

SÉBASTIEN QUIRION

**Animation basée sur la physique :
Extrapolation de mouvements humains plausibles et
réalistes par optimisation incrémentale**

Thèse présentée
à la Faculté des études supérieures de l'Université Laval
dans le cadre du programme de doctorat en génie électrique
pour l'obtention du grade de Philosophiæ Doctor (Ph.D.)

FACULTÉ DE SCIENCES ET DE GÉNIE
UNIVERSITÉ LAVAL
QUÉBEC

2010

©Sébastien Quirion, 2010

Résumé

L'objectif de nos travaux est de faire la synthèse de mouvements plausibles et réalistes de marche humaine dans divers environnements de synthèse. Bien que la solution proposée puisse également s'appliquer aux autres mouvements de locomotion humains ou animaux, nos travaux traitent uniquement du problème de la marche humaine.

Afin de résoudre ce problème, nous avons développé une approche permettant de générer une multitude de variations d'une animation issue de capture de mouvement. Ces variations sont obtenues en adaptant le mouvement original à un environnement de synthèse dont les paramètres, tels que l'inclinaison du sol ou la courbure de la trajectoire, sont variés. Nous sommes donc en mesure de produire un mouvement de marche courbe ou de marche sur un plan incliné à partir d'un mouvement de marche en ligne droite sur un sol horizontal, ce que nous qualifions d'extrapolation de mouvement.

Une animation initiale, obtenue par capture de mouvement, est essentielle à la solution proposée. Adapter ce mouvement à un nouvel environnement de synthèse consiste essentiellement à ajuster les caractéristiques globales du mouvement, telles que l'orientation du personnage et sa vitesse de déplacement. Ce faisant, nous sommes en mesure de conserver les détails plus fins du mouvement qui lui confèrent son aspect humain, tels que le mouvement des bras ou la vitesse avec laquelle un pied entre en contact avec le sol. En conservant les détails fins du mouvement d'origine, la solution proposée assure un certain réalisme dans les mouvements synthétisés.

Dans la solution proposée, l'adaptation du mouvement initial est basée sur le paradigme des contraintes spatio-temporelles, où la synthèse du mouvement est posée comme un problème d'optimisation numérique. En plus d'être une formulation élégante du problème, ce paradigme est tout indiqué pour faire la synthèse de mouvements physiquement plausibles. En combinant ce paradigme avec l'utilisation d'une animation initiale issue de capture de mouvement, nous sommes en mesure de produire des animations de mouvements humains plausibles et réalistes.

En pratique, le problème d'optimisation sous-tendu par l'adaptation d'un mouvement par contraintes spatio-temporelles est fortement non linéaire et opère dans un espace à très grande dimensionnalité. Cette complexité peut fortement ralentir le processus d'optimisation et aller jusqu'à en empêcher la convergence. La solution proposée fait donc appel à plusieurs mécanismes afin de réduire cette complexité. Notons qu'aucun de ces mécanismes ne vient compromettre la polyvalence de l'approche, en limitant la complexité du modèle biomécanique du personnage par exemple. Parmi ces mécanismes, deux sont des contributions originales : une technique d'estimation rapide des forces de réaction du sol et une approche d'optimisation incrémentale. Ces deux mécanismes visent à simplifier le processus d'optimisation en fournissant une solution initiale très proche de la solution optimale.

La technique d'estimation des forces de réaction du sol sert à donner à ces paramètres une valeur initiale qui est relativement proche de leur valeur optimale, ce qui simplifie significativement la tâche d'optimisation subséquente. Cette technique consiste à trouver, pour les phases de support double, les forces de réaction du sol minimisant l'effort interne du personnage. Ce problème peut être exprimé comme une séquence de sous-problèmes de programmation quadratiques. Cette formulation est un aspect central de notre contribution et elle permet d'atteindre la solution très efficacement.

L'approche d'optimisation incrémentale proposée s'inspire des méthodes de continuation. Le mouvement original est considéré comme une solution, un mouvement optimal, pour l'environnement de capture. L'environnement de synthèse est ensuite modifié graduellement, en augmentant l'inclinaison du sol par petits incréments par exemple. À chaque incrément, un nouveau mouvement optimal est trouvé en utilisant la solution de l'incrément précédent comme point de départ. On procède de la sorte jusqu'à l'obtention du mouvement désiré pour l'environnement de synthèse considéré. Si les incréments sont suffisamment petits, la différence entre deux problèmes d'optimisation consécutifs sera petite et il en sera de même pour leur optimum respectif.

Abstract

Our research is directed towards synthesizing plausible and realistic human walking motion in various synthesis environments. Although the solution we propose could be applied to any other kind of locomotion, human or animal, our work exclusively addresses the problem of human walking.

In order to solve this problem, we developed an approach that allows the creation of numerous variations of an original motion obtained via a motion capture system. These variations are created by adapting the original motion to a synthesis environment for which the parameters, such as ground plane inclination or trajectory curvature, can be varied. We are thus able to generate an uphill walking motion or a curved walking motion from an original motion of a person walking straight on a level ground. We use the term *motion extrapolation* to refer to this process.

The original motion, obtained via motion capture, is a fundamental element of the proposed solution. Adaptation of this motion to a new synthesis environment is performed essentially by adjusting the global aspects of the motion, such as the character's orientation or velocity. Accordingly, this approach preserves the finer details of the original motion, such as arm motion irregularities or the speed at which a foot strikes the floor, which greatly impact the naturalness of the synthesized motion.

The proposed solution performs the adaptation of the original motion using the spacetime constraints paradigm, in which the process of motion synthesis is cast as a numerical optimization problem. The spacetime constraints paradigm was originally designed to synthesize physically valid human motion and is an elegant formulation of that problem. By combining the use of spacetime constraints with motion capture data, we are able to synthesize plausible and realistic human motions.

In practice, when using spacetime constraints on a complex character, the underlying optimization problem is highly nonlinear and operates in a space of very high dimensionality. This complexity may severely hinder the capacity of the optimization

process to converge to a solution within a reasonable delay and, ultimately, it may prevent convergence altogether. Therefore, the proposed solution includes a number of techniques to reduce this complexity. We would like to emphasize that none of these techniques limit the scope of application of the proposed solution in any way, such as limiting the complexity of the biomechanical model for instance. Two of these techniques constitute novel contributions to this field of research : a fast ground reaction force estimation technique and an incremental optimization process. Both techniques were designed to simplify the optimization problem by providing an initial guess as close as possible to the optimal solution.

The ground force estimation technique provides initial values for the ground force parameters of the motion that are close to their optimal values, which significantly simplifies the subsequent optimization process. This technique consists in finding, for a character in double limb support, the combination of ground reaction forces minimizing the internal forces of the character. Finding these ground reaction forces is an optimization problem which can be expressed as a succession of quadratic programming subproblems. This formulation is central to our contribution and its implementation converges to the optimal solution very efficiently.

The incremental optimization process is inspired from continuation methods. The original motion is taken to be a solution, an optimal motion, with respect to the capture environment. The synthesis environment is then altered gradually, by increasing the ground plane inclination by small increments for example. After each such increment, a new optimal motion is obtained using the optimal motion of the previous increment as a starting point. The process is iterated until an optimal motion is found for a target synthesis environment. Given that the increments are sufficiently small, the difference between successive optimization problems will be small, as will be the difference between their respective optima.

Avant-propos

J'aimerais d'abord remercier mon directeur de recherche, Denis Laurendeau, pour son intérêt envers ce projet de recherche, son attitude rassurante quant aux divers problèmes et ralentissements rencontrés lors de l'exécution du projet et, surtout, pour son appui indéfectible. Il va sans dire que ce projet n'aurait jamais pu être mené à bien si ce n'avait été de son soutien. Je voudrais aussi remercier mon codirecteur, Mario Marchand. Lors de nos interactions, il a su me transmettre deux notions très importantes en recherche scientifique comme dans bien d'autres aspects de la vie : la rigueur et la passion. On dit souvent que ce qui mérite d'être fait mérite d'être bien fait, on pourrait ajouter que le faire avec passion en fait une aventure plutôt qu'une tâche ardue.

Étant très proche de ma famille et de mes amis, je dois avouer qu'ils ont tous joué un rôle important dans le succès de cette entreprise. Que ce soit par leur soutien moral ou par l'intérêt réel qu'ils portent envers ce projet de recherche, ils ont su entretenir mon intérêt pour cette recherche et ma motivation à mener ce projet à terme. J'ai la chance d'être bien entouré et j'en suis reconnaissant. Je réserve évidemment un merci tout spécial à ma copine Geneviève qui a partagé avec moi les meilleurs moments comme les plus difficiles au long de ces années d'études supérieures.

Plusieurs collègues ont aussi appuyé mes efforts au cours des dernières années. Plus particulièrement, mes nombreuses discussions avec Philippe Lambert ont été un réel moteur dans la progression de mes travaux. Plusieurs autres collègues ont contribué à mes travaux, de près ou de loin, dont Frédéric Jean, Martin Lavallière, Philippe Corbeil, Martin Otis et Philippe Cardou. Je voudrais aussi souligner l'assistance du personnel du LVSN, plus particulièrement Denis Ouellet et Annette Schwerdtfeger, grâce à qui les problèmes quotidiens sont résolus rapidement.

Je voudrais finalement remercier le conseil de recherche en sciences naturelles et en génie du Canada (CRSNG), ayant financé mes recherches par le biais d'une bourse d'études supérieures, et l'organisme Precarn incorporated, dont les bourses ont également contribué au financement de mes études doctorales.

*À Mathis et Olivier qui, contrairement à mes
personnages, n'ont pas mis cinq ans à apprendre à
marcher !*

*The only reason for time is so that
everything doesn't happen at once.
- Albert Einstein*

*No one wants to learn from mistakes,
but we cannot learn enough from
successes to go beyond the state of the
art.
- Henry Petroski*

Table des matières

Résumé	ii
Abstract	iv
Avant-propos	vi
Table des matières	viii
Liste des tableaux	xi
Table des figures	xii
1 Introduction	1
1.1 Objectifs	2
1.1.1 Caractérisation du problème	3
1.2 Contexte	6
1.2.1 Historique	6
1.2.2 État de l'art	10
1.3 Solution proposée	19
1.3.1 Contributions	22
1.3.2 Omissions	23
2 Notation et fondements mathématiques	25
2.1 Notation	25
2.2 Types de vecteurs	26
2.2.1 Vecteur libre	26
2.2.2 Vecteur cartésien/homogène	27
2.2.3 Vecteur glissant	28
2.2.4 Vecteur spatial	28
2.3 Types de coordonnées	29
2.3.1 Coordonnées cartésiennes	29
2.3.2 Coordonnées homogènes	30
2.3.3 Coordonnées plückeriennes	30

2.3.4	Coordonnées spatiales	31
2.3.5	Coordonnées généralisées	34
2.4	Transformations rigides	35
2.4.1	Coordonnées cartésiennes	36
2.4.2	Coordonnées homogènes	38
2.4.3	Coordonnées spatiales	39
3	Modélisation	41
3.1	Modélisation du personnage	41
3.1.1	Corps rigide	41
3.1.2	Articulations	43
3.1.3	Système multicorps	44
3.1.4	Contact avec le sol	49
3.2	Environnement de synthèse	52
3.3	Paramétrisation du mouvement	53
3.3.1	Coordonnées généralisées	53
3.3.2	Coordonnées cartésiennes et coordonnées homogènes	54
3.3.3	Coordonnées spatiales	56
3.4	Dynamique des systèmes multicorps	59
3.4.1	Inertie spatiale	59
3.4.2	Équations de mouvement	59
3.4.3	Dynamique inverse	60
4	Estimation des forces de contact	63
4.1	Problématique	65
4.1.1	Somme des forces de réaction du sol	65
4.2	Approche d'optimisation imbriquée	66
4.2.1	Fonction de coût quadratique	68
4.2.2	Contraintes linéaires	70
4.2.3	Boucle externe	75
4.3	Résultats	76
4.3.1	Qualité des résultats	78
4.3.2	Performance	84
5	Synthèse de mouvements	88
5.1	Optimisation d'un mouvement	89
5.1.1	Fonctions de coût	91
5.1.2	Contraintes cinématiques	95
5.1.3	Contraintes cinétiques	99
5.1.4	Contraintes dynamiques	103
5.1.5	Paramètres du problème d'optimisation	104

5.2	Production du mouvement tuteur	106
5.2.1	Simplification	107
5.2.2	Correction	110
5.3	Synthèse de nouveaux mouvements	110
5.3.1	Problème de base	113
5.3.2	Optimisation incrémentale	113
5.4	Résultats	121
5.4.1	Création du mouvement tuteur	123
5.4.2	Extrapolation de mouvements	130
5.4.3	Discussion	140
6	Conclusion	143
6.1	Réalisations	144
6.2	Travaux futurs	146
	Bibliographie	149
A	Développement de la méthode d'estimation des forces de contact	158
A.1	Équations de \hat{H} , \hat{m} et b	158
A.2	Équations des contraintes linéaires	160
A.2.1	Pied droit	161
A.2.2	Pied gauche	162
A.2.3	Contraintes combinées	163

Liste des tableaux

2.1	Notation utilisée dans cet ouvrage	26
4.1	Erreurs RMS, maximum et RMS relative de notre approche d'estimation des forces de contact par rapport aux forces mesurées par des plateformes de force. La dernière colonne présente à titre comparatif l'erreur RMS relative obtenue par Ren et coll.[70].	83
5.1	Description des divers paramètres utilisés dans la caractérisation d'un problème d'optimisation de mouvement.	105
5.2	Caractérisation du problème de réduction du nombre de degrés de liberté. Puisqu'ils s'agit d'un problème simplifié, certains paramètres ne sont pas requis.	109
5.3	Caractérisation du problème de correction de données issues de capture de mouvement. La valeur initiale des forces de contacts est donnée par notre approche d'estimation des forces de contact, identifiée comme la fonction $EFC(Q'_{sim})$	111
5.4	Caractérisation de la version simplifiée du problème d'extrapolation de mouvement.	114
5.5	Caractérisation du problème de synthèse d'une marche courbe.	115
5.6	Caractérisation du problème intermédiaire lié à λ_i pour l'extrapolation d'une marche courbe.	116
5.7	Caractérisation du problème de synthèse d'une marche sur un plan incliné.	117
5.8	Caractérisation du problème intermédiaire lié à λ_i pour l'extrapolation d'une marche sur un plan incliné.	118

