calculs comme par exemple l'obtention de la matrice des contraintes $\boldsymbol{A}$ de l'équation 2.10 à partir des équations des contraintes de choc, de glissement ou de liaison.

Construction de la matrice des contraintes $\boldsymbol{A}$

Chaque ligne de l'équation 2.10 peut être vue comme un produit scalaire en effet :

$$A_{ij}\,\ddot{x}_j = b_i \tag{4.1}$$

Où i est le numéro de la contrainte et j le numéro de la composante du vecteur accélération. La détermination du vecteur A_{ij} permet donc de construire la matrice $\boldsymbol{A}$. Il est nécessaire de transformer les équations de

contrainte en utilisant l'égalité : $(\vec{a} \Lambda \vec{b}).\vec{c} = \vec{a}.(\vec{b} \Lambda \vec{c})$. Nous pouvons ainsi transformer le membre de gauche de l'équation 2.86 définissant les contraintes de choc en translation.

$$\begin{aligned}&(\overrightarrow{\gamma_{2O_2}} + \overrightarrow{\omega_2} \Lambda [R_{O_2}]\overrightarrow{O_2P} - \overrightarrow{\gamma_{1O_1}} - \overrightarrow{\omega_1} \Lambda [R_{O_1}]\overrightarrow{O_1P}).\vec{n} = \\ &\overrightarrow{\gamma_{2O_2}}.\vec{n} + \overrightarrow{\omega_2}.([R_{O_2}]\overrightarrow{O_2P} \Lambda \vec{n}) - \overrightarrow{\gamma_{1O_1}}.\vec{n} - \overrightarrow{\omega_1}.([R_{O_1}]\overrightarrow{O_1P} \Lambda \vec{n})\end{aligned} \quad (4.2)$$

Le vecteur $\vec{A_i}$ correspondant à la ième ligne de la matrice $\boldsymbol{A}$ est donc défini comme suit :

$$\vec{A_i} = [\underbrace{\vec{0}^t .. \vec{0}^t \ -\vec{n}^t \ -([R_{O_1}]\overrightarrow{O_1P} \Lambda \vec{n})^t}_{\text{6 composantes relatives à la pièce 1}} \ \underbrace{\vec{0}^t .. \vec{0}^t \ \vec{n}^t \ ([R_{O_2}]\overrightarrow{O_2P} \Lambda \vec{n})^t}_{\text{6 composantes relatives à la pièce 2}} \ \vec{0}^t .. \vec{0}^t] \quad (4.3)$$

Il est important de remarquer que ce vecteur ne comprend au maximum que 12 composantes non nulles ce qui rend la matrice $\boldsymbol{A}$ d'autant plus creuse que le nombre de pièces est en augmentation. Des méthodes spéciales de traitement des matrices creuses lesquelles devront être envisagées si le nombre de pièces devient important comme pour la simulation de chaîne ou de poutre flexible par exemple.

4.2.2.4 Gestion des fichiers et interfaçage

La première version de l'outil permettait déjà l'enregistrement et la relecture de fichier pour pouvoir conserver et réutiliser le travail effectué. La géométrie des tracés n'était constituée que de segments, nous avons pour objectif dans cette nouvelle version d'utiliser un plus grand nombre de type de géométrie. Notre souci est aussi de fiabiliser la lecture et l'enregistrement de ces fichiers dans l'outil lui-même mais aussi dans d'autres logiciels. Tout ceci nous a naturellement conduit à l'utilisation du format XML, un fichier de définition (mecanisme.dtd) permet d'imposer une structure unique à tous les fichiers de données (BielleManivelle.xml par exemple). On peut illustrer le lien entre dtd et xml par le graphisme ci-dessous (Fig. 4.30) :

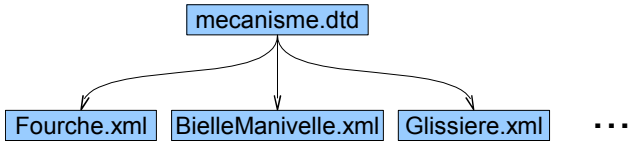


Figure 4.30 : Définition des fichiers xml

Nous donnons en Annexe 4 l'intégralité du fichier mecanisme.dtd et une partie du fichier BielleManivelle.xml.

Un exemple d'interfaçage a été mis en place avec Catia, des macros en Visual Basic nous permettent de retrouver la géométrie ainsi que les liaisons. Nous montrerons l'utilisation de cette interface dans la présentation des essais.

4.2.3 Essais

Les essais réalisés sur cette deuxième version sont surtout focalisés sur la

validation de la dynamique sous contraintes mise en place ainsi que sur le lien entre l'esquisse virtuelle et d'autres outils informatiques. Le premier essai nous a permis de tester ces deux aspects sur un mécanisme simple (bielle-manivelle) mais comprenant cependant un degré d'hyperstatisme. Le second essai va nous permettre de tester sur un système parallèle possédant des singularités, la capacité à manipuler de type de mécanisme avec le retour d'effort. Le dernier essai portera sur une mise en oeuvre complète en conception de l'esquisse virtuelle.

4.2.3.1 Système bielle manivelle

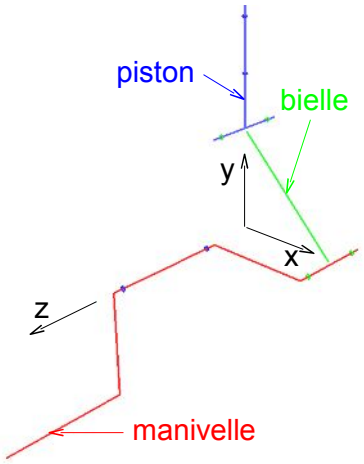


Figure 4.31 : Système bielle-manivelle

Nous avons repris ce système mécanique (Fig. 4.31) très classique que nous avions déjà simulé avec le premier outil, des comparaisons seront possibles. Les liaisons dans le système bielle manivelle seront modélisées soit par deux contacts cercle/segment (avec jeu) soit par des liaisons pivot glissant (sans jeu). Le système bielle manivelle avec 4 liaisons pivot glissant possède plusieurs degrés de mobilité et est hyperstatique. Nous cherchons à montrer que la dynamique sous contraintes gère très bien ces singularités et peut même devenir un outil d'analyse de systèmes mécaniques en mouvement.

Nous allons tester 3 configurations différentes en partant de la même géométrie :

- Configuration 1 : sans liaison, la dynamique est soumise à des contraintes de choc et de glissement.
- Configuration 2 : avec liaison, la dynamique n'est soumise qu'à des contraintes de liaison.
- Configuration 3 : avec liaison et choc, la dynamique est soumise à des contraintes de liaison et de choc. Un segment est ajouté à la géométrie de la pièce fixe afin de limiter le débattement de la bielle.

Configuration 1

Une correction a dû être apportée à la figure 4.28 pour la recherche de la trajectoire, en effet la contrainte de choc impose un rapport entre la vitesse avant

et après choc. Dans notre cas quand la bielle rentre en contact avec le bâti, deux contacts cercle/segment apparaissent comme dans la figure 4.14. Il est facile de montrer que dans ce cas l'axe de la bielle reste à l'horizontale après choc, voir Tableau 14 ci-dessous (le point plus gros sur l'axe en mouvement représente le centre de gravité) :

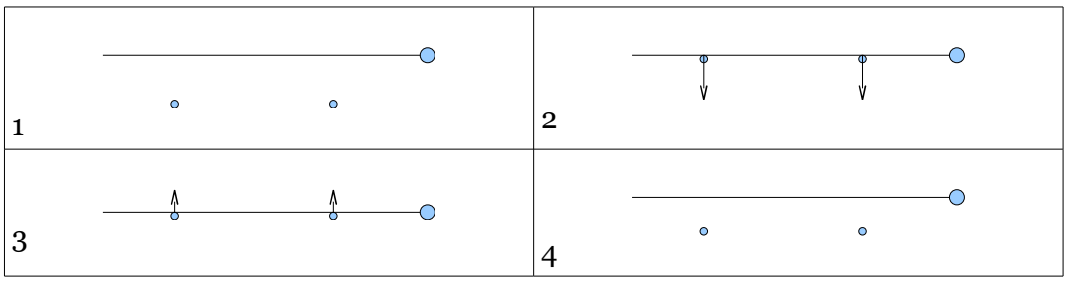


Tableau 14 : Mauvaise gestion du double choc simultané

La position recherchée après choc étant la suivante :

Figure 4.32 : Position à atteindre après un double choc simultané

On comprend que les contraintes de choc ne doivent être appliquées que si l'effort au point de contact est dirigée dans le bon sens, ce qui implique un traitement en deux étapes :

- une première étape au cours de laquelle, toutes les contraintes de choc sont appliquées ce qui permet de déterminer les efforts et de supprimer les contraintes inutiles.
- Une deuxième étape dans laquelle on effectue à nouveau les calculs avec les contacts restants.

La nouvelle recherche de trajectoire devient donc :

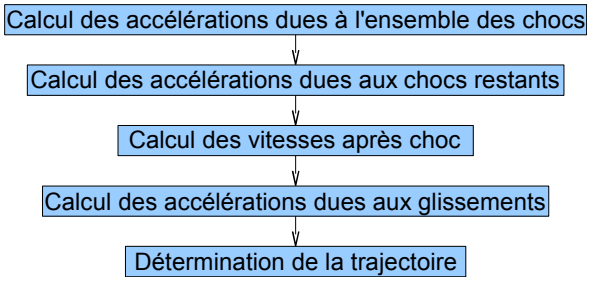


Figure 4.33 : Nouvelle recherche de trajectoire

Nous allons maintenant montrer l'aspect très chaotique des mouvements quand un jeu existe entre les pièces. Nous avons effectué pour cela des tests avec différents rayons de cercle 0.002 m, 0.001 m, 0.0005 m, 0.00025 m soit des jeux respectivement de 4 mm, 2 mm, 1 mm et 0.5 mm. Pour un sketcher, ces dimensions sont difficilement utilisables mais nous avons voulu déterminer les limites du modèle choisi et montrer l'influence du jeu sur le comportement du mécanisme. Ces tests sont réalisés sans retour d'effort, seule la gravité est à l'origine du mouvement des pièces.

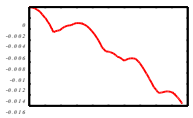
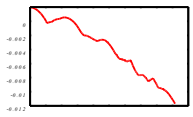
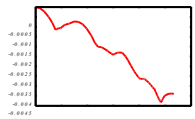
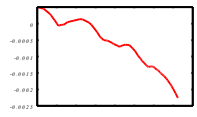
0.002	0.001	0.0005	0.00025
			

Tableau 15 : Allure du déplacement du piston en fonction du jeu

Le tableau ci-dessus représente la descente du piston (translation de la pièce 3 sur y) pour les différents jeux en fonction du temps. Les échelles de temps ne sont pas les mêmes, on peut cependant conclure que ces 4 courbes sont très semblables. Toutes les courbes sont disponibles en dimension plus lisible en Annexe 5.1.

La similarité précédente nous permet de nous focaliser uniquement sur le mécanisme avec un rayon de 0.001.

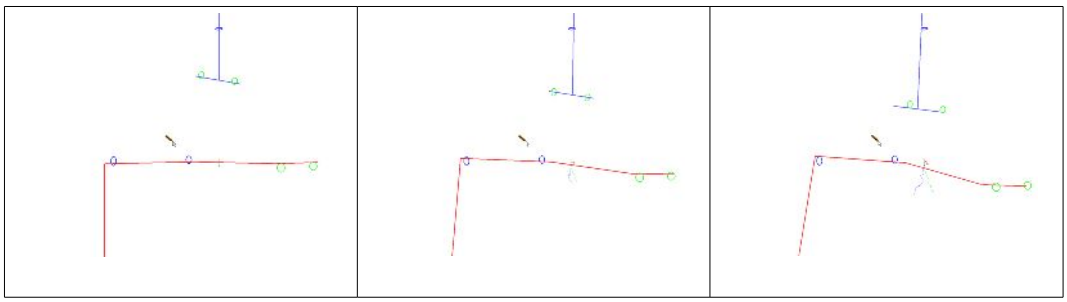


Tableau 16 : Evolution du système bielle manivelle avec jeu au cours du temps

Les trois vues d'écran ci-dessus montrent l'évolution du mécanisme au cours du temps. La première vue sur la gauche est prise juste après le départ, les pièces n'ont pas encore subi de rotation et les doubles contacts dans les liaisons pivot sont nombreuses. Dans la deuxième vue, au centre, la manivelle s'est inclinée ce qui va conduire au glissement de la bielle vers la droite. On aboutit dans la vue de droite à la perte de contact.

Cet exemple simple nous a permis de vérifier que nous avons conservé l'aspect réaliste de la simulation comme nous l'avions dans la première version de l'outil. Tous les contacts sont gérés, les chocs, les glissements ainsi que les pertes de contact. Nous avons cependant cette fois augmenté la validité des résultats en maîtrisant le lieu des contacts en remplaçant les ressorts et amortisseurs par des chocs et glissements.

Configuration 2

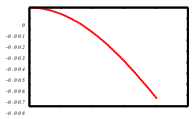


Figure 4.34 : Allure du déplacement du piston sur un temps court

Dans cette configuration, nous avons simplement remplacé les chocs et

glissements dans les liaisons pivots par des liaisons pivot glissant comme indiqué au chapitre 2.2.6. Les mouvements sont beaucoup plus continus comme l'indique la figure ci-dessous, équivalente aux courbes présentées pour la configuration 1.

La qualité de la simulation peut être montrée en laissant la simulation se dérouler et en vérifiant par exemple que les points extrêmes sont toujours atteints. Nous vous présentons ce type de courbe identique à celle ci-dessus pour seulement deux cycles (Fig. 4.36). La figure 4.35 n'est en fait qu'un très gros zoom du point de départ de la courbe (point en haut à gauche) de la figure 4.36.

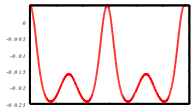


Figure 4.36 : Allure du déplacement du piston sur deux cycles

La stabilité numérique de l'outil est même plus facile à obtenir quand le mécanisme n'est constitué que de liaisons. Par contre cette configuration nous a permis de vérifier un des avantages de la formulation d' Udwadia et Kalaba, en effet, le mécanisme bielle manivelle est mal défini (au sens de la théorie des mécanismes). Il possède des mobilités internes et il est hyperstatique d'ordre 1. Nous allons montrer comment la formulation d'Udwadia et Kalaba contourne ces difficultés.

Si le mécanisme ne contient que des liaisons, les calculs matriciels situés en figure 4.37 sont réalisés, ceux-ci sont présentés en détail en Annexe 6.1.

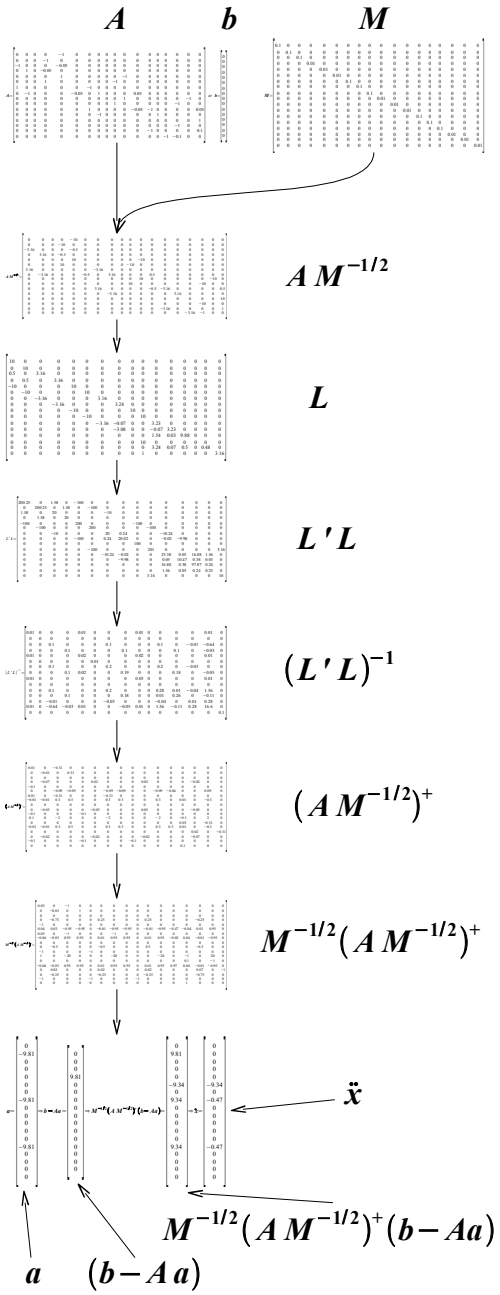


Figure 4.37 : Déroulement des calculs matriciels

La première remarque concerne la matrice A. En effet cette matrice possède deux colonnes vides qui correspondent à deux degrés de mobilité :

- la colonne 3 correspondant à la translation sur z de la manivelle
- la colonne 9 correspondant à la translation sur z de la bielle

Le degré d'hyperstatisme ainsi que le mouvement contraint sont déterminés par la simple recherche de colonnes vides dans la matrice L utilisée lors du calcul de la pseudo-inverse. Dans notre cas la 14ème colonne est vide, ce qui correspond à une contrainte en rotation sur x pour le piston ($-\dot{\omega}_{3/0x}=0$). Pour retrouver cet hyperstatisme, il faut revenir aux contraintes (Fig. 4.38) et chercher à comprendre pourquoi $\dot{\omega}_{3/0z}$ est contraint. En remontant dans la matrice à partir de la colonne de $\dot{\omega}_{3/0z}$, on trouve très rapidement que cette variable

s'annule deux fois ce qui prouve la surdétermination de sa valeur.

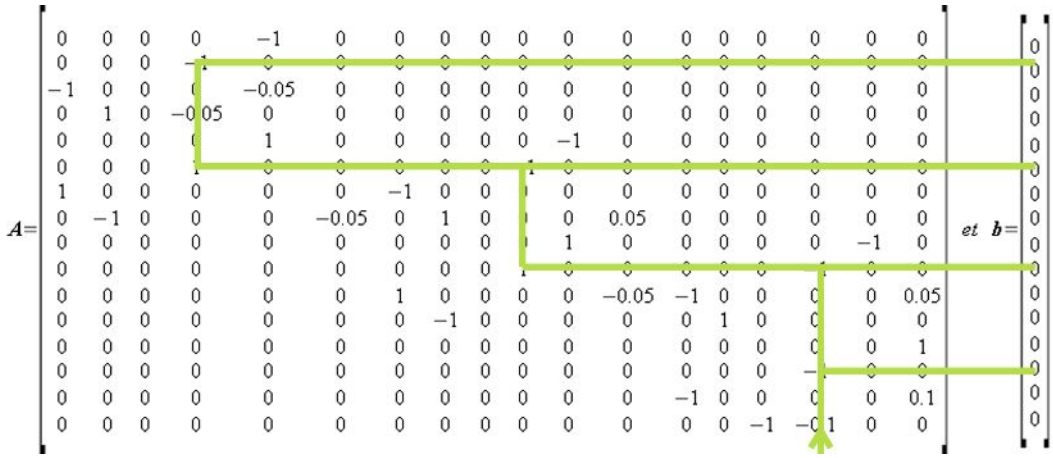


Figure 4.38: Recherche de l'origine de l'hyperstatisme

Ces éléments sont déterminés à chaque pas de temps et il est donc possible de les visualiser directement sur le mécanisme lors de la manipulation de celui-ci.

Les efforts dans les liaisons peuvent, eux aussi, être déterminés et visualisés juste avant l'obtention des accélérations contraintes.

Configuration 3

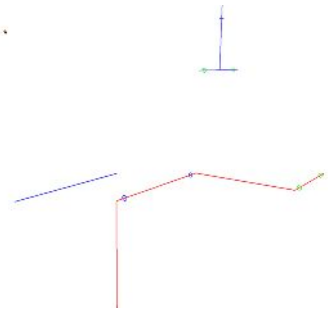


Figure 4.39 : Système bielle manivelle avec choc

Nous avons souhaité dans cette dernière configuration (Fig. 4.39), ajouter une contrainte de choc à nos contraintes de liaison. Nous montrons en Annexe 6.2 que la formulation d'Udwadia et de Kalaba permet une gestion parfaite de ce couplage.

Nous allons montrer l'évolution de la vitesse de rotation de la manivelle en fonction du coefficient d'absorption e . La courbe ci-dessous correspond à l'évolution de la vitesse de rotation de la manivelle dans la configuration 2 (sans choc).

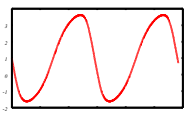


Figure 4.40 : Evolution de la vitesse de rotation de la manivelle

Nous allons comparer cette courbe avec celles avec choc.

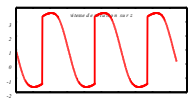
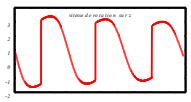
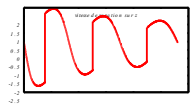
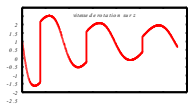
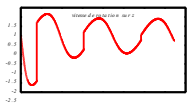
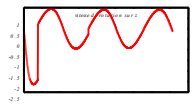
$e = 1.0$	$e = 0.9$	$e = 0.7$
		
$e = 0.5$	$e = 0.3$	$e = 0.1$
		

Tableau 17 : Evolution de la vitesse de rotation de la manivelle en fonction du coefficient d'absorption

Toutes ces courbes se retrouvent en Annexe 5.3. Le saut de vitesse en cas de choc est facile à repérer sur ces courbes. On peut remarquer qu'il n'y a disparition du choc que dans la dernière courbe ($e = 0.1$).

Conclusion sur le système bielle manivelle

Dans cet exemple, nous avons montré que nous pouvions grâce à la dynamique sous contraintes, manipuler des mécanismes soumis à des contraintes diverses avec le même formalisme. Cette unification de la dynamique et de la gestion de contrainte peut donc servir de base à un outil d'esquisse virtuelle.

