

N° d'ordre: D05 - 14

# THÈSE

Présentée devant

**devant l'INSA de Rennes**

pour obtenir

le grade de : DOCTEUR DE L'INSA DE RENNES  
Mention INFORMATIQUE

par

Richard KULPA

Équipe d'accueil : SIAMES - IRISA

École Doctorale : Matisse

Composante universitaire : INSA DE RENNES

Titre de la thèse :

*Adaptation interactive et performante  
des mouvements d'humanoïdes synthétiques :  
aspects cinématique, cinétique et dynamique.*

soutenue le 22 novembre 2005 devant la commission d'examen

M. :	Pascale	SÉBILLOT	Présidente
MM. :	Ronan	BOULIC	Rapporteurs
	Jean-Paul	LAUMOND	
MM. :	Sylvie	GIBET	Examineurs
	Franck	MULTON	
	Bruno	ARNALDI	



*Aucun humanoïde synthétique n'a été blessé  
au cours des adaptations présentées  
dans ce mémoire de thèse.*



## Remerciements

Je remercie tout d'abord Pascale Sébillot qui m'a fait l'honneur de présider mon jury de thèse et j'espère que nous pourrons nous revoir régulièrement (certainement à l'INSA). Je remercie ensuite Jean-Paul Laumond et Ronan Boulic d'avoir acceptés sans réserve d'être mes deux rapporteurs de thèse et pour leurs commentaires sur ce travail mais aussi et surtout pour leur accessibilité au-delà de la thèse. Je remercie aussi Sylvie Gibet d'avoir fait partie de ce jury et je ne doute pas que nous nous reverrons régulièrement au vu des collaborations qui unissent nos laboratoires.

Je remercie Franck Multon pour toutes ces discussions passionnantes qui m'ont permis de progresser et de trouver mes marques dans la recherche. Merci surtout de m'avoir soutenu dans les moments difficiles comme la fin de la rédaction de cette thèse. Je suis content de te compter parmi mes amis et j'espère que nous pourrons continuer longtemps à travailler ensemble. Et puis, je te décerne officiellement la médaille du meilleur pourvoyeur de CDD! ;-)

Merci à Bruno Arnaldi que je connais maintenant depuis de si nombreuses années. Je ne te remercie pas uniquement d'avoir tout de suite accepté d'être mon directeur de thèse mais surtout pour la manière humaine avec laquelle tu m'as toujours considéré. J'espère que nous pourrons continuer à avoir ces longues discussions, certes fortes intéressantes, mais surtout souvent très méta-fumeuses ;-). J'en profite pour remercier Stéphane Donikian que je connais depuis presque aussi longtemps et qui m'a fait découvrir bien malgré lui le monde de la recherche. S'il avait pu imaginer que je n'en partirai plus, m'aurait-il proposé ce stage de fin d'études? Bonne question ;-). Un grand merci pour ta disponibilité malgré ton emploi du temps de ministre, notamment pour les relectures d'articles...

Je tiens à remercier Paul Delamarche qui aurait dû faire partie de ce jury à part entière sans les problèmes de quota. Merci sincèrement de m'avoir toujours considéré comme un membre permanent du laboratoire (tout comme Arlette Delamarche) et de m'avoir fait confiance et laisser explorer les domaines de recherche qui me tenaient à cœur. Cette thèse située à la frontière des deux domaines informatique et biomécanique en est le résultat.

Je ne vais pas m'aventurer à citer tout le monde parce que la liste serait très longue mais je tiens vraiment à remercier l'ensemble des membres actuels et passés de l'équipe SIAMES et du LPBEM (y compris la composante obscure qu'est la physiologie ;-). Notamment pour l'ambiance chaleureuse de travail au quotidien : Ben et Muz (pour leur humour de nageurs), Stef (pour les milliers de sandwiches partagés le midi devant notre écran), Armel (pour son goût des questions ahurissantes), Fab (le fou d'algo), Carole (qui connaît les emplois du temps mieux que personne), Guillaume et Gwenola (qui ne connaît pas le bureau du bonheur?), Kadi (pour ses nombreux conseils), Alain (qui règle tellement de problèmes!)... Merci également aux partenaires industriels avec qui j'ai eu la chance de travailler (ils se reconnaîtront :-)

Je remercie également toute ma "grande" famille pour son soutien tout au long de cette thèse. Merci à tous! (même ceux qui vivent dans leur campagne reculée ;-)

Enfin, j'ai bien sûr gardé le meilleur pour la fin ou devrais-je dire la meilleure (la personne qui mérite d'avoir son nom au même titre que moi sur cette thèse)! Mais les mots ne servent à rien : merci à toi Flo!

# Table des matières

<b>Table des matières</b>	<b>1</b>
<b>Introduction</b>	<b>5</b>
<b>1 Revue de la littérature</b>	<b>9</b>
1.1 Représentation des humanoïdes . . . . .	9
1.1.1 Représentation hiérarchique . . . . .	10
1.1.2 Représentation normalisée . . . . .	13
1.1.3 Bilan . . . . .	15
1.2 Représentation et production de mouvement . . . . .	16
1.2.1 Représentation image par image . . . . .	16
1.2.2 Représentation par postures-clé . . . . .	16
1.2.3 Représentation fréquentielle . . . . .	17
1.2.4 Capture de mouvements . . . . .	18
1.2.4.1 Systèmes de capture de mouvements . . . . .	18
1.2.4.2 Traitement des données issues de capture du mouvement . . . . .	19
1.2.5 Génération de mouvements . . . . .	21
1.2.5.1 Modèles descriptifs . . . . .	21
1.2.5.2 Modèles générateurs . . . . .	24
1.2.5.3 Cartes de chemins probabilistes . . . . .	26
1.2.6 Combinaison de mouvements . . . . .	27
1.2.6.1 Graphes de mouvements . . . . .	27
1.2.6.2 Mélange de mouvements . . . . .	29
1.2.7 Bilan . . . . .	31
1.3 Adaptation de mouvements . . . . .	32
1.3.1 Cinématique inverse . . . . .	32
1.3.1.1 Solution analytique . . . . .	33
1.3.1.2 Linéarisation locale . . . . .	33
1.3.1.3 Optimisation . . . . .	36
1.3.1.4 Approche hybride . . . . .	38
1.3.2 Edition des mouvements . . . . .	40
1.3.2.1 Optimisation des trajectoires . . . . .	40
1.3.2.2 Ajout de cartes de déplacement . . . . .	41
1.3.2.3 Adaptation à un squelette différent . . . . .	43
1.3.2.4 Prise en compte de la dynamique . . . . .	44

1.3.3	Bilan . . . . .	47
<b>2</b>	<b>Contexte et objectifs</b>	<b>51</b>
<b>3</b>	<b>Représentation et spécification des contraintes</b>	<b>59</b>
3.1	Représentation externe des contraintes . . . . .	61
3.1.1	Paramètres génériques . . . . .	61
3.1.1.1	Type de contraintes . . . . .	62
3.1.1.2	Emplacement des contraintes . . . . .	64
3.1.1.3	Portée des contraintes . . . . .	64
3.1.1.4	Importance des contraintes . . . . .	65
3.1.1.5	Activation des contraintes . . . . .	66
3.1.2	Paramètres spécifiques . . . . .	67
3.2	Spécification des contraintes . . . . .	68
3.2.1	Spécification hors-ligne . . . . .	68
3.2.2	Spécification en ligne . . . . .	69
3.3	Représentation interne des contraintes . . . . .	69
3.4	Discussion . . . . .	70
<b>4</b>	<b>Adaptation cinématique</b>	<b>73</b>
4.1	Représentation des données . . . . .	74
4.2	Algorithme d'adaptation . . . . .	76
4.2.1	Algorithme général . . . . .	77
4.2.2	Adaptation de l'intérieur des groupes . . . . .	80
4.2.2.1	Contrôle par priorité des contraintes . . . . .	82
4.2.2.2	Contrôle par mélange des contraintes . . . . .	83
4.2.3	Adaptation de la racine . . . . .	83
4.2.4	Adaptation au sol . . . . .	87
4.2.4.1	Adaptation rapide au sol . . . . .	88
4.2.4.2	Adaptation complète au sol . . . . .	89
4.2.5	Critère d'arrêt et convergence . . . . .	91
4.3	Résultats . . . . .	93
4.4	Discussion . . . . .	96
<b>5</b>	<b>Adaptation cinétique</b>	<b>99</b>
5.1	Représentation des données . . . . .	101
5.1.1	Représentation des données cinétiques . . . . .	101
5.1.2	Découpage en groupes . . . . .	102
5.2	Algorithme d'adaptation . . . . .	103
5.2.1	Algorithme général . . . . .	103
5.2.2	Prise en compte des groupes . . . . .	105
5.2.3	Adaptation des groupes . . . . .	107
5.2.3.1	Détermination du centre de masse local . . . . .	107
5.2.3.2	Adaptation locale d'un groupe . . . . .	108
5.2.4	Adaptation de la racine . . . . .	111
5.2.5	Critère d'arrêt et convergence . . . . .	112
5.3	Fusion des adaptations cinématique et cinétique . . . . .	113



5.3.1	Choix des groupes . . . . .	113
5.3.2	Fusion des adaptations de la racine . . . . .	114
5.3.3	Algorithme final . . . . .	114
5.3.4	Critère d'arrêt et convergence . . . . .	115
5.4	Résultats . . . . .	116
5.5	Discussion . . . . .	119
<b>6</b>	<b>Respect de la dynamique</b>	<b>123</b>
6.1	Analyse . . . . .	124
6.2	Représentation des données . . . . .	126
6.2.1	Représentation des contraintes . . . . .	127
6.2.2	Découpage précis du mouvement de saut . . . . .	127
6.2.3	Représentation des phases . . . . .	128
6.3	Algorithme d'adaptation . . . . .	129
6.3.1	Espace de recherche . . . . .	130
6.3.2	Choix du point dans l'espace . . . . .	132
6.4	Résultats . . . . .	133
6.5	Discussion . . . . .	134
<b>7</b>	<b>Architecture logicielle et applications</b>	<b>137</b>
7.1	Architecture logicielle . . . . .	137
7.2	Applications . . . . .	138
7.2.1	Contrôle direct des contraintes . . . . .	138
7.2.2	Contrôle par modèles paramétriques de mouvement . . . . .	140
7.3	Valorisation . . . . .	142
	<b>Conclusion et perspectives</b>	<b>145</b>
<b>A</b>	<b>Utilisation des contraintes</b>	<b>151</b>
A.1	Contraintes intrinsèquement liées au mouvement . . . . .	151
A.2	Contraintes temps-réel . . . . .	153
<b>B</b>	<b>Exemples de solutions analytiques</b>	<b>155</b>
B.1	Directe . . . . .	155
B.2	Utilisation de rotations . . . . .	156
B.3	Calculs géométriques plus complexes . . . . .	156
<b>C</b>	<b>Adaptation d'un membre pour respecter une position de COM fournie</b>	<b>159</b>
C.1	Problème . . . . .	159
C.2	Calcul de $d^2$ . . . . .	161
C.2.1	Calcul de $A$ . . . . .	161
C.2.2	Calcul de $B$ . . . . .	162
C.2.3	Calcul de $C$ . . . . .	162
C.2.4	Calcul intermédiaire de $d^2$ . . . . .	163
C.2.5	Calcul de $X_c X_p + Y_c Y_p$ . . . . .	163
C.2.6	Calcul final de $d^2$ . . . . .	164
C.3	Calcul final de $l$ . . . . .	164

<b>Glossaire</b>	<b>167</b>
<b>Notations</b>	<b>169</b>
<b>Bibliographie</b>	<b>186</b>
<b>Table des figures</b>	<b>187</b>

# Introduction

L'humain virtuel devient un enjeu socio-économique important puisqu'il est à l'interface de nombreuses disciplines scientifiques. Comprendre et simuler le mouvement humain intéressent effectivement un grand nombre de disciplines parmi lesquelles on peut citer l'animation par ordinateurs, la robotique, la biomécanique, les neurosciences, la paléanthropologie, la médecine... Les développements d'humains virtuels contrôlables sont appuyés par des programmes nationaux et européens qui démontrent l'intérêt de ce champ trans-disciplinaire : pôle de compétitivité "Images et réseaux" de la région Bretagne, pôle de compétitivité "System@tic" d'Île-de-France, les nombreux appels ANR (Agence Nationale de la Recherche) réalisés par le RNTL (Réseau National des Technologies Logicielles) et le RIAM (Recherche et Innovation en Audiovisuel et Multimédia). Ces derniers réseaux ont stipulé explicitement un besoin important d'animation d'humains virtuels dans leurs priorités.

L'humanoïde de synthèse est en effet au centre d'un grand nombre d'applications en plein essor, basées sur la réalité virtuelle et les environnements interactifs en général. Le principe même d'interaction apporte trois principales contraintes au développement d'humanoïdes synthétiques :

- ils doivent être crédibles : ceci implique que leur comportement, leur gestuelle et leur rendu visuel soient d'un niveau de complexité suffisant pour entraîner des réactions réalistes d'utilisateurs interagissant avec eux ;
- les calculs doivent être rapides : dans un cadre interactif, il est nécessaire que l'humanoïde réagisse aux actions d'un utilisateur ou d'autres entités sans impliquer des délais de réponse trop longs (ceci est d'autant plus critique lorsqu'on doit animer une foule d'humanoïdes) ;
- et ils doivent être facilement contrôlables : les personnages autonomes qui doivent exécuter des actions complexes issues d'un scénario interactif et les avatars pilotés par les utilisateurs doivent tous les deux offrir des modes de contrôle intuitifs et permettant une large gamme de mouvements.

C'est dans ce cadre que se situe ce travail de thèse.

Cependant, le mouvement humain est un phénomène très complexe qui est le résultat de nombreux processus mentaux, neuro-musculaires, biomécaniques, physiologiques... qui sont fortement couplés les uns aux autres. Les disciplines expérimentales qui cherchent à percer les secrets du mouvement ont réellement débuté au 19ème siècle avec les travaux de J.E. Marey [Marey 94]. Après plus d'un siècle de recherche et la création de plusieurs disciplines (comme la biomécanique, les neurosciences, la physiologie du mouvement, entre

autres) le problème est loin d'être résolu. Plusieurs grandes théories s'affrontent et aucune n'aboutit pour l'instant à un ensemble de connaissances génériques directement utilisables en animation d'humanoïdes synthétiques.

D'ailleurs, les travaux couplant analyse et synthèse du mouvement humain sont de plus en plus encouragés par les grandes sociétés scientifiques. Par exemple, en robotique, les thématiques des dernières grandes conférences témoignent de l'intérêt du couplage de ces disciplines pour mieux caractériser l'humanoïde et la façon dont il interagit avec des personnes réelles. Un autre exemple concerne la société internationale de biomécanique qui a créé un groupe de recherche sur la simulation du mouvement. Il ne faut donc pas attendre une solution "clé en main" pour animer des humanoïdes synthétiques mais plutôt proposer des méthodes d'animation les plus ouvertes possibles. De plus, il semble pertinent d'avoir une approche pluri-disciplinaire s'inscrivant dans le mouvement scientifique décrit précédemment. L'animation semble avoir toute sa place dans cette dynamique d'analyse/synthèse du mouvement en proposant, entre autres, des environnements de simulation de lois. D'ailleurs, les travaux récemment apparus à la conférence SIGGRAPH montrent que de nombreuses études sont menées en animation pour identifier des lois ou proposer des outils d'extraction automatique de styles dans le mouvement humain.

Bien entendu, à l'heure actuelle, il est presque impossible de définir un modèle d'humanoïde qui tienne compte de tous les aspects de la génération de mouvements : neurosciences, physiologie, biologie, biomécanique... Le développement des systèmes de capture du mouvement a énormément influencé les recherches récentes en animation d'humanoïdes. En effet, un grand nombre de travaux partent de l'hypothèse que les trajectoires capturées contiennent de manière intrinsèque toutes les informations nécessaires. D'ailleurs, Johansson [Johansson 73] montre qu'il est très facile de reconnaître la démarche d'une personne uniquement avec quelques points animés. Ceci montre bien que, sans comprendre précisément les processus mis en jeu, il est possible d'utiliser la capture du mouvement pour rendre les gestes des humanoïdes crédibles.

Cependant, toute intervention sur les trajectoires capturées peut entraîner une altération des informations contenues dans les données originelles. Comme l'animation en environnement interactif implique de pouvoir générer des mouvements répondant à un nombre très important de sollicitations différentes, il est presque impossible de se contenter d'une base de mouvements capturés. En effet, cela impliquerait de capturer tous les mouvements, pour tous les humanoïdes et dans toutes les situations possibles. Ce serait impossible ou extrêmement coûteux. Même si le risque est grand de détériorer les données initiales, les mouvements capturés doivent donc être adaptés en fonction des besoins. Au lieu de tenter de générer des mouvements à partir d'un ensemble incomplet de lois, on fait alors l'hypothèse qu'il est plus facile actuellement de modifier un mouvement. On suppose alors qu'une légère déformation du mouvement ne détériore pas suffisamment les lois intrinsèques.

Le problème consiste alors à gérer un compromis entre les grandes variétés de contraintes auxquelles l'humanoïde doit répondre et le réalisme des animations que l'on peut produire en utilisant une base de données de gestes. Comme nous l'avons précisé auparavant, il faut aussi proposer des méthodes capables de répondre très rapidement aux demandes d'un

utilisateur et, parfois, pour animer un grand nombre d'humanoïdes en même temps. Dans cette thèse, nous proposons une méthode d'adaptation de mouvements capturés pour gérer des contraintes géométriques en temps-réel (taille du personnage, position d'objets dans l'environnement synthétique, consignes d'un utilisateur). Cependant, comme une infinité de mouvements permet de répondre généralement à ces contraintes, la méthode doit être capable de rapidement choisir l'un des plus crédibles (qu'un humain réel aurait certainement effectué). Pour aider à sélectionner un mouvement crédible, nous avons fait l'hypothèse qu'il devait respecter un certain nombre de contraintes cinématiques, cinétiques (sur la position du centre de masse qui intervient fortement dans la notion d'équilibre) et mécaniques.

Le document est organisé de la manière suivante. La section 1 dresse la revue de littérature concernant l'animation d'humanoïdes synthétiques en se focalisant sur les méthodes cherchant à animer le squelette. L'analyse de l'état de l'art nous a conduit à fixer le contexte et les objectifs précis de ce travail (en section 2). Comme l'adaptation de mouvements se base généralement sur des contraintes imposées par l'environnement ou l'utilisateur, nous décrivons dans la section 3 le formalisme utilisé pour les décrire. La section 4 traite alors de la manière de résoudre les contraintes géométriques et la section 5 précise comment tenir compte d'une position imposée du centre de masse. La section 6 montre ensuite comment utiliser cette dernière fonctionnalité pour tenir compte de lois fondamentales de la mécanique dans le cadre de l'adaptation de mouvements de sauts à de nouvelles performances. L'architecture logicielle dans laquelle est intégrée ces travaux ainsi que des exemples d'applications sont fournis dans la section 7. Enfin, une conclusion des travaux présentés dans ce mémoire ainsi que les perspectives de leur utilisation sont présentées dans la section 7.3.



# Chapitre 1

## Revue de la littérature

Afin de parvenir à faire de l'animation d'humanoïdes synthétiques, la première étape consiste à modéliser le personnage à animer. Il convient donc de faire des choix sur les données du corps humain qui sont ou non pertinentes dans le cadre de l'animation de personnages. Cette modélisation est discutée dans la section 1.1.

Une fois la modélisation du personnage choisie, il faut déterminer de quelle manière ces données sont modifiées afin d'obtenir un mouvement réaliste. La section 1.2 se propose de décrire les différentes représentations de mouvement ainsi que les techniques utilisées pour les produire.

Les mouvements ainsi obtenus sont néanmoins dépendants du contexte de leur création. Ainsi, ils sont capturés ou générés pour une morphologie et un environnement particuliers. La section 1.3 propose de décrire les différentes méthodes d'adaptation de ces mouvements afin de pouvoir prendre en compte de nouvelles contraintes.

### 1.1 Représentation des humanoïdes

Le corps humain est composé d'une multitude d'os, de muscles, d'organes, de tissus mous et de peau ainsi que de structures décisionnelles et de transmission des commandes. Il est actuellement impossible de modéliser très finement un tel système. Il faut donc trouver une représentation simplifiée de l'humanoïde afin de diminuer la complexité algorithmique du modèle d'animation sous-jacent tout en assurant un réalisme suffisant des mouvements effectués.

Dans le cadre de l'animation d'humanoïde, Chadwick et coll. [Chadwick 89] ont proposé de décomposer le corps humain en trois couches distinctes : le squelette, les muscles et la peau. Le *squelette* permet de définir la structure osseuse du personnage ainsi que les articulations qui assemblent ces os entre eux. Les *muscles* s'appuient alors sur cette structure squelettique afin de bouger les os. En effet, lorsqu'un muscle se contracte, sa longueur diminue de telle sorte que les os attachés se rapprochent. Ces muscles interviennent également pour améliorer le réalisme de l'enveloppe corporelle de l'humanoïde synthétique en changeant de forme en fonction de leur contraction. Enfin, la *peau* est la dernière couche qui donne la touche finale au rendu du personnage.

Cette décomposition est alors utilisée dans un processus réalisé en deux étapes. La première est l'animation du personnage qui consiste à créer des mouvements (effets des muscles sur le squelette de l'humanoïde). La deuxième s'appuie sur cette animation pour ajouter les détails visuels qui améliorent le réalisme de l'enveloppe corporelle (muscles et peau). Dans le cadre de cette thèse, nous ne traitons que de la partie animation du personnage. L'amélioration visuelle de l'humanoïde n'est donc pas considérée mais peut néanmoins être ajoutée aux mouvements issus de nos travaux.

Cette section décrit la représentation du squelette humain. Il est trop complexe à modéliser dans son intégralité puisqu'il est composé de plus de 200 os [Gray 18]. Pour qu'il soit utilisable dans le cadre de l'animation d'humanoïdes synthétiques, il est donc nécessaire de trouver des représentations simplifiées qui préservent néanmoins les caractéristiques essentielles qui dépendent du domaine d'application. Pour l'animation, c'est évidemment la restitution correcte du mouvement qui est indispensable. Plusieurs représentations ont été proposées dans la littérature et les sous-sections suivantes se proposent de les décrire.

### 1.1.1 Représentation hiérarchique

Une manière simple de représenter un personnage est de définir le squelette uniquement à partir des positions de ces articulations. Johansson [Johansson 73] a montré que cette représentation est suffisante pour percevoir les mouvements et distinguer les différents styles des personnages, par exemple la marche d'un homme ou d'une femme. Cependant, elle peut conduire à une désarticulation du système puisque rien ne garantit que la distance entre ces articulations est constante. C'est la raison pour laquelle une **structure hiérarchique** est utilisée en animation. En effet, chaque os est représenté comme un arc de cette hiérarchie et les nœuds sont par conséquent les articulations qui rattachent ces os entre eux. Le squelette est ainsi représenté par des solides rigides articulés. L'animation d'un humanoïde synthétique peut alors être réalisée en utilisant les trois degrés de liberté de rotation de chaque segment du squelette au cours du temps. Ainsi, la distance entre les articulations et par conséquent la longueur des segments reste constante.

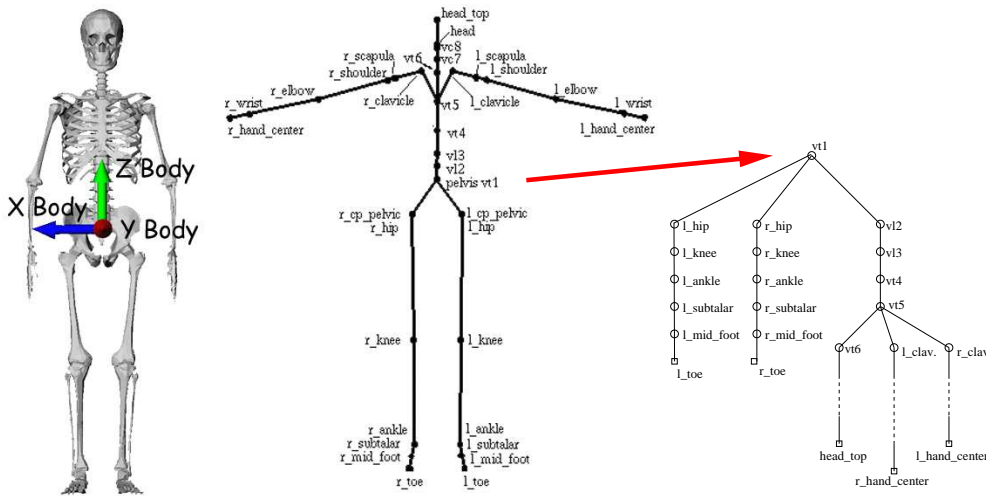


FIG. 1.1 – Représentation hiérarchique du squelette.



Cette représentation hiérarchique du squelette offre donc une formulation intuitive pour l'animation d'humanoïdes synthétiques et fait d'ailleurs l'objet d'une norme : H-Anim [H-Anim 01]. Cependant, quelques simplifications géométriques sont utilisées :

- Les articulations sont associées à une position spatiale qui représente le centre de rotation entre deux objets rigides articulés. C'est la raison pour laquelle seuls les trois degrés de liberté de rotation sont utilisés pour déplacer les segments. Toutefois, sur un squelette humain, la zone de contact entre deux os ne permet pas d'avoir un unique centre de rotation puisque celle-ci est surfacique et non parfaite. Plusieurs travaux se sont donc focalisés sur une représentation plus fidèle d'une partie complexe du squelette telles que la colonne vertébrale [Korein 85, Monheit 91, Furukawa 01], l'épaule [Maurel 00], la main [Rijpkema 91] ou encore le genou [Maciel 02]. Ces études restent toutefois proposées pour des éléments séparés du corps humain.
- Une représentation du squelette articulé sous forme hiérarchique suppose le choix d'une articulation particulière comme racine de l'arborescence. Elle est généralement placée au centre du bassin comme le montre la partie gauche de la figure 1.1. Néanmoins, suivant le contexte de l'animation, ce choix n'est pas toujours judicieux. Ainsi, l'animation d'un humanoïde suspendu à une branche d'arbre par un doigt illustre clairement un squelette hiérarchique dont la racine devrait être associée à l'articulation de l'extrémité du doigt. Cette racine de la hiérarchie est le seul nœud possédant trois degrés de liberté de translation en plus des trois de rotation.
- La complexité du squelette humain est trop importante pour pouvoir contrôler précisément l'ensemble des os du squelette. La manipulation d'un nombre limité d'articulations permet d'alléger la structure tout en conservant un niveau de détail suffisant pour retranscrire les principales caractéristiques d'un mouvement humain. La diminution du nombre d'articulations implique donc le regroupement d'un sous-ensemble d'os du squelette en une partie rigide. On peut citer comme exemple la colonne vertébrale qui à l'origine comporte 24 vertèbres et qui est généralement discrétisée en 5 ou 6 articulations afin de limiter la complexité du système. La figure 1.1 donne un aperçu d'une représentation du squelette humain couramment utilisée. La figure de gauche montre une représentation géométrique de l'ensemble des os composant le squelette humain alors que celle du milieu propose un squelette simplifié avec un nombre d'articulations limité. Enfin, la figure de droite montre la hiérarchie obtenue à partir de la racine placée sur le bassin.

Cette représentation hiérarchique est donc basée sur des articulations parfaites qui sont les centres des rotations qui modifient la posture du personnage. Ces rotations permettent d'avoir deux types de transformations (cf. figure 1.2) : locale qui permet de passer d'un segment parent  $s_i$  au segment suivant (fils)  $s_{i+1}$  et globale qui met en correspondance le repère du monde avec celui d'un segment. Ainsi, la position de points caractéristiques du squelette  $X$  ne dépend que des angles appliqués à chaque degré de liberté  $\theta$  :

$$X = f(\theta) \tag{1.1}$$

La fonction  $f$  est appelée fonction de **cinématique directe**. Cette fonction est la résultante du cumul de toutes les transformations géométriques ayant permis de passer du repère global du monde au repère local de chaque segment. Plusieurs représentations de ces transformations peuvent être utilisées :

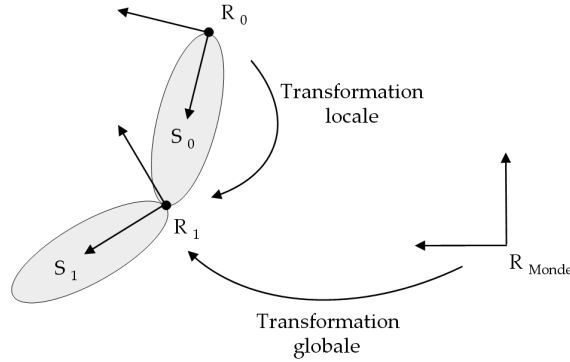


FIG. 1.2 – Transformations locales et globales.

- **Les matrices homogènes** permettent de représenter les repères sous forme matricielle (vecteur en colonne). Il est alors possible de les cumuler en utilisant une multiplication de matrices. Par exemple, la position de l’extrémité du segment  $S$  par rapport à la racine est obtenue grâce à l’équation :

$$S = \left( \prod_{s \in E_s} M_{s_i \rightarrow s_{i+1}} \right) \times \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (1.2)$$

où  $E_s$  est l’ensemble ordonné des segments  $s$  qui part de la racine jusqu’au segment  $S$  dans la hiérarchie et  $M_{s_i \rightarrow s_{i+1}}$  est la transformation locale qui permet de passer du repère du segment  $s$  à celui du segment  $s + 1$ .

Les matrices homogènes permettent de plus de coupler ces rotations avec une translation et une mise à l’échelle dans une matrice  $4 \times 4$ . Cependant, cette représentation pose un problème de singularité pour l’interpolation puisqu’elle peut engendrer des vecteurs colonne nuls ; la matrice n’est alors plus orthonormale.

- **Les angles d’Euler** sont décrits par les trois rotations successives autour de chacun des axes X, Y et Z. Cependant, des variantes existent selon l’ordre des rotations utilisées. Cette représentation très simple et intuitive pose néanmoins un problème de singularité connu sous le nom de “Gimbal Lock”. Cette singularité se traduit par la perte d’un degré de liberté après une séquence de rotations d’angle  $\frac{\pi}{2}$ .
- **Les quaternions** proposés par Lord Williams Hamilton [Hamilton 44, Hamilton 50] au 19<sup>ème</sup> siècle sont une extension des nombres complexes puisque la rotation est symbolisée par  $\mathcal{Q} = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$  où  $w, x, y$  et  $z$  sont des nombres réels et  $\mathbf{i}, \mathbf{j}$  et  $\mathbf{k}$  sont trois axes orthonormés imaginaires tels que  $\mathbf{i}\mathbf{i} = \mathbf{j}\mathbf{j} = \mathbf{k}\mathbf{k} = \mathbf{i}\mathbf{j}\mathbf{k} = -1$ . Les quaternions sont très utilisés en animation [Shoemake 85] parce qu’ils permettent une interpolation qui assure la prise du plus court chemin [Yahia 89] pour aller d’une rotation à une autre.
- Les “**exponential maps**” sont une extension des quaternions où la rotation est représentée à l’aide d’un vecteur de dimension 3 qui décrit l’axe de rotation et dont la norme donne l’angle de rotation (comme cela avait été introduit par Euler au XVIII<sup>ème</sup> siècle). Cette représentation plus simple nécessite toutefois plus de calculs

afin de bénéficier de toutes les propriétés mathématiques des quaternions [Grassia 98] et leur interpolation n'est correcte que dans un voisinage proche où la géodésique est assimilable à un segment.

### 1.1.2 Représentation normalisée

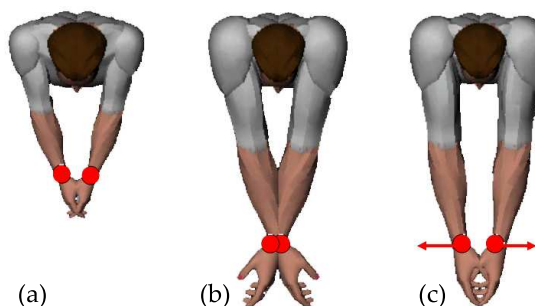


FIG. 1.3 – Problème de morphologies différentes. Les mêmes angles sont appliqués sur les deux personnages des parties (a) et (b). La partie (c) montre la posture qui respecte le contact initial des mains.

Les représentations basées sur l'état des articulations permettent aisément de reproduire un mouvement si on dispose de trajectoires articulaires conformes au squelette. Cependant, rien ne garantit que des angles utilisés sur un squelette donné s'appliquent directement sur un autre, même s'ils partagent la même structure hiérarchique. Les différences de taille des segments suffisent à remettre en cause les angles à appliquer aux articulations pour effectuer une tâche donnée. C'est le cas lorsque deux humanoïdes tentent de frapper dans leurs mains alors qu'ils ont des bras de tailles différentes. En effet, le contact des deux mains ne peut plus être garanti lorsque les mêmes rotations articulaires sont appliquées sur ces deux humanoïdes (cf. parties (a) et (b) de la figure 1.3). Même s'il existe des méthodes pour adapter les trajectoires articulaires d'un personnage pour les rendre compatibles avec un autre humanoïde (cf. figure 1.3c), quelques auteurs ont proposé une représentation du squelette qui évite ce type de problème.

Ménardais [Ménardais 03] a proposé une **représentation normalisée du squelette**. Cette représentation s'appuie sur la structure hiérarchique classiquement utilisée en animation. Cependant, elle distingue trois types de chaînes cinématiques au sein de cette structure du squelette (cf. figure 1.4) :

- les membres sont modélisés comme un seul segment de longueur variable (en rouge) ;
- la colonne vertébrale est modélisée par des splines (en bleu) ;
- et les autres segments, appelés **les segments normalisés**, sont composés d'un seul segment (en noir).

La force de cette représentation réside dans le fait qu'elle intègre des données angulaires et cartésiennes. En effet, les chaînes cinématiques présentées ci-dessus sont calculées directement à partir des positions cartésiennes des articulations. Ainsi une chaîne cinématique  $CC$  composée des segments  $S_j$  est calculée en prenant l'articulation placée à son extrémité

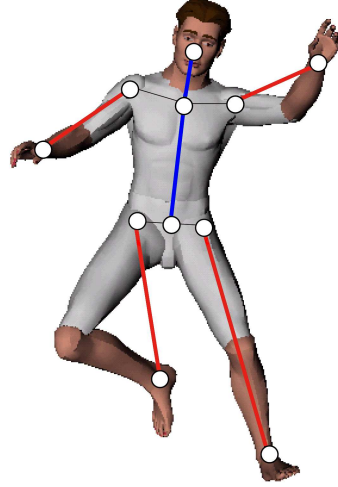


FIG. 1.4 – Représentation normalisée du squelette.

$E$ , appelée articulation distale, relativement à sa racine  $R$  appelée articulation proximale. Ce vecteur  $\overrightarrow{(R, E)}$  est alors normalisé en le divisant par la somme des longueurs initiales  $lg_j$  des segments  $S_j$  appartenant à la chaîne  $CC$  :

$$CCNorm = \frac{E - R}{\sum_j lg_j} \quad (1.3)$$

L'intérêt de cette normalisation est qu'elle permet de retrouver instantanément la position de l'extrémité en inversant le processus de normalisation avec la nouvelle morphologie de personnage. En effet, en multipliant par les longueurs  $lg'_j$  des segments de la nouvelle morphologie, on obtient la nouvelle position  $E'$  de l'extrémité par rapport à la racine :

$$E' = CCNorm \times \sum_j lg'_j \quad (1.4)$$

Cette représentation n'engendre donc aucune perte d'informations lorsque les mêmes données anthropométriques sont utilisées, c'est-à-dire lorsque  $lg$  est égal à  $lg'$ .

Comme le montre la figure 1.4, les articulations des genoux, coudes et vertèbres ne sont pas intégrées dans la représentation normalisée. Cependant, leurs positions peuvent être retrouvées instantanément lorsque c'est nécessaire par exemple pour visualiser la posture du personnage dans un moteur de rendu. Des solutions analytiques sont alors utilisées. Pour les membres, ces dernières s'appuient sur la représentation de la chaîne cinématique :

- un repère qui définit à la fois le vecteur  $\overrightarrow{(R, E)}$  qui va de sa racine  $R$  à son extrémité  $E$  et le demi-plan dans lequel se trouve l'articulation manquante, le genou ou le coude (cf. figure 1.5),
- et une longueur variable qui est la norme du vecteur  $\overrightarrow{(R, E)}$ . La jambe est plus ou moins pliée en fonction de cette longueur.

Pour leur part, les vertèbres sont directement retrouvées en discrétisant les splines qui représentent la colonne vertébrale.

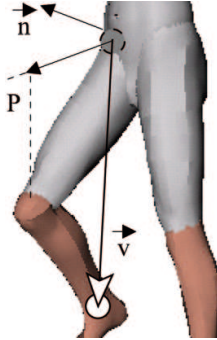


FIG. 1.5 – Représentation normalisée de la jambe. Le demi-plan dans lequel se trouve le genou est indirectement défini par le repère de cette jambe (axe  $P$ ). Les coordonnées locales du genou sont alors retrouvées analytiquement.

Toutes les chaînes cinématiques sont ainsi définies et la position de la racine du squelette est normalisée par la longueur des jambes. Ainsi, le personnage est automatiquement placé à la bonne hauteur par rapport au sol. Cependant, cet avantage est également un inconvénient puisqu'il ne permet pas de gérer des mouvements de marche sur les mains par exemple. La normalisation de la racine devrait donc être adaptée au type de mouvement effectué.

### 1.1.3 Bilan

Dans cette section, nous avons vu que le corps humain est composé d'un grand nombre d'éléments et que les choix effectués en animation consistent généralement à séparer la représentation du squelette de celle des tissus mous. Dans cette thèse, nous ne considérons que la représentation du squelette. Dans le domaine de l'animation, un grand nombre de structures hiérarchiques est utilisé même si elles respectent toutes les normes définies récemment. En effet, ces normes contiennent le nombre maximal d'articulations nécessaires pour l'animation et les segments qui ne sont pas utilisés sont considérés comme fixes, c'est-à-dire qu'ils ne possèdent pas de mouvement relatif par rapport au segment parent dans la chaîne cinématique. Ces normes permettent donc une flexibilité de la représentation des squelettes qui est surtout visible au niveau de la colonne vertébrale et des mains dont le nombre de degrés de liberté à animer est adapté en fonction des besoins.

Toutefois, si la définition des squelettes est assez flexible, il en va autrement de la représentation de la posture. Ainsi, un vecteur d'angles lié à une structure poly-articulée ne peut pas être appliqué sans modification à une nouvelle structure. Ceci implique de mettre en place des modules de conversion qui sont généralement coûteux en temps de calcul et qui altèrent parfois le mouvement. Comme nous le décrivons dans la sous-section 1.1.2, la représentation normalisée du squelette qui est indépendante de la morphologie permet de s'affranchir en grande partie de ces problèmes. Par exemple, dans le cas de la colonne vertébrale, la modélisation par une spline autorise un grand nombre de découpages diffé-

rents. Il suffit alors d'échantillonner la spline conformément au nombre et à l'emplacement des vertèbres utilisées. A une norme flexible de représentation du squelette comme H-Anim [H-Anim 01], il est donc possible d'associer une représentation des postures tout aussi flexible, au moins pour l'humanoïde.

Après avoir vu les différentes modélisations possibles pour le squelette, nous abordons maintenant la manière de représenter et de créer un mouvement.

## 1.2 Représentation et production de mouvement

L'animation d'humanoïdes synthétiques nécessite une représentation précise des mouvements sur lesquels elle s'appuie. Cette section propose donc de décrire, dans un premier temps, ces différentes représentations du mouvement.

Le mouvement naturel est un phénomène encore mal compris qui fait intervenir un grand nombre de facteurs : respect de la physique, capacité des muscles, système de contrôle moteur, comportement, perception... De plus, ces facteurs sont couplés entre eux et intéressent des disciplines scientifiques différentes. En conséquence, il est très difficile de modéliser et de calculer un mouvement en prenant en compte de tous ces facteurs. C'est la raison pour laquelle une première approche consiste à capturer le mouvement d'acteurs dans diverses situations (cf. sous-section 1.2.4). En effectuant ces captures de mouvements, on suppose que tous les facteurs sont intrinsèquement pris en compte, sans explicitement les définir. La seconde approche consiste à générer le mouvement à partir de connaissances issues de la biomécanique et de la neurophysiologie. Cette génération de mouvements est décrite dans la sous-section 1.2.5. Une dernière méthode consiste à représenter le mouvement comme une combinaison d'autres gestes (cf. sous-section 1.2.6).

### 1.2.1 Représentation image par image

Dès la fin du 19<sup>ème</sup> siècle, le mouvement a été décrit comme une succession de postures dans le temps [Marey 94]. Cette décomposition en séquences d'images a permis de décrire très formellement le mouvement humain. Cette approche s'applique effectivement bien à l'analyse du mouvement pour laquelle on dispose de mesures à échantillonnage constant. En animation, le problème est différent, particulièrement dans des environnements calculés en temps-réel. En effet, dans ce contexte, la fréquence d'animation fluctue en fonction de la complexité de la scène à calculer et de la capacité de la machine. La description du mouvement comme une succession de postures espacées d'un intervalle de temps fixe n'est donc pas suffisante.

### 1.2.2 Représentation par postures-clé

Afin de simplifier le travail des animateurs qui décrivaient le mouvement comme une suite ininterrompue d'images, cette technique propose de spécifier uniquement quelques postures du mouvement qui sont jugées importantes. La sélection de ces postures peut être faite manuellement ou automatiquement. Par exemple, Assa et coll. [Assa 05] propose d'effectuer cette sélection en tenant compte des positions, des angles, des vitesses cartésiennes et angulaires de chaque articulation. Le reste de l'animation est alors déduit de ces

postures par interpolation. Cette technique est très générique puisqu'elle peut être utilisée avec toutes les représentations du squelette. La représentation la plus utilisée reste néanmoins la représentation angulaire.

Cette technique est principalement utilisée en représentant les trajectoires articulaires comme des fonctions définies par morceaux [Sturman 86]. Ainsi une trajectoire  $\theta^i(t)$  donnant l'orientation de la  $i^{\text{me}}$  articulation dans le temps est représentée par une série de  $n$  postures-clés  $(\theta_n^i, t_n)$ . Les données intermédiaires sont alors obtenues par interpolation. Bien que la phase de conception soit simplifiée par rapport aux méthodes cherchant à représenter ces trajectoires image par image, l'animateur doit néanmoins effectuer une série d'essais/erreurs pour parvenir au mouvement désiré. En effet, en plus du type d'interpolation utilisé, la qualité du mouvement résultant est très dépendante du choix des postures-clés [Assa 05]. Ainsi un échantillonnage trop faible ne peut pas retranscrire les hautes fréquences auxquelles l'être humain est très sensible. Le mouvement a alors tendance à paraître trop "lissé". De plus, il faut généralement reprendre le processus complètement pour tenir compte d'autres paramètres comme les émotions ou l'adaptation à l'environnement. Le manque de flexibilité de cette représentation se traduit donc par du temps de main d'œuvre d'animateur.

### 1.2.3 Représentation fréquentielle

Au contraire de la représentation par postures-clé, l'utilisation du traitement du signal permet de faire ressortir les propriétés dynamiques contenues intrinsèquement dans chaque trajectoire articulaire. Le principe est de décomposer le mouvement (principalement les angles d'Euler) sous forme de séries de Fourier ou d'ondelettes. Une décomposition des trajectoires angulaires  $\theta_i(t)$  est obtenue sous la forme de :

$$\theta_i(t) = \alpha_0 + \sum_{k \geq 1} \alpha_k \sin(k.t + \Phi_k)$$

où la trajectoire angulaire  $\theta_i$  se décompose comme une somme de  $k$  sinusoides d'amplitude  $\alpha_k$ , de périodes  $\frac{2\pi}{k}$  et de déphasage  $\Phi_k$ . En utilisant un filtre passe-bas, on peut représenter l'ensemble de la trajectoire par une série finie de paramètres  $(\alpha_j, \Phi_j)$ . Cette représentation offre un encodage généralement réduit.

Unuma et coll. [Unuma 91, Unuma 93, Unuma 95] ont utilisé ces séries de Fourier sur les trajectoires angulaires d'un personnage afin d'extraire un certain nombre de paramètres d'une locomotion : par exemple un facteur définissant une marche normale et un facteur décrivant une marche fatiguée. Ces facteurs sont alors utilisés pour générer de nouveaux mouvements en interpolant ou en extrapolant dans le domaine fréquentiel. Une marche normale peut alors être changée en marche fatiguée ou en course de manière fluide.

En résumé, cette représentation offre des paramètres autorisant un contrôle intuitif des mouvements. Contrairement aux représentations par postures-clé, il n'est donc pas nécessaire de mettre en place des modules complexes pour tenir compte d'émotions dans le mouvement. Malgré cet atout, la représentation du mouvement dans l'espace fréquentiel

est restée assez peu utilisée. En effet, elle s’applique particulièrement bien à des mouvements périodiques mais ne permet pas de contrôler intuitivement le geste.

Quelle que soit la représentation choisie, elle ne sert que de support au mouvement. En effet, ce qui donne l’impression qu’un personnage est animé, ce sont les données appliquées à chaque paramètre du mouvement au cours du temps. Par exemple, dans la représentation angulaire, il faut fournir les angles de chacune des postures. Les animateurs arrivent ainsi à créer des mouvements mais c’est un travail très long qui peut aboutir à des mouvements irréalistes en fonction du talent de l’animateur. Il a donc fallu trouver d’autres manières de produire des mouvements. Les sous-sections suivantes traitent de ces différents solutions.

#### 1.2.4 Capture de mouvements

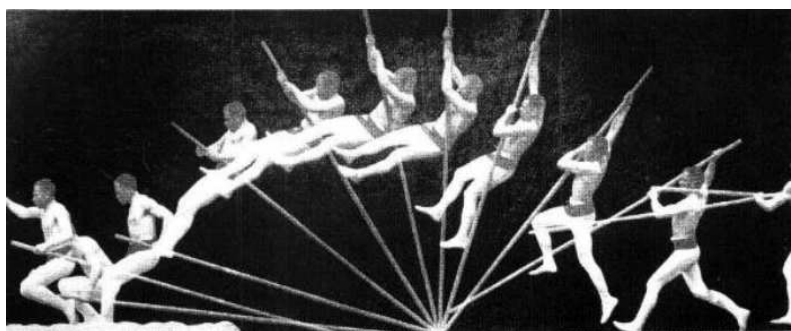


FIG. 1.6 – Etude de Marey sur le saut [Marey 94].

La manière la plus intuitive d’obtenir des mouvements réalistes est de les mesurer directement sur les êtres humains eux-mêmes ; on parle alors de capture de mouvements. Les premiers systèmes étaient utilisés pour comprendre le mouvement humain et isoler un certain nombre de facteurs responsables de la performance motrice. Les études de Marey [Marey 94] en sont un bon exemple (cf. figure 1.6). Récemment, des systèmes de capture de mouvements ont été mis en place afin de capturer le mouvement d’un personnage en trois dimensions. Cependant, comme il est impossible de mesurer directement le déplacement du squelette sans utiliser les rayons X (qui sont invasifs et dangereux pour le sujet), toutes les techniques développées jusqu’alors mesurent le déplacement de points placés sur le sujet. Or, nous avons vu que l’animation d’humanoïdes synthétiques s’appuie avant-tout sur le squelette humain. En plus des systèmes de capture de mouvements décrits dans la section 1.2.4.1, il est donc nécessaire d’utiliser des traitements pour corriger les données capturées afin de les appliquer sur le squelette à animer (cf. section 1.2.4.2).

##### 1.2.4.1 Systèmes de capture de mouvements

Menache et coll. [Menache 99] ont proposé de classer les différents systèmes de capture de mouvements de la manière suivante :

- Les systèmes *extérieur-entrant* utilisent des récepteurs externes (non rattachés au corps humain) pour collecter les informations. Les données acquises sont émises par



des sources placées sur le corps de la personne en mouvement. On retrouve typiquement dans cette catégorie les systèmes d'acquisition à base de caméras (les récepteurs) et de marqueurs passifs qui constituent des sources (cf. figure 1.8a). Les systèmes les plus répandus sont les systèmes Vicon de la société Oxford Metrics et Eagle Digital de la société Motion Analysis. On retrouve aussi dans cette catégorie les systèmes issus du domaine de la vision qui consistent à ne placer aucun marqueur et à travailler, par exemple, sur le calage d'un ensemble de solides rigides dans les images [Gravila 99] ou par la modélisation du personnage par une surface implicite [Niskanen 05] (cf. figure 1.7).

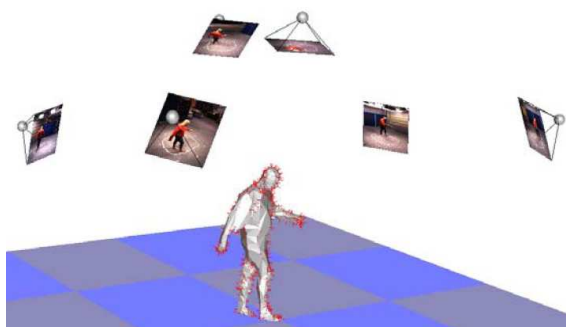


FIG. 1.7 – Capture basée vidéo [Niskanen 05].

- Les systèmes *intérieur-sortant* utilisent des récepteurs placés sur le corps qui collectent des informations émises par des sources externes. Dans les systèmes magnétiques, par exemple, les récepteurs calculent les informations à partir de champs magnétiques ambiants. Les systèmes les plus connus sont Flock of Birds et Motion Star de Ascension Technology Corporation. Chaque capteur fournit sa position et son orientation par rapport à une source placée dans la scène.
- Les systèmes *intérieur-entrant* ont les sources et les récepteurs rattachés au corps du sujet. De telles configurations se retrouvent par exemple dans les systèmes d'acquisition mécaniques. Les récepteurs sont ici des potentiomètres ou goniomètres et l'émetteur est le squelette humain. Par exemple, le système ShapeWrap II, produit par Measureland Shape Advantage, mesure les déformations d'un solide placé sur le corps du sujet (cf. figure 1.8b). Les données ainsi fournies sont les variations d'angles de ces solides au cours du temps.

Tous ces systèmes permettent d'acquérir le mouvement humain avec une plus ou moins grande précision mais ne fournissent pas directement l'évolution des degrés de liberté du squelette. De plus, il est nécessaire d'éliminer un certain nombre d'artefacts introduits par la nature même de ces systèmes. Par exemple, les systèmes *extérieur-entrant* peuvent provoquer des pertes de données dues aux occultations de marqueurs.

#### 1.2.4.2 Traitement des données issues de capture du mouvement

Puisque les systèmes de capture de mouvements ne sont pas invasifs, il est nécessaire de passer de ces marqueurs externes vers les véritables **centres articulaires** qui sont utilisés lors de l'animation du squelette. Plusieurs auteurs [Silaghi 98, Boulic 00] ont proposé de retrouver la position du centre articulaire d'un marqueur dans le repère local

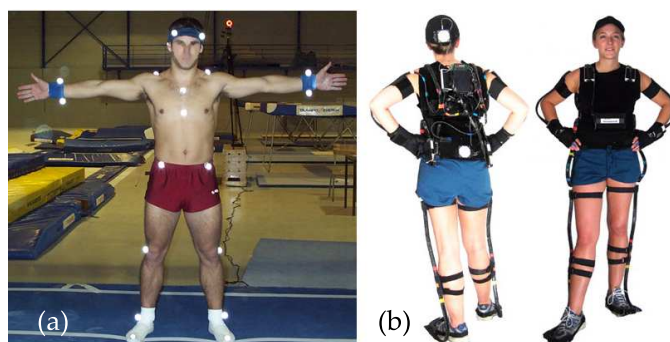


FIG. 1.8 – Systèmes de capture de mouvements. (a) Système opto-électronique Vicon. (b) Système magnétique ShapeWrap II.

du segment auquel il est rattaché. Ces calculs s'appuient sur une hypothèse de distance fixe entre le marqueur et le centre articulaire du segment. Pour les systèmes magnétiques qui renvoient une position et une orientation des marqueurs au cours du temps, Molet et coll. [Molet 96, Molet 97] proposent d'utiliser une phase de calibration afin de retrouver les transformations permettant de passer des repères des capteurs à ceux des repères locaux des segments. O'Brien et coll. [O'Brien 00] minimisent une fonction qui met en relation les vecteurs dans le repère local du marqueur et ceux qui définissent réellement le repère local du segment.

Chaque système de capture de mouvements engendre ses propres artefacts. Il est donc nécessaire de mettre en place des algorithmes de **correction de ces erreurs**. Plusieurs travaux [Boulic 98, Herda 00, Herda 01] ont porté sur la correction des occultations des systèmes opto-électroniques. Une posture du squelette est dans un premier temps estimée à partir des positions de marqueurs fournies par le système. Une visualisation de la posture avec l'enveloppe corporelle permet de prédire les marqueurs occultés et d'apporter ainsi des informations supplémentaires pour aider le système à identifier et suivre les marqueurs.

Certains auteurs ont même proposé des chaînes complètes de traitement des mouvements acquis. Bodenheimer et coll. [Bodenheimer 97] utilise une analyse statistique afin de localiser les erreurs de positionnement et d'orientation les plus flagrantes renvoyées par les capteurs. Ensuite, une posture particulière est acquise afin d'estimer les différentes longueurs de segments ainsi que le positionnement relatif des centres articulaires par rapport aux capteurs. L'animation est au final adaptée sur un squelette de morphologie différente à l'aide d'une méthode de cinématique inverse numérique (cf. sous-section 1.3.1). Enfin, Ménardais et coll. [Ménardais 01, Ménardais 03] ont proposé une chaîne de traitement des mouvements qui permet de corriger les erreurs en utilisant cinq phases successives automatiques puis de les encoder de manière à être directement utilisés dans un moteur d'animation.

Pour conclure sur l'utilisation de données issues de capture de mouvements, on peut dire qu'un grand nombre de modifications est apporté au geste originellement mesuré. Ainsi, la première modification concerne l'imprécision des systèmes de capture du mouvement. Par

exemple, [Ehara 95, Ehara 97] évalue l'imprécision d'un système optique de type Vicon 370 à quelques millimètres sur la mesure de position du centre du marqueur. Cette imprécision évolue en fonction des conditions expérimentales et peut prendre des valeurs très importantes. Nous avons vu qu'il était ensuite nécessaire de calculer les degrés de liberté à appliquer au squelette synthétique pour qu'il reproduise au mieux le mouvement mesuré. Cependant, la topologie du squelette synthétique diffère généralement de celle de l'acteur. La colonne vertébrale, en particulier, est représentée par un nombre restreint de segments rigides alors qu'elle comporte 24 vertèbres pour un être humain réel. Ainsi, l'adaptation des données du squelette de l'acteur vers le squelette synthétique apporte fatalement un biais. Toutefois, ces systèmes permettent d'acquérir rapidement des mouvements qui sont très réalistes, les profils de vitesse et d'accélération correspondant bien à ceux d'un être humain.

Enfin, il est impossible de capturer toutes les variations possibles d'un mouvement en fonction de l'environnement. Par exemple, mesurer les locomotions adaptées à toutes les morphologies et pour tous les sols possibles n'est pas envisageable. Même pour un sous-ensemble restreint de situations, capturer un grand nombre de locomotions différentes serait extrêmement coûteux. Il serait alors plus judicieux de proposer des solutions capables de calculer ces différentes locomotions sans requérir à chaque fois à la capture des mouvements.

### 1.2.5 Génération de mouvements

Depuis les premiers travaux de J. Marey [Marey 94], un grand nombre d'études ont permis d'analyser et de mieux comprendre le mouvement humain. Ces connaissances peuvent être utilisées pour définir des formes caractéristiques de trajectoires et la manière dont elles changent en fonction de différents paramètres. Dès les années 80, ces données ont été utilisées sous le nom de *modèles descriptifs* (cf. sous-section 1.2.5.1). Cependant, comme le mouvement humain répond, entre autres, à des lois de la mécanique, se limiter à une description des trajectoires cinématiques n'est pas suffisant. Les *modèles générateurs* se basent donc sur une description mécanique du corps humain et cherchent à calculer les forces et couples à appliquer pour le faire bouger comme un homme (cf. sous-section 1.2.5.2). Enfin, des techniques de planification de mouvements issues du domaine de la robotique commencent à être utilisées pour animer des personnages humanoïdes. Cette dernière approche pour la génération de mouvements est présentée dans la sous-section 1.2.5.3.

#### 1.2.5.1 Modèles descriptifs

Une première méthode pour générer des mouvements est d'utiliser les connaissances acquises par des analyses menées en biomécanique sur la forme des trajectoires articulaires (cf. sous-section 1.2.5.1.1). Bien évidemment, de telles connaissances n'existent que pour les mouvements les plus étudiés, comme la locomotion. Pour les autres mouvements, il est donc nécessaire de rechercher spécifiquement de nouvelles lois (cf. sous-section 1.2.5.1.2).

##### 1.2.5.1.1 Application de connaissances

Les connaissances sur le mouvement humain sont maintenant très vastes et intéressent différentes disciplines. En biomécanique et en neurosciences, de nombreux travaux cherchent à caractériser la forme des trajectoires issues d'un mouvement et l'influence de divers

paramètres. Par exemple, des études réalisées par Woodworth [Woodworth 99] puis par Fitts [Fitts 54] ont permis d'analyser des gestes de déplacement rapide de la main entre deux cibles (cf. figure 1.9). Une loi existe alors entre la durée du mouvement  $T$ , la distance  $A$  entre les cibles et la précision  $W$  du geste :

$$T = a + b.I_d = a + b.\log_2 \left( \frac{2A}{W} \right)$$

où  $a$  et  $b$  sont des constantes déterminées de façon empirique et  $I_d$  caractérise l'index de difficulté de la tâche.

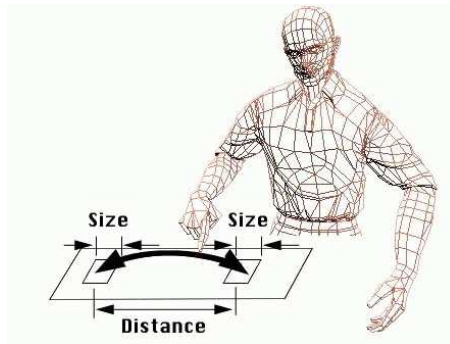


FIG. 1.9 – Exemple de mouvement caractérisé par la loi de Fitts [Fitts 54].

Cette loi a été étendue pour définir la *loi de puissance* qui donne la relation entre la vitesse tangentielle  $V$  et le rayon de courbure  $R$  de mouvements de bras [Viviani 82, Lacquaniti 83, Viviani 95] :

$$V(t) = kR(t)^\beta \quad (1.5)$$

où  $k$  une constante appelée gain de vitesse. Il est montré que la valeur de l'exposant  $\beta$  est constante et fixée à une valeur de  $1/3$  même dans des conditions variées d'exécution, par exemple différentes amplitudes du mouvement [Viviani 95]. D'autres travaux cherchent à étendre cette loi à d'autres mouvements comme la locomotion. Le rayon de courbure  $R$  est alors défini par la trajectoire du déplacement du personnage dans le plan horizontal [Vieilledent 01a, Vieilledent 01b].

Le mouvement le plus étudié est la locomotion pour laquelle on dispose d'un grand nombre de connaissances. En effet, les trajectoires articulaires des membres inférieurs sont bien connues [Alexander 84, Patla 91] et plusieurs études ont montré, entre autres, l'influence de la vitesse [Nilsson 85] ou de l'entraînement [Beaupied 03] sur ces locomotions. C'est aussi l'un des premiers mouvements utilisé pour l'animation puisqu'il permet à l'humanoïde de se déplacer dans l'environnement de synthèse. De nombreux modèles descriptifs de locomotion sont donc apparus dès le début des années 1980 [Zeltzer 82, Zeltzer 83] jusqu'à nos jours [Sun 01]. Cette approche procédurale consiste à décomposer le mouvement en sous-phases, chacune associée à un état d'un automate à états finis. Chaque état contient

une configuration angulaire initiale et finale (une extension ou une flexion maximale). Les configurations intermédiaires sont ensuite interpolées. Comme dans toutes les méthodes à base de cinématique directe, le contrôle des extrémités des chaînes articulées est un problème délicat, comme par exemple le contact entre le pied et le sol dans le cadre de la locomotion. Bruderlin et coll. [Bruderlin 89, Bruderlin 93, Bruderlin 96] ont proposé de changer dynamiquement la racine de la hiérarchie en la plaçant toujours sur un pied au sol, afin de respecter plus facilement sa position.

Cependant, les angles issus de la littérature en biomécanique sont généralement des moyennes qui ne correspondent pas spécifiquement au squelette à animer. Les animations calculées de la sorte peuvent donc comporter un certain nombre d'artefacts. Plusieurs améliorations ont donc été apportées pour assurer un contact sans glissement des pieds sur le sol pendant les phases d'appui par exemple [Girard 85, Girard 87, Boulic 92]. Des techniques d'adaptation de mouvements sont alors utilisées comme la cinématique inverse décrite dans la section 1.3. L'intérêt de ces techniques est que l'on dispose alors d'un modèle paramétrable de locomotion pouvant s'adapter à différents personnages, différentes formes de sol [Cozot 99] et pouvant guider la locomotion par des empreintes sur le sol [van de Panne 97].

Cependant, le style de locomotion est imposé par l'ensemble des connaissances introduites dans le système. Il reste alors difficile de modifier le style pour rendre la démarche d'un personnage plus féminine ou plus stressée, par exemple. Comme nous l'avons vu précédemment, une représentation fréquentielle du mouvement permet de gérer plus facilement certaines de ces caractéristiques [Unuma 95, Bruderlin 95]. Cependant, cette représentation ne permet pas de contrôler le déplacement du personnage aussi aisément qu'avec une connaissance des formes de trajectoires articulaires.

En dehors de la locomotion, il est nécessaire de mener à bien des analyses biomécaniques spécifiques pour extraire des caractéristiques générales de forme de trajectoires ainsi que des opérateurs pour adapter ces formes à divers paramètres de haut niveau. Cette approche a été utilisée par exemple pour le tir au handball [Fradet 03]. On peut aussi définir un langage permettant de composer des mouvements élémentaires qui sont tous finement décrits pour générer des gestes de communication [Gibet 01].

Cependant, comme ces techniques reposent généralement sur des analyses publiées en biomécanique, il est parfois difficile d'utiliser directement les connaissances acquises dans ce domaine pour animer un personnage. A l'inverse, en animation, de plus en plus d'auteurs s'intéressent à mettre en place des techniques quasi-automatiques d'extraction de caractéristiques dans une base de données de mouvements humains.

#### 1.2.5.1.2 Identification de nouvelles lois

L'identification de lois est très complexe parce que le nombre de degrés de liberté et donc de paramètres du mouvement est très important pour l'animation du corps humain. De plus, ces degrés de liberté sont souvent couplés. Dans le cadre de la marche par exemple, une grande majorité des études porte sur la trajectoire relative de la cheville par rapport à la hanche afin de tenir compte simultanément des couplages entre les segments. Cette

trajectoire est appelée *poulaine*.

Une technique d'analyse statistique appelée ACP (Analyse en Composantes Principales) se propose de réduire un système complexe de corrélations en un nombre de paramètres plus restreint. Cette technique conçue par Pearson [Pearson 01] et intégrée aux statistiques mathématiques par Hotelling [Hotelling 33] crée en effet une base de vecteurs orthonormés et ordonnés : les *composantes principales*. La première composante principale contient les informations relatives à la plus forte variance, la deuxième contient les informations relatives à la variance suivante, etc. Le processus est répété jusqu'à l'obtention de la n-ième et dernière composante principale. Les pertes d'informations diminuent d'une étape à la suivante. Ainsi, la reconstruction est exacte si la dimension de la base est la même que le nombre initial de paramètres. Cependant, l'intérêt de l'ACP est justement de réduire cette dimension tout en minimisant l'erreur de reconstruction. Ainsi, seules les premières composantes principales sont généralement considérées. Alexa et coll. [Alexa 00] propose d'ailleurs d'utiliser cette fonctionnalité afin de compresser le mouvement.