Table des figures

1.1	Pourcentage annuel d'articles scientifiques ayant les mots-clés <i>Physics-based animation</i> et <i>Physics-based character animation</i> parmi les publications scientifiques indexées par Google Scholar TM dans la catégorie <i>ingénierie, informatique et mathématique</i> entre 1991 et 2009.	2
1.2	De gauche à droite, un système multicorps représentant un personnage, une de ses articulations et un des corps rigides qui le composent. Ces articulations correspondent à celles généralement utilisées en synthèse de mouvement humains par des approches basées sur la physique. . . .	3
1.3	Illustration du processus d'animation par capture de mouvement, de la séance de capture jusqu'à la transposition du mouvement sur le personnage virtuel.	7
1.4	Illustration du processus d'animation par simulation physique sur un personnage simplifié. À chaque instant, la configuration est déterminée par une simulation appliquant à la configuration précédente les forces données.	8
1.5	Illustration du paradigme des contraintes spatiotemporelles. (a) Les configurations à atteindre aux temps donnés sont posées comme contraintes. (b) Une solution intermédiaire hypothétique ne respectant pas les équations de mouvement de la fusée. (c) Une solution finale hypothétique atteinte optimisant tous les paramètres, soient les positions, orientations et poussées de la fusée à chaque instant. Cette solution respecte toutes les contraintes et minimise une fonction de coût, telle que la consommation de carburant.	9
1.6	Illustration du processus de construction d'un graphe de mouvement et de son utilisation. Les configurations des différents mouvements de la banque de mouvements (en haut à gauche) sont reliées pour former un graphe de mouvements (en haut à droite). On peut ensuite parcourir ce graphe pour faire la synthèse de nouveaux mouvements (en bas).	12

2.1	Le champ de vecteurs d'une translation pure d'un corps rigide est constant (à gauche). Ce champ peut être représenté adéquatement par un seul vecteur libre (à droite). Dans la figure, les flèches rouge et verte représentent le même vecteur libre puisqu'elles sont de même direction et de même module.	27
2.2	Puisqu'elles sont de même direction, de même module et qu'elles se trouvent sur la même droite de support, les flèches rouge et verte représentent le même vecteur glissant.	28
2.3	Les coordonnées plückeriennes $(\vec{d}, \vec{m})^t$ sont utilisées pour décrire une droite dans \mathbb{R}^3 . \vec{d} est un vecteur de module arbitraire (non-nul) parallèle à la droite et \vec{p} rejoint un point arbitraire sur la droite. \vec{m} est le résultat du produit vectoriel $\vec{p} \times \vec{d}$	31
2.4	Les bases plückeriennes définies sur \mathbb{M}^6 (à gauche) et sur \mathbb{F}^6 (à droite).	33
2.5	Les coordonnées du point p changent selon le repère dans lequel il est exprimé.	35
2.6	En haut, le repère 1 subit une rotation d'angle θ autour de l'axe z_1 puis une translation $\vec{\delta}$ pour former le repère 3. En bas, un vecteur exprimé dans le repère 1 subit une rotation d'angle $-\theta$ autour de l'axe z_1 , suite à quoi on lui additionne $-\vec{\delta}$ afin d'obtenir son expression dans le repère 3. Soulignons que le vecteur de translation $\vec{\delta}$ a une orientation différente dans l'image étant donné que cette image est orientée selon le repère 3.	37
3.1	Illustration d'un ellipsoïde de demi-axes a , b et c . Dans cet exemple, les demi-axes b et c sont égaux et inférieurs à a , il s'agit donc d'un ellipsoïde prolate.	42
3.2	Quelques exemples d'articulations couramment utilisées pour modéliser des systèmes mécaniques et biomécaniques. Le corps rigide vert représente le parent, par rapport auquel l'enfant se déplace selon les DDLs de l'articulation. De gauche à droite et de haut en bas nous présentons une articulation prismatique (1 DDL de translation), une articulation rotoïde (1 DDL de rotation), un joint de cardan (2 DDLs de rotation) et une rotule (3 DDLs de rotation).	43
3.3	Dans notre implémentation, un joint de cardan (à gauche) est représenté par deux articulations rotoïdes dont les axes de rotation sont orthogonaux. Le corps rigide entre ces deux articulations est de longueur et de masse nulle (à droite).	44

- 3.4 Trois articulations rotoïdes imbriquées sont utilisées pour modéliser une articulation de type rotule. Dans leur configuration initiale (à gauche), les axes de rotation des articulations sont orthogonaux. Lorsqu'un blocage de cardan est atteint (à droite), les axes de l'articulation externe et de l'articulation interne sont colinéaires. Dans un contexte d'optimisation, cela signifie que les valeurs d'angle données aux articulations externe et interne s'additionnent. On peut donc attribuer une valeur arbitrairement grande à l'angle de l'articulation externe puisque l'angle de l'articulation interne peut toujours compenser pour cette valeur. Une infinité de ces combinaisons d'angles donnent donc une même orientation au corps enfant, un comportement qui peut nuire à la convergence de l'algorithme d'optimisation. 45
- 3.5 Un exemple de système multicorps (à gauche) et le graphe de ses connexions (à droite). Les points du graphe représentent les corps rigides et les liens représentent les articulations. 46
- 3.6 Illustration du graphe des connexions d'un système multicorps et des ensembles qui s'y rattachent. De gauche à droite et de haut en bas nous présentons Ω , l'ensemble de tous les corps rigides, $\phi(11)$, l'ensemble des enfants du corps 11, $\lambda(11)$, le parent du corps 11, $\varphi(11)$, l'ensemble des descendants du corps 11 et $\varpi(11)$, l'ensemble des ancêtres du corps 11. Rappelons que selon notre définition, l'ensemble des descendants et l'ensemble des ancêtres d'un corps i incluent le corps i . Bien que cela ne soit pas standard en théorie des graphes, cette définition nous permet de simplifier quelques équations. 47
- 3.7 L'articulation i est attachée au corps $\lambda(i)$ et au corps i . Le point rouge et le point bleu représentent le point d'attache sur le corps $\lambda(i)$ et sur le corps i respectivement. Notons que dans le cas d'une articulation rotoïde, ces deux points d'attache sont superposés puisque l'articulation a une longueur nulle. L'*orientation* des repères qui y sont attachés sera toutefois différente dans le cas général. 48
- 3.8 Relation entre le repère $\lambda(i)$ et le repère i . Une transformation rigide est appliquée au repère $\lambda(i)$ pour obtenir le repère intermédiaire i' , qui correspond à l'orientation de l'articulation i et à son point d'attache par rapport au corps $\lambda(i)$. Le repère i' est ensuite modifié par l'articulation i , ce qui correspond à une seconde transformation rigide, pour obtenir le repère i 50

3.9	Illustration de deux points de contact sur le pied d'un personnage. Ces points permettent notamment de contrôler différentes parties du pied de façon indépendante, particulièrement quant aux temps de contact. Ainsi, il est possible de permettre au talon de quitter le sol tout en forçant la pointe du pied à rester au sol.	51
3.10	Illustration de surfaces de contact entre un pied et le sol. Dans la figure de gauche, le talon et la pointe du pied sont en contact avec le sol et toute force de contact dont le centre de pression est contenu dans la surface rouge est valide. Dans les deux autres figures, seule la pointe du pied est en contact avec le sol. La surface de contact est limitée à un segment de droite lorsque la boîte englobante ne peut pénétrer dans le sol (au centre), alors qu'elle a une plus grande superficie lorsque la pénétration est permise (à droite). C'est le positionnement des points de contact qui détermine à quel point la boîte englobante peut pénétrer dans le sol. . .	52
4.1	Le vecteur de force de réaction du sol correspond à l'intégrale des forces appliquées par le sol à chaque point de la surface de contact.	64
4.2	Une force appliquée au point \vec{a} affecte le corps rigide de façon identique à une force et un couple appliqués au centre de masse. Cela correspond au changement de repère de la force f , du point \vec{a} au centre de masse du corps rigide.	64
4.3	(à gauche) Vecteur spatial de force de réaction du sol exprimé par rapport au centre de pression \vec{a} , où les couples en x et en y sont nuls par définition du centre de pression. (à droite) Le même vecteur, exprimé par rapport au repère inertiel, comporte généralement des couples non nuls autour des trois axes.	64
4.4	Trois paires de forces différentes dont la somme est égale. Bien qu'elles soient issues de différents efforts musculaires du personnage, leur effet net sur la variation de la quantité de mouvement et du moment angulaire du personnage est identique.	67
4.5	Illustration du partitionnement des différents articulations d'un modèle biomécanique. La racine du modèle est marquée d'un \times dans l'image de gauche. L'ensemble \mathbb{C} contient les articulations qui sont à la fois ancêtres du pied droit et du pied gauche, les ensembles \mathbb{A} et \mathbb{B} contiennent les articulations qui sont uniquement ancêtres du pied droit et du pied gauche respectivement et l'ensemble \mathbb{D} contient toutes les autres articulations du modèle biomécanique.	69
4.6	Différentes surfaces de contact décrites par l'intersection d'un prisme rectangulaire et d'un plan. De gauche à droite et de haut en bas, on présente la surface nulle, un point, un segment de droite, un triangle, un quadrilatère, un pentagone et un hexagone.	71

4.7	L'intersection de 6 demi-plans détermine la surface à l'intérieur de laquelle un centre de pression peut se trouver. La variation des positions relatives de ces demi-plans peut engendrer tous les types de surfaces présentées dans la figure 4.6. Un hexagone, un segment de droite et la surface nulle sont présentés ici en exemples.	71
4.8	Les vecteurs \vec{u} et \vec{v} représentent la position du centre de pression du pied droit et du pied gauche respectivement.	72
4.9	Illustration des repères d et α sur le pied droit. Les deux repères sont de même orientation, mais ont une origine distincte. L'origine du repère α est au centre géométrique du corps rigide. L'origine du repère d est centrée sur l'articulation de la cheville. On peut remarquer que la cheville ne se trouve pas à l'extrémité du corps rigide afin de représenter le talon. Pour le pied gauche, le repère d est remplacé par le repère g et le repère α est remplacé par le repère β	72
4.10	Illustration du vecteur $\vec{\delta}$ et de ses composantes δ_x , δ_y et δ_z	73
4.11	Illustration du modèle biomécanique utilisé pour évaluer notre approche d'estimation des forces de réaction du sol. La figure montre les articulations (à gauche), la longueur des segments (au centre) et la masse des segments (à droite) du modèle en question.	78
4.12	Pied gauche - Comparaison entre la position du centre de pression mesurée par les plateformes de force et celle estimée par l'approche proposée. Trois zones d'intérêt sont grossies afin de mieux juger la qualité de l'estimation.	79
4.13	Pied droit - Comparaison entre la position du centre de pression mesurée par les plateformes de force et celle estimée par l'approche proposée. Quatre zones d'intérêt sont grossies afin de mieux juger la qualité de l'estimation.	80
4.14	Comparaison des forces de réaction du sol mesurées et estimées. Les forces sont évaluées au centre de pression.	81
4.15	Comparaison du couple autour de l'axe z des forces spatiales de réaction du sol mesurées et estimées. Les forces spatiales sont évaluées au centre de pression, les couples en x et y sont donc nuls.	82
4.16	Performance comparative sur 100 essais des différentes formulations du problème d'estimation de forces de réaction du sol. Les valeurs données sont les temps moyens, et leurs écarts-types, pour l'estimation des forces de réaction du sol pour une trame de support double de la séquence de marche. Les temps sont donnés en millisecondes et ont été mesurés sur un processeur Intel Core i7 920.	86

5.1	Illustration de deux mesures de distances entre deux configurations : une somme de distances euclidiennes entre des marqueurs attachés aux corps rigides (à gauche) et une somme des différences des coordonnées généralisées de chaque configuration (au centre et à droite). Les deux exemples de droite montrent qu'une même différence en coordonnées généralisées peut avoir un impact plus ou moins grand selon l'articulation où cette différence est observée.	92
5.2	La fonction $(1 - \cos(2\theta))/2$ a une valeur de 0 lorsque l'angle θ est un multiple entier de π radians, ce qui correspond à avoir les trois axes de rotation orthogonaux. Par contre, elle a une valeur de 1 pour lorsque θ est à mi-chemin entre les multiples entiers de π radians, ce qui correspond à une singularité pour l'articulation à 3 DDLs de rotation.	94
5.3	Illustration d'un mouvement cyclique pour un personnage simplifié. Le mouvement contient 10 trames qui forment un cycle avec un chevauchement de deux trames. Ce chevauchement sert à valider la continuité entre la fin d'un cycle du mouvement et le début d'un nouveau cycle. Cela se fait par le biais de contraintes cinématiques sur les deux premières et les deux dernières configurations du mouvement. Deux trames suffisent à assurer la continuité du mouvement et de sa première dérivée temporelle.	96
5.4	Illustration du principe des temps de contact sur un pied ayant deux points de contact. La figure montre quatre configurations distinctes selon que les points de contact sont actifs (en vert) ou inactifs (en rouge). La matrice A donne les temps de contact correspondants à cette animation fictive de quatre trames. La matrice B donne les temps de contact du corps rigide du pied. Si au moins un des points de contact du pied est actif dans une trame donnée, le pied est considéré être en contact avec le sol dans cette trame.	98
5.5	Trois trame d'une animation où un point de contact glisse sur le sol alors qu'il est actif (en haut) et trois trames d'une animation où le glissement est corrigé (en bas).	99
5.6	Illustration du principe du cône de frottement. La force \vec{f}_1 est contenue dans le cône de frottement puisque la norme de sa composante parallèle au sol est au moins $1/\mu$ fois plus petite que la norme de la composante normale de cette force. Ce n'est pas le cas pour la force \vec{f}_2 qui se trouve à l'extérieur du cône de frottement. Dans cet exemple, le coefficient de frottement est de 0,5. En augmentant la valeur du coefficient de frottement μ , on augmente l'angle solide du cône.	101

5.7	Deux forces distinctes ont des centres de pression distincts. Dans l'exemple présenté, le centre de pression \vec{u} correspond à une force de contact valide puisqu'il est contenu dans l'intersection entre le sol et la boîte englobante du corps rigide. Par opposition, le centre de pression \vec{v} ne correspond pas à une force de contact valide. Rappelons qu'une telle pénétration d'un corps rigide dans le sol est possible lorsque les points de contact sont placés à l'intérieur de la boîte englobante et non sur sa surface.	102
5.8	Processus de création d'un mouvement tuteur à partir de données de capture de mouvement. Le mouvement capturé \mathbf{Q}_{cap} subit une étape de simplification afin de correspondre aux coordonnées articulaires du personnage de synthèse. Les forces de contact du mouvement résultant, soit \mathbf{Q}'_{sim} , sont ensuite estimées par la méthode décrite au chapitre 4. Finalement, une correction est appliquée au mouvement simplifié \mathbf{Q}'_{sim} et aux forces estimées \mathbf{F}_{est} pour produire le tuteur. Le mouvement dynamique défini par \mathbf{Q}_{tut} et \mathbf{F}_{tut} est un mouvement similaire au mouvement capturé, mais il est plausible pour le personnage de synthèse dans l'environnement de synthèse.	107
5.9	Illustration d'un personnage de capture typique (à gauche) et du personnage de synthèse qui lui serait associé (à droite). Pour les deux modèles, la figure donne le nombre de coordonnées articulaires de chaque corps par rapport à son corps parent. Le personnage de capture a 6 DDLs à la racine et 3 DDLs de rotation à chaque articulation, pour un total de 63 coordonnées articulaires. Sur le personnage de synthèse, les articulations sont limitées aux coordonnées articulaires d'un être humain véritable. Les coordonnées articulaires qui ne contribuent pas significativement aux mouvements désirés sont aussi éliminés sur le personnage de synthèse.	108
5.10	Exemple d'une homotopie H définie entre deux fonctions homotopes F et G . Les courbes de niveau de H sont tracées en pointillé.	111
5.11	À gauche, on illustre un personnage sur un sol incliné à θ radians. Cette situation correspond à l'illustration de droite, où le sol est le plan $z = 0$ et où le vecteur d'accélération gravitationnelle et la configuration du personnage ont subi une rotation de $-\theta$ radians.	116
5.12	Paramètres du personnage <i>HalfBoy</i> , qui représente la partie inférieure d'un garçon mesurant 1,20m et pesant 83kg. La figure de gauche indique le nombre de DDLs de chaque corps rigide par rapport à son corps parent. La figure du centre indique la longueur en cm de chaque segment telle que mesurée à partir des données de capture de mouvement. La figure de droite indique la masse de chaque segment en kg.	123
5.13	Mouvement initial utilisé dans notre étude de cas.	124