4.2.3.2 Manipulation haptique d'un système parallèle

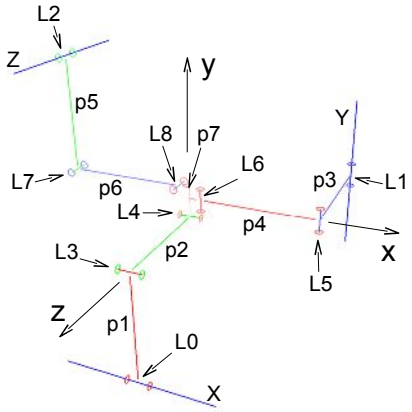


Figure 4.41 : Système parallèle

Notre exemple test part d'un nouveau concept de machine parallèle, nous avons donc cherché à tester un cas de mécanisme parallèle simple. Le système choisi est proche du système T3R1 développé au LAMI mais ne permet que la translation de l'effecteur fixé à la pièce 7. Cette limitation de mouvement nous permettra de manipuler le système avec nos systèmes à retour d'effort qui ne peuvent exercer

de couples.

Analyse du système parallèle

Ce système est constitué de huit pièces, de trois liaisons pivot glissant et de six liaisons pivot ce qui nous donne une matrice $\boldsymbol{A}$ de 42 lignes et 42 colonnes.

Dans la position de départ (celle de la figure 4.41), il n'y a aucune colonne vide, donc pas de mobilité interne dans ce mécanisme. L'hyperstatisme du mécanisme est visible lors du calcul de la matrice $\boldsymbol{L}$ avec l'apparition de trois colonnes nulles. Dans ce cas l'analyse de l'hyperstatisme est moins directe que pour le système bielle manivelle, les trois colonnes vides correspondantes aux contraintes suivantes :

$$-\gamma_{3/0x}+0.15\,\dot{\omega}_{3/0y}+0.02\,\dot{\omega}_{3/0z}=0$$
$$-\gamma_{3/0z}+0.02\,\dot{\omega}_{3/0x}+0.16\,\dot{\omega}_{3/0y}=0$$
$$-\dot{\omega}_{5/0y}=0$$

La troisième contrainte est située dans la liaison pivot glissant L2 et correspond à un blocage en rotation suivant y . Les deux premières équations sont plus difficiles à comprendre. $\dot{\omega}_{3/0y}$ est une variable d'entrée du mécanisme et ne pose donc aucun problème. Dans les 4 variables restantes, seules $\dot{\omega}_{3/0x}$ et $\dot{\omega}_{3/0z}$ sont directement déterminées par deux autres contraintes, ce sont donc ces deux contraintes qu'il faut relâcher si l'on veut supprimer l'hyperstatisme.

Comportement avec une accélération sur l'effecteur en x

Afin de vérifier le bon fonctionnement du système parallèle, nous avons soumis l'effecteur du mécanisme à une accélération sur l'axe x ce qui correspond à l'application d'une force constante suivant x et nous allons vérifier que l'évolution correspondait bien à celle espérée. Les vues suivantes (vue de face et vue de dessus) sont données dans la position de départ et dans la position finale du mouvement enregistré.

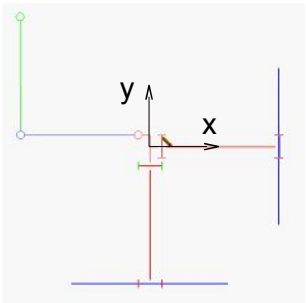


Figure 4.42 : Système en vue de face au départ

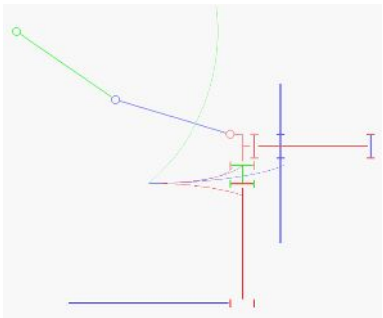


Figure 4.43 : Système den vue de face an fin de mouvement

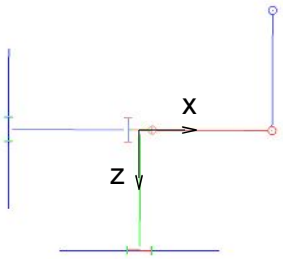


Figure 4.44 : Système en vue de dessus au départ

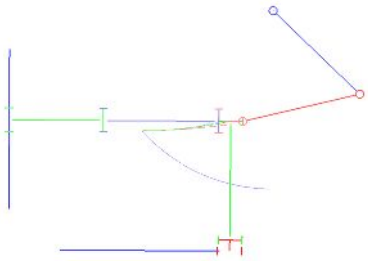


Figure 4.45 : Système en vue de dessus en fin de mouvement

On remarque que les mouvements s'effectuent comme prévu et la pièce 7 est bien restée parallèle à elle-même. La validation du comportement nous permet de nous assurer une manipulation haptique sans comportement aberrant du mécanisme.

Comportement avec une accélération sur la pièce motrice 1 en x

Nous cherchons maintenant avec la même accélération que précédemment, à vérifier le comportement du système dans le cas où l'effort est placé au niveau de la liaison L0 sur la pièce 1. Le comportement est exactement le même que dans le cas précédent, ce qui est logique. Avec les trois courbes ci-dessous, nous montrons deux choses :

- le déplacement de l'effecteur reste identique que l'effort soit placé sur celui-ci ou sur la pièce 1 (Comparaison de la courbe 4.46 et de la courbe 4.47). Les caractéristiques inertielles étant identiques
- le déplacement est aussi identique si l'on applique la force sur la pièce 1 entre celui de l'effecteur et celui de la pièce 1 sur l'axe x (comparaison de la courbe 4.47 et de la courbe 4.48). Cette propriété évidente d'homocinétisme se retrouve suivant les 3 axes en translation. Cette propriété très intéressante pour ce type de système parallèle peut être mise à profit dans des systèmes haptiques.

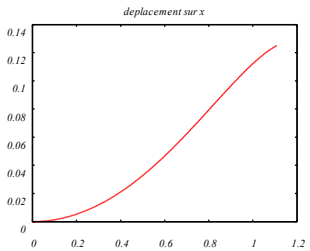


Figure 4.46 : Déplacement sur x de l'effecteur avec une force sur celui-ci

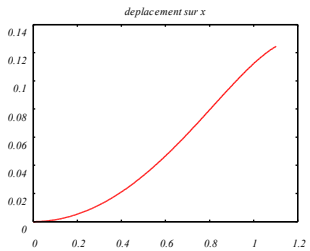


Figure 4.47 : Déplacement sur x de l'effecteur avec une force sur la pièce 1

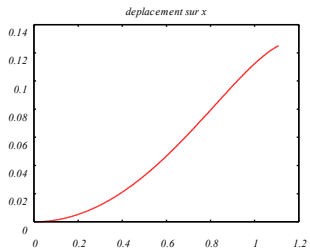


Figure 4.48 : Déplacement sur x de la pièce 1 avec une force sur la pièce 1

Exemple de modification de la longueur d'une pièce

Nous allons montrer comment la dynamique sous contraintes peut être utilisée

pour la modification du mécanisme. Nous voulons par exemple augmenter la longueur de la pièce 1, pour cela, on coupe cette pièce virtuellement en son milieu. Nous nous retrouvons avec les deux pièces 1 et 8 que nous relions par une liaison glissière. La modification de la longueur est simplement réalisée en déplaçant la pièce 8 avec le système haptique. Nous illustrons sur les figures ci-dessous l'évolution du système de la position de départ (Fig. 4.49), en passant par une position intermédiaire (Fig. 4.50) puis la position finale recherchée (Fig. 4.51).

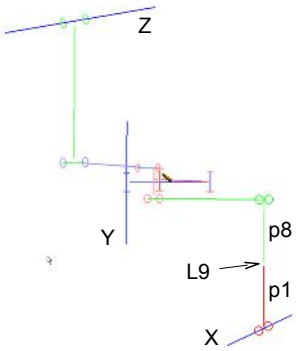


Figure 4.49 : Système parallèle avant déplacement

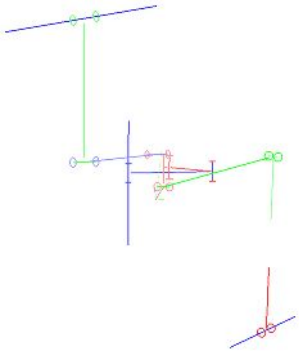


Figure 4.50 : Position intermédiaire du système parallèle

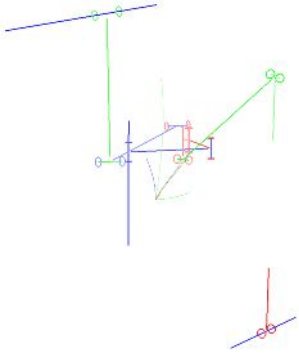


Figure 4.51 : Position recherchée du système parallèle

Dans cet exemple, toutes les liaisons sont libres, il est facile d'imaginer que l'on peut bloquer certains degrés de liberté en fonction des déplacements souhaités.

Cet exemple simple montre une nouvelle fois l'intérêt de l'unification de la dynamique et de la gestion de contraintes.

Premier essai d'interfaçage avec Catia

Nous avons développé en VisualBasic une macro dans Catia, nous permettant de retrouver tous les éléments contenus dans le fichier XML (Fig. 4.52). Les liaisons sont elles aussi transmises ce qui donne la possibilité d'animer le mécanisme. On retrouve la même hiérarchie pour les pièces ce qui nous donnera la possibilité d'habiller les pièces tout en continuant à vérifier les collisions directement dans Catia. La modélisation basée sur les squelettes peut ainsi être poursuivie et la méthode de conception d'Ullman peut enfin être appliquée.

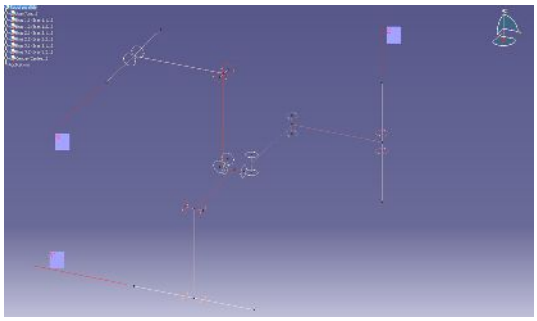


Figure 4.52 : Premier essai d'interfaçage avec Catia

Les géométries transférées (Fig. 4.53) peuvent ensuite être habillées avec des surfaces (Fig. 4.54).

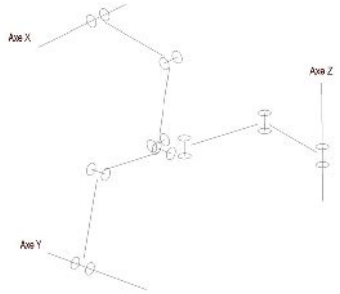


Figure 4.53 : Squelette seul

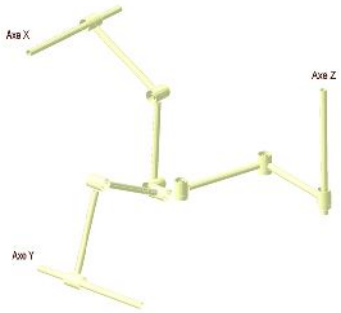


Figure 4.54 : Squelette avec surfaces

Il reste à trouver des techniques permettant l'habillage automatique ou semi-automatique des squelettes de pièces. Comme nous l'avons déjà indiqué, nous réfléchissons encore à la possibilité d'utiliser pour cela soit par exemple le module KBE de Catia soit d'intégrer directement un module intelligent directement dans l'outil d'esquisse.

Conclusion sur la manipulation haptique d'un système parallèle

Cet exemple a permis de prouver que malgré le nombre important de pièces et de contraintes, la simulation du système est possible, la stabilité de la simulation reste bonne. Des optimisations dans les calculs sont encore possibles comme nous l'avons signalé lors de la description de la tâche dynamique au chapitre 4.2.2.3.

Avec les deux exemples précédents, nous avons validé les points les plus importants de l'outil d'esquisse virtuelle : dynamique avec jeu, dynamique avec liaison (avec ou sans choc), modification de l'esquisse et interfaçage avec un autre logiciel. Nous allons maintenant dans le dernier test, montrer comment nous envisageons l'utilisation de l'outil d'esquisse virtuelle au cours de la conception d'un système mécanique.

4.2.3.3 Exemple de conception avec l'esquisse virtuelle

Nous allons maintenant montrer comment notre outil d'esquisse virtuelle peut être utilisé en conception de systèmes mécaniques. L'exemple choisi est celui de l'Exechon structure d'une machine parallèle dont la licence est actuellement achetée par plusieurs sociétés de construction de machine-outil dont PCI à Saint-Etienne. Une société chinoise propose aussi une nouvelle machine avec la licence Exechon, la Links-Exe700.

Figure 4.55 : Machine parallèle chinoise Links-Exe700 sous licence Exechon

La conception d'une nouvelle machine utilisant la licence Exechon pose plusieurs problèmes :

- Une détection rapide des hyperstatismes inhérents à la structure choisie. Il n'est absolument pas évident de déterminer le degré d'hyperstatisme de ce type de mécanisme et cette connaissance est importante pour déterminer la précision des usinages nécessaires pour l'assemblage.
- La vérification des débattements possibles de la structure elle-même mais aussi de celle-ci dans l'environnement global de la machine.
- L'influence des caractéristiques inertielles de chaque pièce sur la dynamique globale de la machine que ce soit pour une manipulation interactive en haptique ou pour une simulation plus fine du comportement.
- L'optimisation des dimensions des pièces ou de la position des liaisons par rapport à la tâche à accomplir ce qui nécessite de pouvoir modifier rapidement la géométrie.
- L'équilibrage statique de l'ensemble en déplaçant les centres de gravité interactivement jusqu'à la stabilisation des mouvements.

Pour répondre à ces différents points, nous nous sommes placés à la place du concepteur de l'Exechon lui-même. La structure cinématique de la machine est imposée, la position et l'orientation des liaisons sont définies par des contraintes simples mais dans l'espace (Fig. 4.56). Nous avons choisi une numérotation des différentes pièces et liaisons que nous avons utilisée dans notre fichier XML (Fig. 4.57). Les carrés représentent les pièces, avec la pièce 0 fixe et la pièce 4 l'effecteur.

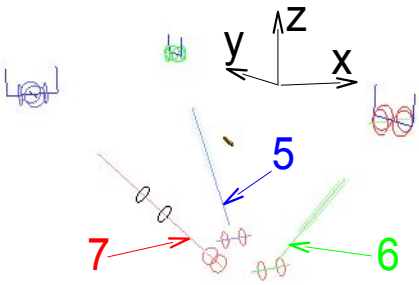


Figure 4.56 : Squelette de l'Exechon

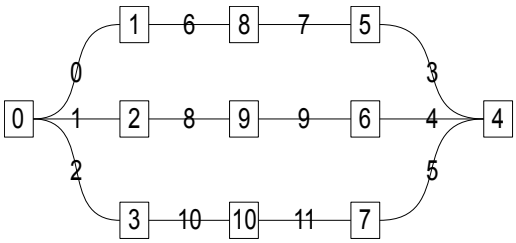


Figure 4.57 : Agencement des pièces et des liaisons

Voici la liste des liaisons présentes dans cette structure :

- liaison pivot : {0, 1, 2, 3, 4, 5, 6, 8, 10}
- liaison glissière : {7, 9}
- liaison pivot glissant : {11}

Soit un total de 9*1 + 2* 1 + 1* 2 = 13 degrés de liberté.

Ce mécanisme contenant deux boucles fermées, on obtient 2*6 équations cinématiques ce qui en final nous donne 13 - 12 = 1 degré de mobilité.

Ce calcul simple n'est valable cependant que si les 12 équations cinématiques ne sont pas dégénérées ce qui malheureusement est le cas. Un spécialiste de la théorie des mécanismes peut le démonter en utilisant par exemple la théorie des faisceaux de droites. Notre outil d'esquisse détecte à chaque pas de temps le rang de la matrice des contraintes ce qui permet de déterminer automatiquement le degré d'hyperstatisme du mécanisme à l'instant t . La matrice des contraintes dans notre exemple est une matrice 9*5 + 2*5 + 1*4 sur (11 – 1)*6 soit une matrice 59 sur 60.

Le rang de cette matrice n'est cependant que de 57 soit deux degrés d'hyperstatisme. L'outil d'esquisse virtuelle nous indique en plus que les deux contraintes qui posent problème sont :

- la contrainte d'arrêt en rotation sur l'axe $\vec{y}$ dans la liaison 9 et
- la contrainte d'arrêt en translation sur l'axe $\vec{x}$ dans la liaison 9

Ces deux degrés d'hyperstatisme permettent, par contre, de faire passer le nombre de mobilité de un à trois ce qui correspond aux trois mouvements de translation pilotés.

Des solutions simples permettraient de lever ces hyperstatismes mais réduiraient d'autant la rigidité de l'ensemble. La première contrainte peut par exemple disparaître en remplaçant la liaison glissière par une liaison pivot glissant. Après essai dans l'outil, on peut confirmer que le comportement dynamique est le même. Pour valider ce type de transformation, il faudrait comparer la rigidité des deux structures.

La deuxième contrainte est difficile à corriger dans la liaison 9 mais peut, par contre, l'être facilement dans une des quatre liaisons pivots (3, 4, 6, 8) de même axe $\vec{x}$ constituant la boucle des deux bras symétriques (Fig. 4.58).

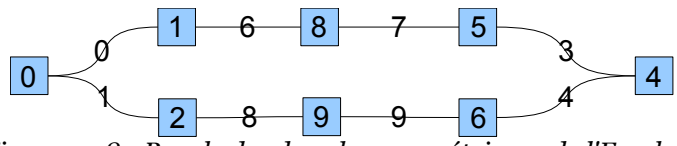


Figure 4.58 : Boucle des deux bras symétriques de l'Exechon

Une de ces 4 liaisons pivots peut être transformée en liaison pivot glissant. On peut aussi technologiquement prévoir un réglage des arrêts en translation dans une de ces liaisons.

Nous avons pu vérifier que l'outil d'esquisse fonctionne malgré l'hyperstatisme du système et qu'il permet de connaître les points de blocages à chaque pas de

temps. Nous pensons indiquer graphiquement la présence de ces blocages sur les contraintes correspondantes, facilitant ainsi la détection et les possibilités de modification.

Débattements possibles de la structure

Nous n'avons simulé que la partie comportant le système parallèle de l'Exechon. La manipulation des deux autres liaisons pivot permettant d'atteindre l'outil terminal nécessite de coupler la rotation du stylet du système à retour d'effort avec l'orientation de l'outil. Nous n'avons pas encore réalisé ce couplage, de plus nous nous attachons principalement à montrer les avantages et les inconvénients de la structure parallèle ce qui justifie de n'étudier que le débattement de la plateforme 4.

Nous avons deux possibilités pour étudier ces débattements, soit en utilisant le retour d'effort en manipulant la plateforme 4, soit en déterminant les mouvements possibles avec une seule entrée libre et en imposant un effort dans la direction du mouvement libre. Pour étudier ces débattements, il faut bloquer deux des liaisons possédant des mouvements de translation et relever le débattement obtenu.

Nous présenterons deux résultats sur les débattements du système parallèle, un avec la plateforme soumise à une force sur $\vec{z}$ et l'autre avec un moment d'axe $\vec{x}$ toujours sur la plateforme. Ce dernier résultat permet de simuler une manipulation haptique avec retour en couple.

La figure 4.59 permet simplement de valider la qualité visuelle des mouvements. Les figures 4.60 et 4.61 sont à mettre en parallèle pour comprendre par exemple que la plateforme pendant son mouvement sur $\vec{z}$ est aussi soumise à une translation sur $\vec{x}$. On peut aussi remarquer un effet dû à la dynamique de l'ensemble des pièces, la vitesse sur $\vec{z}$ subit une baisse de son accélération au cours du temps.

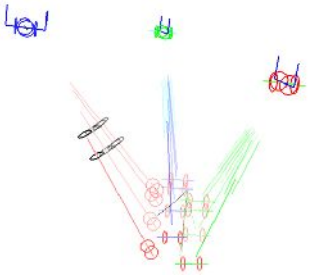


Figure 4.59 : Débattement du système parallèle soumis à une force sur z

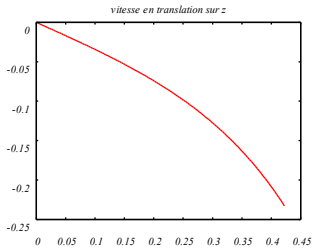


Figure 4.60 : Allure de la vitesse en translation sur z de la plateforme

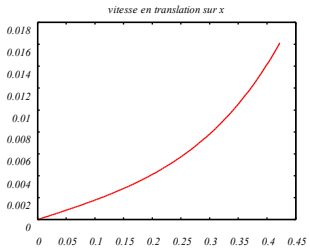


Figure 4.61 : Allure de la vitesse en translation sur x de la plateforme

La figure 4.62 donne une idée du mouvement engendré par le moment suivant $\vec{x}$ sur la plateforme. Le mouvement est stoppé par détection de collision dans un des bras. La figure 4.63 montre un résultat inattendu mais sûrement explicable, qui est la linéarité de la vitesse par rapport au temps ce qui signifie une accélération en rotation constante suivant $\vec{x}$. La figure 4.64 est moins évidente et montre que sous l'effet de ce moment, la plateforme se déplace légèrement en translation sur $\vec{x}$, ce résultat est à peine perceptible à la simple visualisation des mouvements.