Cette réduction des dimensions du vecteur de données permet ainsi de définir des lois sans un nombre trop important de paramètres. Cependant, ces paramètres ne sont pas intuitifs et certains auteurs ont essayé de corrélérer ces paramètres avec des informations de plus haut niveau telles que la vitesse du mouvement par exemple [Gardon 04].

Ces techniques basées sur des connaissances précises du mouvement sont, par nature, très sensibles à la qualité et à l'adéquation des analyses effectuées. En effet, pour reprendre l'exemple de la locomotion, les travaux en biomécanique ont montré qu'il était possible de trouver des différences significatives de trajectoires en fonction du sexe, du statut d'entraînement, de la fatigue, du poids... Il est donc très difficile d'avoir un modèle moyen adaptable à tout type de population et à tout type de situation. De plus, une partie du mouvement est liée à des causes mécaniques (gravité, force de réaction du sol...). Pour des mouvements rapides, faisant intervenir une forte part de dynamique, utiliser ces techniques implique donc de calculer des opérateurs capables de modifier les trajectoires articulaires en fonction des forces externes. Comme la dynamique n'est pas linéaire dans l'espace des trajectoires articulaires, il ne semble pas possible de retrouver ces opérateurs par des analyses statistiques. Les modèles générateurs se proposent donc d'utiliser une description mécanique du corps humain et de calculer les forces et couples à lui appliquer.

### 1.2.5.2 Modèles générateurs

À partir du formalisme de Lagrange ou de celui de Newton, il est possible de dériver un modèle mécanique de tout système poly-articulé [Arnaldi 89, Dumont 90]. Ce modèle est capable de prendre les forces et les couples agissant à tous les niveaux du système pour calculer la configuration angulaire au pas de temps suivant via des intégrateurs numériques optimisés [Villard 95, Cogné 97]. Ces techniques ont largement été employées pour simuler des systèmes inertes. Pour des systèmes utilisant des commandes, il est alors nécessaire de définir des contrôleurs [Lecerf 94] qui calculent les forces et les couples nécessaires pour résoudre une tâche donnée.

Afin d'amener le système à une configuration donnée (une posture imposée par exemple),

une solution assez simple consiste à utiliser un contrôleur de type proportionnel-dérivé :

$$\tau = K_p(\theta - \theta_d) + K_d(\dot{\theta} - \dot{\theta}_d) \quad (1.6)$$

où  $\theta$  est l'état courant du système (la posture courante),  $\theta_d$  est la posture désirée,  $\tau$  est le vecteur de couples à appliquer au système mécanique,  $K_p$  et  $K_d$  sont les gains du contrôleur. Cette approche a, entre autres, été utilisée pour piloter un bras humain [Multon 99]. Cependant, il est très difficile de trouver des valeurs de gains génériques qui satisfassent toutes les situations possibles. Si on cherche à contrôler le bras en imposant une position à l'un des points du squelette, on peut aussi utiliser des multiplicateurs de Lagrange qui agissent alors comme un ressort venant tirer le point du squelette vers sa position. Le problème consiste encore une fois à trouver les bons paramètres pour produire un mouvement naturel.

Pour des tâches plus complexes prenant en compte des minimisations d'énergie ou de couples appliqués aux articulations, il est possible d'utiliser le *contrôle optimal*. Le principe consiste à rechercher les couples qui résolvent la tâche tout en minimisant une fonction de coût. L'avantage de cette méthode est que cette fonction de coût peut intégrer des critères visant à rendre le mouvement plus naturel : minimisation du Jerk [Flash 85], de l'énergie, des forces mises en jeu... Cette méthode a été utilisée pour générer des mouvements sur des humanoïdes de synthèse [Zhao 96, Fang 03]. Cependant, les temps de calcul sont très importants et ne permettent pas d'envisager ces approches pour des applications interactives.

Plutôt que de chercher à minimiser des fonctions, ce qui engendre des temps de calcul importants, une autre approche consiste à travailler dans l'espace d'état  $\{x, \dot{x}\}$  où  $x$  est l'état du système. On analyse alors comment adapter un modèle d'origine pour qu'il vérifie de nouvelles contraintes tout en préservant ses caractéristiques principales dans l'espace d'états [van de Panne 90, Laszlo 96]. Une alternative à toutes ces approches calculatoires est d'effectuer un apprentissage du contrôleur soit à partir d'une base de données de mouvements, soit par renforcement. Généralement, la méthode revient à utiliser un réseau de neurones [Marteau 01, Magdalena 93]. Cet apprentissage peut aussi servir à calculer automatiquement les préconditions et postconditions nécessaires à l'application de contrôleurs dédiés à des tâches limitées, comme s'asseoir ou se lever [Faloutsos 01].

Dans le cas d'un animal et, plus particulièrement d'un humain, le problème est complexe puisque la mécanique n'intervient que partiellement dans le processus de génération du mouvement. Tout mouvement valide du point de vue mécanique n'est pas forcément un mouvement jugé comme naturel pour un humain. Par conséquent, en plus des problèmes de contrôlabilité, il faut tenir compte de cet aspect naturel du mouvement.

Pour apporter plus de réalisme aux modèles générateurs de mouvements humanoïdes, les méthodes les plus courantes s'accordent à utiliser des graphes de postures de contrôle (ou *Pose Control Graph*) [Hodgins 95]. Les nœuds de ces graphes contiennent des postures-clé à atteindre (cf. figure 1.10). Le contrôleur généralement utilisé pour atteindre ces postures est un proportionnel-dérivé. Les gains du contrôleur dépendent partiellement de la morphologie du sujet et leur nombre peut être réduit [Hodgins 97]. Cependant, la détermination de ces gains, même en nombre moins élevé, reste problématique puisqu'ils sont aussi liés à la dynamique propre du mouvement. Ainsi, pour une même succession de postures-clé, on

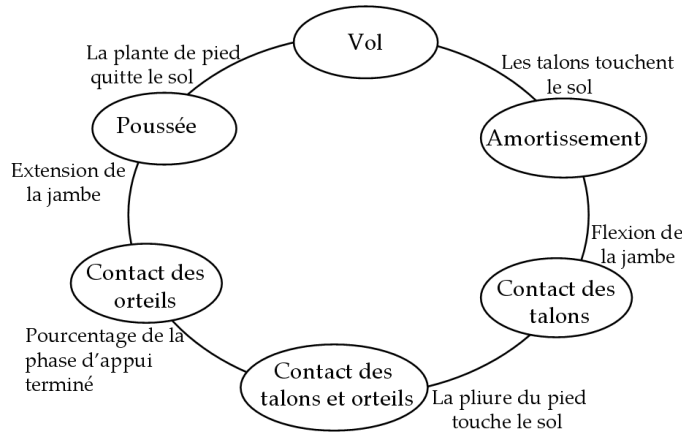


FIG. 1.10 – Exemple de graphe de postures de contrôle pour la course d’un personnage [Hodgins 95].

peut utiliser des gains radicalement différents suivant que le mouvement est effectué rapidement ou lentement. Toutefois, cette technique a l’avantage de tenir compte directement des forces externes et l’utilisateur ne se soucie donc pas de la manière dont le mouvement répond à ces nouvelles contraintes. Ceci a été illustré pour simuler un mouvement de nage dans plusieurs fluides ayant des propriétés différentes [Yang 04]. Si le graphe comporte des postures qui ne sont pas réalistes d’un point de vue mécanique, il est possible de les corriger en utilisant des techniques d’optimisation [Sharon 05]. Cette approche a été utilisée pour simuler une locomotion bipède 2D tout en préservant le style inclus dans les postures du graphe d’origine.

### 1.2.5.3 Cartes de chemins probabilistes

Une alternative aux modèles descriptifs et générateurs est de travailler dans l’espace des configurations  $E$  formé par tous les degrés de liberté du système. Générer un mouvement revient alors à trouver un chemin dans cet espace  $E$  avec les contraintes suivantes :

- une configuration initiale qui correspond à un point de  $E$ ,
- une configuration finale qui correspond également à un point de  $E$ ,
- un ensemble de contraintes liées à l’environnement et aux possibilités de mouvements du système.

Pour résoudre ce problème, la méthode la plus couramment utilisée est appelée “cartes de chemins probabilistes” [Latombe 91]. Elle consiste à choisir aléatoirement des points de l’espace de configurations  $E$  et à ne garder que ceux qui répondent aux critères (par exemple, ne pas être en collision avec l’environnement). La méthode cherche ensuite à relier ces points par des arcs qui eux aussi doivent vérifier les contraintes.

Cette méthode a été utilisée en robotique pour manipuler des objets dans des environnements très contraints (cf. figure 1.11). Laumond et coll. [Laumond 05] ont identifié deux grandes méthodes de résolution de ce problème : représenter tous les degrés de liberté du système (objet manipulé et personnage) dans l’espace de configuration ou résoudre une partie des degrés de liberté du personnage par cinématique inverse [Koga 94, Kallman 03, Yamane 04]. La deuxième méthode permet de réduire les temps de calcul et de bénéficier



des avantages de la cinématique inverse pour produire des mouvements naturels.

- Cette méthode a été généralisée pour la locomotion [Choi 03, Pettré 03, Esteves 06] :
- la position et l’orientation du personnage sont obtenues par cartes de chemins probabilistes, soit sur les empreintes [Choi 03] soit en considérant le corps comme un cylindre [Pettré 03, Esteves 06],
  - les bras manipulateurs sont calculés par cinématique inverse,
  - et les jambes sont obtenues en mélangeant plusieurs mouvements d’une base de données.

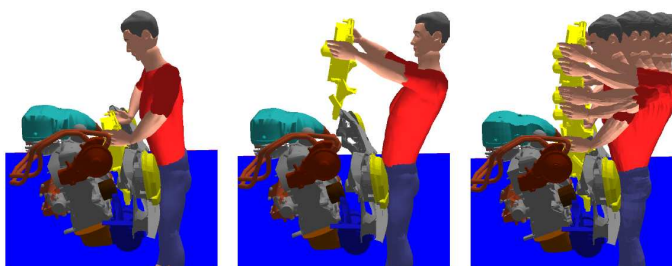


FIG. 1.11 – Planification de mouvements [Laumond 05]. (a) Posture initiale. (b) Posture finale. (c) Succession de postures proposée pour l’animation du personnage.

## 1.2.6 Combinaison de mouvements

Les approches précédentes étaient fondées sur une connaissance générale d’un mouvement donné (forme caractéristique de trajectoires ou postures-clé). À partir de ces données génériques, calculer de nouveaux mouvements revient à trouver les opérateurs capables d’adapter ces données à de nouvelles conditions. Toutefois, vu la très grande variété de mouvements naturels possibles, il est très difficile de proposer des méthodes capables de reproduire tous les styles et toutes les situations. De plus, les connaissances génériques se limitent à un ensemble très restreint de mouvements (généralement la locomotion) et il est très difficile d’extrapoler ces méthodes pour générer d’autres mouvements complexes.

Une alternative consiste à supposer qu’un mouvement peut être représenté comme une combinaison de plusieurs autres. Deux grandes approches sont proposées dans la littérature, les graphes de mouvements et le mélange de mouvements.

### 1.2.6.1 Graphes de mouvements

Avec la multiplication des systèmes de capture du mouvement, il existe maintenant un grand nombre de bases de données de gestes humains. Il faut alors organiser cette base de données pour l’utiliser au mieux. Les contraintes sont de pouvoir gérer un maximum de situations différentes possibles sans pour autant devoir faire une acquisition pour chacune d’entre elles.

La solution la plus couramment utilisée consiste à proposer un graphe de mouvements dans lequel les nœuds sont des configurations du squelette (intégrant posture et parfois vitesse) et les arcs des transitions possibles entre deux configurations [Kovar 02]. Ainsi, pour un unique mouvement, toutes les postures sont potentiellement des nœuds et les

transitions sont liées à la séquentialité dans le mouvement. Toutefois, ces transitions ne sont pas limitées à cette séquentialité mais à une notion de distance entre deux configurations. Par exemple, plusieurs auteurs ont proposé d'utiliser la similarité entre les angles et les vitesses articulaires des postures [Lee 02, Wang 03]. Arikan et coll. [Arikan 02] ont de plus ajouté l'information de vitesse et accélération du torse dans son repère local. Enfin, Brand et Hertzmann [Brand 00] ont, à partir de modèles de Markov cachés, estimé les probabilités liées aux états de transitions sur des mouvements possédant chacun leur style (différents gestes de danse).

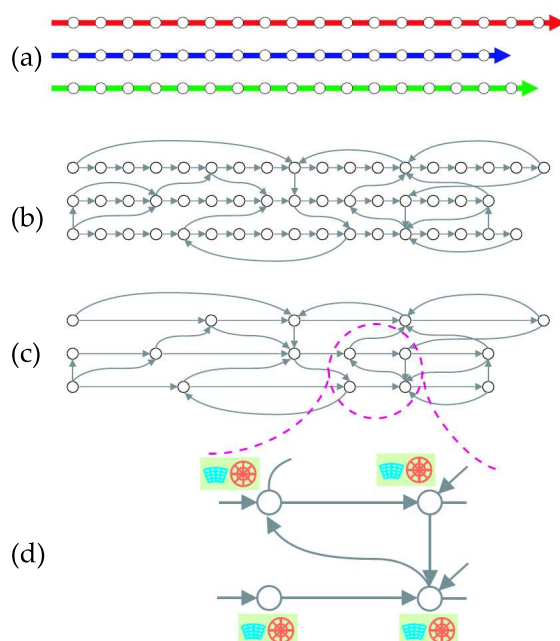


FIG. 1.12 – Graphe de mouvements [Lee 04]. (a) Les trois mouvements composés d'une succession de postures. (b) Les transitions trouvées entre les différentes postures des mouvements. (c) Les nœuds qui n'ont qu'une seule transition de sortie sont regroupés. (d) Une grille est ajoutée au nœud afin de modifier le choix des transitions en fonction de la position de l'objectif (une balle à atteindre).

Grâce à ce calcul de distance entre deux configurations, il est possible de déterminer toutes les transitions possibles dans le graphe de mouvements [Kovar 02] (Schödl et coll. [Schödl 00] appliquent la même technique pour les images vidéo). Ce précalcul est assez long mais s'effectue hors-ligne. Ensuite, générer un mouvement dans ce graphe revient à trouver un chemin menant d'une configuration courante à une configuration souhaitée (cf. figure 1.12b). Lee et coll. [Lee 04] propose d'aller plus loin en regroupant les nœuds qui font partie de la même séquence de mouvements, c'est-à-dire ceux qui se suivent obligatoirement (une seule transition en sortie). Le graphe est alors simplifié et les nœuds restants sont des postures-clé du mouvement (cf. figure 1.12c). Afin d'avoir un contrôle de plus haut niveau, chaque nœud du graphe est associé à des informations qui permettent d'influencer le choix des transitions. Par exemple, afin de pouvoir contrôler un boxeur qui tente d'atteindre une balle déplacée interactivement par l'utilisateur, chaque nœud du graphe contient une grille pour positionner la balle par rapport au personnage (cf. figure 1.12d).

Ces méthodes sont très intéressantes pour l'animation interactive mais nécessitent une base de données très importante. En effet, avec ces méthodes, il n'est pas possible de recalculer de nouvelles configurations. C'est pourquoi, pour couvrir un maximum de configurations différentes, il est nécessaire d'avoir une base de données très volumineuse. À l'inverse, on peut envisager de disposer d'une base de données restreinte et chercher une combinaison de postures, à chaque pas de temps, qui résolve le problème posé.

### 1.2.6.2 Mélange de mouvements

Cette approche part de l'hypothèse qu'une posture peut s'exprimer comme la combinaison de plusieurs autres. Ainsi, marcher tout en saisissant un objet pourrait être la combinaison d'un mouvement de marche naturelle et d'un mouvement de préhension en position statique. Bien entendu, pour effectuer ce mélange, il faut répondre à un certain nombre de contraintes. L'une des plus importantes concerne la synchronisation des différents mouvements à mélanger afin d'assurer une cohérence temporelle. Ainsi, il n'est pas possible de mélanger un saut à cloche-pied droit et un saut à cloche-pied gauche.

Cette synchronisation entre deux mouvements est principalement assurée par des événements identifiés dans chacun d'eux. Par exemple, dans une locomotion, il y a des événements de contact pied-sol, de décollage du pied... Intuitivement, après un contact du pied sur le sol, on peut enchaîner raisonnablement un mouvement qui se base sur un appui de ce même pied (comme un coup de pied par exemple).

Pour mélanger deux mouvements compatibles (deux locomotions par exemple), il faut donc synchroniser ces événements. Le plus évident est de faire correspondre tout d'abord les durées des deux mouvements. Dans cette durée normalisée, il faut ensuite mettre en correspondance les instants où ont lieu les mêmes événements dans les deux mouvements. Cette opération est appelée déformation dynamique du temps (ou *Dynamic Time Warping*). Cette technique présentée par Bruderlin et coll. [Bruderlin 95] pour le domaine de l'animation de personnages permet de compresser ou de dilater des périodes du mouvement (cf. figure 1.13a). Cependant, une attention particulière doit être apportée sur ces changements temporels. En effet, la fonction qui associe le temps réel et le temps adapté doit être monotone afin d'éviter tout retour en arrière dans le temps. De plus, les compressions et dilatations temporelles entraînent des discontinuités au niveau des vitesses du mouvement. Il faut donc en plus s'assurer que cette fonction est de continuité  $C^2$  (cf. figure 1.13b).

La déformation dynamique du temps peut ainsi être utilisée pour synchroniser automatiquement deux mouvements en fonction des contraintes d'appui sur le sol [Ménardais 04b]. Ces auteurs définissent un ensemble d'appuis possibles ainsi qu'un opérateur qui vérifie la compatibilité de ces appuis. Par exemple, un support à gauche uniquement n'est pas compatible avec un support à droite uniquement. Lorsqu'une incompatibilité est détectée, il est alors possible de déformer localement l'échelle de temps d'un ou de plusieurs mouvements. Cette opération est effectuée jusqu'à ce que l'incompatibilité disparaisse. Cette approche se limite cependant aux appuis des pieds sur le sol et devrait être étendue pour gérer d'autres types d'événements pouvant eux-aussi conduire à des incompatibilités.

D'autres critères de synchronisation des mouvements ont été proposés dans la littérature : en faisant correspondre les propriétés cinématiques du mouvement (positions et vitesses) [Ashraf 00, Ashraf 01, Ashraf 03], en détectant les instants où l'accélération des

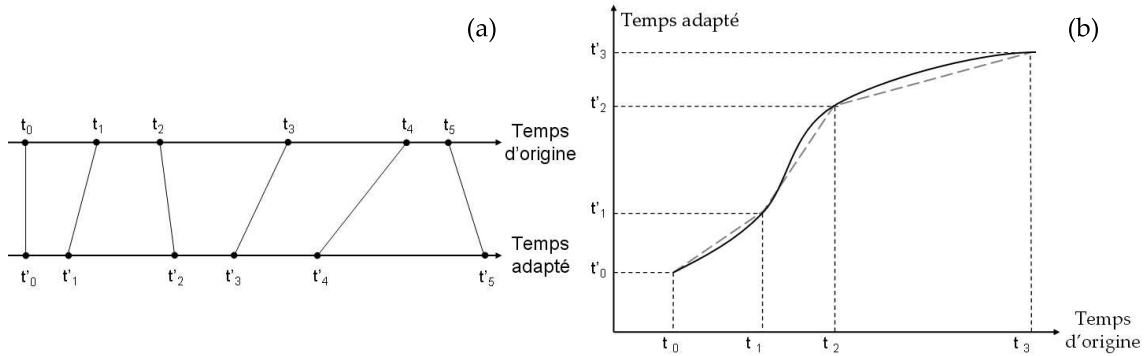


FIG. 1.13 – Déformation dynamique du temps (*Dynamic Time Warping*) [Guo 96]. (a) Les différentes phases des mouvements sont compressées ou dilatées temporellement afin de les faire correspondre. (b) La monotonie et la continuité du recalage temporel sont indispensables pour éviter les discontinuités [Menardais 04a].

extrémités (mains ou pieds) devient nulle ou s'inverse [Bindiganavale 98] ou encore en trouvant des corrélations entre les positions, vitesses et accélérations pour les mouvements cycliques [Rose 96].

Une fois que les échelles temporelles des différents mouvements à combiner sont compatibles (c'est-à-dire que les principaux événements correspondent), il reste à définir un algorithme permettant de mélanger les postures à chaque pas de temps de manière réaliste. Le mélange des trajectoires synchronisées s'effectue principalement par combinaison linéaire (ou somme pondérée) [Guo 96, Wiley 97, Ashraf 00, Park 02, Park 04]. Cette technique a été appliquée à la locomotion humaine en considérant quatre types de démarches [Guo 96] : marche avec des pas courts (**SSW**), avec des pas longs (**LSW**), course à petits pas (**SSR**) et à longs pas (**LSR**). La figure 1.14 montre un exemple d'interpolation appliquée à la locomotion humaine. De manière plus générale, on peut considérer le mouvement humain comme une phrase comportant des verbes (les actions) et des adverbes (les caractéristiques telles que la fatigue, le stress...). Au final, les mouvements de référence de même nature (même action) sont mélangés en interpolant/extrapolant les trajectoires sur l'espace des attitudes caractéristiques [Rose 98]. Cette interpolation peut aussi s'effectuer au niveau des paramètres fréquentiels représentant chaque mouvement [Pettré 03] pour produire des marches avec un virage plus ou moins important par exemple. Toujours concernant la locomotion, il est aussi possible de définir une base de données indexée par des propriétés morphologiques pour adapter la locomotion aux données anthropométriques du personnage [Pronost 05].

Toutes les techniques présentées ci-dessus obligent généralement à connaître la totalité du mouvement à l'avance afin de les recaler temporellement. Cependant, en animation interactive, il est quasiment impossible d'avoir une connaissance à long terme des actions de l'utilisateur. D'autres approches ont donc été proposées pour composer plusieurs mouvements dans ce cas de figure. Dans ces approches, on associe une priorité à chaque mouvement et sur chaque degré de liberté [Boulic 97a, Menardais 04a]. Il est ainsi possible de combiner la cinématique des jambes d'un mouvement de marche à celle des bras

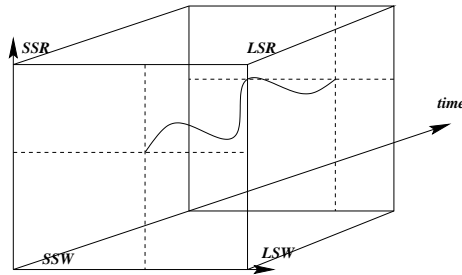


FIG. 1.14 – Une trajectoire articulaire décrite à partir de quatre trajectoires de référence.

lors d'une préhension effectuée à l'arrêt. La continuité du mouvement est assurée par une déformation continue et dérivable des priorités. Au final, le mouvement combiné est une somme pondérée des configurations instantanées de chaque mouvement d'origine. Cependant, cette technique doit préalablement faire appel à une synchronisation par dilatation non-uniforme du temps, comme nous l'avons vu précédemment. Cette déformation doit être capable de proposer une solution quasi-instantanée, avec une connaissance très limitée des événements futurs. Dans certains cas, les résultats peuvent produire des séquences moins réalistes que les approches raisonnant sur la totalité de la séquence.

### 1.2.7 Bilan

Pour représenter le mouvement d'un humanoïde, la quasi-totalité des travaux s'accordent à utiliser des séquences de postures. Ces postures sont alors séparées par un pas de temps fixe pour une représentation image par image ou par des pas de temps variable dans le cas de l'utilisation de postures-clé. La spécification de ces mouvements peut alors se faire en utilisant de la génération de mouvements, de la capture de mouvements ou encore de la combinaison de mouvements.

La génération de mouvements reste peu utilisée parce que sa difficulté de mise en œuvre la destine à des types restreints de gestes. La capture de mouvements, quant à elle, permet d'acquérir des animations qui sont réalistes. Cependant, la nature des systèmes de capture implique des corrections avant de pouvoir utiliser le mouvement. De plus, le coût financier de ces systèmes et de leur utilisation est encore actuellement très important. La combinaison de mouvements permet alors à partir de quelques mouvements capturés d'en produire de nouveaux.

Cette combinaison de mouvements part du principe que l'association de mouvements naturels produit un mouvement également naturel, afin d'éviter de codifier toutes les lois qui gouvernent le contrôle moteur. Cependant, après avoir subi une déformation dynamique du temps puis une combinaison linéaire avec d'autres gestes, un mouvement peut très bien ne plus respecter les lois de la physique. Ainsi, il est possible de diagnostiquer quelles contraintes mécaniques qui doivent être prises en compte pour effectuer ces mélanges [Safonova 05]. Malgré cette limitation, la combinaison de mouvements est très largement employée, même dans les systèmes professionnels d'animation puisqu'elle permet

de calculer intuitivement de nouveaux mouvements sans nécessiter de nouvelles captures. Cependant, contrôler le mouvement finement pour respecter des contraintes de positionnement dans l'espace n'est pas assuré par nature.

A l'inverse, les graphes de mouvements offrent ce type de contrôle. Cependant, le prix à payer est la capture d'un volume très important de mouvements pour échantillonner le plus finement possible l'espace des tâches à effectuer. De plus, si un mouvement est rajouté à la base de données, il est nécessaire de tout recalculer car les transitions peuvent être remises en cause.

Ces deux approches permettent donc de répondre à des besoins différents de l'animation de personnages : interpoler parmi différents styles de gestes pour la combinaison de mouvements et contrôler le personnage simplement en définissant des tâches de haut niveau à effectuer pour les graphes de mouvements. Cependant, il est très difficile avec ces méthodes de tenir compte de ces deux contraintes en même temps. De plus, la base de données nécessaire pour tenir compte de tous les styles et de toutes les tâches serait d'une taille rédhibitoire. Les techniques d'adaptation de mouvements permettent de répondre plus efficacement à ce type de problématique.

### 1.3 Adaptation de mouvements

Puisque le mouvement est composé d'une succession de postures, il est donc possible d'adapter un geste dans sa globalité mais également posture par posture. Cette dernière adaptation des postures est particulièrement appropriée pour les applications interactives dans lesquelles est présent un utilisateur dont le comportement et les interventions sont imprévisibles. Nous avons également vu dans la section 1.2.5 que ces techniques d'adaptation posturale étaient également utilisées par les modèles descriptifs pour corriger, par exemple, les glissements des pieds sur le sol. Ces contraintes comme toutes celles qui proviennent de l'environnement sont cartésiennes alors que nous avons vu dans la section 1.2 que les trajectoires utilisées sont angulaires. Par conséquent, il faut une technique d'adaptation de ces représentations angulaires à des contraintes cartésiennes. Cette technique, appelée cinématique inverse, est décrite dans la sous-section 1.3.1.

Lorsque l'ensemble du mouvement est connu, il est alors possible d'avoir des techniques d'adaptation plus globales qui peuvent parfois utiliser la cinématique inverse. La sous-section 1.3.2 présente ces techniques.

#### 1.3.1 Cinématique inverse

Comme nous venons de le voir, la cinématique inverse est utilisée pour adapter une structure hiérarchique de solides rigides articulés à des contraintes cartésiennes appelées "effecteurs". Cette technique revient à inverser la fonction de cinématique directe  $f$  introduite dans la section 1.1.1 :

$$\theta = f^{-1}(X) \quad (1.7)$$

Le problème est que cette fonction n'est pas bijective et qu'il existe bien souvent un grand nombre de solutions répondant au problème. Prenons le cas d'un bras composé de

deux segments et dont on impose le point de l'épaule et du poignet. Il existe alors une infinité de solutions conduisant à placer le coude sur un cercle [Korein 82]. On dit que le système est redondant. Ce cas de figure arrive obligatoirement lorsque le nombre de degrés de liberté est supérieur au nombre de contraintes. Ainsi, dans l'exemple du bras, on a 3 degrés de liberté à l'épaule et 1 (au moins) au niveau du coude. Si on contraint le poignet, on définit 3 contraintes (une par axe). Il reste donc  $4 - 3 = 1$  degré de liberté qui conduit à un espace de solution à une dimension qui implique effectivement que le coude se situe sur un cercle (cf. figure 1.15a). Pour des chaînes cinématiques plus longues, cet espace de solution devient plus complexe (cf. figure 1.15b).

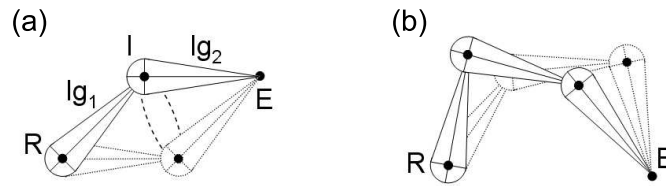


FIG. 1.15 – Cinématique inverse sur une chaîne à deux segments dans un espace à trois dimensions (a) : en contraignant la racine  $R$  et l'effecteur  $E$ , l'articulation  $I$  a une infinité de solutions placées sur un cercle. Dans le cas d'une chaîne à trois segments (b), cet espace de solutions est déjà très complexe.

Le but de la cinématique inverse est donc de choisir l'ensemble des angles  $\theta$  qui permet de respecter les contraintes  $X$  placées sur les effecteurs.

### 1.3.1.1 Solution analytique

Lorsque la structure articulée est très simple, les méthodes analytiques représentent la solution idéale. En effet, elles offrent instantanément la solution au problème de cinématique inverse. Ces méthodes sont généralement appliquées sur des chaînes articulées composées de deux segments tels que le bras ou la jambe [Kondo 91, Lee 99, Wang 99, Tolani 00, Shin 01]. Elles peuvent aussi servir à adapter un personnage sur le sol par exemple. En effet, Ménardais [Ménardais 03] a proposé d'utiliser une solution analytique pour résoudre ce problème. Le contact des pieds sur le sol est assuré par la définition des empreintes à respecter. La solution analytique consiste alors à trouver la hauteur correcte de la racine du personnage en tenant compte de deux critères : les pieds doivent toucher le sol (les longueurs des jambes doivent donc être prises en compte) et la racine doit être la plus proche possible de celle de la posture à adapter. Enfin, les jambes sont à leur tour adaptées de manière analytique.

Cependant, ces méthodes analytiques sont difficiles à mettre en place pour l'ensemble du corps humain puisque sa structure articulée est très complexe, entre 30 à 40 degrés de liberté au minimum. Il faut donc faire appel à d'autres techniques dites "numériques".

### 1.3.1.2 Linéarisation locale

Afin de résoudre la cinématique inverse, il faut inverser la fonction  $f$  (cf. équation 1.7) qui est complexe et clairement non linéaire. Une solution consiste à linéariser localement le

problème autour de la configuration actuelle (*Resolved Motion Rate Control* [Whitney 69]). Les variations angulaires sont alors reliées à celle de l'effecteur par la relation :

$$\Delta X = J\Delta\theta \quad (1.8)$$

Le Jacobien  $J$  fait donc correspondre les variations des degrés de liberté  $\theta$  en variation des paramètres  $X$  de l'effecteur. Il est donc modélisé par une matrice de taille  $M \times N$  qui rassemble les dérivées partielles des paramètres  $X$  de l'effecteur (au nombre de  $M$ ) par rapport aux degrés de liberté  $\theta$  (au nombre de  $N$ ).

$$J_{m,n} = \frac{\delta f_m}{\delta \theta_n}$$

avec  $X = f(\theta)$  et  $f_m$  est le changement du  $m^{\text{ème}}$  paramètre de l'effecteur.

La cinématique inverse est donc résolue en inversant l'équation 1.8 :

$$\Delta\theta = J^{-1}\Delta X \quad (1.9)$$

Cependant, le Jacobien  $J$  n'est pas toujours inversible. Le corps humain possède généralement beaucoup plus de degrés de liberté que l'utilisateur ne fixe de contraintes; le système est dit sous-contraint. C'est alors la pseudo-inverse de Moore-Penrose [Penrose 55] qui est utilisée [Greville 59, Girard 85] :

$$\Delta\theta = J^+\Delta X \quad \text{avec } J^+ = \begin{cases} (J^t J)^{-1} J^t & \text{si } M > N \\ J^t (J J^t)^{-1} & \text{si } M < N \end{cases}$$

Klein et coll. [Klein 83] ont montré que cette pseudo-inverse est optimale, c'est-à-dire que les solutions possèdent une norme euclidienne minimale, pour les systèmes sous-contraints. Pour ces derniers, Liegeois [Liegeois 77] a proposé une forme plus générique qui permet de tenir compte de la redondance des solutions :

$$\Delta\theta = J^+\Delta X + (I - J^+J) \nabla h(\theta)$$

où  $I$  est la matrice identité et  $h$  représente la fonction objectif à minimiser.

Cette fonction  $h$  permet d'appliquer des lois de recherche dans l'espace de solutions fourni par  $\Delta\theta = J^+\Delta X$ . Ces lois sont appelées "tâches secondaires". De nombreuses techniques sont ainsi utilisées pour résoudre le problème de redondance du système : la minimisation d'énergie [Nicolas 04], la minimisation de l'écart de posture par rapport à un mouvement d'origine [Choi 99, Choi 00], l'évitement de collision [Baillieul 86, Nebel 99a, Nebel 99b, Nebel 00], la prise en compte des angles limites aux articulations [Girard 85], le contournement des singularités dans la chaîne cinématique [Sciavicco 87, Sciavicco 96] ou encore la correction d'une locomotion en temps-réel [Boulic 90].

Lorsque le problème nécessite de résoudre deux tâches distinctes, Hanafusa et coll. [Hanafusa 81] ont proposé de les prendre en compte dans l'ordre inverse de leur priorité. Ainsi, lorsque deux contraintes  $\Delta X_1$  (de forte priorité) et  $\Delta X_2$  (de faible priorité) sont appliquées, la variation angulaire est obtenue par l'équation :

$$\Delta\theta = J_1^+(\theta)\Delta X_1 + [J_2(I - J_1^+J_1)]^+(\Delta X_2 - J_2(J_1^+\Delta X_1)) \quad (1.10)$$



Le principal avantage de cette méthode est que la tâche de plus haute priorité  $\Delta X_1$  est assurément vérifiée si c'est possible alors que la tâche de priorité inférieure est respectée au mieux, sans remettre en question  $\Delta X_1$ . Siciliano et coll. [Siciliano 91] a proposé d'étendre ce contrôle par **priorité de contraintes** à un nombre quelconque de tâches. Baerlocher et coll. ont alors fourni des solutions pour diminuer l'ordre de complexité de cette généralisation [Baerlocher 98, Baerlocher 01, Baerlocher 04].

Ces travaux ont été repris par d'autres auteurs pour réaliser des adaptations interactives d'une posture [Yamane 03b] (en utilisant la formulation à deux tâches d'Hanafusa et coll. [Hanafusa 81]) ou d'un mouvement [le Callennec 04] (avec un nombre quelconque de tâches).

A l'inverse, avec une approche utilisant un **mélange des tâches**, aucune d'entre elles n'est vraiment résolue conduisant à un compromis (cf. figure 1.16). Cette expression peut être étendue pour gérer un plus grand nombre de tâches et, donc, de priorités.

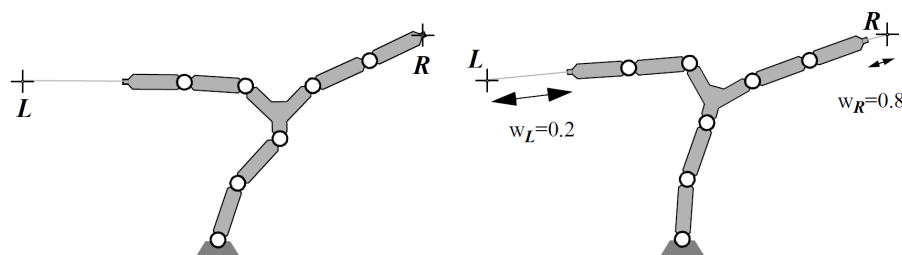


FIG. 1.16 – Cinématique inverse avec contrôle par priorité ou mélange des contraintes [Baerlocher 01].

En animation, Phillips [Phillips 91] a montré l'importance de contrôler le centre de masse pour gérer les notions d'équilibre en quasi-statique (pour les mouvements lents où les accélérations sont négligeables). Ce contrôle permet de gérer des postures complexes tout en vérifiant que la projection du centre de masse sur le sol se trouve bien dans une zone délimitée par les pieds : le polygone de sustentation. Ce dernier définit, dans le plan horizontal, le déplacement maximal du centre de masse qui garantit l'équilibre en quasi-statique. Depuis 1994, ce contrôle du centre de masse est appelé **cinétique inverse** [Boulic 94, Mas 96, Boulic 96]. La figure 1.17 montre la conservation du centre de masse du personnage le long d'un axe vertical au-dessus des pieds. La cinétique inverse est généralement résolue avec une formulation de cinématique inverse avec gestion des priorités [Mas 96, Boulic 96, Boulic 97b, Espiau 98, Baerlocher 00].

Issues du domaine de la robotique [Klein 83, Klein 95], ces méthodes utilisant la linéarisation locale et l'inversion du Jacobien (par inversion, pseudo-inversion ou pseudo-inversion amortie [Maciejewski 90]) se sont largement développées dans le cadre de l'animation d'humanoïdes. En effet, la spécification du mouvement à partir de trajectoires cinématiques externes est plus intuitive que la spécification de configurations angulaires [Badler 93]. Ces méthodes restent donc une référence en matière de cinématique inverse puisqu'elles permettent de tenir compte, entre autres, de lois générales issues de la biomécanique et des neurosciences. Cependant, cette technique nécessite un coût de calcul important qui ne permet pas d'animer plusieurs personnages en temps-réel à 30Hz sur un ordinateur grand

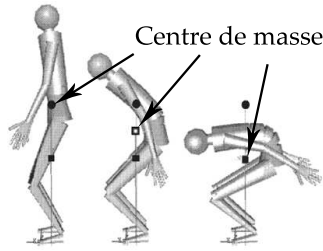


FIG. 1.17 – Utilisation de la cinétique inverse pour contrôler la position du centre de masse sur une même ligne verticale au-dessus des pieds [Boulic 97b]. Elle permet de conserver l'équilibre en posture quasi-statique.

public.

Afin d'éviter l'inversion de la matrice Jacobienne qui est très coûteuse en temps de calcul, une méthode consiste à remplacer la pseudo-inverse par la transposée du Jacobien [Wolovich 84, Das 88, Sciavicco 88]. Cette technique est issue du principe des travaux virtuels introduits par d'Alembert au XVIII<sup>ème</sup> siècle et repris dans le domaine de la robotique par Paul [Paul 81] qui suppose égaux le travail des forces et celui des moments lorsque la variation de posture est faible. L'équation  $\Delta\theta = J^T \cdot \Delta X$  est alors obtenue en faisant une approximation des lois de la dynamique. En effet, cette équation respecte en fait la physique aristotélicienne :  $F = mv$  et  $\tau = I\omega$ .

L'utilisation de la transposée évite l'inversion du Jacobien et permet ainsi de calculer rapidement chaque itération. Cependant, sa convergence est plus lente et la performance globale de cette technique n'est pas toujours meilleure que l'inversion du Jacobien et le résultat est moins précis.

### 1.3.1.3 Optimisation

Afin d'éviter la création et l'inversion coûteuses du Jacobien, une autre approche consiste à utiliser des méthodes d'optimisation non linéaire [Press 92, Gill 81, Girard 85]. Le problème de cinématique inverse se ramène alors à la minimisation d'une fonction de distance entre les paramètres de l'effecteur et ceux de l'objectif.

À partir des nombreuses techniques d'optimisation qui existent pour résoudre ce problème de cinématique inverse, Luenberger [Luenberger 84] a proposé une méthode appelée CCD (Cyclic Coordinate Descent). Elle a été utilisée par Wang et coll. [Wang 91] dans le domaine de la robotique et une version proche du CCD a par ailleurs été utilisée dans le domaine de l'animation d'humanoïdes [Badler 87a, Badler 87b, Zhao 94]. Cette technique est en fait un algorithme itératif qui effectue des adaptations locales aux articulations afin de minimiser des erreurs de position et d'orientation sur les effecteurs. Les premières versions proposaient d'adapter chacun des degrés de liberté un par un. Lander [Lander 98a, Lander 98b] suggère d'utiliser directement une adaptation articulation par articulation. Ainsi, chaque itération parcourt la chaîne cinématique de l'extrémité à la racine et modifie chaque articulation une par une. Lorsqu'une articulation est adaptée, seuls ses degrés de liberté sont considérés comme existants. Le parallèle peut être fait avec

la cinématique inverse qui utilise un Jacobien. En effet, cette adaptation locale revient à ne considérer que les trois colonnes du Jacobien qui sont concernées par les degrés de liberté de l'articulation :

$$\begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = \begin{pmatrix} \frac{\partial f_x}{\partial \theta_i} & \frac{\partial f_x}{\partial \theta_j} & \frac{\partial f_x}{\partial \theta_k} \\ \frac{\partial f_y}{\partial \theta_i} & \frac{\partial f_y}{\partial \theta_j} & \frac{\partial f_y}{\partial \theta_k} \\ \frac{\partial f_z}{\partial \theta_i} & \frac{\partial f_z}{\partial \theta_j} & \frac{\partial f_z}{\partial \theta_k} \end{pmatrix} \begin{pmatrix} d\theta_i \\ d\theta_j \\ d\theta_k \end{pmatrix}$$

Cette adaptation est alors suffisamment simple pour être réalisée directement de manière analytique. La figure 1.18 montre un exemple d'adaptation d'une chaîne cinématique avec une contrainte de position. Cette chaîne est composée de trois segments et seulement deux degrés de liberté marqués par des ronds noirs. La première partie (a) montre la posture initiale qui définit le vecteur d'état  $\theta$  et la contrainte qui est placée au niveau de l'extrémité de la chaîne  $E$  et dont la cible est la position  $Cible$ . L'adaptation du premier segment consiste alors à aligner le segment  $(R, E)$  avec  $(R, Cible)$  illustrée par la partie (b). Enfin, l'adaptation du second segment est réalisée en suivant le même procédé (cf. images (c) et (d) de la figure 1.18).

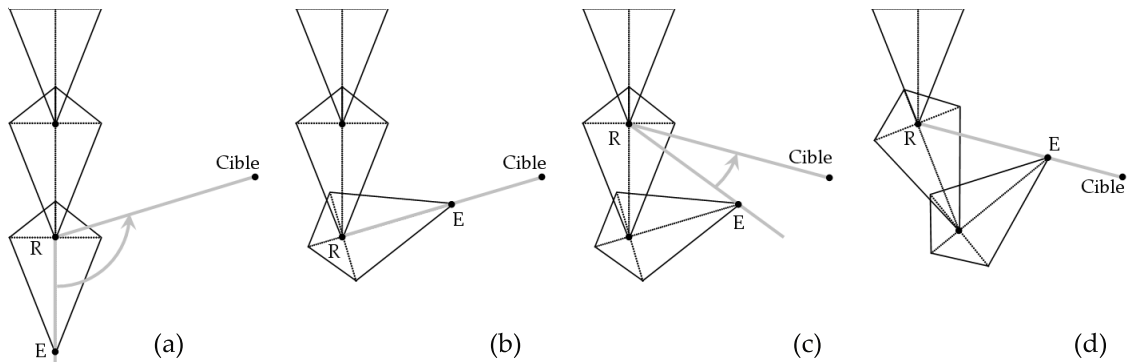


FIG. 1.18 – Exemple d'utilisation du CCD. (a) et (b) Le dernier segment est adapté de manière à aligner  $R$ ,  $E$  et  $Cible$ . (c) et (d) Le premier segment est adapté à son tour.

Malgré un calcul extrêmement rapide de chaque adaptation et le fait que cette méthode converge toujours sans singularité, elle reste très peu utilisée dans le domaine de l'animation d'humanoïdes. En effet, elle comporte de gros inconvénients :

- Il n'est pas possible d'ajouter des tâches secondaires. Cet algorithme ne fournit par l'espace des solutions possibles en tenant compte des redondances mais uniquement une solution. Il faut donc impérativement contrôler l'adaptation afin que l'algorithme converge directement vers la solution souhaitée.
- La convergence du CCD est très rapide au début puis très lente lorsque l'effecteur se rapproche de sa cible. La figure 1.19 montre l'itération qui suit celle présentée dans la figure 1.18. Contrairement à la première itération, l'extrémité ne s'est pas beaucoup rapprochée de la cible. Les itérations suivantes sont dessinées dans la dernière image à droite (d). La convergence ralentit lorsque l'effecteur se rapproche de la cible. C'est la raison pour laquelle Wang et coll. [Wang 91] ont utilisé le CCD uniquement pour

se rapprocher très vite de la solution et a ensuite terminé la cinématique inverse avec une optimisation de type Broyden-Fletcher-Shanno [Luenberger 84].

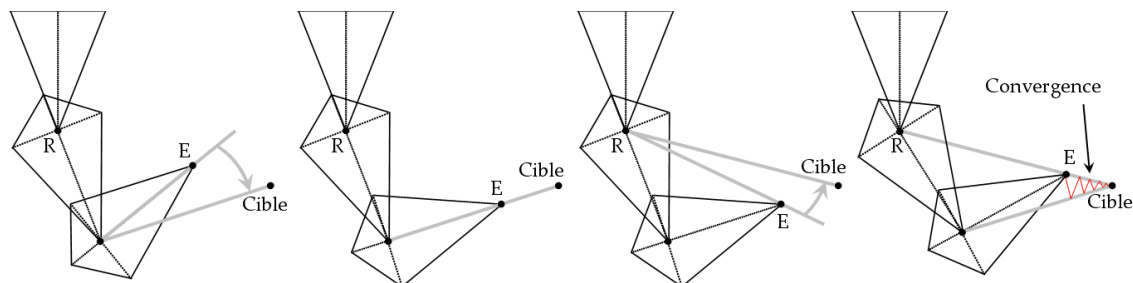


FIG. 1.19 – Cette figure présente la seconde itération qui suit celle de la figure 1.18. L’image de droite montre les adaptations réalisées par les itérations suivantes.

- *L’adaptation n’est pas répartie de manière homogène.* En effet, les premiers paramètres subissent les plus grandes modifications tandis que les derniers sont peu adaptés comme le montre la figure 1.18. Une technique d’amortissement (“*damping*”) peut être employée afin d’améliorer cette répartition. Elle propose de contraindre les variations maximales que peuvent prendre les degrés de liberté à chaque itération. Ainsi, les premiers segments adaptés ne prennent pas toute la déformation et elle est mieux répartie le long de la chaîne cinématique. Le problème est que le nombre d’itérations augmente alors, diminuant ainsi les performances de l’algorithme.

Comme nous venons de le voir, l’utilisation de techniques d’optimisation pour résoudre le problème de cinématique inverse conduit à une solution unique qui ne correspond pas toujours à une posture réaliste du squelette. De plus, même si ces techniques permettent de calculer très vite chaque itération, la lenteur de la convergence implique souvent un nombre d’itérations important.

#### 1.3.1.4 Approche hybride

Récemment, une approche a été proposée pour résoudre ce problème de cinématique inverse dans le cas particulier du mouvement humanoïde [Shin 01]. Cette approche cherche à tirer avantage des techniques décrites précédemment avec, comme objectif, d’offrir un bon compromis entre temps de calcul et réalisme.

L’idée mise en avant dans cet article est, tout d’abord, d’associer une valeur d’importance aux extrémités du corps humain (poignets et chevilles) en fonction de leur proximité avec des objets de l’environnement. Ainsi, par exemple, lorsque les poignets de la femme ne sont pas à proximité d’un objet, l’orientation initiale de ses bras est directement appliquée sur la marionnette (cf. figure 1.20a). A l’inverse, lorsque la table est proche des poignets, leur position doit être conservée (cf. figure 1.20b).

L’algorithme proposé cherche alors une position de la racine du personnage qui essaie de respecter au mieux l’ensemble des contraintes placées aux extrémités en tenant compte de leur importance. Pour chacune de ces contraintes, l’épaule ou la hanche peut se trouver à l’intérieur d’une sphère dont le rayon est la longueur du membre concerné. En considérant que l’orientation du tronc est fixe, l’espace accessible par la racine est la même sphère

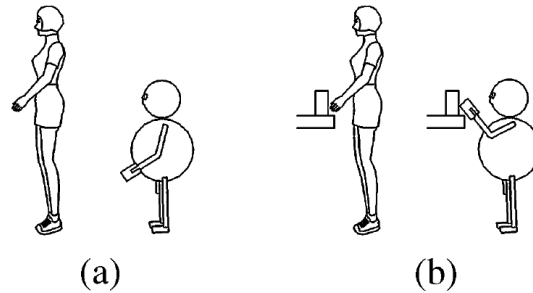


FIG. 1.20 – Importance associée aux extrémités en fonction de leur proximité avec un objet [Shin 01]. (a) Les orientations des bras de la femme sont directement appliquées sur la marionnette puisqu'aucun objet ne se trouve à côté de ces poignets. (b) La conservation de la position des poignets est réalisée puisqu'ils se trouvent à proximité d'un objet.

décalée de la longueur du tronc (cf. figure 1.21). Une estimation de la position de la racine est alors placée à l'intersection des sphères qui définissent son espace accessible et le plus près possible de la position initiale.

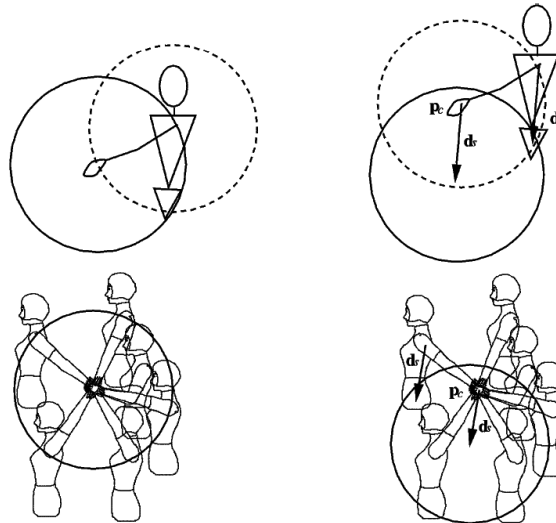


FIG. 1.21 – Espace accessible pour les bras à gauche et la racine à droite [Shin 01].

Un algorithme de descente de gradients conjugués est alors appliqué pour trouver la posture du personnage qui respecte au mieux cette position de la racine et qui utilise des solutions analytiques pour les membres. Le tronc est considéré comme fixe afin d'accélérer l'optimisation.

Cette approche est très intéressante même si elle se limite à un corps simplifié : quatre membres et un tronc rigide. De plus, les contraintes ne peuvent être placées que sur les poignets ou les chevilles. Ce choix se justifie pour l'animation de marionnettes virtuelles mais n'est cependant pas générique. Enfin, avec cette méthode, il n'est pas non plus pos-

sible d'introduire des lois du mouvement ni de contrôler le centre de masse. En résultat, il nous semble que ce type d'approche puisse conduire à des postures non naturelles, comparativement aux méthodes utilisant la linéarisation locale.

### 1.3.2 Edition des mouvements

L'édition de mouvements part de l'hypothèse qu'un geste capturé est naturel et qu'il respecte donc évidemment les lois mécaniques. Cependant, il est nécessaire de le contrôler afin qu'il puisse vérifier de nouvelles contraintes : squelette différent de celui de l'acteur ayant effectué les mouvements, environnement 3D différent (position différente des objets à saisir par exemple) et consignes de l'animateur (guider une partie du squelette de manière interactive). Le problème ici revient donc à savoir comment modifier le geste pour préserver ses caractéristiques originelles tout en vérifiant les nouvelles contraintes.

Les premières méthodes que l'on pourrait rattacher à cette famille d'approches concernent la correction de mouvements de marche pour assurer que le pied est en contact avec le sol [Girard 85, Boulic 92, Ko 93, Unuma 93]. En effet, comme nous l'avons vu dans la sous-section 1.2.5.1, les modèles descriptifs ont rapidement été associés à la cinématique inverse pour assurer que les pieds soient bien en appui sans glissement pendant la phase de contact. Lorsqu'il est question de tâches plus complexes, il est nécessaire de définir un certain nombre de contraintes à respecter dans le mouvement. Ces contraintes sont définies dans l'espace et le temps. On parle alors de contraintes spatio-temporelles ("*spacetime constraints*"). Le problème est qu'il n'est pas concevable de vérifier la contrainte uniquement à l'instant où elle est posée sans tenir compte de la continuité et de la dérivabilité du mouvement.

#### 1.3.2.1 Optimisation des trajectoires

Plusieurs auteurs ont cherché comment appliquer les techniques d'optimisation pour résoudre les contraintes spatio-temporelles [Witkin 88]. Dans ce cas, le problème revient à chercher la succession de postures  $x(t)$  qui minimise une fonction de coût.

L'optimisation des contraintes spatio-temporelles offre une méthode générale pour déformer un mouvement d'origine. Plusieurs facteurs limitent néanmoins le domaine d'utilisation de cette technique :

- Il y a de nombreuses équations de contraintes et variables indépendantes du fait de l'optimisation du système sur un intervalle de temps (et non plus sur une seule posture comme la cinématique inverse).
- La complexité du système (coût de calcul, minima locaux) impose une limitation sur l'échantillonnage de l'intervalle (nombre de postures prises en compte) ainsi que sur la complexité articulaire du sujet animé (nombre de degrés de liberté du squelette).
- L'utilisateur doit spécifier manuellement l'intervalle au cours duquel le mouvement est susceptible d'être déformé. Si celui-ci est mal choisi, une solution incorrecte peut être trouvée. Un trop grand intervalle peut engendrer un ensemble de solutions trop important pouvant induire une convergence du système dans un minimum local non désiré. À l'inverse, un intervalle trop petit peut restreindre de manière trop importante l'espace de solutions, rendant impossible la résolution des contraintes.

- La prise en compte du mouvement sur un intervalle de temps empêche l'utilisation de cette technique lors d'animations en temps-réel.

Plusieurs améliorations et variations à partir des travaux de Witkin et Kass ont été proposées. Cohen [Cohen 92] s'est ainsi attaché à optimiser les algorithmes afin de proposer un outil interactif de déformation du mouvement à l'aide de contraintes spatio-temporelles :

1. Définition de fenêtres spatio-temporelles permettant de diviser le problème (et donc la complexité) en un ensemble de sous-problèmes.
2. Représentation des degrés de liberté par des B-splines permettant d'obtenir une solution  $C^2$  sur tout l'intervalle (et non plus uniquement aux instants discrétisés par le pas de temps).
3. Possibilité de spécifier des contraintes sous forme d'équations conditionnelles ou d'inégalités. Cela offre la possibilité de rajouter des contraintes effectives sous certaines conditions (par exemple une contrainte activée si la distance entre deux articulations est inférieure à un seuil) et limiter le domaine de validité de paramètres (maintenir la hauteur du pied au-dessus du sol).

Liu et coll. [Liu 94] ont reformulé les contraintes spatio-temporelles en utilisant une base d'ondelettes hiérarchiques, afin de réduire la complexité algorithmique. Enfin, Ngo et coll. [Ngo 93] résolvent les contraintes spatio-temporelles en utilisant des algorithmes génétiques.

### 1.3.2.2 Ajout de cartes de déplacement

Pour éviter les coûts et inconvénients de l'optimisation, Bruderlin et coll. [Bruderlin 95] ont proposé une technique qui permet de modifier un mouvement de référence  $m$  afin d'obtenir un nouveau mouvement adapté  $m'$  :

$$m'(t) = m(t) + \Delta(t)$$

La fonction  $\Delta$  représente donc la variation à ajouter aux trajectoires articulaires d'origine afin d'obtenir le mouvement  $m'$  souhaité. Elle se calcule en filtrant la série de positions-clés  $(m'(t_i) - m(t_i), t_i)$ . Cette fonction, appelée **carte de déplacement** (“*displacement map*”) permet d'obtenir la trajectoire  $m'(t)$  déformée en conservant les caractéristiques (hautes fréquences) du mouvement d'origine  $m(t)$ .

Dès lors, l'adaptation du mouvement peut être réalisée en utilisant des contraintes spatio-temporelles uniquement aux instants où elles sont actives [Gleicher 97, Gleicher 98b, Gleicher 01b]. Des techniques de type cinématique inverse sont alors utilisées (cf. section 1.3.1) pour les résoudre. Le résultat de cette opération ne garantit pas une continuité du mouvement. La figure 1.22 montre la trajectoire articulaire à adapter dans la partie (a) et dans l'autre, l'application de la cinématique inverse pour résoudre la contrainte au pas de temps désiré.

Pour corriger ces discontinuités, une carte de déplacements est calculée. La différence entre la nouvelle trajectoire et celle utilisée juste avant d'appliquer la cinématique inverse est filtrée. La figure 1.23a montre cette carte de déplacements filtrée. Elle est alors ajoutée

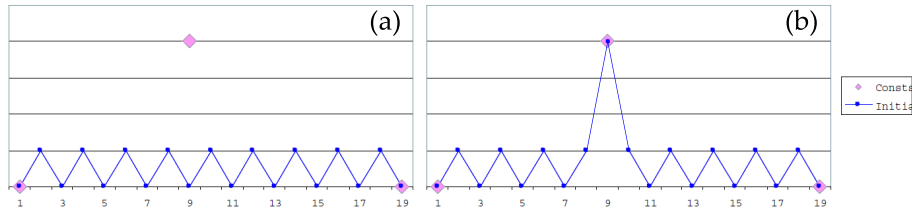


FIG. 1.22 – Discontinuités à résoudre par les cartes de déplacements [Gleicher 01a]. (a) La trajectoire avant adaptation. (b) La contrainte a été vérifiée mais le mouvement est discontinu.

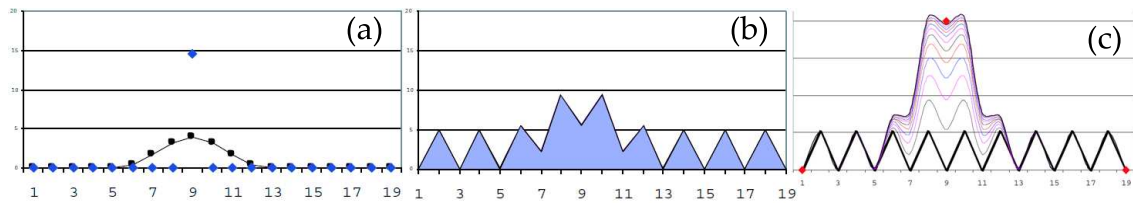


FIG. 1.23 – Filtrage des cartes de déplacements [Gleicher 01a]. (a) La carte de déplacements filtrée, issue de l’adaptation de la figure 1.22. (b) Cette carte est ajoutée à la trajectoire avant adaptation (cf. figure 1.22a). (c) Lancement du processus itératif jusqu’à ce que la contrainte soit vérifiée.

à la trajectoire de référence et fournit la trajectoire de la figure 1.23b. L’opération est alors renouvelée plusieurs fois jusqu’à ce que les contraintes soient respectées (cf. figure 1.23c).

Même s’il faut parfois un grand nombre d’itérations, la complexité de cette méthode se base sur le nombre de contraintes uniquement, contrairement aux méthodes d’optimisation dont la complexité est liée au paramétrage de toutes les trajectoires. Toutefois, le temps de calcul des cartes de déplacement dépend fortement de la méthode de cinématique inverse employée car elle est exécutée un grand nombre de fois. Le Callenec et coll. [le Callenec 04] ont utilisé cette technique pour l’adaptation interactive du mouvement d’un personnage.

En plus d’un ajout de trajectoire, Witkin et Popovic [Witkin 95] ont proposé une méthode appelée “motion warping” qui permet de modifier un mouvement de référence  $m$  afin d’obtenir un nouveau mouvement adapté  $m'$ . Cette adaptation est effectuée en spécifiant une série de contraintes cinématiques sur les trajectoires des paramètres appelées courbes de mouvement (“*motion curves*”). Dans un premier temps, toutes les postures  $m'(t_i)$  sont déterminées de manière à vérifier les contraintes et à se rapprocher au mieux de la posture d’origine  $m(t_i)$ . Le mouvement  $m'$  est alors constitué de ces postures-clés  $(m'(t_i), t_i)$ . La relation entre les mouvements  $m$  et  $m'$  est alors définie par la relation :

$$m'(t) = a(t)m(t) + b(t)$$

où  $a(t)$  et  $b(t)$  sont les valeurs à l’instant  $t$  de deux fonctions caractérisant la déformation (une mise à l’échelle et une translation). En résolvant la relation aux instants  $t_i$ , on obtient deux ensembles de positions clés  $(a(t_i), t_i)$  et  $(b(t_i), t_i)$ . Par interpolation (linéaire, cubique, B-splines cubiques, wavelet B-splines...), il est possible de reconstruire les deux fonctions



et par conséquent le mouvement final déformé  $m'$ . Cette méthode est toutefois plus difficile à contrôler car il faut trouver en plus le paramètre  $a(t)$  à chaque pas de temps.

### 1.3.2.3 Adaptation à un squelette différent

Appliquer un mouvement à un squelette différent peut engendrer des artefacts qu'il est nécessaire de corriger. Ces artefacts se traduisent par le non-respect de contraintes intrinsèquement liées au mouvement, comme le contact sans glissement des pieds en appui sur le sol. Le problème d'adaptation de mouvements à un nouveau squelette peut donc se ramener à la vérification de contraintes spatio-temporelles. Gleicher et coll. [Gleicher 98a] résolvent ces contraintes de manière itérative en utilisant des cartes de déplacement. La figure 1.24 montre le résultat de l'adaptation morphologique obtenu avec cette méthode.

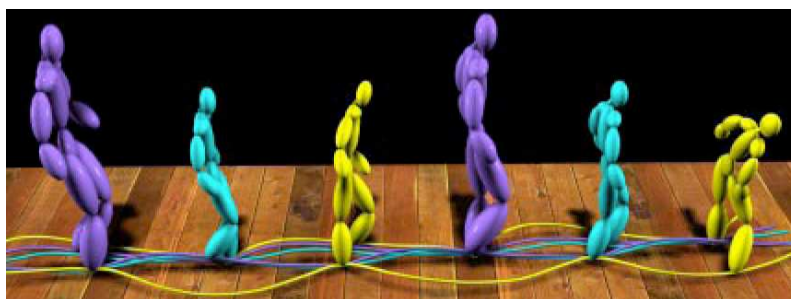


FIG. 1.24 – Adaptation d'un mouvement de marche à différents squelettes [Gleicher 98a].

Komura et coll. [Komura 00, Komura 01] adaptent également le mouvement à partir de contraintes spatio-temporelles. Le squelette est toutefois ici enrichi par la prise en compte des forces liées aux muscles. Cette modélisation plus précise permet de quantifier et adapter certains critères physiologiques.

Monzani et coll. [Monzani 00] utilisent un squelette intermédiaire pour effectuer l'adaptation des trajectoires articulaires sur une nouvelle morphologie. Dans un premier temps, l'utilisateur spécifie les correspondances entre les articulations du sujet acquis (*Performer Skeleton*) et celles du squelette final (*End User Skeleton*). La figure 1.25a montre un exemple de correspondance entre plusieurs articulations du bras et de l'épaule. Dans un deuxième temps, les positions sont mises en correspondance afin de transposer au mieux les orientations du mouvement initial sur le nouveau squelette (figure 1.25b). Au final, une série de contraintes cinématiques est spécifiée afin de conserver certaines caractéristiques du mouvement. La résolution de ces contraintes se fait par l'utilisation de fonctions de déplacement contrôlées par cinématique inverse.

Lee et Shin [Lee 99] proposent de décomposer les trajectoires articulaires en splines hiérarchiques (cf. figure 1.26a). Elles permettent de modifier localement le mouvement et de propager la déformation sur les postures voisines de manière continue. La décomposition hiérarchique permet en outre de travailler localement ou globalement suivant le niveau choisi (modifications des basses ou hautes fréquences). La résolution des contraintes est effectuée à l'aide d'une cinématique inverse numérique combinée avec des résolutions

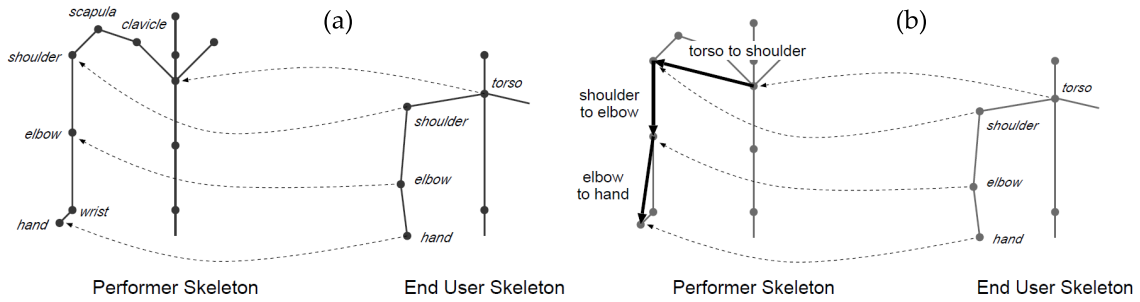


FIG. 1.25 – Utilisation d'un squelette intermédiaire pour l'adaptation morphologique [Monzani 00].

analytiques pour les membres. La figure 1.26b montre une adaptation temps-réel d'un mouvement de marche sur une morphologie qui change dynamiquement.