- 5.14 Illustration du personnage de capture simplifié (à gauche) et du personnage de synthèse qui lui est associé, *HalfBoy* (à droite). Pour les deux modèles, la figure donne le nombre de DDLs de chaque corps par rapport à son corps parent. Le personnage de capture a 6 DDLs à la racine et 3 DDLs de rotation à chaque articulation, pour un total de 30 coordonnées articulaires. Sur *HalfBoy*, les articulations sont limitées aux coordonnées articulaires d'un être humain véritable et les coordonnées articulaires qui ne contribuent pas significativement aux mouvements désirés sont également éliminées. 125
- 5.15 Cette figure montre le résultat de l'étape de simplification de données de capture de mouvements. En haut, nous pouvons y voir quelques configurations du mouvement d'origine (30 coordonnées articulaires) de deux points de vue différents et, en bas, nous pouvons voir les configurations correspondantes suite à la simplification (16 coordonnées articulaires). 126
- 5.16 Cette figure montre le résultat d'une simplification pour un cycle de marche de 28 trames (30Hz) passant de 63 coordonnées articulaires (à gauche) à 43 coordonnées articulaires (à droite). 127
- 5.17 Cette figure montre le résultat de l'étape de correction du mouvement simplifié. En haut, nous pouvons y voir quelques configurations du mouvement simplifié de deux points de vue différents et, en bas, nous pouvons voir les configurations correspondantes sur le mouvement tuteur produit par cette étape de correction (Q_{tut}). 128
- 5.18 Cette figure montre les forces de contact issues de l'étape de correction du mouvement simplifié. Les courbes donnent la position des centres de pression (ligne du haut) et les forces de contact (ligne du bas) pour le pied gauche (colonne de gauche) et le pied droite (colonne de droite) en fonction du temps. Les courbes tracées en noir représentent l'estimation de ces forces telles qu'obtenues par la technique d'estimation des forces de contact présentée au chapitre 4. 129
- 5.19 Un mouvement pour lequel deux configurations sont illustrées est transformé par une fonction Ψ . La barre rouge représente la hauteur de la racine par rapport au sol. Pour chaque configuration, une rotation est appliquée à la racine suite à quoi une translation est appliquée afin de préserver la hauteur de la racine par rapport au sol selon le nouvel angle. Cette dernière étape aide à assurer que le mouvement directeur soit suffisamment proche du sol pour permettre au personnage d'être en contact avec le sol dans le mouvement extrapolé. 131

5.20	En haut, nous présentons quelques trames du mouvement tuteur utilisé pour extrapoler un cycle de marche sur une pente descendante. Le mouvement directeur pour une valeur de $\lambda = 1$ et le mouvement extrapolé sont illustrés au milieu et au bas de la figure respectivement.	133
5.21	Cette courbe montre la progression de l'extrapolation du mouvement en fonction du temps de calcul pour la marche sur une pente descendante. L'axe des abscisses est gradué des valeurs de λ à gauche et des valeurs de l'inclinaison de la pente correspondantes à droite.	134
5.22	En haut, nous présentons quelques trames du mouvement tuteur utilisé pour extrapoler un cycle de marche sur une pente ascendante. Le mouvement directeur pour une valeur de $\lambda = 1$ et le mouvement extrapolé sont illustrés au milieu et au bas de la figure respectivement.	135
5.23	Cette courbe montre la progression de l'extrapolation du mouvement en fonction du temps de calcul pour la marche sur une pente ascendante. L'axe des abscisses est gradué des valeurs de λ à gauche et des valeurs de l'inclinaison de la pente correspondantes à droite.	136
5.24	Cette courbe montre la progression de l'extrapolation du mouvement en fonction du temps de calcul pour la marche sur une pente ascendante. À un angle proche de $-\pi/4$ radians, l'extrapolation de mouvement rencontre une impasse et la progression n'est plus possible. Sans mécanisme à cet effet, l'algorithme d'extrapolation serait alors coincé dans une boucle sans fin.	136
5.25	À gauche, la figure illustre la relation entre la longueur d'un arc de cercle, l'angle qu'il sous-tend et la longueur du rayon du cercle. À droite, la figure illustre que l'angle θ correspond également à l'angle entre les tangentes aux deux extrémités de l'arc de cercle.	137
5.26	À vol d'oiseau, un mouvement pour lequel les configurations illustrées sont transformées par une fonction Φ . De gauche à droite, la figure présente le mouvement original, le segment de droite de longueur d_T , la paramétrisation de la racine d'une configuration, l'application de cette paramétrisation sur l'arc de cercle de longueur d_T et les configurations finales suite à l'ajustement de l'orientation de chaque configuration. . .	138
5.27	En haut, nous présentons quelques trames du mouvement tuteur utilisé pour extrapoler un cycle de marche suivant une trajectoire courbe. Le mouvement directeur pour une valeur de $\lambda = 1$ et le mouvement extrapolé sont illustrés au milieu et au bas de la figure respectivement.	139
5.28	Cette courbe montre la progression de l'extrapolation du mouvement en fonction du temps de calcul pour la marche suivant une trajectoire courbe. L'axe des abscisses est gradué des valeurs de λ à gauche et des valeurs des angles de courbure correspondantes à droite.	140

Chapitre 1

Introduction

L'intérêt envers l'animation basée sur la physique a connu une croissance importante au cours des dix dernières années, tant au niveau académique (figure 1.1) que commercial. De telles animations trouvent des applications non seulement dans l'industrie du film et du jeu vidéo, où elles bonifient la qualité du produit, mais aussi dans toutes les applications de la réalité virtuelle, en permettant une expérience plus immersive.

L'animation basée sur la physique se fonde sur le principe que le mouvement d'entités physiques, tel que la déformation d'un tissu, la formation des vagues ou encore le mouvement de boules de billard suite à des collisions, s'explique par une combinaison de principes physiques que l'on peut simuler numériquement. Lorsque les outils sont disponibles, il est généralement plus simple pour l'animateur d'obtenir de tels mouvements par une simulation physique plutôt que par le biais d'outils d'animation traditionnels.

Diverses techniques permettent aujourd'hui de reproduire de façon relativement convaincante le mouvement d'un grand nombre d'entités inertes, allant du mouvement d'un corps rigide simple, jusqu'aux fourrures, en passant par les fluides. Les défis principaux sur ce front de recherche consistent à augmenter la performance et la précision de ces techniques.

La simulation physique d'une entité *active*, tel qu'un être humain ou un robot par exemple, pose un défi supplémentaire puisqu'elle implique la notion de commande d'un système. Une animation de qualité doit donc respecter les lois de la physique, ce que nous qualifions d'un mouvement *plausible*, à l'aide de commandes *réalistes*. Nous définissons des commandes réalistes comme actionnant l'entité active d'une façon conforme aux attentes de l'observateur. En d'autres termes, l'observateur doit pouvoir imaginer que le mouvement observé a été obtenu par capture de mouvement sur l'entité en

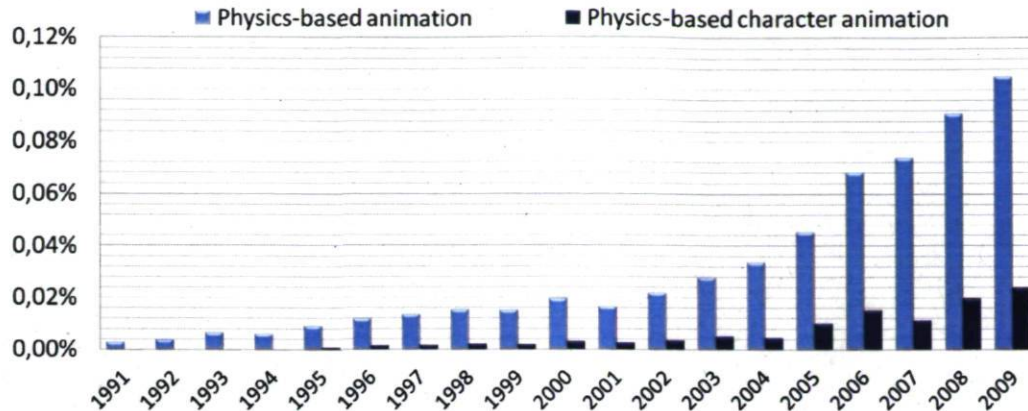


FIGURE 1.1 – Pourcentage annuel d’articles scientifiques ayant les mots-clés *Physics-based animation* et *Physics-based character animation* parmi les publications scientifiques indexées par Google ScholarTM dans la catégorie *ingénierie, informatique et mathématique* entre 1991 et 2009.

question. Si nous cherchons à animer un robot articulé, le réalisme sera généralement facile à atteindre puisque nous avons accès aux algorithmes de commande de celui-ci. Ce n’est toutefois pas le cas pour un mouvement humain ou animal, puisqu’en plus de ne pas avoir d’algorithme de commande absolu en référence, la vision humaine est particulièrement performante pour déceler les défauts des mouvements humains et autres mouvements naturels.

1.1 Objectifs

Ceci nous amène donc au domaine dans lequel s’inscrivent nos recherches, soit la synthèse de mouvements humains *plausibles* et *réalistes*. Plus précisément, nous nous intéressons à produire des mouvements de marche humaine dans différentes conditions, tel que sur des plans inclinés ou encore en suivant une trajectoire courbe. L’approche développée devrait toutefois être applicable à d’autres mouvements d’êtres humains ou d’animaux.

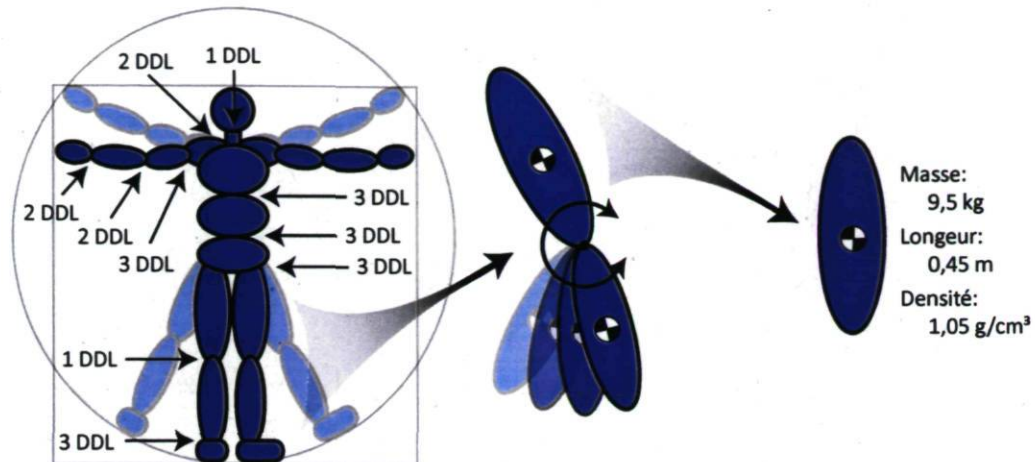


FIGURE 1.2 – De gauche à droite, un système multicorps représentant un personnage, une de ses articulations et un des corps rigides qui le composent. Ces articulations correspondent à celles généralement utilisées en synthèse de mouvement humains par des approches basées sur la physique.

1.1.1 Caractérisation du problème

Modèle biomécanique

Le personnage sur lequel les mouvements sont appliqués est défini par un modèle biomécanique. Nous avons choisi de représenter le corps humain par un système multicorps, soit un ensemble de corps rigides reliés par des articulations. Chaque partie du corps, donc chaque corps rigide, est définie par une longueur (distance entre les articulations), une masse et une densité (uniforme). Le centre de masse et le tenseur d'inertie de chaque corps rigide sont donnés comme ceux d'un ellipsoïde prolate¹ de même longueur, masse et densité. Dans nos figures, le personnage sera généralement illustré à l'aide de ces ellipsoïdes. Les articulations relient deux corps rigides et sont définies par leurs points d'attache sur ces corps rigides ainsi que par leurs degrés de liberté (DDLs), soient les axes autour desquels les articulations peuvent tourner (figure 1.2).

Le modèle biomécanique choisi ne fait pas état des structures internes plus complexes affectant le mouvement, dont les tendons et les ligaments. De plus, les structures représentées sont modélisées sous une forme simplifiée : corps rigides, ellipsoïdes pro-

1. Un ellipsoïde prolate, ou ellipsoïde de rotation, est un ellipsoïde qui peut être obtenu à partir de la rotation d'une ellipse autour de son axe principal. Ses deux axes mineurs sont donc de même grandeur.

lates, etc. Nous avons volontairement choisi un modèle biomécanique simplifié afin de concentrer nos efforts sur le coeur de notre problème sans y ajouter une complexité supplémentaire. En d'autres termes, nous sommes d'avis que la paramétrisation actuelle de notre modèle est nécessaire et suffisante pour bien caractériser le problème. En effet, les mouvements issus de capture de mouvement sont généralement appliqués à des systèmes multicorps et reproduisent des mouvements humains que l'on peut qualifier de réalistes. De plus, ce modèle nous permet également de valider la plausibilité physique d'un mouvement dans un environnement virtuel donné. Il est donc possible de produire des mouvements humains plausibles et réalistes à partir de tels modèles. Nous sommes toutefois d'avis qu'une modélisation plus détaillée pourrait aider à raffiner les mouvements obtenus [58, 46].

De plus, le modèle biomécanique choisi ne contient aucune information quant aux angles limites des articulations et efforts articulaires maximaux que le personnage peut produire. Cela peut être problématique pour une approche d'animation basée sur la physique, puisqu'elle pourrait admettre comme physiquement valide un mouvement qui ne respecte pas les contraintes du corps humain. Toutefois, comme nous le verrons au chapitre 5, la solution que nous proposons ne tend pas vers des mouvements exigeant des forces excessives ou des angles impossibles. Nous avons donc opté pour la simplicité puisque le problème nous le permettait. Notons toutefois que la situation serait probablement différente si nous faisons la synthèse de mouvements plus énergétiques, tels que des mouvements acrobatiques [20].

Environnement virtuel

Toujours dans un esprit de simplicité, les mouvements seront effectués sur un plan infini libre de tout obstacle. Tel que mentionné plus tôt, afin d'être considérés comme plausibles, les mouvements générés devront respecter certaines contraintes posées par les lois de la physique. Nous ne considérons que les contraintes les plus significatives pour le problème traité :

- Les mouvements doivent respecter les équations de mouvement du personnage (i.e. conservation de la quantité de mouvement) ;
- Le personnage est assujéti à une accélération gravitationnelle ;
- Aucune partie du personnage ne peut passer au travers du sol ;
- Le coefficient de frottement entre le sol et le personnage régit les forces de contact possibles ;
- Le sol est infiniment rigide.