Figure 4.62 : Débattement du système parallèle soumis à un moment sur x

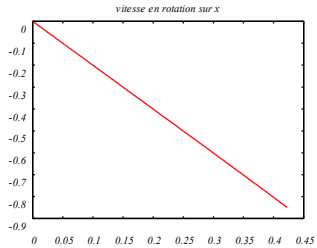


Figure 4.63 : Allure de la vitesse de rotation sur x de la plateforme

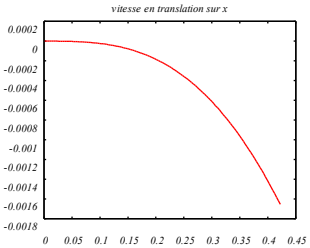


Figure 4.64 : Allure de la vitesse en translation sur x de la plateforme

Modification des caractéristiques inertielles

Plusieurs modes de modifications sont possibles :

- modification interactive de la position du centre de gravité.
- modification des caractéristiques inertielles dans le fichier XML.
- modification automatique pour l'équilibrage statique.

D'autres types de modification peuvent être envisagés mais nous ne présentons que les plus utiles. La modification interactive nécessite simplement que le centre de gravité soit représenté par un point et que celui-ci puisse être manipulé par le système à retour d'effort.

La modification du fichier XML ne présente en soit aucune difficulté et permet quand la liaison bidirectionnelle est établie avec un outil de CAO de récupérer les caractéristiques inertielles au fur à mesure de la définition du produit. Si la liaison bidirectionnelle n'existe pas, les données doivent être modifiées manuellement dans le fichier. Pour chaque pièce, les caractéristiques sont les suivantes :

```
<infoInertie>
        <masse>0.1</masse>
        <cdgX>0.0</cdgX>
        <cdgY>-0.1</cdgY>
        <cdgZ>0.08</cdgZ>
        <inertieX>0.01</inertieX>
        <inertieY>0.01</inertieY>
        <inertieZ>0.01</inertieZ>
        <alphaInertie>0.0</alphaInertie>
        <betaInertie>0.0</betaInertie>
        <gammaInertie>0.0</gammaInertie>
</infoInertie>
```

Dans cet exemple, nous avons modifié la position du centre de gravité du croisillon d'une des trois liaisons cardan afin de le placer à l'intersection des deux liaisons pivot ce qui est technologiquement plus logique.

La modification automatique n'est intéressante que si des contraintes supplémentaires sur la position du centre de gravité dans le repère des pièces. Par exemple, les trois bras ont des formes allongées, le centre de gravité peut donc être paramétré avec un seul paramètre sur cet axe. La recherche de l'équilibre statique est obtenue à partir de l'équation 2.16 en rendant nulle les accélérations contraintes d'où :

$$(AM^{-1/2})^{+}(b-Aa)=M^{1/2}a \qquad (4.4)$$

Cette recherche peut être effectuée par plusieurs techniques : algorithme génétique, CSP, etc.

Modification des dimensions pour une tâche précise

Comme nous l'avons montré dans l'exemple précédent les modifications sont possibles interactivement. En fonction des mouvements recherchés pour la plateforme, on peut modifier les dimensions jusqu'à obtention de la bonne configuration. Il est aussi possible d'obtenir de nouvelles configurations soit en modifiant directement le fichier soit en modifiant celui-ci directement dans un outil de CAO. Nous avons par exemple fait un essai consistant à changer l'écartement entre les trois bras du système parallèle, le tableau 18 présente trois écartements différents.

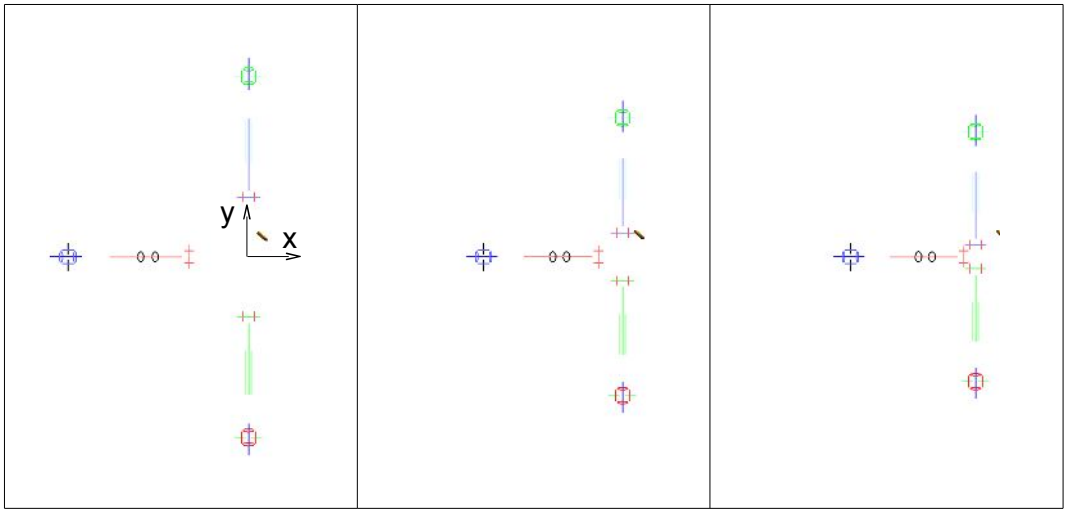


Tableau 18 : Variation de l'écartement des bras du système parallèle

A partir de l'étude de ces variations de dimensions, on peut pour des changements importants de dimensions en déduire l'optimisation globale du mécanisme mais ce type d'étude peut aussi être effectué sur des variations plus faibles proches des tolérances de fabrication et ainsi comprendre l'impact de celles-ci sur le comportement du système.

Nous nous limiterons à l'équivalent de la courbe 4.63 mais pour une plateforme soumise à un moment $\vec{x}$ sur et une force sur $\vec{y}$. L'allure des trois courbes de la vitesse en rotation de la plateforme reste identique (voir tableau 19).

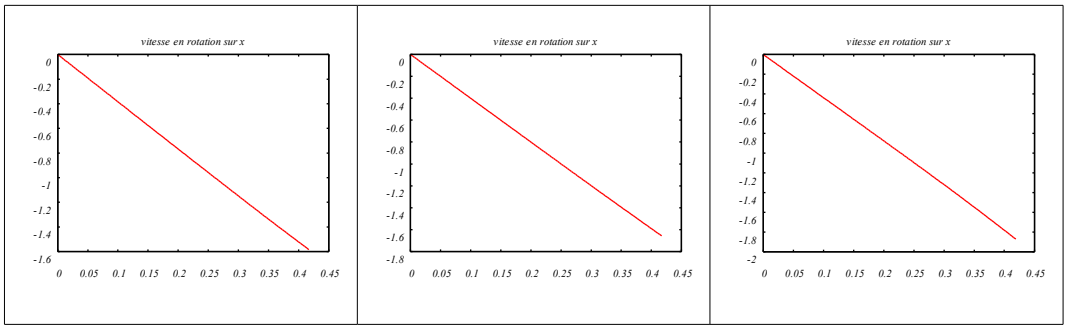


Tableau 19 : Comparaison de l'allure des courbes de la vitesse de rotation de la plateforme

On peut tirer des trois courbes précédentes, l'évolution de la vitesse à 0.4 s : -1.5, -1.6, -1.8 pour des écartements respectifs de 0.05, 0.02, 0.01. On peut donc déduire de ce résultat que plus l'écartement sera faible, plus la machine sera rapide pour la rotation sur $\vec{x}$. Ce résultat logique a une contrepartie qui est l'augmentation des efforts dans les liaisons, justifiant ainsi que les systèmes parallèles qui possèdent des bras très longs par rapport à l'écartement, ont aussi des caractéristiques inertielles très faibles. La figure 4.65 représente un manipulateur cylindrique développé dans le département automatique de l'ONERA/CERT et breveté en 1994. On peut remarquer la faible rigidité de ce type de structure contrairement aux structures des machines parallèles pour l'usinage.

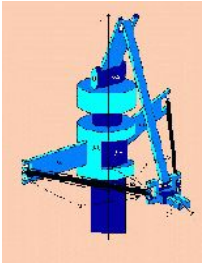


Figure 4.65 : Manipulateur cylindrique

Nous pouvons conclure sur cette étude de variation que l'optimisation d'un critère seul n'est pas très réaliste et qu'il faut par exemple coupler celle-ci avec l'étude des cartes de rigidité effectuée dans notre laboratoire.

Début de définition du système parallèle par lien bidirectionnel avec la CAO

Nous nous limitons à un exemple simple mais qui permet d'expliquer la philosophie du processus de conception que nous mettons en place. Une fois la simulation du squelette validée dans notre outil d'esquisse virtuelle, nous transmettons ce résultat préliminaire à la CAO via un fichier XML.

Comme nous l'avons montré dans la figure 4.54, nous proposons de commencer par définir les surfaces fonctionnelles du mécanisme avant de définir les volumes de celui-ci. Cette technique est actuellement peu compatible avec les outils de CAO volumique et nous envisageons de tester cette intégration dans des modeleurs plus ouverts. Le terme « ouvert » ayant dans notre cas deux sens : le premier correspondant à une plus grande ouverture vers des modèles géométriques plus en cohérence avec notre modélisation basée sur les squelettes et le second ayant pour synonyme le terme « open » signifiant l'accès aux sources du programme. Dans l'exemple que nous présentons, nous supposons que nous avons commencé par définir une partie de la plateforme et que nous souhaitons valider cette première définition (Fig. 4.66).

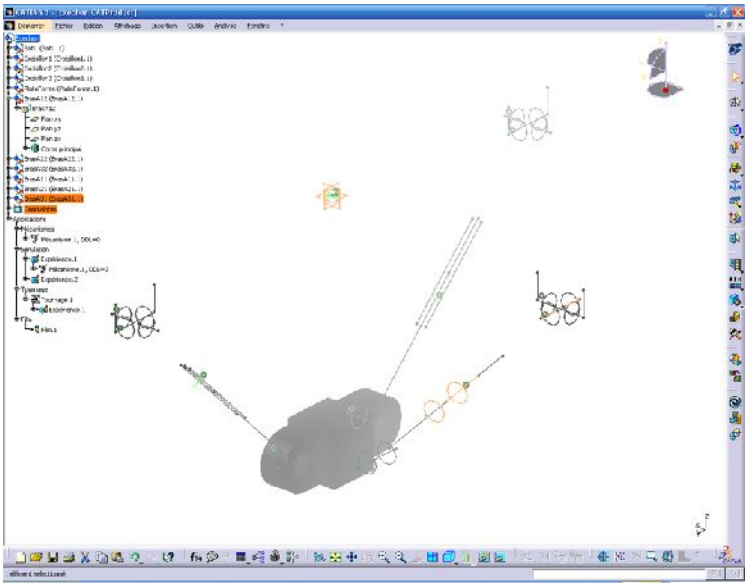


Figure 4.66 : Début de définition de la plateforme dans la CAO

On remarquera sur cette copie d'écran que la structure du produit est automatiquement générée à partir de la lecture du fichier XML ainsi que la récupération des contraintes dues aux liaisons. Dans le module d'analyse cinématique de Catia, les liaisons sont automatiquement et correctement retrouvées à partir des contraintes, ce qui permet d'analyser et de simuler le système. Nous avons choisi de ne définir ensuite qu'une partie de la plateforme, celle-ci est visible sous forme de volume dans la figure 4.66.

La liaison bidirectionnelle impose de pouvoir transmettre cette géométrie à l'outil d'esquisse virtuelle. Pour se faire, nous utilisons le format VRML disponible sous tous les logiciels de CAO. Une nouvelle balise optionnelle a été insérée dans le XML pour indiquer si une représentation VRML d'une pièce était présente ou non. Le résultat de la visualisation de l'Exechon venant de Catia est présenté figure 4.67. La simulation reste exactement la même, nous n'avons fait qu'ajouter une visualisation d'objet donc tout ceci est géré par Java3D et n'intervient pas sur les calculs de détection de collision ou de dynamique.

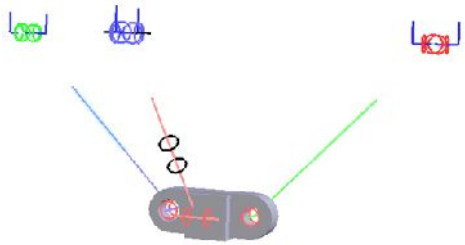


Figure 4.67 : Visualisation avec VRML

Dans cet exemple, la plateforme se déplace normalement avec son squelette. Nous allons montrer un autre cas dans lequel l'objet VRML venant de la CAO restera fixe, c'est l'utilisation de la possibilité dans les outils de CAO de déterminer le volume enveloppe lors d'un mouvement prédéfini. Cette fonctionnalité est utile pour vérifier certains problèmes de détection de collision en dynamique ou d'enlèvement de matière en usinage par exemple. Nous avons généré un volume enveloppe à partir du volume de la plateforme parcourant une trajectoire prédéfinie dans l'outil de CAO. Le résultat complexe de cette forme est

visualisé dans les figures 4.68 et 4.69.

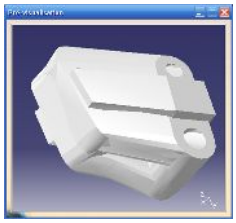


Figure 4.68 : Volume enveloppe de la plateforme, vue 1

Figure 4.69 : Volume enveloppe de la plateforme, vue 2

Cette forme complexe peut elle aussi être transmise à l'outil d'esquisse virtuelle mais dans ce cas, cette forme reste fixe par rapport au bâti.

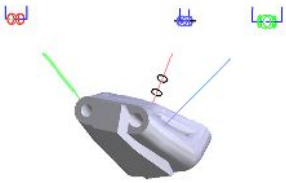


Figure 4.70 : Volume enveloppe dans l'outil d'esquisse virtuelle

Nous terminons notre présentation de proposition d'utilisation de l'outils d'esquisse virtuelle pour la conception d'un système parallèle de type Exechon par une vue 3D de l'intérieur de la structure de l'Exechon (Fig. 4.71). La question que l'on peut se poser en regardant la complexité de la solution retenue est : comment passer de l'utilisation de l'esquisse virtuelle à ce résultat ?

Notre réponse est simple, l'image ci-dessous n'a jamais été un objectif à atteindre avec nos outils, nous travaillons uniquement dans les phases avancées de conception. Nous devons valider une structure, des déplacements et pourquoi pas une dynamique, notre résultat est ensuite repris dans un autre logiciel pour aboutir au résultat de la figure 4.71.

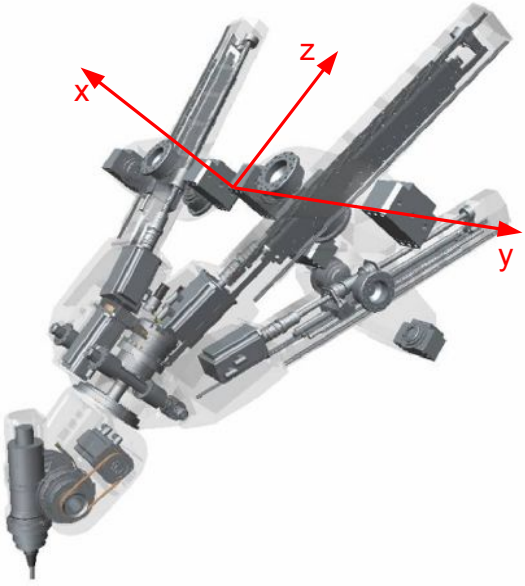


Figure 4.71 : Vue 3D de l'intérieur de la structure de l'Exechon

Conclusion sur la conception avec l'esquisse virtuelle

Nous avons essayé de montrer comment l'outil d'esquisse virtuelle pouvait être utilisée pour la conception d'un système parallèle de type Exechon. De nombreuses fonctionnalités sont encore à implémenter ou à déterminer comme pour l'équilibrage statique automatique par exemple. Nous avons surtout montré que cet outil est adapté aux phases amont de conception tout en permettant de s'intégrer dans les phases suivantes grâce au lien bidirectionnel avec la CAO. Nous pouvons aussi intégrer des éléments au format VRML afin de visualiser par exemple l'encombrement du bâti.

Conclusion générale et perspectives

Nous terminerons la présentation de ce travail par une conclusion générale sur celui-ci puis par les perspectives importantes de l'esquisse virtuelle dans le domaine de la conception de systèmes mécaniques, les avantages et les inconvénients de l'utilisation de la dynamique sous contraintes pour l'animation et la modification du système créé.

Conclusion générale

Notre satisfaction dans la réalisation de ces deux outils d'esquisse virtuelle est à mettre en parallèle avec les difficultés rencontrées sur la programmation de certains concepts, simples en théorie, mais complexes à mettre en oeuvre. Nous avons dû développer de nombreuses parties de code avec plusieurs langages de programmation et en utilisant différentes librairies. Les périodes de doute et de satisfaction ont alterné jusqu'à l'obtention des résultats que nous estimons très prometteurs.

Nous pensons avoir posé les bases d'un outil d'esquisse virtuelle permettant d'effectuer le lien tant attendu entre le concept et la définition du produit en validant principalement deux points :

- le fait que le retour d'effort était une condition nécessaire à la création de systèmes mécaniques directement dans l'espace (condition que nous estimons extrapolable à toutes formes de graphismes en 3D) ;
- la nécessité d'unifier la dynamique et les contraintes pour aboutir à une esquisse virtuelle directement simulable et modifiable en 3D.

Ces résultats doivent maintenant être complétés par de nombreux tests et améliorations, ils peuvent aussi être réutilisés dans d'autres cadres que celui de

l'esquisse virtuelle.

Perspectives

En commençant ce travail, nous avions déjà une idée assez précise de l'opportunité de celui-ci. Le manque d'outil dédié à la conception a été pour nous le point de départ de cette recherche et ce point est resté un fil conducteur lors des choix effectués. Les deux outils développés ont permis de montrer deux points importants :

- Le premier a permis de valider le fait que le retour d'effort était indispensable au tracé d'esquisse virtuelle en 3D ;
- Le deuxième nous a conduit à formuler une dynamique sous contraintes permettant de simuler des systèmes mécaniques et de les modifier.

Ce ne sont pas les seuls résultats intéressants obtenus dans ce travail mais ce sont ceux qui ont servi de fil conducteur et que nous devions valider. Le premier point peut être mise en oeuvre avec la baisse des prix des systèmes haptiques. Nous avons déjà testé la validité de l'Omni de Sensable mais nous souhaitons aussi essayer le nouveau périphérique orienté jeu de la société Novint, ou les périphériques français de la société Haption. Tous ces périphériques sont directement intégrables puisqu'ils sont compatibles avec Haptic Library. Le deuxième point peut constituer une rupture dans les sketchers actuels en réalisant la liaison entre la dynamique et les contraintes. Nous sommes conscients du travail qui reste à réaliser sur le code actuel mais les fondements théoriques sont posés.

Lien entre les deux outils d'esquisse

Nous devons refaire le lien entre le tracé à main levée et les géométries simples utilisées dans le deuxième outil. Un travail a débuté dans ce sens, la bibliographie sur ce sujet est fournie. Le nombre de géométrie que nous souhaitons reconnaître est très faible : un segment ou un arc de cercle, par contre nous devons déterminer le plus rapidement possible l'intention du concepteur avec en plus l'utilisation de contraintes géométriques sur les autres entités présentes ou sur des directions importantes (x, y, z). Par exemple, comme dans les sketchers utilisés en CAO en 2D, si le concepteur déplace le stylet de la gauche vers la droite en suivant approximativement une droite, il faut que l'on détermine à partir des premiers points que son intention est de créer une droite puis que celle-ci est sur l'axe x.

L'analyse de la courbure obtenue à partir des premiers points, devrait nous permettre de basculer soit sur le segment soit sur l'arc de cercle mais cette idée simple n'est pas très robuste. Nous devons l'améliorer et essayer une technique plus classique dans le domaine de la reconnaissance de forme qui est la transformation de Hough. Il est possible d'ajouter des contraintes sur celle-ci de type 'passe par' ou autre.

Amélioration des performances

Il reste un travail important afin de rendre plus robuste le code développé afin de pouvoir passer à des tests plus importants sur l'effet de l'outil d'esquisse virtuelle sur la conception de systèmes mécaniques.

L'optimisation la plus simple à réaliser est la gestion correcte de la mémoire. Celle-ci nécessite en premier lieu de supprimer toute nouvelle création d'objet dans les boucles. L'utilisation d'outil de Profiling facilite cette tâche ingrate mais il n'existe actuellement aucun outil qui permet d'effectuer ce type d'analyse dans un code mélangeant du Java et du C++. Il nous faut donc soit redéfinir l'interface graphique en C++ avec une librairie comme Ogre3D, soit découpler temporairement le graphisme du calcul et ne travailler qu'avec des fichiers en entrée et en sortie.

Les outils de Profiling permettent ensuite de déterminer les pertes de temps ce qui permet d'optimiser le code en conséquence sans a priori sur les opérations qui ralentissent l'exécution. Nous avions ainsi vu à quel point les opérations mathématiques comme le calcul des sinus étaient à éviter.

Sans les analyses précédentes, nous pouvons aussi chercher, à partir du code, où se situent les opérations inutiles. Lors des multiples tests que nous avons effectués, nous avons rencontré des matrices relativement creuses comme la matrice des contraintes, il peut être intéressant de revoir le stockage de ces éléments ainsi que leurs manipulations.

La détection de collision peut aussi être améliorée par des pré-tests permettant de détecter plus rapidement les possibilités ou non de collision. Ceci est d'autant plus important que le nombre de géométries augmente. Une technique simple peut être employée consistant à indiquer quelles sont les géométries susceptibles de rentrer en contact. L'inconvénient de cette technique est le manque de généricité et un travail de sélection manuel important.

Nous souhaitons aussi effectuer des essais pour remplacer les threads actuels par la librairie Intel Threading Building Blocks [RE07]. L'augmentation de performances est importante, les processeurs multi-coeurs sont tranformés en véritable cluster.