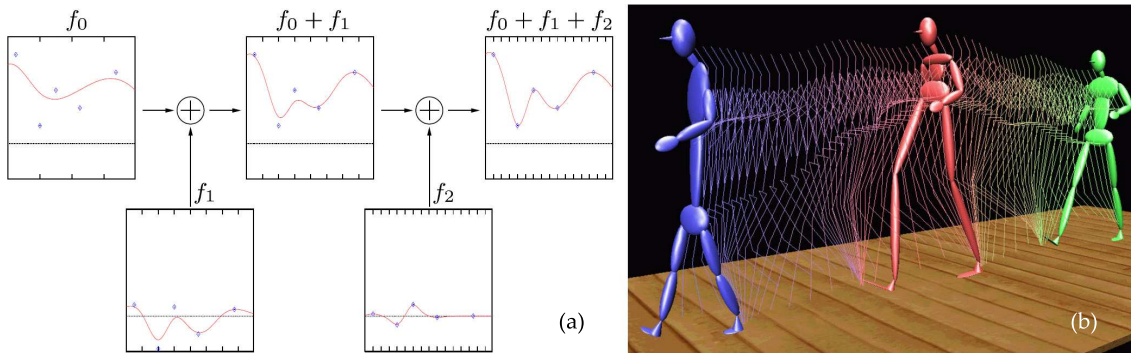


FIG. 1.26 – Utilisation de splines hiérarchiques (a) pour effectuer une adaptation morphologique (b) en temps-réel [Lee 99].

### 1.3.2.4 Prise en compte de la dynamique

Toutes les méthodes d'adaptation présentées ci-dessus ne concernent que la modification de trajectoires pour vérifier des contraintes spatio-temporelles. Cependant, le mouvement humain comprend d'autres types de contraintes, liées principalement au respect des lois de la mécanique. Ainsi, la première approche [Witkin 88] consiste à coupler des contraintes spatio-temporelles sur les positions  $x(t)$  et les forces  $f(t)$  mises en jeu. Le problème revient alors à résoudre une fonction d'énergie  $G$  qui est définie comme l'intégrale d'une fonction de pénalité à minimiser sur l'intervalle. Ainsi pour obtenir une contrainte de moindre effort, il est possible de définir et minimiser une fonction objectif représentant la consommation d'énergie sur ce même intervalle :

$$G = h \sum_i |f_i \dot{q}_i|$$

où  $h$  est le pas de temps de discrétisation sur l'intervalle,  $f_i$  est le vecteur des forces à l'instant  $i$  et  $q_i$  le vecteur position des membres du squelette articulé au même instant.

$f_i$  et  $q_i$  sont les variables indépendantes à déterminer. Chacune de ces fonctions sont intégrées dans des contraintes physiques à résoudre. L'optimisation utilise Newton-Raphson de premier ordre sur les contraintes et de second ordre sur la fonction objectif. Le Jacobien  $J_{ij}$  définit le noyau dans lequel les contraintes sont résolues et le Hessien  $H_{ij}$  améliore la résolution des contraintes :

$$J_{ij} = \frac{\partial C_i}{\partial S_j}$$

$$H_{ij} = \frac{\partial^2 G}{\partial S_i \partial S_j}$$

où  $C_i$  sont les contraintes,  $S_j$  est le vecteur de paramètres et  $G$  est la fonction objectif. Pour résoudre ces contraintes, on peut aussi faire appel à un processus d'optimisation qui engendre lui-aussi des temps de calcul très importants. Cette méthode a principalement été employée sur des systèmes mécaniques comportant peu de degrés de liberté comparativement à l'homme. Pour réduire le temps de calcul, il est donc nécessaire d'avoir recours à des modèles simplifiés de squelette pour effectuer l'optimisation du système mécanique [Popović 99]. D'ailleurs, plusieurs auteurs se sont focalisés sur le contrôle du mouvement du centre de masse en phase aérienne [Sulejmanpašić 05] ou avec différents types d'appui [Pollard 00].

Ces approches optimisent généralement des couples articulaires. La résolution passe principalement par le calcul des dérivées premières de ces couples qui engendrent des complexités en  $o(n^2)$  où  $n$  est le nombre de degrés de liberté. Pour réduire cette complexité à  $o(n)$ , Fang et coll. [Fang 03] ont proposé une nouvelle formulation de contraintes dynamiques particulièrement bien adaptée à la gestion des contacts (dont le contact au sol). D'une manière générale, le principe revient à remplacer les couples articulaires par les accélérations angulaires multipliées par la masse pour chaque segment :

$$h(x) = \int_{t=t_s}^{t_f} \left( \sum_{i=1}^n (m_i \ddot{q}_i(x, t))^2 \right) dt \quad (1.11)$$

où  $m_i$  est la masse liée au segment  $i$  et  $\ddot{q}_i$  son accélération angulaire.

Pour chacune de ces approches basées sur de l'optimisation spatio-temporelle, il est toutefois nécessaire d'optimiser les forces mises en jeu au niveau des articulations au cours du temps, ce qui entraîne des temps de calcul importants. Pour réduire l'espace de recherche, des contraintes sur les moments de forces ont été introduites. Elles consistent à se rapprocher d'une forme caractéristique qui inclut donc non seulement les contraintes mécaniques mais aussi partiellement le style [Liu 02, Abe 04] :

$$\min_S \sum_{i=0}^n (C_i - S(t_i))^2 \quad (1.12)$$

où  $S(t_i)$  est une spline définissant une contrainte sur l'évolution des couples au cours du temps,  $C_i$  est le couple contenu dans le mouvement courant.

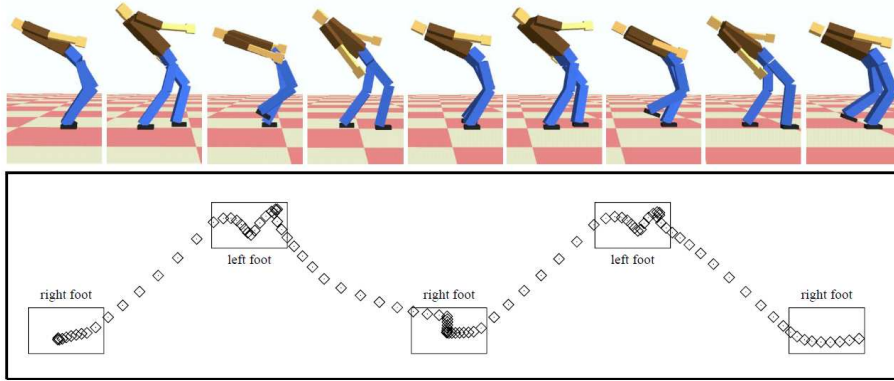


FIG. 1.27 – Maintien de l'équilibre en conservant le ZMP dans le polygone de sustentation [Tak 00].

Au lieu d'optimiser un ensemble de paramètres sur la durée de la séquence, un autre principe consiste à filtrer le mouvement pour prendre en compte un ensemble de contraintes dynamiques. Le plus courant concerne le respect de l'équilibre qui peut être formalisé comme le maintien d'un point virtuel *Zero Moment Point* ou *ZMP* dans le polygone de sustentation (cf. figure 1.27). Le *ZMP* est défini comme le point où la résultante des moments est nulle [Vukobratovic 90] :

$$\sum_i m_i (r_i - Z) \times \ddot{r}_i = \sum_i (r_i - Z) \times m_i \times g \quad (1.13)$$

où  $g$  est l'accélération due à la gravité,  $Z$  est le *ZMP*,  $m_i$  et  $r_i$  sont respectivement la masse et la position du segment  $i$ . Il est ainsi possible de définir une contrainte d'égalité à vérifier à chaque image [Aydin 99a, Aydin 99b, Tak 00, Shin 03]. Pendant les phases aériennes, le centre de masse doit suivre une trajectoire parabolique d'accélération égale à  $g$  et le moment cinétique doit rester constant. Ces deux contraintes ont aussi été ajoutées à un processus itératif exécutant séquentiellement une résolution aux instants des contraintes spatio-temporelles et un filtrage pour assurer la continuité (de manière comparable à ce qui est proposé avec les cartes de déplacements [Gleicher 01b]). Ces méthodes ont largement été utilisées en robotique pour simuler la locomotion de robots [Sardain 04].

La méthode a été étendue pour ne plus avoir ce type de processus itératif en cherchant des solutions optimales à chaque pas de temps. On parle alors de filtre dynamique [Tak 02, Yamane 03a]. En effet, les auteurs utilisent un double filtrage des paramètres pour vérifier les contraintes dynamiques. Un premier filtre de Kalman permet de modifier les paramètres du mouvement initial  $x(t) = [P(t), q^0(t), q^1(t), \dots, q^n(t)]$  où  $P(t)$  est la position et l'orientation de la racine du personnage et  $q^i(t)$  sont les états des autres degrés de liberté. Pour tenir compte de la dynamique, on définit alors le vecteur d'état  $s = [x \ \dot{x} \ \ddot{x}]$ . Les contraintes cinématiques et dynamiques sont représentées par des égalités :

- amener des points du squelette à une position imposée :  $h_{ik}(s^*(t)) = z_{ik}(t)$ ,
- respecter l'équilibre en assurant que le *ZMP* se trouve dans le polygone de sustentation :  $h_e(s^*(t)) = z_e(t)$ ,
- et limiter les couples articulaires pour ne pas dépasser des valeurs maximales :

$$h_c(s^*(t)) = z_c(t).$$

Le filtre de Kalman prédit l'état suivant  $s^*(t)$  qui est corrigé pour tenir compte de toutes les contraintes :

$$s(t) = s^*(t) + K_k \{ [z_{ik}(t) - h_{ik}(s^*(t))] + [z_e(t) - h_e(s^*(t))] + [z_c(t) - h_c(s^*(t))] \} \quad (1.14)$$

où  $K_k$  est le gain du filtre de Kalman. Le résultat de ce filtre peut engendrer une incohérence entre  $x(t)$ ,  $\dot{x}(t)$  et  $\ddot{x}(t)$  qui est corrigée par un second filtre. Cette méthode s'applique bien à des petites modifications du mouvement d'origine mais peut poser des problèmes si les adaptations sont trop importantes.

Une autre alternative à l'optimisation consiste à étendre les graphes de postures (*Pose Control Graph*) des modèles générateurs en supposant que les postures sont données à chaque pas de temps grâce aux données capturées. Ainsi, pour des gains élevées de contrôleurs, sans autre modification, le mouvement produit est très proche de celui qui est mesuré avec juste un léger décalage temporel. Cela a été utilisé pour tenir compte localement de contacts et de collisions [Zordan 02, Arikan 05]. Il est aussi possible d'ajouter des masses sur le dos d'un personnage en ne contrôlant qu'un sous-ensemble des degrés de liberté [Oshita 01]. Dans toutes ces approches, la méthode consiste à ajouter des couples articulaires à ceux calculés par les contrôleurs proportionnels-dérivés. Ce couple peut par exemple être calculé comme une impulsion qui est filtrée pour revenir au mouvement initial [Arikan 05].

### 1.3.3 Bilan

L'animation de personnages répond à des contraintes très différentes en fonction du type d'application. Dans des applications interactives, la capture de mouvements peut être utilisée directement pour animer des avatars en temps-réel. Cependant, les temps de calcul ne permettent pas de traiter les trajectoires pour adapter le mouvement à de nouvelles contraintes pour de multiples personnages. A l'opposé, dans le domaine de l'audiovisuel, et en particulier la production de films de synthèse, il est généralement nécessaire d'adapter les données mesurées à un squelette synthétique et à un environnement différents. De plus, dans ce domaine, le mouvement mesuré est modifié manuellement pour accentuer un style imposé par le personnage. Dans ce type d'application, le réalisme est le point-clé et le temps de calcul devient moins primordial. Il est alors tout à fait envisageable d'utiliser des méthodes d'édition de mouvements pouvant tenir compte de la dynamique pour simplifier le travail des animateurs. En effet, ces derniers peuvent se focaliser sur le style et les tâches que le personnage doit réaliser sans se soucier des lois de la physique qui doivent être respectées. Dans ce cas, les méthodes basées sur un modèle mécanique de squelette et incluant un processus d'optimisation peuvent fournir des résultats intéressants. Il existe maintenant des logiciels du commerce capables de résoudre efficacement ce type de problème (Endorphin de Natural Motion).

Cependant, même si le temps de calcul n'est pas primordial, s'il y a un grand nombre de personnages à animer, des techniques moins coûteuses peuvent être envisagées. En effet, si la base de données de mouvements utilisée est assez volumineuse, il est possible de chercher un geste qui correspond au mieux à la situation et d'effectuer une simulation

mécanique uniquement lors des phases de transition [Zordan 05] ou sur un modèle simplifié [Arikan 05]. La deuxième solution semble la plus intéressante car c'est celle qui nécessite le moins de captures de mouvements ; c'est donc la moins onéreuse.

Même si les temps de calculs sont bien inférieurs à ceux nécessaires à l'optimisation d'un système mécanique, ces techniques ne sont pas utilisables dans des environnements temps-réel où l'utilisateur peut intervenir à chaque instant pour modifier les contraintes de l'environnement. Par exemple, dans le domaine des jeux interactifs, le personnage synthétique doit être capable de frapper un ballon qui est lui-même piloté indirectement par un utilisateur. Les graphes de mouvements peuvent répondre à ce type de problématique en préservant complètement le style initial contenu dans les mouvements capturés. Cependant, cette méthode nécessite une grande base de données qui engendre un coût de production et un volume de stockage importants. A l'opposé, les modèles descriptifs se contentent de très peu de données lors de l'animation temps-réel mais requièrent des analyses spécifiques pour chaque mouvement. Si on souhaite préserver un style particulier dans le geste, il est nécessaire d'utiliser des techniques de type ACP. Cependant, les variables qui sont issues de ces analyses ne sont pas intuitives, ce qui rend le contrôle du mouvement très difficile.

L'utilisation de contraintes spatio-temporelles permet au contraire un contrôle intuitif du mouvement. Nous avons vu que le principal problème est alors de résoudre efficacement ces contraintes aux temps imposés et ensuite d'assurer la continuité. Les cartes de déplacement sont maintenant largement utilisées pour répondre à ce type de problème car elles offrent des temps de calcul bien inférieurs à ceux obtenus avec des techniques d'optimisation. Toutefois, ces temps de calcul sont encore généralement trop élevés pour utiliser ces techniques en environnement temps-réel. Ceci est dû en partie au fait que ces techniques raisonnent sur un geste global alors que des modifications peuvent être requises à n'importe quel instant, de manière imprévisible. Une autre raison concerne le nombre d'itérations nécessaire et le temps passé à chaque itération.

Concernant le nombre d'itérations, pour des applications critiques du point de vue du temps de calcul, une alternative est de résoudre le problème à chaque pas de temps en faisant varier de manière continue l'influence de chaque contrainte. Par contre, le temps passé à chaque itération est fortement lié à la méthode de cinématique inverse employée pour résoudre les contraintes. Le problème que cette cinématique inverse doit résoudre provient du fait que les contraintes sont généralement exprimées dans l'espace Cartésien alors que le mouvement est encodé par des trajectoires articulaires. Il serait certainement pertinent de trouver une représentation qui permette de simplifier cette résolution de contraintes. De plus, si cette résolution était capable de respecter directement les lois de la mécanique, un plus haut niveau de réalisme pourrait être atteint sans avoir recours aux techniques d'optimisation de systèmes dynamiques.

Or, en raison des redondances, il existe une infinité de postures qui permettent de résoudre des contraintes imposées dans l'espace Cartésien. Il est donc impossible de résoudre ce problème de manière analytique si on cherche à gérer le corps humain dans son ensemble. Le tableau 1.1 résume les principaux avantages et inconvénients de chaque méthode de résolution.

Les algorithmes de type CCD requièrent très peu de temps de calcul par itération. Cependant, le nombre d'itérations est généralement très élevé pour parvenir à la solution

Méthode	Temps de calcul	Réalisme	Contrôle du COM	Lois biomécaniques
CCD	++	+	-	+
Jacobien	+	+++	+++	+++
Hybride	++	++	-	+

TAB. 1.1 – Comparaison des méthodes principales de cinématique inverse.

qui peut alors comporter une déformation non-homogène le long de la chaîne cinématique [Welman 93]. Comme cette méthode converge vers une seule solution, la prise en compte de lois biomécaniques se limite à définir l'ordre dans lequel les segments sont adaptés. Par exemple, le système peut mobiliser les segments des plus légers aux plus lourds pour minimiser l'énergie cinétique mise en jeu.

Les approches utilisant le Jacobien de la fonction cinématique nécessitent un nombre plus restreint d'itérations. Ce nombre dépend principalement du pas de temps et de la norme du vecteur  $\Delta X$  exprimant la translation entre la position courante et désirée. Cependant, le temps de calcul nécessaire pour chaque itération est beaucoup plus important car il faut calculer numériquement ou symboliquement la Jacobienne puis l'inverser. Cette inversion peut se faire soit avec la transposée (qui requière un plus grand nombre d'itérations) soit avec la pseudo-inverse car le système est redondant. La tâche secondaire permet de prendre en compte des lois générales de la biomécanique mais implique de calculer l'opérateur de projection sur le noyau de la fonction  $(I - J^+ J)$  qui nécessite lui aussi un grand nombre de calculs. Toutefois, les résultats sont beaucoup plus réalistes que ceux obtenus avec des approches de type CCD. De plus, les masses peuvent être prises en compte dans la fonction à inverser, de manière à avoir un contrôle sur la position du centre de masse.

L'approche proposée par [Shin 01] offre des temps de calcul inférieurs à ceux obtenus avec les autres méthodes grâce à l'utilisation de solutions analytiques et à une simplification du squelette justifiée par le domaine d'application : la manipulation de marionnettes virtuelles. Ce travail ne permet donc pas d'utiliser un squelette complexe ni de gérer des contraintes placées sur n'importe quel point du corps. Enfin, cette méthode ne prend pas en compte la répartition de masses pour contrôler la position du centre de gravité. Cependant, cette méthode semble être un premier pas vers un algorithme ayant un très bon rapport qualité sur temps de calcul.

Récemment, une approche originale [Grochow 04] a été proposée pour résoudre des problèmes de type cinématique inverse en se basant sur une base de données de mouvements capturés. Le système revient à rechercher la posture la plus crédible dans la base de données pour résoudre les contraintes et, éventuellement, à extrapoler. Cependant, il est nécessaire d'avoir des bases de données volumineuses qui peuvent en plus poser des problèmes de continuité entre deux postures successives.

Ces techniques de cinématique inverse sont essentielles au contrôle d'un personnage en interaction avec son environnement. Comme ces techniques sont des briques d'approches plus globales d'animation de personnages, elles doivent offrir des postures réalistes pour un temps de calcul moindre.





## Chapitre 2

# Contexte et objectifs

Les travaux mis en avant dans ce mémoire s'inscrivent dans le domaine de l'animation et de la simulation de personnages humanoïdes en temps-réel. De nombreuses applications se placent dans ce cadre, comme les jeux vidéos, les simulations comportementales ou encore les simulations en réalité virtuelle. Ce dernier domaine est d'ailleurs l'une des thématiques mises en avant par l'appel d'offre de l'ANR (Agence Nationale de la Recherche) réalisé par le RNTL (Réseau National des Technologies Logicielles) cette année 2005. Les personnages peuvent alors être contrôlés interactivement par l'utilisateur ou posséder leur propre comportement et ainsi réagir instantanément à des stimuli provenant d'autres personnages ou encore de l'utilisateur.

Lorsque les personnages sont **contrôlés interactivement par l'utilisateur**, ce dernier donne en temps-réel des instructions sur les mouvements à effectuer. Il est alors essentiel que ces mouvements soient automatiquement et instantanément adaptés au nouvel environnement du personnage. Ainsi, dans un jeu vidéo par exemple, un joueur peut contrôler son personnage en temps-réel et décider de lui faire ouvrir des portes qui peuvent avoir des poignées placées à différentes hauteurs. Un module d'adaptation de mouvements en temps-réel est donc nécessaire.

Les personnages peuvent également **avoir leur propre autonomie** et prendre ainsi leurs propres décisions en temps-réel. Dans ce cas, il faut les doter d'un modèle comportemental. Généralement, ce dernier est composé de trois parties : la perception, la décision et l'action. La perception permet d'observer l'environnement du personnage et d'en déduire, dans la partie décision, les actions à réaliser. Enfin, la dernière partie se charge d'exécuter ces actions. Il est alors courant d'utiliser des modèles de mouvement qui permettent d'alléger le travail de la couche comportementale (cf. figure 2.1a). Le problème est que la définition de modèles paramétriques de mouvements est complexe. En effet, à partir de peu de paramètres, pour des raisons de contrôlabilité par l'utilisateur, ces modèles doivent être en mesure de trouver le bon mouvement à utiliser et de l'adapter. C'est la raison pour laquelle ils sont généralement limités à un type de mouvement. Nous proposons donc de définir un niveau d'adaptation de mouvements afin de simplifier la création des modèles de mouvements. Ces derniers n'auront alors plus qu'à trouver le mouvement le plus approprié et à en déduire, à partir des paramètres fournis par la couche comportementale, les contraintes qui doivent être respectées (cf. figure 2.1b).

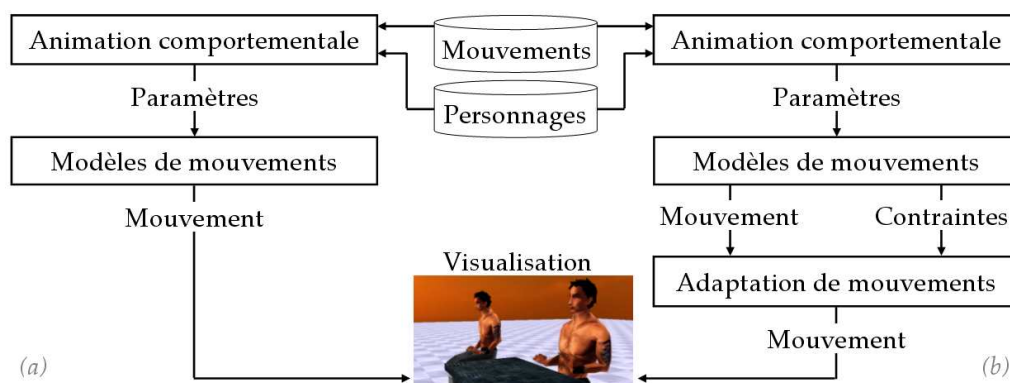


FIG. 2.1 – Utilisation de l’adaptation de mouvements pour l’animation comportementale. Dans la partie (a), les modèles de mouvements doivent tout gérer. Avec un moteur d’adaptation de mouvements illustré dans la partie (b), les modèles ne s’occupent plus que du choix des mouvements et de l’interprétation des paramètres en contraintes.

L’adaptation de mouvements est donc nécessaire pour ces deux types d’applications puisqu’elle permet de modifier un mouvement pour l’adapter au nouvel environnement du personnage. Cependant, cette adaptation doit aussi permettre de prendre en compte la morphologie du personnage. En effet, que les mouvements aient été capturés (cf. section 1.2.4), qu’ils aient été créés par des animateurs à l’aide de logiciels spécialisés tels que 3DSMax ou FilmBox, ou qu’ils soient générés à l’aide des techniques présentées dans la section 1.2.5, ils restent dépendants de la morphologie initiale du personnage pendant cette capture ou cette création. Il faut donc les refaire dès que l’un de ces deux paramètres change, ce qui est le cas dès que l’on souhaite animer de nombreux personnages différents dans des environnements virtuels variés. Notre travail s’inscrit donc dans l’adaptation de ces mouvements afin de pouvoir réutiliser un même geste dans des conditions différentes. Par exemple, à partir d’un mouvement unique de préhension d’un objet devant soi, nous souhaitons pouvoir adapter le mouvement de manière à attraper l’objet quelle que soit sa position, dans la limite du raisonnable bien entendu.

Deux grandes approches sont proposées dans la littérature afin de résoudre ce type de problème. La première consiste à créer de grandes bases de données de mouvements. Ces mouvements sont alors mélangés par des techniques d’interpolation (cf. section 1.2.6.2) ou encore utilisés dans des graphes de mouvements (appelés “*motion graphs*”) qui permettent de précalculer toutes les transitions entre leurs différentes postures (cf. section 1.2.6.1). Cela permet alors de pouvoir très rapidement passer d’un mouvement à un autre de manière réaliste. Cependant, cela nécessite d’importantes bases de données pour pouvoir adapter le geste du personnage à des contextes très différents. Ces techniques ne sont donc pas compatibles avec des applications interactives comme les jeux vidéos qui ont une ressource mémoire très limitée. De plus, le précalcul de toutes les transitions prend un temps considérable et l’ajout d’un nouveau mouvement dans la base de données nécessite de refaire le calcul de toutes les transitions vers ce nouveau mouvement. Les travaux présentés dans ce mémoire se rapprochent par conséquent davantage de la seconde approche qui consiste à

faire de l'adaptation de mouvements en respectant des contraintes fournies (cf. section 1.3). Ces contraintes peuvent être de différentes natures même si les contraintes géométriques, c'est-à-dire celles qui concernent les positions et les angles sont les plus intuitives à utiliser.

Dans sa thèse effectuée au sein de l'équipe SIAMES, Stéphane Ménardais [Ménardais 03] a proposé une chaîne d'animation cinématique permettant à la fois de faire l'adaptation aux différentes morphologies de personnages (cf. section 1.3.2.3) mais aussi de faire la synchronisation et le mélange de ces mouvements (cf. section 1.2.6.2). La figure 2.2 montre qu'à chaque pas de temps, les postures issues des mouvements utilisés dans les actions sont mélangées en temps-réel afin de n'avoir qu'une seule posture finale, celle du personnage. Ce travail permet donc en temps-réel de s'affranchir des problèmes de morphologie initiale des mouvements de référence, qu'ils soient capturés ou créés par des animateurs.

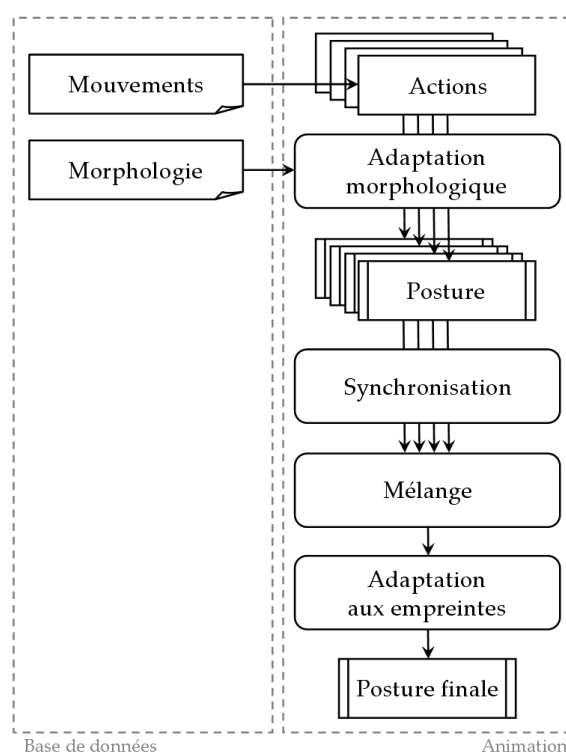


FIG. 2.2 – Chaîne d'animation proposée par Stéphane Ménardais [Ménardais 03].

Cette chaîne d'animation propose néanmoins qu'une adaptation au sol. **L'objectif de cette thèse est donc de proposer des méthodes d'adaptation des mouvements à des contraintes génériques** qui peuvent provenir de l'environnement du personnage (interaction avec le décor ou d'autres personnages) ou simplement du choix de l'utilisateur. Nous proposons donc de combiner une nouvelle méthode d'adaptation de mouvements avec la chaîne d'animation présentée ci-dessus afin d'avoir au final une adaptation à des contraintes quelconques et pour tout type de morphologie humanoïde (cf. figure 2.3).

Dans la mesure où nos travaux concernent exclusivement les applications interactives (interaction avec l'utilisateur ou entre plusieurs personnages dans le cadre d'animations

comportementales), les mouvements de référence doivent être mélangés en temps-réel sans que l'on ait connaissance de l'avenir du geste. En effet, le lancement d'une nouvelle action peut survenir à n'importe quel moment, entraînant l'activation d'un nouveau mouvement dans l'animation du personnage. Il faut donc le mélanger avec les mouvements déjà en cours d'animation (cf. section 1.2.6.2). Or, à un pas de temps donné, un mouvement fournit une posture. Ce sont donc les postures issues de chacun de ces mouvements qui doivent être mélangées pour être appliquées sur le personnage. Ainsi, seule la posture résultant de ce mélange est connue du module d'adaptation. Par conséquent, les adaptations proposées dans cette thèse doivent modifier cette posture appelée **posture de référence**. Il est toutefois possible d'utiliser les travaux de cette thèse dans des applications non interactives et disposant de la connaissance du futur. Il suffit alors comme dans les techniques de cartes de déplacement présentées dans la section 1.3.2.2 d'utiliser des filtres afin de passer de l'adaptation posturale à une adaptation de mouvement.

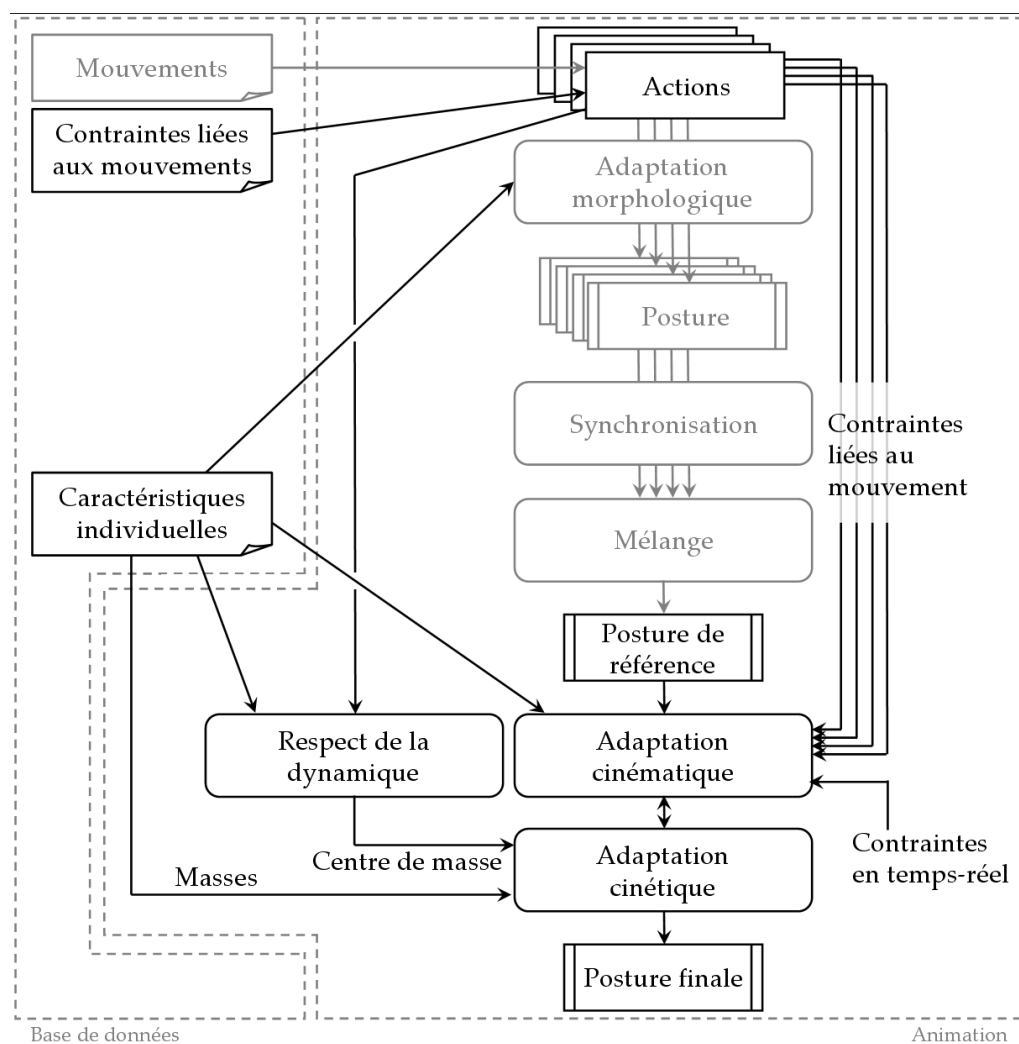


FIG. 2.3 – Chaîne complète d'adaptation des mouvements.

Les contributions de cette thèse (en noir dans la figure 2.3) sont décomposées en cinq parties détaillées ci-après :

- représentation des données (contraintes et caractéristiques individuelles),
- prise en compte de ces données par les actions,
- adaptation cinématique aux contraintes,
- adaptation cinétique aux contraintes,
- et respect des lois de la dynamique.

La partie “Prise en compte de ces données par les actions” n’est pas décrite dans ce mémoire parce qu’elle ne consiste qu’à lire les nouvelles données et à les fournir aux modules concernés. Ce mémoire se concentre donc sur les quatre autres parties qui sont le cœur de l’adaptation de mouvements.

### Représentation des contraintes

Le premier travail présenté dans le chapitre 3 concerne l’identification des grandes familles de contraintes qui sont utilisables afin d’adapter facilement le mouvement. Nous souhaitons que notre moteur d’adaptation puisse être utilisé par le plus grand nombre aussi, il faut que son contrôle soit extrêmement simple. Les contraintes sont alors décomposées en deux grandes familles :

- Celles qui sont *intrinsèquement liées à la nature du mouvement*. Par exemple, dans un mouvement d’applaudissement, le contact des mains est indispensable pour préserver la nature même du geste. De la même manière, lorsqu’un personnage s’assoit sur une chaise, il est important d’avoir une contrainte afin que les fesses soient en contact avec l’assise de la chaise. Ces contraintes doivent donc être associées directement au mouvement au même titre que les trajectoires. C’est la raison pour laquelle la base de données illustrée à gauche dans la figure 2.3 contient un nouveau format de fichiers contenant les contraintes liées aux mouvements.
- Celles que l’utilisateur souhaite ajouter en temps-réel. Ces contraintes sont principalement imposées à cause d’une interaction entre le personnage et le décor (par exemple la hauteur de la poignée d’une porte) ou encore avec d’autres personnages.

Toutes ces contraintes sont de nature géométrique et cinématique. De par l’environnement géométrique qui nous entoure, ce sont celles qui sont les plus intuitives à contrôler. La contribution suivante consiste justement à adapter la posture de référence à ce type de contraintes.

### Adaptation cinématique aux contraintes

Le module d’adaptation cinématique aux contraintes est la clé de voûte des travaux de ce mémoire. En effet, son rôle est d’adapter la posture de référence conformément à ces contraintes. Cependant, l’utilisateur doit pouvoir contrôler très simplement cette adaptation. Par conséquent, il ne doit pas contraindre tous les degrés de liberté du corps humain, surtout en temps-réel. La chaîne cinématique représentant le squelette du personnage est alors un système sous-contraint : il y a moins de contraintes que de degrés de liberté. Une infinité de solutions existe donc toujours pour respecter ces contraintes. L’algorithme d’adaptation doit alors respecter les contraintes imposées tout en préservant au mieux la posture de référence. Ainsi, tous les degrés de liberté restants sont retrouvés à partir de cette posture.

Nous souhaitons pouvoir animer et adapter des dizaines de mouvements de personnages en temps-réel. La notion de performance est par conséquent un critère très important pour la définition de l’algorithme présenté dans le chapitre 4.

### Adaptation cinétique aux contraintes

L’utilisation d’une adaptation cinématique aux contraintes permet à l’utilisateur d’adapter très intuitivement et rapidement la posture de référence. Cependant, la posture obtenue n’est pas toujours réaliste selon les contraintes fournies. En effet, lorsqu’un personnage se penche en avant pour attraper une cible qui se trouve très loin devant lui par exemple, il arrive un moment où il devrait perdre l’équilibre comme l’illustre la figure 2.4. Cette notion d’équilibre n’est pas prise en compte par la cinématique puisqu’elle nécessite la connaissance des masses des segments du personnage. La cinétique est justement la prise en compte de ces masses. Grâce à elles, il est possible de déterminer la position du centre de masse du personnage, couramment appelé COM dans la littérature. Ce dernier représente le centre de gravité de l’humanoïde ce qui permet de gérer son équilibre dans le cas de posture quasi-statique ou encore de vérifier certaines lois de la physique comme cela est utilisé dans la dernière contribution “Respect des lois de la dynamique”. Cette adaptation cinétique aux contraintes est décrite dans le chapitre 5.



FIG. 2.4 – Exemple d’adaptation cinématique sans gestion de l’équilibre.

L’utilisation des adaptations cinématique et cinétique dans la même chaîne de traitements ne peut se faire de manière indépendante. En effet, chacune d’entre elles modifie la posture afin de répondre à ses propres contraintes, au risque de défaire l’adaptation réalisée par l’autre module. Il faut, par conséquent, trouver un algorithme commun qui permette de les utiliser ensemble. Cet algorithme est détaillé à la fin du chapitre 5.

### Respect des lois de la dynamique

Grâce aux adaptations précédentes, il est possible de vérifier un certain nombre de contraintes mais aussi de contrôler la position du centre de masse.

Le chapitre 6 propose d’introduire une adaptation des deux phases d’un saut (prise d’impulsion et phase aérienne) afin de respecter au mieux les lois de la physique. En effet,

le centre de masse du personnage doit répondre à certaines règles. Pendant la phase aérienne par exemple, il doit suivre une trajectoire parabolique due au fait que le personnage est soumis uniquement à la gravité. Lors de la prise d'impulsion, son accélération décrit les forces mises en jeu pour effectuer le saut. Le chapitre 6 propose d'utiliser un système simplifié qui est limité au contrôle du centre de masse. Ainsi, cette adaptation des sauts est effectuée en déterminant la nouvelle trajectoire du centre de masse. A chaque pas de temps, cette position du COM est respectée par l'adaptation cinétique présentée dans le chapitre 5.

Le chapitre 6 propose d'introduire une adaptation d'un saut afin de respecter au mieux les lois de la physique. En effet, le saut est un mouvement très utilisé en animation, par exemple dans les jeux vidéo, et son réalisme dépend très fortement du respect des lois de la dynamique. Pendant les deux phases qui le composent (prise d'impulsion et phase aérienne), le centre de masse du personnage doit répondre à certaines règles. Pendant la phase aérienne par exemple, il doit suivre une trajectoire parabolique due au fait que le personnage est soumis uniquement à la gravité. Lors de la prise d'impulsion, son accélération décrit les forces mises en jeu pour effectuer le saut. Le chapitre 6 propose d'utiliser un système simplifié qui est limité au contrôle du centre de masse. Ainsi, cette adaptation des sauts est effectuée en déterminant la nouvelle trajectoire du centre de masse. A chaque pas de temps, cette position du COM est respectée par l'adaptation cinétique présentée dans le chapitre 5.

Le domaine d'application de ces travaux est l'adaptation des animations de dizaines d'humanoïdes en temps-réel. Notre but est donc de trouver le meilleur compromis entre réalisme des adaptations d'un côté et rapidité d'exécution de l'autre.





## Chapitre 3

# Représentation et spécification des contraintes

Toutes les adaptations de mouvements présentées dans ce mémoire s'appuient sur l'utilisation de contraintes. Dans le cadre d'interaction entre différents personnages ou encore entre un personnage et son environnement, les contraintes qui sont naturellement appliquées sont de nature géométrique (position et orientation) et cinématique (vitesse et accélération). Ces contraintes sont très intuitives à définir même par un utilisateur qui n'est pas animateur de profession.

Dans la mesure où nos algorithmes d'adaptation travaillent sur une posture à un instant donné, toutes ces contraintes deviennent des contraintes géométriques. En effet, le contrôle de la vitesse et de l'accélération est simplement la conséquence de l'enchaînement des contraintes géométriques à chaque pas de temps. C'est la raison pour laquelle nous avons fait le choix de respecter ce type de contraintes dans l'ensemble des travaux présentés dans ce mémoire.

Dans le chapitre précédent, nous avons vu que deux grandes familles de contraintes peuvent être spécifiées : les contraintes intrinsèquement liées au mouvement et celles spécifiées en temps-réel. Dans ce chapitre, nous proposons une représentation unique de toutes ces contraintes qui permet de répondre à un certain nombre de critères :

- cette représentation doit **fonctionner avec les deux grandes familles de contraintes** présentées précédemment.
- elle doit de plus être **compatible avec les différents modes de contrôle des contraintes**. Le mode de contrôle par priorité respecte les contraintes dans l'ordre décroissant de leur importance alors que le mode de contrôle par mélange tient compte de toutes les contraintes simultanément (cf. section 1.3.1.2).
- les contraintes doivent **pouvoir s'appliquer à n'importe quel point du corps humain**, y compris un point sur sa peau par exemple. Dans la littérature, il est usuel de contraindre les articulations du personnage. En effet, comme elles font partie de la chaîne cinématique qui représente le squelette humain, il est plus facile de les contraindre.
- il doit être possible d'appliquer des **contraintes plus complexes** que celles classiquement utilisées, à savoir des contraintes de position ou d'orientation. Il est donc

important que leur représentation permette d'en ajouter de nouvelles ultérieurement.

Cependant, les parties traitant de l'adaptation cinétique et dynamique utilisent en plus une ou plusieurs contraintes spécifiques. Par exemple, pour contrôler l'équilibre, il faut connaître la position souhaitée du centre de gravité du personnage ainsi que la répartition des masses de son corps. Pour contrôler le saut effectué par un humanoïde, il faut pouvoir modifier sa longueur ou la hauteur maximale qu'il peut atteindre. Ces contraintes sont très spécifiques puisqu'elles ne servent pas à contraindre une partie du corps humain mais uniquement des données externes comme la trajectoire du centre de masse. De plus, elles sont peu nombreuses et parfaitement identifiées aussi il n'est pas nécessaire d'avoir une représentation générique pour elles. Enfin, elles ne sont utilisées que lorsque l'adaptation cinétique ou dynamique est activée. Elles seront par conséquent détaillées dans les chapitres 5 et 6 et ne sont pas traitées dans le reste de ce chapitre.

Pour que de nouvelles contraintes puissent être ajoutées sans que l'algorithme d'adaptation n'ait besoin d'être modifié, nous proposons dans ce chapitre d'ajouter un nouveau niveau de représentation des contraintes. Ce niveau décrit les contraintes qui sont utilisées en interne par l'algorithme d'adaptation. Ainsi, deux niveaux coexistent comme l'illustre la figure 3.1 :

- la représentation externe qui définit l'ensemble des contraintes mis à la disposition de l'utilisateur ;
- la représentation interne qui décrit les contraintes qui sont gérées par l'algorithme d'adaptation.

Un opérateur de transformation de la représentation externe vers la représentation interne est donc nécessaire pour chaque type de contraintes manipulé par l'utilisateur.

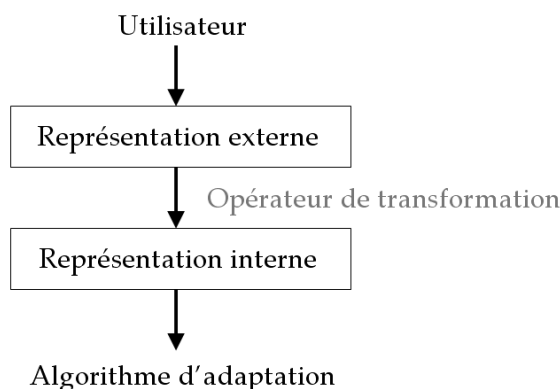


FIG. 3.1 – Deux niveaux de représentation des contraintes.

L'ajout d'une nouvelle contrainte signifie donc la création d'une représentation externe de cette contrainte ainsi qu'un opérateur de transformation vers la représentation interne. L'algorithme reste inchangé et peut directement prendre en compte ce nouveau type de contraintes. De plus, l'algorithme peut ainsi être spécialisé afin d'offrir de meilleures performances.

La section 3.1 décrit la représentation externe des contraintes qui est fournie à l'utilisateur. Cette représentation des contraintes est unique cependant, suivant la famille de contraintes, la spécification de ces contraintes est différente. En effet, les contraintes intrinsèquement liées au mouvement sont associées hors-ligne au mouvement alors que les contraintes temps-réel sont créées et modifiées au moment de l'exécution du programme. Ces deux manières de spécifier les contraintes sont définies dans la section 3.2. La section 3.3 décrit la représentation interne des contraintes qui est utilisée par l'algorithme d'adaptation. Enfin, une discussion sur le contenu de ce chapitre est proposée dans la section 3.4.

### 3.1 Représentation externe des contraintes

Dans tous les mouvements de préhension, une contrainte doit être appliquée sur la main. Cependant, la position de la cible ainsi que le moment de saisie de l'objet sont variables d'un mouvement à l'autre. Aussi, une contrainte nécessite la définition de paramètres. Certains de ces paramètres sont communs à toutes les contraintes (par exemple, l'instant d'activation d'une contrainte) alors que d'autres dépendent du type de la contrainte. En effet, une contrainte cherchant à atteindre une position de l'espace cartésien n'a pas le même objectif qu'une autre cherchant à conserver une orientation. La première nécessite la définition de la position de la cible alors que la deuxième a juste besoin de l'orientation souhaitée. Cette section est par conséquent décomposée en deux parties : les paramètres génériques (décrits dans la sous-section 3.1.1) qui sont communs à toutes les contraintes et les paramètres spécifiques qui dépendent de la nature des contraintes (décrits dans la sous-section 3.1.2).

Une contrainte  $C_i$  est donc modélisée par l'ensemble de ces paramètres génériques  $PG_i$  et spécifiques  $PS_i$  :

$$C_i = \{PG_i, PS_i\}$$

#### 3.1.1 Paramètres génériques

Les paramètres génériques d'une contrainte sont les suivants :

$$PG_i = \{Type_i, Id_i, Seg_i, Decal_i, Portee_i, Imp_i, Prec_i, Act_i, Desact_i, dAct_i, dDesact_i\}$$

Le paramètre  $Type_i$  est le type de la contrainte qui définit la nature de l'objectif à atteindre : une position de l'espace cartésien ou une orientation par exemple.  $Id_i$  est l'identificateur de la contrainte qui permet de la retrouver parmi l'ensemble des contraintes utilisées. Nous avons choisi d'utiliser une chaîne de caractères afin de pouvoir associer des informations sémantiques à l'identification. Ainsi, une contrainte sur la main qui fait une préhension peut, par exemple, s'appeler 'mainActive'. Les paramètres  $Seg_i$  et  $Decal_i$  sont utilisés pour donner la position de la contrainte  $CP_i$  sur le corps humain, c'est-à-dire le point contraint. Ils représentent le segment et le décalage local dans le repère de ce segment.  $Portee_i$  définit la portée de la contrainte sur la chaîne cinématique. Le paramètre  $Imp_i$  représente l'importance de la contrainte par rapport aux autres. La précision souhaitée entre le point contraint  $CP_i$  et son objectif est définie par le paramètre  $Prec_i$ . Il permet de respecter plus ou moins finement certaines contraintes et moins précisément d'autres.

Enfin, le paramètre  $Act_i$  (resp.  $Desact_i$ ) est l'instant d'activation de la contrainte (resp. de la désactivation) et  $dAct_i$  (resp.  $dDesact_i$ ) représente le temps nécessaire pour que la contrainte soit activée (resp. désactivée).

Les principaux paramètres sont détaillés dans les sous-sections suivantes.

### 3.1.1.1 Type de contraintes

Les contraintes sont utilisées pour adapter le mouvement. Cela signifie que chaque contrainte est associée à un objectif. Le paramètre  $Type_i$  est justement utilisé pour décrire le comportement attendu par l'algorithme d'adaptation face à une contrainte  $C_i$ . Dans la littérature, deux types de contraintes sont principalement utilisés : les contraintes de position et d'orientation. Nous proposons ici d'étendre cette représentation des contraintes afin de gérer une plus grande famille d'adaptations. Ainsi, trois grands types de contraintes sont proposés : les contraintes de distance, les contraintes d'espace et celles d'orientation. Cependant, cette liste pourra être encore étendue à de nouveaux types dans le futur si le besoin apparaît.



FIG. 3.2 – Utilisation de contraintes de distance. La figure montre une contrainte de contact (distance nulle) des coudes sur une table.

Les **contraintes de distance** sont les plus utilisées. En effet, elles permettent de contrôler la distance entre un point de l'humanoïde et une cible mais également de prendre en compte les contacts, lorsque la distance est nulle. Ainsi, trois types de contraintes de distance sont utilisés :

- par rapport à une position 3D. La position  $CP_i$  de la contrainte doit être conservée à une certaine distance du point 3D fourni. Cette contrainte est par exemple utilisée lors de la préhension d'un objet. Avec une distance nulle, il est possible d'assurer le contact du point  $CP_i$  avec l'objet. La figure 3.2 illustre l'utilisation de ce type de contrainte.
- par rapport à un point du corps humain. Cette fois, au lieu de considérer un point fixe de l'environnement, cette contrainte cherche à placer le point contraint  $CP_i$  à une certaine distance de l'autre point du corps humain fourni.
- par rapport à une autre contrainte. Ce type permet de rapprocher ou d'éloigner les positions de deux contraintes pour les maintenir à une distance donnée l'une de l'autre. Bien évidemment, il est nécessaire que la contrainte réciproque existe.

Les **contraintes d'espace** sont utilisées pour contraindre un point du corps dans une zone définie en 3D. Par exemple, en appliquant le mouvement d'un personnage discutant

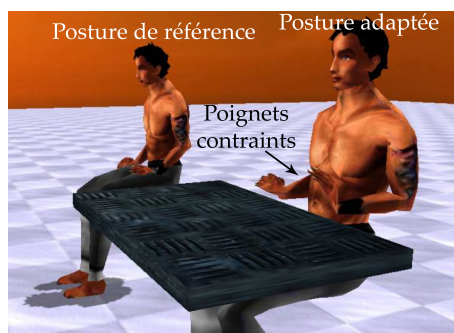


FIG. 3.3 – Utilisation de contraintes d'espace. Dans cet exemple, le mouvement de référence est celui d'une personne assise qui parle avec les mains. Son environnement étant libre, il posait régulièrement ses mains sur ses cuisses. Au premier plan, le même mouvement est appliqué sur un personnage qui a une table devant lui. Les poignets sont contraints à rester au-dessus de la table.

assis sur une chaise à un environnement où l'humanoïde est finalement dans un fauteuil, il faut pouvoir contraindre les coudes afin qu'ils restent au-dessus des bras du fauteuil. Il suffit donc de décaler l'espace autorisé des coudes vers le haut afin que ces derniers ne pénètrent plus dans les accoudoirs. Deux types de contrainte d'espace sont proposés :

- espace autorisé. Cette zone en 3D définit l'espace dans lequel le point contraint  $CP_i$  doit évoluer. La figure 3.3 montre l'utilisation d'une telle contrainte.
- espace interdit. À l'inverse de la contrainte précédente, cet espace définit la zone dans laquelle le point contraint ne peut se trouver. Dans l'exemple décrit précédemment d'un personnage discutant dans un fauteuil, ce type de contrainte peut être utilisé afin de définir l'espace des bras du fauteuil dans lequel le point  $CP_i$  ne peut pénétrer.

Les **contraintes d'orientation** permettent de définir classiquement l'orientation souhaitée pour un segment. Lors de l'animation d'un serveur portant un plateau, il est impératif d'imposer une contrainte d'orientation sur la main afin que celle-ci reste horizontale par exemple.

Enfin, il reste à définir les **contraintes de support**. Ces contraintes sont utilisées pour assurer le respect des appuis des pieds du personnage sur le sol conformément à la phase de support en cours : bipodale (les deux pieds au sol), monopodale (un seul pied au sol) ou aucun appui (phase aérienne). Ces contraintes sont séparées des autres parce qu'elles doivent toujours être vérifiées quelle que soit l'adaptation demandée. Ainsi, elles ne possèdent pas de priorité et sont toujours résolues. Ces contraintes de support ne sont donc pas traitées de la même manière que les autres.

Les types de contraintes que nous venons de définir permettent de spécifier un comportement. Ce dernier doit être pris en compte par les algorithmes d'adaptation présentés dans les chapitres 4, 5 et 6. Cependant, ces comportements sont associés à des objectifs différents. Une contrainte de distance par rapport à une position, par exemple, a besoin de cette valeur de distance ainsi que d'une position 3D. À l'inverse, une contrainte d'espace autorisé a seulement besoin de la définition de cet espace. Ces objectifs sont finalement des paramètres spécifiques et sont définis dans la sous-section 3.1.2. Cependant, un grand nombre de paramètres sont uniquement utilisés pour caractériser la contrainte (son em-

placement, son activation...). Ils sont indépendants de l'objectif visé et par conséquent communs à tous les types de contraintes. Ce sont ces paramètres qui sont décrits dans les sous-sections suivantes.

### 3.1.1.2 Emplacement des contraintes

Nous souhaitons pouvoir contraindre n'importe quel point du corps humain, y compris un point sur la peau. Cela permet, par exemple, de gérer une parade avec un point quelconque de l'avant-bras comme illustré dans la figure 3.4, ce qui est impossible pour les techniques se limitant aux centres articulaires.



FIG. 3.4 – Exemple d'une parade effectuée avec un point quelconque de l'avant-bras.

Dans la revue de littérature, en animation, nous avons vu que le squelette est représenté par des solides rigides articulés. Aussi, il est nécessaire de spécifier le segment  $Seg_i$  sur lequel va être rattachée la contrainte  $C_i$ . Afin de pouvoir contraindre n'importe quel point du solide attaché au segment  $S_i$ , nous utilisons un vecteur 3D défini localement dans le repère de ce segment afin d'obtenir le décalage  $Decal_i$  entre la position  $CP_i$  de la contrainte et la racine du segment  $Seg_i$ . Ainsi, pour définir une contrainte placée sur le coude, il faut soit l'associer au segment de l'avant-bras avec un décalage nul soit utiliser le segment du bras avec un décalage égal à sa longueur.

L'utilisation de ce décalage pour contraindre un point quelconque est illustré par la figure 3.5. La partie (a) montre la posture de référence avec le décalage, la contrainte et la cible à atteindre. La partie (b) montre la posture adaptée de façon à vérifier cette contrainte.

### 3.1.1.3 Portée des contraintes

Dans la hiérarchie du squelette, tous les segments en amont de celui de la contrainte peuvent être utilisés pour l'adaptation. Cependant, l'utilisateur peut décider de contrôler le nombre de segments mis en jeu. La figure 3.6 montre trois contraintes avec des portées différentes. La contrainte  $C_1$  est placée sur la main et peut solliciter le bras et le tronc. A l'inverse, la contrainte  $C_2$  placée sur l'avant-bras ne peut mobiliser que le bras. Enfin, la contrainte  $C_3$  est appliquée sur la tête et utilise l'ensemble de la chaîne cinématique. Dans

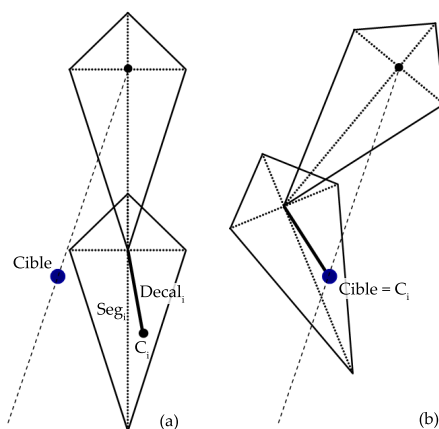


FIG. 3.5 – Décalage utilisé pour spécifier la position des contraintes. La partie (a) montre la posture de référence et la partie (b), la posture finale adaptée pour vérifier la contrainte.

le cas d'une préhension, par exemple, ce paramètre  $Portee_i$  d'une contrainte  $C_i$  permet de n'utiliser que le bras ou tout le tronc suivant la manière dont il est défini.

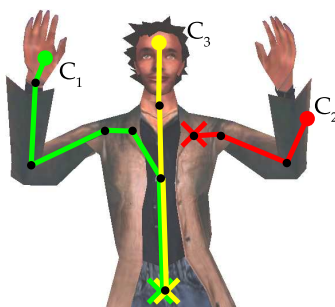


FIG. 3.6 – Portée des contraintes sur la chaîne cinématique. La contrainte  $C_1$  peut solliciter le bras et le tronc pour faire son adaptation alors que la contrainte  $C_2$  ne peut utiliser que le bras. La contrainte  $C_3$  est un autre exemple de portée d'une contrainte placée sur la tête.

#### 3.1.1.4 Importance des contraintes

Lorsque l'on souhaite adapter une posture à plusieurs contraintes, il est important de différencier les contraintes prioritaires des autres. En effet, si l'utilisateur souhaite que son personnage essaie d'attraper un objet avec sa main droite sans lâcher la poignée qu'il tient avec sa main gauche, il faut donner plus d'importance à cette deuxième contrainte qu'à la première. Cette importance est spécifiée par le paramètre  $Imp_i$  pour une contrainte  $C_i$  et est une valeur réelle strictement positive. Cette manière de quantifier l'importance d'une contrainte est classique dans la littérature. On retrouve d'ailleurs deux grands modes de contrôle des contraintes en fonction de ces priorités (cf. section 1.3.1) :

- par priorité de contraintes [Baerlocher 01]. Les contraintes sont respectées dans l'ordre de leur importance. Ce paramètre est alors appelé une priorité. Ainsi, la contrainte

de plus forte priorité est respectée la première et la contrainte suivante tente d'être vérifiée sans remettre en question la précédente. Cette adaptation est réitérée pour toutes les contraintes.

- par mélange de contraintes. Cette fois, chaque contrainte est prise en compte en fonction de la valeur de son importance comparée à la somme de toutes les autres. Ce paramètre est alors appelé un poids. Toutes les contraintes sont donc prises en compte pour l'adaptation et mélangées en fonction de leur poids. Un compromis est donc trouvé entre toutes ces contraintes.

Les travaux mis en avant dans ce mémoire sont compatibles avec ces deux modes de contrôle afin d'offrir le maximum de fonctionnalités à l'utilisateur. Avec les mêmes contraintes, l'utilisateur peut décider de choisir le mode de contrôle souhaité au moment de l'exécution. La figure 3.7 illustre l'utilisation de ces deux modes de contrôle. Dans ces trois images, deux contraintes  $C_1$  et  $C_2$  sont placées sur les mains (droite et gauche respectivement). Les deux sphères représentent les cibles de ces contraintes. La partie (a) montre l'utilisation du mode par priorité de contraintes. La contrainte sur la main droite a une priorité supérieure et est par conséquent complètement vérifiée tandis que l'autre contrainte est respectée au mieux. Dans les parties (b) et (c), un contrôle par mélange de contraintes est utilisé. Les priorités  $P_1 = 4$  et  $P_2 = 1$  sont utilisées dans la partie (b) impliquant que la contrainte  $C_1$  est respectée au  $\frac{P_1}{P_1+P_2} = \frac{4}{5}$  alors que la contrainte  $C_2$  est respectée à  $\frac{1}{5}$ . Dans la partie (c), les deux contraintes ont la même priorité et se partagent donc l'adaptation de la posture.

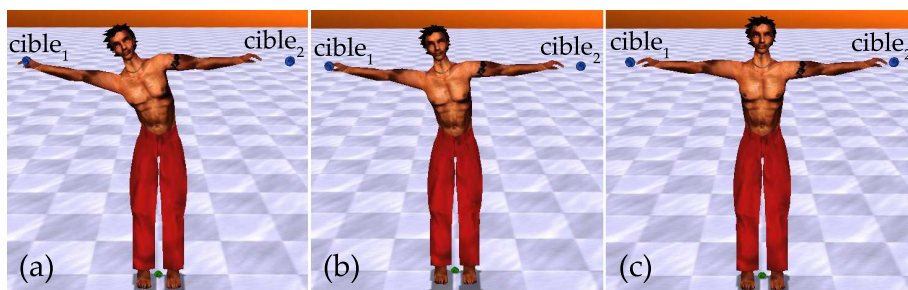


FIG. 3.7 – Deux contraintes  $C_1$  et  $C_2$  sont appliquées sur les deux mains. Les sphères représentent les cibles de ces contraintes. La priorité des contraintes respecte toujours la contrainte de plus forte priorité comme illustrée par la partie (a). La partie (b) montre le mélange de contraintes avec un poids de valeur 4 pour la main droite et 1 pour la gauche. La partie (c) utilise cette fois des poids de même valeur pour les deux contraintes sur les poignets.

### 3.1.1.5 Activation des contraintes

Lors d'un mouvement, les contraintes ne doivent pas être respectées en permanence. Il faut donc pouvoir spécifier l'instant d'activation de la contrainte, c'est-à-dire le moment à partir duquel cette contrainte doit être vérifiée mais également l'instant de sa désactivation. Cependant, comme les contraintes doivent être modifiées de manière continue pour ne pas créer de discontinuité dans le geste, des phases d'activation et de désactivation sont nécessaires. Pendant ces phases, la contrainte est progressivement prise en compte ou désactivée. La figure 3.8 illustre ces différentes phases. On y retrouve les quatre paramètres :



- $Act_i$  : l'instant d'activation de la contrainte,
- $dAct_i$  : la durée de la phase d'activation,
- $Desact_i$  : l'instant de désactivation de la contrainte,
- et  $dDesact_i$  : la durée de la phase de désactivation.

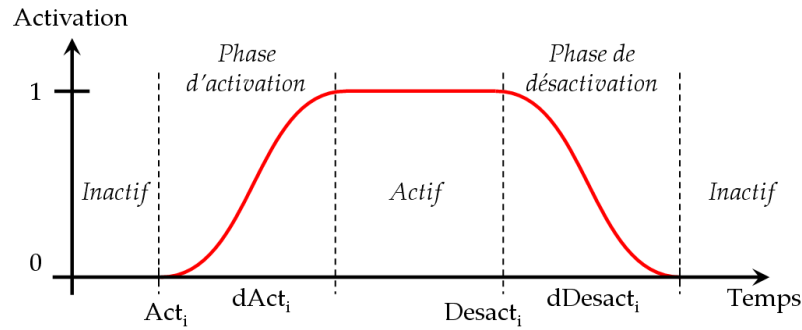


FIG. 3.8 – Coefficient d'activation des contraintes.

Lorsque l'on crée et utilise une contrainte en temps-réel, ce paramètre n'est pas obligatoire. En effet, ce type de contraintes a pour vocation d'être utilisé dans un contexte interactif. L'activation et la désactivation des contraintes sont alors spécifiées en direct par l'utilisateur ou la couche comportementale.