Les différents paramètres de l'environnement, dont l'inclinaison du sol, son coefficient de frottement ou l'accélération gravitationnelle, sont appelés à être modifiés afin de faire

la synthèse de différents mouvements. Toutefois, nous ne considérons pas la résistance de l'air lors de la synthèse de mouvement. Nous ne sommes donc pas en mesure de simuler les effets des vents ou autres effets atmosphériques.

Mouvement

Tel que mentionné plus tôt, nous cherchons à faire la synthèse de mouvements de marche humaine sous différents environnements. Pour favoriser un temps de traitement raisonnable, ces mouvements seront généralement échantillonnés à une fréquence de 15 ou 30 trames par seconde. Bien que la capture de mouvement puisse généralement échantillonner 120 trames par seconde ou plus, une fréquence de 30 trames par seconde suffit généralement à reproduire des mouvements humains d'apparence fluide. Les mouvements reproduits à une fréquence de 15 trames par seconde sont légèrement saccadés, mais s'avèrent très utiles pour tester plus rapidement nos solutions.

Dans cet ouvrage, nous ferons la distinction suivante entre les termes *configuration* et *configuration dynamique* :

Configuration Nous utilisons le terme *configuration* pour désigner la cinématique du personnage à un temps t . La configuration du personnage est donnée par les valeurs prises par chacune des coordonnées articulaires du modèle biomécanique au temps t . La configuration au temps t d'un modèle biomécanique à n coordonnées articulaires sera donnée par un vecteur \vec{q}_t à n composantes. Selon notre usage, les premières et secondes dérivées de ces valeurs ne font pas partie de la configuration, mais sont obtenues par différences finies sur des configurations consécutives ;

Configuration dynamique Nous utilisons le terme *configuration dynamique* pour désigner la dynamique du personnage. La configuration dynamique englobe la *configuration* du personnage ainsi que les forces, internes et externes, appliquées sur le personnage au temps t . La configuration dynamique au temps t d'un modèle biomécanique à n coordonnées articulaires et à m corps rigides sera donnée par un vecteur \vec{q}_t à n dimensions représentant la configuration, un vecteur $\vec{\tau}_t$ à n dimensions exprimant les efforts articulaires ainsi que m vecteurs de forces $\hat{f}_{ext,i,t}$ en coordonnées spatiales exprimant les forces externes agissant sur chacun des m corps rigides du personnage. En pratique, nous considérons que seuls les deux pieds peuvent subir des forces externes, par leur contact avec le sol, et nous ne considérons donc que 2 des m vecteurs de forces externes. Notons qu'au lieu de considérer la gravité comme une force externe, nous soustrayons l'accélération gravitationnelle de l'accélération du personnage, ce qui est dynamiquement équivalent.

Nous utiliserons le terme *mouvement* pour désigner une séquence de configurations, tout en précisant *mouvement dynamique* lorsque nous souhaitons plutôt faire référence à une séquence de configurations dynamiques.

Spécifions que nous cherchons à faire la synthèse de mouvements cycliques. Aussi, les paramètres de l'environnement de synthèse demeurent constants tout au long du mouvement. Nous ne visons donc pas, par exemple, à faire la synthèse d'un mouvement de transition qui commence sur un plancher horizontal et qui se termine sur un plan incliné. Finalement, nous ne cherchons pas à produire une approche de synthèse de mouvement en temps réel. Les mouvements dynamiques obtenus pourraient éventuellement être appliqués en temps réel par un régulateur de type proportionnel-intégral-dérivé (PID) ou autre, mais cela ne fait pas partie des objectifs de cet ouvrage.

1.2 Contexte

Afin de situer la solution que nous allons proposer dans son contexte, nous présentons un bref historique des principales techniques d'animation de personnages. Cet historique est suivi d'une revue des recherches récentes les plus pertinentes à notre problème.

1.2.1 Historique

Des débuts de l'animation par ordinateur en 1964 [39] jusqu'à aujourd'hui, la pratique d'animation la plus connue et probablement la plus utilisée est l'animation par trames clés (*key frames* en anglais) en cinématique directe. Cette approche consiste à diriger le mouvement du personnage en fixant la valeur de ses paramètres à certaines trames, les trames clés, et à utiliser une méthode d'interpolation pour reconstruire la position du personnage à chaque trame du mouvement. Bien que simple d'implémentation et très souple, la qualité de l'animation produite dépend entièrement de la capacité de l'animateur à reproduire des mouvements plausibles et réalistes, ainsi que de la qualité de la méthode d'interpolation.

Bien que la capture de mouvement comme technique d'animation ait été popularisée dans les années 1990, elle peut être répertoriée dans la littérature scientifique dès le début des années 1980 [9]. La capture de mouvement permet de fixer les différents paramètres d'un personnage à partir des mouvements d'un acteur (figure 1.3). Dans le cadre de nos travaux, nous voyons la capture de mouvement comme une technique

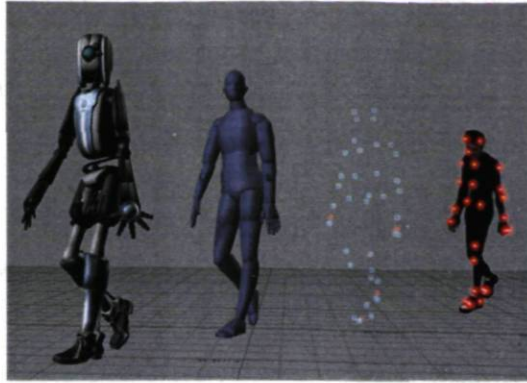


FIGURE 1.3 – Illustration du processus d'animation par capture de mouvement, de la séance de capture jusqu'à la transposition du mouvement sur le personnage virtuel.

permettant d'estimer les angles des diverses articulations d'un personnage modélisé par un système multicorps, soit un assemblage de corps rigides reliés par des articulations. Notons toutefois que la capture de mouvement peut aussi être utilisée pour estimer d'autres types de paramètres, tels que la déformation de la surface du visage d'un acteur par l'action de ses muscles [86, 30, 12] ou encore la déformation de la surface du corps en entier, si le corps humain est considéré comme un corps déformable [61, 16]. La capture de mouvement est essentiellement une approche par trames clés en cinématique directe, toutefois ces trames clés estiment avec une bonne précision les configurations d'un mouvement réel au lieu d'être fixées manuellement. Il est aussi possible d'éliminer la phase d'interpolation en faisant de chaque trame du mouvement une trame clé. L'avantage évident de ce type d'approche est le réalisme des animations obtenues, dans la mesure où le mouvement est reproduit dans un contexte similaire à celui de la capture, donc dans un environnement virtuel similaire à l'environnement de capture et sur un personnage de proportions similaires à l'acteur. En contrepartie, le mouvement appliqué à un contexte différent du contexte de capture perd rapidement son réalisme. Un mouvement capturé a donc une réutilisabilité relativement faible. Des outils d'édition de capture de mouvement ont été créés afin de corriger partiellement le problème, mais, comme pour les animations par trames clés, la qualité du résultat dépend ultimement du talent de l'animateur et de l'ampleur des retouches apportées.

Éventuellement, ces outils d'édition ont été bonifiés par les premiers outils d'animation utilisant la cinématique inverse. La cinématique inverse permet à l'animateur de spécifier une ou plusieurs contraintes sur les configurations du personnage, en fixant sa main à une poignée de porte par exemple, tout en laissant au logiciel d'animation le fardeau de produire des configurations répondant à ces contraintes. Lorsque le problème est sous contraint, ce qui est généralement le cas, une infinité de configurations peuvent répondre aux contraintes, le réalisme de l'animation dépend donc fortement du

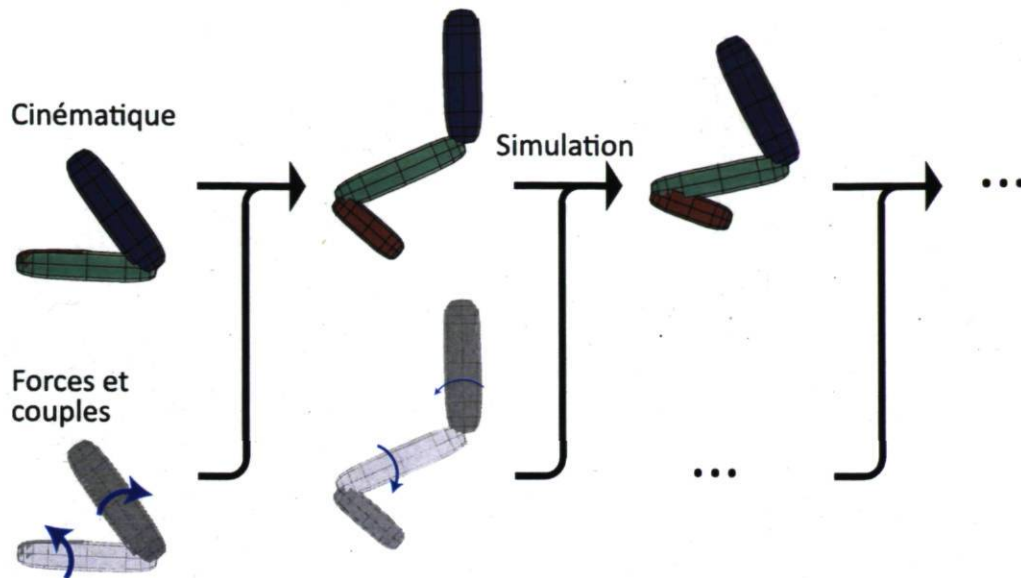


FIGURE 1.4 – Illustration du processus d’animation par simulation physique sur un personnage simplifié. À chaque instant, la configuration est déterminée par une simulation appliquant à la configuration précédente les forces données.

choix de la solution. Les approches de cinématique inverse vont généralement intégrer une méthode favorisant des configurations plus proches de configurations connues par rapport à des configurations plus extravagantes [27, 10, 51]. Tel que mentionné plus tôt, les outils de cinématique inverse sont particulièrement intéressants pour modifier des animations issues de capture de mouvement, pour corriger des glissements de pieds par exemple [36].

Les premières méthodes d’animation basées sur la physique sont elles aussi apparues dès les années 1980. Un attrait majeur de ces méthodes est que les animations qu’elles produisent respectent d’emblée les lois de la physique. La première vague de travaux visant l’animation de personnages articulés était centrée sur la dynamique directe [5, 25, 54, 85]. L’animation y est déterminée par le résultat d’une simulation physique où l’on fixe l’état initial du personnage ainsi que les forces appliquées tout au long de l’animation (figure 1.4). Une faille évidente de ce type d’approches est la complexité du contrôle de l’animation, même pour une entité inerte. À titre d’exemple, si nous souhaitons que deux balles lancées dans les airs entrent en contact à un temps donné, il nous faut ajuster les conditions initiales avec beaucoup d’attention.

Les méthodes de *forces de contraintes* [7, 37], qui permettent de poser des contraintes cinématiques sur les entités à animer, ont été développées afin de faciliter le contrôle des animations basées sur la physique. Le principe de ces approches est d’introduire

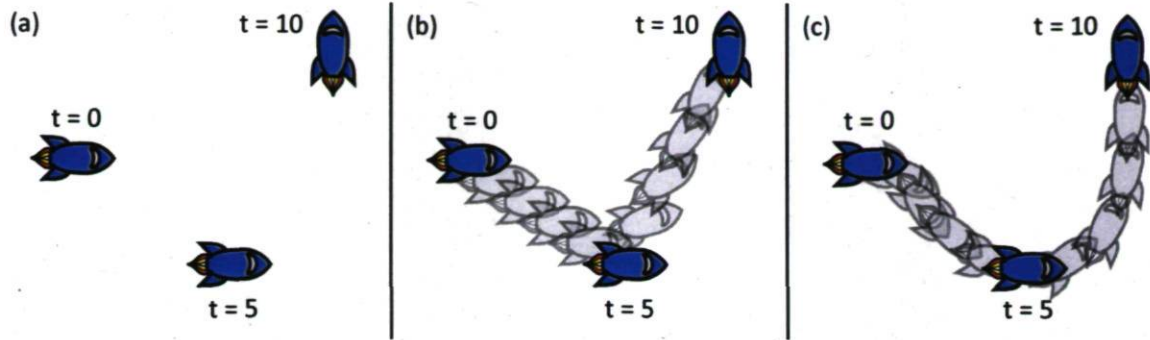


FIGURE 1.5 – Illustration du paradigme des contraintes spatiotemporelles. (a) Les configurations à atteindre aux temps donnés sont posées comme contraintes. (b) Une solution intermédiaire hypothétique ne respectant pas les équations de mouvement de la fusée. (c) Une solution finale hypothétique atteinte optimisant tous les paramètres, soient les positions, orientations et poussées de la fusée à chaque instant. Cette solution respecte toutes les contraintes et minimise une fonction de coût, telle que la consommation de carburant.

des forces produisant les mouvements répondant aux contraintes cinématiques. Dans un premier temps, la résolution d'un problème de dynamique inverse permet de déduire les forces de contraintes et, dans un second temps, ces forces sont appliquées en simulation. Ces approches combinent donc la dynamique inverse, pour calculer les forces de contraintes, et la dynamique directe, pour produire l'animation finale.

En 1988, Witkin et Kass [88] ont proposé les *contraintes spatiotemporelles*, un paradigme plus élégant et plus souple pour poser des contraintes sur des animations basées sur la physique (figure 1.5). Au lieu de considérer l'animation comme un problème de dynamique directe ou de dynamique inverse, ce paradigme traite l'animation comme un problème d'optimisation sujet à diverses contraintes, dont le respect des équations de mouvement. Le terme "spatiotemporelles" réfère au fait que l'animation est considérée dans sa totalité, comme un ensemble de configurations dynamiques à différents temps, contrairement aux approches de dynamique directe, où l'animation est produite chronologiquement du temps 0 au temps T . Dans ce paradigme, les paramètres du mouvement dynamique à tous les instants forment le vecteur à optimiser en respectant les contraintes.

Sans rendre le problème trivial, voir l'animation comme un problème d'optimisation permet une grande souplesse. Les contraintes sur le mouvement, tant cinématiques que dynamiques, prennent la forme de fonctions de contraintes. À titre d'exemple, les contraintes d'équation de mouvement assurent que les paramètres de forces correspondent aux accélérations dans le mouvement. Le choix de la fonction de coût peut être plus ambiguë, mais il permet tout de même de déterminer comment l'animation

doit procéder, en minimisant les accélérations des articulations par exemple. L'envers de la médaille est la complexité du problème d'optimisation à résoudre pour générer l'animation. Dans le cas général, il s'agit d'optimiser une fonction quadratique ou non linéaire sous contraintes d'égalités et d'inégalités nonlinéaires et cela dans un espace à très grande dimensionnalité. Afin d'être applicables en pratique, ces approches nécessitent donc un processeur performant ainsi qu'un algorithme d'optimisation non linéaire efficace. Néanmoins, un grand nombre d'approches d'animation récentes sont fondées sur le paradigme des *contraintes spatiotemporelles*.