Ajout de géométries et de contraintes

Nous n'avons pas implanté la gestion des chocs avec des surfaces par faute de temps et par choix préférant valider cette gestion avec les segments et les cercles. Le passage aux autres éléments définis dans le tableau 7 ne pose aucun problème théorique, seules des difficultés d'implémentation sont à craindre c'est pourquoi nous n'effectuerons ce travail qu'après une amélioration du code actuel.

L'utilisation de formes paramétrées plus complexes (Bspline, Nurbs, etc.) ne pose pas de difficultés en dehors de la gestion du nombre de contacts possibles. La méthode que nous avons mise au point s'applique à toute entité définie par des paramètres variant de 0 à 1.

Pour les contraintes, en dehors des nouvelles contraintes liées à la prise en compte de nouvelles géométries, nous devons aussi implémenter les contraintes non idéales. Ce qui nous permettra d'intégrer tout type de frottement (sec ou visqueux), des contraintes de roulement sans glissement, etc.

Interfaçage avec la CAO

Nous avons développé une interface avec d'autres logiciels grâce à notre fichier au format XML. Nous réfléchissons actuellement aux informations à ajouter à ce fichier qui nous permettraient de faire un lien bidirectionnel entre l'outil d'esquisse et la CAO. Actuellement le lien fonctionne pour les géométries simples (point, segment et cercle) et pour les liaisons qui sont converties en contraintes d'assemblage dans l'outil de CAO. Nous avons ajouté le fait de pouvoir visualiser des fichiers VRML dans l'outil d'esquisse mais nous souhaitons maintenant ajouter des informations facilitant le travail à plusieurs sur le même système. Nous espérons même pouvoir faciliter le travail à plusieurs sur la même pièce ce qui n'est pas encore prévu dans les outils de CAO, en ajoutant lors de la construction des entités des informations sur le nom du concepteur, les droits de modifications, les règles de conception utilisée, etc.

Pour nous, le squelette du mécanisme ne devrait pas pouvoir être modifié dans la CAO sauf cas extrême validé par l'ensemble des concepteurs. Ceci signifie que c'est le squelette qui pilote la définition du produit et qu'en cas de modification de celui-ci, la définition doit automatiquement être mise à jour. Une solution assez simple par ajout d'attributs dans le format XML est envisagée.

Nous veillons aussi à limiter le nombre d'informations transmises de la CAO à l'outil d'esquisse virtuelle, en effet celui-ci n'a pas pour vocation à remplacer des outils beaucoup plus complexes mais à permettre une interaction avec le squelette de l'objet créé. Nous avons ajouté la possibilité de visualiser la géométrie des pièces créées en CAO via VRML parce que cette option ne perturbe pas les performances de l'outil et nous ferons de même avec d'autres options si l'utilité et les possibilités se présentent. Le rendu des cartes de rigidité peut, par exemple, être envisagé soit graphiquement mais en prenant le risque de surcharger la visualisation soit avec un retour d'effort simulant la flexibilité du mécanisme.

Passage du concept à la définition du produit

Notre outil d'esquisse virtuelle doit permettre à terme d'obtenir le squelette complet du mécanisme avec ses surfaces fonctionnelles et les caractéristiques des différentes sections de celui-ci. Les surfaces fonctionnelles sont par exemple déterminées pour limiter les pressions de contact et les sections sont calculées pour limiter la déformation globale du système. Cette démarche est celle proposée dans la méthode de conception d'Ullman. Notre outil doit donc gérer ces différents éléments géométriques et technologiques, ceux-ci doivent pouvoir se retrouver dans le fichier XML afin de piloter la suite de la définition dans un outil de CAO.

Dans l'état actuel de notre développement l'habillage du squelette est effectué dans la CAO ce qui nous permet de pouvoir utiliser les modules KBE de Catia, par exemple, pour utiliser des règles de conception afin d'automatiser l'habillage du squelette. Cette solution n'est pas pour nous optimale pour plusieurs raisons :

- Cette démarche devient liée à un outil propriétaire ce qui limite la réutilisation avec d'autres outils ;
- Les outils de CAO actuels ne sont pas adaptés à la modélisation basée sur le squelette. L'utilisation de la modélisation volumique est pour nous une

impasse pour ce type de conception ;

- La conception collaborative n'a jamais été prévue en CAO ce qui est aussi une importante limitation pour le concurrent design à la Ullman ;
- L'utilisation de moteur de recherche 3D est beaucoup plus efficace sur des modèles pilotés par le squelette. Des recherches de pièces existantes, s'appuyant à la fois sur la géométrie du squelette et sur les éléments technologiques, sont plus simples et plus efficaces dans ce cas.

Nous avons déjà réussi à intégrer les contraintes à la dynamique, l'étape suivante consistera à intégrer des outils support du KBE directement afin d'accroître les possibilités de notre développement.

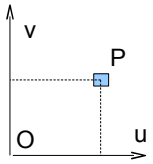
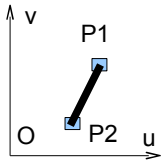
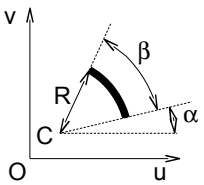
De nombreuses perspectives sont donc ouvertes à la suite de notre travail et nous imaginons que de nombreuses autres sont possibles auxquelles nous n'avons pas encore pensées. Malheureusement le temps n'est pas compressible et il faudra faire des choix en fonction des éléments les plus intéressants à développer. Nous souhaitons aussi travailler avec les autres laboratoires qui développent ce type d'outils.

ANNEXES

A.1 Définition des géométries

Nous pésentons ici le paramétrage des différentes géométries utilisées en commençant par les géométries 2D définies dans l'espace des paramètres u et v. Ces géométries sont utilisées pour définir des parcours délimitant des zones de contact sur les géométries 3D de dimension 3.

Géométrie 2D

Point2D	Segment2D	Cercle2D
 *Figure A.1 : Paramétrage du Point2D*	 *Figure A.2 : Paramétrage du Segment2D*	 *Figure A.3 : Paramétrage du Cercle2D*

Géométrie3D

Dimension 0 (Point)

Point3D

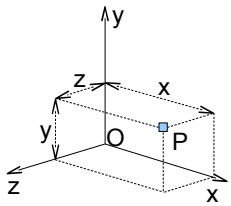


Figure A.4 : Paramétrage du Point3D

Dimension 1 (Courbe)

Segment3D

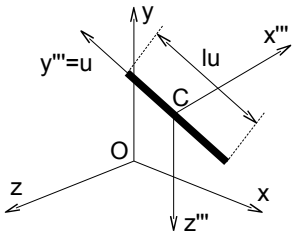


Figure A.5 : Paramétrage du Segment3D

Cercle3D

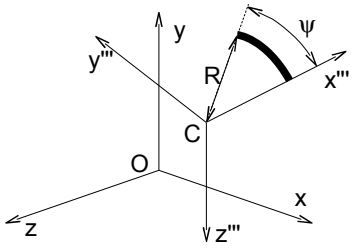


Figure A.6 : Paramétrage du Cercle3D

Le changement de repère se fait en utilisant les angles d'Euler :

- rotation autour de y d'un angle α
- rotation autour de x' d'un angle β
- rotation autour de y'' d'un angle γ

Dimension 2 (Surface)

Plan3D

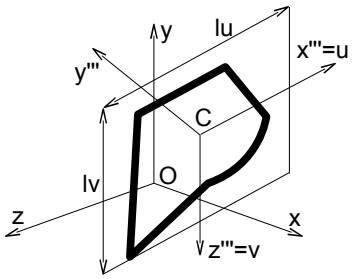


Figure A.7 : Paramétrage du Plan3D

Un parcours 2D peut délimiter la surface plane.

Cylindre3D

Comme pour le cercle 3D, on peut utiliser un arc et non un cercle. Sur la figure le cercle est complet donc $\psi = 2\pi$.

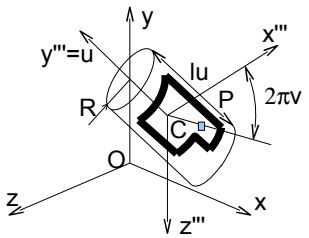


Figure A.8 : Paramétrage du Cercle3D

Sphere3D

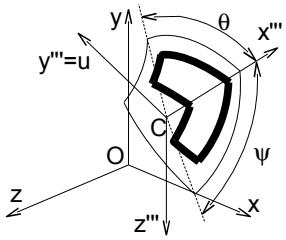


Figure A.9 : Paramétrage de la Sphere3D

Cone3D

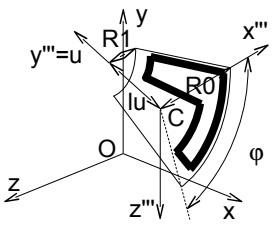


Figure A.10 : Paramétrage du Cone3D

A.2 Hiérarchie de sphères

Une hiérarchie de sphères a été utilisée dans le premier prototype afin de rechercher le point le plus près d'un tracé par rapport à l'extrémité du curseur et pour la détection de collision entre deux tracés. Nous ne présenterons ici que la recherche du point le plus proche sans entrer dans les détails. Ces techniques sont relativement classiques dans le domaine de la réalité virtuelle.

La figure A.11 ci-dessous représente un tracé à main levée, le stylo est ensuite approché du tracé et l'on souhaite déterminer la distance la plus courte afin de créer un effort d'attraction si l'extrémité du stylo est proche du tracé.

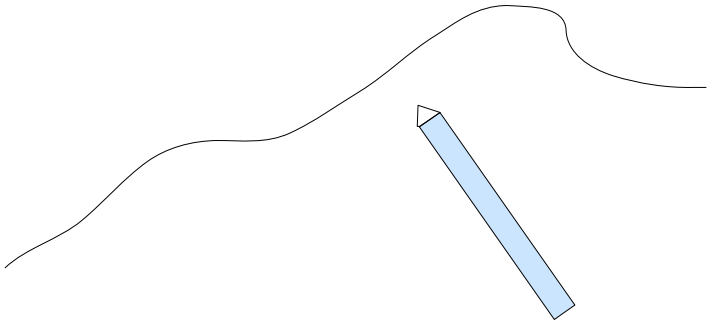


Figure A.11 : Recherche de la plus courte distance

Le tracé est modélisé par un ensemble de segments, chaque segment va être englobé dans une sphère comme représenté dans la figure A.12.

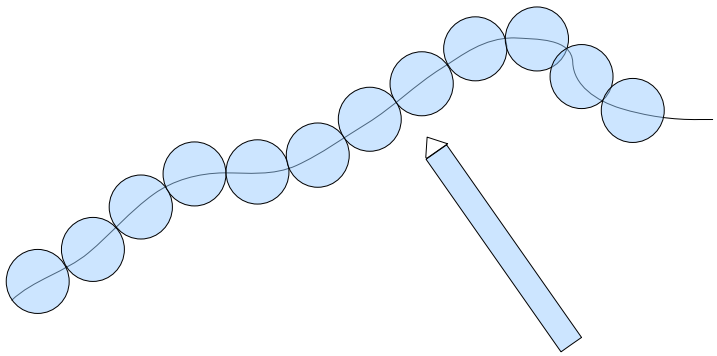


Figure A.12 : Premier niveau de la hiérarchie de sphères

Les deux premières sphères sont à leur tour englobées dans une nouvelle sphère et ainsi de suite.

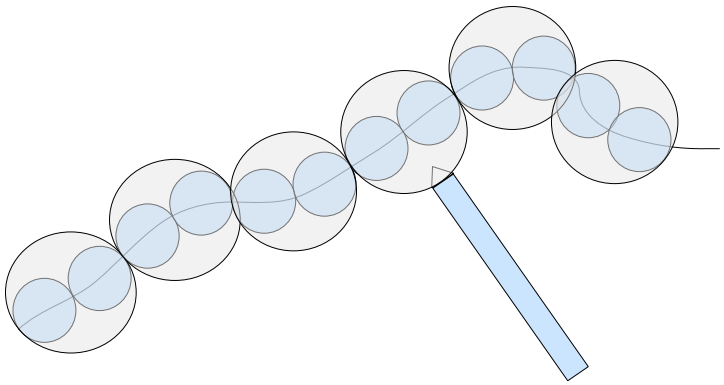


Figure A.13 : Deuxième niveau de la hiérarchie de sphères

Ce travail est itérativement exécuté jusqu'à l'obtention de la sphère la plus englobante.

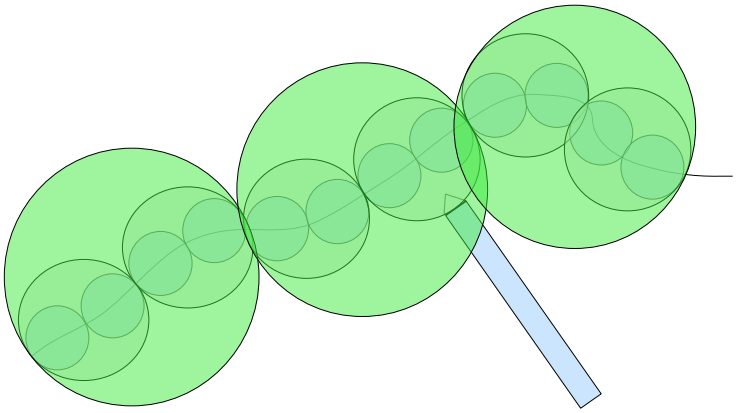


Figure A.14 : Troisième niveau de la hiérarchie de sphères

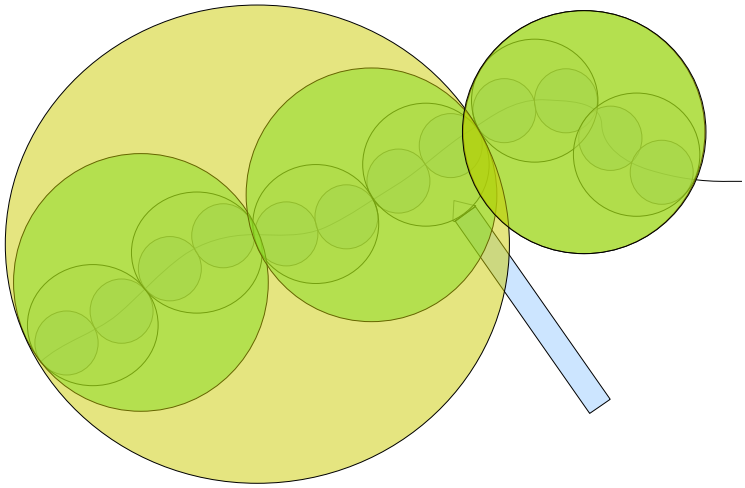


Figure A.15 : Quatrième niveau de la hiérarchie de sphères

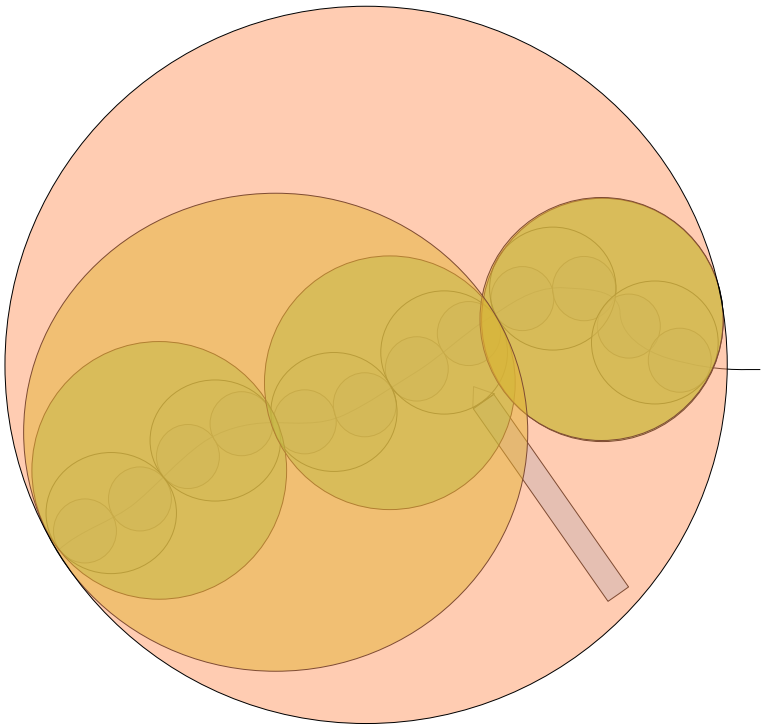


Figure A.16 : Dernier niveau de la hiérarchie de sphères

Cette hiérarchie de sphères peut être représentée par un arbre binaire (Fig. A.17) et une recherche en profondeur d'abord va être utilisée.

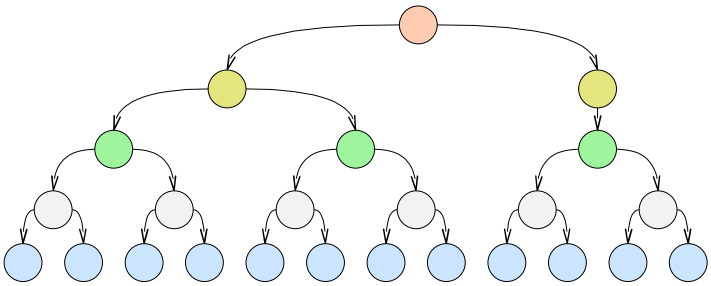


Figure A.17 : Arbre binaire

A.3 Arithmétique des intervalles

Nous ne présenterons ici que les éléments de base implémentés dans notre code ainsi que certaines problèmes qui sont apparus lors de calculs effectués avec l'arithmétique des intervalles.

Un intervalle $\tilde{a}$ sera noté $[a_l\ a_h]$ avec $a_l \leqslant a_h$

A.3.1 Fonctions de base

Addition de deux intervalles :

$$\tilde{a}=\tilde{b}+\tilde{c} \Rightarrow \begin{matrix} l_a=l_b+l_c \\ h_a=h_b+h_c \end{matrix} \quad (1)$$

Résultat obtenu en 2 additions.

Si l'on veut utiliser les mêmes opérations avec des réels, comme par exemple effectuer l'addition d'un intervalle et d'un réel, il suffit de remplacer ce réel par un intervalle centré sur ce réel et de largeur nulle.

Multiplication de deux intervalles :

$$\tilde{a}=\tilde{b}\ x\ \tilde{c} \Rightarrow \begin{matrix} l_a=min(l_b x\, l_c, l_b x\, h_c, h_b x\, l_c, h_b x\, h_c) \\ h_a=max(l_b x\, l_c, l_b x\, h_c, h_b x\, l_c, h_b x\, h_c) \end{matrix} \quad (2)$$

Résultat obtenu en 4 multiplications et 5 comparaisons.

Multiplication d'un intervalle par un réel :

$$\tilde{a}=\tilde{b}\ x\ c \Rightarrow \begin{matrix} l_a=min(l_b x\, c, h_b x\, c) \\ h_a=max(l_b x\, c, h_b x\, c) \end{matrix} \quad (3)$$

Résultat obtenu en 2 multiplications et 1 comparaison.

Contrairement à l'addition, on remarque qu'il est préférable d'effectuer un calcul spécifique pour la multiplication d'un intervalle par un réel.

A.3.2 Fonctions spéciales

Carré d'un intervalle

Nous allons simplement montrer le type d'erreur qu'il ne faut pas commettre quand on utilise l'arithmétique des intervalles. Si l'on écrit par exemple que pour calculer le carré d'un intervalle, il suffit de multiplier l'intervalle par lui-même

$\tilde{a}^2 = \tilde{a} \, x \, \tilde{a}$, on peut aboutir au résultat suivant si $\tilde{a} = [-2 \; 3]$ alors $\tilde{a}^2 = [-6 \; 9]$. Ce résultat est faux, les bornes de cet intervalle devraient toujours être positives ou nulles et donc $\tilde{a}^2 = [0 \; 9]$. On peut donc noter que $\tilde{a}^2 \neq \tilde{a} \, x \tilde{a}$ et qu'il est nécessaire d'affecter un résultat en fonction des signes des bornes.

Fonctions trigonométriques

Comme nous l'avons indiqué précédemment, nous avons optimisé ce calcul afin de limiter le nombre d'appels à des fonctions trigonométriques. Si on cherche $\tilde{a} = \sin(\tilde{b})$, nous avons choisi d'écrire que si $b_h \leqslant 0.01$ alors $\tilde{a} = \tilde{b}$. Dans l'autre cas, il faut une nouvelle fois faire très attention à tous les cas possibles. Par exemple si $\tilde{b} = [\frac{\pi}{6} \; \frac{5\pi}{6}]$, il faut éviter de conclure trop rapidement que $\tilde{a} = [\frac{1}{2} \; \frac{1}{2}]$, en effet sur l'ensemble de l'intervalle, le sinus passe par la valeur 1 et donc, $\tilde{a} = [\frac{1}{2} \; 1]$.

L'arithmétique des intervalles doit aussi être capable de gérer les erreurs d'arrondis dans les calculs en fonction du codage des chiffres en binaire. Les librairies comme IASolver, gèrent ce problème mais en contrepartie, ne contiennent qu'une petite partie des fonctions qui nous sont utiles.

A.3.3 Hiérarchie d'intervalles

L'utilisation d'une hiérarchie d'intervalles qui consiste à découper en deux un intervalle de manière récursive, a fait apparaître certaines aberrations lors de simples multiplications par exemple. Admettons que nous définissions une hiérarchie d'intervalles sur l'intervalle $\tilde{a} = [1 \; 5]$ et que l'on effectue une multiplication par l'intervalle $\tilde{b} = [4 \; 5]$. Au niveau le plus élevé de la hiérarchie, le résultat donne $\tilde{c} = [4 \; 25]$. Au niveau suivant, nous obtenons pour $\tilde{a}$ deux intervalles : $\tilde{a}_1 = [1 \; 3]$ et $\tilde{a}_2 = [3 \; 5]$ d'où les deux nouveaux résultats : $\tilde{c}_1 = [4 \; 15]$ et $\tilde{c}_2 = [12 \; 25]$. On remarque qu'il y a un recouvrement de ces deux résultats qui correspondra à un double balayage inutile de l'intervalle $[12 \; 15]$. On peut définir un taux de recouvrement correspondant à la largeur de l'intervalle balayé inutilement sur la largeur de l'intervalle résultat au niveau le plus élevé, dans notre cas ce taux = $\frac{3}{21} = 0.142$. Si l'on effectue les mêmes multiplications au niveau inférieurs (4 multiplications), le taux de recouvrement passe à 0.428 .