### 3.1.2 Paramètres spécifiques

Les paramètres spécifiques sont les objectifs associés aux types de contraintes présentés précédemment :

- **contrainte de distance.** Ce type de contrainte nécessite bien évidemment une valeur de cette distance  $PS\_Dist_i$  à respecter. Cependant, en fonction de la cible, de nouveaux paramètres sont nécessaires :
  - par rapport à une position 3D. Un vecteur 3D définissant cette position doit être spécifié. Les paramètres sont alors définis comme suit :
 
$$PS_i = \{PS\_Dist_i, PS\_Pos_i\}.$$
  - par rapport à un point du corps humain. Le segment du corps humain contenant ce point ainsi que le vecteur 3D définissant le décalage local dans ce segment sont nécessaires. Les paramètres sont alors définis comme suit :
 
$$PS_i = \{PS\_Dist_i, PS\_Seg_i, PS\_Decal_i\}.$$
  - par rapport à un autre point contraint  $CP_j$ . L'identificateur de cette contrainte  $C_j$  est suffisant. L'accès aux paramètres génériques de la contrainte permet ainsi de retrouver la position exacte du point contraint. Les paramètres sont alors définis comme suit :
 
$$PS_i = \{PS\_Dist_i, PS\_Id_i\}$$
 où  $PS\_Id_i = Id_j$  pour une contrainte  $C_j$ .
- **contrainte d'espace.** Le premier paramètre à définir est bien évidemment l'espace en trois dimensions utilisé par cette contrainte. Nous avons décidé de le représenter à l'aide d'un parallélépipède afin que sa spécification soit intuitive pour l'utilisateur. Bien évidemment, cette représentation peut être étendue selon les besoins. Cet espace est donc défini par les points opposés : le point minimum  $PS\_MinPos_i$  suivant les

trois axes et son équivalent maximum  $PS\_MaxPos_i$ . Cependant, il faut également ajouter un booléen  $PS\_Aut_i$  qui précise si cet espace est autorisé ou interdit. Les paramètres spécifiques sont alors définis comme suit :

$$PS_i = \{PS\_MinPos_i, PS\_MaxPos_i, PS\_Aut_i\}.$$

- **Contrainte d'orientation.** Le paramètre est l'orientation souhaitée pour cette contrainte :  $PS_i = \{PS\_Orient_i\}$

## 3.2 Spécification des contraintes

La représentation des contraintes est la même quelle que soit leur nature, qu'elles soient intrinsèquement liées au mouvement ou spécifiées en temps-réel par l'utilisateur indépendamment du mouvement. Cependant, leur spécification est différente. En effet, les contraintes intrinsèquement liées au mouvement doivent être définies en même temps que le mouvement auquel elles sont associées (cf. sous-section 3.2.1). Les autres contraintes sont ajoutées en temps-réel par l'utilisateur en fonction de ses besoins (cf. sous-section 3.2.2).

### 3.2.1 Spécification hors-ligne

Dans la mesure où les contraintes intrinsèquement liées au mouvement doivent être acquises en même temps que le mouvement lui-même, il faut les stocker dans un fichier de contraintes associé au fichier du mouvement. Afin de pouvoir faire évoluer ces contraintes et leur représentation en fonction des besoins, nous avons décidé d'utiliser une grammaire de fichier basée sur une représentation XML. Un exemple d'utilisation de cette grammaire est fourni dans l'annexe A.

Pour permettre à l'utilisateur de spécifier ces contraintes associées au mouvement, nous avons créé un logiciel (appelé S4DMaker) permettant de visualiser le mouvement et d'y incorporer directement les contraintes. Ce logiciel est présenté dans la section 7.1. Le fichier des contraintes est alors automatiquement associé à celui du mouvement comme le montre la figure 3.9.

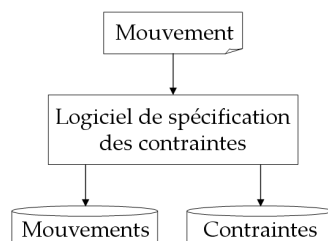


FIG. 3.9 – Entrées / sorties du logiciel de spécification des contraintes intrinsèquement liées au mouvement.

Pour utiliser ces contraintes lors de l'animation, il suffit d'utiliser l'identificateur  $Id_i$  de la contrainte  $C_i$  correspondante et de modifier les paramètres souhaités. Par exemple, pour un mouvement de préhension auquel est associée une contrainte de nom *mainPrehension*, il suffit de retrouver cette contrainte et ensuite de modifier le paramètre spécifique  $PS_i$

qui contient la position finale de la main. Ainsi, la préhension sera adaptée à la nouvelle position de l'objet.

### 3.2.2 Spécification en ligne

Les contraintes temps-réel sont ajoutées directement par l'utilisateur au cours de l'animation, par exemple pour respecter l'environnement qui évolue dynamiquement. La spécification de ces contraintes se fait donc par une interface logicielle (appelée *S4DMaker*) qui permet de créer de nouvelles contraintes et de modifier leurs paramètres.

## 3.3 Représentation interne des contraintes

La grande variété de contraintes décrites précédemment offre plus de contrôle à l'utilisateur. Cependant, elle augmente aussi la complexité de l'algorithme nécessaire pour les prendre toutes en considération. De plus, l'ajout de nouvelles contraintes nécessitent alors de changer le module d'adaptation. Nous proposons donc d'utiliser une représentation interne et unique des contraintes afin d'avoir un algorithme générique basé sur des contraintes de position.

Des opérateurs de transformation des représentations externes des contraintes vers la représentation interne sont donc nécessaires. Les paramètres spécifiques  $PS_i$  qui définissent les objectifs de la contrainte  $C_i$  sont donc convertis en une position 3D désirée appelée  $Cible_i$ . Cette cible est donc la position que la contrainte devrait avoir après adaptation.

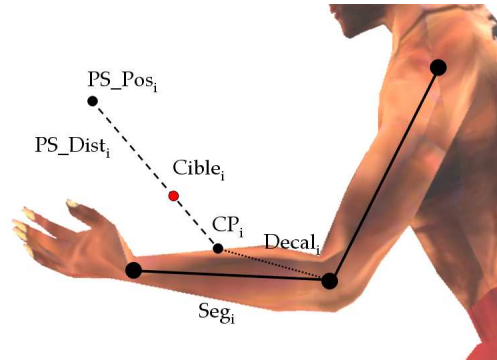


FIG. 3.10 – Conversion d'une contrainte de distance.

Le premier type de contrainte que l'utilisateur peut contrôler est la **contrainte de distance**. Sa conversion se fait en deux étapes. La première consiste à calculer la position de référence  $PosRef$ . Cette position est fournie par les paramètres spécifiques. Le calcul dépend donc de leur type :

- par rapport à une position 3D : c'est directement cette position qui est utilisée :  $PosRef = PS\_Pos_i$ .
- par rapport à un point du corps humain : la position de référence est calculée en appliquant le décalage  $PS\_Decal_i$  dans le repère local du segment  $PS\_Seg_i$ .
- par rapport à une autre contrainte  $C_j$  : le segment  $Seg_j$  et le décalage  $Decal_j$  sont retrouvés à partir du nom  $Id_j$  de cette contrainte. Le calcul est alors le même que

pour le type précédent.

À partir de cette position de référence, il est alors possible de déterminer la cible  $Cible_i$  associée à la contrainte  $C_i$ . En effet, ce point est situé sur le vecteur qui part du point contraint  $CP_i$  et qui va à la position de référence  $PosRef$ . La cible  $Cible_i$  est alors placée à une distance  $PS_{Dist_i}$  de la position de référence (cf. figure 3.10). Si cette distance est nulle, la cible est alors placée sur la position de référence  $PosRef$ .

Le second type de contrainte utilisé représente les **contraintes d'espace**. Considérons le cas d'un espace autorisé (cf. figure 3.11). Si la contrainte est dans cet espace, sa cible  $Cible_i$  est sa propre position  $CP_i$  puisqu'elle est correcte. Aussi, on part de cette position et on compare axe par axe (X, Y et Z) la valeur de la coordonnée avec celle des positions  $PS_{MinPos_i}$  et  $PS_{MaxPos_i}$ . Si la valeur n'est pas dans l'intervalle entre ces deux points, on la seuille. Ainsi, on se retrouve à projeter la position courante de la contrainte sur cet espace autorisé. La même technique est appliquée pour les espaces interdits en inversant le seuillage des coordonnées.

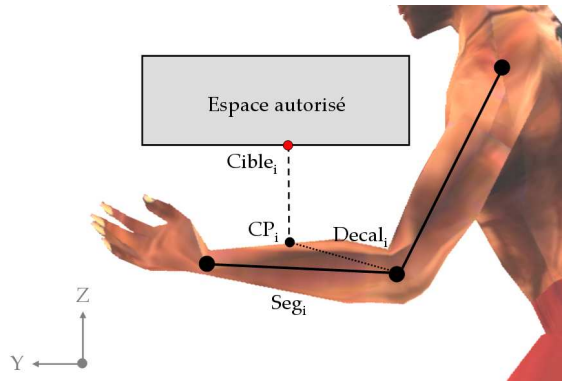


FIG. 3.11 – Conversion d'une contrainte d'espace autorisé à partir d'un point contraint  $CP_i$  hors de cette zone.

Enfin, le dernier type de contrainte contrôlé par l'utilisateur est la **contrainte d'orientation**. La cible  $Cible_i$  est placée de façon à ce que le vecteur entre la contrainte  $C_i$  et cette cible soit orienté conformément à l'orientation fournie par le paramètre  $PS_{Ori_i}$  (cf. figure 3.12).

### 3.4 Discussion

Dans ce chapitre, deux niveaux de contraintes sont proposés afin que l'algorithme d'adaptation soit indépendant du type de contraintes utilisé :

- la représentation externe qui définit les contraintes qui peuvent être contrôlées par l'utilisateur ou la couche comportementale ;
- la représentation interne des contraintes utilisée par l'algorithme d'adaptation ;

Un opérateur de conversion d'une représentation à l'autre est donc nécessaire pour chaque type de contrainte. Enfin, ces deux niveaux de représentations peuvent être utilisés avec les contraintes intrinsèquement liées au mouvement et avec celles spécifiées en temps-réel.

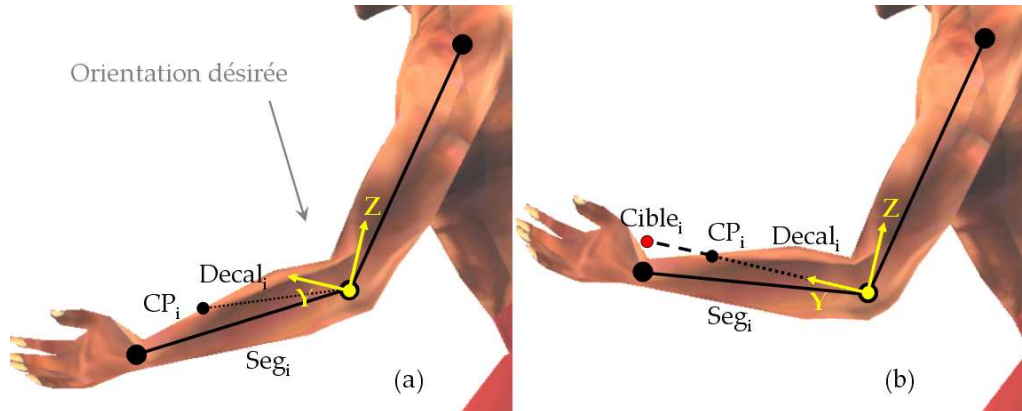


FIG. 3.12 – Conversion d’une contrainte d’orientation. (a) Posture de référence. (b) Posture adaptée à l’orientation désirée.

L’être humain est très sensible aux changements de vitesse et d’accélération. Pour qu’une animation puisse être considérée comme réaliste, il faut donc que le mouvement du personnage soit fluide. Par conséquent, l’adaptation du mouvement ne doit pas engendrer de discontinuité. Étant donné que les algorithmes d’adaptation proposés essaient de respecter au mieux les contraintes fournies, ces dernières ne peuvent pas être déplacées de manière discontinue sans engendrer d’artefacts visuels. Il est donc important que les paramètres des contraintes et en particulier leurs cibles soient modifiés de manière continue. C’est aussi la raison pour laquelle les paramètres d’activation et de désactivation des contraintes utilisent une spline afin de garantir une continuité  $C^2$ .

Il est important de distinguer la cible  $Cible_i$  de la contrainte  $C_i$  et sa position  $CP_i$  calculée à partir du segment  $Seg_i$  et du décalage local  $Decal_i$ . Respecter une contrainte signifie que l’adaptation essaie de placer la position  $CP_i$  sur la cible associée  $Cible_i$ . Il est donc aussi important d’assurer une continuité dans le déplacement de la contrainte que dans celui de la cible. En général, le point contraint  $CP_i$  n’est pas déplacé. Cependant, pour réussir des adaptations très complexes, il peut être utile de combiner la modification simultanée des deux paramètres. Par exemple, si une personne a les avant-bras appuyés sur le rebord d’une table et qu’un deuxième personnage éloigne la table, les avant-bras devraient “glisser” sur le rebord de la table. Cette adaptation peut alors être réalisée en plaçant une contrainte par avant-bras. Leur cible serait alors conservée sur le rebord de la table et le point contraint serait déplacé le long de l’avant-bras.

Les modifications des paramètres faites par l’utilisateur pourraient être filtrées de manière à automatiquement garantir une continuité des contraintes. Cela permettrait à l’utilisateur de pouvoir placer ces contraintes sans être très rigoureux. Cependant, avec cette couche de filtrage automatique, il perdrait du même coup une partie de leur contrôle. De plus, il serait très difficile de garantir que le filtre utilisé corresponde exactement aux attentes de l’utilisateur. Par conséquent, nous préférons lui laisser l’accès direct et total aux contraintes.

L'utilisation de contraintes géométriques permet implicitement de pouvoir contraindre les vitesses et les accélérations. En effet, les adaptations sont faites à chaque pas de temps aussi une contrainte a une trajectoire au cours du temps. Nous avons vu qu'il fallait que cette courbe soit au minimum continue mais il est possible d'aller plus loin en imposant une vitesse ou une accélération par exemple. Il pourrait être intéressant de créer des contraintes de plus haut niveau qui convertirait des paramètres cinématiques en contraintes géométriques afin que les vitesses et les accélérations puissent être définis explicitement.

## Chapitre 4

# Adaptation cinématique

Conformément à la chaîne d'animation présentée dans la figure 2.3, le module d'adaptation cinématique doit permettre d'adapter la posture de référence aux contraintes. Dans le chapitre précédent, les contraintes de l'utilisateur sont converties en une représentation interne. Ce module d'adaptation propose donc un **algorithme de cinématique inverse** afin d'appliquer ces contraintes internes sur la chaîne cinématique des personnages animés.

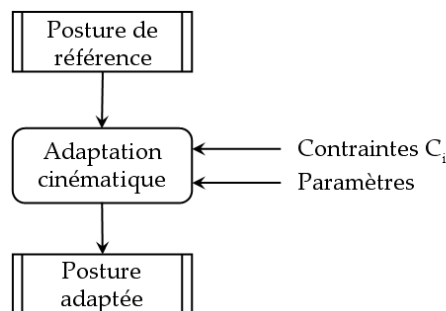


FIG. 4.1 – Entrées / sorties du module d'adaptation cinématique.

Le module d'adaptation cinématique prend donc en entrée cette posture de référence ainsi que les contraintes (cf. figure 4.1). L'utilisateur ne peut pas contraindre toutes les parties du corps humain à chaque instant, surtout en temps-réel. Par conséquent, le système est sous-contraint : il y a moins de contraintes que de degrés de liberté. Une infinité de solutions existe donc pour respecter ces contraintes. Il faut alors trouver un moyen d'identifier la solution qui est la plus appropriée. Nous avons choisi de chercher à préserver au mieux la posture de référence. En effet, elle représente la posture que devrait avoir le personnage si aucune contrainte n'était appliquée. Le but de notre module d'adaptation est donc de **respecter les contraintes fournies tout en préservant au mieux cette posture de référence**.

Une entrée supplémentaire est présentée dans la figure 4.1 : les paramètres. Ils offrent à l'utilisateur un contrôle plus précis de l'adaptation.

La section 4.1 décrit la représentation des données utilisée par l'algorithme d'adaptation. Cet algorithme est à son tour détaillé dans la section 4.2. La section 4.3 donne les résultats obtenus ainsi que des valeurs de performance de notre module d'adaptation. Enfin, une discussion sur ce travail est présentée dans la section 4.4.

## 4.1 Représentation des données

Dans cette section, nous proposons une représentation du personnage selon trois niveaux :

- la représentation du squelette qui est commune à tous les humanoïdes ;
- la représentation des caractéristiques individuelles de chacun des personnages ;
- et enfin, le découpage en groupes du squelette.

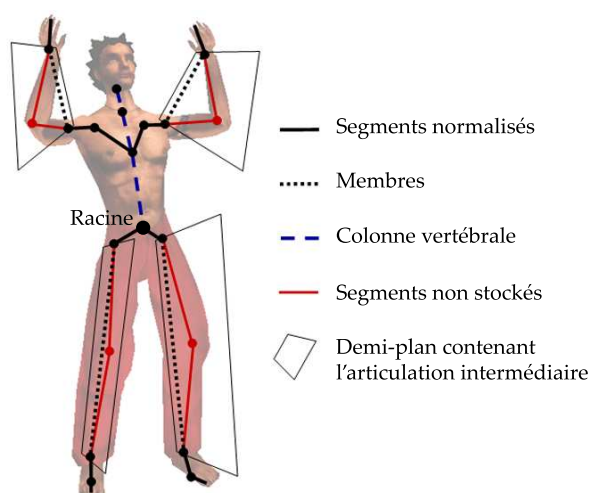


FIG. 4.2 – Squelette normalisé.

Parmi toutes les représentations du squelette présentées dans la revue de littérature (cf. section 1.1), nous avons choisi d'utiliser la **représentation normalisée** illustrée dans la figure 4.2. En effet, elle offre deux avantages majeurs :

- elle est indépendante de la morphologie. Le module d'adaptation morphologique présenté dans la figure 2.3 s'appuie sur cette représentation ;
- et surtout elle contient à la fois des données cartésiennes et angulaires. Cette fonctionnalité permet un calcul simplifié des solutions analytiques. Les membres par exemple sont modélisés par un segment de longueur variable (cf. figure 4.2) au lieu de deux segments liés par une articulation ayant un degré de liberté.

En plus de cette représentation du squelette, certaines **caractéristiques individuelles des personnages** sont définies. Ces données sont directement utilisées par les différents modules d'adaptation de notre chaîne d'animation (cf. figure 2.3) :

- **la morphologie**. Elle représente les dimensions des segments du personnage. Cette donnée est essentielle pour le module d'adaptation morphologique.



- **les butées articulaires.** Elles définissent les limites angulaires pour chaque articulation.
- **les limites musculaires.** Elles représentent la souplesse du personnage. Au-delà des butées articulaires, ces paramètres définissent la faculté à étendre les muscles. Par exemple, ce ne sont pas les limites osseuses des articulations des hanches qui empêchent de réaliser un grand écart mais bien les adducteurs. Il faut par conséquent prendre en compte ce type de paramètres afin d'être plus précis dans l'adaptation de la posture. Contrairement aux butées articulaires qui sont constamment activées, ces limites musculaires ne sont utilisées que dans certaines circonstances. Elles sont donc définies par configuration de la posture, par exemple l'écart frontal maximal autorisé lorsque les jambes sont tendues.
- **les données cinétiques.** Ces données comprennent entre autres les masses du personnage. Ces dernières peuvent être décrites segment par segment ou alors uniquement par la masse totale. Dans ce dernier cas, les masses de chaque segment sont retrouvées automatiquement en utilisant des tables anthropométriques. Les autres données cinétiques sont décrites dans la section 5.

Cette représentation est bien évidemment ouverte à l'insertion de tous les paramètres qui peuvent définir les caractéristiques individuelles des personnages. Nous avons présenté ici ceux qui sont utilisés par nos trois modules d'adaptation. Dans le cadre d'applications comprenant des acteurs virtuels autonomes, c'est-à-dire des personnages qui ont leur propre comportement, cette représentation des individualités peut être partagée avec la couche comportementale. En effet, d'autres paramètres tels que l'agressivité du personnage par exemple peuvent être nécessaires mais ces paramètres n'entrent pas dans le cadre de l'adaptation de mouvements présentée dans ce mémoire.

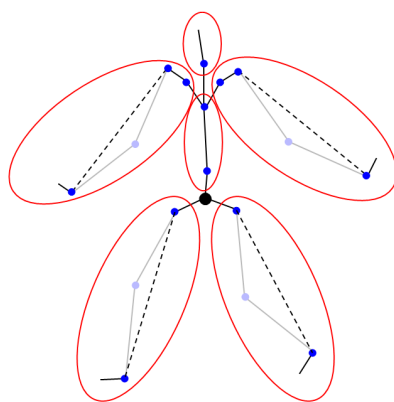


FIG. 4.3 – Groupes du corps humain utilisés pour l'adaptation.

Le squelette humain est composé de plusieurs chaînes cinématiques (cf. section 1.1). Ces chaînes partent toutes de la racine du personnage et vont jusqu'aux extrémités (cf. figure 4.2). Le problème de cette représentation est que les segments du tronc sont partagés par plusieurs chaînes cinématiques. Ainsi, lorsque deux contraintes sont utilisées, une sur chaque bras par exemple, la modification apportée par chacune des contraintes sur sa

chaîne cinématique implique les segments du tronc ce qui peut conduire à des modifications antagonistes de ces segments.

Nous proposons donc de **découper le squelette en groupes** qui définissent des chaînes cinématiques simples. Ce découpage présenté dans la figure 4.3 permet de séparer les chaînes cinématiques afin de leur appliquer une adaptation personnalisée. Il permet également de choisir l'ordre dans lequel ces adaptations sont effectuées.

En restant conforme à la hiérarchie du squelette qui part de la racine du personnage vers les extrémités, nous avons défini une hiérarchie équivalente pour les groupes. Nous sommes donc partis de la racine du personnage pour définir le premier niveau (cf. figure 4.4). Puis en suivant les liens de la hiérarchie, les jambes et le tronc forment le second niveau. Enfin, le dernier niveau est composé des bras et de la tête.

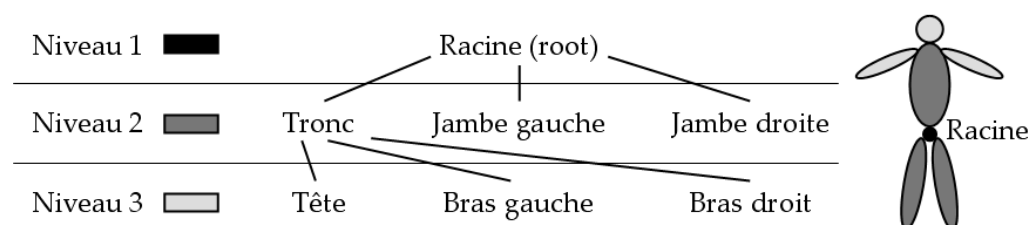


FIG. 4.4 – Hiérarchie des groupes du corps humain.

Ces trois niveaux de représentation du personnage sont utilisés dans la section suivante qui décrit l'algorithme d'adaptation cinématique.

## 4.2 Algorithme d'adaptation

Dans la section 1.3.1 de la revue de littérature, nous avons décrit les grandes familles de cinématique inverse. Chacune d'entre elles présente des avantages et des inconvénients comme le résume la section 1.3.3.

L'algorithme d'adaptation présenté dans ce chapitre doit être capable d'animer de nombreux personnages en temps-réel. Aussi, nous proposons d'utiliser un algorithme itératif de type CCD (cf. section 1.3.1.3) qui permet d'effectuer très rapidement les calculs de chacune des itérations. De plus, cet algorithme est combiné à des solutions analytiques comme le proposent Shin et coll. [Shin 01]. Cependant, contrairement à ces auteurs, nous souhaitons pouvoir :

- contraindre n'importe quel point du corps du personnage,
- utiliser complètement les chaînes cinématiques du corps humain
- et enfin fonctionner avec les deux modes de contrôle des contraintes : par mélange et par priorité (cf. section 1.3.1.2).

Dans ce contexte d'animation d'humanoïdes, l'algorithme itératif de type CCD est très peu utilisé étant donné les inconvénients majeurs qu'il présente (cf. section 1.3.1.3). Nous devons donc proposer des solutions afin de résoudre les problèmes suivants :

- l'algorithme ne prend pas en compte les règles biomécaniques d'organisation segmentaire : on ne sollicite le tronc que si les membres les plus légers et donc les moins coûteux en terme énergétique ne sont pas suffisants ;
- la répartition de l'adaptation le long des chaînes cinématiques n'est pas uniforme. La figure 4.5 montre une adaptation effectuée sur une structure cinématique simple : le serpent. La partie (a) montre la posture de référence et la partie (b) le résultat attendu. Cependant, le résultat de l'utilisation du CCD fournit la posture illustrée dans la partie (c). L'effet "zig-zag" obtenu provient de cette mauvaise répartition de l'adaptation le long de la chaîne cinématique ;
- enfin, le temps de calcul de chaque itération est très faible mais le nombre d'itérations est très important diminuant les performances globales de cet algorithme.

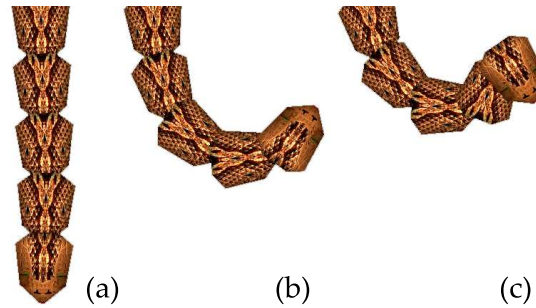


FIG. 4.5 – Mauvaise répartition de l'adaptation en utilisant le CCD. La partie (a) montre la posture de référence et la partie (b) illustre le type de résultat attendu en utilisant une contrainte sur l'extrémité de la chaîne. En fait, l'adaptation obtenue avec le CCD est illustrée par la figure (c).

Pour résoudre le problème de l'organisation segmentaire, nous proposons d'appliquer le CCD sur les groupes définis dans la section 4.1 et non sur les segments. L'utilisation de ces groupes permet en effet d'avoir deux niveaux d'ordonnancement des adaptations : l'ordre de prise en compte des groupes et l'ordre d'adaptation du contenu des groupes. Cet ordonnancement des groupes est également utilisé pour traiter le problème de la mauvaise répartition des adaptations le long de la chaîne cinématique. En effet, il permet d'appliquer les premières adaptations qui sont les plus importantes aux groupes et aux segments les plus mobiles. Enfin, le nombre d'itération va être diminué à l'aide de solutions analytiques qui permettent de placer instantanément des parties du corps dans la meilleure configuration, celle qui minimise l'erreur entre les contraintes et leurs cibles.

#### 4.2.1 Algorithme général

Comme nous venons de le voir, l'algorithme général d'adaptation cinématique est composé d'une boucle principale qui effectue un certain nombre d'itérations jusqu'à l'obtention de la posture qui répond au problème posé. A l'intérieur de cette boucle, notre algorithme va faire des adaptations locales aux groupes. Cependant, il reste à définir l'ordre de prise en compte de ces groupes.

Pour démontrer l'importance de l'ordre d'adaptation des groupes, considérons le cas d'une personne qui souhaite saisir une cible devant elle, comme le montre la figure 4.6a. Pour qu'elle puisse atteindre son objectif, elle va devoir utiliser son bras et son tronc

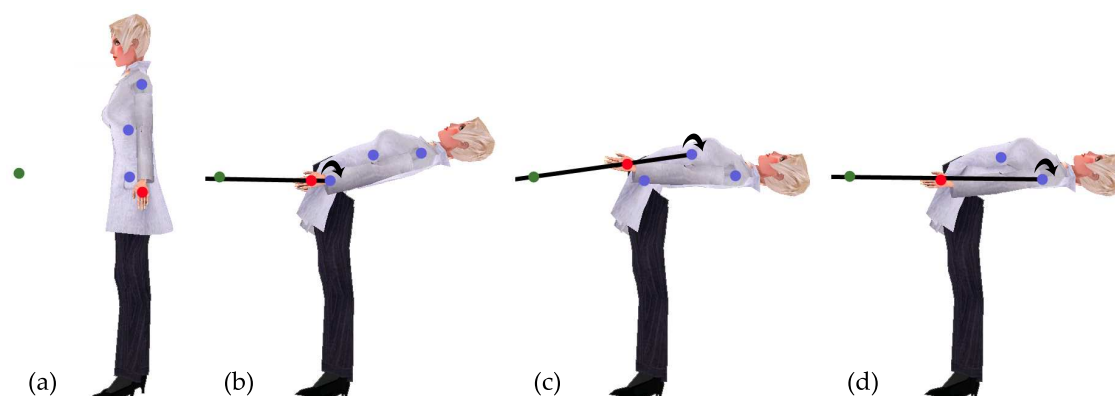


FIG. 4.6 – Groupes pris en compte de la racine aux extrémités de la chaîne cinématique. Une contrainte est appliquée sur la main avec pour but d’atteindre la cible placée devant le personnage. Le tronc est sollicité avant le bras. La partie (a) est la posture de référence. Les parties (b), (c), et (d) montrent les étapes successives pour atteindre la cible. La posture finale n’est pas réaliste.

(pour les besoins de cette démonstration, aucune butée articulaire n’est appliquée). Si on parcourt la chaîne cinématique de la racine aux extrémités, l’adaptation va traiter en premier l’abdomen qui va donc basculer en arrière afin d’aligner la racine de ce segment, la contrainte et la cible (cf. figure 4.6b). C’est ensuite au tour du torse de pencher encore plus en arrière toujours pour respecter cet alignement (cf. figure 4.6c). Enfin, le bras va être adapté à son tour (cf. figure 4.6d). On constate immédiatement que la posture n’est pas réaliste. De plus, augmenter le nombre d’itérations ne corrigera pas le problème.

A l’inverse, si on parcourt la chaîne cinématique en partant des extrémités vers la racine, on obtient le résultat fourni par la figure (cf. figure 4.7). La première image (a) est à nouveau la posture de référence. Le bras est adapté le premier (b) puis le torse (c) et enfin l’abdomen (d). Cette fois la posture est crédible et l’utilisation de nouvelles itérations (la partie (e) montre le calcul suivant du bras) permet d’affiner le respect de la contrainte sur la main.

L’ordre est donc primordial pour obtenir une posture réaliste. C’est d’ailleurs l’absence de tâches secondaires qui explique qu’il faille appliquer directement la bonne heuristique. En effet, on utilise toujours les membres les plus légers en premier puisqu’ils requièrent moins d’énergie pour être déplacés. De plus, ce sont aussi ceux qui offrent la plus grande amplitude de mouvement permettant ainsi d’atteindre plus rapidement les objectifs. L’algorithme d’adaptation cinématique fonctionne donc du niveau le plus bas (3) vers le niveau le plus haut (1) de la hiérarchie des groupes présentée dans la figure 4.4. L’ordre des groupes au sein de chacun de ces niveaux n’a cependant pas d’importance puisque ces groupes sont indépendants. L’algorithme général d’adaptation cinématique est par conséquent le suivant :

*fonction adaptationCinématique()*

*it = 0*

*Faire*

*adaptationCmGroupe(Tête)*

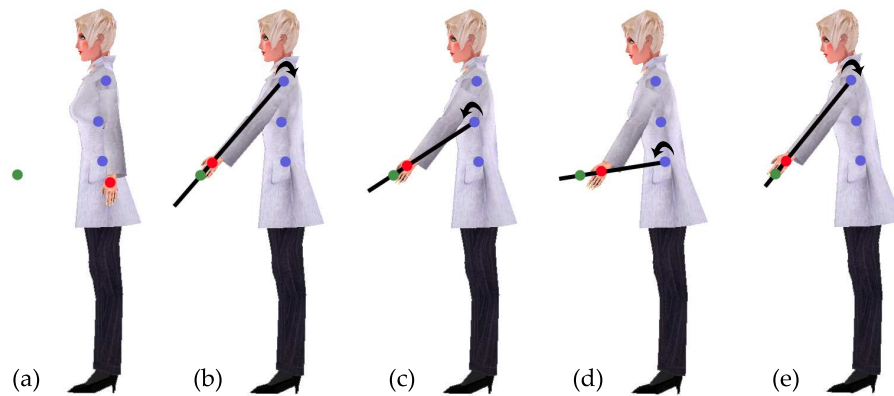


FIG. 4.7 – Groupes pris en compte des extrémités à la racine de la chaîne cinématique. Une contrainte est appliquée sur la main avec pour but d’atteindre la cible placée devant le personnage. Le tronc est sollicité avant le bras. La partie (a) est la posture de référence. Les parties (b), (c), (d) et (e) montrent les étapes successives pour atteindre la cible. La posture finale est réaliste.

```

adaptationCmGroupe(Bras droit)
adaptationCmGroupe(Bras gauche)
Si (Pied droit non en contact avec le sol)
    adaptationCmGroupe(Jambe droite)
Fin du Si
Si (Pied gauche non en contact avec le sol)
    adaptationCmGroupe(Jambe gauche)
Fin du Si
adaptationCmGroupe(Tronc)
adaptationCmRacine()
terminé = calculCmErreur()
it = it + 1

```

*Tant que ( (it < maxIt) & (¬terminé) )*  
 où *it* est l’itération courante, *maxIt* est le nombre maximum d’itérations souhaité et *terminé* est un booléen qui a pour valeur *vrai* lorsque l’adaptation est achevée.

La paramètre *maxIt* est très utile puisqu’il peut varier en fonction de la puissance de la machine sur laquelle est exécutée l’adaptation. En effet, l’un des intérêts des algorithmes itératifs est qu’ils peuvent être arrêtés à tout moment. Bien évidemment, un arrêt précoce diminue la précision de l’adaptation. Dans l’exemple 4.7, le résultat de la partie (d) est celui qui serait obtenu en n’utilisant qu’une seule itération de l’algorithme d’adaptation.

Puisque les contraintes de contact des pieds d’appui sur le sol sont prioritaires sur toutes les autres, les groupes des jambes ne peuvent être utilisés pour l’adaptation cinématique que s’ils ne servent pas au respect du support du personnage.

### 4.2.2 Adaptation de l'intérieur des groupes

Le découpage du squelette en groupes a également été réalisé afin de pouvoir contrôler l'ordre de prise en compte des segments à l'intérieur de ces groupes. Cet ordre s'appuie sur les mêmes règles biomécaniques que celles utilisées pour l'ordonnement des groupes entre eux. En effet, les segments les plus mobiles et les plus légers sont mobilisés en premier. Ainsi, pour les groupes des bras et des jambes, les extrémités sont sollicitées les premières et le parcours se termine à la racine du groupe. A l'inverse, pour le tronc, la rotation de l'abdomen est la plus mobile et offre la plus grande amplitude aussi le parcours de ce groupe est effectué de l'abdomen jusqu'au torse. Cette utilisation des segments les plus mobiles en priorité est également justifiée par le fait que le CCD modifie toujours davantage les premiers segments adaptés.

Après l'ordre de prise en compte des segments à l'intérieur d'un groupe, il reste à déterminer la manière de les adapter. Les groupes contiennent des parties du corps qui peuvent être de trois types (conformément à la structure présentée dans la figure 1.4) : les membres, la colonne vertébrale et les autres segments qui sont normalisés.

Les **membres** sont résolus à l'aide de solutions analytiques. Ces solutions permettent d'accélérer les temps de calcul puisque les deux segments qui composent le membre sont adaptés instantanément. Leur utilisation est également importante pour la crédibilité des postures obtenues. En effet, dans l'exemple fourni dans la section précédente (cf. figures 4.7), si l'adaptation du bras n'avait pas été faite en un seul calcul, il y aurait d'abord eu l'adaptation de l'avant-bras puis du bras. La figure 4.8 montre cette adaptation pour le même exemple. La partie (a) est la même posture de référence que dans la figure 4.7a. L'avant-bras, le bras puis le torse et enfin l'abdomen sont alors adaptés successivement (cf. partie (b), (c), (d) et (e) de la figure 4.7). La posture obtenue peut parfaitement être réalisée par une personne cependant elle n'est pas réaliste en terme de confort. En effet, lorsqu'une préhension est effectuée naturellement, le tronc n'est sollicité que si le bras ne suffit pas. De plus, l'être humain s'arrange pour que la rotation du tronc soit toujours minimale. Ainsi, quelle que soit la position de la cible, le tronc n'est incliné que de l'amplitude nécessaire pour que la main puisse atteindre l'objectif. Il est donc primordial que les solutions analytiques qui décrivent les membres respectent instantanément la contrainte si elle est accessible ou alors offrent la solution optimale, c'est-à-dire celle qui minimise l'erreur. Les solutions analytiques vont également être utilisées pour adapter les contraintes placées en aval des membres, c'est-à-dire sur les mains ou les pieds.

Dans la représentation normalisée, la **colonne vertébrale** est modélisée par deux splines, une pour l'abdomen et une pour le torse. Ces splines sont contrôlées par les deux points de contrôle placés aux extrémités. Ce découpage en deux parties du tronc est classique en animation et s'appuie sur les différences d'amplitudes angulaires qui ne sont pas les mêmes pour toutes les vertèbres [Chaffin 99]. Nous allons par conséquent considérer le tronc comme étant formé de deux segments rigides au regard de l'algorithme d'adaptation cinématique. Bien entendu, la représentation sous-jacente reste composée de splines ce qui va permettre d'avoir automatiquement le dos qui s'arrondit lorsque le personnage se penche par exemple.

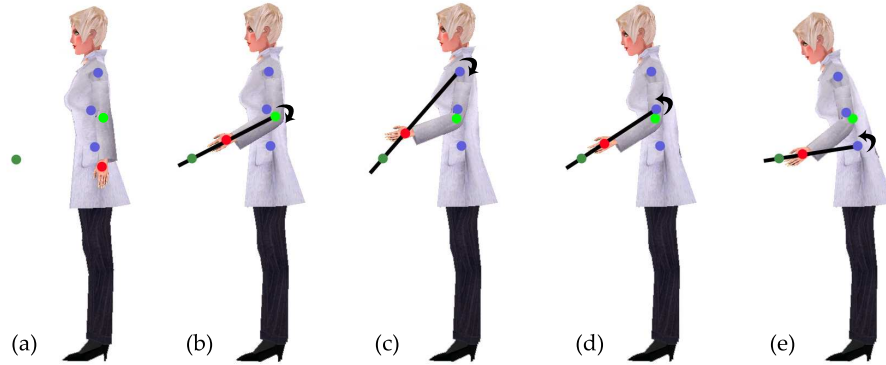


FIG. 4.8 – Adaptation des deux segments du bras indépendamment l'un de l'autre. La partie (a) est la posture de référence. Les parties (b), (c), (d) et (e) montrent les étapes successives pour atteindre la cible. Le tronc est plus penché que nécessaire et du coup le bras est plié : la posture n'est pas réaliste.

Enfin, les **segments normalisés** sont bien évidemment considérés comme des segments rigides et interviennent dans l'algorithme de la même manière que dans un CCD classique.

L'algorithme d'adaptation d'un groupe distingue donc deux cas, le traitement des membres et celui des autres parties du corps. Pour adapter les différents segments du groupe, il faut donc les parcourir et faire le calcul adéquat en fonction de l'appartenance du segment à un membre ou non :

**fonction adaptationCmGroupe( $G$ )**

```

Pour tous les segments  $S$  du groupe  $G$ 
  Si  $S$  est dans un membre
    Solution analytique : méthode adaptationCmMembre()
  Sinon
    Pour toutes les contraintes  $C_i$ 
      Si  $((C_i$  est active) Et  $(S \in Portee_i)$  Et  $(\varepsilon_i < Prec_i))$ 
        Rotation du segment : méthode adaptationCmSegment()
      Fin du si
    Fin du pour
  Fin du si
Fin du pour

```

où  $\varepsilon_i$  est la distance entre la cible  $Cible_i$  et la position de la contrainte  $CP_i$ . Cette erreur est utilisée afin d'adapter uniquement les contraintes qui ne sont pas déjà respectées ou bien qui demandent une précision d'adaptation plus grande.

Cet algorithme d'adaptation des groupes s'appuie donc sur deux méthodes : *adaptationCmMembre()* et *adaptationCmSegment()*. Ces méthodes sont dépendantes du mode de contrôle des contraintes utilisés : le contrôle par priorité ou mélange des contraintes (cf. section 3.1.1.4). Afin que notre algorithme fonctionne avec les différents modes de contrôle des contraintes, chacune des deux méthodes d'adaptation existent pour chacun de ces

modes. La section suivante 4.2.2.1 décrit l'adaptation des contraintes pour le mode de priorité des contraintes tandis que la section 4.2.2.2 détaille celle correspondant au mélange des contraintes.

#### 4.2.2.1 Contrôle par priorité des contraintes

Le contrôle par priorité des contraintes respecte les contraintes de plus forte priorité et s'approche au mieux des suivantes. Les contraintes sont donc triées dans l'ordre décroissant de leur priorité. Cette sous-section détaille dans un premier temps l'adaptation d'un segment normalisé et ensuite d'un membre conformément à ce mode de contrôle des contraintes.

Les **segments normalisés** sont adaptés de la même manière que les segments rigides dans la version originale du CCD . En effet, on applique des rotations pour aligner la racine du segment  $Seg_i$  avec la position de la contrainte  $CP_i$  et de sa cible  $Cible_i$ . L'articulation proximale du segment  $Seg_i$ , c'est-à-dire la racine de ce segment est appelée  $A_i$ . Ainsi, l'adaptation des segments normalisés consiste à aligner les vecteurs  $\overrightarrow{(A_i, CP_i)}$  et  $\overrightarrow{(A_i, Cible_i)}$  (cf. figure 4.9b). L'utilisation des butées articulaires est effectuée à ce moment de l'adaptation. En effet, la rotation calculée est ainsi seuillée par ces butées articulaires qui proviennent de la représentation des caractéristiques individuelles de chaque personnage (cf. section 4.1).

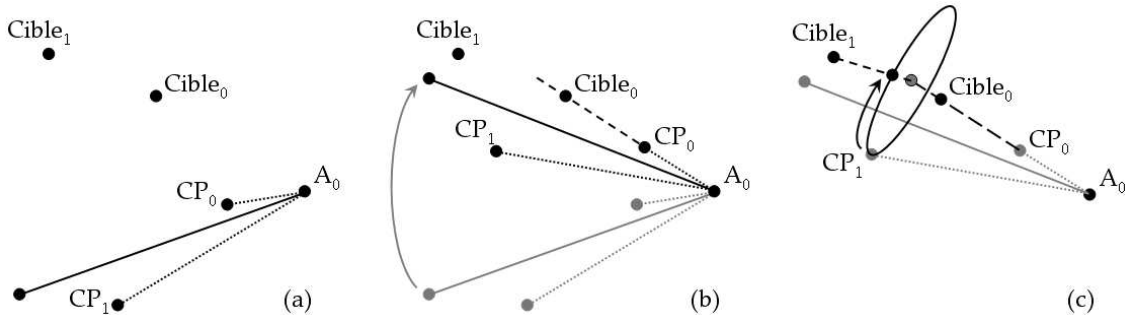


FIG. 4.9 – Adaptation d'un segment normalisé en mode priorité de contraintes. La partie (a) montre la posture de référence. La partie (b) illustre l'adaptation de la posture pour respecter la première contrainte  $C_0$ . La seconde contrainte  $C_1$  est respectée dans la partie (c) sans pour autant modifier la première.

Dans le cadre d'un contrôle des contraintes par priorité, cet alignement n'est fait que pour la première contrainte, celle de plus forte priorité :  $C_0$ . Toutes les modifications suivantes ne doivent plus changer cette première adaptation. On applique par conséquent une rotation autour de l'axe  $\overrightarrow{(A_0, CP_0)}$  de façon à ne pas modifier la position  $CP_0$ . La solution choisie est alors le point du cercle de solution présentée dans la figure 4.9c qui place la contrainte  $CP_1$  le plus près possible de la cible  $Cible_1$ . Le segment normalisé est alors complètement contraint et toutes les autres contraintes sont par conséquent ignorées.

Les **membres** sont adaptés en utilisant des solutions analytiques. Ces solutions doivent être définies en suivant cette règle : si la contrainte est accessible, la solution doit instanta-



nément respecter cette contrainte. Autrement, si la contrainte ne l'est pas, la solution doit placer le membre dans la configuration optimale, c'est-à-dire celle qui minimise l'erreur entre la position de la contrainte et celle de sa cible. Cette règle est essentielle afin que l'algorithme d'adaptation converge (cf. section 4.2.5).

En plus de minimiser l'erreur, les solutions analytiques sont définies de manière à prendre en compte la représentation des contraintes présentée dans le chapitre 3. Elles doivent par conséquent être capables d'adapter n'importe quel point du corps humain. Pour cela, elles sont créées pour tenir compte des paramètres  $Seg_i$  et  $Decal_i$  qui définissent la position de la contrainte  $CP_i$ .

Des exemples de ces solutions analytiques sont fournis en annexe B.

#### 4.2.2.2 Contrôle par mélange des contraintes

Lorsqu'une somme pondérée de toutes les contraintes est effectuée, le résultat est une somme pondérée des adaptations de chacune des contraintes prises séparément.

Ainsi, pour adapter les **segments normalisés**, l'algorithme d'adaptation calcule en premier la rotation  $Rot_i$  correspondant aux adaptations de chacune des contraintes  $C_i$  de façon isolée. Ensuite, la somme pondérée de ces rotations est calculée afin d'obtenir la rotation finale  $Rot$  à appliquer au segment :

$$Rot = \frac{\sum Imp_i * Rot_i}{\sum Imp_i}$$

Les rotations  $Rot_i$  sont toutes modélisées par un couple (axe de rotation, angle). Chaque rotation est limitée par les butées articulaires et son amplitude ne peut pas dépasser  $2\pi$ . La combinaison linéaire des axes de rotation et des angles fournit bien un axe et un angle moyen de rotation.

La rotation finale est seuillée par les valeurs des butées articulaires présentes dans la représentation des caractéristiques individuelles.

L'adaptation des **membres** se fait une fois de plus à l'aide de solutions analytiques. Elles respectent les mêmes critères que ceux définis précédemment pour le mode de contrôle par priorité de contraintes 4.2.2.1, à savoir de toujours minimiser l'erreur entre la contrainte et sa cible. Des exemples de ces solutions analytiques sont fournis en annexe B.

#### 4.2.3 Adaptation de la racine

L'adaptation des groupes ne suffit pas. En effet, la position de la racine de la hiérarchie est importante puisque tous les groupes en dépendent. Prenons l'exemple de la posture illustrée dans la figure 4.10a. Dans cette posture de référence (celle que l'on souhaite conserver au mieux), la jambe de support est fléchie. Une contrainte est placée sur la cheville et la cible correspondante est symbolisée par la sphère. Si on se contente de faire une adaptation des groupes sans déplacer le bassin et par conséquent la racine de la hiérarchie, la posture obtenue ne permet pas de vérifier la contrainte (cf. figure 4.10b). Il faut par conséquent pouvoir contrôler le déplacement de la racine afin d'avoir une adaptation la plus générique possible (méthode *adaptationCmRacine()*). Nous allons donc introduire un nouveau paramètre *CmRacine* qui représente le compromis entre le respect des contraintes et celui de la posture de référence.

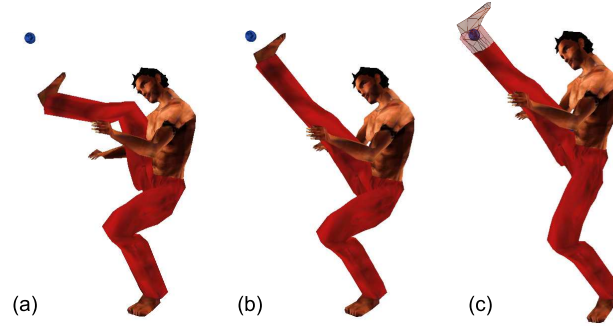


FIG. 4.10 – Déplacement de la racine pour résoudre une contrainte placée sur la cheville droite. La cible est représentée par la sphère. La partie (a) montre la posture de référence. Dans la partie (b), l’adaptation cinématique a été réalisée sans le déplacement du bassin. La contrainte n’est alors pas résolue. Enfin, dans la dernière partie (c), la translation de la racine est utilisée et la jambe de support est étirée de façon à ce que la contrainte soit vérifiée.

Cette translation de la racine a donc pour but de se rapprocher au mieux des contraintes qui ne sont pas encore respectées. Dans notre exemple du coup de pied, la racine est déplacée en direction de la cible non atteinte par la contrainte placée sur la cheville. La figure 4.10c montre la posture obtenue en tenant compte de la translation de la racine. Pour que ce paramètre puisse être facilement utilisé, il faut qu’il se base automatiquement sur l’erreur associée à une contrainte. Cette erreur  $\vec{\varepsilon}_i$  est simplement le vecteur qui va de la position  $CP_i$  de la contrainte  $C_i$  à sa cible  $Cible_i$  :

$$\vec{\varepsilon}_i = \overrightarrow{(CP_i, Cible_i)}$$

Il est important de remarquer que notre erreur est ici un vecteur. Classiquement, l’erreur est calculée en utilisant une distance ce qui équivaut à la norme de notre vecteur  $\vec{\varepsilon}_i$ . L’utilisation d’un vecteur comme erreur est essentielle puisque ce vecteur contient à la fois la distance entre la contrainte et sa cible mais également la direction vers laquelle déplacer la racine.

Lorsque plusieurs contraintes sont actives, il suffit alors de faire la somme des erreurs de chacune de ces contraintes pondérée par sa priorité :

$$\vec{\varepsilon} = \frac{\sum_i Imp_i * \vec{\varepsilon}_i}{\sum_i Imp_i}$$

Le paramètre  $CmRacine$  définit le pourcentage du vecteur d’erreur utilisé par la translation de la racine. La figure 4.11 illustre l’utilisation de ce paramètre avec le même exemple que celui de la figure 4.10 . Le pourcentage varie de 0% pour le personnage au fond de l’image à 100% pour celui au premier plan. Ainsi, le personnage au devant de l’image respecte parfaitement la contrainte placée sur sa cheville alors que le personnage au fond n’atteint pas du tout la cible.

Lorsqu’une ou plusieurs contraintes sont appliquées sur le bassin, ce déplacement de la racine n’est plus le seul paramètre dont il faut tenir compte. En effet, lorsqu’un personnage s’assoit sur une chaise, une contrainte de contact des fesses avec le siège est appliquée. Comme elle est attachée au bassin, elle ne peut être respectée que par la translation de

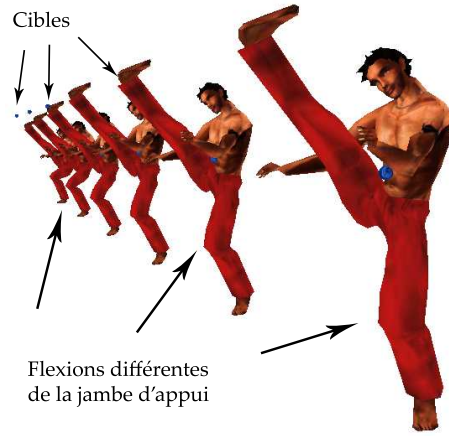


FIG. 4.11 – Utilisation d'un pourcentage pour déplacer la racine de la hiérarchie. Le paramètre  $CmRacine$  varie de 0% pour le personnage au fond de l'image à 100% pour celui au premier plan. La contrainte de ce dernier est parfaitement vérifiée alors que le personnage au fond est très loin de la cible.

la racine. Si le déplacement de la racine est uniquement contrôlé par l'erreur des autres contraintes, jamais cette contrainte de contact entre les fesses et le siège ne peut être respectée. Il faut donc partager l'adaptation de la racine entre l'ensemble  $E_b$  des contraintes qui sont appliquées sur le bassin d'un côté et l'ensemble  $E_g$  de toutes les autres contraintes appliquées sur les groupes du corps. Le déplacement de la racine est alors obtenu par la somme pondérée de ces deux ensembles de contraintes :

$$\vec{\varepsilon} = \frac{Imp_{E_b} * \vec{\varepsilon}_{E_b} + Imp_{E_g} * \vec{\varepsilon}_{E_g}}{Imp_{E_b} + Imp_{E_g}} \quad (4.1)$$

où  $Imp_{E_b}$  (resp.  $Imp_{E_g}$ ) est la somme des priorités des contraintes de l'ensemble  $E_b$  (resp.  $E_g$ ).

D'autres règles peuvent également être intégrées dans l'adaptation de la racine de la hiérarchie. Ces règles peuvent être issues des études sur le contrôle moteur ou bien découler des caractéristiques personnelles de chaque individu.

A titre d'exemple, nous avons choisi et intégré une règle qui améliore le réalisme des adaptations et qui permet surtout de montrer l'utilisation de la représentation des caractéristiques individuelles (cf. section 4.1). Cette règle montre notamment que l'utilisation des butées articulaires n'est pas suffisante pour caractériser une posture et qu'il faut utiliser les limites musculaires. En effet, pour ramasser un objet à terre, certaines personnes ont besoin de plier les genoux tandis que d'autres se penchent simplement en avant. Or, cette différence n'est pas due aux butées articulaires. Même les personnes qui ne peuvent pas toucher le sol sans plier les genoux peuvent approcher les cuisses du buste en étant accroupies par exemple. Les responsables de ce phénomène sont en partie les muscles ischio-jambiers qui partent du haut du mollet jusqu'à la fesse (cf. figure 4.12a). Lorsqu'une personne se penche en avant, ce muscle atteint sa tension maximale due au basculement du bassin (cf. figure 4.12b). En pliant le genou, la longueur du muscle diminue (cf. figure 4.12c).

Nous proposons d'intégrer une règle qui tient compte de ces muscles ischio-jambiers.

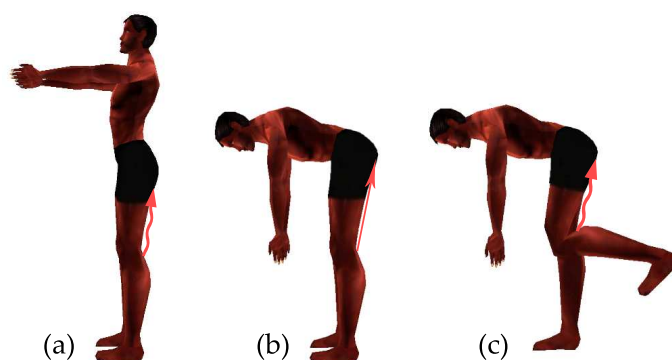


FIG. 4.12 – A moins d’être souple, les ischio-jambiers empêchent de se pencher lorsque la jambe est tendue. (a) Lorsque le personnage se tient droit, les ischio-jambiers (représentés ici schématiquement) sont détendus. (b) Lorsqu’une personne se penche, les ischio-jambiers sont tendus. Une douleur se ressent derrière les genoux. (c) En pliant la jambe, plus de tension, ce n’est donc pas un problème de butée articulaire.

Elle permet ainsi de contraindre le mouvement d’un personnage qui se penche en avant en fonction de ses limites musculaires définies dans sa représentation des caractéristiques individuelles.

Cette règle ne s’applique que lorsqu’au moins une jambe est tendue dans la posture de référence. Ensuite, il est seulement nécessaire de baisser le bassin et par conséquent de descendre la racine de la hiérarchie lorsque l’angle entre l’abdomen et la cuisse de cette jambe tendue est inférieur à un certain seuil donné par le paramètre de souplesse du personnage animé. Il est ainsi possible d’animer plusieurs personnages ayant la même morphologie et le même geste mais avec des caractéristiques de souplesse différentes comme l’illustre la figure 4.13.

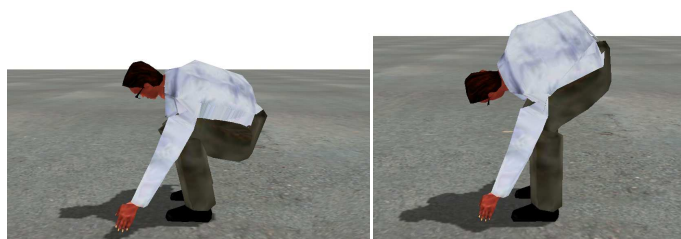


FIG. 4.13 – La souplesse d’un personnage est une donnée de la représentation des caractéristiques individuelles présentées à la section 4.1. Elle permet de différencier les mouvements des personnages en fonction de leurs aptitudes. Cette figure présente deux personnalisations de la souplesse.

Grâce à l’adaptation des groupes et de la racine qui vient d’être décrite, nous sommes à présent en mesure d’adapter la posture aux contraintes fournies par l’utilisateur. Cependant, les contraintes de contact des pieds sur le sol n’ont pas été traitées. En effet, elles sont séparées des autres contraintes parce qu’elles doivent impérativement être respectée indépendamment des autres contraintes. La section suivante décrit la gestion de l’adaptation au sol qui va être intégrée dans l’algorithme d’adaptation cinématique.

#### 4.2.4 Adaptation au sol

L'adaptation au sol a pour but de modifier la posture du personnage de manière à placer les pieds dans des empreintes calculées automatiquement en tenant compte :

- des empreintes présentes dans la posture de référence. Ce sont les empreintes issues du mélange des mouvements de référence. Par conséquent, elles sont uniquement dépendantes des environnements d'origine ;
- de la phase de support du personnage : bipodal, monopodal ou aucun appui ;
- de la configuration du sol pour le pas de temps courant.

Ainsi, dans le cas d'un appui bipodal, l'objectif de l'adaptation au sol est de passer de la posture de référence présentée dans la figure 4.14a à celle qui respecte les contacts des pieds sur le sol (cf. figure 4.14b).

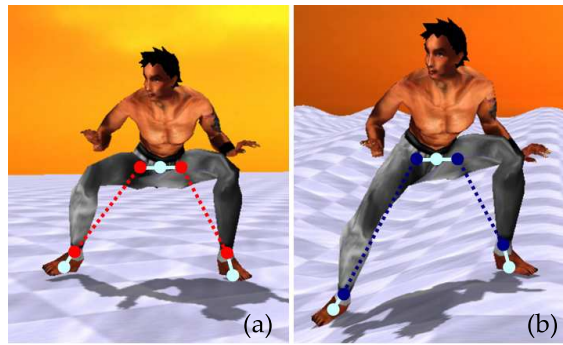


FIG. 4.14 – Exemple d'adaptation d'une posture au sol. (a) est la posture de référence, (b) est celle adaptée au nouvel environnement.

La posture de référence doit correspondre exactement à la posture qui serait utilisée pour l'animation du personnage si aucune contrainte n'était appliquée. En effet, c'est cette posture que l'on souhaite préserver au mieux. Il faut donc que cette posture de référence soit adaptée au sol par une méthode, appelée "Adaptation complète au sol", qui est appliquée avant d'utiliser la boucle itérative (cf. figure 4.15).

Étant donné que le positionnement du personnage dépend directement de la racine de la hiérarchie, son adaptation peut rompre les contacts des pieds avec le sol. Ce n'est pas le cas des adaptations de groupes puisque nous interdisons par l'intermédiaire de l'algorithme d'adaptation cinématique de contraindre une jambe en appui (qui sert donc pour le support). Une nouvelle adaptation au sol n'est donc nécessaire qu'après celle de la racine (cf. figure 4.15). Cependant, cette adaptation peut s'appuyer sur celle qui a déjà été réalisée sur la posture de référence et contrôler cette fois uniquement la hauteur de la racine pour qu'elle permette toujours de respecter les contraintes d'appui.

Ces deux adaptations sont détaillées dans les sous-sections suivantes 4.2.4.1 et 4.2.4.2.

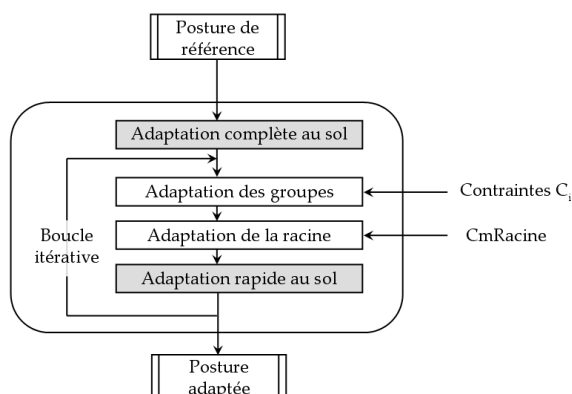


FIG. 4.15 – Algorithme général incluant l’adaptation au sol.

#### 4.2.4.1 Adaptation rapide au sol

Stéphane Ménardais [Ménardais 03] a proposé une adaptation au sol qui est rapide et adaptée à la représentation normalisée du squelette. Cette adaptation répond à deux critères :

- la position de la racine doit être cohérente avec les contraintes d’appui des pieds sur le sol. Elle doit tenir compte de la morphologie du personnage et en particulier de la longueur des jambes ;
- les jambes doivent être adaptées pour respecter cette nouvelle position de la racine.

Elle est effectuée en plusieurs étapes successives en partant de la posture de référence (cf. figure 4.16a). Les trois premières permettent de trouver la hauteur de la racine qui est compatible avec les empreintes conformément à la morphologie du personnage et les suivantes permettent d’adapter les jambes à cette nouvelle position de la racine :

1. tout d’abord, les chevilles sont calculées à partir des empreintes à respecter (cf. figure 4.16b) ;
2. ensuite, la largeur du bassin est retranchée de toutes les positions afin d’exprimer les articulations relativement à la racine (cf. figure 4.16c) ;
3. la nouvelle hauteur de la racine (sur l’axe vertical passant par la position de référence de la racine) est calculée afin de tenir compte des deux hauteurs maximales qu’elle peut atteindre lorsque les jambes sont tendues et placées sur les chevilles à respecter, ainsi que de la hauteur initiale qu’il faut préserver si possible. La bonne hauteur de la racine est par conséquent le minimum de ces trois hauteurs (cf. figure 4.16d) ;
4. la largeur du bassin est alors ajoutée pour retrouver les positions finales des chevilles (cf. figure 4.16e) ;
5. l’adaptation des jambes est alors automatique grâce à la représentation normalisée (cf. figure 4.16f).

L’inconvénient de cette adaptation au sol est qu’elle est “passive”. En effet, dans la réalité, l’être humain anticipe les variations du sol et se déplace en conséquence. Nous

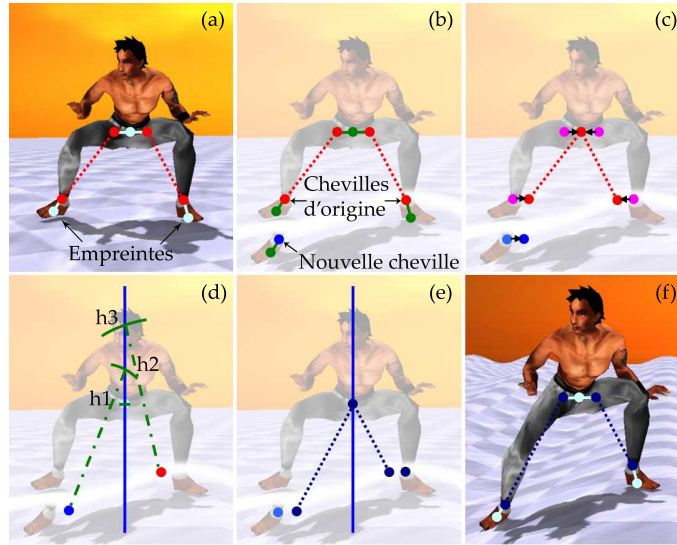


FIG. 4.16 – Adaptation rapide au sol. Plusieurs étapes : (a) la posture de référence. (b) les positions initiales et désirées des chevilles sont déterminées. (c) La largeur du bassin est supprimée de toutes les positions afin d'exprimer les articulations relativement à la racine. (d) Trois hauteurs sont calculées : celle qui préserve la racine d'origine ( $h1$ ) et les deux hauteurs maximales que la racine peut atteindre si les jambes sont tendues. (e) La hauteur minimale est choisie afin de respecter les trois critères. (e) La largeur du bassin est ajoutée pour retrouver les positions finales des chevilles. (f) L'adaptation des jambes est alors automatique avec la représentation normalisée.

proposons donc une sur-couche à cette adaptation avec un contrôle de la position de la racine.

#### 4.2.4.2 Adaptation complète au sol

Pour prendre en compte cette anticipation, nous utilisons les variations de positions des empreintes afin de modifier la position souhaitée de la racine du personnage. Comme l'adaptation rapide au sol préserve au mieux la racine de la posture de référence, cette modification de la position de la racine est donc effectuée préalablement (cf. figure 4.17). L'intérêt de laisser ces deux adaptations séparées est que le même module adaptation rapide au sol est ensuite utilisé dans la boucle itérative (cf. figure 4.15).