1.2.2 État de l'art

Au cours des vingt dernières années, plusieurs chercheurs se sont intéressés à produire plus facilement des animations alliant réalisme et souplesse. Nous présentons ici les principales recherches dans le domaine en les regroupant sous différents paradigmes. Notons que cette classification n'est pas absolue, puisque certaines approches empruntent leurs idées à plus d'un paradigme, mais elle aide néanmoins à obtenir une vue d'ensemble de ce domaine de recherche.

Animation basée sur les données

Les méthodes d'animation en cinématique, par opposition aux approches basées sur la physique, ont généralement l'avantage d'être rapides d'exécution. Si ces approches utilisent des données issues de capture de mouvement, de façon intégrale ou avec de légères modifications, les animations produites ont le potentiel d'être à la fois plausibles et réalistes. Tel que mentionné plus tôt, la condition à respecter pour obtenir une animation plausible et réaliste à partir de capture de mouvement est la correspondance entre l'environnement de capture et l'environnement de synthèse. À partir d'une banque de mouvements arbitrairement grande, il serait possible d'extraire le mouvement souhaité pour l'environnement de synthèse désiré.

Lamouret et van de Panne [43] ont appliqué cette idée pour faire suivre à la lampe *Luxo* une trajectoire préétablie sur un parcours comportant des marches et des plans inclinés. Pour chaque saut, une banque de mouvements est consultée afin d'y trouver le saut répondant le mieux aux caractéristiques du saut nécessaire pour faire avancer le personnage à la cadence voulue. Une fois ce saut trouvé, il est modifié légèrement afin de répondre aux contraintes cinématiques du mouvement. Puisque la capture de mouvement n'était pas applicable pour un personnage fictif de la sorte, la banque de

mouvements de *Luxo* fût peuplée par des sauts de longueurs et hauteurs variables synthétisés en simulation physique. Le mouvement final est obtenu par la concaténation des différents sauts retenus.

La qualité de l'animation d'une telle approche dépend de deux paramètres : l'ampleur des modifications nécessaires pour adapter chaque saut et la qualité de la transition entre deux sauts. Ces deux paramètres peuvent théoriquement être améliorés par une banque de mouvements plus densément peuplée. Toutefois, générer des transitions de qualité à partir d'une banque de mouvements est un problème qui peut être mieux abordé par une autre approche, soit les *graphes de mouvements*.

Par un concours de circonstances exceptionnel, le concept à la base des graphes de mouvements a été avancé simultanément à la conférence SIGGRAPH 2002 par Kovar et al. [42], Lee et al. [45], Arikan et Forsyth [4], Li et al. [48] et Pullen et Bregler [66]. Plusieurs recherches ont ensuite construit sur cette idée [76, 31, 68, 72, 8, 95, 94]. À sa plus simple expression, le graphe de mouvements est une banque de mouvements augmentée de liens entre des paires de mouvements représentant des transitions (figure 1.6). La construction du graphe consiste à déterminer quels mouvements peuvent être liés par une transition et à trouver les trames de ces mouvements produisant des transitions fluides. Il est également possible de générer quelques trames de transition entre les deux mouvements, pour assurer une transition encore plus fluide. Pour faire la synthèse d'un mouvement, on cherche un parcours du graphe minimisant une fonction de coût, telle que le respect d'une trajectoire de marche préétablie. Lorsque le graphe est bien peuplé et suffisamment connecté, il est possible d'en obtenir des animations très réalistes et qu'un observateur humain pourrait juger plausibles. Par contre, les mouvements pouvant être produits sont limités par le contenu de la banque de mouvements initiale. À titre d'exemple, un graphe de mouvements contenant diverses animations de marche en ligne droite pourrait difficilement produire un mouvement de marche sur une trajectoire courbe. De plus, pour produire de bons résultats, les graphes de mouvements doivent être peuplés d'un grand nombre de mouvements. Notons que Sung et al. [81] ont également utilisé les graphes de mouvements pour animer des foules de façon réaliste.

Afin d'augmenter la souplesse des graphes de mouvements, Ren et al. [69] ont récemment proposé d'augmenter le nombre de connexions dans un graphe de mouvement en générant des transitions par la méthode des contraintes spatiotemporelles. Les transitions créées de la sorte permettent de lier de façon convaincante des mouvements qui auraient été disjoints dans un graphe de mouvements simple. Cet ajout permet de faire la synthèse de mouvements suivant plus fidèlement une trajectoire désirée ou une séquence d'actions souhaitée.

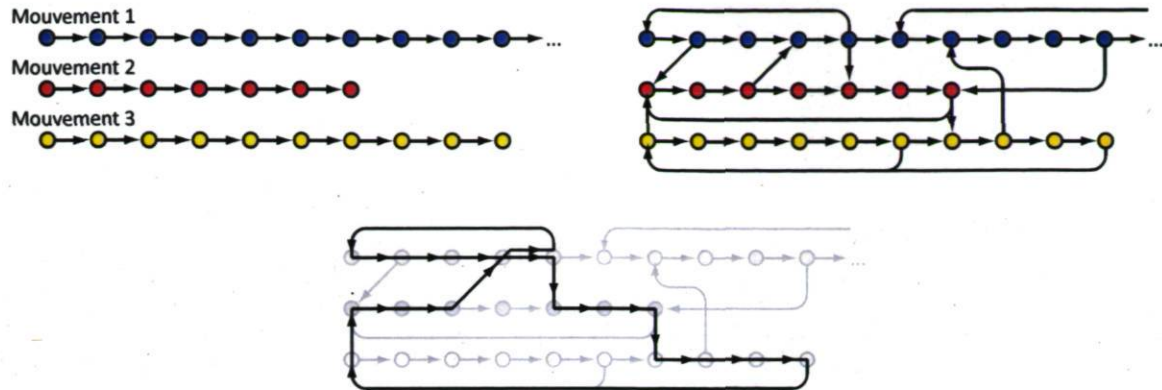


FIGURE 1.6 – Illustration du processus de construction d'un graphe de mouvement et de son utilisation. Les configurations des différents mouvements de la banque de mouvements (en haut à gauche) sont reliées pour former un graphe de mouvements (en haut à droite). On peut ensuite parcourir ce graphe pour faire la synthèse de nouveaux mouvements (en bas).

Toujours dans les techniques d'animation basées sur les données, Kovar et Gleicher [41] ont proposé une méthode de paramétrisation des mouvements contenus dans une banque de mouvements. Selon cette approche, les mouvements similaires sont regroupés et traités comme des échantillons d'un espace paramétrique de ce type de mouvement. Des mouvements intermédiaires peuvent ensuite être interpolés dans cet espace.

Hsu et al. [34] utilisent une approche similaire aux graphes de mouvements, mais mettent plus d'emphasis sur les techniques de contrôle du mouvement à produire. Lee et Elgammal [46] proposent une autre approche basée sur les données. Ils utilisent des exemples de mouvements pour estimer un *manifold*² à faible dimensionnalité, sur lequel se trouvent les configurations des mouvements d'intérêt. De nouveaux mouvements peuvent ensuite être interpolés à partir du *manifold*.

En général, nous observons que les approches basées sur les données ont le potentiel de produire de très bons résultats lorsque le mouvement à synthétiser peut être reproduit par une suite de mouvements contenus dans une banque de mouvements [42, 45, 4, 48, 66, 69] ou par une interpolation entre ces mouvements [41, 46]. Puisque notre problématique porte plutôt sur l'extrapolation de mouvements, ces méthodes sont cependant peu appropriées.

2. Un *manifold* peut être vu comme un objet géométrique à n dimensions contenue dans un espace à $(n + m)$ dimensions.

Contrôleurs en cinématique

Sun et Metaxas [79] sont parmi les seuls à avoir proposé une forme de contrôleur n'étant pas basé sur la dynamique. À l'instar des méthodes basées sur les données, ce contrôleur utilise un mouvement issu de capture de mouvement duquel il extrait des signaux d'angles caractéristiques, dont les angles décrits dans le plan sagittal par les hanches, les genoux et les chevilles. Le contrôleur fait ensuite la synthèse d'un mouvement respectant ces angles caractéristiques tout en respectant certaines contraintes cinématiques, telles que le contact entre le sol et les pieds. Cette approche profite d'un temps d'exécution très rapide et d'une grande souplesse, puisqu'elle permet de générer des démarches dans une multitude d'environnements différents. Toutefois, les modifications appliquées aux données de mouvement initiales peuvent être très grossières et ne sont pas contraintes par la physique. Les mouvements obtenus n'ont donc aucune garantie d'être réalistes ou même plausibles.

Contrôleurs sous simulation physique

Les approches d'animation en dynamique directe ont l'avantage de produire des animations plausibles d'emblée, de permettre d'adapter des mouvements à certaines variations de l'environnement de synthèse et possiblement de répondre à des forces externes inattendues. Le personnage y est généralement activé par un contrôleur de type *proportionnel-dérivé* cherchant à atteindre des configurations dictées par un mouvement idéal.

Plusieurs chercheurs ont tenté de produire des contrôleurs par design ou par apprentissage automatique. Nous les qualifions de contrôleurs synthétiques. Dans ces approches, le mouvement à suivre peut être donné par des courbes paramétriques [28, 67, 90], des machines à états finis [32, 18, 17, 19, 75] ou une combinaison des deux [44, 83]. Bien que les animations résultantes soient plausibles, elles ne sont généralement pas d'apparence naturelle.

Pour ajouter du réalisme, il est possible d'utiliser des données issues de capture de mouvement et de créer des contrôleurs basés sur les données. C'est ce qu'ont proposé Zordan et Hodgins [96] pour animer le haut du corps d'un personnage à partir de capture de mouvement ; les mouvements de l'acteur étant imités par le personnage tout en respectant les contraintes physiques de ce dernier. Yin et al. [92], da Silva et al. [14] et Muico et al. [57] ont appliqué ce principe avec succès à des mouvements de locomotion pour un personnage entier. Abe et Popović [3] ont également utilisé ce principe pour

faire interagir un personnage avec son environnement.

Récemment, da Silva et al. [15] ont proposé une méthode d'interpolation de contrôleur. Leur approche permet de combiner plusieurs contrôleurs selon une certaine pondération afin d'atteindre un objectif donné.

Bien que cela dépasse les objectifs de nos recherches, mentionnons tout de même que certaines recherches ont été menées afin de rendre ces contrôleurs robustes à des forces externes inattendues telles que l'impact d'un ballon ou d'un coup de pied [97, 40, 1, 53]. Cette idée a ensuite été appliquée aux méthodes d'animation basées sur les données [74, 98, 93]. Prenant la forme de commandes hybrides, l'animation est purement cinématique jusqu'au moment d'application de la force externe, suite à quoi des commandes en dynamique directe sont appliquées pour répondre à l'impact. Ce contrôleur tente ensuite de faire une transition vers un mouvement de la banque de mouvements pour finalement lui céder le contrôle de l'animation.

Finalement, nous croyons qu'il est important de mentionner les travaux de Grzeszczuk et al. [29]. Ils ont développé un environnement de simulation baptisé NeuroAnimator qui est en fait un réseau de neurones entraîné à imiter un environnement de simulation physique. Le NeuroAnimator est moins général qu'un simulateur physique *traditionnel* puisqu'il ne peut simuler adéquatement la dynamique d'entités sur lesquelles il n'a pas été entraîné, mais est significativement plus performant dans ces simulations. Une approche inspirée du NeuroAnimator est donc tout indiquée dans certaines situations où la rapidité de la simulation prime la précision du résultat.

Contraintes spatiotemporelles

L'approche des contraintes spatiotemporelles peut s'appliquer à plusieurs problématiques propres à l'animation. Witkin et Kass [88] l'ont d'abord présentée comme une technique permettant de générer une animation originale à partir de très peu d'information, soient quelques positions à respecter et une ligne directrice selon laquelle le mouvement sera jugé optimal. Rose et al. [71], et éventuellement Ren et al. [69], ont quant à eux utilisé cette approche pour générer des transitions plausibles et réalistes entre deux mouvements d'une banque de mouvements. Popović et Witkin [65] et Popović [64] ont construit des outils d'édition de mouvements capturés sur la base des contraintes spatiotemporelles. L'attrait de ces approches est qu'elles conservent la plausibilité du mouvement malgré les modifications qui y sont apportées. Popović et al. [63] ont suggéré l'utilisation des contraintes spatiotemporelles dans un système d'animation de corps rigides s'exécutant en temps interactif (plus de 15 trames par seconde).

Plus récemment, Liu et al. [50] ont utilisé les contraintes spatiotemporelles pour résoudre des animations impliquant plusieurs personnages, notamment en introduisant des contraintes entre les différents personnages.

Quelques années après l'introduction des contraintes spatiotemporelles, Ngo et Marks [59] ont noté la complexité du problème d'optimisation sous-jacent à cette approche et l'importance d'avoir une animation initiale près de l'optimum afin d'assurer la convergence. Un effet de bord de cette situation est que l'optimisation du mouvement est relativement locale et ne favorise pas la découverte de mouvements fondamentalement originaux. Ils ont donc proposé l'utilisation d'algorithmes génétiques comme solution d'optimisation globale afin de découvrir une variété de mouvements pour des personnages ayant de 3 à 5 coordonnées articulaires dans un environnement à 2 dimensions. En dépit de la nature aléatoire des algorithmes génétiques, la simplicité des personnages et de l'environnement de synthèse leur a permis d'en obtenir plusieurs mouvements intéressants.

Tout récemment, Wampler et Popović [84] ont repris l'idée d'une optimisation globale avec succès pour animer des créatures fictives ayant de 10 à 32 coordonnées articulaires dans un environnement à 3 dimensions. Notons que leur approche vient généraliser les contraintes spatiotemporelles en ce sens qu'ils optimisent non seulement le mouvement, mais également les paramètres de la créature, dont la longueur de ses membres. Un autre apport important de leurs travaux est qu'ils présentent les contraintes spatiotemporelles comme un problème de *programmation non linéaire mixte*, permettant d'optimiser tant sur des valeurs continues que discrètes. Une variable discrète, telle que les temps de contacts³, peut donc faire partie des paramètres du mouvement à optimiser. Le traitement des variables discrètes ajoute toutefois une lourdeur au système, ce qui se traduit dans leurs expériences par un temps de calcul de l'ordre de 50 fois plus long par rapport à un problème similaire où les paramètres discrets sont fixés *a priori*.