Les pertes de temps de calcul sont donc importantes. Nous avons essayé de modifier l'opérateur de multiplication de l'équation 2 mais on obtient un résultat encore plus gênant avec des zones non-recouvertes. La seule solution consiste en la mise en place d'un nouvel opérateur spécifique à la multiplication d'une hiérarchie d'intervalles. Nous proposons de prendre la moyenne de la borne haute de $\tilde{c}_1$ et de la borne basse de $\tilde{c}_2$. Plus formellement, si $\tilde{a}_1 = [l_a \; m_a]$

et $\tilde{a}_2=[m_a\ h_a]$, $\tilde{b}=[l_b\ h_b]$ alors :

$$\begin{aligned} l_c &= min(l_a x\, l_b, l_a x\, h_b, m_a x\, l_b, m_a x\, h_b) \\ m_c &= \frac{max(l_a x\, l_b, l_a x\, h_b, m_a x\, l_b, m_a x\, h_b)}{2} \\ &+ \frac{min(m_a x\, l_b, m_a x\, h_b, h_a x\, l_b, h_a x\, h_b)}{2} \\ h_c &= max(m_a x\, l_b, m_a x\, h_b, h_a x\, l_b, h_a x\, h_b) \end{aligned} \quad (4)$$

Et l'on peut dans ce cas obtenir la même borne intermédiaire, d'où les deux résultats : $\tilde{c}_1=[l_c\ m_c]$ et $\tilde{c}_2=[m_c\ h_c]$. Cette nouvelle multiplication ne nécessite qu'une division par deux supplémentaire pour deux résultats.

A.4 Format de fichier

Nous donnons un exemple du fichier mecanisme.dtd ainsi qu'une partie du fichier BielleManivelle.xml. La ligne suivante dans le fichier xml:

```
<!DOCTYPE mecanisme SYSTEM "mecanisme.dtd">
```

impose le respect du fichier mecanisme.dtd.

A.4.1 mecanisme.dtd

```
<?xml version='1.0' encoding='us-ascii'?>

<!--
   DTD for a mechanism.
-->

<!ELEMENT mecanisme (infoMecanisme, piece+, liaison+)>
<!ELEMENT infoMecanisme (nomMecanisme)>
<!ELEMENT nomMecanisme (#PCDATA)>
<!ELEMENT piece (infoPiece, infoInertie, infoGeometrie)>
<!ELEMENT infoPiece (nomPiece*, numeroPiece)>
<!ELEMENT nomPiece (#PCDATA)>
<!ELEMENT numeroPiece (#PCDATA)>
<!ELEMENT infoInertie (masse, cdgX, cdgY, cdgZ, inertieX, inertieY, inertieZ, alphaInertie, betaInertie,
gammaInertie)>
<!ELEMENT masse (#PCDATA)>
<!ELEMENT cdgX (#PCDATA)>
<!ELEMENT cdgY (#PCDATA)>
<!ELEMENT cdgZ (#PCDATA)>
<!ELEMENT inertieX (#PCDATA)>
<!ELEMENT inertieY (#PCDATA)>
<!ELEMENT inertieZ (#PCDATA)>
<!ELEMENT alphaInertie (#PCDATA)>
<!ELEMENT betaInertie (#PCDATA)>
<!ELEMENT gammaInertie (#PCDATA)>

<!ELEMENT infoGeometrie (geometrie2D*, geometrie3D)>

<!ELEMENT geometrie2D (infoPoint2D, infoSegment2D*, infoCercle2D*, infoParcours2D*)>

<!ELEMENT infoPoint2D (point2D+)>
<!ELEMENT point2D (uPoint2D, vPoint2D)>
<!ELEMENT uPoint2D (#PCDATA)>
<!ELEMENT vPoint2D (#PCDATA)>

<!ELEMENT infoSegment2D (segment2D+)>
<!ELEMENT segment2D (p1segment2D, p2segment2D)>
<!ELEMENT p1segment2D (#PCDATA)>
<!ELEMENT p2segment2D (#PCDATA)>

<!ELEMENT infoCercle2D (cercle2D+)>
<!ELEMENT cercle2D (centreCercle2D, alphaCercle2D, betaCercle2D, rayonCercle2D)>
<!ELEMENT centreCercle2D (#PCDATA)>
<!ELEMENT alphaCercle2D (#PCDATA)>
<!ELEMENT betaCercle2D (#PCDATA)>
<!ELEMENT rayonCercle2D (#PCDATA)>

<!ELEMENT infoParcours2D (parcours2D+)>
<!ELEMENT parcours2D (entitesParcours2D+)>
<!ELEMENT entitesParcours2D (numeroEntiteParcours2D, typeEntiteParcours2D)>
```

```
<!ELEMENT numeroEntiteParcours2D (#PCDATA)>
<!ELEMENT typeEntiteParcours2D (#PCDATA)>

<!ELEMENT geometrie3D (dimension0, dimension1, dimension2*, infoParcours3D*)>

<!ELEMENT dimension0 (infoPoint3D)>

<!ELEMENT infoPoint3D (point3D+)>
<!ELEMENT point3D (xPoint3D, yPoint3D, zPoint3D, utilePoint3D)>
<!ELEMENT xPoint3D (#PCDATA)>
<!ELEMENT yPoint3D (#PCDATA)>
<!ELEMENT zPoint3D (#PCDATA)>
<!ELEMENT utilePoint3D (#PCDATA)>

<!ELEMENT dimension1 (infoSegment3D*, infoCercle3D*)>

<!ELEMENT infoSegment3D (segment3D+)>
<!ELEMENT segment3D (centreSegment3D, alphaSegment3D, betaSegment3D, gammaSegment3D,
luSegment3D)>
<!ELEMENT centreSegment3D (#PCDATA)>
<!ELEMENT alphaSegment3D (#PCDATA)>
<!ELEMENT betaSegment3D (#PCDATA)>
<!ELEMENT gammaSegment3D (#PCDATA)>
<!ELEMENT luSegment3D (#PCDATA)>

<!ELEMENT infoCercle3D (cercle3D+)>
<!ELEMENT cercle3D (centreCercle3D, alphaCercle3D, betaCercle3D, gammaCercle3D, phiCercle3D,
rayonCercle3D)>
<!ELEMENT centreCercle3D (#PCDATA)>
<!ELEMENT alphaCercle3D (#PCDATA)>
<!ELEMENT betaCercle3D (#PCDATA)>
<!ELEMENT gammaCercle3D (#PCDATA)>
<!ELEMENT phiCercle3D (#PCDATA)>
<!ELEMENT rayonCercle3D (#PCDATA)>

<!ELEMENT dimension2 (infoPlan3D*, infoCylindre3D*, infoSphere3D*, infoCone3D*)>

<!ELEMENT infoPlan3D (plan3D+)>
<!ELEMENT plan3D (centrePlan3D, alphaPlan3D, betaPlan3D, gammaPlan3D, luPlan3D, lvPlan3D,
parcours2DPlan3D)>
<!ELEMENT centrePlan3D (#PCDATA)>
<!ELEMENT alphaPlan3D (#PCDATA)>
<!ELEMENT betaPlan3D (#PCDATA)>
<!ELEMENT gammaPlan3D (#PCDATA)>
<!ELEMENT luPlan3D (#PCDATA)>
<!ELEMENT lvPlan3D (#PCDATA)>
<!ELEMENT parcours2DPlan3D (#PCDATA)>

<!ELEMENT infoCylindre3D (cylindre3D+)>
<!ELEMENT cylindre3D (centreCylindre3D, alphaCylindre3D, betaCylindre3D, gammaCylindre3D,
phiCylindre3D, luCylindre3D, rayonCylindre3D, parcours2DCylindre3D)>
<!ELEMENT centreCylindre3D (#PCDATA)>
<!ELEMENT alphaCylindre3D (#PCDATA)>
<!ELEMENT betaCylindre3D (#PCDATA)>
<!ELEMENT gammaCylindre3D (#PCDATA)>
<!ELEMENT phiCylindre3D (#PCDATA)>
<!ELEMENT luCylindre3D (#PCDATA)>
<!ELEMENT rayonCylindre3D (#PCDATA)>
<!ELEMENT parcours2DCylindre3D (#PCDATA)>

<!ELEMENT infoSphere3D (sphere3D+)>
<!ELEMENT sphere3D (centreSphere3D, alphaSphere3D, betaSphere3D, gammaSphere3D, phiSphere3D,
tetaSphere3D, rayonSphere3D, parcours2DSphere3D)>
<!ELEMENT centreSphere3D (#PCDATA)>
<!ELEMENT alphaSphere3D (#PCDATA)>
<!ELEMENT betaSphere3D (#PCDATA)>
<!ELEMENT gammaSphere3D (#PCDATA)>
<!ELEMENT phiSphere3D (#PCDATA)>
<!ELEMENT tetaSphere3D (#PCDATA)>
<!ELEMENT rayonSphere3D (#PCDATA)>
<!ELEMENT parcours2DSphere3D (#PCDATA)>

<!ELEMENT infoCone3D (cone3D+)>
<!ELEMENT cone3D (centreCone3D, alphaCone3D, betaCone3D, gammaCone3D, phiCone3D, luCone3D,
```

```
rayon0Cone3D, rayon1Cone3D, parcours2DCone3D)>
<!ELEMENT centreCone3D (#PCDATA)>
<!ELEMENT alphaCone3D (#PCDATA)>
<!ELEMENT betaCone3D (#PCDATA)>
<!ELEMENT gammaCone3D (#PCDATA)>
<!ELEMENT phiCone3D (#PCDATA)>
<!ELEMENT luCone3D (#PCDATA)>
<!ELEMENT rayon0Cone3D (#PCDATA)>
<!ELEMENT rayon1Cone3D (#PCDATA)>
<!ELEMENT parcours2DCone3D (#PCDATA)>

<!ELEMENT infoParcours3D (parcours3D+)>
<!ELEMENT parcours3D (fermeParcours3D, entitesParcours3D+)>
<!ELEMENT fermeParcours3D (#PCDATA)>
<!ELEMENT entitesParcours3D (numeroEntiteParcours3D, typeEntiteParcours3D)>

<!ELEMENT liaison (typeLiaison, nomLiaison, numeroLiaison, numeroPiece1, numeroPiece2, xLiaison,
yLiaison, zLiaison, alphaLiaison, betaLiaison, gammaLiaison)>
<!ELEMENT typeLiaison (#PCDATA)>
<!ELEMENT nomLiaison (#PCDATA)>
<!ELEMENT numeroLiaison (#PCDATA)>
<!ELEMENT numeroPiece1 (#PCDATA)>
<!ELEMENT numeroPiece2 (#PCDATA)>
<!ELEMENT xLiaison (#PCDATA)>
<!ELEMENT yLiaison (#PCDATA)>
<!ELEMENT zLiaison (#PCDATA)>
<!ELEMENT alphaLiaison (#PCDATA)>
<!ELEMENT betaLiaison (#PCDATA)>
<!ELEMENT gammaLiaison (#PCDATA)>
```

A.4.2 BielleManivelle.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
   Document   : BielleManivelle.xml
   Created on : 8 février 2007, 11:56
   Author     : Administrateur
   Description:
      Purpose of the document follows.
-->
<!DOCTYPE mecanisme SYSTEM "mecanisme.dtd">
<mecanisme>
  <infoMecanisme>
    <nomMecanisme>Système 1</nomMecanisme>
  </infoMecanisme>

  <piece>
    <infoPiece>
      <nomPiece>Bati</nomPiece>
      <numeroPiece>0</numeroPiece>
    </infoPiece>

    <infoInertie>
      <masse>0.1</masse>
      <cdgX>0.0</cdgX>
      <cdgY>0.0</cdgY>
      <cdgZ>0.0</cdgZ>
      <inertieX>0.01</inertieX>
      <inertieY>0.01</inertieY>
      <inertieZ>0.01</inertieZ>
      <alphaInertie>0.0</alphaInertie>
      <betaInertie>0.0</betaInertie>
      <gammaInertie>0.0</gammaInertie>
    </infoInertie>

    <infoGeometrie>
      <geometrie3D>
        <dimension0>
          <infoPoint3D>
            <point3D>
```