L'anticipation se base sur les variations des positions des empreintes. En effet, lorsque par exemple une personne en train de marcher doit poser son pied davantage sur le côté, elle anticipe en se décalant également dans cette direction. La racine de la hiérarchie du personnage doit donc être déplacée en accord avec les variations des positions des empreintes :  $\overrightarrow{\Delta Emp_D}$  pour le pied droit et  $\overrightarrow{\Delta Emp_G}$  pour le gauche. Ainsi, le déplacement  $\overrightarrow{\Delta Racine}$  est donc la moyenne des variations des positions des empreintes :

$$\overrightarrow{\Delta Racine} = \frac{\overrightarrow{\Delta Emp_D} + \overrightarrow{\Delta Emp_G}}{2}$$

La figure 4.18 illustre l'utilisation de cette anticipation. En effet, le mouvement de marche sur sol plat (a) est adapté de façon à tenir compte d'une marche d'escalier (b). La

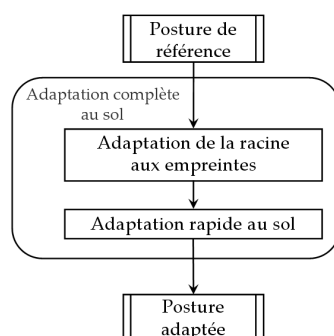


FIG. 4.17 – Schéma général du module d'adaptation complète au sol.

trajectoire de la racine en haut est adaptée dès le changement de trajectoire du pied ce qui rend l'animation plus fluide et surtout plus réaliste puisque le personnage ne tend pas sa jambe trop tôt.

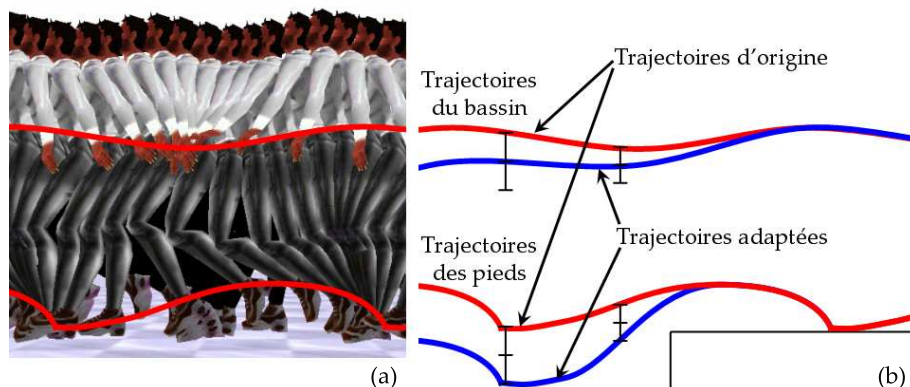


FIG. 4.18 – Schéma général du module d'adaptation complète au sol.

Le *module d'adaptation de la racine aux empreintes* permet donc de proposer une nouvelle position pour la racine qui doit être préservée si possible par le *module d'adaptation rapide au sol*. En se basant sur l'algorithme présenté à la section 4.2.1, l'adaptation cinématique finale est donc réalisée de la manière suivante :

```

fonction adaptationCinématique()
  adaptationCompleteSol()
  it = 0
  Faire
    adaptationCmGroupe(Tête)
    adaptationCmGroupe(Bras droit)
    adaptationCmGroupe(Bras gauche)
    adaptationCmGroupe(Jambe droite)
    adaptationCmGroupe(Jambe gauche)
  
```



```

    adaptationCmGroupe(Tronc)
    adaptationCmRacine()
    adaptationRapideSol()
    terminé = calculCmErreur()
    it = it + 1
    Tant que ( it < maxIt ) & ( ¬terminé )
    
```

La section suivante décrit la manière dont le critère d'arrêt *terminé* est calculé et discute de la convergence de l'algorithme

#### 4.2.5 Critère d'arrêt et convergence

Dans la mesure où les contraintes n'atteignent pas nécessairement leur cible, il est impossible d'utiliser un critère de distance minimale à respecter entre les cibles et les contraintes pour considérer que la posture est adaptée. En effet, la posture présentée dans la figure 4.19 est bien adaptée (conformément au mode de contrôle par mélange des contraintes) aux deux contraintes antagonistes de même priorité placées sur ces poignets et dont les cibles sont représentées par les sphères. Pourtant, la distance entre les contraintes et les cibles n'est pas nulle. Il faut donc prendre en compte les variations d'erreurs afin de déterminer à partir de quel instant la posture est stable et est donc adaptée. Cela implique que l'algorithme converge sans osciller. Cette question de la convergence sera abordée juste après le calcul du critère d'arrêt.

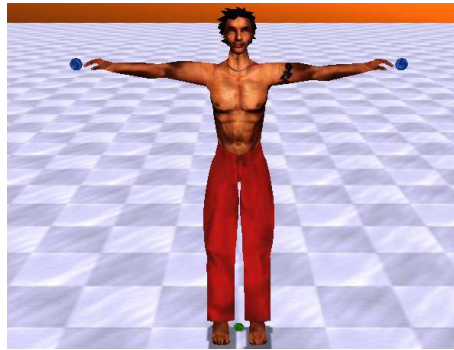


FIG. 4.19 – Respect de deux contraintes antagonistes.

En conservant les notations utilisées pour le module d'adaptation de la racine 4.2.3, l'erreur utilisée pour le critère d'arrêt est :

$$\vec{\varepsilon}_i = \overrightarrow{(CP_i, Cible_i)}$$

Afin de déterminer l'instant à partir duquel l'adaptation est terminée, l'erreur globale  $CmErreur_k$  de la posture pour l'itération  $k$  est calculée en utilisant les erreurs  $\vec{\varepsilon}_i$  de chaque contrainte  $C_i$  :

$$CmErreur_k = \frac{\sum_i Imp_i * \|\vec{\varepsilon}_i\|}{\sum_i Imp_i} \quad (4.2)$$

Il suffit donc de prendre en compte l'erreur à l'itération  $k$  courante et celle pour l'itération précédente  $k - 1$  afin d'obtenir la variation de l'erreur globale  $\Delta CmErreur_k$  entre deux itérations :

$$\Delta CmErreur_k = CmErreur_k - CmErreur_{k-1} \quad (4.3)$$

L'erreur globale initiale  $CmErreur_0$  est l'erreur entre les positions des contraintes de la posture de référence et celles de leurs cibles. Elle est calculée juste avant de commencer la boucle itérative. L'adaptation cinématique est alors terminée lorsque cette variation de l'erreur globale passe sous un seuil appelé  $CmSeuil$ .

Bien évidemment, l'algorithme doit converger pour pouvoir utiliser la variation d'erreur comme critère d'arrêt. Cette convergence existe si chaque modification apportée par notre algorithme minimise l'erreur globale de la posture. En effet, dans ce cas, la fonction d'erreur est décroissante et comme l'erreur calculée est une distance, cette fonction est toujours positive. Par conséquent, l'adaptation cinématique converge toujours. Il suffit donc que les modifications apportées par l'algorithme d'adaptation lors d'une itération minimisent l'erreur globale. Chaque erreur associée à une contrainte est déjà minimisée. En effet, les solutions analytiques minimisent par définition cette erreur et les autres parties utilisent des rotations afin d'aligner la contrainte avec la cible, c'est-à-dire de minimiser la distance entre les deux. Enfin, la racine utilise les vecteurs d'erreur afin de rapprocher les contraintes des cibles. L'erreur  $\vec{\varepsilon}_i$  associée à une contrainte  $C_i$  est donc toujours minimale.

Ainsi, lorsqu'une seule contrainte est présente, son erreur  $\vec{\varepsilon}_i$  est l'erreur globale (cf. équation 4.2). Elle est donc bien minimisée. Lorsque plusieurs contraintes sont présentes, le risque vient alors de deux contraintes antagonistes. Prenons l'exemple d'un segment dont l'extrémité est soumise à deux contraintes  $C_0$  et  $C_1$  de priorité  $Imp_0 = 2$  et  $Imp_1 = 1$ . Considérons que, dans la posture de référence, la contrainte est à équidistance des deux cibles comme l'illustre la figure 4.20a. A la première itération, la rotation  $Rot_0$  utilisée pour l'adaptation est la somme pondérée des deux rotations initiales  $\alpha_0$  et  $\beta_0$  :

$$Rot_0 = \frac{2\alpha_0 + \beta_0}{3}$$

Or,  $\alpha_0 = -\beta_0$  puisque les deux contraintes sont placées de manière symétrique, la rotation appliquée est donc :

$$Rot_0 = \frac{\alpha_0}{3}$$

A l'itération suivante, la configuration est celle présentée dans la figure 4.20b. La rotation  $Rot_1$  utilisée pour cette itération est donc :

$$Rot_1 = \frac{2\alpha_1 + \beta_1}{3}$$

Or, les nouvelles valeurs des angles sont :

$$\begin{aligned} \alpha_1 &= \alpha_0 - Rot_0 \\ &= \alpha_0 - \frac{\alpha_0}{3} = \frac{2}{3}\alpha_0 \\ \beta_1 &= \beta_0 - Rot_0 \\ &= -\alpha_0 - \frac{\alpha_0}{3} = -\frac{4}{3}\alpha_0 \end{aligned}$$

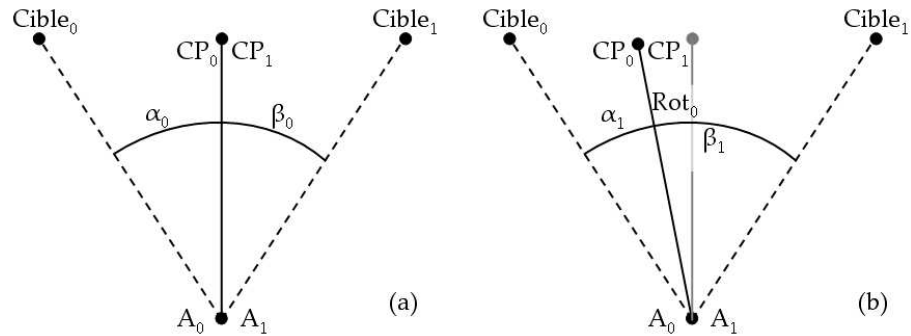


FIG. 4.20 – Calcul de la convergence pour deux contraintes antagonistes.

La valeur de  $Rot_1$  est donc nulle. En utilisant des sommes pondérées, même en utilisant des contraintes antagonistes, l'algorithme converge.

Cette convergence peut aussi être expliquée en s'appuyant sur celle du CCD. En effet, lorsque seuls les membres sont nécessaires à une adaptation, la solution analytique trouve directement la solution. Sinon, c'est que ce membre est tendu en direction de la cible et est donc assimilable à un segment rigide. Dans ce cas, on se retrouve dans un contexte d'adaptation de type CCD dont la convergence est démontrée.

### 4.3 Résultats

Cette section présente les résultats de l'utilisation de notre algorithme d'adaptation cinématique en terme de performance. Le critère de réalisme est traité à la fin du chapitre 5 puisque, dans la réalité, il n'est pas possible de différencier l'adaptation cinématique et cinétique puisque le personnage respecte forcément l'équilibre dans des postures statiques.

Des applications plus complexes sont décrits dans la section 7.2.



FIG. 4.21 – Adaptation à des contraintes placées sur la peau des avant-bras. La personnage a pour consigne de placer ces avant-bras sur le rebord de la table.

Notre algorithme sait respecter des contraintes inaccessibles en s'en approchant le plus possible. La figure 4.22 montre l'adaptation de la posture à deux contraintes placées sur

le poignet et le coude. La posture de référence est montrée dans la partie (a). La partie (b) montre une adaptation pour laquelle les deux contraintes sont accessibles. La partie (c) illustre des contraintes qui ne peuvent être atteintes.

Enfin, il est possible de contraindre n'importe quel point du corps humain en utilisant notre algorithme d'adaptation. La figure 4.21 montre l'exemple d'un personnage qui appuie ses avant-bras sur le rebord d'une table.

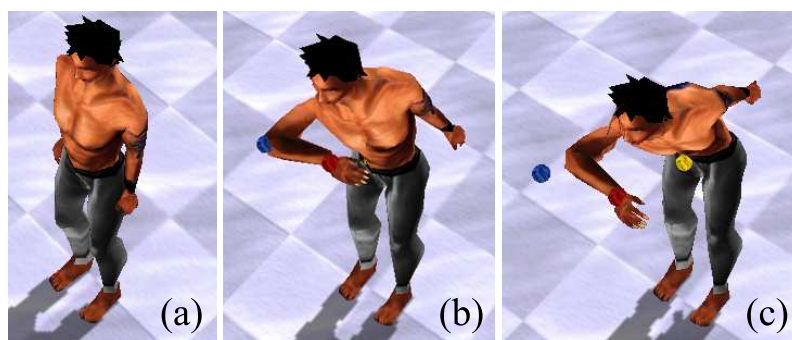


FIG. 4.22 – L'algorithme d'adaptation cinématique sait gérer des contraintes qui ne sont pas accessibles. La posture de référence est montrée dans la partie (a). Les parties (b) et (c) montrent les adaptations lorsque les contraintes sont ou non accessibles.

Afin de juger des **performances de notre algorithme**, nous avons observé le temps de calcul pour les différentes adaptations présentées dans la figure 4.23. Ces expérimentations ont été effectuées sur un portable équipé d'un Pentium4 2.8GHz et de 512Mo de mémoire vive.

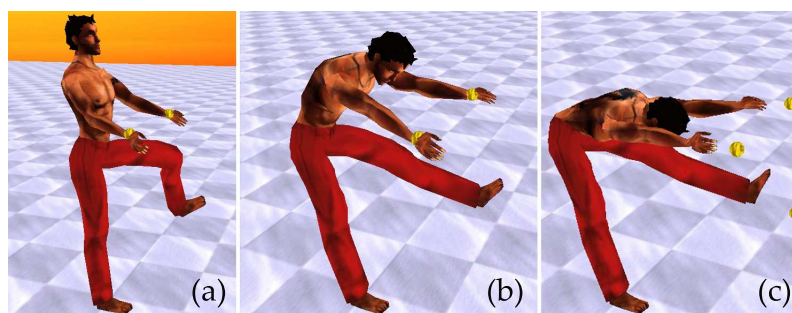


FIG. 4.23 – Différentes configurations utilisées pour les calculs de performance.

Le premier exemple (cf. figure 4.23a) utilise deux contraintes placées sur les poignets du personnage et une autre sur sa cheville. Les cibles sont facilement accessibles aussi seules les solutions analytiques sont nécessaires. C'est la raison pour laquelle l'erreur des adaptations est toujours nulle comme le montre le tableau 4.1. Dans ce tableau (ainsi que dans les suivants), les deux premières colonnes définissent les paramètres d'optimisation utilisés :  $CmSeuil$  est le seuil utilisé pour arrêter l'adaptation et  $Prec_i$  est la précision associée aux contraintes. La colonne *Nombre d'itérations* comptabilise les itérations qui

ont été effectuées au cours de l'adaptation. La colonne *Temps* définit le temps de calcul nécessaire pour toutes les itérations en  $\mu s$ . L'*Erreur globale* est la moyenne des erreurs des contraintes de l'adaptation. Enfin, la colonne *Nombre de pers. à 25Hz* donne le nombre de personnages qui peuvent être animés à 25Hz.

<i>CmSeuil</i> (mm)	<i>Prec<sub>i</sub></i> (mm)	Nombre d'itérations	Temps ( $\mu s$ )	Erreur globale (mm)	Nombre de pers. à 25Hz
0.01	1	2	88	0	454
0.1	1	2	88	0	454
1	1	2	88	0	454
10	1	2	88	0	454
0.01	10	2	88	0	454
0.01	100	2	88	0	454

TAB. 4.1 – Performances pour 3 groupes utilisés par 3 contraintes proches et un groupe utilisé pour l'appui sur le sol.

La deuxième configuration (cf. figure 4.23b) permet de constater que le nombre d'itérations n'est jamais élevé lorsque les contraintes sont quasiment accessibles (cf. tableau 4.2). En effet, pour 10 à 12 itérations, il est possible de solliciter tout le corps avec une précision d'adaptation allant de 2 et 5 millimètres.

<i>CmSeuil</i> (mm)	<i>Prec<sub>i</sub></i> (mm)	Nombre d'itérations	Temps ( $\mu s$ )	Erreur globale (mm)	Nombre de pers. à 25Hz
0.01	1	12	544	2	73
0.1	1	12	548	2	73
1	1	11	515	2	77
10	1	4	228	20	175
0.01	10	10	440	5	90
0.01	100	3	117	43	341

TAB. 4.2 – Performances pour 4 groupes utilisés par 3 contraintes lointaines et un groupe utilisé pour l'appui sur le sol.

Les contraintes qui ne sont pas accessibles (cf. figure 4.23c) sont toujours celles qui nécessitent le plus de temps de calcul (cf. tableau 4.3). En effet, elles demandent un étirement complet de la chaîne cinématique. Les itérations peuvent alors atteindre le nombre de 50 pour avoir la meilleure adaptation possible à ces contraintes. La valeur de l'erreur est ici donnée par rapport à la posture qui respecte au mieux les différentes contraintes. En effet, la véritable erreur globale n'a pas de sens au regard de contraintes qui ne peuvent être atteintes. Elle peut en effet varier de 0 à l'infini sans que cela ne modifie la posture finale.

$CmSeuil$ (mm)	$Prec_i$ (mm)	Nombre d'itérations	Temps ( $\mu s$ )	Erreur globale (mm)	Nombre de pers. à 25Hz
0.01	1	48	2258	0(49)	17
0.1	1	48	2345	0(49)	17
1	1	24	1240	9(58)	32
10	1	8	443	48(97)	90
0.01	10	48	2262	0(49)	17
0.01	100	11	537	42(91)	74

TAB. 4.3 – Performances pour 4 groupes utilisés par 3 contraintes inaccessibles et un groupe utilisé pour l'appui sur le sol.

## 4.4 Discussion

Dans ce chapitre, nous avons décrit une **nouvelle méthode d'adaptation cinématique** d'une posture de référence à des contraintes placées n'importe où sur le corps humain. Dans notre contexte d'applications interactives, cette méthode est utilisée directement à chaque pas de temps pour obtenir des adaptations de mouvements. Elle permet ainsi d'animer jusqu'à 540 personnages à 25Hz sur un portable grand public. Toutefois, cette méthode peut également être associée à des techniques classiques telles que les cartes de déplacement lorsque les applications offrent la connaissance complète des mouvements.

Cependant, pour que cette adaptation de posture puisse être utilisée pour l'animation de mouvements, il faut que la continuité des postures soit vérifiée lorsque les contraintes sont déplacées de manière continue. Or, ce sont les successions d'orientations des segments en sens contraire lors de l'adaptation qui entraînent une mauvaise répartition de l'adaptation et surtout des discontinuités entre deux postures successives. Notre algorithme a justement apporté des solutions pour supprimer cette mauvaise répartition des adaptations qui créent des "zig-zags" dans les chaînes cinématiques. En effet, il utilise des solutions analytiques qui réduisent la longueur des chaînes mais surtout qui permettent de placer instantanément les extrémités des chaînes dans leur meilleure configuration, celle qui minimise l'erreur avec les contraintes. Notre algorithme assure ainsi que toutes les rotations sont minimales et effectuées dans le même sens, suivant le même ordre grâce à l'utilisation des groupes qui ordonnent les adaptations. Ainsi, la répartition est toujours la même et une variation de la contrainte entraîne juste une variation de ces rotations dans le sens qui convient.

En plus de ce nouvel algorithme de cinématique inverse, nous avons proposé dans ce chapitre une **représentation du personnage en plusieurs couches** qui permet d'obtenir des adaptations réalistes et variées des mouvements. En effet, il est important, par exemple dans le cadre de l'animation de foules, d'avoir des adaptations différentes pour chaque personnage. Cette représentation répond donc à ces exigences en définissant plusieurs niveaux qui sont utilisés en fonction des besoins :

- le **squelette du personnage** qui est commun à tous les humanoïdes. Cette représentation définit la structure osseuse qui peut être adaptée, c'est-à-dire la chaîne cinématique du personnage ;

- les **groupes** du corps humain. Ils permettent de découper le squelette afin d’ordonner les adaptations des différentes parties du corps humain. Ces groupes permettent également d’ouvrir la représentation à d’autres formes de créature. Ainsi, en utilisant une telle représentation, il est plus facile d’utiliser l’algorithme d’adaptation sur une structure cinématique différente, par exemple celle d’une araignée ;
- les **caractéristiques individuelles** des humanoïdes. Ces caractéristiques comprennent la morphologie, les butées articulaires mais aussi les masses qui serviront à gérer la cinétique ainsi que d’autres paramètres comme la souplesse. En effet, nous avons montré que les butées articulaires seules ne permettaient pas de toujours choisir la posture la plus réaliste. De plus, ce niveau de représentation permet de personnaliser les adaptations à chaque individu.

Grâce à la représentation des caractéristiques individuelles des personnages, l’algorithme peut accéder automatiquement à des informations spécifiques à chaque individu. L’utilisateur peut alors contrôler le module d’adaptation cinématique simplement avec les quelques paramètres suivants :

- une posture de référence qui est fournie automatiquement par le moteur d’animation en fonction des actions que l’utilisateur a choisies pour son personnage ;
- des contraintes qui permettent de changer le mouvement ;
- un paramètre pour donner le seuil d’erreur utilisé pour savoir quand la posture est correctement adaptée et qu’il n’est plus nécessaire de poursuivre l’adaptation ;
- et enfin, un paramètre de contrôle de la translation de la racine de l’humanoïde. Ce dernier paramètre permet de simuler la motivation du personnage. Pour l’utilisation des mêmes contraintes, ce paramètre va permettre de choisir l’énergie que le personnage est prêt à investir dans son geste.

A la fin de la revue de littérature concernant les techniques de cinématique inverse (cf. section 1.3.3), un tableau a été présenté contenant les différents avantages et inconvénients de ces méthodes. Nous pouvons à présent comparer l’algorithme présenté dans ce chapitre (dernière ligne du tableau : “Contribution”) avec ces méthodes :

Méthode	Temps de calcul	Réalisme	Lois biomécaniques
CCD	++	+	+
Jacobien	+	+++	+++
Hybride	++	++	+
Contribution	+++	++(+)	++

TAB. 4.4 – Comparaison des méthodes principales de cinématique inverse.

Dans le tableau 4.4, un des signes + a été mis entre parenthèses au niveau du réalisme de l’adaptation. En effet, nous nous plaçons dans le contexte d’applications interactives où le futur n’est pas connu. A ce titre, l’adaptation présentée est parfaitement réaliste et mérite les trois +. Cependant, lorsque le futur peut être connu et dans le cas de larges adaptations, l’animation peut ne pas paraître réaliste. Par exemple, pour une préhension d’une cible lointaine, le bras est tendu avant de solliciter le tronc ce qui se justifie au niveau de la posture mais qui donne une impression irréaliste lorsque l’on passe à l’anima-

tion, c'est-à-dire l'enchaînement des postures. En effet, une discontinuité de l'accélération apparaît au moment où le bras se retrouve tendu et que le tronc commence à être mobilisé. Cela n'intervient que lorsque les mouvements deviennent dynamiquement différents. Dans notre exemple, le mouvement de référence n'utilise pas le tronc. Aucun profil de vitesse ni d'accélération n'est donc présent et, par conséquent, il ne peut être respecté. Ce problème de discontinuité peut être facilement corrigé en utilisant des techniques classiques de filtrage, comme Gleicher et coll. [Gleicher 98b] l'ont fait avec les cartes de déplacement par exemple (cf. section 1.3.2.2).



## Chapitre 5

# Adaptation cinétique

Le module d'adaptation cinématique présenté dans le chapitre précédent permet d'adapter une posture de manière à respecter un certain nombre de contraintes géométriques. Cependant, la posture obtenue n'est pas toujours réaliste suivant les contraintes fournies par l'utilisateur. En effet, lorsqu'un personnage se penche en avant pour attraper une cible qui se trouve très loin devant lui par exemple, il devrait perdre l'équilibre comme l'illustre la figure 5.1a. Cette notion d'équilibre n'est pas prise en compte par les contraintes géométriques puisqu'elle nécessite, entre autres, la connaissance des masses des segments du personnage. La cinétique est justement la prise en compte de ces masses. La figure 5.1a montre une posture qui pourrait être obtenue par cinématique inverse (si l'utilisateur appliquait un paramètre *CmRacine* à 100%). Cette posture est réalisable mais pas sans chuter. Si le contexte de l'animation veut que le personnage reste sur ses pieds dans une posture quasi-statique, il faut utiliser de la cinétique inverse qui consiste à adapter la posture afin de respecter une position de centre de masse (cf. section 1.3.1.2). Les figures 5.1b et 5.1c montrent deux postures qui sont adaptées par cinétique inverse.

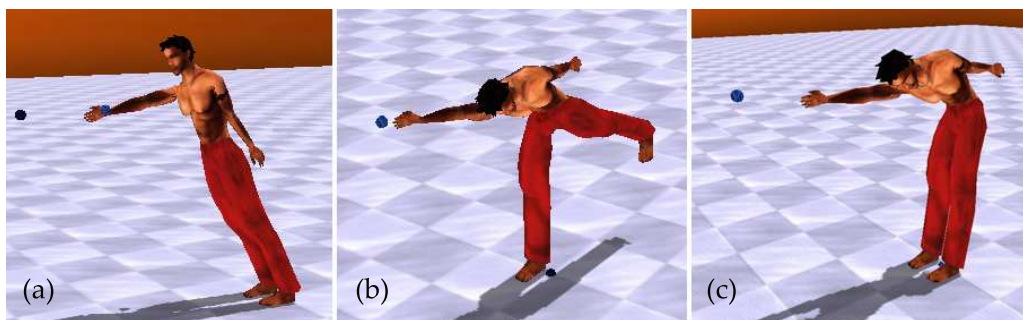


FIG. 5.1 – La cinétique permet de prendre en compte les notions d'équilibre. La posture (a) est adaptée sans cinétique, le personnage devrait tomber. Les parties (b) et (c) présentent deux manières de respecter l'équilibre en utilisant la cinétique.

Le contrôle du centre de masse, appelé COM, est donc très important pour l'animation de personnages. Nous proposons dans ce chapitre un module d'adaptation, décrit dans la figure 5.2, qui prend en compte la gestion de ce centre de masse par l'utilisation de la

**cinétique inverse.** La position du COM peut être fournie par l'utilisateur ou un module de calcul extérieur. Cependant, nous proposons également de spécifier le centre de masse à respecter  $COM'$  à partir de la posture de référence (en grisé sur la figure 5.2). C'est le mode par défaut de notre module d'adaptation cinétique qui est donc utilisé lorsqu'aucun centre de masse n'est fourni.

Le respect du centre de masse  $COM'$  fourni par l'algorithme d'adaptation peut être effectué de deux manières :

- le respect en 3D de  $COM'$  : le centre de masse de la posture  $COM$  est alors placé sur la position de  $COM'$ . Ce mode est utilisé pour les phases aériennes des sauts par exemple ;
- le respect de  $COM'$  dans le plan horizontal : le centre de masse de la posture  $COM$  est alors placé à la verticale de  $COM'$ . Cette utilisation permet de traiter les problèmes d'équilibre pour les postures quasi-statiques par exemple.

Le paramètre  $CnMode$  permet de choisir parmi ces deux modes d'adaptation.

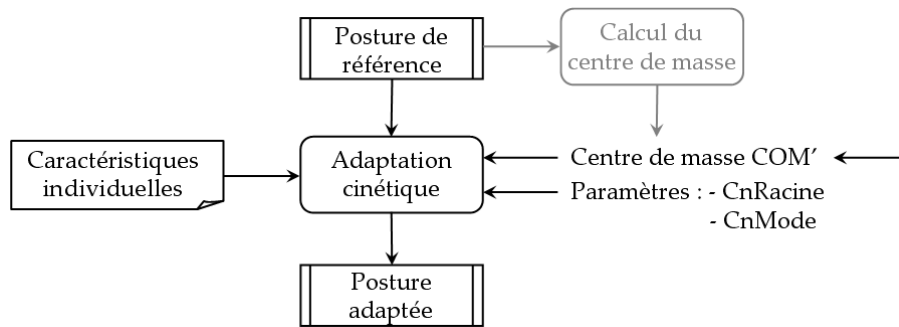


FIG. 5.2 – Schéma général de l'adaptation cinétique.

L'algorithme de cinétique inverse est complémentaire de celui de cinématique inverse. Il est donc naturel de vouloir les utiliser ensemble. Cependant, il n'est pas possible de se contenter de les utiliser séquentiellement. En effet, les modifications apportées par la cinétique inverse peuvent rompre les contraintes vérifiées par la cinématique inverse. Il faut donc les intégrer dans un même algorithme. Il est possible de les intégrer tous les deux dans une boucle itérative globale mais les performances ne seraient alors plus à la hauteur des espérances et surtout il y aurait un fort risque d'oscillation dû aux actions antagonistes des deux adaptations. Nous proposons donc un algorithme commun dans lequel sont fusionnés les algorithmes de cinématique et de cinétique inverse.

Ce chapitre décrit dans un premier temps la représentation des données utilisée par l'algorithme de cinétique inverse (cf. section 5.1) puis l'algorithme lui-même (cf. section 5.2). Dans un deuxième temps, ce chapitre décrit la fusion des algorithmes de cinématique et de cinétique inverse dans la section 5.3. Enfin, la section 5.4 donne des résultats obtenus par cet algorithme commun ainsi que des valeurs de performance. Une discussion sur ce chapitre est décrite dans la section 5.5.

## 5.1 Représentation des données

L'algorithme de cinétique inverse utilise plusieurs données cinétiques. La sous-section 5.1.1 détaille les informations nécessaires ainsi que la manière de les représenter. La sous-section 5.1.2 montre alors comment le découpage en groupes du corps humain présenté dans la section 4.1 peut être également utilisé dans le contexte de l'adaptation cinétique.

### 5.1.1 Représentation des données cinétiques

Le centre de masse  $COM$  du personnage est calculé à partir du centre de masse local  $COM_s$  de chacun des segments  $s$  du corps humain. Il est effectué à l'aide d'une somme pondérée de ces centres de masse locaux :

$$COM = \frac{\sum_s (m_s COM_s)}{\sum_s m_s} \quad (5.1)$$

où  $m_s$  est la masse du segment  $s$ .

Les segments peuvent être représentés de diverses manières, par exemple par un cône tronqué ou un cylindre. La position du  $COM_s$  est alors placée entre l'articulation proximale  $Prox_s$  (racine du segment) et l'articulation distale  $Dist_s$  (extrémité du segment). La figure 5.3 montre une représentation d'un segment par un cône tronqué ainsi qu'un exemple de positionnement du centre de masse local.  $COM_s$  est donc défini par l'équation :

$$COM_s = Prox_s + R_s(Dist_s - Prox_s)$$

où  $R_s$  est le pourcentage du vecteur  $\overrightarrow{(Prox_s, Dist_s)}$  qui permet de retrouver la position du centre de masse local  $COM_s$ .

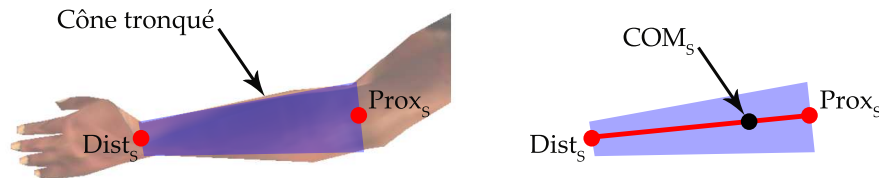


FIG. 5.3 – Le segment est modélisé par un cône tronqué. Le centre de masse  $COM_s$  du segment est placé entre l'articulation proximale  $Prox_s$  et distale  $Dist_s$ .

Puisque les positions des articulations sont fournies par la posture elle-même, seuls les masses des segments et le pourcentage  $R_s$  doivent être obtenus. Des tables anthropométriques existent pour retrouver ces paramètres. Elles permettent en effet d'avoir les informations associées à chaque segment simplement à partir de quelques paramètres globaux tels que la hauteur du personnage ou encore sa masse totale. Trois catégories de tables anthropométriques existent en fonction de la provenance de leurs informations :

- à partir de données cadavériques. Ces tables ont été créées dès les années 50 à partir de la découpe de cadavres d'hommes âgés [Dempster 55, Chandler 75, Clauser 69];
- à partir de données évaluées par tomographie et imagerie médicale. Ces informations ont été recueillies par des techniques de radiographie par rayons gamma sur une population de sportifs [Zatsiorsky 90, de Leva 96];

- et par évaluation de modèles géométriques : contrairement aux tables précédentes qui effectuent des mesures directes, ces tables permettent de retrouver les données à partir des propriétés géométriques des segments. Ainsi, des équations de régression permettent de retrouver ces données à partir de plusieurs mesures anatomiques comme le périmètre de la cuisse [Jensen 78, Jensen 94, Pavol 02, Wicke 03].

Un exemple de table anthropométrique est fourni par le tableau 5.1 issu des études de Dempster [Dempster 55]. La première ligne fournit la masse d'un segment en fonction de la masse totale.  $R_s$  est fourni dans la deuxième ligne. Avec cette table, l'utilisateur n'a plus qu'à spécifier la masse globale du personnage. Dans la mesure où ce paramètre est spécifique à chaque humanoïde, il est stocké dans la représentation des caractéristiques individuelles (cf. section 4.1) ainsi que le nom de la table utilisée.

Segment	Tronc	Bras	Avant-bras	Main	Cuisse	Mollet	Pied	Tête
%Masse	0.497	0.028	0.016	0.006	0.1	0.0465	0.0145	0.081
$R_s$	0.66	0.436	0.43	0.506	0.433	0.433	0.5	1

TAB. 5.1 – Données anthropométriques issues de Dempster [Dempster 55].

### 5.1.2 Découpage en groupes

Pour améliorer les performances de l'adaptation cinématique, un niveau de représentation du corps humain en groupes a été proposé dans la section 4.1. Cette représentation est à nouveau utilisée pour l'adaptation cinétique. En effet, le centre de masse est calculé en faisant la somme pondérée des centres de masse des segments. Nous proposons de regrouper ces centres de masse pour tous les segments appartenant au même groupe. Ainsi, le centre de masse  $COM_g$  d'un groupe  $g$  qui contient un ensemble  $ES$  de segments est défini par l'équation :

$$COM_g = \frac{\sum_{s \in ES} (m_s COM_s)}{\sum_{s \in ES} m_s} \quad (5.2)$$

À partir de cette équation et de celle définissant le centre de masse global 5.1, le calcul du  $COM$  est donc :

$$COM = \frac{\sum_g (m_g COM_g)}{\sum_g m_g} \quad (5.3)$$

$$(5.4)$$

Nous proposons donc d'utiliser une hiérarchie de centres de masse basée sur celle des groupes (cf. figure 4.4) mais avec un niveau de plus qui correspond aux segments (cf. figure 5.4). La partie (a) montre le niveau des centres de masse des segments :  $COM_s$ . Le calcul des centres de masse des groupes de niveau 3, à savoir les bras et la tête, est montré dans la partie (b). Ils sont calculés à partir des centres de masse des segments  $COM_s$ . La partie (c) montre l'utilisation des centres de masse du niveau précédent afin d'obtenir les centres de masse des groupes de niveau 2 : les jambes et le tronc. Enfin, la partie (d) montre le calcul final du centre de masse global.

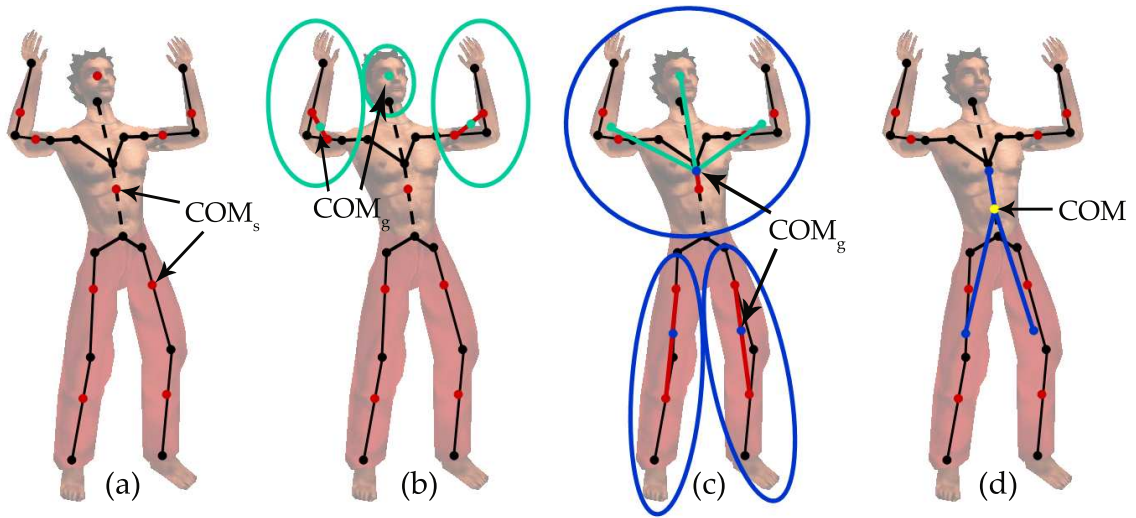


FIG. 5.4 – Hiérarchie des centres de masse. (a) Centres de masse des segments. (b) Centres de masse des groupes de niveau 3. (c) Centres de masse des groupes de niveau 2. (d) Centres de masse du niveau 1, c'est-à-dire le centre de masse global.

Cette hiérarchie est très importante puisqu'elle intègre intrinsèquement le fait qu'adapter le tronc modifie aussi les centres de masse des groupes du niveau en-dessous, c'est-à-dire les bras et la tête et par répercussion au niveau le plus bas, les centres de masse des segments concernés.

## 5.2 Algorithme d'adaptation

A l'instar de l'adaptation cinématique, nous proposons d'utiliser un algorithme itératif afin de résoudre le problème d'adaptation cinétique. De plus, nous allons tirer profit de la représentation hiérarchique des centres de masse présentée dans la section 5.1.2. Le principe de cet algorithme itératif est d'adapter localement les groupes. Ces derniers sont ainsi adaptés un par un en considérant que les autres sont fixes pendant leur adaptation.

Cette similitude entre les algorithmes de cinématique et de cinétique inverse est bien évidemment intentionnelle puisque ces deux algorithmes doivent fonctionner ensemble.

### 5.2.1 Algorithme général

L'objectif de la cinétique inverse est d'adapter la posture de référence afin de placer son centre de masse  $COM$  sur la position fournie par l'utilisateur  $COM'$  (à droite de la figure 5.2). Il existe deux méthodes de prise en compte de ce centre de masse. La première consiste à ne respecter sa position que sur le plan horizontal, c'est-à-dire avec les composantes  $X$  et  $Y$  ( $Z$  étant orienté vers le haut). Cette utilisation permet de respecter **l'équilibre du personnage** en situation quasi-statique. C'est l'utilisation la plus courante de la cinétique inverse. La seconde consiste à respecter **totalemtent la position du centre de masse en 3D**. Par exemple, lorsqu'un personnage saute, la trajectoire de son

centre de masse au cours du temps doit suivre une parabole puisque seule la gravité est appliquée si on ignore les forces de frottement de l'air.

L'algorithme proposé utilise donc le niveau le plus haut de la hiérarchie des centres de masse afin de déterminer l'erreur à corriger (cf. figure 5.4). En effet, c'est le centre de masse global qui est présent dans ce premier niveau. Le but est donc de le déplacer vers  $COM'$ . Cette erreur est ensuite répercutée sur les groupes par des adaptations locales en suivant l'ordre fixé par la hiérarchie de groupes. Toutefois, l'adaptation cinétique peut également utiliser la racine de la hiérarchie. Plusieurs stratégies d'adaptation sont alors possibles. En effet, en partant d'une posture de référence qui peut être issue de l'adaptation cinématique par exemple (cf. figure 5.5a), il est possible de solliciter uniquement les groupes pour essayer de respecter au mieux la position du centre de masse fourni (cf. figure 5.5b). Cependant, leur utilisation ne suffit pas forcément à rééquilibrer la posture. Il est également possible de déplacer uniquement la racine de la hiérarchie afin d'assurer le respect du centre de masse fourni (cf. figure 5.5c). Dans ce cas, il est quasiment toujours possible de rééquilibrer la posture. Cependant, dans le cadre d'une utilisation combinée avec l'adaptation cinématique, le personnage se retrouve très loin de la contrainte modélisée par la sphère à gauche. La meilleure solution est alors de combiner ces deux méthodes d'adaptation afin de respecter le centre de masse imposé tout en se rapprochant au maximum de la contrainte à atteindre (cf. figure 5.5d). Le choix de la stratégie employée est configurée par le paramètre  $CnRacine$ . A l'instar de  $CmRacine$  (cf. section 4.2.3), ce paramètre représente le pourcentage de l'erreur qui doit être utilisé par le déplacement de la racine.

Afin de pouvoir imposer le fait qu'une partie de l'adaptation est effectuée par la racine et le reste par les groupes, il est important de déplacer la racine en premier. En effet, nous allons montrer que l'adaptation de chaque groupe tend à toujours respecter au mieux le centre de masse fourni. Si c'est accessible par la modification des groupes, le centre de masse  $COM'$  est alors atteint et la racine n'a plus d'adaptation à faire. C'est la raison pour laquelle l'algorithme général de l'adaptation cinétique utilise dans un premier temps la racine afin de prendre une partie de l'adaptation à son compte puis adapte les groupes. Afin de compléter l'adaptation si les groupes sont insuffisants, elle est entièrement finalisée par l'utilisation du déplacement de la racine à 100% de l'erreur restante :

### ***fonction adaptationCinétique()***

*Faire*

*adaptationCnRacine(CnRacine)*

*adaptationRapideSol()*

*adaptationCnGroupes()*

*adaptationCnRacine(100%)*

*adaptationRapideSol()*

*it = it + 1*

*Tant que ( (it < maxIt) & (¬terminé) )*

où *it* désigne l'itération courante, *maxIt* le nombre maximum d'itérations souhaité et *terminé* un booléen de valeur *vrai* lorsque l'adaptation est achevée.

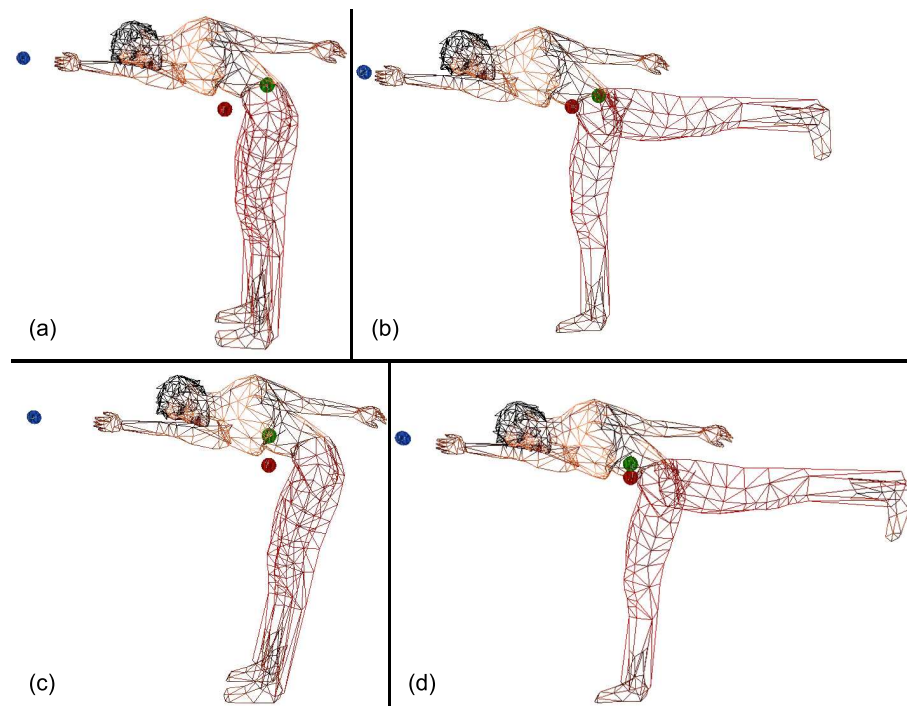


FIG. 5.5 – Stratégies de l'adaptation cinétique. La sphère rouge représente le COM de la posture, la verte la position demandée par l'utilisateur (contrainte sur le centre de masse). L'équilibre est donc respecté lorsque les deux sphères sont l'une au-dessus de l'autre. (a) Posture de référence issue de l'adaptation cinématique. (b) L'adaptation en utilisant uniquement les groupes n'est pas suffisante pour respecter l'équilibre de la posture. (c) Seule l'adaptation de la racine est utilisée et l'équilibre est vérifié. Par contre, le personnage est loin de sa cible. (d) L'utilisation combinée des adaptations des groupes et de la racine permet de respecter l'équilibre tout en se rapprochant de la cible.

### 5.2.2 Prise en compte des groupes

Afin d'être le plus générique possible, l'utilisateur doit pouvoir **sélectionner les groupes** qu'il souhaite impliquer dans l'adaptation cinétique. Par défaut, ils sont tous sélectionnés. Ce choix des groupes est également utilisé pour la fusion des deux adaptations cinématique et cinétique présentées dans la section 5.3.

Bien évidemment, lorsqu'aucun groupe n'est sélectionné et donc ne peut être utilisé, l'adaptation doit entièrement être réalisée par le déplacement de la racine. L'algorithme est par conséquent le suivant :

```

fonction adaptationCinétique()
    it = 0
    EGC = ensembleGroupesPourCinétique()
    Si (EGC = ∅)
        Faire
            adaptationCnRacine(100%)
            adaptationRapideSol()
            it = it + 1
        Tant que ( it < maxIt ) & ( ¬terminé )
    Sinon

```

```

Faire
  adaptationCnRacine(CnRacine)
  adaptationRapideSol()
  adaptationCnGroupes()
  adaptationCnRacine(100%)
  adaptationRapideSol()
  it = it + 1
Tant que ( it < maxIt ) & ( ¬terminé )
Fin du si

```

où la fonction *ensembleGroupesPourCinétique()* fournit l'ensemble des groupes sélectionnés par l'utilisateur.

L'ensemble *EGC* définit donc l'ensemble des groupes qui sont utilisés pour l'adaptation. Il est à présent important de connaître **l'ordre de prise en compte de ces groupes**. En effet, comme pour l'adaptation cinématique, l'ordre d'adaptation des groupes est important pour le réalisme de la posture finale.

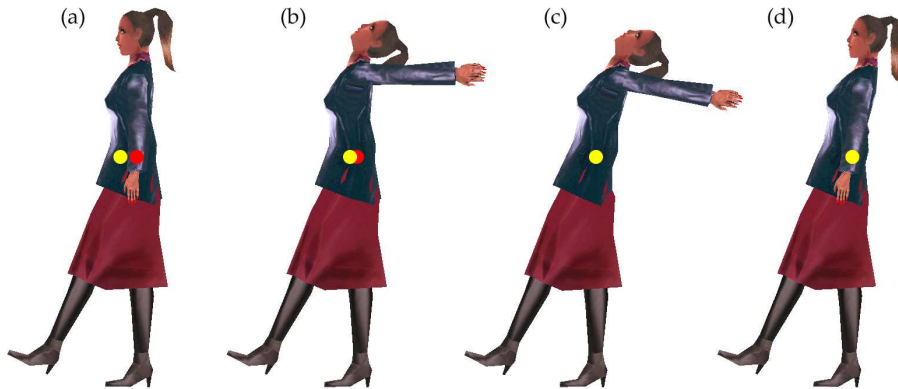


FIG. 5.6 – L'ordre de prise en compte des groupes est important. La sphère rouge représente le COM à respecter. La jaune est celle de la posture. (a) Posture de référence. (b) et (c) Utilisation des groupes les plus légers d'abord. Les bras et la tête sont complètement horizontaux pour essayer de contrebalancer la jambe mais leurs masses sont insuffisantes et le tronc est alors utilisé. (d) Le tronc qui est le groupe le plus lourd est utilisé en premier. Il arrive très facilement à contrebalancer la jambe avec une très petite rotation vers l'arrière du personnage.

Comme l'adaptation est cinétique, le choix de l'ordre dépend directement des masses des groupes. Considérons les deux ordres possibles illustrés par la figure 5.6. Si l'algorithme parcourt les groupes des plus légers au plus lourds, l'adaptation commence par les bras et la tête. Ainsi, la posture obtenue est celle de la figure 5.6b. En effet, leurs masses étant inférieures à celle d'une seule jambe (cf. la table anthropométrique 5.1), ils doivent être mobilisés au maximum afin de déplacer au mieux le centre de masse total. Le tronc est ensuite utilisé et la posture illustrée dans la figure 5.6c est obtenue. Bien évidemment, cette posture n'est pas réaliste. A l'inverse, si les groupes les plus lourds sont sollicités les premiers, le tronc sera tout de suite adapté. Avec sa masse très importante (quasiment égale à la moitié de celle du corps humain), une petite rotation en arrière du tronc permet de tout de suite compenser l'avance de la jambe (cf. figure 5.6d).

Il est donc indispensable de parcourir les groupes en partant des plus lourds aux plus



légers. Le parcours de la hiérarchie des groupes va donc se faire du niveau le plus haut (1) au niveau le plus bas (3). Les groupes sont donc adaptés dans l'ordre suivant : le tronc, les jambes, les bras et enfin la tête.

### 5.2.3 Adaptation des groupes

L'adaptation d'un groupe va permettre de contrôler son centre de masse local et intrinsèquement de modifier le centre de masse global. Cependant, il est difficile en local d'effectuer une adaptation pour résoudre un problème global. L'adaptation d'un groupe  $G$  est donc décomposée en deux étapes : la détermination de son centre de masse local désiré  $COM'_G$  qui doit être respecté au mieux et ensuite l'adaptation correspondante des membres à l'intérieur du groupe (cf. figure 5.7). Les deux sous-sections suivantes décrivent ces deux étapes.

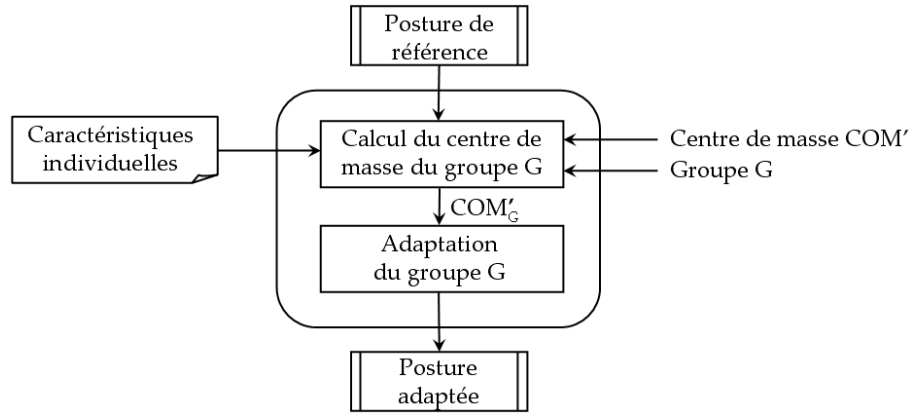


FIG. 5.7 – Adaptation d'un groupe.

#### 5.2.3.1 Détermination du centre de masse local

De par la nature même de l'algorithme itératif d'optimisation choisi, lorsqu'un groupe  $G$  est adapté, les autres sont considérés comme fixes. Puisque le centre de masse global est calculé à partir de la somme des centres de masse locaux (cf. équation 5.3) et que tous les autres groupes sont fixes, il est possible d'en déduire la position idéale du centre de masse  $COM'_G$  du groupe adapté  $G$  afin que le centre de masse global  $COM$  soit placé à la position fournie par l'utilisateur  $COM'$ . Il suffit en effet d'isoler le groupe dans l'équation 5.3 :

$$COM' = \frac{\sum_{g \neq G} (m_g COM_g)}{\sum_g m_g} + \frac{m_G COM'_G}{\sum_g m_g}$$

On en déduit donc la position désirée du centre de masse local du groupe  $G$  :

$$COM'_G = \frac{\sum_g m_g * COM' - \sum_{g \neq G} (m_g COM_g)}{m_G} \quad (5.5)$$

Lorsque ce calcul est appliqué au tronc, la masse  $m_G$  ne doit pas simplement être la masse du tronc. En effet, les groupes des bras et de la tête sont des fils du tronc dans la

hiérarchie. Par conséquent, lorsque le tronc est adapté, ces autres groupes le sont également. Leurs masses doivent être également prises en compte. La masse  $m_G$  est donc la somme des masses du tronc, des bras et de la tête lorsque  $G$  est le groupe du tronc.

### 5.2.3.2 Adaptation locale d'un groupe

Conformément au parcours hiérarchique défini dans la section 5.2.2, l'adaptation cinétique se fait du niveau d'abstraction le plus élevé (le centre de masse global), vers celui le plus précis (les centres de masse des segments). Le principe de l'adaptation d'un groupe reprend cette idée et à partir d'une modification globale du groupe répercute cette adaptation à ses différentes parties. Ainsi, le tronc et la tête sont représentés de manière abstraite par un unique segment. Les groupes des bras et des jambes sont, quant à eux, modélisés comme une chaîne à deux segments.

Deux types de solutions analytiques sont donc nécessaires pour résoudre notre problème : une pour le tronc et la tête d'une part et une pour les membres d'autre part.

**Le tronc et la tête** sont modélisés comme un segment, il suffit donc d'utiliser une rotation afin de rapprocher le centre de masse local  $COM_G$  de ce groupe  $G$  vers la position souhaitée  $COM'_G$  conformément au calcul précédent (cf. section 5.2.3.1). Cependant, cette rotation dépend du paramètre  $CnMode$  qui définit le contrôle désiré sur le centre de masse :

- le respect en 3D de  $COM'_G$ . Le centre de masse du groupe  $COM_G$  doit être placé le plus près possible de  $COM'_G$ . La rotation appliquée cherche donc à aligner les points *Racine*,  $COM_G$  et  $COM'_G$  (cf. figure 5.8b) ;
- le respect de  $COM'_G$  dans le plan horizontal. Le centre de masse du groupe  $COM_G$  doit être placé à la verticale de  $COM'_G$ . La figure 5.8c montre l'utilisation de cette rotation.

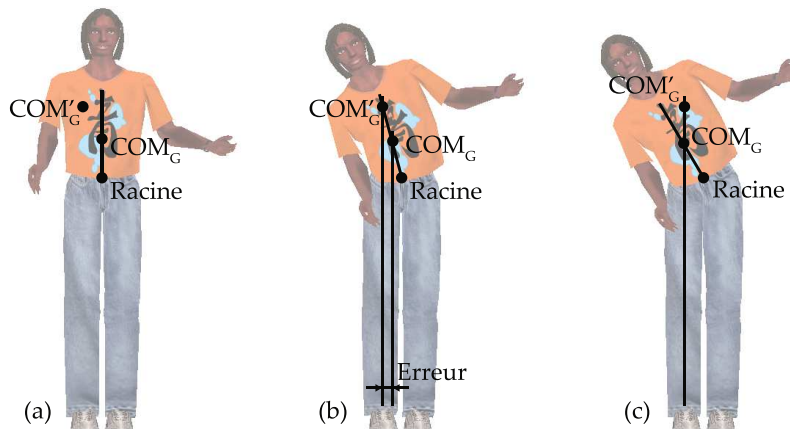


FIG. 5.8 – Adaptation cinétique du tronc. (a) La posture de référence. (b) La rotation qui rapproche au mieux les positions  $COM_G$  et  $COM'_G$  n'est pas correcte. (c) La rotation correcte est celle qui les place l'un au-dessus de l'autre.

**Les membres** sont composés de deux segments. La solution analytique doit donc pouvoir modifier ces deux segments, appelés respectivement  $S_1$  et  $S_2$  en partant de la racine vers l'extrémité, afin de placer le centre de masse du membre  $COM_G$  à la position calculée  $COM'_G$ . Cette adaptation est effectuée en deux étapes :

- extension du membre pour placer  $COM_G$  à la même distance que  $COM'_G$  de l'articulation proximale (cf. figure 5.9a) ;
- puis rotation du membre pour placer  $COM_G$  sur la position  $COM'_G$  (cf. figure 5.9b). La figure 5.9c montre la posture finale obtenue.

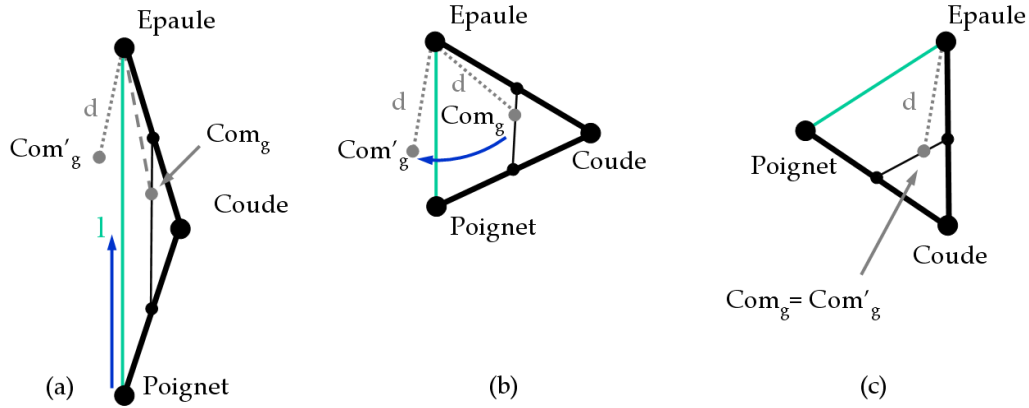


FIG. 5.9 – Adaptation des membres en deux étapes. (a) Posture de référence. (b) La longueur variable du membre est changée de manière à placer  $COM_G$  à la même distance que  $COM'_G$  de l'épaule. (c) Une rotation est effectuée pour placer  $COM_G$  sur  $COM'_G$ .

Pour décrire ces deux étapes, considérons l'exemple du bras. L'étape d'**extension du membre** permet de trouver instantanément sa bonne longueur, c'est-à-dire la distance entre le poignet et l'épaule dans notre représentation normalisée du squelette (cf. section 4.1). La figure 5.10 montre le bras composé des segments  $S_1$  de longueur  $l_1$  et  $S_2$  de longueur  $l_2$ . Ces longueurs sont fournies par la morphologie du personnage. La position de leur centre de masse est calculée à l'aide de la table anthropométrique fournie par la représentation des caractéristiques individuelles (cf. section 5.1). Ainsi, le centre de masse  $COM_1$  est retrouvé par l'équation :  $COM_1 = Epaule + R_1(Coude - Epaule)$ . Le même calcul est appliqué pour retrouver  $COM_2$  sur le segment  $S_2$ .

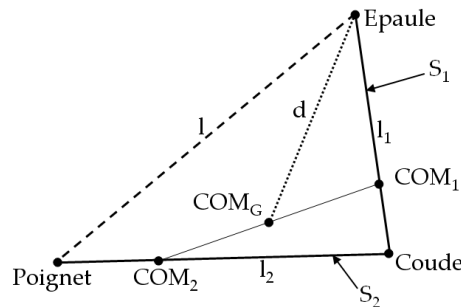


FIG. 5.10 – Position du centre de masse d'un membre.  $COM_1$  (resp.  $COM_2$ ) est la position du COM du segment  $S_1$  (resp.  $S_2$ ).  $COM_g$  est la position du COM du bras.

La position du centre de masse du groupe  $COM_G$  est alors calculée en utilisant un nouveau ratio  $R_3$  qui dépend des masses  $m_1$  et  $m_2$  des deux segments  $S_1$  et  $S_2$ . Ces masses

sont également fournies par la table anthropométrique.

$$R_3 = \frac{m_2}{m_1 + m_2}$$

Avec ces notations, le problème d'extension du bras peut être reformulé de la manière suivante : il faut trouver la nouvelle longueur  $l$  (entre l'épaule et le poignet) à partir d'une valeur  $d$  qui représente la distance entre l'épaule et le centre de masse à atteindre  $COM'_G$ . En effet, l'utilisation de cette valeur  $d$  assure alors que le centre de masse du membre  $COM_G$  est à la même distance que  $COM'_G$  par rapport à l'épaule, excepté lorsqu'il est hors d'atteinte. Dans ce cas particulier, le bras est tendu et la distance maximale  $dMax$  qu'il peut y avoir entre l'épaule et le centre de masse du bras est calculée à l'aide de l'équation suivante (cf. figure 5.11) :

$$\begin{aligned} dMax &= COM_1 + R_3(COM_2 - COM_1) \\ dMax &= R_1 l_1 + R_3 [(R_2 l_2 + l_1) - R_1 l_1] \end{aligned}$$

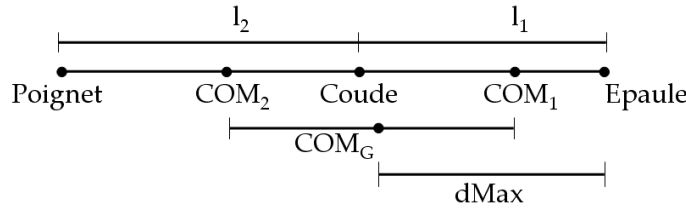


FIG. 5.11 – Lorsque le bras est tendu, la distance entre le centre de masse et l'épaule est maximale :  $dMax$ .

Par conséquent, si la distance  $d$  entre le centre de masse à atteindre  $COM'_G$  et l'épaule est supérieure à cette valeur  $dMax$ , alors  $COM'_G$  n'est pas accessible et la longueur  $l$  recherchée est fixée à la longueur du bras. Autrement, cette longueur  $l$  peut être calculée en utilisant l'équation suivante (détaillée en annexe C) :

$$l = \sqrt{\frac{d^2 - F}{G}} \quad (5.6)$$

avec  $F$  et  $G$  deux constantes telles que :

$$\begin{aligned} F &= l_1^2 \left[ (R_1 + R_3)^2 + \right. \\ &\quad \left. R_3 [R_2(R_1 R_3 - R_1 - R_3) + R_1(R_1 R_3 - 2(R_1 + R_3))] \right] \\ &\quad + l_2^2 R_2 R_3 [R_3(R_2 + R_1 - 1) - R_1] \\ G &= R_2 R_3 (R_3 + R_1 - R_1 R_3) \end{aligned}$$

Il ne reste alors plus qu'à effectuer la **rotation du membre** conformément à celle appliquée pour le tronc (cf. sous-section 5.8). La figure 5.12 illustre l'adaptation de la jambe afin de respecter le centre de masse  $COM'_G$  fourni (sphère claire).

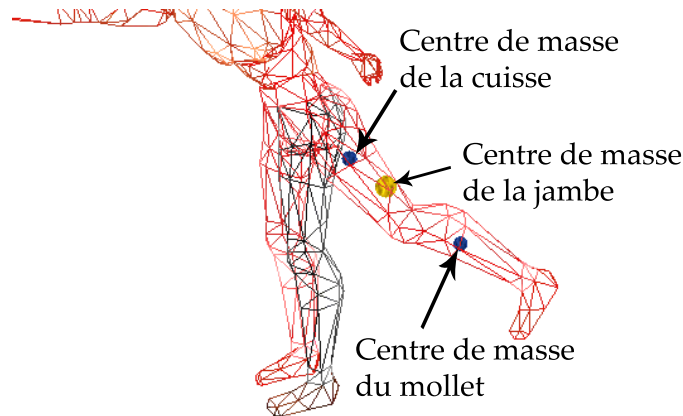


FIG. 5.12 – Adaptation de la jambe pour respecter la position demandée de  $COM'_G$  (sphère claire). Les sphères foncées représentent les positions des centres de masse de la cuisse et du mollet.

#### 5.2.4 Adaptation de la racine

Plusieurs stratégies sont possibles pour l'adaptation cinétique, comme l'a montré la section 5.2.1. En effet, il est possible d'utiliser la racine de la hiérarchie, les groupes qui composent le corps humain ou encore de combiner ces deux formes d'adaptation. L'utilisateur a donc besoin de pouvoir paramétrer la stratégie qu'il souhaite employer pour son adaptation. L'algorithme utilise, pour cela, un paramètre appelé  $CnRacine$ . Ce paramètre définit le pourcentage d'utilisation de la racine pour l'adaptation cinétique de la posture. Le reste de l'adaptation est bien évidemment pris en charge par les groupes dans la limite de leur capacité.

Ce pourcentage  $CnRacine$  est utilisé directement sur le vecteur allant du centre de masse actuel de la posture  $COM$  vers celui qu'il faut respecter :  $COM'$ . Ainsi, cette adaptation est réalisée uniquement au premier niveau de la hiérarchie. La nouvelle position du centre de masse est donc :  $COM_n = CnRacine \times (COM' - COM)$ . Ce déplacement de la racine peut rompre les contacts des pieds sur le sol. Il est donc toujours associé, comme pour la cinématique, à une adaptation rapide au sol.