Tel que mentionné plus tôt, l'approche des contraintes spatiotemporelles est basée sur un problème d'optimisation très complexe. Puisque l'animation est optimisée simultanément sur toute sa durée et sur l'ensemble des coordonnées articulaires du personnage, cette complexité augmente rapidement avec la durée de l'animation, la fréquence d'échantillonnage et le nombre de coordonnées articulaires du personnage. Afin d'alléger le problème, Cohen [13] a suggéré l'utilisation de *fenêtres spatiotemporelles*. Les

3. L'information sur les temps de contacts est nécessaire à toute animation par contraintes spatiotemporelles où le personnage entre en contact avec le sol. Ces données binaires indiquent, pour un temps t , si le corps rigide i est en contact avec le sol. Cette information sert généralement à déterminer s'il doit y avoir une force de réaction au sol agissant sur le corps rigide i au temps t . Elle est habituellement fournie comme une constante, posant une contrainte supplémentaire sur le mouvement à optimiser.

fenêtres délimitent un intervalle temporel et, s'il y a plus d'un personnage à animer, un sous-ensemble des personnages. L'optimisation se fait sur un ensemble de fenêtres qui se recouvrent partiellement et qui couvrent la totalité des paramètres de l'animation. Liu et al. [52] ont plutôt abordé le problème par une approche basée sur les ondelettes. Les valeurs d'une coordonnée articulaire à tous les temps sont traitées comme des échantillons d'un signal continu. En représentant ce signal par des ondelettes, il est possible de le définir par un moins grand nombre de paramètres si les hautes fréquences doivent être retranchées du signal ou limitées à certaines sections du signal. De plus, les résultats présentés par Liu et al. [52] semblent indiquer que la représentation des signaux par ondelettes mène à un problème d'optimisation mieux conditionné, donc plus rapide à résoudre, même dans le cas où le nombre de paramètres n'a pas été diminué. Popović et Witkin [65, 64] proposent plutôt d'optimiser le mouvement pour une version simplifiée du personnage, ne conservant que les coordonnées articulaires les plus significatifs au mouvement désiré, pour ensuite reconstruire le mouvement sur le personnage complet. Bien que l'animation optimisée sur le personnage simplifié respecte la physique, ce n'est généralement pas le cas de l'animation finale sur le personnage complet. Safonova et al. [73] proposent une technique similaire, mais utilisent une approche d'analyse en composantes principales (ACP) afin d'effectuer l'optimisation dans un espace à dimensionnalité réduite.

L'approche proposée par Fang et Pollard [20, 21] ne tient compte que de la quantité de mouvement et de moment angulaire du personnage en entier. Elles omettent les contraintes liées aux efforts articulaires, ce qui simplifie significativement la complexité du problème de contraintes spatiotemporelles. En d'autres termes, les mouvements obtenus par ces approches sont physiquement plausibles, mais permettent au personnage de produire des forces internes d'amplitude potentiellement infinie. Dans les cas où l'on ne cherche pas à minimiser ou à contraindre l'effort interne du personnage, cela peut s'avérer un compromis très intéressant.

Liu et al. [49] traitent le problème inverse aux contraintes spatiotemporelles. En partant d'une animation produite par capture de mouvement, ils cherchent les paramètres du modèle pour lesquels le mouvement observé optimise une fonction de coût donnée. Les paramètres du modèle décrivent certaines caractéristiques des mouvements du personnage, dont l'utilisation préférentielle d'un muscle par rapport à un autre par exemple. Il est ensuite possible de faire la synthèse de mouvements originaux respectant ce style de mouvement en utilisant les paramètres trouvés dans une approche de contraintes spatiotemporelles *classique*.

Autres méthodes d'animation par optimisation

Wu et Popović [11] ont utilisé une autre approche d'optimisation pour produire des animations plausibles et réalistes de différents oiseaux. Ils optimisent les paramètres de chaque battement d'ailes par recuit simulé. Leur fonction de coût comporte un terme pénalisant les écarts par rapport à la trajectoire désirée, ainsi que plusieurs termes favorisant la stabilité de l'oiseau, sa propension à ne pas envoyer ses ailes vers l'arrière et autres caractéristiques d'un vol d'apparence naturel.

Bien que basées sur l'optimisation numérique, l'approche proposée par Wu et Popović a quelques différences importantes avec les approches de contraintes spatiotemporelles. Premièrement, l'optimisation sert à déterminer les commandes à appliquer pour obtenir un mouvement adéquat, alors qu'une approche de contraintes spatiotemporelles optimise directement le mouvement et les forces en cause. Deuxièmement, ils traitent un problème d'optimisation non contraint. Cette absence de contraintes provient de deux facteurs :

- L'animation finale est obtenue par simulation physique. Il n'y a donc pas lieu de poser des contraintes sur le respect des équations de mouvement, sur les forces de contact, sur le frottement, etc ;
- La trajectoire à suivre est traitée comme un tuteur, une ligne à suivre, et non comme des positions exactes à respecter. Le terme portant sur le respect de la trajectoire n'est donc pas une contrainte, il fait partie de la fonction de coût.

De plus, cette approche se distingue des autres approches basées sur la dynamique directe en ce sens que les commandes sont optimisées sur mesure pour la trajectoire souhaitée au lieu d'apprendre des commandes plus générales et de chercher à en appliquer la combinaison idéale.

Extrapolation de mouvement

Nous qualifions d'approche d'extrapolation de mouvement les méthodes d'animation appliquant des modifications à une animation issue de capture de mouvement pour produire un nouveau mouvement. Cela regroupe des approches de cinématique, de dynamique directe et de contraintes spatiotemporelles, dont certaines approches que nous avons déjà mentionnées [92, 14, 26, 64], tout en se distinguant des approches produisant de nouveaux mouvements par interpolation d'autres mouvements [41, 46]. Le caractère commun de ces approches est qu'elles visent toutes à faire la synthèse de nouveaux mouvements tout en conservant les caractéristiques propres au mouvement initial. Dans cette section, nous présentons les recherches en extrapolation de mouvement les plus

pertinentes par rapport à nos travaux.

Les approches proposées par McCann et al. [55] et Hsu et al. [33] permettent de moduler le temps d'exécution d'un mouvement connu tout en conservant la plausibilité physique du mouvement. Ces approches de *modulation temporelle* peuvent servir à augmenter ou diminuer la durée du mouvement entier ou seulement de certains segments du mouvement initial. S'il s'agit bien d'une forme d'extrapolation de mouvement, la modulation temporelle d'un mouvement ne répond pas directement aux objectifs spécifiques de nos recherches puisqu'elle ne permet pas de faire la synthèse de configurations significativement originales par rapport au mouvement initial.

Hsu et al. [35] ont obtenu des résultats intéressants avec leur approche de *transposition de style*. Leur approche détermine d'abord une fonction liant deux styles de marche distincts, une marche normale et une marche de côté par exemple. Cette fonction peut ensuite être utilisée pour transformer d'autres instances de marche en lui appliquant l'autre style. Par exemple, en appliquant cette fonction à une marche normale le long d'une trajectoire courbe, on obtient une marche de côté qui suit cette même trajectoire. Cette approche est toutefois limitée à modifier le style avec lequel le mouvement se fait et ne permet pas d'extrapoler une marche suivant une nouvelle trajectoire par exemple.

Witkin et Popović [89] ont proposé le *motion warping*. Cette approche d'animation cinématique module les courbes décrivant les valeurs des coordonnées articulaires en fonction du temps. Des fonctions lisses sont appliquées aux courbes initiales afin de les moduler. Cette modulation se fait de façon à respecter certaines *trames clés* tout en préservant les hautes fréquences des courbes initiales. Cette approche est élégante par sa simplicité et permet de produire des variations convaincantes d'un mouvement donné. Toutefois, comme toute approche cinématique, elle ne garantit pas le respect des lois de la physique, ce qui est une limite importante dans l'extrapolation de certains mouvements. À titre d'exemple, une personne qui tourne rapidement d'un côté devra incliner son corps de ce côté afin de conserver son équilibre. Le *motion warping* ne permettra pas d'obtenir ce type de comportement s'il extrapole le mouvement à partir d'une marche en ligne droite.

Les approches de *filtrage dynamique de mouvement*, telles que proposées par Pollard et Reitsma [62] et par Tak et Ko [82], visent principalement à corriger des animations cinématiques pour les rendre physiquement plausibles, mais elles peuvent également être considérées comme des approches d'extrapolation de mouvement. Tel que démontré par Tak et Ko [82], en appliquant un filtre dynamique à une animation cinématique on peut produire un mouvement original physiquement plausible. Le filtrage dynamique modifie les configurations du mouvement de façon chronologique en une passe,

contrairement aux approches de contraintes spatiotemporelles qui traitent simultanément les paramètres du mouvement dynamique en entier. L'avantage principal de ce type d'approches est que leur temps de calcul n'augmente que linéairement par rapport au nombre de configurations du mouvement. De plus, le filtrage dynamique pourrait aussi être utilisé dans des applications interactives. Par contre, le filtrage en ligne d'un mouvement peut éventuellement mener à des discontinuités dans le mouvement ou à une divergence entre le mouvement souhaité et le mouvement obtenu par filtrage.

Par opposition au filtrage dynamique, les approches de contraintes spatiotemporelles profitent d'une vue globale du mouvement, ce qui élimine les problèmes de discontinuités et de divergence, mais souffrent d'une plus grande complexité algorithmique. C'est pourquoi les approches d'extrapolation de mouvement basées sur ce paradigme tentent d'alléger le problème, soit en utilisant une version simplifiée du personnage [65, 64] ou en ne traitant que des mouvements ballistiques⁴ [2, 78]. Les résultats obtenus par ces approches sont néanmoins des mouvements originaux plausibles et réalistes.

Finalement, Yin et al. [91] proposent une approche inspirée des *méthodes de continuation* pour extrapoler des contrôleurs en simulation physique. L'approche proposée est utilisée pour simplifier un problème d'optimisation qui serait autrement très complexe. Celle-ci consiste essentiellement à débiter à l'optimum d'une version simplifiée du problème d'optimisation initial, à faire varier graduellement le problème simplifié et à trouver l'optimum de ce nouveau problème en se servant de l'optimum du problème précédent comme solution initiale. À sa conclusion, on obtient l'optimum du problème initial. Par cette approche, Yin et al. [91] sont capables d'extrapoler avec succès des contrôleurs de marche sur des plans inclinés, sur un sol glissant et sous d'autres variantes de l'environnement. Leurs contrôleurs n'arrivent toutefois pas à produire des mouvements suffisamment réalistes.

1.3 Solution proposée

Rappelons que l'objectif de nos travaux est de faire la synthèse de mouvements plausibles et réalistes de marche humaine dans différentes conditions. En considérant l'état de l'art dans le domaine, nous sommes d'avis qu'une méthode d'extrapolation de mouvement par contraintes spatiotemporelles serait à privilégier.

Notre raisonnement est le suivant. Il nous semble évident que pour produire un

4. L'absence de forces de contact au sol durant une grande partie d'un mouvement balistique diminue significativement la complexité du problème.

mouvement réaliste il est préférable de baser notre approche sur des mouvements issus de capture de mouvement. Pour y ajouter la souplesse désirée, soit d'effectuer un même mouvement sous différentes variantes d'un environnement, une méthode d'extrapolation de mouvement s'impose. De plus, afin de garantir que cette extrapolation produise des mouvements plausibles, elle doit être basée sur la physique.

Ce processus d'élimination nous présente donc trois options : l'extrapolation de contrôleurs en simulation physique, l'extrapolation de mouvement par filtrage dynamique ou l'extrapolation de mouvement par contraintes spatiotemporelles. Nous éliminons la première option étant donné le faible réalisme des animations produites par les approches basées sur des contrôleurs en simulation physique. Finalement, nous éliminons le filtrage dynamique étant donné les possibilités de divergence ou de discontinuités.

Bien qu'elle réponde à nos critères, l'extrapolation de mouvement par contraintes spatiotemporelles comporte des limites importantes, particulièrement la complexité du problème d'optimisation non linéaire sous-jacent. Cette complexité peut ralentir considérablement le processus d'optimisation ou le confiner à des minima locaux indésirables. D'ailleurs, tel que mentionné dans la section précédente, les approches actuelles s'adressent à des problèmes relativement simples, tels que des mouvements balistiques [2, 78], ou résolvent le mouvement pour une version simplifiée du personnage [65, 64].

Afin de simplifier le problème sans restreindre le type de mouvement ou la complexité du personnage, l'approche que nous proposons vise à fournir au processus d'optimisation une solution initiale aussi proche que possible de l'optimum recherché. À cette fin, nous avons développé une méthode permettant d'estimer très rapidement les forces de réaction au sol en support double et nous avons développé une stratégie inspirée des méthodes de continuation pour les contraintes spatiotemporelles.

L'idée de notre approche est simple mais puissante. Dans un premier temps, une animation issue de capture de mouvement est rendue plausible par contraintes spatiotemporelles. La tâche est plus ou moins difficile selon la qualité de la capture de mouvement et la similarité entre les proportions du personnage par rapport à celles de l'acteur ayant effectué le mouvement, mais elle sera toujours facilitée par notre estimation des forces de contact au sol. À sa conclusion, cette première étape produit un mouvement plausible et réaliste reproduisant assez fidèlement le mouvement capturé. De plus, nous avons maintenant un mouvement dynamique, soit un mouvement pour lequel nous connaissons la cinématique et la cinétique.

Puisque le mouvement est capturé sur un acteur, il est forcément plausible dans le monde réel. Cependant, lorsqu'appliqué sur le personnage virtuel dans l'environnement

de simulation, le mouvement peut demander une certaine adaptation pour être physiquement plausible. Cela est principalement causé par une modélisation simplifiée de la biomécanique du personnage et ainsi que de l'environnement. Le bruit d'acquisition et les problèmes de traitement des données peuvent aussi introduire des défauts qui doivent être corrigés par contraintes spatiotemporelles.

Ce mouvement dynamique fait ensuite office de mouvement de base à partir duquel nous pouvons extrapoler d'autres mouvements. Par une approche similaire aux méthodes de continuation, l'extrapolation de mouvement se fait graduellement, en variant les paramètres de l'environnement par petits incréments. À chaque incrément, on résout un nouveau problème de contraintes spatiotemporelles en utilisant l'optimum de l'incrément précédent comme solution initiale. Puisque le problème varie de façon continue, les solutions de deux incréments consécutifs seront très similaires dans la mesure où les incréments sont suffisamment petits.

À titre d'exemple, supposons que nous souhaitons extrapoler un mouvement de marche sur un plan de 20° à partir d'un mouvement de marche sur un plancher horizontal issu de capture de mouvement. La première étape consiste à rendre ce mouvement plausible pour le personnage dans son environnement virtuel. On pose ensuite le même problème de contraintes spatiotemporelles, mais pour un plan incliné de 1° , en utilisant le mouvement dynamique obtenu à la première étape comme point de départ pour l'optimisation. De la même façon, on pose ensuite un problème similaire pour une pente de 2° , en utilisant comme point de départ l'optimum trouvé pour la pente de 1° . On procède de cette façon jusqu'à ce qu'on résolve le mouvement pour une pente de 20° .

Tel que nous le verrons au chapitre 5, l'implémentation actuelle de cette solution permet la synthèse de mouvements de marche plausibles et réalistes pour un modèle simplifié de personnage humain pour lequel le haut du corps est omis. Le problème de synthèse de mouvement pour un personnage anthropomorphique complet semble toutefois trop complexe pour notre implémentation actuelle et nous n'avons pas réussi à obtenir des résultats significatifs en ce sens. Il peut alors sembler inadéquat de parler d'un mouvement réaliste, mais nous pouvons néanmoins comparer le mouvement des jambes issu de capture de mouvement avec les mouvements synthétisés.