```
<!--0-->
  <xPoint3D>0.0</xPoint3D>

  <yPoint3D>0.0</yPoint3D>

  <zPoint3D>0.075</zPoint3D>

  <utilePoint3D>0</utilePoint3D>
</point3D>

<point3D>
<!--1-->
  <xPoint3D>0.0</xPoint3D>

  <yPoint3D>0.0</yPoint3D>

  <zPoint3D>0.025</zPoint3D>

  <utilePoint3D>0</utilePoint3D>
</point3D>
```

A.5 Courbes des tests

Les courbes des tests sont placées ici pour une meilleure lecture du document.

A.5.1 Descente du piston en fonction du jeu

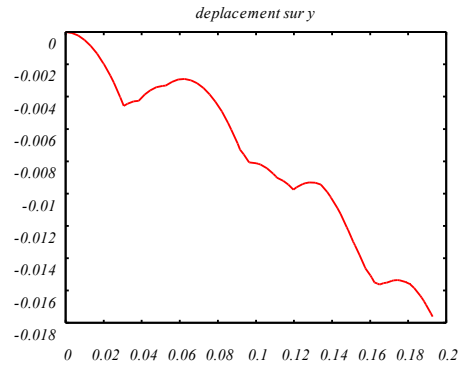
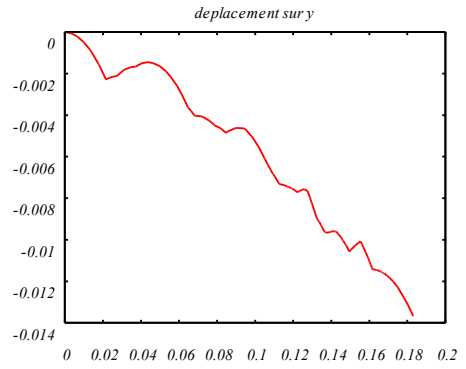
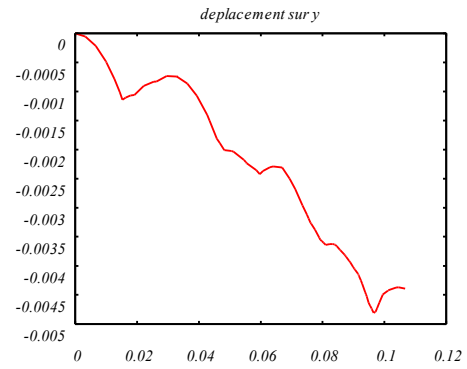
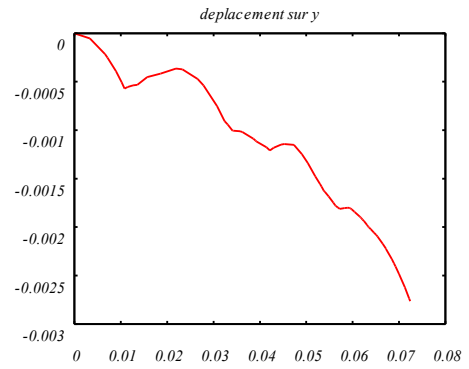
0.002	0.001
	
0.0005	0.00025
	

Tableau 20 : Evolution du déplacement du piston en fonction du jeu

A.5.2 Descente du piston avec des liaisons

Les deux courbes suivantes permettent de montrer que l'utilisation de contraintes de liaison donne une évolution beaucoup plus lissée du déplacement du piston sur y . La première courbe obtenue sur un temps similaire aux précédentes, permet de vérifier que les courbes avec choc ont une forme globale équivalente.

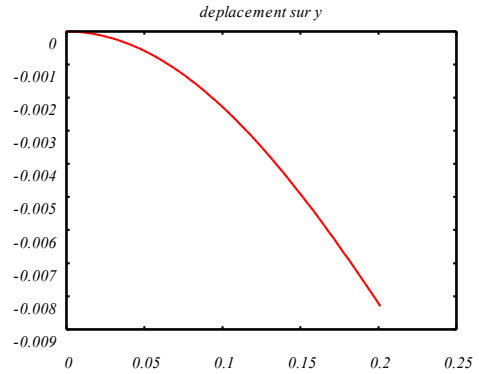


Figure A.18 : Déplacement du piston avec des liaisons sur un temps court

En nous limitant sur deux cycles, on peut remarquer la régularité du mouvement lors de l'utilisation de contraintes de liaison.

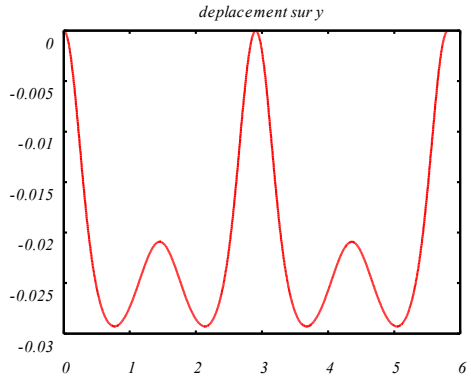


Figure A.19 : Déplacement du piston avec liaisons sur deux cycles

A.5.3 Vitesse de rotation de la manivelle

Afin de montrer la qualité du couplage entre des contraintes de liaison et des contraintes de choc, nous présentons l'évolution de la vitesse de rotation de la manivelle en commençant par le cas sans choc :

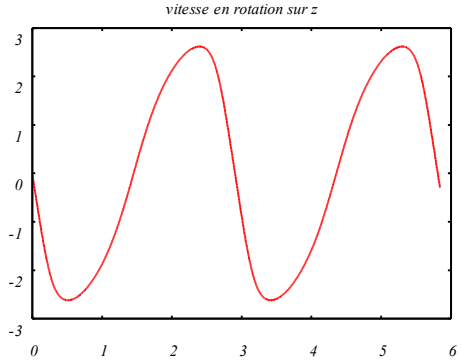


Figure A.20 : Vitesse de rotation de la manivelle sans choc

La courbe précédente peut être comparée avec celles ci-dessous dans lesquelles un choc a été ajouté sur la manivelle. Dans ces courbes, nous avons fait évoluer le coefficient d'absorption e .

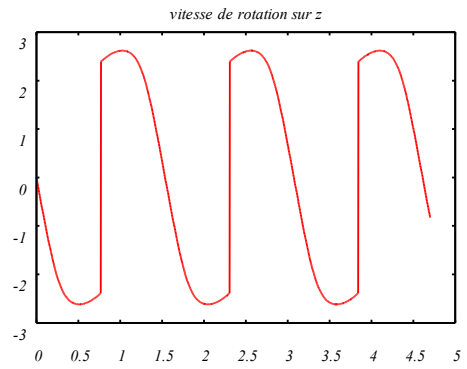
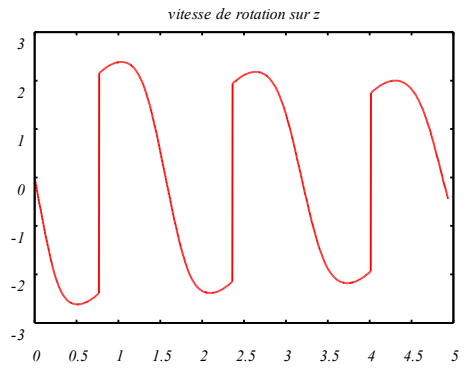
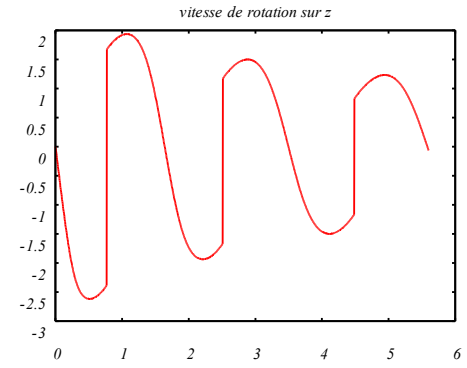
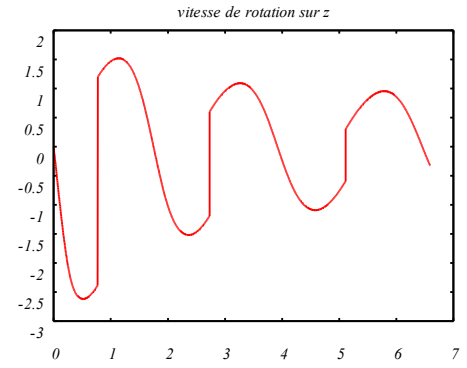
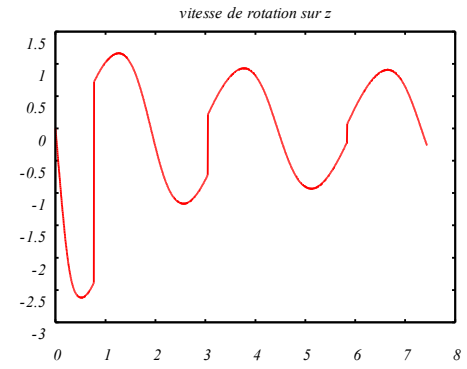
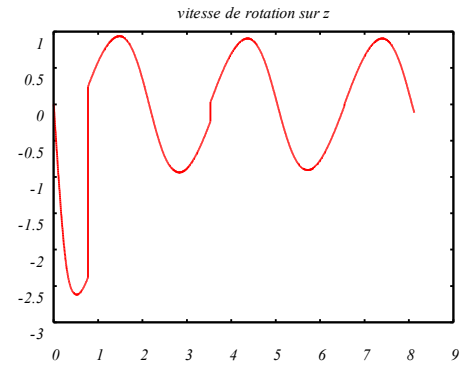
$e = 1.0$	$e = 0.9$
	
$e = 0.7$	$e = 0.5$
	
$e = 0.3$	$e = 0.1$
	

Tableau 21 : Evolution de la vitesse de rotation de la manivelle en fonction du coefficient d'absorption

A.6 Exemple de détermination des accélérations

Nous donnons ici des détails sur les calculs matriciels mis en oeuvre pour la dynamique sous contraintes avec la formulation d'Udwadia et de Kalaba.

A.6.1 Système bielle manivelle avec liaisons

Cet exemple est donné pour le système bielle manivelle avec liaisons dans sa position de départ afin de faciliter la compréhension.

$$A=\begin{bmatrix}
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & -0.05 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & -0.05 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & -0.05 & 0 & 1 & 0 & 0 & 0 & 0.05 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -0.05 & -1 & 0 & 0 & 0 & 0.05 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0.1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -0.1 & 0 & 0
\end{bmatrix} \quad et \quad b=\begin{bmatrix}
0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0
\end{bmatrix}$$

Nous avions choisi les mêmes masses (0.1 Kg) et mêmes inerties pour toutes les pièces ce qui nous a conduit à la matrice d'inertie suivante :

$$M=\begin{bmatrix}
0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01
\end{bmatrix}$$

et donc

$$AM^{-1/2}=\begin{bmatrix}
0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-3.16 & 0 & 0 & 0 & -0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 3.16 & 0 & -0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
3.16 & 0 & 0 & 0 & 0 & 0 & -3.16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -3.16 & 0 & 0 & 0 & -0.5 & 0 & 3.16 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & -10 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 3.16 & 0 & 0 & 0 & 0 & -0.5 & -3.16 & 0 & 0 & 0 & 0 & 0.5 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -3.16 & 0 & 0 & 0 & 0 & 0 & 3.16 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3.16 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3.16 & -1 & 0 & 0
\end{bmatrix}$$

Le calcul de la pseudo-inverse de cette matrice en utilisant la méthode de Courrieu implique le calcul de la matrice L :

$$\begin{bmatrix}
10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.5 & 0 & 3.16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.5 & 0 & 3.16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-10 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -10 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -3.16 & 0 & 0 & 0 & 3.16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -3.16 & 0 & 0 & 0 & 3.24 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -3.16 & -0.07 & 0 & 0 & 3.23 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -3.08 & 0 & 0 & -0.07 & 3.23 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.54 & 0.03 & 9.88 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.24 & 0.07 & 0.5 & 0 & 0.48 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 3.16
\end{bmatrix}$$

Cette matrice fait apparaître une colonne vide correspondant au degré d'hyperstatisme du système bielle manivelle. Ce degré d'hyperstatisme se retrouve bien en rotation autour de l'axe $\vec{x}$.
Cette colonne vide est supprimée pour obtenir la matrice L puis cette matrice est multipliée par sa transposée. Ce produit pourra ensuite être inversé. La matrice produit est dans ce cas une matrice 15x15 symétrique :

$$L'L=\begin{bmatrix}
200.25 & 0 & 1.58 & 0 & -100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 200.25 & 0 & 1.58 & 0 & -100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1.58 & 0 & 20 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.58 & 0 & 20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-100 & 0 & 0 & 0 & 200 & 0 & 0 & 0 & -100 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -100 & 0 & 0 & 0 & 200 & 0 & 0 & 0 & -100 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -10 & 0 & 0 & 0 & 20 & 0.24 & 0 & 0 & -10.24 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -100 & 0 & 0.24 & 20.02 & 0 & 0 & -0.02 & -9.98 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -100 & 0 & 0 & 0 & 201 & 0 & 0 & 0 & 0 & 3.16 \\
0 & 0 & 0 & 0 & 0 & 0 & -10.24 & -0.02 & 0 & 0 & 23.38 & 0.05 & 16.88 & 1.56 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -9.98 & 0 & 0 & 0.05 & 10.47 & 0.38 & 0.03 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 16.88 & 0.38 & 97.87 & 0.24 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.56 & 0.03 & 0.24 & 0.23 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.16 & 0 & 0 & 0 & 0 & 10
\end{bmatrix}$$

L'inversion de cette matrice donne :

$$(L'L)^{-1}=\begin{bmatrix}
0.01 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0.01 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.1 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0.1 & 0 & -0.01 & -0.64 & 0 \\
0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0.1 & 0 & -0.03 & 0 \\
0.01 & 0 & 0 & 0 & 0.02 & 0 & 0 & 0 & 0.02 & 0 & 0 & 0 & 0 & 0.01 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.1 & 0 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0.2 & 0 & -0.03 & 0 & 0 \\
0 & 0 & 0 & 0.1 & 0.02 & 0 & 0 & 0.19 & 0 & 0 & 0 & 0.18 & 0 & -0.05 & 0 \\
0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.03 & 0 & 0 & 0 & 0 & 0.01 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.1 & 0 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0.28 & 0.01 & -0.04 & 1.56 & 0 \\
0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0.18 & 0 & 0 & 0.01 & 0.26 & 0 & -0.11 & 0 \\
0 & 0 & -0.01 & 0 & 0 & 0 & -0.03 & 0 & 0 & 0 & -0.04 & 0 & 0.01 & 0.28 & 0 \\
0.01 & 0 & -0.64 & -0.03 & 0.01 & 0 & 0 & -0.05 & 0.01 & 0 & 1.56 & -0.11 & 0.28 & 16.6 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1
\end{bmatrix}$$

La pseudo-inverse donne alors :

$$(AM^{-1/2})^{+}=\begin{bmatrix}
0.01 & 0 & -0.31 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & -0.01 & 0 & 0.31 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & -0.07 & 0 & 0 & 0 & 0.02 & 0 & 0 & 0 & 0.02 & 0 & 0 & 0 & -0.02 & 0 & 0\\
-0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & -0.09 & -0.09 & 0 & 0 & -0.09 & -0.09 & 0 & 0 & -0.09 & -0.04 & 0 & 0 & 0.09 & 0\\
0.01 & 0 & -0.31 & 0 & 0 & 0 & -0.31 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-0.01 & -0.01 & 0.3 & 0.3 & 0 & 0 & 0.3 & 0.3 & 0 & 0 & 0.3 & 0 & 0.01 & 0 & -0.3 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & -0.05 & 0 & 0 & 0 & -0.05 & 0 & 0 & 0 & 0.05 & 0 & 0 & 0 & -0.05 & 0 & 0\\
-0.1 & 0 & 0 & 0 & -0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0.1 & 0 & -2 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & -2 & 0 & -0.1 & 0 & 2 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.03 & 0 & -0.31 & 0\\
-0.01 & -0.01 & 0.3 & 0.3 & 0 & 0 & 0.3 & 0.3 & 0 & 0 & 0.3 & 0.3 & 0.01 & 0 & -0.3 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.02 & 0 & -0.31\\
0 & -0.02 & 0 & 0 & 0 & -0.02 & 0 & 0 & 0 & -0.02 & 0 & 0 & 0 & -0.07 & 0 & 0\\
-0.1 & 0 & 0 & 0 & -0.1 & 0 & 0 & 0 & -0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0
\end{bmatrix}$$

Avant de déterminer les accélérations, on multiplie la pseudo-inverse à gauche par $M^{-1/2}$:

$$M^{-1/2}(A\,M^{-1/2})^{+}=\begin{bmatrix}
0.05 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & -0.03 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & -0.75 & 0 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0.25 & 0 & 0 & 0 & -0.25 & 0 & 0\\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0.04 & 0.03 & -0.95 & -0.95 & 0 & -0.01 & -0.95 & -0.95 & 0 & -0.01 & -0.95 & -0.47 & -0.04 & 0.01 & 0.95 & 0\\
0.05 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-0.04 & -0.03 & 0.95 & 0.95 & 0 & 0.01 & 0.95 & 0.95 & 0 & 0.01 & 0.95 & -0.02 & 0.04 & -0.01 & -0.95 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & -0.5 & 0 & 0 & 0 & -0.5 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & -0.5 & 0 & 0\\
-1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
1 & 0 & -20 & 0 & 0 & 0 & -20 & 0 & 0 & 0 & -20 & 0 & -1 & 0 & 20 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & -1 & 0\\
-0.04 & -0.03 & 0.95 & 0.95 & 0 & 0.01 & 0.95 & 0.95 & 0 & 0.01 & 0.95 & 0.97 & 0.04 & -0.01 & -0.95 & 0\\
0 & 0.02 & 0 & 0 & 0 & 0.02 & 0 & 0 & 0 & 0.02 & 0 & 0 & 0 & 0.07 & 0 & -1\\
0 & -0.25 & 0 & 0 & 0 & -0.25 & 0 & 0 & 0 & -0.25 & 0 & 0 & 0 & -0.75 & 0 & 0\\
-1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}$$

Il reste à déterminer $(b-A\,a)$, comme nous n'avons soumis le mécanisme qu'à de la gravité :

$$a=\begin{bmatrix}
0\\ -9.81\\ 0\\ 0\\ 0\\ 0\\ 0\\ -9.81\\ 0\\ 0\\ 0\\ 0\\ 0\\ -9.81\\ 0\\ 0\\ 0\\ 0
\end{bmatrix}\Rightarrow b-Aa=\begin{bmatrix}
0\\ 0\\ 0\\ 9.81\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0
\end{bmatrix}\Rightarrow M^{-1/2}(A\,M^{-1/2})^{+}(b-Aa)=\begin{bmatrix}
0\\ 9.81\\ 0\\ 0\\ 0\\ -9.34\\ 0\\ 9.34\\ 0\\ 0\\ 0\\ 0\\ 0\\ 9.34\\ 0\\ 0\\ 0\\ 0
\end{bmatrix}\Rightarrow \ddot{x}=\begin{bmatrix}
0\\ 0\\ 0\\ 0\\ 0\\ -9.34\\ 0\\ -0.47\\ 0\\ 0\\ 0\\ 0\\ 0\\ -0.47\\ 0\\ 0\\ 0\\ 0
\end{bmatrix}$$

Dans cette configuration particulière seule la quatrième contrainte (arrêt en translation sur y dans la liaison pivot glissant bâti/manivelle) va avoir une influence sur les accélérations, les autres composantes de $(b-A\,a)$ étant nulles.

A.6.2 Système bielle manivelle avec liaisons et choc

Nous ne présenterons que la matrice A et le vecteur b pour les contraintes juste au moment du choc :

$$A=\begin{bmatrix}
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & -0.05 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & -0.05 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0.03 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & -0.05 & 1 & 0 & 0 & 0 & 0 & 0.03 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 & 0.7 & 0 & 0 & 0 & 0 & -0.7 & -0.7 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 & -0.7 & 0 & 0 & 0 & 0 & -0.7 & 0.7 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.7 & -0.7 & 0 & 0 & 0 & -0.05 & -0.7 & 0.7 & 0 & 0 & 0 & 0.03 \\
0 & 0 & 0 & 0 & 0 & 0 & -0.7 & -0.7 & 0 & 0 & 0 & 0 & 0.7 & 0.7 & 0 & 0 & 0 & -0.03 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0.1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -0.1 & 0 & 0 \\
0 & -1.0 & 0 & 0.08 & 0 & 0.03 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \quad et \quad b=\begin{bmatrix}
0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.18 \\ 0.1 \\ 0 \\ 0 \\ 0 \\ -0.14 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1425200
\end{bmatrix}$$

La dernière ligne correspond au choc avec un $dt=10^{-7}s$ et un coefficient d'absorption de 0.99 d'où la valeur très élevée correspondante dans le vecteur $\boldsymbol{b}$: 1425200. Cette accélération n'est utilisée que pour déterminer la vitesse après choc, on peut écrire que l'incrément en vitesse est l'accélération multipliée par dt.

$$\boldsymbol{x}=\begin{bmatrix}
0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 47506664 \\ 1187670 \\ -1187668 \\ 0 \\ 0 \\ 0 \\ 33592284 \\ 0 \\ -5 \\ 0 \\ 0 \\ 0 \\ 0
\end{bmatrix} \Rightarrow \Delta \boldsymbol{V}=\boldsymbol{x}\,x\,1.10^{-7}=\begin{bmatrix}
0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 4.75 \\ 0.11 \\ -0.11 \\ 0 \\ 0 \\ 0 \\ 3.35 \\ 0 \\ -5 \\ 0 \\ 0 \\ 0 \\ 0
\end{bmatrix} \Rightarrow \boldsymbol{V}_{avant}=\begin{bmatrix}
0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -2.38 \\ -0.06 \\ 0.06 \\ 0 \\ 0 \\ 0 \\ -1.68 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0
\end{bmatrix} \quad devient \quad \boldsymbol{V}_{après}=\begin{bmatrix}
0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2.36 \\ 0.06 \\ -0.06 \\ 0 \\ 0 \\ 0 \\ 1.67 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0
\end{bmatrix}$$

La combinaison de contraintes de liaison avec des contraintes de choc est donc parfaitement gérée avec cette formulation.

Bibliographie

[AB01] Abdel-Malek, Blackmore, Joy, Swept Volumes: Foundations, Perspectives, and Applications, International Journal of Shape Modeling Vol. 12, No. 1 World Scientific Publishing Company 87-127 (2006)

[AK01] Akao Y., QFD prendre en compte les besoins du client dans la conception du produit, Association Française de Normalisation (2000), ISBN : 978-2124683116

[AL01] Christine J. Alvarado, A Natural Sketching Environment: Bringing the Computer into Early Stages of Mechanical Design, Department of Electrical Engineering, Massachusetts Institute of Technology, May (2000)

[AL02] John Alex, Hybrid Sketching: A New Middle Ground Between 2- and 3-D, Department of Achitecture, Massachussets Institute of Technology, February (2005)

[AL03] Allard, Cotin, Faure, Bensoussan, Poyer, Duriez, Delingette, Grisoni, SOFE - An Open Source Framework for Medical Simulation, Medecine Meets Virtual Reality MMVR'15 (2007)

[AN01] Angelidis, Cani, Adaptive Implicit Modeling using Subdivision Curves and Surfaces as Skeletons, Proceedings of the seventh ACM symposium on Solid modeling and applications 45-52 (2002)

[AP01] Apitz, Guimbretière, CrossY: a crossing-based drawing application, Proceedings of the 17th annual ACM symposium on User interface software and technology ACM Press 3-12 (2004)

[AT01] Atman, Cardella, Turns, Adams, Comparing freshman and senior engineering design processes: an in-depth follow-up study, Design Studies 26 Elsevier 325-357 (2005)

[AT02] Atkins, Fiser, Jacobs, Experience-dependent visual cue integration based on consistencies between visual and haptic percepts, Vision Research 41 Pergamon 449-461 (2001)

[BA01] Barbic, James, Real-Time Subspace Integration for St Venant-Kirchhoff Deformable Models, ACM Transactions on Graphics (SIGGRAPH 2005) Vol. 24 N. 3 ACM Press 982-990 (2005)

[BA02] Barone, The Process Improvement Impact on the Styling Workflow of

an Industrial Design Company, Proceedings of the Eurographics Workshop on Sketch-Based Interfaces and Modeling 2004 (SBM'04) (2004)

[BA03] Basdogan, Lum, Salcedo, Chow, Kupiec, Kostrewski, Autostereoscopic and Haptic Vizualization for Space Exploration and Mission Design, Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002 271-276 (2002)

[BA04] Bartz, Wu, Advanced Virtual Medecine: Techniques and Applications for Medecine-oriented Computer Graphics, Tutorial 6 25th Annual Conference of the European Association for Computer Graphics, Grenoble (2004)

[BA05] Baraff, Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies, Computer Graphics 23, Number 3 223-232 (1989)

[BA06] Xavier Baguenard, Propagation de contraintes sur les intervalles, application à l'étalonnage des robots, L.I.S.A., Université d'Angers, décembre (2005)

[BA07] Barbier, Galin, Fast distance computation between a point and cylinders, cones, line swept spheres and cone-spheres, Journal of Graphics tools Volume 9, Number 2 11-19 (2004)

[BA08] Bascoul C., Pour une Conception Mécanique Assistée par Ordinateur, 4° Colloque sur la Conception Mécanique Intégrée La Plagne 119-126 (1995)

[BA09] Bascoul C., Ray P., Virtual Sketch in Mechanical Design, EUROMECH Colloquium 476 Real-time Simulation and Virtual Reality Applications of Multibody Systems (2006)

[BA10] Bascoul C., Gogu G., Ray P., Virtual Sketch in Mechanical Design, Proceedings of the Workshop Virtual Reality Technology and Applications Transilvania University of Brasov (2006)

[BA11] Bascoul C., Ray P., Virtual Sketch in Mechanical Design, Proceedings of Virtual Concept 2006 (2006)

[BE01] Bedny, Seglin, Meister, Activity Theory: history, research and application, Theoretical Issues in Ergonomics Sciencs 1, No 2 Taylor & Francis 168-206 (2000)

[BE02] Benami, Jin, Creative stimulation in Conceptual Design, ASME Design Engineering Technical Conferences and Computer and Information in Engineering Conference DETC2002/DTM-34023 1-13 (2002)

[BI01] Bimber, Raskar, Modern Approches to Augmented Reality,

EUROGRAPHICS 2004 Tutorial 8 (2004)

[BI02] Bilda, Demirkan, An insight on designers' sketching activities in traditional versus digital media, Design Studies 24 Elsevier 27-50 (2003)

[BI03] Bimber, Encarnacao, Stork, A multi-layered architecture for sketch-based interaction within virtual environments, Computer & Graphics 24 Pergamon 851-867 (2000)

[BI04] Bilda, Gero, Purcell, To sketch or not to sketch? That is the question, Design Studies 27 Elsevier 587-613 (2006)

[BL01] Blanch, Ferley, Cani, Gascuel, Non-Realistic Haptic Feedback for Virtual Sculpture, Evasion, INRIA Rhone-Alpes (2004)

[BL02] Blundell, Schwartz, Creative 3-D display and interaction interfaces, Wiley (2005), ISBN : 0-471-48271-4

[BO01] Bordegoni, Cugini, Design Products with your hands, Proceedings of Virtual Concept Vol. 1 Springer (2005)

[BO02] Bourguignon, Chaine, Cani, Drettakis, Relief, A Modeling by Drawing Tool, Eurographics Workshop on Sketch-Based Interfaces and Modeling Eurographics Digital Library (2004)

[BO03] Bordegoni, Colombo, Formentini, Haptic technologies for the conceptual and validation phases of product design, Computers & Graphics 30 Elsevier 377-390 (2006)

[BR01] Brostow, Essa, Steedly, Kwatre, Novel Skeletal Representation for Articulated Creatures, Computer Vision - ECCV 2004, LNCS 3023 Volume 3023, Part III Springer Berlin / Heidelberg 66-78 (2004)

[BR02] Bradshaw, O'Sullivan, Adaptive Medial-Axis Approximation for Sphere-Tree Construction, ACM Transactions on Graphics Vol. 23, No. 1 ACM Press 1-26 (2004)

[BR03] Joel Brown, Real-Time Soft Tissue and Suture Simulation, Department of Computer Science, Stanford University, November (2003)

[BR04] Brown, Sorkin, Bruyns, Latombe, Montgomery, Stephanides, Real-Time Simulation of Deformable Objects: Tools and Application, Computer Animation, 2001. The Fourteenth Conference on Computer Animation. Proceedings 228-258 (2001)

[BR05] Bruyninckx, Khatib, Gauss' Principle and Dynamics of Redundant

and Constrained Manipulators, Proceedings of the 2000 I.C.R.A. Volume: 3 2563-2568 (2000)

[BU01] Burak Kara, Stahovich, An image-based, trainable symbol recognizer for hand-drawn sketches, Computers & Graphics 29 Elsevier 501-517 (2005)

[BU02] Burdea G., Coiffet P., La réalité virtuelle, Hermes Sciences Publicat. (1993), ISBN : 978-2866013868

[CA01] Yves Cartonnet, Les relations entre la vision stéréoscopique et les activités cognitives de conception en bureau d'études mécaniques, Lettres et sciences humaines, Université de Provence, 8 décembre (1992)