Dans l'algorithme général présenté dans la section 5.2.1, l'adaptation de la racine est tout d'abord utilisée avec le paramètre  $CnRacine$  pour qu'une partie de l'adaptation cinétique soit gérée par le déplacement de la racine. Les groupes sont alors adaptés afin de prendre en charge le reste de l'adaptation. Cependant, il est possible que l'adaptation des groupes ne suffise pas à corriger complètement la position du centre de masse. En effet, si une jambe est utilisée pour contrebalancer une préhension d'un objet placé loin devant le personnage comme dans la figure 5.5b, le centre de masse ne peut pas être déplacé suffisamment. La masse de la jambe, en effet, est largement inférieure à celles du tronc et du bras réunies. Une dernière adaptation de la racine est donc réalisée avec une valeur de 100% de l'erreur restante. L'algorithme assure ainsi que l'adaptation cinétique respecte presque complètement la position souhaitée du centre de masse avec juste une seule itération. En effet, seule l'adaptation au sol peut légèrement déplacer la jambe d'appui et par conséquent le centre de masse. Les itérations suivantes ne servent donc qu'à affiner l'adaptation cinétique.

### 5.2.5 Critère d'arrêt et convergence

A l'instar de l'adaptation cinématique, l'adaptation cinétique est terminée lorsque la variation de l'erreur  $\Delta CnErreur_k$  pour l'itération  $k$  est inférieure à un seuil  $CnSeuil$  fourni par l'utilisateur. Cette variation de l'erreur  $\Delta CnErreur_k$  est calculée à partir de l'erreur  $CnErreur_k$  qui est la distance entre le centre de masse  $COM$  de la posture de référence et celui à atteindre  $COM'$  qui est fourni par l'utilisateur :

$$\Delta CnErreur_k = CnErreur_k - CnErreur_{k-1}$$

Cette distance est calculée soit en utilisant la norme du vecteur  $\overrightarrow{\|(COM, COM')\|}$  lorsque la cinétique inverse est utilisée en 3D pour contrôler par exemple la trajectoire du centre de masse lors d'une phase de saut, soit en calculant la distance entre les projections des deux centres de masse  $COM$  et  $COM'$  sur le sol lorsque seul l'équilibre doit être vérifié.

En théorie, il serait possible d'utiliser directement un critère d'arrêt basé sur l'erreur  $CnErreur_k$  et pas sur la variation d'erreur  $\Delta CnErreur_k$  puisqu'une seule contrainte est appliquée : le centre de masse. Cependant, l'algorithme serait alors sensible au fait que l'on puisse ou non atteindre le centre de masse  $COM'$  fourni par l'utilisateur. Si ce dernier est placé trop loin du personnage, les contraintes de contact des pieds sur le sol empêchent de l'atteindre et l'algorithme ne converge pas. Nous avons donc choisi un critère basé sur une variation d'erreur afin d'offrir l'algorithme le plus stable. Un centre de masse placé si loin reste cependant une erreur de l'utilisateur puisque le but de la cinétique inverse est d'obtenir des postures réalistes ce qui n'est évidemment pas le cas dans ces conditions. Aussi, nous proposons de combiner ces deux approches afin d'avoir un algorithme stable mais qui permet, lorsque le centre de masse peut être atteint, de s'arrêter dès que l'erreur de position du centre de masse  $CnErreur_k$  devient plus petit que le seuil d'erreur fourni par l'utilisateur :  $CnErreurMax$ . Il est ainsi intuitif de donner une précision d'adaptation grâce à ce dernier paramètre. On conserve malgré tout une sécurité de convergence avec le paramètres  $CnSeuil$ . Le critère d'arrêt est alors :  $terminé = (\Delta CnErreur_k < CnSeuil) | (CnErreur_k < CnErreurMax)$ .

Dans la mesure où toutes les modifications apportées par les différentes adaptations diminuent la distance entre le COM actuel et celui désiré, notre critère d'arrêt définit une fonction décroissante. De plus, l'erreur est toujours une valeur strictement positive puisqu'elle représente une distance, la fonction est par conséquent bornée par la valeur zéro. La fonction converge donc toujours.

Même si l'adaptation cinétique présentée dans ce chapitre peut être utilisée toute seule pour assurer l'équilibre par exemple, elle prend vraiment toute son ampleur lorsqu'elle est couplée à l'adaptation cinématique. Dans la section suivante 5.3, nous détaillons donc la fusion des deux adaptations cinématique et cinétique. Nous montrerons alors directement les résultats de cette fusion ainsi que les performances de l'ensemble.

### 5.3 Fusion des adaptations cinématique et cinétique

Lors de l'adaptation d'une posture à des contraintes, il est important de pouvoir à la fois respecter ces dernières mais également de bénéficier du réalisme apporté par l'utilisation de la cinétique. Cette section propose donc un algorithme commun aux deux adaptations. Les notions de performance sont toujours au centre de nos préoccupations. Il n'est donc pas souhaitable de se contenter d'imbriquer les deux algorithmes dans une même boucle itérative entraînant l'utilisation de boucles imbriquées plus coûteuses en temps de calcul.

Nous proposons d'utiliser une approche de type *Push-Pull* (cf. figure 5.13) pour l'animation de personnages. Cette technique, utilisée en radiosité, permet une cohérence de l'ensemble de la hiérarchie. En effet, la radiosité sur les surfaces émettrices doit être mise à jour en tenant compte de l'énergie apportée à tous les niveaux de la hiérarchie. Ceci est généralement effectué par un parcours en profondeur de la hiérarchie (cf. figure 5.13) : lors de la descente, les valeurs d'irradiance sont ajoutées jusqu'au niveau des feuilles de la hiérarchie, où leur valeur cumulée est multipliée par la valeur de réflectance de la feuille, pour obtenir la radiosité de cette feuille. Lors de la remontée, la radiosité d'un élément est calculée comme la moyenne des radiosités de ses fils pondérée par les aires des sous-surfaces associées. A l'issue de la phase de *Push-Pull*, la radiosité est représentée de manière cohérente à tous les niveaux hiérarchiques du maillage.

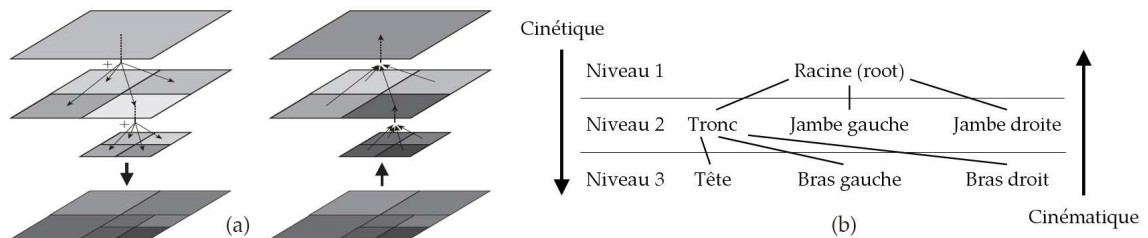


FIG. 5.13 – (a) Algorithme Push-Pull utilisé pour le calcul de la radiosité. (b) L'algorithme fusionné de cinématique et cinétique inverse.

L'adaptation cinématique parcourt la hiérarchie de groupes du plus bas niveau au plus haut en effectuant les adaptations aux contraintes. Au retour, dans le sens inverse de la hiérarchie, nous effectuons les calculs de cinétique. Nous décrivons donc dans cette section l'algorithme d'adaptation cinématique et cinétique.

#### 5.3.1 Choix des groupes

L'adaptation cinétique ne doit pas aller à l'encontre de l'adaptation cinématique. Elle ne doit donc pas adapter les groupes utilisés par la cinématique. Par exemple, lors d'une préhension, il n'est pas acceptable de changer la configuration du bras pour équilibrer la posture. Aussi, c'est l'ensemble des groupes non contraints par la cinématique qui est sélectionné pour être utilisé par la cinétique inverse. Dans un souci de généricité, l'utilisateur peut également décider de n'utiliser que certains groupes non contraints.

### 5.3.2 Fusion des adaptations de la racine

Au cours de l'adaptation cinématique, la racine est déplacée de manière à se rapprocher des cibles lorsque celles-ci sont difficiles à atteindre. L'adaptation cinétique l'utilise aussi afin de rapprocher le centre de masse de la posture avec la position imposée par l'utilisateur. Si les deux adaptations sont utilisées en même temps sans précaution, l'algorithme a peu de chance de converger. En effet, si les adaptations sont de sens opposés, elles vont s'annuler l'une l'autre et la posture va osciller d'un côté et de l'autre.

L'adaptation cinétique est la plus importante. En effet, l'adaptation cinématique seule est remplacée par cet algorithme commun lorsque l'utilisateur souhaite une posture réaliste qui respecte par exemple les notions d'équilibre. Il est donc indispensable de respecter plus strictement la cinétique que la cinématique. Il est en effet normal que le personnage ne puisse pas atteindre des cibles placées trop loin comme l'illustre les exemples (c) et (d) de la figure 5.5. En revanche, il est primordial qu'il ne tombe pas.

L'adaptation cinétique de la racine est donc prioritaire. Ainsi, seules les composantes (X, Y ou Z) non utilisées par cette adaptation peuvent être sollicitées par l'adaptation cinématique de la racine. Par exemple, lors de la gestion de l'équilibre, seules les deux composantes du plan horizontal sont utilisées pour la cinétique. Par conséquent, l'adaptation cinématique peut déplacer la racine sur l'axe vertical. Par contre, lorsque la position du centre de masse doit être parfaitement respectée dans les trois dimensions comme dans l'exemple d'un saut, alors la cinématique n'a aucune liberté pour utiliser la racine.

### 5.3.3 Algorithme final

Grâce à la similarité des algorithmes proposés pour les adaptations cinématique et cinétique et comme une attention particulière est portée pour ne pas avoir d'interaction antagoniste entre ces deux adaptations, il est possible d'utiliser directement le contenu des boucles des deux algorithmes dans une seule et même boucle itérative :

**fonction adaptationCinématiqueEtCinétique()**

```

adaptationCompleteSol()
it = 0
EGC = ensembleGroupesPourCinétique()
Faire
  adaptationCmGroupe(Tête)
  adaptationCmGroupe(Bras droit)
  adaptationCmGroupe(Bras gauche)
  adaptationCmGroupe(Jambe droite)
  adaptationCmGroupe(Jambe gauche)
  adaptationCmGroupe(Tronc)
  adaptationCmRacine()
  adaptationRapideSol()
  Si (CnErreurk > CnErreurMax)
    Si (EGC = ∅)
      adaptationCnRacine(100%)
      adaptationRapideSol()
  Sinon
    adaptationCnRacine(CnRacine)

```



```

    adaptationRapideSol()
    adaptationCnGroupes()
    adaptationCnRacine(100%)
    adaptationRapideSol()
  Fin du si
Fin du si
it = it + 1
Tant que ( it < maxIt ) & ( ¬terminé )

```

Cet algorithme revient à effectuer les deux adaptations séquentiellement au cours d'une itération. Cependant, le critère d'erreur *CnErreurMax* sur le centre de masse (expliqué dans la sous-section 5.2.5) est directement inséré dans la boucle itérative afin d'éviter de faire des adaptations cinétiques inutiles. En effet, l'adaptation cinématique requiert généralement beaucoup plus d'itérations que la cinétique, surtout lorsque les contraintes se rapprochent de leurs cibles. Par conséquent, plusieurs itérations seraient calculées pour rien au niveau de la cinétique entraînant une augmentation des temps de calcul. Ce terme est donc enlevé du critère d'arrêt final mais intervient indirectement en ne calculant pas l'adaptation cinétique si l'erreur au niveau de la position du centre de masse est insuffisante.

### 5.3.4 Critère d'arrêt et convergence

Chaque adaptation, cinématique et cinétique, possède son propre critère d'arrêt. Il est donc nécessaire de tenir compte de ces deux critères afin d'évaluer la fin de l'adaptation de l'algorithme fusionné. Toutefois, nous venons de voir que le critère d'erreur sur la position du centre de masse est déjà intégré dans la boucle. Par conséquent, nous proposons d'utiliser le critère d'arrêt suivant :

$$\text{terminé} = (\Delta CmErreur_k \leq CmSeuil) \ \& \ (\Delta CnErreur_k \leq CnSeuil)$$

Il est alors important de s'assurer de la convergence de cet algorithme. En effet, les deux adaptations sont exécutées indépendamment l'une de l'autre. Il ne faut donc pas qu'elles soient antagonistes. Cependant, les deux seuils d'erreur utilisés  $CmErreur_k$  et  $CnErreur_k$  ne sont pas réellement indépendants. En effet, l'adaptation cinématique des groupes du personnage va entraîner une modification de la posture et par conséquent de la position de son COM et de l'erreur cinétique. Ainsi, si l'algorithme d'adaptation cinématique converge, alors les variations de posture deviennent négligeables et la variation de position du COM devient également négligeable. La variation de l'erreur sur le COM est alors infime et l'algorithme a convergé. Or l'algorithme d'adaptation cinématique converge pour deux raisons. La première est que l'ensemble  $Em$  des degrés de liberté utilisés par l'adaptation cinématique est complètement disjoint de celui utilisé par l'adaptation cinétique  $En$  :  $Em \cap En = \emptyset$ . En effet, lors de la fusion des deux adaptations, une attention particulière a été prise afin d'assurer cette indépendance : une adaptation ne peut donc pas modifier les groupes contrôlés par l'autre adaptation. La seconde raison est que l'adaptation cinématique rapproche le personnage des cibles à atteindre et par conséquent également son COM. Les corrections apportées par l'adaptation cinétique vont donc compenser ce décalage en déplaçant le COM du personnage dans la direction opposée aux cibles. L'adaptation cinématique se retrouve donc plus loin des cibles mais toujours du même côté. Les

adaptations successives ne vont donc qu'augmenter la déformation des chaînes cinématiques jusqu'à un point de convergence lorsque l'adaptation cinématique ne peut plus se rapprocher des cibles. La variation de la posture est alors négligeable ainsi que celle du COM, l'algorithme a convergé.

## 5.4 Résultats

Cette section présente des résultats de l'utilisation de l'algorithme d'adaptation cinématique seul ou bien couplé avec la cinématique. Le premier résultat est celui de la figure 5.5 qui montre les différentes stratégies qui peuvent être employées pour l'adaptation cinétique. Cet exemple a en effet été créé à partir de l'algorithme décrit ci-dessus.

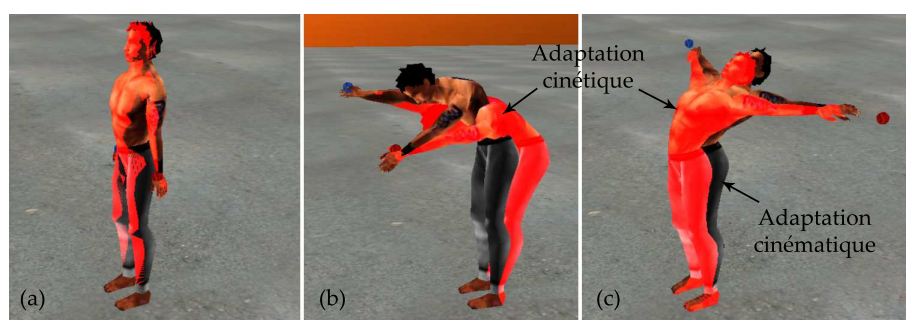


FIG. 5.14 – Comparaison d'une posture avec ou sans adaptation cinétique. (a) Posture de référence. (b) et (c) Deux configurations différentes. La posture en couleurs normales est adaptée par la cinématique seule. L'autre posture en rouge est également adaptée par la cinématique.

Le second résultat montre la différence entre une adaptation cinématique seule et l'utilisation de l'algorithme fusionné. À partir de la même posture de référence (cf. figure 5.14a), deux configurations différentes des contraintes sont fournies (cf. figure 5.14b et 5.14c). La posture en rouge est celle qui est soumise à l'adaptation cinématique et cinétique.

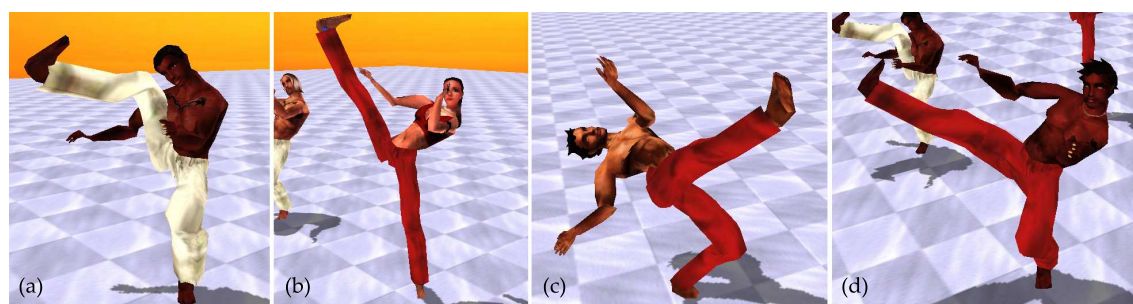


FIG. 5.15 – Utilisation des adaptations cinématique et cinétique de la racine. À partir de la posture de référence (a), plusieurs configurations sont réalisées qui utilisent les deux adaptations cinématique et cinétique (b, c et d).

Le troisième résultat montre l'utilisation des adaptations cinématique et cinétique de

la racine du personnage. En effet, nous avons discuté précédemment de l'importance de contrôler correctement cette double adaptation de la racine. Les exemples des figures 5.15b, 5.15c et 5.15d montrent des configurations qui nécessitent à la fois une adaptation cinétique pour conserver l'équilibre mais également cinématique pour pouvoir atteindre les cibles demandées.

Afin de juger du **réalisme des adaptations** proposées, nous allons comparer des postures réelles avec des postures adaptées par notre algorithme. Trois postures sont donc nécessaires :

- la posture de comparaison : c'est la posture capturée à laquelle on se compare (en jaune dans les figures de cette section). De cette posture sont extraites les contraintes à respecter. En effet, lors de l'acquisition de cette posture, la personne capturée devait toucher un certain nombre de cibles avec ces mains et/ou ces pieds.
- la posture de référence : c'est celle que l'on souhaite adapter. Cette posture a été également capturée.
- la posture adaptée : c'est la posture fournie par notre algorithme d'adaptation cinématique et cinétique à partir de la posture de référence et des contraintes extraites de la posture de comparaison.

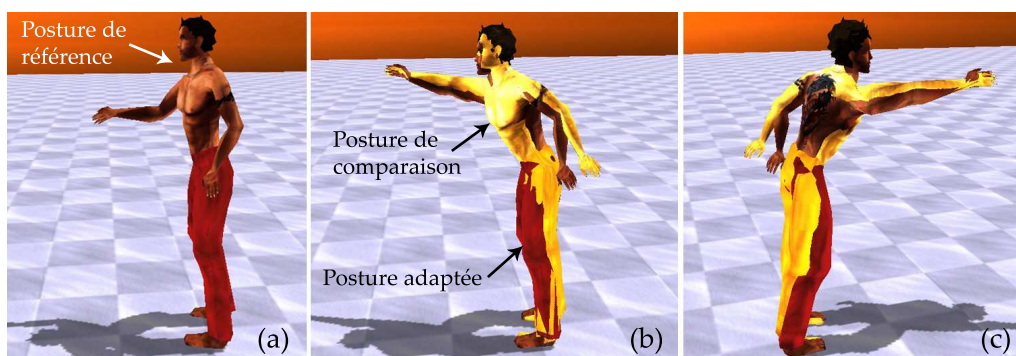


FIG. 5.16 – Comparaison d'une posture réelle (en jaune) avec celle fournie par notre adaptation cinématique. (a) Posture de référence. (b, c) Différentes vues de la comparaison entre les postures réelle et adaptée.

La figure 5.16 montre un exemple d'adaptation d'une posture de préhension. La cible est placée loin devant le personnage afin de solliciter son tronc. La posture de référence (cf. figure 5.16a) est issue d'une saisie d'un objet proche. Les parties (b) et (c) de la figure montrent deux vues différentes de l'adaptation. Le personnage en jaune représente la posture de comparaison.

La figure 5.17 présente cette fois une adaptation à trois contraintes placées sur les poignets et la cheville droite. La partie (a) montre une fois de plus la posture de référence. Les autres parties montrent différentes vues de l'adaptation.

Afin de juger des **performances de l'algorithme d'adaptation cinématique et cinétique**, nous avons observé le temps de calcul pour deux des configurations présentées dans les résultats de l'adaptation cinématique : celle de la figure 4.23a qui montre une

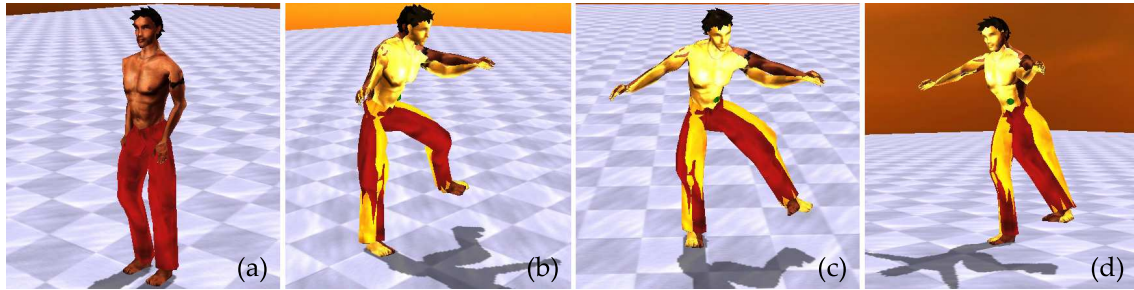


FIG. 5.17 – Comparaison d’une posture réelle (en jaune) avec celle fournie par notre adaptation cinématique. (a) Posture de référence. (b, c, d) Différentes vues de la comparaison entre les postures réelle et adaptée.

adaptation à des contraintes accessibles et celle de la figure 4.23c qui montre la pire des configurations puisqu’aucune des contraintes n’est accessible. Cette dernière configuration donne les mêmes résultats que celle de la figure 4.23b puisqu’avec le recul du personnage pour maintenir son équilibre, les contraintes sont dans les deux cas inaccessibles. Ces expérimentations ont été également effectuées sur un portable équipé d’un Pentium4 2.8GHz et de 512Mo de mémoire vive.

Le premier exemple (cf. figure 4.23a) utilise deux contraintes placées sur les poignets du personnage et une sur sa cheville gauche. Les cibles sont facilement accessibles dans cet exemple. Dans les tableaux 5.2 et 5.3, les deux premières colonnes définissent les paramètres d’optimisation utilisés :  $CnSeuil$  est le seuil utilisé pour arrêter l’adaptation cinématique et  $CnErreurMax$  est la précision pour effectuer ou non l’adaptation cinématique lors d’une itération. Pour tous ces exemples, nous avons pris les paramètres cinématiques à des valeurs de  $CmSeuil = Prec_i = 1mm$ . La colonne *Nombre d’itérations* comptabilise les itérations qui ont été effectuées au cours de l’adaptation. La colonne *Temps* définit le temps de calcul nécessaire pour toutes les itérations en  $\mu s$ . *L’Erreur COM* est l’erreur de position du centre de masse global. Enfin, la colonne *Nombre de pers. à 25Hz* donne le nombre de personnages qui peuvent être animés à 25Hz.

$CnSeuil$ (mm)	$CnErreurMax$ (mm)	Nombre d’itérations	Temps ( $\mu s$ )	Erreur COM (mm)	Nombre de pers. à 25Hz
0.01	1	5	284	0.26	140
0.1	1	5	284	0.26	140
1	1	5	284	0.26	140
10	1	5	284	0.26	140
0.01	10	3	188	7.9	212
0.01	100	2	141	57	283

TAB. 5.2 – Performances pour 3 groupes utilisés par 3 contraintes proches.

Les contraintes qui ne sont pas accessibles (cf. figure 4.23c) sont toujours celles qui nécessitent le plus de temps de calcul (cf. tableau 5.3). En effet, elles demandent un étirement complet de la chaîne cinématique et une adaptation importante de la cinétique

pour maintenir l'équilibre. Il est intéressant de constater dans ces deux exemples que l'erreur finale du centre de masse ne dépasse jamais l'erreur maximale fixée par le paramètre  $CnErreurMax$ .

$CnSeuil$ (mm)	$CnErreurMax$ (mm)	Nombre d'itérations	Temps ( $\mu s$ )	Erreur COM (mm)	Nombre de pers. à 25Hz
0.01	1	22	1701	0	23
0.1	1	22	1707	0	23
1	1	14	1131	0.07	35
10	1	14	1135	0.07	35
0.01	10	19	1474	2	27
0.01	100	20	1524	89	26

TAB. 5.3 – Performances pour 4 groupes utilisés par 3 contraintes inaccessibles.

## 5.5 Discussion

Dans ce chapitre, nous avons décrit un **algorithme de cinétique inverse** qui permet d'adapter les postures de plusieurs dizaines, voire de centaines de personnages en temps-réel. A l'instar de la cinématique inverse, il peut être utilisé pour l'adaptation de mouvements aussi bien pour les applications interactives, quand aucune connaissance sur le futur n'est disponible, que pour celles qui offrent une connaissance complète des mouvements. En effet, deux postures adaptées successivement par cet algorithme sont modifiées de manière continue si le centre de masse l'est aussi. Le déplacement de la racine est proportionnel au vecteur d'erreur de la position du centre de masse. Ainsi, lorsqu'il évolue de manière continue, le vecteur d'erreur, et par conséquent la racine, sont également déplacés de manière continue (à condition que le paramètre  $CnRacine$  évolue lui-même de manière continue). Les adaptations des membres ne sont effectuées que par des rotations pour s'approcher du centre de masse demandé. Par conséquent, si son déplacement est continu, les rotations utilisées afin de se placer dans la bonne configuration le sont aussi. L'algorithme de cinétique inverse assure donc la continuité de ces adaptations lorsque le centre de masse est déplacé de manière continue.

A l'instar de l'adaptation cinématique, une attention particulière a aussi été portée sur la simplicité d'utilisation. En effet, cette cinétique inverse est simplement contrôlée par les paramètres suivants :

- une posture de référence ;
- un paramètre  $CnErreurMax$  qui permet de spécifier le seuil d'erreur maximal souhaité sur le centre de masse ;
- un autre paramètre  $CnSeuil$  qui est utilisé uniquement pour assurer la stabilité de l'algorithme même lors de son emploi par un utilisateur non averti. Ce paramètre est le seuil de variation d'erreur en-dessous duquel l'algorithme s'arrête ;
- un paramètre  $CnMode$  qui permet de choisir le type de contrôle du centre de masse utilisé ;

- et enfin, un paramètre *CnRacine* qui permet de spécifier la stratégie d'adaptation cinétique.

Ce dernier paramètre est très important puisqu'il permet de choisir entre une adaptation réalisée uniquement avec les groupes, une autre effectuée seulement avec le déplacement du bassin ou encore une combinaison de toutes ces adaptations comme le montre la figure 5.5. Cela permet notamment d'avoir des adaptations différentes pour chaque personnage ou suivant le contexte. L'adaptation cinétique repose de plus sur l'utilisation de la même représentation des données que l'adaptation cinématique et en particulier celle des caractéristiques individuelles. Il est ainsi possible de paramétrer chaque personnage (par exemple en changeant ses masses corporelles) afin de personnaliser son adaptation. Par extension, il est possible de lui ajouter des masses externes. Par exemple, si un personnage porte un objet, la masse de cet objet peut être temporairement ajoutée à celle de la main.

Nous avons également proposé un nouvel algorithme permettant de tirer profit à la fois de cette adaptation cinétique et de l'adaptation cinématique. Cette **fusion des adaptations cinématiques et cinétiques** est réalisée à l'aide d'un algorithme de type *Push-Pull*. Il offre l'avantage d'avoir à la fois une adaptation facile d'une posture à des contraintes géométriques mais également d'assurer un réalisme de la posture finale par l'introduction de la cinétique et par conséquent des notions d'équilibre. De plus, cette fusion des algorithmes a été intégrée dans **une seule et même boucle itérative** qui permet ainsi de pouvoir adapter près de 280 personnages en temps-réel en utilisant quatre groupes du corps humain pour l'adaptation cinématique (les deux bras et une jambe étant utilisés pour vérifier des contraintes, l'autre jambe pour l'appui sur le sol) tout en respectant l'équilibre du personnage par le contrôle du centre de masse. Cette fusion a été réalisée de manière à ce qu'aucune interaction n'existe entre les deux adaptations cinématique et cinétique. Ainsi, chacun contrôle de manière continue ses propres parties du corps et le résultat est par conséquent modifié de manière continue. Ce point est très important pour l'utilisation de cet algorithme pour l'animation.

Enfin, cet algorithme commun d'adaptation cinématique et cinétique est **contrôlé par les mêmes paramètres** que chacun des algorithmes séparément afin que son utilisation soit aussi simple que chacune des adaptations. De plus, l'algorithme propose de détecter automatiquement les groupes non utilisés par l'adaptation cinématique afin de les impliquer dans la cinétique inverse. L'utilisateur a alors le choix de tous les utiliser ou alors d'en sélectionner une partie. Dans la pratique, l'utilisation de la tête pour la cinétique inverse n'est pas une bonne idée. En effet, sa faible amplitude ne lui permet pas de déplacer significativement le centre de masse du personnage. Ainsi, si elle est utilisée et qu'aucun autre groupe ne parvient à respecter l'équilibre, elle se penche de manière à compenser au maximum l'erreur sur la position du centre de masse. La posture est alors irréaliste et l'implication de la tête est inutile dans la mesure où c'est le déplacement de la racine qui est nécessaire pour corriger une telle situation.

Dans la discussion du chapitre 4, un tableau a été présenté pour comparer l'algorithme d'adaptation cinématique seul avec les différentes méthodes vues dans la revue de la littérature (cf. section 1.3.3). Une nouvelle colonne est ajoutée à ce tableau concernant le contrôle du centre de masse. Puisque notre algorithme respecte complètement le COM

imposé à la fois en tenant compte de la racine et des groupes, trois '+' sont marqués dans le tableau 5.4 en face de la ligne "Contribution". Le signe (+) du tableau 5.4 est expliqué dans la section 4.4.

Méthode	Temps de calcul	Réalisme	Contrôle du COM	Lois biomécaniques
CCD	++	+	-	+
Jacobien	+	+++	+++	+++
Hybride	++	++	-	+
Contribution	+++	++(+)	+++	++

TAB. 5.4 – Comparaison des méthodes principales de cinématique et cinétique inverse.





## Chapitre 6

# Respect de la dynamique

Le module d'adaptation cinématique et cinétique présenté dans le chapitre précédent permet d'adapter un mouvement à des contraintes tout en veillant au respect de la position de son centre de masse, par exemple pour gérer l'équilibre du personnage. Cependant, ce centre de masse peut également être utilisé pour respecter la dynamique sur le système simplifié COM, il est même l'un des paramètres fondamentaux pour qu'un mouvement paraisse réaliste [Safonova 05]. Nous proposons donc, dans ce chapitre, une méthode d'adaptation des sauts qui tire pleinement profit de l'adaptation cinématique et cinétique afin de respecter la dynamique de ce geste. Ainsi, cette méthode alimente l'adaptation cinématique et cinétique avec la position du centre de masse au cours du temps (cf. figure 6.1).

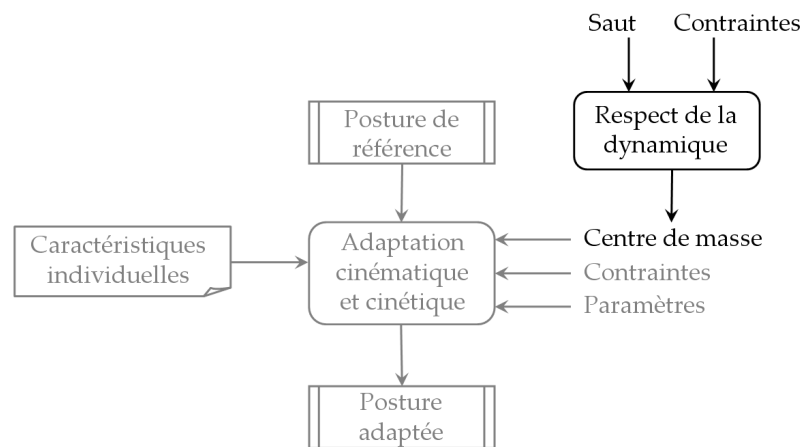


FIG. 6.1 – Schéma général du respect de la dynamique.

Afin que l'utilisateur soit en mesure d'adapter des sauts, il faut qu'il puisse poser des contraintes sur les mouvements, par exemple sauter plus loin ou plus haut. L'adaptation proposée dans ce chapitre modifie donc un saut de référence conformément aux contraintes spécifiées par l'utilisateur tout en respectant au mieux la dynamique du geste (cf. figure 6.1).

La section 6.1 analyse les différentes phases du saut et fournit les éléments qui peuvent

être utilisés pour l'adaptation. Grâce à ces éléments, la section 6.2 propose une représentation des données utilisées pour adapter le saut. La section 6.3 décrit quant à elle l'algorithme d'adaptation qui se base sur cette représentation. Enfin, les résultats sont donnés dans la section 6.4 et une discussion sur cette adaptation des sauts est proposée dans la section 6.5.

## 6.1 Analyse

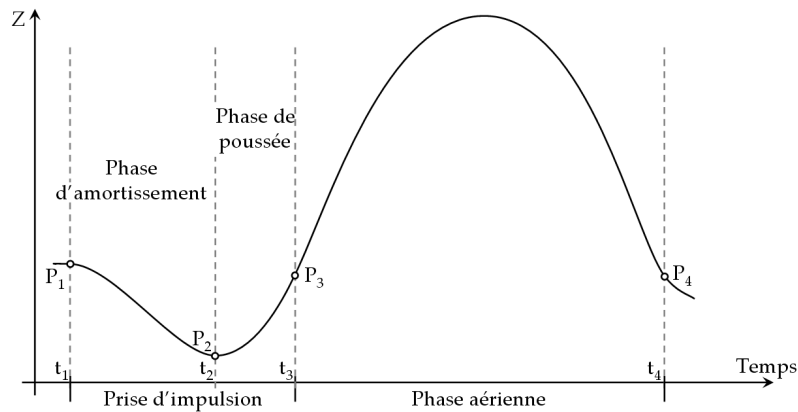


FIG. 6.2 – Trajectoire du centre de masse lors d'un saut. Deux phases sont présentes : la prise d'impulsion et la phase aérienne.

Un saut peut être décomposé en deux phases comme le montre la figure 6.2 :

- la **prise d'impulsion** est la période du saut pendant laquelle le personnage a les pieds sur le sol. Cette période très courte est une phase préparatoire du saut. En effet, c'est lors de ce court laps de temps que le personnage doit appliquer les forces suffisantes pour réussir son saut. Après cette impulsion, puisque les pieds ne sont pas en contact avec le sol, il n'est plus possible de modifier la trajectoire du COM.
- la **phase aérienne** est la période pendant laquelle l'utilisateur applique les contraintes afin de modifier ce saut. Il peut ainsi aisément choisir les points par lesquels doit passer le centre de masse pour obtenir un saut plus ou moins haut, ou plus ou moins long en fonction d'un obstacle ou d'un objectif donné.

Pour adapter un saut à des contraintes, il faut donc adapter la phase aérienne mais également la prise d'impulsion puisque c'est elle qui donne les moyens d'atteindre le nouveau but fixé par les contraintes.

Pendant la **phase aérienne**, la trajectoire du centre de masse du personnage a une forme parabolique (cf. figure 6.2). En effet, la relation fondamentale de la dynamique qui lie l'accélération de ce COM avec les forces externes  $\sum \vec{F}_{ext} = m \vec{a}$  nous permet d'écrire l'équation suivante puisque seule la force de la pesanteur (le poids) est appliquée (en ignorant les frottements de l'air) :

$$\sum \vec{F}_{ext} = \vec{P} \quad \implies \quad \vec{a} = \vec{g} \quad (6.1)$$

En décomposant cette équation suivant les trois axes, l'accélération est nulle pour les axes X et Y et est égale à la gravité  $g$  pour l'axe vertical Z. En intégrant deux fois ces équations, on obtient la trajectoire du centre de masse du personnage pendant la phase aérienne :

$$\begin{aligned}x &= \dot{x}_3.t + x_3 \\y &= \dot{y}_3.t + y_3 \\z &= -\frac{1}{2}.g.t^2 + \dot{z}_3.t + z_3\end{aligned}\tag{6.2}$$

où  $\dot{x}_3$ ,  $\dot{y}_3$  et  $\dot{z}_3$  sont les vitesses initiales de la phase aérienne du saut suivant les trois axes et  $x_3$ ,  $y_3$  et  $z_3$  définissent la position initiale du centre de masse de cette même phase. Ces données sont celles du point  $P_3$  de la figure 6.2.

Cette équation définit un polynôme de degré 2 qui ne dépend que des données initiales du saut, à savoir la vitesse et la position du point  $P_3$ . Toute la phase aérienne est donc uniquement dépendante des informations présentes sur ce point  $P_3$  qui est aussi la fin de la phase d'impulsion. C'est la raison pour laquelle la phase aérienne n'est utilisée que pour interpréter les contraintes et que la vraie difficulté de l'adaptation réside dans la phase d'impulsion. A l'inverse, adapter la phase d'impulsion revient donc à prendre en compte les modifications apportées par la phase aérienne sur le point  $P_3$ .

La **phase d'impulsion** est décomposée en deux parties : la phase d'amortissement et la phase de poussée (cf. figure 6.2). Lors de la première phase, le personnage baisse son centre de masse afin d'être au niveau du point  $P_2$  dans une posture qui lui permet d'effectuer une poussée suffisante lors de la seconde phase pour parvenir à réaliser le saut souhaité. Il est donc capital de conserver ces deux parties et, par conséquent, la forme de la trajectoire du centre de masse pendant l'impulsion. Ainsi, le personnage peut toujours se placer dans une posture au point  $P_2$  qui lui permet d'atteindre la position et la vitesse requises à la fin de l'impulsion, en  $P_3$ .

Afin d'adapter cette phase d'impulsion, il est possible de contrôler les forces appliquées lors de cette période. En effet, l'impulsion est la résultante des forces externes au cours du temps :  $\int_t^{t+\Delta t} \sum \vec{F}_{ext}.dt$ . Cependant, il n'est pas possible d'utiliser des solutions de type masse-ressort puisque le système moteur humain est actif et non passif. En effet, l'impulsion n'est pas simplement une accumulation/restitution d'énergie mais un système plus complexe où de nouvelles forces sont ajoutées lors de la phase de poussée (produites par le personnage).

Une étudiante du laboratoire LPBEM (Laboratoire de Physiologie et de Biomécanique de l'Exercice Musculaire) de l'Université Rennes 2, Carole Durocher, a proposé lors de sa thèse de travailler sur l'adaptation de ces forces [Durocher 05]. Ce travail est décomposé en deux grandes parties. La première consiste à retrouver les forces présentes dans le mouvement de référence. Un module de dynamique inverse est utilisé pour les déterminer à partir de la trajectoire du COM. Ces forces sont ensuite modélisées par des splines hiérarchiques. La deuxième partie du travail consiste à optimiser les points de contrôle de ces splines pour respecter les contraintes fixées par l'utilisateur. Le système utilise une approximation linéaire tangente afin de respecter une tâche primaire (l'impulsion nécessaire au saut) et

une tâche secondaire (la position finale du centre de masse). La nouvelle trajectoire du centre de masse est alors retrouvée en intégrant deux fois les splines modifiées. Cependant, cette approche repose sur des double intégrations et dérivations numériques qui rendent la résolution très instable.

C'est la raison pour laquelle nous proposons dans ce chapitre d'adapter directement la trajectoire du centre de masse global du personnage. De plus, pour éviter les problèmes liés à la double dérivation, **cette trajectoire** est uniquement représentée de manière **symbolique**. Enfin, nous proposons une méthode originale pour adapter les deux parties de la phase d'impulsion afin de conserver la forme de la trajectoire tout en respectant les consignes fournies par les contraintes appliquées sur la phase aérienne.

Dans le cadre de ce mémoire, nous traitons le saut vertical à titre d'exemple. L'adaptation du centre de masse le long de l'axe vertical Z est en effet la plus compliquée à réaliser puisque les forces présentes sont le poids et la force de réaction du sol.

## 6.2 Représentation des données

La figure 6.3 montre les différentes étapes utilisées pour effectuer l'adaptation. Les modules dessinés en noir permettent de passer du saut aux données nécessaires pour l'adaptation et sont décrits dans cette section. Le module d'adaptation du saut dessiné en gris est présenté dans la section suivante 6.3.

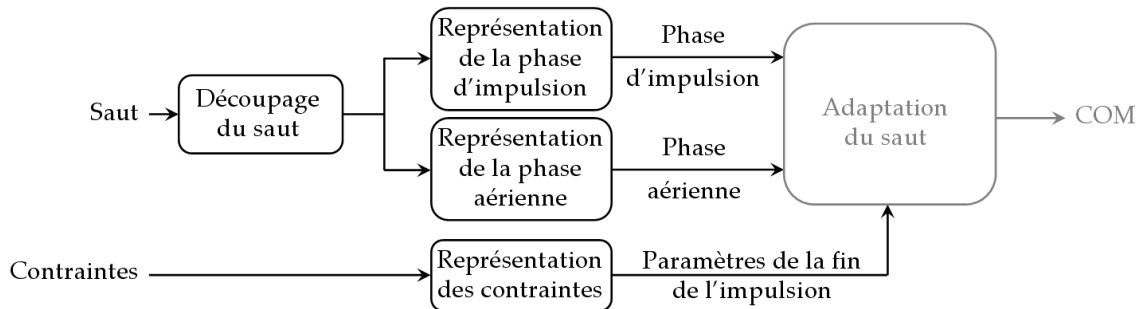


FIG. 6.3 – Schéma général de la représentation des données.

Avant tout, il convient de définir les contraintes qui peuvent être appliquées sur le saut. La première sous-section 6.2.1 définit donc ces contraintes et explique comment elles sont représentées. Ensuite, il faut représenter le saut lui-même. L'analyse présentée dans la section précédente a mis en avant les deux phases essentielles du saut : la prise d'impulsion et la phase aérienne. Aussi, avant de proposer une représentation de ces deux parties, il faut déjà être en mesure de les séparer correctement. C'est la raison pour laquelle la sous-section 6.2.2 présente une méthode pour effectuer cette séparation. Une représentation de chacune des phases est alors proposée dans la sous-section 6.2.3.

### 6.2.1 Représentation des contraintes

La section précédente a montré que les contraintes sont toujours appliquées sur la phase aérienne. Elles peuvent être de différentes natures comme sauter plus haut, aller plus loin ou encore arriver à une position donnée à la fin du saut. En fait, toutes les contraintes qui permettent de définir la trajectoire du centre de masse pendant la phase aérienne peuvent être utilisées. Ainsi, deux niveaux de spécification de contraintes peuvent être définis : le niveau le plus bas qui permet d'accéder à tous les paramètres du saut : le temps, la position et la vitesse du point initial  $P_3$  de la phase aérienne (cf. figure 6.2) ; et le niveau supérieur qui offre la possibilité de donner des modèles de contraintes. A l'instar des modèles paramétriques de mouvements, ces modèles de contraintes permettent de transformer des paramètres de haut niveau, tels que la hauteur maximale du saut ou bien encore sa longueur, en consignes de bas niveau : le temps, la position et la vitesse du point initial  $P_3$  de la phase aérienne. Dans l'exemple de la figure 6.4, la contrainte proposée permet de spécifier la hauteur maximale atteinte par le centre de masse pendant son saut. Cette hauteur maximale est alors convertie de manière à retrouver les paramètres du saut. Pour cet exemple, décrit ci-après, seule la modification de la vitesse du point  $P_3$  est nécessaire.

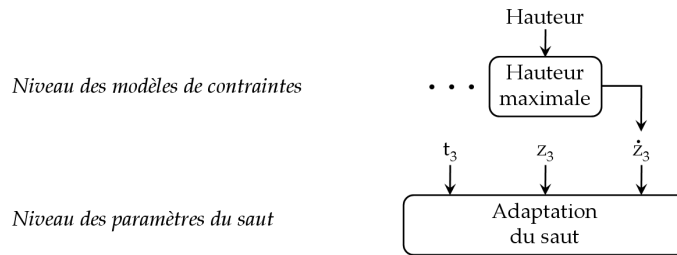


FIG. 6.4 – Deux niveaux de représentation des contraintes.

Puisque nous traitons l'exemple du saut vertical, nous détaillons seulement un exemple de contrainte : la hauteur maximale du saut. Afin de modifier cette hauteur maximale, il faut déjà déterminer l'instant pour lequel le centre de masse est le plus haut, c'est-à-dire l'instant où sa vitesse verticale est nulle :  $\dot{z} = g.t_m + \dot{z}_3 = 0$ . Cet instant est donc :  $t_m = \frac{\dot{z}_3}{g}$ . L'équation 6.2 permet alors de trouver la vitesse  $\dot{z}_3'$  nécessaire au début du saut afin de respecter la hauteur  $h$  demandée :

$$\dot{z}_3' = \frac{h + \frac{1}{2}gt_m^2 - z_3}{t_m}$$

Puisque chaque contrainte appliquée est convertie en paramètre du point  $P_3$ , il est important de connaître précisément ce point. La sous-section suivante propose donc de découper avec précision le saut.

### 6.2.2 Découpage précis du mouvement de saut

Dans la section 6.1, nous avons montré que la définition du point  $P_3$  est essentielle puisque la trajectoire parabolique de la phase aérienne dépend uniquement des données

liées à ce point : son temps, sa position ou sa vitesse. Il est donc important que ce point soit placé avec précision afin de retrouver les vraies informations du mouvement de référence.

Il est toutefois facile pour l'utilisateur d'estimer ce point  $P_3$ . En effet, c'est le moment où le personnage décolle les pieds du sol. Cependant, il est très difficile de l'estimer avec précision surtout sur des mouvements capturés ou créés avec une fréquence élevée. Aussi, nous proposons une méthode pour affiner cette position. Pour cela, une fenêtre de points autour de celui proposé par l'utilisateur va être étudiée. Dans la figure 6.5, cette fenêtre contient les trois points qui sont modélisés par des cercles. Le nombre de points pris en compte n'est pas fondamental pour le bon fonctionnement du découpage.

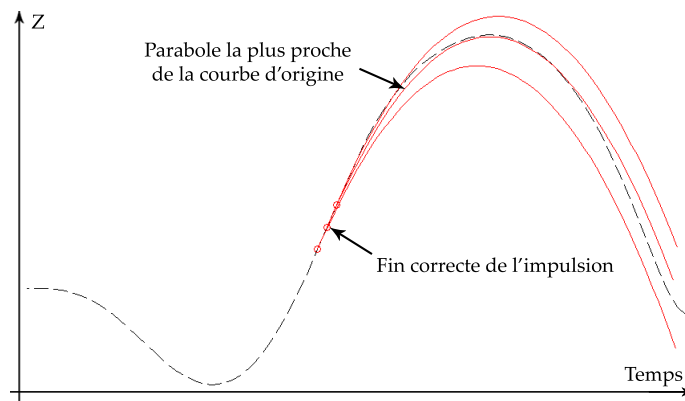


FIG. 6.5 – Adaptation du saut à différentes vitesses d'impulsion.

La position et la vitesse de chacun des points de la fenêtre traitée sont appliquées à l'équation 6.2. La parabole obtenue est donc la trajectoire du centre de masse lorsque le point traité est la fin de l'impulsion. Le véritable point  $P_3$  est celui qui minimise l'erreur entre cette parabole et la trajectoire de référence du centre de masse (cf. figure 6.5). Puisque ces courbes ont la même forme générale et passent par le même point, une méthode simple de calcul de cette erreur peut être utilisée. Dans nos travaux, nous avons opté pour la méthode des moindres carrés. Dans la figure 6.5, le point choisi est celui du milieu parmi les trois sélectionnés.

### 6.2.3 Représentation des phases

La trajectoire du centre de masse pendant la **phase aérienne** suit une parabole (cf. équation 6.2). Elle est donc naturellement représentée par le polynôme de degré 2 qui correspond à cette équation.

La prise d'impulsion est plus complexe. Elle contient deux parties : la phase d'amortissement et la phase de poussée (cf. figure 6.2). Puisque ces deux parties sont essentielles à la prise d'impulsion, il faut donc les conserver. Nous proposons donc d'identifier et de contrôler directement le point  $P_2$  qui est à l'intersection de ces deux phases. Ainsi, en conservant les caractéristiques de ce point, c'est-à-dire une dérivée nulle et un temps  $t_2$

entre  $t_1$  et  $t_3$ , nous sommes assurés de conserver la forme de l'impulsion.

Afin d'avoir une grande précision dans la modélisation de la courbe d'impulsion, nous considérons les valeurs de temps, position, vitesse et accélération au début et à la fin de l'impulsion :  $P_1(t_1, z_1, \dot{z}_1, \ddot{z}_1)$  et  $P_3(t_3, z_3, \dot{z}_3, \ddot{z}_3)$ . Puisque ces informations sont à ajouter au temps et à la position du point  $P_2(t_2, z_2)$ , huit paramètres sont disponibles et il est par conséquent possible d'utiliser un polynôme de degré 7 pour modéliser la courbe (la dérivée du point  $P_2$  doit rester nulle, il n'est donc pas nécessaire de la contrôler). La figure 6.6a montre la trajectoire de référence de l'impulsion en bleu et celle modélisée par le polynôme en vert. Cette modélisation permet également d'avoir une bonne approximation de la trajectoire de l'accélération du centre de masse (cf. figure 6.6b).

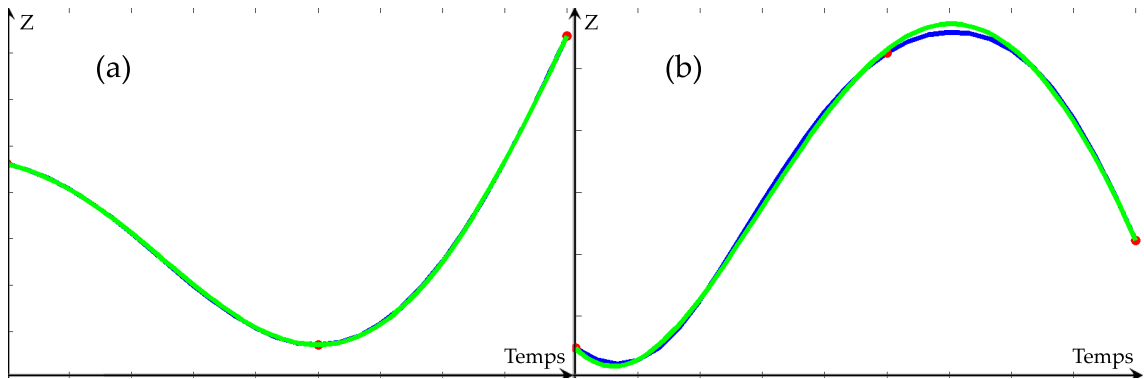


FIG. 6.6 – La trajectoire de référence est en bleu et celle modélisée en vert. (a) Représentation du saut par un polynôme de degré 7. (b) L'accélération est donc modélisée par un polynôme de degré 5.

L'intérêt de cette représentation de la trajectoire du centre de masse est qu'elle ne dépend plus que de trois points de contrôle :  $P_1$ ,  $P_2$  et  $P_3$ . Le point  $P_1$  provient soit de la trajectoire de référence soit du dernier état (position, vitesse accélération) du COM du personnage avant le saut. Le point  $P_3$ , quant à lui, est modifié par les contraintes et par l'utilisateur. Son temps est laissé sous le contrôle de l'utilisateur qui peut ainsi définir la durée de la phase d'impulsion. Puisque cette durée peut varier, le temps du point  $P_2$  est normalisé par rapport à la durée de l'impulsion.

Il faut dès lors trouver le point  $P_2$  qui convient. La section suivante se propose d'expliquer la méthode d'adaptation de l'impulsion à l'aide de ce point  $P_2$ .

### 6.3 Algorithme d'adaptation

Dans la section précédente, nous avons vu que la pose de contraintes implique une modification des caractéristiques du point  $P_3$ . Ce nouveau point modifié est appelé  $P'_3$ . De la même manière, le point  $P_1$  modifié est appelé  $P'_1$ . L'inconnue de ce problème d'adaptation est donc le point  $P'_2$  et en particulier, son temps  $t'_2$  et sa hauteur  $z'_2$ .

Le problème à résoudre peut donc être décomposé en deux étapes. La première consiste à trouver l'espace dans lequel ce point  $P'_2$  peut évoluer tout en respectant la dynamique et les contraintes appliquées sur le saut. La section 6.3.1 définit cet espace appelé *espace de recherche*. La seconde étape consiste à déterminer dans cet espace de recherche le point le plus approprié pour le réalisme du mouvement. Cette dernière étape est détaillée dans la section 6.3.2.

### 6.3.1 Espace de recherche

Sans aucune hypothèse, le point  $P'_2$  peut être déplacé librement dans l'intervalle de temps entre les deux points  $P'_1$  et  $P'_3$  et en Z entre  $-\infty$  à  $+\infty$ . Nous proposons néanmoins d'appliquer une règle pour limiter l'espace de recherche à des valeurs admissibles du point de vue mécanique : **l'accélération du centre de masse ne peut jamais être inférieure à la gravité  $g$** . Même si le personnage se laisse complètement tomber pendant la phase d'amortissement (période durant laquelle l'accélération est négative suivant l'axe vertical) et que la force de réaction du sol est par conséquent nulle, l'accélération de son centre de masse ne peut pas être supérieure à la gravité conformément à l'équation 6.1 puisque seul le poids s'applique alors. La seule manière d'avoir une accélération plus grande serait d'attacher les pieds du personnage au sol afin qu'il puisse tirer sur ses jambes pour augmenter encore cette accélération. Autrement, les pieds décolleraient du sol.

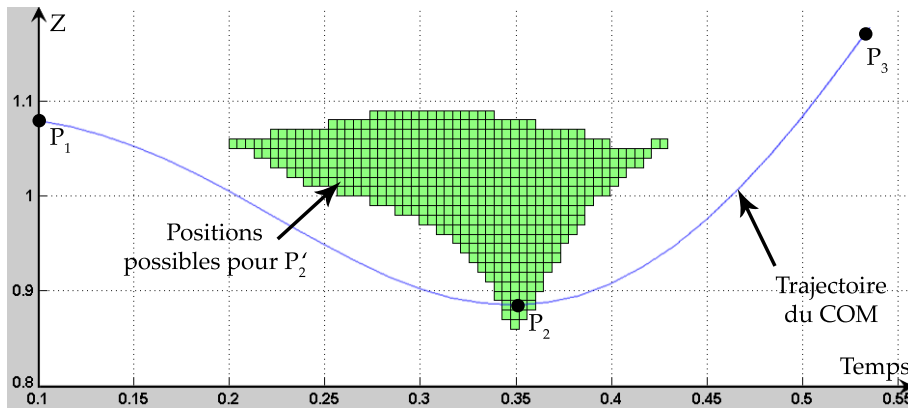


FIG. 6.7 – Positions possibles du point  $P'_2$  en ayant une accélération inférieure à la gravité.

La figure 6.7 montre l'espace possible des positions du point  $P'_2$  lorsque cette règle est appliquée. Cette fois, l'espace de recherche est parfaitement borné. Cette forme triangulaire est due aux trois zones qui ne peuvent pas être utilisées par ce point de contrôle  $P'_2$  (cf. figure 6.8) :

- la zone au-dessus du triangle : si le point de contrôle était placé là, cela signifierait que pour prendre une impulsion, le personnage commencerait par monter son centre de masse. Cela n'a évidemment aucun sens et requiert une composante d'accélération négative puis positive énorme pour respecter les conditions imposées à la fin de l'impulsion ;
- la zone en bas à gauche : cette zone définit les points pour lesquels la phase de descente du centre de masse serait accompagnée d'une accélération supérieure à la



- gravité. Nous avons vu que ce n'était pas possible sans attacher les pieds au sol ;
- et la zone en bas à droite : si le point de contrôle était placé à cet endroit, l'accélération pour atteindre le dernier point de contrôle devrait être très importante empêchant par la même de respecter les conditions finales de l'impulsion.

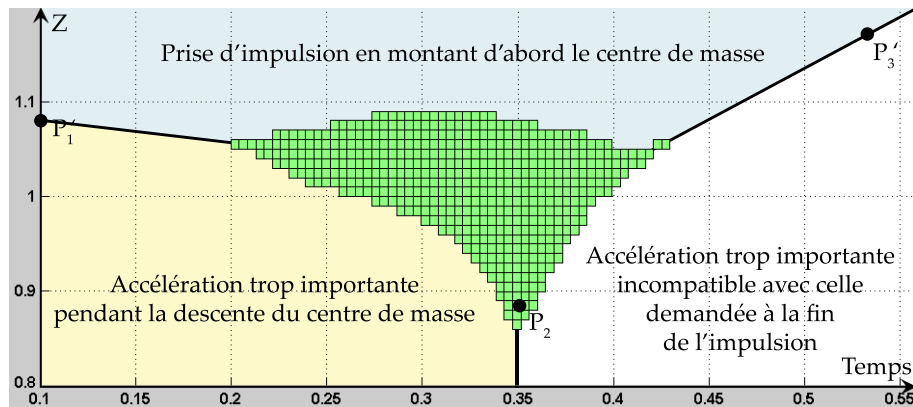


FIG. 6.8 – Explication des zones automatiquement supprimées.

L'intérêt de cet espace de recherche est qu'il peut maintenant être utilisé quelles que soient les modifications apportées sur les extrémités. Par exemple, la figure 6.9 montre le nouvel espace de recherche obtenu en changeant de façon importante la vitesse de la fin de la prise d'impulsion :  $\dot{z}_3' = \dot{z}_3 * 0.5$ .

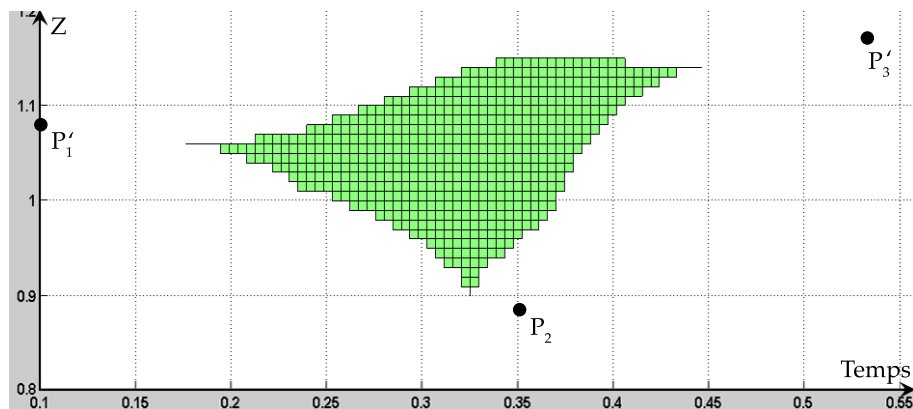


FIG. 6.9 – Positions possibles du point P<sub>2</sub>' avec une vitesse finale divisée par 2.

Par l'utilisation de cette règle sur l'accélération du centre de masse, nous avons pu définir un espace de recherche borné pour le point P<sub>2</sub>'. Il reste encore à trouver la position de ce point la plus appropriée pour respecter les contraintes de l'utilisateur.

### 6.3.2 Choix du point dans l'espace

Dans la mesure où l'espace de recherche est borné, il est à présent possible d'utiliser une loi afin de trouver le meilleur point au regard de la dynamique. Cependant, cette loi est très complexe et nécessite de tenir compte de paramètres difficilement mesurables liés au contrôle moteur (biomécanique et neurophysiologie). En effet, les capacités de chaque individu influent sur les forces qu'il peut mettre en jeu et par conséquent sur les accélérations qu'il peut engendrer. Ainsi, un athlète spécialisé dans le saut en hauteur et une personne sédentaire ne peuvent pas sauter à la même hauteur indépendamment de la technique employée.

Une étude biomécanique complète sur la comparaison des différentes lois applicables dans l'espace de recherche serait nécessaire afin d'identifier et de prendre correctement en compte tous ces paramètres (qui peuvent être stockés dans la représentation des caractéristiques individuelles (cf. section 4.1) s'ils sont spécifiques à chaque personnage). Le but de cette thèse n'est pas de trouver de telles lois mais d'offrir les méthodes et les outils pour les tester. Aussi, nous proposons une loi afin de montrer le potentiel de la méthode et l'identification de la meilleure loi est laissée en perspective de ce travail.

Nous souhaitons proposer une loi qui préserve au mieux le mouvement de référence, c'est-à-dire que le point  $P'_2$  recherché doit être le plus proche possible du point  $P_2$  dans l'espace 2D formé par le temps et l'altitude  $Z$ . De plus, il faut que les performances de calcul soient compatibles avec des applications interactives. Aussi, nous proposons d'utiliser une fonction de coût (représentant la loi choisie) qui est directement dépendante de la distance entre le point  $P'_2$  et le point  $P_2$  de la trajectoire de référence. Ainsi, le meilleur point  $P'_2$  est celui qui est le plus proche du point de référence  $P_2$  dans l'espace de recherche. La figure 6.10 montre alors le résultat de l'application de la fonction de coût dans l'espace de recherche. Plus la couleur tend vers le bleu, meilleur est le point.

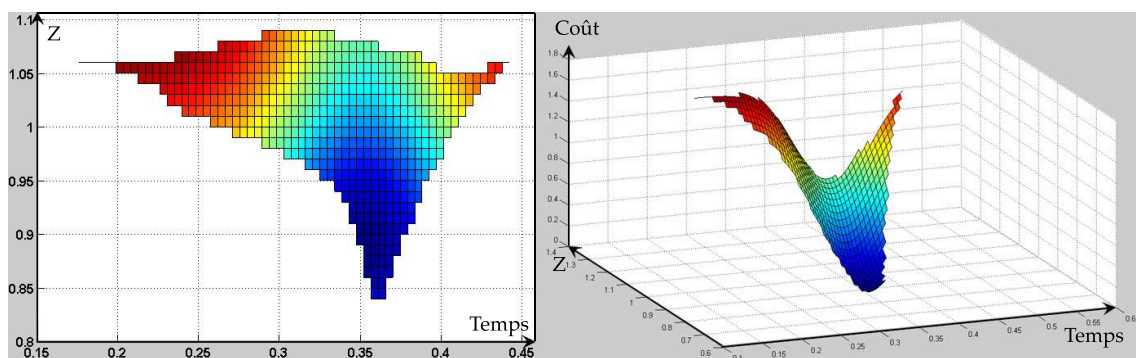


FIG. 6.10 – Utilisation de la règle de proximité du point d'origine.

La vue 3D à droite de la figure 6.10 montre la continuité de la fonction de coût et identifie clairement le point qui est le plus approprié : celui qui a la valeur de coût la plus faible (le plus bas sur la figure). Cette continuité dans l'espace de recherche permet d'utiliser une méthode d'optimisation de type descente de gradient sans risque de tomber

dans un minimum local.

En utilisant cette méthode, le point  $P'_2$  est alors trouvé ce qui définit complètement le polynôme de la trajectoire du centre de masse lors de la prise d'impulsion. Puisque que la phase aérienne est également contrainte par la dynamique, nous disposons donc d'une trajectoire du centre de masse respectant à la fois les contraintes et la dynamique pour l'ensemble du saut. Il ne reste alors plus qu'à fournir la position du centre de masse au module d'adaptation cinématique et cinétique à chaque pas de temps afin d'adapter le mouvement de saut du personnage. La section suivante donne quelques résultats de cette adaptation de sauts qui respecte la dynamique.

## 6.4 Résultats

L'exemple traité est celui du saut vertical. Aussi, nous avons modifié la vitesse à la fin de l'impulsion afin d'atteindre des hauteurs différentes. La figure 6.11 montre les différentes trajectoires du centre de masse obtenues avec la méthode d'adaptation présentée ci-dessus. Ces données sont conformes à la réalité puisqu'un personnage qui veut sauter plus haut diminue sa phase de poussée pour accélérer la vitesse à la fin de l'impulsion [Allard 00]. Afin de comparer les mouvements adaptés avec des mouvements réels, une étude statistique sera mise en place très prochainement.