Nous sommes d'avis que la complexité du problème d'optimisation pourrait être allégée davantage en modifiant quelques aspects de l'implémentation, notamment sur la modélisation des articulations. Nous y reviendrons dans la section 1.3.2. Étant donné la taille imposante du code utilisé, il est également possible que nous soyons victimes d'une erreur de programmation qui ne se produit que dans les cas les plus complexes et qui est donc difficile à identifier et à corriger.

Malgré cette limite de l'implémentation, nos résultats démontrent tout de même le potentiel de la solution proposée. En effet, que l'on traite un personnage anthropomorphe ou un personnage dont le haut du corps est omis, tous les aspects du problème demeurent les mêmes. Il s'agit de faire la synthèse d'un mouvement de marche bipède sous les mêmes contraintes. C'est d'ailleurs pourquoi nous sommes d'avis qu'une implémentation révisée de la solution proposée pourrait produire des résultats plus intéressants que l'implémentation actuelle. De plus, malgré que la dynamique du personnage simplifiée s'avère fort différente de celle d'un personnage anthropomorphe, notre approche permet tout de même de produire des mouvements réalistes puisque le processus d'optimisation fait tendre le mouvement de synthèse vers le mouvement capturé.

Soulignons que notre technique d'estimation des forces de contact n'est pas affectée par cette limite. Comme nous le verrons au chapitre 4, le problème est formulé en termes de sous-problèmes de programmation quadratiques dont la complexité est indépendante du nombre de coordonnées articulaires du personnage ou de son graphe de connexions. La complexité du personnage influence uniquement le temps requis pour le calcul des matrices et vecteurs utilisés pour définir ces sous-problèmes.

1.3.1 Contributions

Nos travaux apportent deux contributions significatives à l'état de l'art actuel : une méthode d'estimation rapide des forces de contact en support double, soit lorsque les deux pieds du personnage sont au sol, et une stratégie d'extrapolation progressive de mouvement par contraintes spatiotemporelles.

Notre technique d'estimation des forces de contact au sol permet de trouver une solution rapide au problème mathématiquement sous-contraint de la répartition des forces de contact en support double, soit quand le personnage a les deux pieds au sol en même temps. Cette technique sera détaillée au chapitre 4. Soulignons que cette technique peut être utilisée dans tout domaine où le problème du support double se pose, tant en animation basée sur la physique qu'en robotique ou en kinésiologie. Sa portée n'est donc pas limitée à notre approche d'extrapolation de mouvements. À notre connaissance, aucune approche similaire n'a été proposée dans la littérature scientifique.

Notre stratégie d'extrapolation de mouvement permet de résoudre un problème d'optimisation complexe en résolvant une séquence de problèmes d'optimisation beaucoup plus simples. Cela nous permet d'extrapoler avec succès des mouvements relativement complexes sur des personnages à plusieurs coordonnées articulaires, un problème que les approches de contraintes spatiotemporelles actuelles ont du mal à résoudre. Les deux

étapes de cette stratégie seront détaillées au chapitre 5.

1.3.2 Omissions

L'approche que nous proposons aurait pu profiter d'autres améliorations que nous avons omis de développer ou d'implémenter suite à des contraintes de temps ou simplement pour mieux focaliser nos efforts de recherche. Les paragraphes suivants offrent un rapide survol des principales améliorations possibles à notre approche. Nous reviendrons sur ces aspects plus en détail dans la section 6.2, après avoir présenté notre approche avec plus de détail.

Tout d'abord, notre technique d'estimation des forces de contact au sol s'applique uniquement au cas du support double⁵. Il pourrait être intéressant de la généraliser à un support arbitraire (triple, quadruple, etc.). Cela ne s'avérerait probablement pas d'une grande utilité pour les mouvements de locomotion humaine, qui n'ont généralement pas plus de deux points d'appui, mais permettrait de généraliser la technique à des mouvements de créatures réelles ou fictives à plus de deux pattes.

Ensuite, comme nous le verrons au chapitre 5, les incréments de notre approche d'extrapolation progressive sont déterminés et ajustés par une méthode relativement simple qui peut augmenter significativement le temps de calcul nécessaire au processus d'extrapolation. Il serait possible de développer une méthode d'ajustement des incréments qui fasse un meilleur usage de l'information fournie par les incréments précédents afin de choisir plus judicieusement les incréments ultérieurs. À titre d'exemple, cela pourrait se faire par une régression linéaire ou quadratique sur la valeur des n incréments précédents ayant fait progresser l'extrapolation.

Dans nos modèles biomécaniques, les articulations sphériques, donc à trois degrés de liberté (DDLs) de rotation, sont modélisées comme une séquence de trois articulations à un DDL reliées par des corps fictifs de longueur et de masse nulles. Les coordonnées articulaires de ces articulations correspondent alors aux angles d'Euler. Bien que cela soit une pratique courante en animation, cette paramétrisation des rotations comporte des singularités qui peuvent nuire au processus d'optimisation. Une paramétrisation des rotations par *exponential map* serait préférable à ce niveau puisqu'elle ne comporte pas de singularité, sans ajouter de paramètres supplémentaires comme le ferait une représentation par quaternions. Cela ajoute toutefois une complexité considérable à

5. Nous traitons également les phases de support simple, mais ce problème possède une solution analytique qui ne requiert pas notre technique.

l'implémentation du système.

Finalemment, contrairement à Wampler et Popović [84], nous optimisons uniquement des paramètres continus. Cela veut donc dire que les temps de contact des pieds sont constants dans notre approche. Cela peut limiter la classe des mouvements que notre approche permet d'extrapoler.

Chapitre 2

Notation et fondements mathématiques

Ce chapitre présente au lecteur la notation mathématique qui sera employée tout au long du document ainsi que les notions mathématiques essentielles à une bonne compréhension du reste de l'ouvrage. L'introduction de ces notions est très succincte en ce sens qu'elle ne couvre que les éléments utiles au reste du document. À titre d'exemple, la théorie derrière les coordonnées homogènes est beaucoup plus vaste que la portion introduite dans le chapitre, mais cette portion est suffisante au développement des chapitres suivants.

2.1 Notation

Le tableau 2.1 présente et décrit les éléments principaux de la notation utilisée tout au long du présent ouvrage. La compréhension de certains éléments requiert des explications supplémentaires. Ces explications seront données au lecteur dans le texte lorsque la notation en question sera rencontrée pour la première fois.

D'autres éléments de notation, dont l'usage est limité à des portions précises du document, seront expliqués à l'intérieur du texte.

Notation	Explication
\mathbf{p}, \mathbf{A}	vecteur/matrice quelconque (coordonnées généralisées ou autres)
\vec{p}, A	vecteur/matrice cartésien(ne) (3 dimensions)
\tilde{p}, \tilde{A}	vecteur/matrice homogène (4 dimensions)
\hat{p}, \hat{A}	vecteur/matrice spatial(e) (6 dimensions)
A^t, A^{-1}	transposée et inverse multiplicatif d'une matrice A
\mathcal{R}_i	identifie le repère i
$[\vec{p}]_i, [\tilde{p}]_i, [\hat{p}]_i$	expression du vecteur p dans \mathcal{R}_i
${}^j R_i, {}^j \vec{t}_i$	matrice et vecteur euclidien de changement de repère (de \mathcal{R}_i vers \mathcal{R}_j)
${}^j \tilde{W}_i$	matrice homogène de changement de repère
${}^j \hat{X}_i$	matrice spatiale de changement de repère d'une variable cinématique
${}^j \hat{X}_i^*$	matrice spatiale de changement de repère d'une variable cinétique
\hat{x}	opérateur de produit vectoriel spatial d'une variable cinématique
\hat{x}^*	opérateur de produit vectoriel spatial d'une variable cinétique
\dot{x}, \ddot{x}	première et seconde dérivées temporelles de la variable x

TABLE 2.1 – Notation utilisée dans cet ouvrage

2.2 Types de vecteurs

Dans le cadre de cet ouvrage, nous manipulerons différents types de vecteurs : libres, glissants, spatiaux, cartésiens et homogènes. Nous donnons ici une description sommaire de chacun de ces vecteurs. Soulignons que le type de vecteur peut ne pas être explicitement spécifié dans le document si le contexte lève toute ambiguïté.

2.2.1 Vecteur libre

Un vecteur *libre* est entièrement défini par une direction et une amplitude. On peut visualiser un vecteur libre par un point dans l'espace ou par une flèche partant de l'origine et se terminant sur ce point. En 3 dimensions, un vecteur libre est représenté par un minimum de 3 coordonnées.

En dynamique des corps rigides, les vecteurs libres servent à représenter des quantités qui ne requièrent pas la spécification d'un point d'application. À titre d'exemple, une translation et un couple se représentent à l'aide de vecteurs libres.

Précisons ce que nous entendons lorsque nous disons que ces quantités ne nécessitent

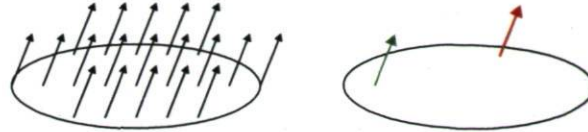


FIGURE 2.1 – Le champ de vecteurs d’une translation pure d’un corps rigide est constant (à gauche). Ce champ peut être représenté adéquatement par un seul vecteur libre (à droite). Dans la figure, les flèches rouge et verte représentent le même vecteur libre puisqu’elles sont de même direction et de même module.

pas la spécification d’un point d’application. Prenons la translation d’un corps rigide à titre d’exemple. Celle-ci peut être évaluée en tout point du corps rigide et même au-delà des limites du corps rigide. En ce sens, la translation d’un corps rigide peut être vue comme un champ de vecteurs, soit une fonction associant un vecteur libre à chaque point de l’espace. Il en va de même pour les autres quantités cinématiques et cinétiques décrivant la dynamique d’un corps rigide, y compris les quantités qui ne peuvent être représentées par un vecteur libre.

La figure 2.1 illustre l’idée d’un champ de vecteurs constant ; peu importe le point de l’espace où le champ est évalué, le vecteur obtenu est le même. Pour revenir à l’exemple de la translation d’un corps rigide, en absence de rotation, tous les points d’un corps rigide se déplacent de la même distance et dans la même direction. Ces champs peuvent donc être représentés adéquatement par un seul vecteur libre.

Les vecteurs libres pourront aussi nous servir à spécifier des positions dans \mathbb{R}^3 , telles que la position de la pointe du pied du personnage ou de son centre de masse par exemple.

2.2.2 Vecteur cartésien/homogène

Les vecteurs cartésiens et homogènes sont en fait des vecteurs libres. Un vecteur libre est dit *cartésien* lorsqu’il est exprimé en coordonnées cartésiennes et il est dit *homogène* lorsqu’il est exprimé en coordonnées homogènes. On parle alors des coordonnées cartésiennes et homogènes du même vecteur. Nous aborderons les différents types de coordonnées dans la prochaine section.



FIGURE 2.2 – Puisqu’elles sont de même direction, de même module et qu’elles se trouvent sur la même droite de support, les flèches rouge et verte représentent le même vecteur glissant.

2.2.3 Vecteur glissant

Tel qu’illustré à la figure 2.2, un vecteur *glissant* est essentiellement un vecteur libre contraint de se trouver sur une droite de support donnée. En 3 dimensions, un vecteur glissant est représenté par un minimum de 5 paramètres, soient 4 paramètres pour définir une droite et un paramètre pour spécifier le module du vecteur.

En dynamique des corps rigides, un vecteur glissant sert à représenter des quantités pour lesquelles il faut connaître la droite d’action du vecteur. À titre d’exemple, une rotation pure et une force appliquée à un corps rigide sont exprimés par des vecteurs glissants.

2.2.4 Vecteur spatial

Un vecteur spatial est représenté par un minimum de 6 paramètres. Un vecteur spatial est toujours exprimé en coordonnées spatiales. Les vecteurs libres et les vecteurs glissants peuvent être représentés en coordonnées spatiales (voir section 2.3.4). Un vecteur spatial général est issu de la somme d’un vecteur libre et d’un vecteur glissant lorsqu’ils sont exprimés en coordonnées spatiales.

Un vecteur spatial sert à représenter simultanément les composantes de translation et de rotation des diverses quantités utilisées en dynamique des corps rigides. Lorsqu’elles sont exprimées par un vecteur spatial, nous qualifierons ces quantités de *spatiales*. Les quantités spatiales utilisées dans ce document sont les suivantes :

Vélocité spatiale La vélocité spatiale regroupe la vitesse de translation et la vitesse angulaire d’un corps rigide.

Accélération spatiale L’accélération spatiale est la dérivée temporelle de la vélocité spatiale.

Quantité de mouvement spatiale Une quantité de mouvement spatiale regroupe

une quantité de mouvement et un moment angulaire.

Force spatiale Une force spatiale regroupe une force et un couple.

Un vecteur spatial peut aussi être vu comme un visseur ou un torseur. De fait, à l'exception d'une notation différente, un visseur (torseur) et un vecteur spatial peuvent représenter les mêmes quantités et se manipulent essentiellement de la même façon. La notation spatiale est toutefois beaucoup plus utilisée dans la littérature du domaine de l'animation basée sur la physique. C'est principalement pour cette raison que nous avons préféré la notation spatiale aux visseurs (torseurs) dans nos travaux et dans le présent document.

Les tenseurs cartésiens présentés par Balafoutis et Patel [6] forment une autre notation permettant d'exprimer les quantités cinématiques et cinétiques d'un corps rigide par un seul objet mathématique. Sans enlever de mérite à cette approche élégante, nous avons ici aussi préféré l'utilisation de la notation spatiale étant donné son utilisation beaucoup plus fréquente dans le domaine de l'animation basée sur la physique.

2.3 Types de coordonnées

Afin d'exprimer et de manipuler différents types de vecteurs, nous devons faire usage de différents types de coordonnées. Ceux-ci sont décrits dans cette section.

2.3.1 Coordonnées cartésiennes

Les coordonnées cartésiennes sont généralement utilisées dans les espaces réels à n dimensions, \mathbb{R}^n . Elles paramétrisent de façon unique un vecteur libre dans cet espace selon une base orthonormée. Dans le cadre de cet ouvrage, nous les utiliserons exclusivement dans \mathbb{R}^3 .