[CA02] Carr, Beatson, Cherrie, Mitchell, Fright, McCallum, Reconstruction and Representation of 3D Objects with Radial Basis Functions, Computer Graphics Proceedings SIGGRAPH 2001 67 - 76 (2001)

[CA03] Carkett, 'He's different, he's got 'Star Trek' vision': supporting the expertise of conceptual design engineers, Design Studies 25 Elsevier 459-475 (2004)

[CH01] Chablat, Wenger, A New Six Degree-of-Freedom Haptic Device based on the Orthoglide and the Agile Eye, Proceedings of Virtual Concept Vol. 1 Springer (2005)

[CH02] Chimka, Altman, Graphical Representation of Engineering Design Behavior, Frontiers in Education Conference, 1998. FIE '98. 28th Annual Vol. 1 160-165 (1998)

[CH04] Choi, Wang, Kim, Exact Collision Detection of Two Moving Ellipsoids Under Rational Motions, Computer Graphics Group, University of Hong Kong (2002)

[CH05] Chuang, Ahuja, Lin, Tsai, Chen, A potential-based generalized cylinder representation, Computers & Graphics 28 Elsevier 907-918 (2004)

[CH06] Chen, Yang, Lian, On the development of a haptic system for rapide product development, Computer-Aided Design 37 Elsevier 559-569 (2005)

[CH07] Choi, Sun, Heng, An efficient and scalable deformable model for virtual reality-based medical applications, Artificial Intelligence in Medecine 32 Elsevier 51-69 (2004)

[CH08] Chen, Chang, Chien, Chou, Explicit Equations of Motion for Dynamical Systems with Multiple Constraints, Japanese Journal of Applied Physics 45, N° 6A 5286-5292 (2006)

[CH09] Chen, Yong, Zhng, Paul, Sun, Computing minimum distance between two implicit algebraic surfaces, Computer-Aided Design 38 Elsevier 1053-1061 (2006)

[CH10] Chevalier, Lecrinier, Dessin de construction de l'analyse à la définition, Hachette (1972)

[CI01] Joselita Ciaravino, Un art paradoxal, la notion de disegnio en Italie (XVème-XVIème siècles), L'Harmattan (2004), ISBN : 2-7475-7132-7

[CO01] Cornea, Silver, Yuan, Balasubramanian, Computing Hierarchical Curve-Skeletons of 3D Objects, Pattern Analysis and Machine Intelligence 22(11) IEEE Trans. 1241-1251 (2000)

[CO02] Cornea, Silver, Min, Curve-Skeleton Applications, IEEE Visualization Conference (Vis'05) (2005)

[CO03] Company, Piquer, Contero, On the Evolution of Geometrical Reconstruction as a Core Technology to Sketch-Based Modeling, Eurographics Workshop on Sketch-Based Interfaces and Modeling Eurographics Digital Library (2004)

[CO04] Jonathan Cohen, Systems for Sketching in 3D, Senior thesis, Brown University, May (2000)

[CO05] Cotin, Delingette, Ayache, A Hybrid Elastic Model allowing Real-Time Cutting, Deformations and Force-Feedback for Surgery Training and Simulation, "Visual Computer" Journal 16, No 8 437-452 (2000)

[CO06] Stéphane Cotin, Modèles anatomiques déformables en temps réel, INRIA Sophia Antiplois, Université de Nice Sophia Antipolis, novembre (1997)

[CO07] Conti, Khatib, Baur, Interactive rendering of deformable objects based on a filling sphere modeling approach, Proceedings of the 2003 IEE I.C.R.A 3716-3721 (2003)

[CO08] Courrieu, Fast Computation of Moore-Penrose Inverse Matrices, Neural Information Processing - Letters and Reviews 8, N° 2 25-29 (2005)

[CR01] Cross, Creative Cognition in Design: Processes of Exceptional Designers, Proceedings of the 4th conference on Creativity & cognition ACM Press 14-19 (2002)

[CR02] Cross, Expertise in design: an overview, Design Studies 25 Elsevier 427-441 (2004)

[CR03] Nigel Cross, Comprendre la pensée du concepteur, Mardaga (2002), ISBN : 2-87009-803-0

[CR04] Nigel Cross, Engineering Design Methods, John Wiley & Sons (1989), ISBN : 0-471-92215-3

[CS01] Csabai, Stroud, Xirouchakis, Container spaces and functional features for top-down 3D layout design, Computer_Aided Design Elsevier 1011-1035 (2002)

[DA01] Davis, Sketch Understanding in Design: Overview of Work at the MIT AI Lab, 2002 AAAI Spring Symposium Technical Report SS-02-08 The AAAI Press 24-31 (2002)

[DA02] Dachille, Qin, Kaufman, A novel haptics-based interface and sculpting system for physics-based geometric design, Computer-Aided Design 33 Elsevier 403-420 (2001)

[DA03] Frédéric Danesi, Contribution à la définition et à l'intégration de notions intuitives dans le dialogue Homme - Machine en C.A.O., Institut de Formation Technique Supérieure, Université de Reims Champagne - Ardenne, novembre (2002)

[DA04] Darses, Détienne, Visser, Les activités de conception et leur assistance, Ergonomie, P.U.F., (2004), ISBN : 2-13-051404-9, 545-563

[DE01] Debrunne, Desbrun, Cani, Barr, Dynamic Real-Time Deformations using Space & Time Adaptative Sampling, Computer Graphics Proceedings Annual Conference Series ACM Press / ACM SIGGRAPH (2001)

[DE02] Mathieu Desbrun, Modélisation et Animation de Matériaux Hautement Déformables en Synthèse d'Images, iMAGIS/GRAVIR-IMAG, Institut National Polytechnique de Grenoble, Décembre (1997)

[DE03] De Boeck, Raymaeckers, Van Reeth, Introducing Touch in a Rigid Body Simulation Environment: a Design Overview, Proceedings of PURS 2000 1-8 (2000)

[DE04] Dey, Zhao, Approximate Medial Axis as a Voronoi Subcomplex, Proc. 7th ACM Sympos. Solid Modeling and Applications 356-366 (2002)

[DE05] Demirbilek, Demirkan, Universal product design involving elderly users: a participatory design model, Applied Ergonomics 35 Elsevier 361-370 (2004)

[DE06] De Sapio, Khatib, Task-Level Approaches for the Control of

Constrained Multibody Systems, MULTIBODY SYSTEM DYNAMICS VOL 16; NUMBER 1 Springer Science 73-102 (2006)

[DE07] De Falco, Pennestri, Vita, The Udwadia-Kalabia Formulation: A Report on its Numerical Efficiency in Multibody Dynamics Simulations and on its Teaching Effectiveness, Multibody Dynamics 2005 ECCOMAS Thematic Conference (2005)

[DE08] De Varennes, Le principe gaussien de la moindre contrainte n'est pas exact, Portugaliae Mathematica 14 (1955)

[DI01] Diehl, Lindemann, Realisation of a low cost VR-CAD-System using Open Source Software, Proceedings of Virtual Concept 2005 Vol. 1 Springer (2005)

[DI02] Diehl, Müller, Lindemann, From raw 3D-Sketches to exact CAD product models - Concept for an assistant-system, Eurographics Workshop on Sketch-Based Interfaces and Modeling Eurographics Digital Library (2004)

[DI03] Diehl, Bremer, Lindemann, Learning how to sketch in three dimensions, Research in Interactive Design Vol. 2 Springer (2006)

[DI04] G. E. Dieter, Engineering Design. A materials and Processing Approach, McGraw-Hill (1983), ISBN : 978-0073661360

[DO01] DO, Design sketches and sketch design tools, Knowledge-Based Systems 18 Elsevier 1-23 (2005)

[DO02] DO, Sketch That Scene For Me - Creating Virtual Worlds by Freehand Drawing, Proceedings of the 18th conference on Education in Computer Aided Architectural Design in Europe 265-268 (2000)

[DO03] DO, Drawing marks, acts, and reacts: Toward a computational sketching interface for architectural design, Artificial Intelligence for Engineering Design, Analysis and Manufacturing 16 Cambridge University Press 149-171 (2002)

[DO04] Döring, Brix, Reessing, Application of computational kinematics in the digital mechanism and gear library DMG-Lib, Mechanism and Machine Theory 41 Elsevier 1003-1015 (2006)

[DO05] Dominjon, Lécuyer, Burkhardt, Richard, Richir, Influence of Control/Display Ration on the Perception of Mass of Manipulated Objects in Virtual Environments, Proceedings of the IEEE International Conference on Virtual Reality (2005)

[DO06] Dorfmüller, Wirth, Real-Time Hand and Head Tracking for Virtual Environments Using Infrared Beacons, Lecture Notes In Computer Science Vol. 1537 Springer-Verlag 113-127 (1998)

[DR01] Druesne, Dulong, Villon, Real time simulation of non linear mechanical model, Proceedings of Virtual Concept 2005 Vol. 1 Springer (2005)

[DR02] Druon, Crosnier, Brigandat, Efficient Cellular Automata for 2D/3D Free-Form Modeling, Journal of WSCG Vol. 11, No. 1 102-108 (2003)

[DU01] Dumitrescu, Catalano, Giannini, Falcidieno, Vergeest, Curve and skeleton based shape deformations to support product design, Proceedings of CAID & CD 2005 Conference 302-307 (2005)

[DU02] Christion Duriez, Contact frottant entre objets déformables dans des simulations temps-réel avec retour haptique, Laboratoire IBISC, Université d'Evry, Décembre (2004)

[DU03] Laurent Dupont, Paramétrage quasi-optimal de l'intersection de deux quadriques : théorie, algorithme et implantation, UFR Mathématique et Informatique, Université Nancy 2, octobre (2004)

[DU04] Du, Qin, A shape design system using volumetric implicit PDEs, Computer-Aided Design Vol. 36, Issue 11 Elsevier 1101-1116 (2004)

[EG01] Kevin Egan, Techniques for Real-Time Rigid Body Simulation, Department of Computer Science, Brown University, (2003)

[ER01] Kenny Erleben, Stable, Robust, and Versatile Multibody Dynamics Animation, The Department of Computer Science, University of Conpenhagen, November (2004)

[EV01] Evans, Cheshire, Dean, An investigation into the use of haptic modelling during industrial design activity, IDATER conference 2000 Loughborough University (2000)

[FA01] Farrugia, Borg, Camlleri, Spiteri, Bartolo, A Cameraphone-Based Approach for the Generation of 3D Models from Paper Sketches, Eurographics Workshop on Sketch-Based Interfaces and Modeling Eurographics Digital Library 33-42 (2004)

[FE01] Ferley, Cani Gascuel, Attali, Skeletal Reconstruction of Branching Shapes, Implicit Surfaces John C. Hart and Kees van Overveld 127-142 (1996)

[FE02] Ferley, Cani, Gascuel, Resolution Adaptive Volume Sculpting, Graphical Models (GMOD) Volume 63 459-478 (2002)

[FE03] Fernandes, Raja, Eyre, Immersive learning system for manufacturing industries, Computers in Industry 51 Elsevier 31-40 (2003)

[FR01] Frisken, Perry, Creature Feature: An Intuitive System for Designing 3D Models, Mitsubishi Electric Research Laboratories, (2003)

[FU01] Funkhauser, Min, Kazhdan, Chen, Halderman, Dobkin, A search engine for 3D models, ACM Transact ions on Graphics Volume 22 , Issue 1 ACM Press 83-105 (2003)

[FU02] Fuchs P., Moreau G., Le traité de la réalité virtuelle. Troisième édition en quatre volumes, Presse de l'Ecole des Mines (2006), ISBN : 2-911762-62-2

[GA01] Gao, Zhang, A C-tree decomposition algorithm for 2D and 3D geometric constraint solving, Computer-Aided Design 38 1-13 (2006)

[GA02] Ganovelli, Dingliana, O'Sullivan, BucketTree: Improving Collision Detection Between Deformable Objects, Proceedings of SCCG2000 156 - 163 (2000)

[GA03] Gauss, Über ein neues allgemeines Grundgesetz der Mechanik, Journal für die reine und angewandte Mathematik 4 (1829)

[GE01] Gero, Kannengiesser, The situated function-behaviour-structure framework, Design Studies 25 Elsevier 373-391 (2004)

[GE02] Ge, An Error in Gauss' and Appell's Proof of Gauss' Principle of Least Constraint, Chinese Society of Mechanical Engineers 22; part 4 281-286 (2001)

[GI02] Giannini, Monti, Design intent-oriented modelling tools for aesthetic design, Journal of WSCG 11, N°1, WSCG'2003 (2003)

[GO01] Götekin, cavusoglu, Tendick, Sastry, GiPSi: An Open Source/Open Architecture Software Development Framework for Surgical Simulation, gipsi.case.edu/gipsi.html, Case Western Reserve University (2004)

[GO02] Goldschmidt, Smolkov, Variances in the impact of visual stimuli on design problem solving performance, Design Studies 27 Elsevier 549-569 (2006)

[GR01] Greer, Wood, Jensen, Effort flow analysis: a methodology for directed product evaluation, Design Studies 25 Elsevier 193-214 (2004)

[GR02] Gregory, Ehmann, Lin, inTouch: Interactive Multiresolution Modeling and 3D Painting with a Haptic Interface, Proceedings of IEEE Virtual Reality Conference 2000 (2000)

[GR03] Gregory, Lin, Gottscjalk, Taylor, Fast and accurate collision detection for haptic interaction using a three degree-of-freedom force-feedback device, Computational Geometry 15 Elsevier 69-89 (2000)

[HA01] Hauth, Numerical Techniques for Soft Object Simulation, Tutorial 6 (2004)

[HA02] Hasegawa, Sato, Real-time Body Simulation for Haptic Interactions Based on Contact Volume of Polygonal Objects, Eurographics 2004 23, Number 3 (2004)

[HA03] Han, Lee, A case-based framework fir reuse of previous design concepts in conceptual synthesis of mechanisms, Computers in Industry 57 Elsevier 305-318 (2006)

[HE01] Hezen, Ailenei, Di Fiore, Van Reeth, Patterson, Sketching with a Low-latency Electronic Ink Drawing Tablet, Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia ACM Press 51-60 (2005)

[HI01] Hirayama, Aoyama, 3D modelling system based on line sketch, Proceedings of Virtual Concept 2005 (2005)

[HI02] Laurent Hilde, Algorithmes de résolution des équations du mouvement pour l'animation basée sur la physique, Laboratoire d'Informatique Fondamentale de Lille, Université des Sciences et Technologies de Lille, 20 décembre (2002)

[HI03] Hisada, Belyaev, Kunii, A Skeleton-based Approach for Detection of Percptually Salient Features on Polygonal Surfaces, Computer Graphics forum 21 Blackwell Publishers 689-700 (2002)

[HI04] Hitzer, Kamiwaai - Interactive 3D Sketching with Java based on CL(4,1) Conformal Model of Euclidean Space, Advances in Applied Clifford Algebras 13 No. 1 11-45 (2003)

[HI05] Hirtz, Stone, McAdams, Szykman, Wood, A Functional Basis for Engineering Design: Reconciliation and Evolving Previous Efforts, NIST Technical Note 1447, (2002)

[HI06] Hitzer, Kamiwaai - Interactive 3D sketching with Java based on Cl(4,1) conformal model of euclidean space, Advances in Applied Clifford Algebras 13 No. 1 11-45 (2003)

[HO01] Horvath, van der Vegte, Nucleus-based Product Conceptualization --

Part 1: Principles and Formalization, International Conference on Engineering Design ICED 03 (2003)

[HO02] Horvath, van der Vegte, Nucleus-based Product Conceptualization -- Part 2: Application in Designing for Use, International Conference on Engineering Design ICED 03 (2003)

[HO03] Hoeben, Stappers, Direct talback in computer supported tools for the conceptual stage of design, Knowledge-Based Systems XX Elsevier (2005)

[HO04] Horvath, Vergeest, Kczogi, Development and Application of Design Concept Ontologies for Contextual Conceptualization, Proc. of 1998 ASME Design Engineering Technical Conferences DETC98/CIE-5701 (1998)

[HO05] Hoffman, Lomonosov, Sitharam, Decomposition Plans for Geometric Constraint Systems, Part I: Performance Measures for CAD, Journal of Symbolic Computation 31 367-408 (2001)

[HO06] Hoffman, Lomonosov, Sitharam, Decomposition Plans for Geometric Constraint Systems, Part II: New Algorithms, Journal of Symbolic Computation 31 409-427 (2001)

[HO07] Ho, Eastman, An investigation of 2D and 3D spatial and mathematical abilities, Design Sudies 27 Elsevier 505-524 (2006)

[HS01] Hsiao, Huang, A neural network based approach for product design, Design Studies 23 Elsevier 67-84 (2002)

[HU01] Huang, Way, Shih, Physical-based Model of Ink Diffusion in Chinese Ink Paintings, Proceedings of WSCG 2003 (2003)

[HU02] Stéphane Huot, Une nouvelle approche pour la conception créative : De l'interprétation du dessin à main levée au prototypage d'interactions non-standard, ENSTIM de Nantes, Université de Nantes, juillet (2005)

[HU03] Huang, Bo, Chen, An integrated computational intelligence approach to product concept generation and evaluation, Mechanism and Machine Theory 41 Elsevier 567-583 (2006)

[JA01] Robert Carl Jagnow, Virtual Sketching with Haptic Displacement Maps, Department of Electrical Engineering, Massachussets Institute of Technology, June (2001)

[JA02] James, Pai, A Unified Traitement of Elastostatic Contact Simulation for Real Time Haptics, Haptics-e 2, Number 1 (2001)

[JA03] James, Pai, BD-Tree: Output-Sensitive Collision Detection for Reduced Deformable Models, ACM Transactions on Graphics 23 ACM Press 393-398 (2004)

[JA04] Jansson, Vergeest, A discrete mechanics model for deformable bodies, Journal of Computer-Aided Design Vol. 34, N° 12 919-928 (2002)

[JA05] Johan Jansson, Automated Computational Modeling, Department of Applied Mechanics, Chalmers University of Technology, (2006)

[JA06] Jansson, Vergeest, Kuczogi, Horvath, Combining Deformable and Rigid Body Mechanics Simulation, Computer Animation, 2001 Proceedings of the Fourteenth Conference on Computer Animation 147-156 (2003)

[JE01] Christophe Jermann, Résolution de contraintes géométriques par rigidification récursive et propagation d'intervalles, UFR Sciences, Université de Nice - Sophia Antipolis, décembre (2002)

[JO01] Jonson, Design ideation: the conceptual sketch in the digital age, 26 613-623 (2005)

[JO02] Jones, Bokinsky, Tretter, Negishi, A comparaison of learning with haptic and visual modalities, The Electronic Journal of Haptics Research Vol. 3, No. 6 (2005)

[JO03] Johnson, Cohen, Unified Distance Queries in a Heterogeneous Model Environment, Proceedings of DETC2004 DETC2004/DAC-57461 (2004)

[JO04] Johnson, Cohen, Distance Extrema for Spline Models Using Tangent Cones, Proceedings of Graphics Interface 2005 Canadian Human-Computer Communications Society 169-175 (2005)

[JO05] Joskowicz, Sacks, Computer-Aided Mechanical Design Using Configuration Spaces, IEEE Computers in Science and Engineering Nov/Dec 1999 14-21 (1999)

[KA01] Kallio, 3D6B Editor: Projective 3D Sketching with Line-Based Rendering, 2nd Eurographics Workshop on Sketch-Based Interfaces and Modeling (2005)

[KA02] Karpenko, Hughes, Raskar, Free-from Sketching with Variational Implicit Surfaces, Computer Graphics Forum 21 585-594 (2002)

[KA03] Kavan, Zara, Fast Collision Detection for Skeletally Deformable Models, EUROGRAPHICS 2005 Vol. 24, No. 3 Blackwell Publishing 363-372 (2005)

[KA04] Kavan, Zara, Spherical Blend Skinning: A Real-time Deformation of Articulated Models, Proceedings of the 2005 symposium on Interactive 3D graphics and games ACM Press 9-16 (2005)

[KA05] Kalabia, Udwadia, Xu, Constrained Motion Revisited, Applied Mathematics and Computation 70 Elsevier 67-76 (1995)

[KA07] Kaufman, Edmunds, Pai, Fast Frictional Dynamics for Rigid Bodies, ACM Transactions on Graphics Volume 24 , Issue 3 ACM Press 946-956 (2005)

[KA08] Kalabia, Udwadia, Analytical dynamics with constraint forces that do work in virtual displacements, Applied Mathematics and Computation 121 211-217 (2001)

[KH01] Kho, Garland, Sketching Mesh Deformation, Proceedings of the 2005 symposium on Interactive 3D graphics and games 147-154 (2005)

[KI01] Kim, Neumann, Interactive Multiresolution Hair Modeling and Editing, ACM Transactions on Graphics Volume 21 , Issue 3 ACM Press 620-629 (2002)

[KI02] Arthur Kirkpatrick, Force plus graphics is not equal to vision plus haptics: Towards usable haptic environments, Department of Computer and Information Science, University of Oregon (2000)

[KI03] Kim, Rossignac, Collision Prediction for Polyhedra under Screw Motions, Proceedings of the eighth ACM symposium on Solid modeling and applications ACM Press 4-10 (2003)

[KI04] Axel Kilian, Defining Digital Space Through a Visual Language, Department of Architecture, M.I.T., February (2000)

[KN01] Kneller, Constraints in discrete mechanical systems, Workshop on Constraints ERCIM/COMPULOG (1998)

[KO01] Kolcarek, Sochor, Haptic Rendering using Velocity Driven Level of Detail, Virtual reality interaction and physical simulation (VRIPHYS'04) 149-158 (2004)

[LA01] Larsen, Gottschalk, Lin, Manocha, Fast Promimity Queroes with Swept Sphere Volumes, , Department of Computer Science (1999)

[LA02] Joseph J. La Viola, Mathematical Sketching: A New Approach to Creating and Exploring Dynamic Illustrations, Department of Computer Science, Brown University, May (2005)

[LA03] Caroline Larboulette, Traitement temps-réel des déformations de la peau et des tissus sous-cutanés pour l'animation de personnages, SIAMES - EVASION, INSA de Rennes, novembre (2004)

[LA04] Larsson, Akenine-Möller, Collision Detection for Continuously Deforming Bodies, Eurographics Conference 2001 325-333 (2001)

[LA05] Lai, Chang, A distributed linking system for supporting idea association during the conceptual design stage, Design Studies 27 Elsevier 685-710 (2006)

[LA06] Laville F., La cognition située: Une nouvelle approche de la rationalité limitée, Revue économique Vol. 51, No. 6 1301-1331 (2000)

[LE01] Lennerz, Schömer, Efficient Distance Computation for Quadric Curves and Surfaces, Proceedings of the Geometric Modeling and Processing - Theory and Applications (GMP'02) IEEE Computer Society 60 (2002)