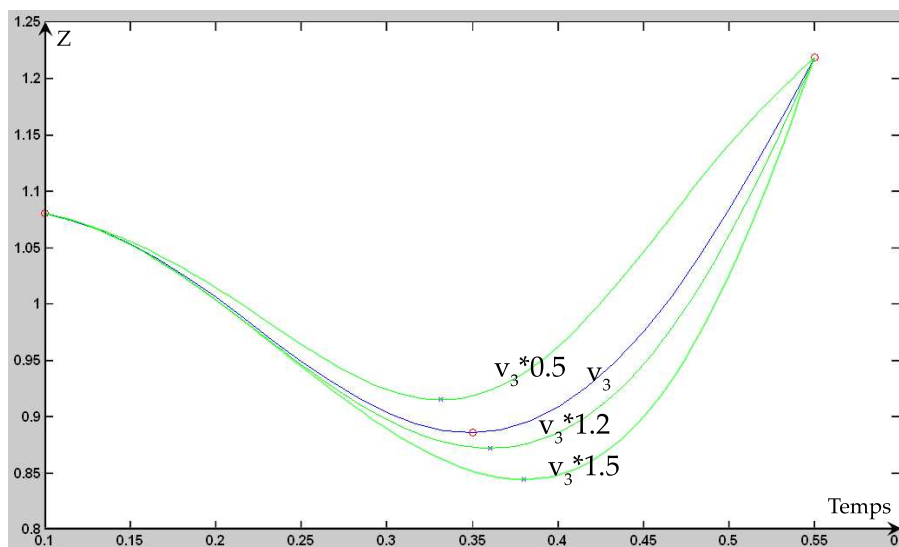


FIG. 6.11 – Adaptation du saut à différentes vitesses d'impulsion.

L'utilisation de cette trajectoire du centre de masse pour l'animation complète de personnages se fait par l'intermédiaire du module de cinétique inverse présenté dans la section 5. En effet, à chaque pas de temps, on récupère la position souhaitée du centre de masse et on la fournit à l'adaptation cinétique qui modifie la posture en conséquence. La figure 6.12 montre un exemple d'adaptation de saut. Afin que l'illustration du mouvement soit claire, un décalage temporel a été ajouté linéairement à chaque pas de temps. Ainsi,

le personnage se déplace vers la droite au fur et à mesure que le temps avance. Enfin, pour que le geste ne soit pas déformé par le changement du temps du point  $P'_2$ , un étirement du temps (“*time warping*”) est appliqué sur le mouvement de référence.

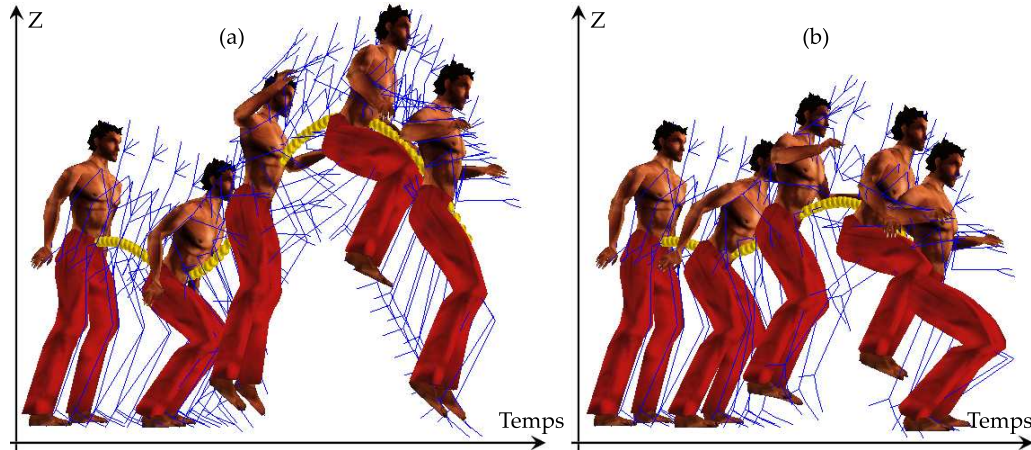


FIG. 6.12 – Exemple d’adaptation de saut. (a) Saut de référence. (b) La hauteur maximale du saut est extrêmement diminuée.

## 6.5 Discussion

Dans ce chapitre, nous avons proposé une **méthode d’adaptation des sauts qui respecte la dynamique**. Contrairement aux deux adaptations cinématique et cinétique présentées dans les chapitres 4 et 5, ce module de respect de la dynamique nécessite la connaissance du mouvement à adapter dans son ensemble. Cela signifie que dans le cadre de notre chaîne d’animation, un seul mouvement peut être utilisé dans l’immédiat. Ainsi, les postures fournies par le mélange des mouvements sont celles du mouvement de référence : le saut en cours d’adaptation. Le stockage ainsi que le mélange des informations nécessaires à ce module de respect de la dynamique est une perspective de ce travail.

Ce module de respect de la dynamique s’appuie sur les adaptations cinématiques et cinétiques présentées précédemment. En effet, il adapte la trajectoire du centre de masse du personnage afin de vérifier les contraintes fournies par l’utilisateur et le respect de la dynamique. Tous ces travaux ne sont donc pas des adaptations indépendantes mais bien un **ensemble cohérent d’adaptations du mouvement**.

Enfin, la méthode d’adaptation proposée peut être utilisée comme **outil de validation de loi**. En effet, elle permet de définir un espace borné dans lequel se trouve forcément le point de contrôle qui sépare les deux parties de la phase d’impulsion. Il est alors possible d’utiliser cet outil pour tester différentes lois sur l’adaptation de l’impulsion. Des travaux sont actuellement en cours pour intégrer d’autres lois fondamentales de la dynamique, comme le respect d’un moment cinétique constant en phase aérienne. L’objectif de ce travail est d’offrir une alternative aux méthodes impliquant d’optimiser des couples (à

appliquer à un modèle mécanique) ou à l'utilisation de techniques de filtre dynamique.



## Chapitre 7

# Architecture logicielle et applications

Les différents travaux de cette thèse portent sur l'adaptation de mouvements à des contraintes. Dans le chapitre 2, nous avons défini l'objectif de coupler les travaux de cette thèse avec ceux de Stéphane Ménardais [Ménardais 03] concernant le mélange et la synchronisation de mouvements, afin d'obtenir la chaîne complète d'animation présentée dans la figure 2.3. Le but de ce chapitre est donc de montrer, dans un premier temps, comment l'ensemble de ces travaux a été intégré dans une même architecture logicielle (cf. section 7.1) et ensuite de donner des exemples d'applications dans la section 7.2.

### 7.1 Architecture logicielle

La chaîne complète décrite dans la figure 2.3 a été intégrée au sein d'une architecture logicielle unique présentée sous la forme d'une bibliothèque C++ : MKM (*Manageable Kinematic Motions*) dont le site Web est présent à l'adresse [www.irisa.fr/siames/MKM](http://www.irisa.fr/siames/MKM).

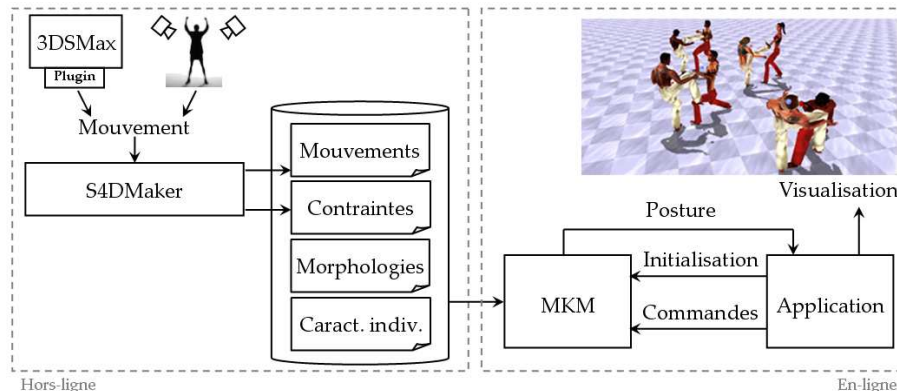


FIG. 7.1 – Chaîne d'animation complète MKM.

Cette bibliothèque a été conçue de manière à être indépendante de la plate-forme d'exécution ainsi que du moteur de rendu. Elle est composée de deux grandes parties comme le montre la figure 7.1. La première est une **partie hors-ligne**. Elle est utilisée pour la création et la correction des mouvements ainsi que pour la spécification des contraintes

intrinsèquement liées à ces mouvements. Pour effectuer ces traitements, nous avons donc créé un logiciel appelé S4DMaker. Ce logiciel dont l'interface est présentée dans la figure 7.2 permet de lire des fichiers de mouvements qui peuvent provenir du système de capture infrarouge de mouvements Vicon ou bien encore du logiciel 3DSMax. Il offre alors des services de découpage, filtrage, cyclification et réorientation des mouvements. Il permet en plus de spécifier les contraintes qui sont liées au mouvement et assure la sauvegarde cohérente de toutes ces informations dans une base de données unique.

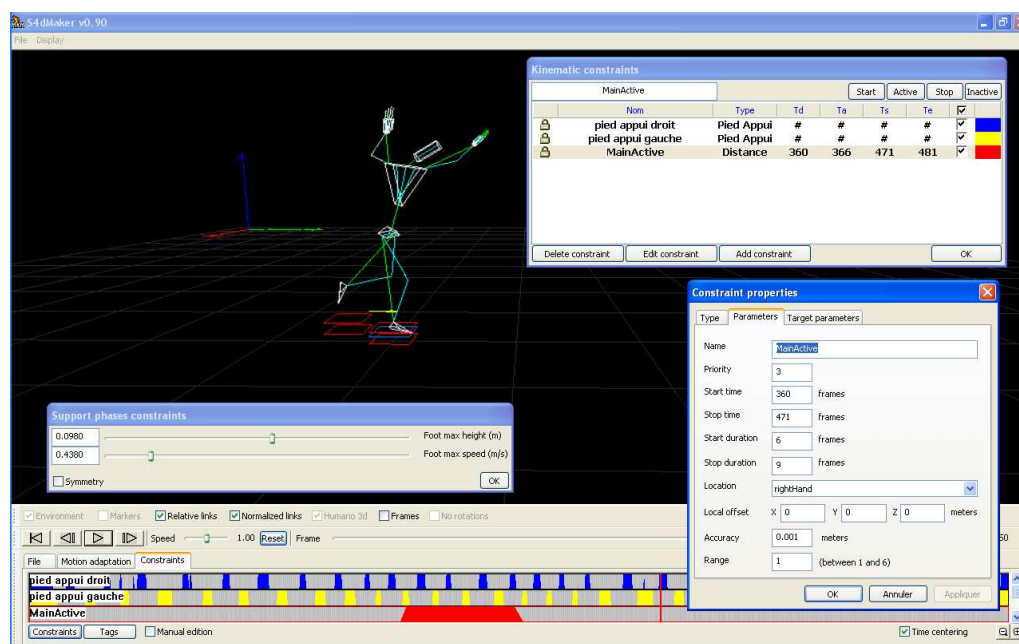


FIG. 7.2 – Interface du logiciel S4DMaker pour le traitement des mouvements et des contraintes.

La **partie en-ligne** contient le moteur d'animation temps-réel de MKM. C'est la partie qui est contrôlée par l'application afin d'être utilisée dans le cadre d'animations comportementales, dans le domaine de la réalité virtuelle ou encore dans des applications interactives. Les démonstrations présentées dans la section suivante 7.2 en font partie.

## 7.2 Applications

Afin d'illustrer le potentiel de ce moteur d'animation MKM, plusieurs applications sont montrées dans les sous-sections suivantes. Elles sont décomposées en deux grandes familles : celles qui contrôlent directement les mouvements et les contraintes, et celles qui utilisent des modèles de mouvements créés à partir des travaux de cette thèse.

### 7.2.1 Contrôle direct des contraintes

L'application illustrée par la figure 7.3 montre 40 personnages de morphologies différentes contrôlés interactivement par l'utilisateur à l'aide de deux contraintes placées sur les poignets. Deux autres contraintes sont appliquées afin que les pieds restent en contact avec



le sol. Les deux contraintes des poignets de tous les personnages sont alors interactivement modifiées par l'utilisateur à l'aide de la souris. L'algorithme d'adaptation cinématique et cinétique assure alors le respect des contraintes tout en imposant que le centre de masse reste à la verticale de sa position d'origine (assurant l'équilibre en quasi-statique). La vidéo de cette application peut être trouvée à l'adresse : [www.irisa.fr/siames/Richard.Kulpa/These/Control40p.avi](http://www.irisa.fr/siames/Richard.Kulpa/These/Control40p.avi).



FIG. 7.3 – Contrôle interactif de 40 personnages avec adaptation cinématique et cinétique.

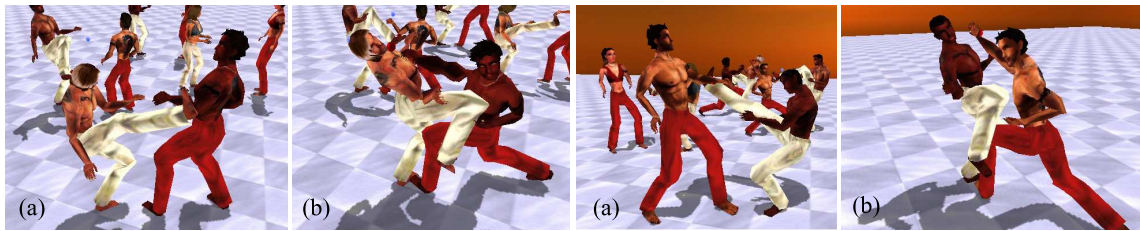


FIG. 7.4 – Cours de karaté entre deux équipes de quatre personnages.

Les contraintes peuvent également être spécifiées par l'intermédiaire d'un script. Cela permet par exemple de gérer des scénarios. La figure 7.4 montre un cours de karaté suivi par deux équipes : les karatékas en pantalon rouge contre ceux en pantalon blanc. Dans ce scénario, les attaquants en blanc donnent en premier un coup de pied paré par un point de l'avant-bras de l'adversaire, comme illustré dans la partie (a). Ensuite, ce sont les attaquants en rouge qui tentent de donner un coup de point à l'adversaire. Ce dernier l'évite d'un mouvement de buste et de tête. Bien évidemment, chaque personnage possède ses propres caractéristiques morphologiques. Cette démonstration montre l'utilisation des adaptations de contraintes dans un contexte où l'interaction entre les personnages entraîne une évolution des contraintes en temps-réel. La vidéo de cette application peut être trouvée à l'adresse : [www.irisa.fr/siames/Richard.Kulpa/These/Karate.avi](http://www.irisa.fr/siames/Richard.Kulpa/These/Karate.avi).

Puisque l'algorithme de cinétique inverse utilise les masses des segments du corps humain, il est possible de les augmenter pour simuler la présence d'un objet dans la main par exemple. Dans l'exemple illustré par la figure 7.5, le personnage est placé sur un sol

qui bouge en temps-réel. De plus, deux contraintes sont placées sur ses pieds afin qu'il conserve son appui bipodal. Enfin, deux contraintes sont appliquées sur ses poignets afin d'adapter sa posture. Dans la première image (a), les contraintes placées sur les poignets sont de masse nulle. Le personnage respecte les contraintes tout en conservant son centre de masse à la verticale de la position d'origine assurant ici l'équilibre quasi-statique. Les deux sphères rouge et bleue près de son bassin représentent respectivement le centre de masse que l'on souhaite respecter et celui de la posture adaptée. Dans la partie (b), les contraintes sont associées à une masse de 5Kg et dans l'image (c), à 10Kg. Comme tous les membres sont contraints, le personnage est obligé de reculer le bassin pour conserver son centre de masse sur la même droite verticale. La vidéo de cette application peut être trouvée à l'adresse : [www.irisa.fr/siames/Richard.Kulpa/These/Masses.avi](http://www.irisa.fr/siames/Richard.Kulpa/These/Masses.avi).

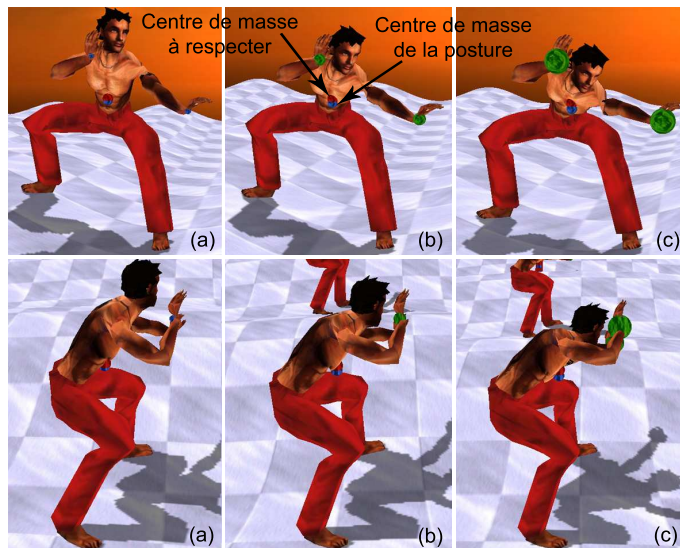


FIG. 7.5 – Ajout de masses supplémentaires sur le personnage.

### 7.2.2 Contrôle par modèles paramétriques de mouvement

Comme leur nom l'indique, les animations comportementales offrent une couche logicielle qui permet de donner un caractère autonome aux personnages. Ils peuvent ainsi prendre leurs propres décisions. Une fois que les actions à réaliser sont déterminées, il reste à contrôler les personnages afin qu'ils les exécutent. Les modèles paramétriques de mouvement permettent d'avoir une couche intermédiaire d'abstraction du contrôle du personnage. Par exemple, au lieu d'adapter un mouvement d'ouverture de porte en modifiant les trajectoires articulaires du personnage, il est plus facile d'utiliser des paramètres comme la position de la poignée de la porte et la durée du geste. Le modèle paramétrique de mouvement se charge alors de prendre en compte ces paramètres et de les transformer en réelles actions du personnage. Dans cette section, nous proposons trois modèles basés sur l'algorithme d'adaptation de mouvements.

Le premier modèle présenté ici concerne la **préhension** (cf. figure 7.6). Ce modèle a

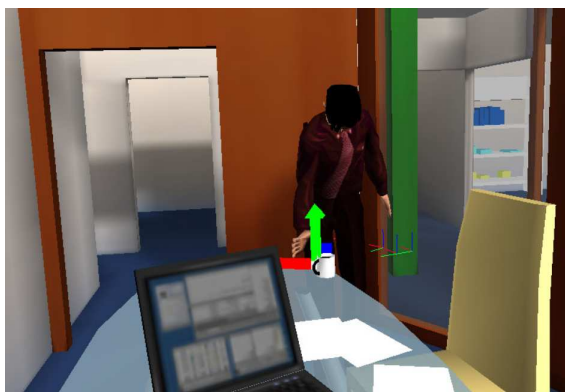


FIG. 7.6 – Modèle de préhension.

été développé dans le cadre du projet RIAM AVA-Motion. À partir de la position de la cible, de la direction, de l'orientation et de la vitesse d'arrivée de la main sur cette cible, des contraintes sont automatiquement ajoutées. Le système détermine la trajectoire que la main doit suivre au cours du temps par l'ajout d'une spline sur la trajectoire d'origine. Cette trajectoire est mise à jour à chaque pas de temps afin de prendre en compte les éventuels changements de position de la cible ou du personnage. Ensuite, la contrainte est automatiquement respectée grâce à l'algorithme d'adaptation des contraintes.

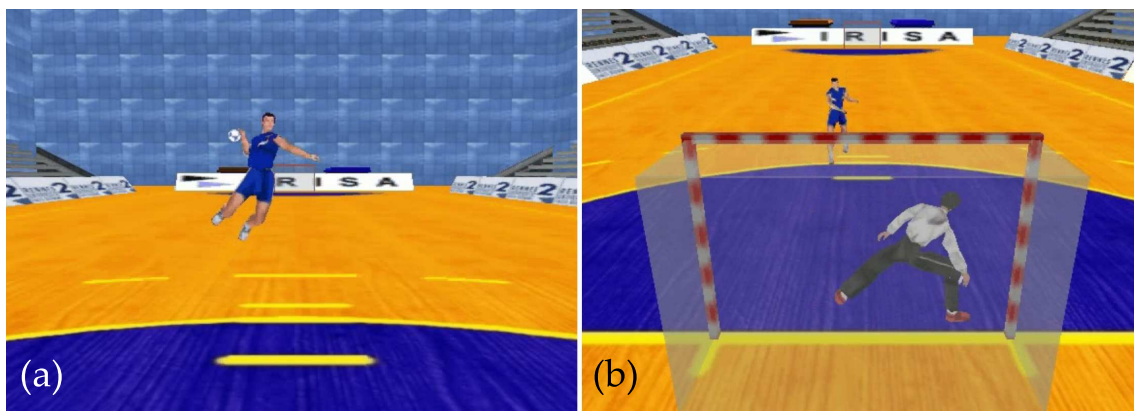


FIG. 7.7 – Modèle de tir au handball.

Le second modèle concerne **le tir au handball**. En effet, nous avons mené une étude visant à déterminer les paramètres que prend en compte un gardien de but afin d'effectuer son geste d'arrêt. Il est très difficile pour un tireur de reproduire deux fois le même tir et il lui est impossible de ne modifier avec précision qu'un seul paramètre du tir à la fois. Par exemple, il ne peut pas contrôler précisément la hauteur de sa main au moment du lâcher de balle. Une analyse biomécanique a été menée par une doctorante du LPBEM, Laëtitia Fradet, afin d'identifier les grandes familles de geste ainsi que les opérateurs de modification à appliquer [Fradet 03]. Nous avons alors créé un modèle de tir au handball

afin de ne modifier qu'un seul paramètre à la fois. Les gardiens étudiés ont ainsi été plongés dans un environnement virtuel avec vision stéréoscopique et ont été soumis à des tirs par des attaquants virtuels [Bideau 03b, Bideau 04]. Des analyses statistiques ont alors été réalisées par un autre doctorant du LPBEM, Benoît Bideau, sur les réactions du gardien de handball [Bideau 03a]. Les résultats ont montré que les gardiens réagissaient de la même manière aux tirs dans l'environnement virtuel que dans le réel. La figure 7.7a montre l'environnement virtuel vu par le gardien et la figure 7.7b montre une vue utilisée pour les entraîneurs. La vidéo de cette application peut être trouvée à l'adresse : [www.irisa.fr/siames/Richard.Kulpa/These/handball.avi](http://www.irisa.fr/siames/Richard.Kulpa/These/handball.avi).

Le dernier modèle proposé permet de contrôler l'orientation de n'importe quel mouvement de déplacement. Dans la démonstration présentée dans la figure 7.8, le mouvement utilisé est celui d'une personne ivre. Ce modèle prend en entrée une contrainte de destination et la transforme en commande de contrôle du personnage. Cette commande gère l'orientation du personnage au fur et à mesure de son déplacement. Elle s'appuie sur l'adaptation au sol présentée dans la section 4.2.4 ainsi que sur l'utilisation des contraintes de contact des pieds sur les empreintes calculées. Dans cet exemple, la barre verticale est l'objectif que doit atteindre le personnage. L'utilisateur contrôle alors interactivement cette barre afin de perturber le personnage ivre. L'intérêt d'un tel modèle de mouvement est qu'il fonctionne avec tous les mouvements qui possèdent une succession d'appuis différents. La vidéo de cette application peut être trouvée à l'adresse : [www.irisa.fr/siames/Richard.Kulpa/These/drunkMan.avi](http://www.irisa.fr/siames/Richard.Kulpa/These/drunkMan.avi).



FIG. 7.8 – Modèle générique d'adaptation de l'orientation d'un mouvement.

### 7.3 Valorisation

La chaîne complète d'adaptation et d'animation de personnages MKM a déjà fait l'objet de deux dépôts APP (Agence de Protection des Programmes) : IDDN.FR.001.290016.002.-S.A.2003.000.41200. Elle a été développée dans le cadre de deux projets RIAM :

- *AVA-Motion* dont le but était l'utilisation de l'animation réaliste de personnages en temps-réel pour les fictions interactives. Les modèles de mouvement ont notamment été développés dans le cadre de ce projet. Les partenaires de ce projet étaient les sociétés Kineo et Dæsign et les laboratoires SIAMES de l'IRISA et LPBEM (Laboratoire de Physiologie et de Biomécanique de l'Exercice Musculaire) de l'Université Rennes 2.

- *Semocap* dont le but était la réalisation d'un système de capture de mouvements économique à base de caméscopes grand public. L'utilisation des contraintes était alors un élément primordial afin de pouvoir corriger les erreurs faites par ce système de capture. Les partenaires de ce projet étaient les sociétés Artefacto et Asica et les laboratoires LPBEM de l'Université Rennes 2 et le projet MOVI de l'INRIA Rhône-Alpes.

Enfin, MKM a fait partie des logiciels présents sur le stand INRIA lors de l'exhibition de la conférence SIGGRAPH 2005. Il est actuellement en cours d'évaluation par deux sociétés de jeu vidéo dont une de réputation internationale.



# Conclusion et perspectives

Les travaux présentés dans cette thèse concernent l'**adaptation de mouvements d'humanoïdes synthétiques** à des contraintes en respectant la cinématique, la cinétique et la dynamique. Ces adaptations peuvent être réalisées sur des dizaines voire des centaines de personnages en temps-réel sur un ordinateur "grand public". Elles peuvent de plus être personnalisées pour chaque humanoïde.

Le contrôle de ces adaptations est réalisé par l'intermédiaire d'un ensemble de contraintes géométriques. Nous avons ainsi proposé une **représentation de ces contraintes** à deux niveaux :

- les contraintes manipulées par l'utilisateur (représentation externe) ;
- et une représentation interne dédiée à l'algorithme d'adaptation.

Il est ainsi possible d'ajouter de nouvelles contraintes (dans la représentation externe) simplement en leur associant un opérateur de transformation vers la représentation interne. Leur prise en compte est automatique puisque l'algorithme est indépendant de la représentation fournie à l'utilisateur.

Les algorithmes d'adaptation proposés respectent au mieux les contraintes imposées par l'utilisateur. Le réalisme des animations obtenues dépend alors, en partie, de la manière dont celui-ci les spécifie et les contrôle. En perspective, il serait donc intéressant de détecter automatiquement ces contraintes dans les captures de mouvements afin d'alléger le travail de spécification. De plus, une communication avec un environnement informé pourrait permettre d'ajouter et de supprimer en temps-réel des contraintes dues à la proximité d'un objet, par exemple.

Il serait enfin intéressant d'ajouter une nouvelle famille de contraintes qui ne peuvent pas se ramener uniquement à un problème géométrique comme, par exemple, la minimisation d'énergie ou la réaction à une force (qui intervient au niveau des accélérations).

Afin d'avoir une personnalisation et une crédibilité des adaptations de mouvements, nous avons défini une **représentation des personnages en trois niveaux** :

- la structure du squelette qui est commune à tous les humanoïdes ;
- les groupes qui divisent le squelette. Ils sont utilisés pour optimiser les algorithmes de cinématique et de cinétique inverse ;
- et enfin les caractéristiques individuelles des personnages qui définissent par exemple la morphologie et les masses corporelles.

Cette représentation pourrait être enrichie pour gérer l'expressivité du personnage. Les données à rajouter pourraient ainsi être partagées avec un module comportemental capable de les interpréter. Un grand nombre de critères interviennent sur la génération du mou-

vement : morphologiques, d'apprentissage, sociaux, médicaux (comme la prise en compte d'une pathologie)... Ces caractéristiques pourraient aussi venir s'ajouter à celles proposées dans ce mémoire. La difficulté revient encore une fois à proposer des méthodes de calcul capables d'en tenir compte. Par exemple, un doctorant du LPBEM (Laboratoire de Physiologie et de Biomécanique de l'Exercice Musculaire) de l'Université Rennes 2, Nicolas Fusco, a montré qu'il était possible de reproduire la cinématique d'une marche d'hémiplégique en jouant principalement sur les valeurs des butées articulaires [Fusco 05].

La représentation normalisée qui avait été initialement proposée par Stéphane Ménardais [Ménardais 03] ne permet pas de modéliser les mains et le visage. Cette représentation est en cours d'évolution afin de prendre en compte ces autres parties du corps. L'ajout du visage dans MKM fait notamment l'objet d'une collaboration avec Catherine Pelachaud de l'IUT de Montreuil. Les algorithmes présentés dans ce mémoire doivent eux-aussi être adaptés pour en tenir compte.

L'adaptation cinématique proposée dans ce mémoire est un algorithme de **cinématique inverse** qui permet de modifier le moins possible la posture de référence du personnage synthétique afin de vérifier les contraintes. Cette adaptation des postures permet d'animer les personnages sans connaissance à l'avance de l'évolution du mouvement réalisé ou de l'environnement. Cette utilisation directe des adaptations posturales à chaque pas de temps permet de répondre à des contraintes d'applications interactives. L'humanoïde piloté de cette manière peut aussi bien interagir avec un utilisateur qu'avec d'autres personnages synthétiques. Cependant, cette adaptation peut également être utilisée lorsque la connaissance du mouvement complet est disponible à l'avance. En effet, les techniques d'adaptation de mouvements présentées dans la littérature (comme par exemple les cartes de déplacement présentés dans la section 1.3.2.2) peuvent utiliser directement les travaux décrits dans ce mémoire. Contrairement aux autres méthodes utilisées dans la littérature, l'approche présentée dans ce mémoire offre le double avantage d'avoir un temps de calcul très court et de tenir compte de la position du centre de masse pour améliorer le réalisme.

Cette adaptation cinématique utilise un paramètre qui permet d'avoir un compromis entre le respect des contraintes à tout prix et le minimum de modification de la posture de référence. Une perspective intéressante serait de déterminer automatiquement ce paramètre en fonction du type de mouvement et des caractéristiques individuelles des personnages. Des analyses statistiques sont donc nécessaires pour déterminer une valeur moyenne de ce paramètre ainsi que la manière dont il évolue en fonction de divers critères tels que la fatigue, etc. Ces paramètres pourront alors être intégrés à la représentation des caractéristiques individuelles.

Au-delà de ce paramètre, il serait intéressant d'intégrer des lois de confort qui pourraient influencer l'adaptation. De manière générale, la prise en compte de lois plus génériques est une perspective très intéressante. Dans le domaine des APS (Activités Physiques et Sportives) par exemple, il serait possible de comparer plusieurs lois afin d'évaluer les différences de performance du geste sportif.

Enfin, une collaboration avec Lionel Reveret de l'équipe Evasion de l'INRIA Rhone-Alpes vient d'être initiée pour étendre ces techniques d'adaptation pour les utiliser sur les animaux, et en particulier les quadrupèdes.



Un algorithme de **cinétique inverse** est également proposé dans ce mémoire. Il permet d'adapter la posture du personnage tout en respectant une position du centre de masse imposée par l'utilisateur. Le réalisme de l'adaptation dépend donc en partie du contrôle de ce centre de masse. Nous proposons par défaut d'utiliser la position du centre de masse du mouvement de référence. Il serait peut-être intéressant de recalculer une trajectoire plausible de ce centre de masse en fonction du problème à résoudre, comme le suggère [le Callennec 04]. C'est en partie ce que le module de dynamique de ce mémoire effectue.

Une autre extension intéressante de ce travail serait d'intégrer la prise en compte du ZMP (cf section 1.3.2.4) pour tenir compte des aspects dynamiques dans l'équilibre. Cependant, sa gestion nécessite la connaissance des points d'application des forces, de leur intensité et de leur direction, afin de pouvoir calculer les moments de forces. Ces données doivent donc être rajoutées dans la représentation des contraintes liées au mouvement.

Nous proposons également dans cette thèse un algorithme qui permet de prendre en compte **simultanément la cinématique et la cinétique** tout en préservant des performances suffisantes pour répondre au cahier des charges. Cet algorithme permet d'adapter jusqu'à 280 personnages (à 25Hz sur un ordinateur équipé d'un Pentium4 2.8GHz et de 512Mo de mémoire vive) avec deux contraintes placées sur les poignets et une sur la cheville. Dans cet algorithme, nous avons choisi de respecter la position du centre de masse, comme si cette contrainte avait une priorité plus élevée que les autres. Dans certains cas, on pourrait toutefois tolérer un léger décalage du centre de masse, à condition que les conditions d'équilibre soient toujours respectées. Cela reviendrait à intégrer dans l'algorithme une gestion de priorités comme on pourrait la trouver dans les approches avec des Jacobiens [Baerlocher 00]. Le plus gros problème consiste à gérer les priorités entre cinématique et cinétique inverse, ce qui impliquerait certainement de revoir l'algorithme.

Enfin, la section 6 a montré que grâce au module de cinétique inverse, il était possible d'avoir indirectement un **contrôle mécanique du mouvement**, même s'il se fonde uniquement sur le système "centre de masse". Cette utilisation d'un modèle simplifié avait déjà été proposée dans la littérature [Shin 03]. L'approche présentée dans ce mémoire propose des premiers pas vers un système inspiré de celui que nous utilisons pour la cinématique inverse. En effet, en plus des contraintes géométriques, chaque mouvement peut être associé à des contraintes mécaniques intrinsèques. Dans ce mémoire, nous avons développé l'exemple du saut qui comprend une phase d'amortissement, de poussée et de vol. Pendant la phase de vol, nous appliquons les mêmes contraintes que [Shin 03]. En revanche, pendant l'appui, il est primordial de respecter la succession d'amortissements et de poussées qui caractérise la prise d'impulsion avant un saut. A plus long terme, l'idée est de généraliser cette approche en définissant hors-ligne les contraintes mécaniques propres à tout mouvement. Ces contraintes mécaniques spécifiques (par exemple la succession des phases pour un saut ou d'autres lois biomécaniques) viendraient s'ajouter aux lois générales de la dynamique : accélération égale à  $g$  pendant une chute, conservation du moment cinétique en phase aérienne...

Les trois adaptations de mouvements présentées dans ce mémoire forment ainsi un ensemble cohérent. En effet, il est possible d'utiliser les adaptations cinématiques et ci-

nétiques de manière isolées mais elles tirent toute leur force de leur combinaison. De la même manière, l'adaptation dynamique utilise pleinement les fonctionnalités offertes par la cinétique inverse pour résoudre efficacement ces contraintes mécaniques. Cette succession d'adaptations est particulièrement bien adaptée pour gérer des **niveaux de détail des adaptations**. Ainsi, suivant que le personnage est loin de la caméra ou non, il est possible d'utiliser au choix : aucune adaptation, une adaptation cinématique, une adaptation cinématique et cinétique ou enfin les trois adaptations.

Ces travaux d'adaptation peuvent servir de support à la **création de modèles de mouvements**. Dans les résultats de cette thèse, trois modèles sont d'ailleurs proposés qui peuvent être classés en deux catégories : les modèles dédiés à un mouvement en particulier (la préhension ou le tir au handball par exemple) et les modèles qui s'appliquent à tous les mouvements (contrôleur générique de déplacement décrit dans la section 7.2.2). Une perspective intéressante serait de proposer un générateur quasi-automatique de modèles de mouvement. Un langage pourrait alors décrire les fonctionnalités nécessaires à mettre en œuvre pour gérer complètement ce modèle.

A plus courte échéance, les modules d'adaptations présentés dans cette thèse pourront être utilisés de manière à améliorer la communication entre la couche comportementale et le moteur d'animation des personnages. En effet, afin de gérer correctement l'anticipation, les couches décisionnelles ont besoin d'informations sur l'état du personnage à l'instant présent mais également dans un avenir proche pour faire un choix parmi plusieurs stratégies d'action. Ces informations doivent donc remonter de la couche de contrôle du personnage vers celle du comportement (partie (a) de la figure 7.9). Par exemple, considérons le cas d'un personnage autonome qui marche dans une pièce et qui doit s'arrêter à un endroit précis. Afin d'avoir une animation réaliste, une anticipation des cycles de marche à venir est nécessaire pour diminuer progressivement la longueur des pas, par exemple. Le contrôle du personnage doit donc indiquer le nombre de cycles restants à faire avant la destination. La couche comportementale peut alors prendre la décision de modifier la longueur des pas par l'intermédiaire de commandes qu'elle donne au contrôle du personnage (partie (b) de la figure 7.9).

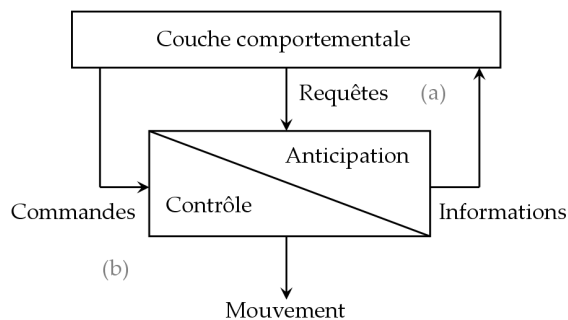


FIG. 7.9 – Deux modes de communication entre le comportement et le contrôle. (a) Le comportement lance des requêtes pour savoir la faisabilité de telle ou telle action ou l'état actuel ou futur du personnage. (b) Lorsque les informations recueillies sont suffisantes, des commandes sont utilisées pour contrôler l'animation.

Cette communication pourrait être étendue à la planification de chemins. Il serait alors

possible de tester plusieurs solutions en simulant rapidement les mouvements nécessaires et en retournant une valeur de coût. Celui-ci pourrait être un critère de confort ou de réalisme (comme le propose [Arikan 05] via un module appelé “oracle”).

Dans la revue de littérature, nous avons vu que les graphes de mouvements représentent une technique très utilisée en animation pour combiner des mouvements afin d’en créer de nouveaux. Toutefois, pour obtenir une adaptation suffisamment précise, la base de données de mouvements doit être très importante. Une perspective intéressante de ce travail serait alors de coupler ces graphes avec les modèles paramétrables de mouvements. Au lieu de simplement trouver les meilleures transitions entre les postures de mouvements prédéfinis, des fonctions de coût intégrant les probabilités d’adaptation pourraient être utilisées.



## Annexe A

# Utilisation des contraintes

Afin d'illustrer l'utilisation des deux familles de contraintes (celles qui sont intrinsèquement liées au mouvement et celles qui sont créées en temps-réel), cette annexe propose deux exemples. Le premier concerne l'utilisation d'une contrainte inhérente au mouvement : une parade avec l'avant-bras. Le second exemple traite de contraintes créées et modifiées en temps-réel dans le cadre d'une application interactive.

### A.1 Contraintes intrinsèquement liées au mouvement

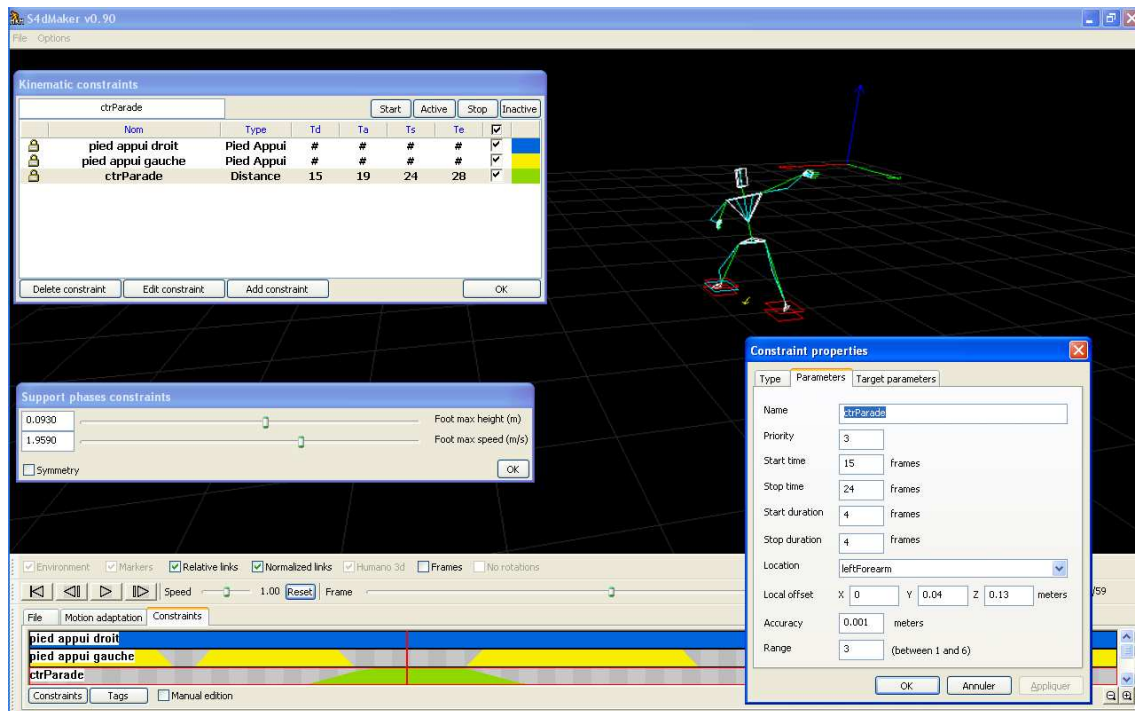
Cette section montre la spécification et l'utilisation d'une contrainte placée sur l'avant-bras dans un mouvement de parade du bras (cf. figure A.1).



FIG. A.1 – Exemple de parade d'un coup de pied avec l'avant-bras.

Dans la mesure où cette contrainte est intrinsèquement liée au mouvement, elle doit être **spécifiée hors-ligne**. A l'aide du logiciel S4DMaker (cf. figure A.2), nous créons une contrainte de position appelée *ctrParade*.

Les boîtes de dialogue s'adaptent automatiquement afin que les paramètres à remplir correspondent aux paramètres génériques et surtout spécifiques au type de contrainte utilisé, dans notre exemple, une contrainte de position.

FIG. A.2 – Ajout de la contrainte *ctrParade* dans le S4dMaker.

Au moment de la sauvegarde, cette contrainte est stockée dans un fichier XML associé à celui du mouvement :

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<ConstraintsSet xmlns="ContraintesXML">
  <Distance>
    <name>ctrParade</name>
    <priority>3</priority>
    <accuracy>0.001</accuracy>
    <start_time>0.499995</start_time>
    <stop_time>0.799992</stop_time>
    <start_duration>0.119999</start_duration>
    <stop_duration>0.122574</stop_duration>
    <range>6</range>
    <bodyPoint>
      <segment>rightSternoClavicular</segment>
      <localOffset>
        <X>0</X>
        <Y>0.04</Y>
        <Z>0.13</Z>
      </localOffset>
    </bodyPoint>
  </Distance>
</ConstraintsSet>
```

```

<specif>
  <distance>0</distance>
  <position>
    <X>0</X>
    <Y>0</Y>
    <Z>0</Z>
  </position>
</specif>
</Distance>
</ConstraintsSet>

```

La spécification est alors terminée et cette contrainte peut être gérée à chaque fois que le mouvement est **utilisé par le moteur d'animation**. Les contraintes que l'on souhaite modifier doivent alors être retrouvées parmi celles qui sont associées au mouvement. Le paramètre  $Id_i$  existe justement à cette fin. L'appel se fait par l'intermédiaire d'une fonction C++ du type :

```
C0 = getConstraint("ctrParade");
```

A chaque pas de temps, la cible associée à cette contrainte est déplacée de manière à suivre la position de la cheville de l'adversaire :

```
C0->setTargetPosition(advAnkle)
```

où *advAnkle* est la position de la cheville de l'adversaire.

Tous les paramètres inchangés conservent leurs valeurs spécifiées hors-ligne (par exemple, l'activation et le désactivation de la contrainte).

Le **résultat** est fourni par la vidéo [www.irisa.fr/siames/Richard.Kulpa/These/Karate.avi](http://www.irisa.fr/siames/Richard.Kulpa/These/Karate.avi). La figure A.3 montre une succession d'images de la parade du coup de pied :



FIG. A.3 – Séquence d'images de la parade du coup de pied.

## A.2 Contraintes temps-réel

L'exemple utilisé pour illustrer l'utilisation de contraintes spécifiées en temps-réel est une application interactive où l'utilisateur contrôle deux contraintes placées sur les poignets. Les pieds sont contraints sur le sol et le geste utilisé est un mouvement de repos.

Les contraintes appliquées au poignet ne sont pas présentes dans le mouvement. Il faut donc les créer et les modifier en temps-réel. La **création** se fait en initialisant une nouvelle

instance d'une classe de contraintes de position avec les paramètres génériques et spécifiques présentés dans la section 3.

Pour modifier les contraintes créées, il est toujours possible de les récupérer par l'intermédiaire de leur identification  $Id_i$ , cependant, dans la pratique, l'utilisateur les possède déjà puisqu'elles ont été créées directement par lui. Il peut donc **changer les paramètres** des contraintes comme il le souhaite. Dans notre exemple, la position des cibles des contraintes des poignets est modifiée en fonction des mouvements de la souris.

Dans ce cadre d'applications interactives, il n'est pas possible de spécifier à l'avance les paramètres d'activation et de désactivation des contraintes. L'utilisateur peut alors directement les activer et les désactiver par l'intermédiaire de deux méthodes C++ *start()* et *stop()*.

La séquence d'images illustrée par la figure A.4 montre différentes postures obtenues sur 40 personnages dans le cadre de cette application interactive.



FIG. A.4 – Application interactive où l'utilisateur contrôle les poignets. Les pieds sont contraints sur le sol.



## Annexe B

# Exemples de solutions analytiques

Cette annexe se propose de donner les grandes familles de solutions analytiques utilisées pour résoudre l'adaptation des membres à des contraintes. Ces familles regroupent ces solutions selon la manière dont elles sont résolues :

- **directe** : ces solutions analytiques sont utilisées pour les contraintes placées directement sur les données de la représentation normalisée du squelette. Pour les membres, ce sont les articulations des poignets et des chevilles.
- **par utilisation de rotations** : cette famille regroupe les solutions analytiques qui ne sont résolues que par l'application de rotations. Ce sont par exemple les cas où aucune contrainte n'est placée sur l'avant-bras ou le mollet. En effet, dans ces exemples, aucune modification de la longueur variable du membre n'est requise et seule des rotations doivent répondre au problème.
- ou à l'aide de **calculs géométriques plus complexes** : ce sont toutes les autres solutions qui nécessitent des adaptations plus complexes.

Les sections suivantes proposent de donner un exemple de solution analytique pour chacune de ces familles. Ces exemples utilisent les notations suivantes :

- $Racine_M$  est l'articulation racine du membre, c'est-à-dire l'épaule ou la hanche ;
- $Inter_M$  est l'articulation intermédiaire du membre, à savoir le coude ou le genou ;
- et  $Extr_M$  représente l'articulation extrémité du membre, c'est-à-dire le poignet ou la cheville.

### B.1 Directe

Ces solutions sont utilisées pour les contraintes qui sont placées sur les poignets et les chevilles. En effet, ces articulations font partie de la représentation normalisée du squelette et peuvent donc être modifiées directement. Prenons l'exemple d'une contrainte  $C_0$  placée sur le poignet et dont la cible est une position donnée  $Cible_0$ . Pour résoudre ce problème (illustré par la figure B.1), il suffit de changer le vecteur normalisé du bras de manière à placer cette contrainte sur la cible :

- Le vecteur  $Z$  du repère du membre est dirigé vers la cible  $\overrightarrow{(Racine_M, Cible_0)}$  ;
- La longueur variable du membre est égale à la norme du vecteur  $\overrightarrow{(Racine_M, Cible_0)}$  divisée par la longueur maximale du membre. Cette longueur est seuillée afin de ne pas dépasser la valeur de 1 qui correspond au bras tendu.

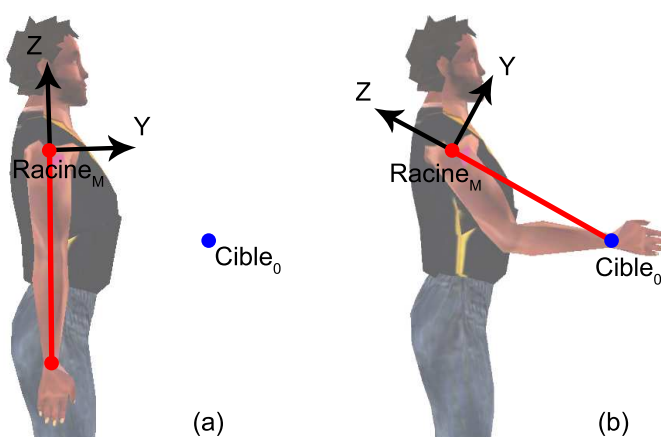


FIG. B.1 – Respect d’une contrainte sur le poignet. La partie (a) montre la posture de référence. La partie (b) illustre l’adaptation faite pour respecter la contrainte.

## B.2 Utilisation de rotations

Lorsqu’une contrainte est placée sur le bras ou la cuisse, aucune adaptation n’est nécessaire sur la longueur variable du bras. Une rotation est donc suffisante pour résoudre le problème : passer du vecteur  $\overrightarrow{(Racine_M, CP_0)}$  au vecteur  $\overrightarrow{(Racine_M, Cible_0)}$  (cf. figure B.2 pour l’exemple du bras).

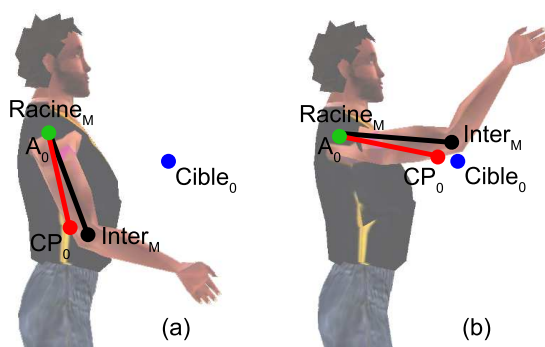


FIG. B.2 – Respect d’une contrainte sur le bras. La partie (a) est la posture de référence. La partie (b) est la posture adaptée afin de respecter au mieux la contrainte.

## B.3 Calculs géométriques plus complexes

Ces solutions sont les plus utilisées puisqu’elles permettent de résoudre le plus grand nombre de configurations. Considérons l’exemple d’une contrainte  $C_0$  placée sur l’avant-bras (cf. figure B.3). Cette contrainte est à une distance fixe du coude et du poignet puisque le décalage  $Decal_0$  est un vecteur 3D fixe dans le repère de l’avant-bras. Il est donc possible de déterminer l’une des deux positions au choix.

Nous choisissons de déterminer la meilleure position pour le coude. Ce dernier se trouve à une distance constante de l'épaule (de la longueur du bras) et de la position de la cible  $Cible_0$  (nous traitons à part le cas où la contrainte est inaccessible). Le coude se trouve donc à l'intersection des deux sphères :  $S_1$  centrée sur l'épaule  $Racine_M$  et de rayon la longueur du bras et  $S_2$  centrée sur la cible  $Cible_0$  et de rayon  $Decal_0$  (cf. partie (b) de la figure B.3). Puisque l'intersection de deux sphères est un cercle, il y a une infinité de solutions. Afin de respecter au mieux la posture de référence, nous allons choisir le point du cercle qui permet au bras de conserver au mieux cette orientation de référence (cf. partie (c) de la figure B.3). Il ne reste alors plus qu'à retrouver la position du poignet pour modifier la représentation normalisée du bras.

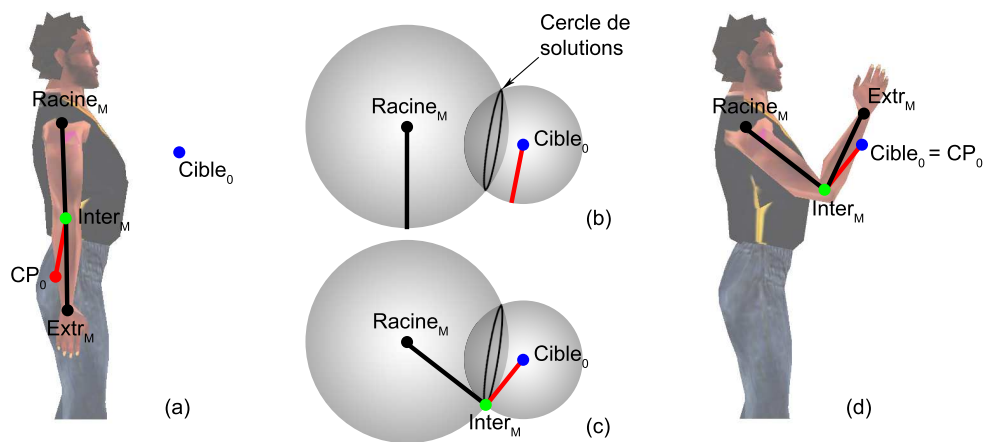


FIG. B.3 – Adaptation d'une contrainte sur l'avant-bras lorsque la cible est atteignable. La posture de référence est fournie dans la partie (a). Le coude est à une distance fixe de la contrainte et de l'épaule, donc à l'intersection de deux sphères illustrées en partie (b). Dans la partie (c), le coude est choisi parmi le cercle de solutions afin de conserver au mieux l'orientation du bras de la posture de référence. La partie (d) montre la posture adaptée.



## Annexe C

# Adaptation d'un membre pour respecter une position de COM fournie

Dans cette annexe, nous expliquons l'adaptation d'un membre de manière à placer son centre de masse à la position demandée par notre algorithme de cinétique inverse. Nous considérons ici l'exemple du bras.

### C.1 Problème

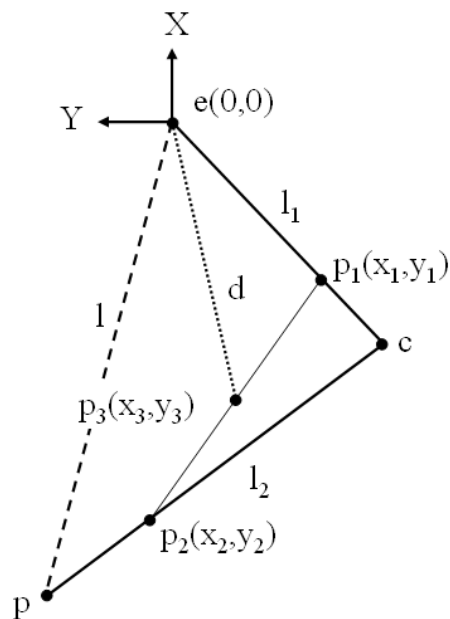


FIG. C.1 – Calcul du COM dans un segment de longueur variable.

Le problème posé est de trouver la longueur  $l$  en fonction de la distance  $d$  fournie entre l'épaule et le centre de masse imposé. En effet, la position de ce COM est connue et il faut trouver l'extension du bras qui respecte la distance entre le COM ( $p_3$ ) et l'épaule ( $e$ ). Ensuite, une rotation permet de finaliser l'adaptation.

Les paramètres suivants sont connus par la morphologie du personnage ou les tables anthropométriques :

- $l_1$  : la longueur du bras ;
- $l_2$  : la longueur de l'avant-bras ;
- $r_1$  : le ratio qui permet de trouver le COM du bras  $p_1$  sur le segment de l'épaule ( $e$ ) au coude ( $c$ ) ;
- $r_2$  : le ratio qui permet de trouver le COM de l'avant-bras  $p_2$  sur le segment du coude ( $c$ ) au poignet ( $p$ ) ;
- $r_3$  : le ratio qui permet de trouver  $p_3$  sur le segment de  $p_1$  à  $p_2$ .

Les deux premiers paramètres permettent de définir les équations suivantes :

$$X_c^2 + Y_c^2 = l_1^2 \quad (\text{C.1})$$

$$X_p^2 + Y_p^2 = l^2 \quad (\text{C.2})$$

$$(X_c - X_p)^2 + (Y_c - Y_p)^2 = l_2^2 \quad (\text{C.3})$$

$$(\text{C.4})$$

Les ratios permettent de retrouver les points  $p_1$ ,  $p_2$  et  $p_3$ . Les deux premiers ratios sont fournis par les tables anthropométriques. Le ratio  $r_3$  est considéré comme égal à  $\frac{m_2}{m_1+m_2}$  où  $m_1$  est la masse du bras et  $m_2$  celle de l'avant-bras. Le calcul du centre de masse  $p_3$  se fait alors simplement comme suit :

$$p_3 = \frac{m_1 p_1 + m_2 p_2}{m_1 + m_2}$$

$$p_3 = (1 - r_3)p_1 + r_3 p_2$$

$$p_3 = p_1 + r_3(p_2 - p_1)$$

Afin de simplifier les calculs, toutes les coordonnées sont données dans le repère de l'épaule.

Les coordonnées du premier point  $p_1$  sont :

$$X_1 = r_1(X_c - X_e) = r_1 X_c \quad (\text{C.5})$$

$$Y_1 = r_1(Y_c - Y_e) = r_1 Y_c \quad (\text{C.6})$$

Les coordonnées du point  $p_2$  sont :

$$X_2 = X_c + r_2(X_p - X_c)$$

$$Y_2 = Y_c + r_2(Y_p - Y_c)$$

soit :

$$X_2 = (1 - r_2)X_c + r_2X_p \quad (\text{C.7})$$

$$Y_2 = (1 - r_2)Y_c + r_2Y_p \quad (\text{C.8})$$

Les coordonnées du point  $p_3$  sont :

$$X_3 = X_1 + r_3(X_2 - X_1)$$

$$Y_3 = Y_1 + r_3(Y_2 - Y_1)$$

soit :

$$X_3 = (1 - r_3)X_1 + r_3X_2 \quad (\text{C.9})$$

$$Y_3 = (1 - r_3)Y_1 + r_3Y_2 \quad (\text{C.10})$$

## C.2 Calcul de $d^2$

Notre résolution se base sur des calculs de distance, nous partons donc de l'équation suivante :

$$d^2 = X_3^2 + Y_3^2 \quad (\text{C.11})$$

Avec les équations (C.9) et (C.10), nous obtenons :

$$\begin{aligned} d^2 &= (1 - r_3)^2 X_1^2 + r_3^2 X_2^2 + 2r_3(1 - r_3)X_1X_2 \\ &+ (1 - r_3)^2 Y_1^2 + r_3^2 Y_2^2 + 2r_3(1 - r_3)Y_1Y_2 \end{aligned}$$

Soit :

$$d^2 = (1 - r_3)^2(X_1^2 + Y_1^2) + r_3^2(X_2^2 + Y_2^2) + 2r_3(1 - r_3)(X_1X_2 + Y_1Y_2) \quad (\text{C.12})$$

Traitons séparément les trois composantes de cette addition tels que  $d^2 = A + B + C$ .

### C.2.1 Calcul de $A$

A partir des coordonnées du point  $p_1$ , nous avons :

$$X_1^2 + Y_1^2 = (r_1l_1)^2 = r_1^2l_1^2$$

Donc avec l'équation (C.12) :

$$A = (1 - r_3)^2 r_1^2 l_1^2 \quad (\text{C.13})$$

### C.2.2 Calcul de $B$

Calculons d'abord  $X_2^2 + Y_2^2$  avec les équations (C.7) et (C.8) :

$$\begin{aligned} X_2^2 + Y_2^2 &= (1 - r_2)^2 X_c^2 + r_2^2 X_p^2 + 2r_2(1 - r_2)X_c X_p \\ &+ (1 - r_2)^2 Y_c^2 + r_2^2 Y_p^2 + 2r_2(1 - r_2)Y_c Y_p \end{aligned}$$

En regroupant les termes membre à membre et en utilisant les équations (C.1) et (C.2), nous obtenons :

$$X_2^2 + Y_2^2 = (1 - r_2)^2 l_1^2 + r_2^2 l^2 + 2r_2(1 - r_2)(X_c X_p + Y_c Y_p)$$

Avec l'équation (C.12),  $B$  devient alors :

$$B = r_3^2 [(1 - r_2)^2 l_1^2 + r_2^2 l^2 + 2r_2(1 - r_2)(X_c X_p + Y_c Y_p)]$$

Soit :

$$B = r_3^2 (1 - r_2)^2 l_1^2 + r_2^2 r_3^2 l^2 + 2r_2 r_3^2 (1 - r_2)(X_c X_p + Y_c Y_p) \quad (\text{C.14})$$

### C.2.3 Calcul de $C$

Calculons d'abord  $X_1 X_2$  avec les équations (C.5) et (C.7) :

$$X_1 X_2 = r_1 X_c [(1 - r_2)X_c + r_2 X_p]$$

En regroupant les termes :

$$X_1 X_2 = r_1 [(1 - r_2)X_c^2 + r_2 X_c X_p]$$

En faisant de même avec  $Y_1 Y_2$ , nous avons :

$$X_1 X_2 + Y_1 Y_2 = r_1 [(1 - r_2)X_c^2 + r_2 X_c X_p + (1 - r_2)Y_c^2 + r_2 Y_c Y_p]$$

Avec l'équation (C.1) :

$$X_1 X_2 + Y_1 Y_2 = r_1 [(1 - r_2)l_1^2 + r_2(X_c X_p + Y_c Y_p)]$$

Avec l'équation (C.12),  $C$  devient alors :

$$C = 2r_1 r_3 (1 - r_3) [(1 - r_2)l_1^2 + r_2(X_c X_p + Y_c Y_p)]$$

Soit, en développant :

$$C = 2r_1 r_3 (1 - r_3)(1 - r_2)l_1^2 + 2r_1 r_2 r_3 (1 - r_3)(X_c X_p + Y_c Y_p) \quad (\text{C.15})$$



### C.2.4 Calcul intermédiaire de $d^2$

À partir des équations (C.13),(C.14) et (C.15), nous avons :

$$\begin{aligned} d^2 &= (1 - r_3)^2 r_1^2 l_1^2 \\ &+ r_3^2 (1 - r_2)^2 l_1^2 + r_2^2 r_3^2 l^2 + 2r_2 r_3^2 (1 - r_2) (X_c X_p + Y_c Y_p) \\ &+ 2r_1 r_3 (1 - r_3) (1 - r_2) l_1^2 + 2r_1 r_2 r_3 (1 - r_3) (X_c X_p + Y_c Y_p) \end{aligned}$$

En regroupant les membres, nous trouvons :

$$\begin{aligned} d^2 &= l_1^2 [(1 - r_3)^2 r_1^2 + r_3^2 (1 - r_2)^2 + 2r_1 r_3 (1 - r_3) (1 - r_2)] \\ &+ r_2^2 r_3^2 l^2 \\ &+ (X_c X_p + Y_c Y_p) [2r_2 r_3^2 (1 - r_2) + 2r_1 r_2 r_3 (1 - r_3)] \end{aligned}$$

En développant :

$$\begin{aligned} d^2 &= l_1^2 [(1 + r_3^2 - 2r_3) r_1^2 + r_3^2 (1 + r_2^2 - 2r_2) + 2r_1 r_3 (1 - r_3 - r_2 + r_2 r_3)] \\ &+ r_2^2 r_3^2 l^2 \\ &+ (X_c X_p + Y_c Y_p) (2r_2 r_3^2 - 2r_2^2 r_3^2 + 2r_1 r_2 r_3 - 2r_1 r_2 r_3^2) \end{aligned}$$

Soit :

$$\begin{aligned} d^2 &= l_1^2 (r_1^2 + r_1^2 r_3^2 - 2r_1^2 r_3 + r_3^2 + r_2^2 r_3^2 - 2r_2 r_3^2 + 2r_1 r_3 - 2r_1 r_3^2 - 2r_1 r_2 r_3 + 2r_1 r_2 r_3^2) \\ &+ r_2^2 r_3^2 l^2 \\ &+ (X_c X_p + Y_c Y_p) (2r_2 r_3^2 - 2r_2^2 r_3^2 + 2r_1 r_2 r_3 - 2r_1 r_2 r_3^2) \end{aligned}$$

Soit :

$$\begin{aligned} d^2 &= l_1^2 [(r_1 + r_3)^2 + r_1^2 r_3^2 - 2r_1^2 r_3 + r_2^2 r_3^2 - 2r_2 r_3^2 - 2r_1 r_3^2 - 2r_1 r_2 r_3 + 2r_1 r_2 r_3^2] \\ &+ r_2^2 r_3^2 l^2 \\ &+ 2(X_c X_p + Y_c Y_p) (r_2 r_3^2 - r_2^2 r_3^2 + r_1 r_2 r_3 - r_1 r_2 r_3^2) \end{aligned} \quad (\text{C.16})$$

### C.2.5 Calcul de $X_c X_p + Y_c Y_p$

En développant l'équation (C.3), nous obtenons :

$$X_p^2 + X_c^2 - 2X_p X_c + Y_p^2 + Y_c^2 - 2Y_p Y_c = l_2^2$$

Soit :

$$2(X_p X_c + Y_p Y_c) = (X_p^2 + Y_p^2) + (X_c^2 + Y_c^2) - l_2^2$$

Avec les équations (C.1) et (C.2), nous obtenons :

$$2(X_p X_c + Y_p Y_c) = l^2 + l_1^2 - l_2^2 \quad (\text{C.17})$$

### C.2.6 Calcul final de $d^2$

En prenant compte de l'équation (C.17), l'équation (C.16) devient alors :

$$\begin{aligned} d^2 &= l_1^2 [(r_1 + r_3)^2 + r_1^2 r_3^2 - 2r_1^2 r_3 + r_2^2 r_3^2 - 2r_2 r_3^2 - 2r_1 r_3^2 - 2r_1 r_2 r_3 + 2r_1 r_2 r_3^2] \\ &+ r_2^2 r_3^2 l^2 \\ &+ (l^2 + l_1^2 - l_2^2)(r_2 r_3^2 - r_2^2 r_3^2 + r_1 r_2 r_3 - r_1 r_2 r_3^2) \end{aligned}$$

En développant, nous avons :

$$\begin{aligned} d^2 &= l_1^2 [(r_1 + r_3)^2 + r_1^2 r_3^2 - 2r_1^2 r_3 - r_2 r_3^2 - 2r_1 r_3^2 - r_1 r_2 r_3 + r_1 r_2 r_3^2] \\ &+ l^2 (r_2^2 r_3^2 + r_2 r_3^2 - r_2^2 r_3^2 + r_1 r_2 r_3 - r_1 r_2 r_3^2) \\ &+ l_2^2 (r_2^2 r_3^2 - r_2 r_3^2 - r_1 r_2 r_3 + r_1 r_2 r_3^2) \end{aligned}$$

Soit :

$$\begin{aligned} d^2 &= l_1^2 [(r_1 + r_3)^2 + r_1^2 r_3^2 - 2r_1^2 r_3 - r_2 r_3^2 - 2r_1 r_3^2 - r_1 r_2 r_3 + r_1 r_2 r_3^2] \\ &+ l_2^2 (r_2^2 r_3^2 - r_2 r_3^2 - r_1 r_2 r_3 + r_1 r_2 r_3^2) \\ &+ l^2 (r_2 r_3^2 + r_1 r_2 r_3 - r_1 r_2 r_3^2) \end{aligned}$$

Pour diminuer le nombre de multiplications (pour les temps de calcul), il est possible d'écrire cette équation sous la forme :

$$\begin{aligned} d^2 &= l_1^2 [(r_1 + r_3)^2 + r_3 [r_2 (r_1 r_3 - r_1 - r_3) + r_1 (r_1 r_3 - 2(r_1 + r_3))]] \\ &+ r_2 r_3 l_2^2 [r_3 (r_2 + r_1 - 1) - r_1] \\ &+ l^2 r_2 r_3 (r_3 + r_1 - r_1 r_3) \end{aligned}$$

Cette équation est donc :

$$d^2 = F + Gl^2 \quad (\text{C.18})$$

avec  $F$  et  $G$  deux valeurs constantes telles que :

$$\begin{aligned} F &= l_1^2 [(r_1 + r_3)^2 + r_3 [r_2 (r_1 r_3 - r_1 - r_3) + r_1 (r_1 r_3 - 2(r_1 + r_3))]] \\ &+ r_2 r_3 l_2^2 [r_3 (r_2 + r_1 - 1) - r_1] \\ G &= r_2 r_3 (r_3 + r_1 - r_1 r_3) \end{aligned}$$

### C.3 Calcul final de $l$

$l$  est obtenu en vérifiant que  $G \neq 0$  :

$$l^2 = \frac{d^2 - F}{G}$$

Or  $r_1$ ,  $r_2$  et  $r_3$  appartiennent à l'intervalle  $]0,1[$ , donc pour que  $G$  soit égal à 0, il faut que :

$$\begin{aligned} r_3 + r_1 - r_1 r_3 &= 0 \\ r_3 &= \frac{-r_1}{1 - r_1} \end{aligned}$$

Puisque  $r_1$  appartient à l'intervalle  $]0, 1[$ , alors  $(1 - r_1)$  aussi et la fraction précédente est forcément négative ce qui n'est pas possible puisque  $r_3 \in ]0, 1[$ , donc  $G \neq 0$ .

$l$  étant une distance, on veut une valeur positive et nous avons donc le résultat final :

$$l = \sqrt{\frac{d^2 - F}{G}} \quad (\text{C.19})$$



# Glossaire

**Articulation distale** L'articulation extrémité du segment, par exemple le poignet pour l'avant-bras.

**Articulation intermédiaire d'un membre** L'articulation intermédiaire du membre est celui qui est à l'intersection de ces deux segments, à savoir le coude ou le genou.

**Articulation proximale** L'articulation "mère" du segment, par exemple le coude pour l'avant-bras.

**Contraintes de support** Ce sont les contraintes utilisées pour assurer le respect des appuis des pieds du personnage sur le sol conformément à la phase de support en cours : bipodal (les deux pieds au sol), monopodal (un seul pied au sol, le droit ou le gauche) ou aucun appui (phase de saut).

**Longueur variable d'un membre** C'est le paramètre de longueur variable associé à un membre dans la représentation normalisée du squelette.

**Mélange de contraintes** Le respect de toutes les contraintes en fonction de leur importance. Cette adaptation est réalisée en utilisant une somme pondérée par les poids de chaque contrainte.

**Mode de contrôle des contraintes** La façon dont les contraintes sont prises en compte. Deux modes sont décrits dans cette thèse : la priorité de contraintes et le mélange de contraintes.

**Mouvement de référence** Le mouvement adapté aux contraintes. Ce mouvement est en fait constitué de la succession des postures de référence à chaque pas de temps.

**Posture de référence** La posture que l'on souhaite adapter afin de respecter les contraintes fournies. Cette posture, que l'on souhaite préserver au mieux, permet de contraindre les degrés de liberté libres de notre système sous-contraint. C'est la posture qui serait utilisée pour animer le personnage à chaque pas de temps si aucune adaptation n'était faite.

**Priorité de contraintes** Le respect des contraintes dans l'ordre inverse de leur priorité.  
Ce mode est plus connu sous le nom de "Task-Priority".

# Notations

$COM$	Le centre de masse ( <i>Center Of Mass</i> )
$C_i$	La $i^{\text{ème}}$ contrainte spécifiée
$PG_i$	Les paramètres génériques de la contrainte $C_i$
$PS_i$	Les paramètres spécifiques à la contrainte $C_i$
$Id_i$	L'identification de la contrainte $C_i$
$Seg_i$	Le segment attaché à la contrainte $C_i$
$Decal_i$	La position relative de la contrainte $C_i$ par rapport à $S_i$
$Portee_i$	La portée en nombre de segments de la contrainte $C_i$
$Imp_i$	L'importance de la contrainte $C_i$
$Prec_i$	La précision de prise en compte de la contrainte $C_i$
$Act_i$	L'instant d'activation de la contrainte $C_i$
$dAct_i$	La durée de la phase d'activation de la contrainte $C_i$
$Desact_i$	L'instant de désactivation de la contrainte $C_i$
$dDesact_i$	La durée de la phase de désactivation de la contrainte $C_i$
$Type_i$	Le type de la contrainte $C_i$
$CP_i$	La position de la $i^{\text{ème}}$ contrainte spécifiée
$Cible_i$	La cible de la contrainte $CP_i$
$A_i$	L'articulation proximale du segment $S_i$
$Root_M$	La racine ( <i>Root</i> ) d'un membre : l'épaule ou la hanche
$Extr_M$	L'extrémité d'un membre : le poignet ou la cheville
$Inter_M$	L'articulation intermédiaire d'un membre : le coude ou le genou
$CmRacine$	Le paramètre de contrôle cinématique de la racine
$CnRacine$	Le paramètre de contrôle cinétique de la racine
$CnMode$	Le mode de contrôle du $COM$ : sur le plan horizontal ou en 3D
$CmSeuil$	Le critère d'arrêt de l'adaptation cinématique
$CnSeuil$	Le critère d'arrêt de l'adaptation cinétique
$CnErreurMax$	La précision de calcul du centre de masse
$\vec{\varepsilon}_i$	L'erreur vectorielle entre la contrainte $C_i$ et sa cible $Cible_i$
$\vec{\varepsilon}$	L'erreur vectorielle moyenne des contraintes
$CmErreur_k$	La moyenne des valeurs absolues des erreurs $\vec{\varepsilon}_i$ pour l'itération $k$
$\Delta CmErreur_k$	La variation de l'erreur entre deux itérations
$CnErreur_k$	L'erreur sur la position du centre de masse pour l'itération $k$
$\Delta CnErreur_k$	La variation de l'erreur $CnErreur_k$ entre deux itérations
$EGC$	Ensemble des groupes utilisés pour l'adaptation cinétique





# Bibliographie

- [Abe 04] Y. Abe, C. K. Liu & Z. Popović. *Momentum-based parameterization of dynamic character motion*. In SCA'04 : Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 173–182, Grenoble, France, 2004. ACM Press.
- [Alexa 00] M. Alexa & W. Müller. *Representing animations by principal components*. Computer Graphics Forum, Eurographics 2000, vol. 19, no. 3, pages 411–426, 2000.
- [Alexander 84] R.M. Alexander. *Walking and Running*. In American Scientist, volume 72, pages 348–354, 1984.
- [Allard 00] P. Allard & J. P. Blanche. *Analyse du mouvement humain par la biomécanique*, volume 2ème édition. Décarié éditeurs Inc., 2000.
- [Arikan 02] O. Arikan & D. A. Forsyth. *Interactive motion generation from examples*. In SIGGRAPH'02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques, pages 483–490, San Antonio, Texas, 2002. ACM Press.
- [Arikan 05] O. Arikan, D. A. Forsyth & J. F. O'Brien. *Pushing people around*. In SCA'05 : Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 59–66, Los Angeles, California, 2005. ACM Press.
- [Arnaldi 89] B. Arnaldi. *Conception du noyau d'un système d'animation de scènes tridimensionnelles intégrant les lois de la mécanique*. Thèse, Université de Rennes 1, 1989.
- [Ashraf 00] G. Ashraf & K. C. Wong. *Generating Consistent Motion Transition via Decoupled Framespace Interpolation*. Computer Graphics Forum, vol. 19, no. 3, 2000.
- [Ashraf 01] G. Ashraf & K.C. Wong. *Constrained Framespace Interpolation*. In Computer Animation 2001 : Proceedings of the 14th conference on Computer Animation, pages 61–72, November 2001.
- [Ashraf 03] G. Ashraf & K. C. Wong. *Semantic Representation and Correspondence for State-Based Motion Transition*. IEEE Transactions on Visualization and Computer Graphics, vol. 9, no. 4, pages 481–499, 2003.
- [Assa 05] J. Assa, Y. Caspi & D. Cohen-Or. *Action Synopsis : Pose Selection and Illustration*. In Proceedings of SIGGRAPH'05 : ACM Transactions on Graphics, vol. 24, no. 3, pages 667–676, Juillet 2005.

- [Aydin 99a] Y. Aydin & M. Nakajima. *Balance control and mass centre adjustment of articulated figures in interactive environments*. The Visual Computer, vol. 15, no. 3, pages 113–123, 1999.
- [Aydin 99b] Y. Aydin & M. Nakajima. *Realistic Articulated Character Positioning and Balance Control in Interactive Environments*. In CA'99 : Proceedings of the Computer Animation, page 160. IEEE Computer Society, 1999.
- [Badler 87a] N. I. Badler, K. H. Manoochehri & D. Baraff. *Multi-dimensional input techniques and articulated figure positioning by multiple constraints*. In SI3D'86 : Proceedings of the 1986 workshop on Interactive 3D graphics, pages 151–169, Chapel Hill, North Carolina, USA, 1987. ACM Press.
- [Badler 87b] N. I. Badler, K. H. Manoochehri & G. Walters. *Articulated figure positioning by multiple constraints*. IEEE Computer Graphics and Applications, vol. 7, no. 6, pages 28–38, Juin 1987.
- [Badler 93] N. I. Badler, C. B. Phillips & B. L. Webber. *Simulating humans : computer graphics animation and control*. Oxford University Press, Inc., New York, NY, USA, 1993.
- [Baerlocher 98] P. Baerlocher & R. Boulic. *Task-Priority Formulations for the Kinematic Control of Highly Redundant Articulated Structures*. IEEE IROS, 1998.
- [Baerlocher 00] P. Baerlocher & R. Boulic. *Kinematic Control of the Mass Properties of Redundant Articulated Bodies*. In ICRA, pages 2557–2562, 2000.
- [Baerlocher 01] P. Baerlocher. *Inverse Kinematics Techniques for the Interactive Posture Control of Articulated Figures*. Thèse, École Polytechnique Fédérale de Lausanne, Suisse, Décembre 2001.
- [Baerlocher 04] P. Baerlocher & R. Boulic. *An inverse kinematics architecture enforcing an arbitrary number of strict priority levels*. The Visual Computer, vol. 20, no. 6, pages 402–417, 2004.
- [Baillieul 86] J. Baillieul. *Avoiding obstacles and resolving kinematic redundancy*. In Proceedings of 198 International Conference on Robotics and Automation, pages 1698–1704, Avril 1986.
- [Beaupied 03] H. Beaupied. *Etude mécanique et énergétique de la marche, de la course et de la transition marche-course : influence de la spécialité athlétique*. Thèse, Université de Rennes 2, Mars 2003.
- [Bideau 03a] B. Bideau. *La réalité virtuelle pour l'analyse des duels sportifs : application au duel entre un gardien et un tireur au handball*. Thèse, Université Rennes 2, Décembre 2003.
- [Bideau 03b] B. Bideau, R. Kulpa, S. Ménardais, L. Fradet, F. Multon, P. Delamarche & B. Arnaldi. *Real handball goalkeeper vs. virtual handball thrower*. Presence : Teleoper. Virtual Environment, vol. 12, no. 4, pages 411–421, 2003.
- [Bideau 04] B. Bideau, F. Multon, R. Kulpa, L. Fradet & B. Arnaldi. *Virtual reality applied to sports : do handball goalkeepers react realistically to simulated*

- synthetic opponents?* In VRCAI'04 : Proceedings of the 2004 ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Applications in Industry, pages 210–216, Singapore, 2004. ACM Press.
- [Bindiganavale 98] R. Bindiganavale & N. I. Badler. *Motion Abstraction and Mapping with Spatial Constraints*. In CAPTECH'98 : Proceedings of the International Workshop on Modelling and Motion Capture Techniques for Virtual Environments, pages 70–82. Springer-Verlag, 1998.
- [Bodenheimer 97] B. Bodenheimer, C. Rose, S. Rosenthal & J. Pella. *The Process of Motion Capture : Dealing with the Data*. In Computer Animation and Simulation'97, proceedings of Eurographics International Workshop., pages 3–18, Septembre 1997.
- [Boulic 90] R. Boulic, N. Magnenat Thalmann & D. Thalmann. *A global human walking model with real-time kinematic personification*. The Visual Computer, vol. 6, no. 6, pages 344–358, Décembre 1990.
- [Boulic 92] R. Boulic & D. Thalmann. *Combined Direct and Inverse Kinematic Control for Articulated Figure Motion Editing*. Computer Graphics forum, vol. 11, no. 4, pages 189–202, 1992.
- [Boulic 94] R. Boulic, R. Mas & D. Thalmann. *Inverse Kinetics for Center of Mass Position Control and Posture Optimization*. Race Workshop on “Combined real and synthetic image processing for broadcast and video production (Monalisa Project)”, September 1994.
- [Boulic 96] R. Boulic, R. Mas & D. Thalmann. *A robust approach for the control of the center of mass with inverse kinetics*. Computers and Graphics, vol. 20, no. 5, pages 693–701, Septembre 1996.
- [Boulic 97a] R. Boulic, P. Becheiraz, L. Emering & D. Thalmann. *Integration of motion control techniques for virtual human and avatar real-time animation*. In VRST'97 : Proceedings of the ACM Symposium on Virtual Reality Software and Technology, pages 111–118, Lausanne, Switzerland, 1997. ACM Press.
- [Boulic 97b] R. Boulic, R. Mas-Sanso & D. Thalmann. *Complex Character Positioning Based on a Compatible Flow Model of Multiple Supports*. IEEE Transactions on Visualization and Computer Graphics, vol. 3, no. 3, pages 245–261, jul-sep 1997.
- [Boulic 98] R. Boulic, P. Fua, L. Herda, M. Silaghi, J-S. Monzani, L. Nedel & D. Thalmann. *An Anatomic Human Body for Motion Capture*. In Proceedings of EMMSEC'98, 1998.
- [Boulic 00] R. Boulic, M-C. Silaghi & D. Thalmann. *Visualization of Local Movements for Optimal Marker Positioning*. In AMDO'00 : Proceedings of the First International Workshop on Articulated Motion and Deformable Objects, pages 133–144. Springer-Verlag, 2000.
- [Brand 00] M. Brand & A. Hertzmann. *Style Machines*. In Proceedings of ACM SIGGRAPH'00, Computer Graphics Proceedings, Annual Conference Series, pages 183–192, Juillet 2000.

- [Bruderlin 89] A. Bruderlin & T. W. Calvert. *Goal-Directed, Dynamic Animation of Human Walking*. Proceedings of SIGGRAPH'89, Computer Graphics, vol. 23, no. 3, pages 233–242, Juillet 1989.
- [Bruderlin 93] A. Bruderlin & T. Calvert. *Interactive animation of personalized human locomotion*. In Graphics Interface, pages 17–23, Mai 1993.
- [Bruderlin 95] A. Bruderlin & L. Williams. *Motion Signal Processing*. In Proceedings of SIGGRAPH'95, Computer Graphics Proceedings, Annual Conference Series, pages 97–104, Août 1995.
- [Bruderlin 96] A. Bruderlin & T. Calvert. *Knowledge-driven, interactive animation of human running*. In GI'96 : Proceedings of the Conference on Graphics Interface, pages 213–221, Toronto, Ontario, Canada, 1996. Canadian Information Processing Society.
- [Chadwick 89] J. E. Chadwick, D. R. Haumann & R. E. Parent. *Layered construction for deformable animated characters*. In SIGGRAPH'89 : Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques, pages 243–252. ACM Press, 1989.
- [Chaffin 99] D. B. Chaffin, G. B. Anderson & B. J. Martin. Occupational biomechanics, volume 3ème édition. Wiley Interscience, New York, USA, 1999.
- [Chandler 75] R. F. Chandler, C. E. Clauser, J. T. McConville, H. M. Reynolds & J. W. Young. *Investigation of inertial properties of the human body*. In AMRL Technical report, pages 74–137. Wright-Patterson Air Force Base, 1975.
- [Choi 99] K-J. Choi, S-H. Park & H-S. Ko. *Processing motion capture data to achieve positional accuracy*. Graphic Models Image Processing, vol. 61, no. 5, pages 260–273, 1999.
- [Choi 00] K-J. Choi & H-S. Ko. *Online motion retargetting*. Journal of Visualization and Computer Animation, vol. 11, no. 5, pages 223–235, 2000.
- [Choi 03] M. G. Choi, J. Lee & S. Y. Shin. *Planning Biped Locomotion using Motion Capture Data and Probabilistic Roadmaps*. ACM Transactions on Graphics, vol. 22, no. 2, pages 182–203, 2003.
- [Clauser 69] C. E. Clauser, J. T. McConville & J. W. Young. Weight, volume and center of mass of segments of the human body. Wright-Patterson Air Force Base, 1969.
- [Cogné 97] L. Cogné. *Simulation de systèmes physiques lagrangiens : de la représentation symbolique aux évaluations numériques séquentielles et parallèles*. Thèse, Université de Rennes 1, Décembre 1997.
- [Cohen 92] M. F. Cohen. *Interactive spacetime control for animation*. Proceedings of SIGGRAPH'92, Computer Graphics, vol. 26, no. 2, pages 293–302, Juillet 1992.
- [Cozot 99] R. Cozot, F. Multon, B. Valton & B. Arnaldi. *Animation Levels of Detail Design for Real-Time Virtual Human*. Computer Animation and Simulation'99, pages 35–44, Mai 1999.

- [Das 88] H. Das, J.-J. Slotine & T. B. Sheridan. *Inverse kinematic algorithms for redundant systems*. In Proceedings of the International Conference on Robotics and Automation, pages 43–48, 1988.
- [de Leva 96] P. de Leva. *Adjustements to Zatsiorsky-Seluyanov's segment inertia parameters*. Journal of Biomechanics, vol. 29, no. 9, pages 1223–1230, 1996.
- [Dempster 55] W. T. Dempster. Space requirements for the seated operator. Wright-Patterson Air Force Base, Ohio, 1955.
- [Dumont 90] G. Dumont. *Animation de scènes tridimensionnelles : la mécanique des solides comme modèle de synthèse du mouvement*. Thèse, Université de Rennes 1, 1990.
- [Durocher 05] C. Durocher. *Analyse, modélisation et simulation de l'impulsion au sol dans les activités physiques*. Thèse, Université de Rennes 2, Septembre 2005.
- [Ehara 95] Y. Ehara, H. Fujimoto, S. Miyazaki, S. Tanaka & S. Yamamoto. *Comparison of the performance of 3D camera systems*. Technical Note, Gait & Posture, vol. 3, pages 166–169, 1995.
- [Ehara 97] Y. Ehara, H. Fujimoto, S. Miyazaki, M. Mochimaru, S. Tanaka & S. Yamamoto. *Comparison of the performance of 3D camera systems II*. Technical Note, Gait & Posture, vol. 5, pages 251–255, 1997.
- [Espiau 98] B. Espiau & R. Boulic. *On the Computation and Control of the Mass Center of Articulated Chains*. RR-3479, pages 1–38, Août 1998.
- [Esteves 06] C. Esteves, G. Arechavaleta, J. Pettré & J.-P. Laumond. *Animation Planning for Virtual Characters Cooperation*. To appear in ACM Transactions on Graphics, page A paraître, 2006.
- [Faloutsos 01] P. Faloutsos, M. van de Panne & D. Terzopoulos. *Composable controllers for physics-based character animation*. In SIGGRAPH'01 : Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pages 251–260. ACM Press, Août 2001.
- [Fang 03] A. C-H. Fang. *Efficient synthesis of physically valid human motion for computer animation*. Thèse, Brown University, 2003. Adviser-Nancy S. Pollard.
- [Fitts 54] P. M. Fitts. *The information capacity of the human motor system in controlling the amplitude of movement*. Journal of the Experimental Psychology, vol. 47, no. 1, pages 381–391, 1954.
- [Flash 85] T. Flash & N. Hogan. *The coordination of arm movements : an experimentally confirmed mathematical model*. Journal of Neurosciences, vol. 5, pages 1688–1703, 1985.
- [Fradet 03] L. Fradet. *Analyse biomécanique, modélisation et simulation du tir au handball*. Thèse, Université de Rennes 2, 2003.
- [Furukawa 01] D. Furukawa. *Human Spine Posture Estimation from Video Images Based on Connected Vertebra Spheres Model*. In Proceedings of the 14th Conference on Computer Animation'01, pages 174–181, 2001.

- [Fusco 05] N. Fusco, G. Nicolas, F. Multon & A. Crétual. *Simulation of hemiplegic subjects' locomotion*. Lecture Notes in Computer Science, vol. Sous presse, 2005.
- [Gibet 01] S. Gibet, T. Lebourque & P.F. Marteau. *High level Specification and Animation of Communicative Gestures*. Journal of Visual Languages and Computing, vol. 12, no. 6, pages 657–687, Décembre 2001.
- [Gill 81] P. Gill, W. Murray & M. Wright. *Practical optimization*. Academic press, New York, 1981.
- [Girard 85] M. Girard & A. A. Maciejewski. *Computational modeling for the computer animation of legged figures*. In SIGGRAPH'85 : Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques, pages 263–270. ACM Press, 1985.
- [Girard 87] M. Girard. *Interactive design of 3D computer-animated legged animal motion*. In IEEE Computer Graphics and Applications, pages 39–51, Juin 1987.
- [Glardon 04] P. Glardon, R. Boulic & D. Thalmann. *A coherent locomotion engine extrapolation beyond experimental data*. Proceedings of CASA, 2004.
- [Gleicher 97] M. Gleicher. *Motion Editing with Spacetime Constraints*. In Symposium on Interactive 3D Graphics'97, pages 139–148, Avril 1997.
- [Gleicher 98a] M. Gleicher. *Retargetting Motion to New Characters*. In Proceedings of SIGGRAPH'98, Computer Graphics Proceedings, Annual Conference Series, pages 33–42, Juillet 1998.
- [Gleicher 98b] M. Gleicher & P. Litwinowicz. *Constraint-based Motion Adaptation*. The Journal of Visualization and Computer Animation, vol. 9, no. 2, pages 65–94, 1998.
- [Gleicher 01a] M. Gleicher. *Comparing constraint-based motion editing methods*. Graphical models, vol. 63, no. 2, pages 107–134, 2001.
- [Gleicher 01b] M. Gleicher. *Motion Path Editing*. In ACM Symposium on Interactive 3D Graphics'01, pages 195–202, Mars 2001.
- [Grassia 98] F. S. Grassia. *Practical Parameterization of Rotations Using the Exponential Map*. Journal of Graphics Tools, vol. 3, no. 3, pages 29–48, 1998.
- [Gravila 99] D.M. Gravila. *The visual analysis of human movement : A survey*. Computer Vision and Image Understanding, vol. 3, no. 1, pages 82–98, Janvier 1999.
- [Gray 18] H. Gray. *Anatomy of the human body*. Lea & Febiger, 1918.
- [Greville 59] T. N. E. Greville. *The pseudoinverse of a rectangular or singular matrix and its application to the solution of systems of linear equations*. SIAM Review, vol. 1, no. 1, pages 38–43, Janvier 1959.
- [Grochow 04] K. Grochow, S. L. Martin, A. Hertzmann & Z. Popović. *Style-based inverse kinematics*. ACM Trans. Graph., vol. 23, no. 3, pages 522–531, 2004.

- [Guo 96] S. Guo & J. Robergé. *A high-level control mechanism for human locomotion based on parametric frame space interpolation*. In Proceedings of the Eurographics Workshop on Computer Animation and Simulation'96, pages 95–107, Poitiers, France, 1996. Springer-Verlag New York, Inc.
- [H-Anim 01] H-Anim. *Information technology  $\dot{\cup}$  Computer graphics and image processing  $\dot{\cup}$  Humanoid animation (H-Anim)*, ISO/IEC FCD 19774. <http://www.h-anim.org>, 2001.
- [Hamilton 44] W. R. Hamilton. *On a new Species of Imaginary Quantities connected with a theory of Quaternions*. Proceedings of the Royal Irish Academy, vol. 2, pages 424–434, 1844.
- [Hamilton 50] W. R. Hamilton. *On Quaternions and the Rotation of a Solid Body*. Proceedings of the Royal Irish Academy, vol. 4, pages 38–56, 1850.
- [Hanafusa 81] H. Hanafusa, T. Yoshikawa & Y. Nakamura. *Analysis and Control of Articulated Robot with Redundancy*. In IFAC, 8th Triennial World Congress, volume 4, pages 1927–1932, 1981.
- [Herda 00] L. Herda, P. Fua, R. Plänklers, R. Boulic & D. Thalmann. *Skeleton-Based Motion Capture for Robust Reconstruction of Human Motion*. In CA'00 : Proceedings of the Computer Animation, page 77. IEEE Computer Society, 2000.
- [Herda 01] L. Herda, P. Fua, R. Plänkers, R. Boulic & D. Thalmann. *Using skeleton-based tracking to increase the reliability of optical motion capture*. Human Movement Science Journal, vol. 20, no. 3, pages 313–341, Juin 2001.
- [Hodgins 95] J. K. Hodgins, W. L. Wooten, D. C. Brogan & J. F. O'Brien. *Animating Human Athletics*. In Proceedings of ACM SIGGRAPH'95, Computer Graphics Proceedings, Annual Conference Series, pages 71–78, Août 1995.
- [Hodgins 97] J. K. Hodgins & N. S. Pollard. *Adapting Simulated Behaviors For New Characters*. In Proceedings of ACM SIGGRAPH'97, Computer Graphics Proceedings, Annual Conference Series, pages 153–162, Août 1997.
- [Hotelling 33] H. Hotelling. *Analysis of a complex of statistical variables into principal components*. Journal of Educational Psychology, vol. 24, pages 417–441, 1933.
- [Jensen 78] R. K. Jensen. *Estimation of the biomechanical properties of three body types using a photogrammetric method*. Journal of Biomechanics, vol. 11, pages 349–358, 1978.
- [Jensen 94] R. K. Jensen & P. Fletcher. *Distribution of mass to the segments of elderly males and females*. Journal of Biomechanics, vol. 27, pages 89–96, 1994.
- [Johansson 73] G. Johansson. *Visual Perception of biological motion and a model for its analysis*. Perception and Psychophysics, vol. 14, no. 2, pages 201–211, 1973.

- [Kallman 03] M. Kallman, A. Aubel, T. Abaci & D. Thalmann. *Planning collision-free reaching motions for interactive object manipulation and grasping*. In P. Brunet & D. Fellner, editeurs, ACM SIGGRAPH/Eurographics, volume 22. Eurographics association, 2003.
- [Klein 83] C. Klein & C. Huang. *Review of pseudoinverse control for use with kinematically redundant manipulators*. IEEE Transactions on Systems, Man and Cybernetics, vol. SMC-13, no. 3, pages 245–250, Mars 1983.
- [Klein 95] C. Klein & S. Ahmed. *Repeatable pseudoinverse control for planar kinematically redundant manipulators*. IEEE Transactions on Systems, Man and Cybernetics, vol. 25, no. 12, pages 1657–1662, 1995.
- [Ko 93] H. Ko & N.I. Badler. *Straight-Line Walking Animation Based on Kinematic Generalization that Preserves the Original Characteristics*. In Proceedings of Graphics Interface'93, pages 9–16, Mai 1993.
- [Koga 94] Y. Koga, K. Kondo, J. Kuffner & J-C. Latombe. *Planning motions with intentions*. In SIGGRAPH'94 : Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, pages 395–408. ACM Press, 1994.
- [Komura 00] T. Komura, Y. Shinagawa & T. L. Kunii. *Creating and retargetting motion by the musculoskeletal human body model*. The Visual Computer, vol. 16, no. 5, pages 254–270, 2000.
- [Komura 01] T. Komura & Y. Shinagawa. *Attaching physiological effects to motion-captured data*. In GRIN'01 : No description on Graphics Interface, pages 27–36, Ottawa, Ontario, Canada, 2001. Canadian Information Processing Society.
- [Kondo 91] K. Kondo. *Inverse kinematics of a human arm*. Journal of Robotics and Systems, vol. 8, no. 2, pages 115–175, 1991.
- [Korein 82] J. U. Korein & N. I. Badler. *Techniques for generating the goal-directed animation of articulated structures*. IEEE Computer Graphics and Application, vol. 2, no. 9, pages 71–81, 1982.
- [Korein 85] J. U. Korein. *A geometric investigation of reach*. MIT Press, Cambridge, MA, USA, 1985.
- [Kovar 02] L. Kovar, M. Gleicher & F. Pighin. *Motion graphs*. In Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, pages 473–482, San Antonio, Texas, 2002. ACM Press.
- [Lacquaniti 83] F. Lacquaniti, C. Terzuolo & P. Viviani. *The law relating the kinematic and figural aspects of drawing movements*. Acta Physiological, vol. 54, pages 115–130, 1983.
- [Lander 98a] J. Lander. *Making kine more flexible*. Game Developer Magazine, pages 15–22, Novembre 1998.
- [Lander 98b] J. Lander. *Oh my god, I inverted Kine!* Game Developer Magazine, Septembre 1998.



- [Laszlo 96] J. F. Laszlo, M. van de Panne & E. L. Fiume. *Limit Cycle Control and its Application to the Animation of Balancing and Walking*. In Proceedings of SIGGRAPH'96, Computer Graphics Proceedings, Annual Conference Series, pages 155–162, Août 1996.
- [Latombe 91] J.-C. Latombe. *Robot motion planning*. Kluwer Academic Publishers, 1991.
- [Laumond 05] J.-P. Laumond, E. Ferré, G. Arechavaleta & C. Esteves. *Mechanical part assembly planning with virtual mannequins*. In ISATP'2005 : IEEE International Symposium on Assembly and Task Planning, Juillet 2005.
- [le Callennec 04] B. le Callennec & R. Boulic. *Interactive motion deformation with prioritized constraints*. In SCA '04 : Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 163–171, Grenoble, France, 2004. ACM Press.
- [Lecerf 94] C. Lecerf. *Contrôle de mouvement de systèmes mécaniques en animation*. Thèse, Université de Rennes 1, France, Juin 1994.
- [Lee 99] J. Lee & S. Y. Shin. *A hierarchical approach to interactive motion editing for human-like figures*. In SIGGRAPH'99 : Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pages 39–48, 1999.
- [Lee 02] J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins & N. S. Pollard. *Interactive Control of Avatars Animated With Human Motion Data*. ACM Transactions on Graphics, vol. 21, no. 3, pages 491–500, Juillet 2002.
- [Lee 04] J. Lee & K. H. Lee. *Precomputing avatar behavior from human motion data*. In SCA'04 : Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer Animation, pages 79–87, Grenoble, France, 2004. ACM Press.
- [Liegeois 77] A. Liegeois. *Automatic Supervisory control of the Configuration and Behavior of Multibody Mechanisms*. IEEE Transactions on Systems, Man and Cybernetics, vol. SMC-7, no. 12, Décembre 1977.
- [Liu 94] Z. Liu, S. J. Gortler & M. F. Cohen. *Hierarchical spacetime control*. In SIGGRAPH'94 : Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, pages 35–42. ACM Press, Juillet 1994.
- [Liu 02] C. K. Liu & Z. Popović. *Synthesis of complex dynamic character motion from simple animations*. In SIGGRAPH'02 : Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, pages 408–416, San Antonio, Texas, 2002. ACM Press.
- [Luenberger 84] D. G. Luenberger. *Linear and nonlinear programming*. Addison-Wesley, 1984.
- [Maciejewski 90] A. A. Maciejewski. *Dealing with the Ill-Conditioned Equations of Motion for Articulated Figures*. IEEE Computer Graphics and Applications, vol. 10, no. 3, pages 63–71, Mai 1990.

- [Maciel 02] A. Maciel, L. Porcher Nedel & C. M. Dal Sasso Freitas. *Anatomy-Based Joint Models for Virtual Human Skeletons*. In CA'02 : Proceedings of the Computer Animation, page 220. IEEE Computer Society, 2002.
- [Magdalena 93] L. Magdalena & F. Monasterio. *Fuzzy Controlled Gait Synthesis for a Bipedal Walking Machine*. Fuzzy Systems, vol. 2, pages 1334–1339, Mars 1993. San Francisco, US.
- [Marey 94] M. Marey. *Mécanique animale*, volume 119. New-York : Appleton & Co., 1894.
- [Marteau 01] P. F. Marteau, S. Gibet & F. Juillard. *Non Parametric Learning of Sensori-Motor Maps. Application to the control of multi Joint Systems*. In WSES/IEEE conf. on Neural, Fuzzy and Evolutionary Computation, July 2001.
- [Mas 96] R. Mas. *Balance control of multiple supported articulated systems for computer animation*. Thèse, Université des Iles Baléares, 1996.
- [Maurel 00] W. Maurel & D. Thalmann. *Human Upper Limb Modeling including Scapulo-Thoracic Constraint and Joint Sinus Cones*. Computers and Graphics, vol. 24, no. 2, pages 203–218, 2000.
- [Menache 99] A. Menache. *Understanding motion capture for computer animation and video games*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.
- [Ménardais 01] S. Ménardais, F. Multon & B. Arnaldi. *Amélioration des trajectoires acquises par des systèmes optiques pour l'animation de personnages synthétiques*. Revue Internationale de CFAO, pages 99–103, 2001.
- [Ménardais 03] S. Ménardais. *Fusion et adaptation temps réel de mouvements acquis pour l'animation d'humanoïdes synthétiques*. Thèse, Université de Rennes 1, Janvier 2003.
- [Menardais 04a] S. Menardais, R. Kulpa, F. Multon & B. Arnaldi. *Synchronization for dynamic blending of motions*. In Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pages 325–336, Août 2004.
- [Ménardais 04b] S. Ménardais, F. Multon, R. Kulpa & B. Arnaldi. *Motion Blending for Real-Time Animation while Accounting for the Environment*. In CGI'04 : Proceedings of the Computer Graphics International, pages 156–159. IEEE Computer Society, 2004.
- [Molet 96] T. Molet, R. Boulic & D. Thalmann. *A real time anatomical converter for human motion capture*. In Proceedings of the Eurographics Workshop on Computer Animation and Simulation'96, pages 79–94, Poitiers, France, 1996. Springer-Verlag New York, Inc.
- [Molet 97] T. Molet, Z. Huang, R. Boulic & D. Thalmann. *An Animation Interface Designed for Motion Capture*. In CA'97 : Proceedings of the Computer Animation, page 77. IEEE Computer Society, 1997.
- [Monheit 91] G. Monheit & N. I. Badler. *A Kinematic Model of the Human Spine and Torso*. IEEE Computer Graphics and Applications, vol. 11, no. 2, pages 29–38, 1991.

- [Monzani 00] J-S. Monzani, P. Baerlocher, R. Boulic & D. Thalmann. *Using an Intermediate Skeleton and Inverse Kinematics for Motion Retargetting*. Computer Graphics Forum, vol. 19, no. 3, Août 2000. ISSN 1067-7055.
- [Multon 99] F. Multon, J. L. Nougaret, B. Arnaldi, G. Hégron & L. Millet. *A Software System to Carry-out Virtual Experiments on Human Motion*. In Proceedings IEEE, Computer Animation, pages 16–23, Mai 1999.
- [Nebel 99a] J. C. Nebel. *Keyframe animation of articulated figures using autocollision-free interpolation*. In 17th Eurographics UK Conference'99, Avril 1999.
- [Nebel 99b] J. C. Nebel. *Keyframe interpolation with self-collision avoidance*. In Eurographics Workshop on Computer Animation and Simulation, pages 77–86, Septembre 1999.
- [Nebel 00] J. C. Nebel. *Realistic collision avoidance of upper limbs based on neuroscience models*. Computer Graphics Forum, vol. 19, no. 3, 2000.
- [Ngo 93] J. T. Ngo & J. Marks. *Spacetime constraints revisited*. In Proceedings of SIGGRAPH'93, pages 343–350, 1993.
- [Nicolas 04] G. Nicolas, F. Multon & G. Berillon. *Inverse kinematics for the calculation of plausible bipedal locomotion using anthropological knowledge*. In Proceedings of CASA, pages 103–110, July 2004.
- [Nilsson 85] J. Nilsson, A. Thorstensson & J. Halbertsam. *Changes in leg movements and muscle activity with speed of locomotion and mode of progression in humans*. Acta Physiol Scand, vol. 123, no. 4, pages 457–475, Avril 1985.
- [Niskanen 05] M. Niskanen, E. Boyer & R. Horaud. *Articulated Motion Capture from 3-D Points and Normals*. In Torr Clocksin Fitzgibbon, editeur, British Machine Vision Conference, volume 1, pages 439–448. BMVA, British Machine Vision Association, September 2005.
- [O'Brien 00] J. O'Brien, R. E. Bodenheimer, G. Brostow & J. K. Hodgins. *Automatic joint parameter estimation from magnetic motion capture data*. In Proceedings of Graphics Interface'00, pages 53–60, 2000.
- [Oshita 01] M. Oshita & A. Makinouchi. *A Dynamic Motion Control Technique for Human-like Articulated Figures*. Computer Graphics Forum, vol. 20, no. 3, 2001.
- [Park 02] S.I. Park, H. J. Shin & S. Y. Shin. *On-line Locomotion Generation Based on Motion Blending*. In ACM SIGGRAPH'02 Symposium on Computer Animation, pages 105–112, Juillet 2002.
- [Park 04] S. I. Park, H. J. Shin, T. H. Kim & S. Y. Shin. *On-line motion blending for real-time locomotion generation*. Computer Animation and Virtual Worlds, vol. 15, no. 3-4, pages 125–138, 2004.
- [Patla 91] A.E. Patla. *Adaptability of human gait*, chapitre Visual Control of Human Locomotion, pages 55–97. Elsevier Science Publishers, 1991.
- [Paul 81] R. P. Paul. *Robot manipulators : Mathematics, programming and control*. MIT Press, Cambridge, MA, USA, 1981.

- [Pavol 02] M. J. Pavol, T. M. Owings & M. D. Grabiner. *Body segment inertial parameter for the general population of older adults*. Journal of Biomechanics, vol. 35, no. 5, pages 707–712, Mai 2002.
- [Pearson 01] K. Pearson. *Principal Components Analysis*. The London, Edinburgh and Dublin Philosophical Magazine & Journal, vol. 6, no. 2, page 566, 1901.
- [Penrose 55] F. Penrose. *A generalized inverse for matrices*. In Proceedings of the Cambridge Philosophical Society 51, pages 406–413, 1955.
- [Pettré 03] J. Pettré, J.-P. Laumond & T. Siméon. *A 2-stages locomotion planner for digital actors*. In SCA'03 : Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pages 258–264, San Diego, California, 2003. Eurographics Association.
- [Phillips 91] C. B. Phillips & N. I. Badler. *Interactive behaviors for bipedal articulated figures*. Proceedings of SIGGRAPH'91 on Computer Graphics, vol. 25, no. 4, pages 359–362, Juillet 1991.
- [Pollard 00] N. Pollard & F. Behmaram-Mosavat. *Force-Based Motion Editing for Locomotion Tasks*. In Proceedings of IEEE International Conference on Robotics and Automation, pages 663–669, Avril 2000.
- [Popović 99] Z. Popović & A. Witkin. *Physically based motion transformation*. In SIGGRAPH'99 : Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pages 11–20. ACM Press/Addison-Wesley Publishing Co., 1999.
- [Press 92] W. H. Press, S. A. Teukolsky, W. T. Vetterling & B. P. Flannery. *Numerical recipes in c*. Cambridge University Press, 2nd edition edition, 1992.
- [Pronost 05] N. Pronost, G. Dumont, G. Berillon & G. Nicolas. *Morphological and stance interpolations in database for simulating bipedalism of virtual humans*. Visual Computer, 2005.
- [Rijpkema 91] H. Rijpkema & M. Girard. *Computer animation of knowledge-based human grasping*. In Proceedings of ACM SIGGRAPH'91, pages 339–348, Juillet 1991.
- [Rose 96] C. F. Rose, B. Guenter, B. Bodenheimer & M. F. Cohen. *Efficient Generation of Motion Transitions using Spacetime Constraints*. In Proceedings of SIGGRAPH'96, Computer Graphics Proceedings, Annual Conference Series, pages 147–154, Août 1996.
- [Rose 98] C. Rose, M. F. Cohen & B. Bodenheimer. *Verbs and Adverbs : Multi-dimensional Motion Interpolation*. IEEE Computer Graphics and Applications, vol. 18, no. 5, pages 32–40, sep–oct 1998.
- [Safonova 05] A. Safonova & J. K. Hodgins. *Analyzing the Physical Correctness of Interpolated Human Motion*. In Proceedings of SCA'05, pages 171–180, Los Angeles, CA, USA, Juillet 2005.
- [Sardain 04] P. Sardain & G. Bessonnet. *Forces acting on a biped robot. Center of pressure-zero moment point*. IEEE Transactions on Systems, Man and Cybernetics, Part A, vol. 34, no. 5, pages 630–637, Septembre 2004.

- [Schödl 00] A. Schödl, R. Szeliski, D. H. Salesin & I. Essa. *Video Textures*. In Siggraph'00, pages 489–498, 2000.
- [Sciavicco 87] L. Sciavicco & B. Siciliano. *A dynamic solution to the inverse kinematic problem of redundant manipulators*. IEEE International Conference on Robotics and Automation, pages 1081–1086, Mars 1987.
- [Sciavicco 88] L. Sciavicco & B. Siciliano. *A solution algorithm to the inverse kinematic problem for redundant manipulators*. IEEE International Conference on Robotics and Automation, vol. 4, pages 403–410, Mars 1988.
- [Sciavicco 96] L. Sciavicco & B. Siciliano. *Modeling and Control of Robot Manipulators*. In McGraw-Hill, New-York, 1996.
- [Sharon 05] D. Sharon & M. van de Panne. *Synthesis of Controllers for Stylized Planar Bipedal Walking*. ICRA 2005 : International Conference on Robotics and Automation, pages 18–22, Avril 2005.
- [Shin 01] H. J. Shin, J. Lee, S. Y. Shin & M. Gleicher. *Computer puppetry : An importance-based approach*. ACM Transactions on Graphics, vol. 20, no. 2, pages 67–94, 2001.
- [Shin 03] H. J. Shin, L. Kovar & M. Gleicher. *Physical Touch-Up of Human Motions*. In Pacific Conference on Computer Graphics and Applications, pages 194–203, 2003.
- [Shoemake 85] K. Shoemake. *Animating rotation with quaternion curves*. In SIGGRAPH'85 : Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques, pages 245–254. ACM Press, 1985.
- [Siciliano 91] B. Siciliano & J.-J. Slotine. *A general framework for managing multiple tasks in highly redundant robotic systems*. In ICAR'91, volume 2, page 1211–1215, 1991.
- [Silaghi 98] M-C. Silaghi, R. Plankers, R. Boulic, P. Fua & D. Thalmann. *Local and Global Skeleton Fitting Techniques for Optical Motion Capture*. In CAPTECH'98 : Proceedings of the International Workshop on Modeling and Motion Capture Techniques for Virtual Environments, pages 26–40. Springer-Verlag, 1998.
- [Sturman 86] D. Sturman. *Interactive keyframe animation 3-D articulated models*. In Graphics Interface'86, Tutorial on Computer Animation, 1986.
- [Sulejmanpašić 05] A. Sulejmanpašić & J. Popović. *Adaptation of performed ballistic motion*. ACM Transactions on Graphics, vol. 24, no. 1, pages 165–179, 2005.
- [Sun 01] H. C. Sun & D. N. Metaxas. *Automating gait generation*. In SIGGRAPH'01 : Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pages 261–270. ACM Press, 2001.
- [Tak 00] S. Tak, O-Y. Song & H-S. Ko. *Motion Balance Filtering*. Computer Graphics Forum, vol. 19, no. 3, 2000.
- [Tak 02] S. Tak, O-Y Song & H-S. Ko. *Spacetime Sweeping : An Interactive Dynamic Constraints Solver*. In CA'02 : Proceedings of the Computer Animation, pages 261–270. IEEE Computer Society, 2002.

- [Tolani 00] D. Tolani, A. Goswami & N. I. Badler. *Real-Time Inverse Kinematics Techniques for Anthropomorphic Limbs*. Graphical Models, vol. 62, no. 5, pages 353–388, 2000.
- [Unuma 91] M. Unuma & R. Takeuchi. *Generation of human motion with emotion*. In Proceedings of Computer Animation'91, pages 77–88, 1991.
- [Unuma 93] M. Unuma & R. Takeuchi. *Generation of human walking motion with emotion for computer animation*. In Trans. Inst. Electron. Inf. Commun. Eng., Août 1993.
- [Unuma 95] M. Unuma, K. Anjyo & R. Takeuchi. *Fourier Principles for Emotion-based Human Figure Animation*. In Proceedings of SIGGRAPH'95, Computer Graphics Proceedings, Annual Conference Series, pages 91–96, Août 1995.
- [van de Panne 90] M. van de Panne, E. Fiume & Z. Vranesic. *Reusable motion synthesis using state-space controllers*. In SIGGRAPH'90 : Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques, pages 225–234, Dallas, TX, USA, 1990. ACM Press.
- [van de Panne 97] M. van de Panne. *From Footprints to Animation*. Computer Graphics Forum, vol. 16, no. 4, pages 211–224, 1997.
- [Vieilledent 01a] S. Vieilledent, Y. Kerlirzin, S. Dalbera & A. Berthoz. *The curvature of locomotor paths determines the walking velocity in human*. Archives of Physiology, Biochemistry, vol. 109, page 50, 2001.
- [Vieilledent 01b] S. Vieilledent, Y. Kerlirzin, S. Dalbera & A. Berthoz. *Relationship between velocity and curvature of a locomotor trajectory in human*. Neuroscience Letters, vol. 305, pages 65–69, 2001.
- [Villard 95] D. Villard. *D'un formalisme énergétique à la résolution des équations du mouvement : l'usage de la différentiation et des graphes*. Thèse, Université de Rennes 1, 1995.
- [Viviani 82] P. Viviani & C. A. Terzuolo. *Trajectory determines movement dynamics*. In Neurosciences, volume 7, pages 431–437, 1982.
- [Viviani 95] P. Viviani & T. Flash. *Minimum-jerk, two-thirds power law, and isochrony : Converging approaches to movement planning*. Journal of Experimental Psychology : Human Perception and Performance, vol. 21, no. 1, pages 32–53, 1995.
- [Vukobratovic 90] M. Vukobratovic. Biped locomotion. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1990.
- [Wang 91] L.-C. T. Wang & C. C. Chen. *A Combined Optimization Method for Solving the Inverse Kinematics Problem of Mechanical Manipulators*. IEEE Transactions on Robotics and Applications, vol. 7, no. 4, pages 489–499, Août 1991.
- [Wang 99] X.G. Wang. *A behavior-based inverse kinematics algorithm to predict arm prehension for computer-aided ergonomic evaluation*. Journal of biomechanics, vol. 32, no. 5, pages 453–460, Mai 1999.

- [Wang 03] J. Wang & B. Bodenheimer. *An evaluation of a cost metric for selecting transitions between motion segments*. In SCA'03 : Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 232–238, San Diego, California, 2003. Eurographics Association.
- [Welman 93] C. Welman. *Inverse Kinematics and Geometric Constraints For Articulated Figure Manipulation*. Thèse, Simon Fraser University, 1993.
- [Whitney 69] D. E. Whitney. *Resolved motion rate control of manipulators and human prostheses*. IEEE Transactions on Man-Machine Systems, vol. MMS-10, no. 2, pages 47–53, Juin 1969.
- [Wicke 03] J. Wicke & B. Lopers. *Validation of the volume function within Jensen's elliptical cylinder model*. Journal of Applied Biomechanics, vol. 19, no. 1, pages 3–12, Février 2003.
- [Wiley 97] D. J. Wiley & J. K. Hahn. *Interpolation Synthesis of Articulated Figure Motion*. IEEE Computer Graphics and Applications, vol. 17, no. 6, pages 39–45, Novembre 1997.
- [Witkin 88] A. Witkin & M. Kass. *Spacetime Constraints*. Proceedings of SIGGRAPH'88 on Computer Graphics, vol. 22, no. 4, pages 159–168, Août 1988.
- [Witkin 95] A. P. Witkin & Z. Popović. *Motion warping*. In Proceedings of SIGGRAPH'95, pages 105–108, Août 1995.
- [Wolovich 84] W. A. Wolovich & H. Elliot. *A computational technique for inverse kinematics*. In Proceedings of the 32th Conference on Decision and Control, pages 1359–1362, Décembre 1984.
- [Woodworth 99] R. S. Woodworth. *The accuracy of voluntary movement*. In Psychological review 3, pages 1–106, 1899.
- [Yahia 89] H. Yahia & A. Gagalowicz. *Interactive animation of object orientations*. In Proceedings of the 2nd International Conference. Pixim 89, pages 265–275, September 1989.
- [Yamane 03a] K. Yamane & Y. Nakamura. *Dynamics filter - Concept and implementation of online motion generator for human figures*. IEEE Transactions on robotics and automation, vol. 19, no. 3, pages 421–432, Juin 2003.
- [Yamane 03b] K. Yamane & Y. Nakamura. *Natural Motion Animation through Constraining and Deconstraining at Will*. IEEE Transactions on Visualisation and Computer Graphics, vol. 9, no. 3, pages 352–360, Septembre 2003.
- [Yamane 04] K. Yamane, J. Kuffner & J. K. Hodgins. *Synthesizing animations of human manipulation tasks*. In Proceedings of SIGGRAPH'04, New York, NY, 2004. ACM Press.
- [Yang 04] P-F. Yang, J. Laszlo & K. Singh. *Layered dynamic control for interactive character swimming*. In SCA'04 : Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pages 39–47, Grenoble, France, 2004. ACM Press.

- [Zatsiorsky 90] V. Zatsiorsky, V. Seluyanov & L.G. Chugunova. *Contemporary problems of Biomechanics*. In Methods of determining mass-inertial characteristics of human body segments. 273-291, Moscow : Mir publishers 1990.
- [Zeltzer 82] D. Zeltzer. *Motor Control Techniques for Figure Animation*. IEEE Computer Graphics and Applications, vol. 2, no. 9, pages 53–59, Novembre 1982.
- [Zeltzer 83] D. Zeltzer. *Knowledge-based animation*. In ACM SIGGRAPH/SIGART, pages 187–192, April 1983.
- [Zhao 94] J. Zhao & N. I. Badler. *Inverse Kinematics Positioning Using Nonlinear Programming for Highly Articulated Figures*. ACM Transactions on Graphics, vol. 13, no. 4, pages 313–336, Octobre 1994.
- [Zhao 96] X. Zhao, D. Tolani, B-J. Ting & N. I. Badler. *Simulating human movements using optimal control*. In Proceedings of the Eurographics Workshop on Computer Animation and Simulation'96, pages 109–120, Poitiers, France, Août 1996. Springer-Verlag New York, Inc.
- [Zordan 02] V. B. Zordan & J. K. Hodgins. *Motion capture-driven simulations that hit and react*. In SCA'02 : Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 89–96, San Antonio, Texas, Juillet 2002. ACM Press.
- [Zordan 05] V. B. Zordan, A. Majkowska, B. Chiu & M. Fast. *Dynamic response for motion capture animation*. ACM Transactions on Graphics : Proceedings of Siggraph'05, vol. 24, no. 3, pages 697–701, 2005.



# Table des figures

1.1	Représentation hiérarchique du squelette . . . . .	10
1.2	Transformations locales et globales . . . . .	12
1.3	Problème de morphologies différentes . . . . .	13
1.4	Représentation normalisée du squelette . . . . .	14
1.5	Représentation normalisée de la jambe . . . . .	15
1.6	Etude de Marey sur le saut [Marey 94] . . . . .	18
1.7	Capture basée vidéo [Niskanen 05] . . . . .	19
1.8	Systèmes de capture de mouvements . . . . .	20
1.9	Exemple de mouvement caractérisé par la loi de Fitts [Fitts 54] . . . . .	22
1.10	Exemple de graphe de postures de contrôle pour la course d'un person- nage [Hodgins 95] . . . . .	26
1.11	Planification de mouvements [Laumond 05] . . . . .	27
1.12	Graphe de mouvements . . . . .	28
1.13	Déformation dynamique du temps ( <i>Dynamic Time Warping</i> ) . . . . .	30
1.14	Une trajectoire articulaire décrite à partir de quatre trajectoires de référence.	31
1.15	Cinématique inverse sur une chaîne à deux ou trois segments . . . . .	33
1.16	Cinématique inverse avec contrôle par priorité ou mélange des contraintes .	35
1.17	Utilisation de la cinétique inverse pour contrôler la position du centre de masse . . . . .	36
1.18	Exemple d'utilisation du CCD . . . . .	37
1.19	Lenteur de la convergence du CCD près de la cible . . . . .	38
1.20	Importance associée aux extrémités [Shin 01] . . . . .	39
1.21	Espace accessible pour les bras et la racine [Shin 01] . . . . .	39
1.22	Discontinuités à résoudre par les cartes de déplacements . . . . .	42
1.23	Filtrage des cartes de déplacements . . . . .	42
1.24	Adaptation d'un mouvement de marche à différents squelettes . . . . .	43
1.25	Utilisation d'un squelette intermédiaire pour l'adaptation morphologique . .	44
1.26	Utilisation de splines hiérarchiques (a) pour effectuer une adaptation mor- phologique (b) en temps-réel [Lee 99] . . . . .	44
1.27	Maintien de l'équilibre en conservant le ZMP dans le polygone de sustenta- tion [Tak 00] . . . . .	46
2.1	Utilisation de l'adaptation de mouvements pour l'animation comportementale	52
2.2	Chaîne d'animation proposée par Stéphane Ménardais . . . . .	53
2.3	Chaîne complète d'adaptation des mouvements . . . . .	54

2.4	Exemple d'adaptation cinématique sans gestion de l'équilibre . . . . .	56
3.1	Deux niveaux de représentation des contraintes . . . . .	60
3.2	Utilisation de contraintes de distance . . . . .	62
3.3	Utilisation de contraintes d'espace . . . . .	63
3.4	Exemple d'une parade effectuée avec un point quelconque de l'avant-bras . .	64
3.5	Décalage utilisé pour spécifier la position des contraintes . . . . .	65
3.6	Portée des contraintes sur la chaîne cinématique . . . . .	65
3.7	Deux modes de contrôle des priorités . . . . .	66
3.8	Coefficient d'activation des contraintes . . . . .	67
3.9	Entrées / sorties du logiciel de spécification des contraintes intrinsèquement liées au mouvement . . . . .	68
3.10	Conversion d'une contrainte de distance . . . . .	69
3.11	Conversion d'une contrainte d'espace . . . . .	70
3.12	Conversion d'une contrainte d'orientation . . . . .	71
4.1	Entrées / sorties du module d'adaptation cinématique . . . . .	73
4.2	Squelette normalisé . . . . .	74
4.3	Groupes du corps humain utilisés pour l'adaptation . . . . .	75
4.4	Hierarchie des groupes du corps humain . . . . .	76
4.5	Mauvaise répartition de l'adaptation en utilisant le CCD. . . . .	77
4.6	Groupes pris en compte de la racine aux extrémités de la chaîne cinématique.	78
4.7	Groupes pris en compte des extrémités à la racine de la chaîne cinématique	79
4.8	Adaptation des deux segments du bras indépendamment l'un de l'autre . . .	81
4.9	Adaptation d'un segment normalisé en mode priorité de contraintes . . . . .	82
4.10	Déplacement de la racine pour résoudre les contraintes . . . . .	84
4.11	Utilisation d'un pourcentage pour déplacer la racine de la hiérarchie . . . .	85
4.12	Utilisation des ischio-jambiers . . . . .	86
4.13	Utilisation de la souplesse d'un personnage . . . . .	86
4.14	Exemple d'adaptation d'une posture au sol . . . . .	87
4.15	Algorithme général incluant l'adaptation au sol . . . . .	88
4.16	Adaptation rapide au sol . . . . .	89
4.17	Schéma général du module d'adaptation complète au sol . . . . .	90
4.18	Schéma général du module d'adaptation complète au sol . . . . .	90
4.19	Respect de deux contraintes antagonistes . . . . .	91
4.20	Calcul de la convergence pour deux contraintes antagonistes . . . . .	93
4.21	Adaptation à des contraintes placées sur la peau des avant-bras . . . . .	93
4.22	Adaptation à des contraintes non accessibles . . . . .	94
4.23	Différentes configurations utilisées pour les calculs de performance . . . . .	94
5.1	La cinétique permet de prendre en compte les notions d'équilibre . . . . .	99
5.2	Schéma général de l'adaptation cinétique . . . . .	100
5.3	Modélisation du segment par un cône tronqué . . . . .	101
5.4	Hierarchie des centres de masse . . . . .	103
5.5	Stratégies de l'adaptation cinétique. . . . .	105
5.6	Ordre de prise en compte des groupes . . . . .	106
5.7	Adaptation d'un groupe . . . . .	107

5.8	Adaptation cinétique du tronc . . . . .	108
5.9	Adaptation des membres en deux étapes . . . . .	109
5.10	Position du centre de masse d'un membre . . . . .	109
5.11	Distance maximale entre le centre de masse et l'épaule . . . . .	110
5.12	Adaptation de la jambe pour respecter le centre de masse . . . . .	111
5.13	Algorithme Push-Pull utilisé pour le calcul de la radiosité . . . . .	113
5.14	Comparaison d'une posture avec ou sans adaptation cinétique . . . . .	116
5.15	Utilisation des adaptations cinématique et cinétique de la racine . . . . .	116
5.16	Comparaison d'une posture réelle avec celle fournie par notre adaptation cinématique . . . . .	117
5.17	Comparaison d'une posture réelle avec celle fournie par notre adaptation cinématique . . . . .	118
6.1	Schéma général du respect de la dynamique . . . . .	123
6.2	Trajectoire du centre de masse lors d'un saut . . . . .	124
6.3	Schéma général de la représentation des données . . . . .	126
6.4	Deux niveaux de représentation des contraintes . . . . .	127
6.5	Adaptation du saut à différentes vitesses d'impulsion . . . . .	128
6.6	Représentation du saut par un polynôme de degré 7 . . . . .	129
6.7	Positions possibles du point $P'_2$ en ayant une accélération inférieure à la gravité	130
6.8	Explication des zones automatiquement supprimées . . . . .	131
6.9	Positions possibles du point $P'_2$ avec une vitesse finale divisée par 2 . . . . .	131
6.10	Utilisation de la règle de proximité du point d'origine . . . . .	132
6.11	Adaptation du saut à différentes vitesses d'impulsion . . . . .	133
6.12	Exemple d'adaptation de saut . . . . .	134
7.1	Chaîne d'animation complète MKM . . . . .	137
7.2	Interface du logiciel S4DMaker . . . . .	138
7.3	Contrôle interactif de 40 personnages avec adaptation cinématique et cinétique	139
7.4	Cours de karaté entre deux équipes de quatre personnages . . . . .	139
7.5	Ajout de masses supplémentaires sur le personnage . . . . .	140
7.6	Modèle de préhension . . . . .	141
7.7	Modèle de tir au handball . . . . .	141
7.8	Modèle générique d'adaptation d'orientation . . . . .	142
7.9	Deux modes de communication entre le comportement et le contrôle . . . . .	148
A.1	Exemple de parade d'un coup de pied avec l'avant-bras . . . . .	151
A.2	Ajout de la contrainte <i>ctrParade</i> dans le S4DMaker . . . . .	152
A.3	Séquence d'images de la parade du coup de pied . . . . .	153
A.4	Application interactive où l'utilisateur contrôle les poignets . . . . .	154
B.1	Respect d'une contrainte sur le poignet . . . . .	156
B.2	Respect d'une contrainte sur le bras . . . . .	156
B.3	Adaptation d'une contrainte sur l'avant-bras lorsque la cible est atteignable	157
C.1	Calcul du COM dans un segment de longueur variable . . . . .	159





---

**Adaptation interactive et performante  
des mouvements d'humanoïdes synthétiques :  
aspects cinématique, cinétique et dynamique.**

---

De nos jours, la capture de mouvements est une technique très répandue pour obtenir des animations réalistes d'humanoïdes synthétiques. Cependant, ces mouvements doivent être adaptés à de nouvelles contraintes comme la morphologie et l'environnement du personnage afin de pouvoir les utiliser dans toutes les situations. Le problème est alors de les modifier tout en préservant leurs propriétés intrinsèques. Nous proposons donc une méthode d'adaptation de ces mouvements à de nouvelles contraintes en vérifiant les aspects cinématique, cinétique et dynamique. Ces travaux répondent aux problèmes suivants : trouver une posture réaliste parmi celles qui vérifient les contraintes, contrôler la position du centre de masse et adapter les postures en conséquence, enfin être capable d'appliquer ces méthodes sur de nombreux personnages en temps-réel. Ce travail peut être utilisé dans les applications interactives par exemple dans des domaines tels que l'éducation, l'audiovisuel et le sport.

**Mots clés** : Animation temps-réel d'humanoïdes, adaptation de mouvements, cinématique inverse, cinétique inverse, dynamique, représentation de contraintes

---

**Interactive and fast motion adaptation  
of human-like synthetic characters :  
kinematic, kinetic and dynamic considerations.**

---

Nowadays, motion capture is widely used to animate synthetic human-like figures. However, the motion must be adapted to new constraints, such as the morphology and the environment of the synthetic character. Indeed, adaptation prevents from capturing motions for all the possible situations. However, the problem consists in modifying a motion while preserving its intrinsic properties. We propose a method that adapts a motion to new constraints while taking kinematics, kinetics and dynamics into account. This thesis addresses the following problems : finding a realistic solution among all the possible postures that verify the above constraints, controlling the position the center of mass and calculating a compatible posture, and being able to handle many characters in real-time with those methods. This work can consequently be used in video games and virtual reality that require interactions. The main applications are education, entertainment and sports.

**Keywords** : Real-time human animation, motion adaptation, inverse kinematics, inverse kinetics, dynamics, constraints representation.