[LE02] Lennerz, Schönmer, Warken, A Framework for Collision Detection and Response, 11th European Simulation Symposium and Exhibition (ESS'99) 309-314 (1999)

[LE03] Anatole Lécuyer, Contribution à l'étude des retours haptique et pseudo-haptique et de leur impact sur les simulations d'opérations de montage/démontage en aéronautique, Laboratoire de Robotique de Versailles, Université Paris XI, décembre (2001)

[LE04] Lécuyer, McIntyre, Mégard, Coquillart, Gentaz, Bilan du Projet ECOVIA : Etude du Couplage Visuo-haptique, Journées ROBEA : Robotiques et Entités Artificielles (2006)

[LI01] Liu, Dodds, McCartney, Hinds, Manipulation of CAD surface models with haptics based on shape control functions, Computer-Aided Design 37 Elsevier 1447-1458 (2005)

[LI02] Liu, Tang, Evolutionary design in a multi-agent design environment, Applied Soft Computing Volume 6, Issue 2 Elsevier 207-220 (2005)

[LI03] Hod Lipson, Computer Aided 3D Sketching for conceptual design, Laboratory for Computer Graphics and CAD, Israel Institute of Technology, September (1998)

[LI04] Lim, Prieto, Wright, A study of sketching behaviour to support free-form surface modelling from on-line sketching, Design Studies Volume 25, Issue 4 Elsevier 393-413 (2003)

[LI05] Udo Lindemann, Human Behavior in Design, Springer (2003), ISBN : 3-540-40632-8

[LO01] Loock, Schömer, A virtual Environment for Interactive Simulation From Rigid Bodies to Deformable Cables, 5th world multiconference on systemics, cybernetics and informatics vol. 3 325-32 (2001)

[MA01] Malik, A Sketching Interface for Modeling and Editing Hairstyles, Eurographics Workshop on Sketch-Based Interfaces and Modeling 185-194 (2005)

[MA02] Mann, Bedi, Generalization of the Imprint Method to General Surfaces of Revolution for NC Machining, Computer-aided design Vol. 34, No 5 Elsevier 373-378 (2001)

[MC01] McKoy, Vargas-Hernandez, Summers, shah, Influence od Design Representation on Effectiveness of Idea Generation, Proceedings of ASME Design Theory and Methodology Conference (2001)

[MC02] McGrown, Green, Rodgers, Visible ideas: information patterns of conceptual activity, Design Studies 19 Elsevier 431-453 (1998)

[MC03] Robert McCarter, Frank Lloyd Wright, Phaidon (2002), ISBN : 0-7148-9313-7

[ME01] Meseure, Kheddar, Faure, Détection de collision et Calcul de la réponse, Action spécifique CNRS N° 90, CNRS (2003)

[ME02] Frédéric Messine, L'Optimisation Globale par Intervalles :de l'Etude Théorique aux Applications, ENSEEIHT-IRIT, UMR-CNRS 5055, février (2006)

[ME03] Menezes, Lawson, How designers perceive sketches, Design Studies 27 Elsevier 571-585 (2006)

[MI01] Millar, Network models for haptic perception, Infant Behavior & Development Volume 28, Issue 3 Elsevier 250-265 (2005)

[NA01] Joris Naudet, Forward Dynamics of Multibody Systems: A Recursive Hamiltonian Approach, Faculteit Ingenieurswetenschappen, Vrije Universiteit Brussel, September (2005)

[NE01] Nelson, Johnson, Cohen, Haptic Rendering of Surface-to-Surface Sculpted Model Interaction, International Conference on Computer Graphics and Interactive Techniques ACM SIGGRAPH 2005 Courses ACM Press 97 (2005)

[NE02] Neff, Finding the distance between two circles in three-dimensional

space, IBM Journal of Research and Development Vol. 34, N° 5 IBM Corp. 770-775 (1990)

[NE03] Nealen, Müller, Keiser, Boxerman, Carlson, Physically Based Deformable Models in Computer Graphics, Computer Graphics Forum Volume 25, Number 4 Blackwell Publishing 809-836 (2006)

[NE04] Nelson, Cohen, Interactive Mechanical Design Variation for Haptics and CAD, Computer Graphics Forum Volume 18, Number 3 Blackwell Publishing 287-296 (1999)

[NE05] Newsome S., Spillers W., Finger S., Design Theory '88, Springer-Verlag (1988), ISBN : 3-540-96976-4

[OH01] Oh, Kim, Progressive reconstruction of 3D objects from a single free-hand line drawing, Computers & Graphics 27 Pergamon 581-592 (2003)

[OH02] Ohtake, Belyanev, Seidel, A Multi-scale Approach to 3D Scattered Data Interpolation with Compactly Supported Basis Functions, Proceedings of the Shape Modeling International 2003 IEEE Computer Society 153 (2003)

[OH03] Ohtake, Belyaev, Alexa, Turk, Seidel, Multi-level Partition of Unity Implicits, International Conference on Computer Graphics and Interactive Techniques ACM Press 463-470 (2003)

[OR01] O'Reilly, Srinivasa, On a decomposition of generalized constraint forces, The Royal Society 457 1307-1313 (2001)

[OS01] O'Sullivan, Dingliana, Real-Time Collision Detection and Response Using Sphere-Trees, Image Synthesis Group, Trinity College Dublin (1999)

[OS02] O'Sullivan, Dingliana, Collisions and Perception, ACM Transactions on Graphics Volume 20 , Issue 3 ACM Press 151-168 (2001)

[OU01] Jianjun Oung, Design and implementation of an object-oriented geometric constraint solver, CISE department, University of Florida, (2001)

[PA01] Jill Bazemore Pable, A description of student's experiences with a studio-based quick sketching course using traditional and computer-based instruction practice strategies, College of Education, University of South Florida, August (2000)

[PA02] Paredis, Diaz-Calderon, Sinha, Khosla, Composable Models for Simulation-Based Design, Institute for Complex Enginnered Systems, Carnegie Mellon University (2000)

[PA04] Pai, Reissel, Haptic Interaction with Multiresolution Image Curves, Computers and Graphics Vol. 21 Elsevier 405-411 (1997)

[PA05] Pasko, Adzhiev, Sourin, Savchenko, Function Representation in Geometric Modeling: Concepts, Implementation and Applications, The Visual computer vol. 11, No 8 Springer 429-446 (1995)

[PA06] Park, Gero, Qualitative Representation and Reasoning about Shapes, Visual and Spatial Reasoning in Design Gero, J. S. and Tversky, B. 55-68 (1999)

[PA07] Pahl, Beitz, Konstruktionslehre, Springer Verlag (1977), ISBN : 978-3-540-34060-7

[PA08] Pahl, Beitz, Engineering design, Springer (1984), ISBN : 978-1846283185

[PE01] Pereira, Jorge, Branco, Ferreira, Towards Calligraphic Interfaces: Sketching 3D Scenes with Gestures and Context Icons, WSCG'2000 (2000)

[PE02] Peternell, Pottmann, Steiner, Zhao, Swept volumes, Computer-Aided Design Appl. 2 599-608 (2005)

[PI01] Picinbono, Delingette, Ayache, Non-Linear Anisotropic Elasticity for Real-Time Surgery Simulation, Epidaure, INRIA Sophia-Antipolis (2000)

[PI02] Pithioux, Lopez, Meier, Monserrat, Juan, Alcaniz, ParSys: a new particle system for the introduction of on-line physical behaviour to three-dimensional synthetic objects, Computers & Graphics 29 Elsevier 135-144 (2005)

[PO01] Popovic, Expertise development in product design - strategic and domain-specific knowledge connections, Design Studies 25 Elsevier 527-545 (2004)

[PU01] Purcell, Gero, Drawings and the design process, Design Studies 19 Elsevier 389-430 (1998)

[PY01] Pylyshyn, Mental Imagery: In search of a theory, Behavioral and Brain Sciences (2001)

[RA01] Fesharakifard, Fuchs, Dautin, A string-based haptic interface for CAVE-liked immersive virtual space, Proceedings of Virtual Concept Vol. 1 Springer (2005)

[RA02] Raymaekers, Vansichen, Van Reeth, Improving Sketching by Utilizing Haptic Feedback, Papers from the 2002 American Association for Artificial

Intelligence Spring Symposium (AAAi 2002) 113-117 (2002)

[RA03] Raviv, Elber, Three-dimensional freeform sculpting via zero sets of scalar trivariate functions, Computer-Aided Design 32 Elsevier 513-526 (2000)

[RA04] Chris Raymaekers, Haptic Feedback in Virtual Environments: Towards a Multi-modal Interface, Scool voor Informatietechnologie, transnationale Universiteir Limburg, (2002)

[RE01] Redon, Kheddar, Coquillard, CONTACT: Arbitrary in-between motions for collision detection, 10th IEEE International Workshop on Robot and Human Interactive Communication, 2001 106-111 (2001)

[RE02] Redon, Kheddar, Coquillart, Gauss'Least Constraints Principle and Rigid Body Simulations, IEEE International Conference on Robotics and Automation, 2002 Volume: 1 517- 522 (2002)

[RE03] Redon, Kheddar, Coquillart, Fast Continuous Collision Detection between Rigid Bodies, Eurographics Vol. 21, No. 3 Blackwell Publishing 279-288 (2002)

[RE04] Steven Joseph Rekuc, Eliminating Design Alternative under Interval-based Uncertainty, School of Mechanical Engineering, Georgia Institute of Technology, July (2005)

[RE05] Stéphane Redon, Algorithmes de simulation dynamique interactive d'objets rigides, Laboratoire Systèmes Complexes, Université d'Evry, octobre (2002)

[RE06] Renouf M., Acary V., Dumont G., 3D frictional contact and impact multibody dynamics: a comparaison of algorithms suitable for real-time applications, MULTIBODY DYNAMICS 2005 ECCOMAS Thematic Conference (2005)

[RE07] Reinders James, Threading Building Blocks book, O'Reilly (2007), ISBN : 0-596-51480-8

[RI01] Richard, Vaz-Cerniglia, Portalier, Evolution des procédures d'exploration haptique chez des sujets voyants, aveugles tardifs et aveugles précoces, Revue européenne de psychologie appliquée 54 Elsevier 227-236 (2004)

[RO01] Rodgers, Green, McGrown, Using concept sketches to track design progress, Design Studies 21 Elsevier 451-464 (2000)

[RO02] Roy, Bharadwaj, Design with part behaviors: behavior model,

representation and application, Computer-Aided Design 34 Elsevier 613-636 (2002)

[RO03] Rodenacker, Wolf G., Methodisches Konstruieren, Springer (1976), ISBN : 3-540-07513-5

[RU01] Ruspini, Khatib, Collision/Contact Models for the Dynamic Simulation of Compex Environmnets, IEEE/RSJ International Conference on Intelligent Robots and Systems IROS'97 (1997)

[RU02] Russel S. Open Dynamics Engine v0.5 User Guide 2006

[SA01] Safin, Boulanger, Leclercq, A Virtual Desktop's First Evaluations for An Augmented Design Process, Proceedings of Virtual Concept 2005, Biarritz, France (2005) 1-8

[SC01] Schirski, Kuhlen, Hopp, Adomeit, Pischinger, Bischof, Virtual Tubelets--efficiently visualizing large amounts of particle trajectories, Computer & Graphics 29 Elsevier 17-27 (2005)

[SC02] Schmidt, Wyvill, Sousa, Jorge, ShapeShop: Sketch-Based Solid Modeling with BlobTrees, EUROGRAPHICS Workshop on Sketch-Based Interfaces a (2005)

[SC03] Jean-Marc Schwartz, Calcul rapide de forces et de déformations mécaniques non-linéaires et visco-élastiques pour la simulation de chirurgie, Faculté des sciences et de génie, Université de Laval, Octobre (2003)

[SC04] Acali, Wright, Shillito, 3D Modeling is not for WIMPs, HCI International 2003 Lawrence Erlbaum Associates (2003)

[SC05] Schömer, Reichel, Warken, Efficient collision Detection for Curved Solid Objects, Seventh ACM Symposium on Solid Modeling and Applications (SM 02) ACM Press 321-328 (2002)

[SH01] Shillito, Scali, Whright, Haptics: For a more experimential quality in a computer interface, 5th European Academy of Design Conference, Barcelona (2003)

[SI01] Sitharam, Peters,Zhou, Solving minimal, wellconstrained, 3D geometric constraint systems; combinatorial optimization of olgebraic complexity, (2004)

[SI02] Singh, Jain, Jain, Concurrent optimal adjustement of nominal dimensions and selection of tolerances considering alternative machines, Computer-Aided Design 38 Elsevier 1074-1087 (2006)

[SN01] Snibbe, MacLean, Shaw, Roderick, Verplank, Scheeff, Haptic Techniques for Digital Medial, Proceedings of the 14th ACM Symposium on User Interface Software and Technology UIST 2001 ACM Press (2001)

[SO01] Sobek, Use of Journals to Evaluate Student Design Processes, Proceedings of the 2002 ASEE Conference, Montreal (2003)

[SO02] Sourin, Functionally based virtual embossing, The Visual Computer Springer-Verlag (2001)

[ST01] Stork, Santos, Fleish, Recommandations for Base Hardware and System Software, SmatSketches, Information Society Technologies (2002)

[ST02] Stickland, McKay, Scanlon, The development of a three dimensional imaging system and its application in computer aided design workstations, Mechatronics 13 Pergamon 521-532 (2003)

[ST03] Starling, College, Performance-Based Computational Synthesis of Parametric Mechanical Systems, Department of Engineering, University of Cambridge, Janvier (2004)

[ST04] Stewart, Buttolo, Chen, Marsan, Haptic Surface Manipulation, ICDA Group, Ford Research Laboratory (2001)

[SU01] Suwa, Purcell, Gero, Macroscopic analysis of design processes based on a scheme for coding designer's cognitive actions, Design Studies 19 Elsevier 455-483 (1998)

[SU02] Su, Collins, McCarthy, An extensible Java applet for spatial linkage synthesis, Proceedings of DETC'02 ASME (2002)

[TA01] Tarrin, Coquillart, Hasegawa, Bouguilla, Sato, The Stringed Haptic Workbench: a New Haptic Workbench Solution, Computer Graphics Forum Vol. 22, N. 3 Blackwell Publishing 583-589 (2003)

[TA02] Tatarinov, Les équations de mécanique classique en formes laconiques, Département de Mécanique Théorique, Université de Moscou (2005)

[TH01] Thompson, Johnson, Cohen, Direct Haptic Rendering Of Sculptured Models, Proceedings Symposium on Interactive 3D Graphics, Providence (1997)

[TH02] David Charles Thompson, Feasibility of a Skeletal Modeler for Conceptual Mechanical Design, University of Texas, Austin, December (2000)

[TO01] Tolba, Dorsey, McMillan, A Projective Drawing System, Proceedings of I3D 2001, Cambridge, MA 02139 (2004)

[TS01] Tsagarakis, Petrone, Mayoral, Caldwell, A Haptic Aesthesis API and its Application to the Pre-Operative Planning for total Hip Arthroplasty, Proceedings of Virtual Concept 2005, Biarritz, France (2005) 1-7

[TU01] Turro, Khatib, Haptically Augmented Teleoperation, Proc. IEEE International Conference on Robotics and Automation 386-392 (2001)

[UD01] Udwadia, Kalaba, On the foundations of analytical dynamics, International Journal of Non-Linear Mechanics 37 Pergamon 1079-1090 (2002)

[UL01] Ullman, Wood, Craig, The Importance of Drawing in the Mechanical Design Process, Computer & Graphics 14, N° 2 263-274 (1990)

[UL02] David G. Ullman, The mechanical design process, McGraw-Hill (1992), ISBN : 0-07-065739-4

[VA01] Van der Lugt, How sketching can affect yhe idea generation process in design group meetings, Design Studies 26 Elsevier 101-122 (2005)

[VA02] Varley, Martin, Suzuki, Frontal geometry from sketches of engineering objects: is line labelling necessary?, Computer-Aided Design Elsevier 1-23 (2005)

[VA03] Vance, Larochelle, Dorozhkin, VRSpatial: Designing spatial Mechanisms using Virtual Reality, Proceedings of DETC'02 ASME (2002)

[VA04] Michael James Van Wie, Designing Product Architecture: a Systematic Method, University of Texas, Austin, December (2002)

[VA05] Van Wie, Bryant, Bohm, McAdams, Stone, A model of function-based representations, Artificial Intelligence for Engineering Design, Analysis and Manufacturing Volume 19 , Issue 2 Cambridge University Press 89-111 (2005)

[VE01] Verstijnen, Hennessey, van Leeuwen, Harnel, Goldschmidt, Sketching and creative discovery, Design Studies 19 Elsevier 519-546 (1998)

[VE02] Verplank, Gurevich, Mathews, THE PLANK: Designing a simple haptic controller, Proceeding of the Conference on New Instrument for Musical Expression National University of Singapore 1-4 (2002)

[VE03] Yoann VERNAT, Formalisation et qualification de modèles par contraintes en conception préliminaire, Ecole Nationale Supérieure d'Arts et Métiers, Centre de Bordeaux, novembre (2004)

[VI01] Visser, Stappers, van de Lugtr, Contextmapping: Experiences from practice, CoDesign Volume 1, Number 2 Taylor and Francis Ltd 119-149 (2005)

[VI02] Willemien Visser, Dynamic Aspects of Design Cognition: Elements for a cognitive Model of Design, Eiffel, INRIA Rocquencourt (2004)

[VI03] Leonardo Vita, Sciluppo e implementazione di formulazioni per l'analisi dinamica di sistemi mulitibody, Facoltà di Ingegneria, Università Degli Studi di Roma, (2004)

[VI04] Vivek, McAdams, Stone, De Crescenzo, Visual representations as an aid to concept generation, Proceedings of IDETC/CIE 2006 DETC2006-99572 (2006)

[WA01] Wang, Shen, Xie, Neelamkavil, Pardasani, Collaborative conceptual design - state of the art and future trends, Computer-Aided Design 34 Elsevier 981-996 (2002)

[WA02] Wall, Harwin, A high bandwidth interface for haptic human computer interaction, Mechatronics 11 Pergamon 371-387 (2001)

[WA03] Wall, Paynter, Shillito, Wright, Scali, The Effect of Haptic and Stereo in a 3D Target Acquisition Task, Proceedings of Eurohaptics 2002 23-29 (2002)

[WA04] Wang, Becker, Jones, Glover, Benford, Greenhalgh, Vloeberghs, A virtuel reality surgery simulation of cutting and retraction in neurosurgery with force-feedback, Computer Methods and Programs in Biomedicine 84 Alsevier 11-18 (2006)

[WO01] Woods, Newell, Visual, haptic and cross-modal recognition of objects and scenes, Journal of Physiology Vol. 98, No. 1-3 Elsevier 147-159 (2004)

[XI01] Xiao, Zhang, Contact Constraint Analysis and Determination of Geometrically Valid Contact Formations from Possible Contact Primitives, IEEE Transactions on Robotics and Automation Volume: 13, Issue: 3 456-466 (1997)

[YA01] Yang, Concept generation and sketching: Correlations with design outcome, Proceedings of DETC 2003 1-6 (2003)

[YA02] Yang, Lian, Chen, Haptic function evaluation of multi-material part design, Computer-Aided Design Volume 37, Issue 7 Elsevier 727-736 (2005)

[YA03] Yau, Hsu, Development of a Dental Training System Based on Point-based Models, Computer-Aided Design and Applications 3, No. 6 779-787 (2006)

[YE01] Ye, Campbell, Page, Badni, An investigation into the implementation of virtual reality technologies in support of conceptual design, Design Studies 27

Elsevier 77-97 (2006)

[YO01] Yoon, Narayanan, Predictors of success in diagrammatic problem solving, Diagrams. International conference No 3 Springer-Verlag 301-315 (2004)

[YO02] Yoshizawa, Belyaev, Seidel, Free-form Skeleton-driven Mesh Deformations, ACM Solid Modeling 2003 247-253 (2003)

[YO03] Yoshida, Miyazaki, Hoshino, Ozeki, Hasegawa, Yasuda, Yokoi, Spatial Sketch System for Car Styling Design, XIXth Congress of International Society for Photogrammetry and Remote Sensing Vol. XXXIII, Part B5 919-925 (2000)

[ZA01] Gabriel Zachmann, Virtual Reality in Assemblty Operation -- Collision Detection, Simulation Algorithms, and Interaction Techniques, Fachbereich Intormatik, Technischen Universität Darmstadt, Mai (2000)

[ZH01] Zhang, Tor, Britton, Deng, Functional design of mechanical products based on behavior-driven function-environment-structure modeling framework, Innovation in Manufacturing Systems and Technology (IMST) (2002)

[ZH02] Zhao, Zhou, Feng, A theory of degrees of freedom for mechanisms, Mechanism and Machine Theory 39 Elsevier 621-643 (2004)

Nous essayons de montrer comment l'esquisse virtuelle peut être une aide pendant les phases avancées de la conception de systèmes mécaniques. Le titre de ce travail mérite des précisions :
Le terme esquisse doit être rapproché du terme disegno qui ai né pendant la renaissance en Italie.
La conception de systèmes mécaniques implique la prise en compte du mouvement des pièces. Ce point est celui qui différencie le plus notre travail des autres recherches sur l'esquisse virtuelle.
Les phases avancées de conception sont les phases les plus critiques puisque la quasi totalité du coût du produit y est définie.
Aider un concepteur dans son travail, c'est essayer de savoir comment il travaille.
Nous avons créé un premier prototype qui permet d'effectuer un croquis à main levée en 3d avec retour d'effort. La dynamique utilisée contenant des limitations ; nous avons développé notre dynamique et notre gestion de collision que nous décrirons, ainsi que leur application dans le deuxième prototype.

We try to show how the virtual sketching can be a help during the advanced mechanical systems design phases. The title of this work deserves some precise details:
The sketching term is to be brought closer the disegno term which was born during the Renaissance in Italy.
The mechanical systems design implies the taking into account of parts movement. This point is that which differentiates more our work of other research on the virtual sketching.
The advanced phases of design are the most critical phases since in these phases the quasi totality of the product cost is defined.
To help a designer in his work, it is to try to know how work a designer.
In this work a first prototype has been developed which makes it possible to carry out a freehand sketch in 3d with force feedback. The dynamics used feel certain limitations; we chose to develop our dynamics and our management of collision. We will describe the dynamics used and its application in the second prototype.