Étant donné un repère $\mathcal{R}_O = \vec{i}, \vec{j}, \vec{k}$ orthonormal, soit un repère pour lequel les vecteurs de base \vec{i} , \vec{j} et \vec{k} sont de norme unitaire et sont mutuellement orthogonaux, les coordonnées cartésiennes $(x, y, z)^t$ représentent le vecteur \vec{p} par l'égalité suivante :

$$\vec{p} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = x\vec{i} + y\vec{j} + z\vec{k}. \quad (2.1)$$

2.3.2 Coordonnées homogènes

Les coordonnées homogènes sont utilisées dans les espaces projectifs à $n + 1$ dimensions, \mathbb{P}^n , pour paramétriser un vecteur libre par une classe d'équivalence. Une classe d'équivalence est un ensemble de points dans \mathbb{P}^n dont chaque élément représente le même vecteur libre. En coordonnées homogènes, tout point $(x_1, x_2, \dots, x_{n+1})^t$, avec $x_{n+1} \neq 0$, est équivalent au point $(x_1/x_{n+1}, x_2/x_{n+1}, \dots, 1)^t$. Les coordonnées cartésiennes du vecteur libre paramétrisé par cette classe d'équivalence sont :

$$(x_1/x_{n+1}, x_2/x_{n+1}, \dots, x_n/x_{n+1})^t.$$

Puisque les vecteurs libres traités dans nos travaux sont tridimensionnels, nous utiliserons les coordonnées homogènes exclusivement dans \mathbb{P}^3 . De plus, au lieu de considérer les classes d'équivalence, nous nous intéresserons uniquement aux vecteurs homogènes dont la dernière coordonnée est égale à 1. Ainsi, le vecteur homogène $\tilde{p} = (x, y, z, 1)^t$ sera pour nous l'équivalent du vecteur cartésien $\vec{p} = (x, y, z)^t$. Ceci nous permet de définir une relation d'équivalence entre le vecteur cartésien \vec{p} et sa contrepartie homogène \tilde{p} :

$$\tilde{p} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \vec{p} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (2.2)$$

$$\vec{p} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tilde{p}. \quad (2.3)$$

Comme nous le verrons plus tard, l'utilisation des coordonnées homogènes nous permet d'exprimer une transformation rigide par une seule matrice. Cela nous permet de simplifier l'expression de plusieurs équations tout au long du document.

2.3.3 Coordonnées plückeriennes

Nous ne faisons pas directement usage des coordonnées plückeriennes dans nos travaux. Elles forment toutefois la base sur laquelle les coordonnées spatiales sont établies. Il convient donc de les définir.

Les coordonnées plückeriennes décrivent une droite dans \mathbb{R}^3 à l'aide de 6 paramètres, que l'on peut voir comme deux vecteurs cartésiens. Tel qu'illustré à la figure 2.3, le

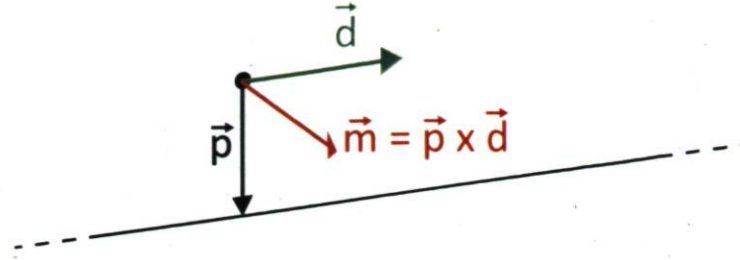


FIGURE 2.3 – Les coordonnées plückeriennes $(\vec{d}, \vec{m})^t$ sont utilisées pour décrire une droite dans \mathbb{R}^3 . \vec{d} est un vecteur de module arbitraire (non-nul) parallèle à la droite et \vec{p} rejoint un point arbitraire sur la droite. \vec{m} est le résultat du produit vectoriel $\vec{p} \times \vec{d}$.

vecteur \vec{d} est un vecteur de module arbitraire (non nul) parallèle à la droite et le vecteur \vec{m} est le résultat du produit vectoriel $\vec{p} \times \vec{d}$, où \vec{p} est un point arbitraire sur la droite. Le vecteur \vec{m} peut être vu comme le couple induit autour de l'origine par une force \vec{d} appliquée au point \vec{p} .

Les coordonnées $(\vec{d}, \vec{m})^t$ ne peuvent être issues que d'une seule droite et forment ensemble des coordonnées plückeriennes pour cette droite. À l'image des coordonnées homogènes, les coordonnées plückeriennes ne paramétrisent pas une droite de façon unique, mais plutôt par une classe d'équivalence¹. De fait, toute droite $(h\vec{d}, h\vec{m})^t$, avec $h \neq 0$, est équivalente à la droite $(\vec{d}, \vec{m})^t$. Modifier le facteur h revient en fait à modifier le module du vecteur \vec{d} , qui est arbitraire par définition.

L'intérêt des coordonnées plückeriennes en dynamique des corps rigides est qu'elles expriment une quantité à l'aide d'une composante de translation à 3 dimensions, soit le vecteur \vec{d} , et d'une composante angulaire à 3 dimensions, soit le vecteur \vec{m} , ce qui est tout à fait indiqué pour exprimer les quantités cinématiques et cinétiques d'un corps rigide. Même si nous ne les abordons pas sous le paradigme de la paramétrisation d'une droite, les coordonnées spatiales sont en fait des coordonnées plückeriennes que nous utilisons afin d'exprimer des vecteurs spatiaux.

2.3.4 Coordonnées spatiales

Les coordonnées spatiales sont généralement mieux introduites par un exemple. Soit un corps rigide sur lequel on choisit un point \vec{O} . La vitesse de translation du point \vec{O}

1. En fait, en adoptant une définition plus large pour les coordonnées homogènes, il est possible de montrer que les coordonnées plückeriennes sont une forme de coordonnées homogènes

est donnée par \vec{v}_O et la vitesse angulaire² du corps rigide est donnée par $\vec{\omega}$. On peut alors dire que le corps rigide se déplace avec une vitesse de translation \vec{v}_O en tournant à une vitesse angulaire $\vec{\omega}$ autour d'un axe passant par le point \vec{O} . Cette remarque est vraie peu importe le point \vec{O} choisit.

En définissant les coordonnées cartésiennes suivantes $\vec{\omega} = (\omega_x, \omega_y, \omega_z)^t$ et $\vec{v}_O = (v_{Ox}, v_{Oy}, v_{Oz})^t$, nous pouvons également exprimer cette vitesse de déplacement en coordonnées spatiales par le vecteur $\hat{v}_O = (\omega_x, \omega_y, \omega_z, v_{Ox}, v_{Oy}, v_{Oz})^t$. Un tel vecteur est exprimé par rapport à une base $\mathcal{D}_O = \{\hat{d}_{Ox}, \hat{d}_{Oy}, \hat{d}_{Oz}, \hat{d}_x, \hat{d}_y, \hat{d}_z\}$ définie sur l'espace \mathbb{M}^6 , soit l'espace des mouvements spatiaux. Les vecteurs de cette base sont illustrés à la figure 2.4. Les vecteurs \hat{d}_{Ox} , \hat{d}_{Oy} et \hat{d}_{Oz} représentent une rotation unitaire autour des axes Ox , Oy et Oz respectivement. Les vecteurs \hat{d}_x , \hat{d}_y et \hat{d}_z représentent une translation unitaire dans les directions x , y et z respectivement. Les coordonnées spatiales $(\omega_x, \omega_y, \omega_z, v_{Ox}, v_{Oy}, v_{Oz})^t$ représentent le vecteur de vitesse spatiale \hat{v}_O par l'égalité suivante :

$$\hat{v}_O = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \\ v_{Ox} \\ v_{Oy} \\ v_{Oz} \end{pmatrix} = \omega_x \hat{d}_{Ox} + \omega_y \hat{d}_{Oy} + \omega_z \hat{d}_{Oz} + v_{Ox} \hat{d}_x + v_{Oy} \hat{d}_y + v_{Oz} \hat{d}_z. \quad (2.4)$$

Une accélération spatiale \hat{a}_O est définie comme la dérivée temporelle d'une vitesse spatiale. Elle peut donc également être exprimée dans l'espace \mathbb{M}^6 selon la base \mathcal{D}_O .

Notons que si les trois premières composantes du vecteur spatial \hat{v}_O sont nulles ($\vec{\omega} = \vec{0}$), le vecteur spatial \hat{v}_O exprime un vecteur libre (une vitesse de translation dans cet exemple). Aussi, si les vecteurs $\vec{\omega}$ et \vec{v}_O ont un produit scalaire nul ($\vec{\omega} \cdot \vec{v}_O = 0$), alors le vecteur spatial \hat{v}_O exprime un vecteur ligne (la vitesse angulaire d'un corps rigide dans cet exemple). Cette observation s'applique à tous les vecteurs spatiaux issus des bases \mathcal{D}_O et \mathcal{E}_O définies dans cette section et illustrées à la figure 2.4.

La force la plus générale pouvant être appliquée à un corps rigide peut être décrite par une force \vec{f} appliquée le long d'une droite passant par un point \vec{O} et d'un couple \vec{n}_O autour du point \vec{O} . Cette force spatiale peut également être exprimée en coordonnées spatiales. Un tel vecteur sera exprimé par rapport à une base $\mathcal{E}_O = \{\hat{e}_x, \hat{e}_y, \hat{e}_z, \hat{e}_{Ox}, \hat{e}_{Oy}, \hat{e}_{Oz}\}$ définie sur l'espace des forces spatiales \mathbb{F}^6 . Les vecteurs de

2. La vitesse angulaire d'un corps rigide est identique pour tous les points appartenant à ce corps rigide, c'est pourquoi la vitesse angulaire du corps rigide n'est pas exprimée par rapport à un point spécifique comme l'est la vitesse de translation d'un corps rigide.

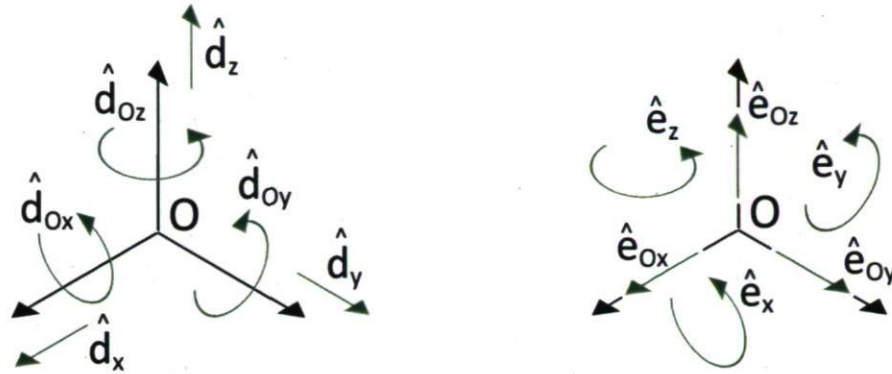


FIGURE 2.4 – Les bases plückeriennes définies sur M^6 (à gauche) et sur F^6 (à droite).

cette base sont similaires à ceux de la base \mathcal{D}_O et ils sont également illustrés à la figure 2.4. Les vecteurs \hat{e}_x , \hat{e}_y et \hat{e}_z représentent un couple unitaire dans les directions x , y et z respectivement. Les vecteurs \hat{e}_{Ox} , \hat{e}_{Oy} et \hat{e}_{Oz} représentent une force unitaire appliquée le long des axes Ox , Oy et Oz respectivement. Les coordonnées spatiales $(n_{Ox}, n_{Oy}, n_{Oz}, f_x, f_y, f_z)^t$ représentent le vecteur de force spatiale \hat{f}_O par l'égalité suivante :

$$\hat{f}_O = \begin{pmatrix} n_{Ox} \\ n_{Oy} \\ n_{Oz} \\ f_x \\ f_y \\ f_z \end{pmatrix} = n_{Ox}\hat{e}_x + n_{Oy}\hat{e}_y + n_{Oz}\hat{e}_z + f_x\hat{e}_{Ox} + f_y\hat{e}_{Oy} + f_z\hat{e}_{Oz}. \quad (2.5)$$

Les coordonnées spatiales nous permettent de formuler nos équations et de nos algorithmes de façon beaucoup plus succinctes. À titre d'exemple, un vecteur de vitesse spatiale exprime simultanément regroupe à la fois la vitesse de translation et la vitesse angulaire d'un corps rigide. Cela nous permet, dans plusieurs cas, de remplacer deux équations par une seule traitant sur ce vecteur de vitesse spatiale. Il en va de même pour les vecteurs de forces spatiales.

La notation spatiale introduit différents objets mathématiques et opérateurs qui lui sont spécifiques. Les objets mathématiques, tels que les tenseurs d'inertie spatiaux et les matrices de transformation rigides, seront introduits dans le texte lorsqu'ils seront utilisés pour la première fois.

Les opérateurs qui nous seront utiles en mécanique des systèmes multicorps sont l'addition de deux vecteurs spatiaux, la multiplication d'un vecteur spatial ou d'une matrice spatiale par un scalaire, la multiplication vecteur-matrice et matrice-matrice et

le produit vectoriel entre deux vecteurs spatiaux. À l'exception du produit vectoriel, tous les autres opérateurs sont identiques à leurs homologues en coordonnées cartésiennes. Le produit vectoriel nécessite toutefois une description plus détaillée.

En coordonnées cartésiennes, le produit vectoriel est souvent vu comme un opérateur binaire, soit un opérateur nécessitant deux opérandes. Il peut aussi être vu comme un opérateur unaire. Ainsi, le produit $\vec{u} \times \vec{v}$ peut être vu comme $(\vec{u} \times) \vec{v}$, où $\vec{u} \times$ est la matrice antisymétrique suivante :

$$\vec{u} \times = \begin{pmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{pmatrix}. \quad (2.6)$$

Il suffit ensuite de faire le produit entre cette matrice et le vecteur \vec{v} . C'est sous la forme d'opérateur unaire que nous allons introduire le produit vectoriel en coordonnées spatiales.

Définissons deux opérateurs de produit vectoriel spatiaux distincts, que nous représentons par les symboles $\hat{\times}$ et $\hat{\times}^*$:

$$\hat{v}_O \hat{\times} = \begin{pmatrix} \vec{\omega} \\ \vec{v}_O \end{pmatrix} \hat{\times} = \begin{pmatrix} \vec{\omega} \times & 0 \\ \vec{v}_O \times & \vec{\omega} \times \end{pmatrix} \quad (2.7)$$

$$\hat{v}_O \hat{\times}^* = \begin{pmatrix} \vec{\omega} \\ \vec{v}_O \end{pmatrix} \hat{\times}^* = \begin{pmatrix} \vec{\omega} \times & \vec{v}_O \times \\ 0 & \vec{\omega} \times \end{pmatrix}. \quad (2.8)$$

Le premier produit vectoriel donne une matrice que l'on peut multiplier par un élément de \mathbf{M}^6 alors que le second, soit $\hat{v}_O \hat{\times}^*$, donne une matrice que l'on peut multiplier par un élément de \mathbf{F}^6 . À titre d'exemple, à l'aide des vecteurs spatiaux $\hat{m} \in \mathbf{M}^6$ et $\hat{f} \in \mathbf{F}^6$, nous sommes en mesure d'exprimer les produits vectoriels $\hat{v}_O \hat{\times} \hat{m}$ et $\hat{v}_O \hat{\times}^* \hat{f}$.

Dans ce document, nous ne présentons qu'un bref survol de la notation spatiale. Le lecteur intéressé est invité à consulter un ouvrage de référence sur le sujet, tel que [22] ou [23].

2.3.5 Coordonnées généralisées

Au sens large, tout ensemble de paramètres qui déterminent de façon unique l'état d'un système, soit la configuration du personnage dans notre cas, sont des coordonnées généralisées de ce système. Nous ferons toutefois usage du terme coordonnées généralisées exclusivement pour désigner l'ensemble des valeurs des différentes coordonnées articulaires du personnage.

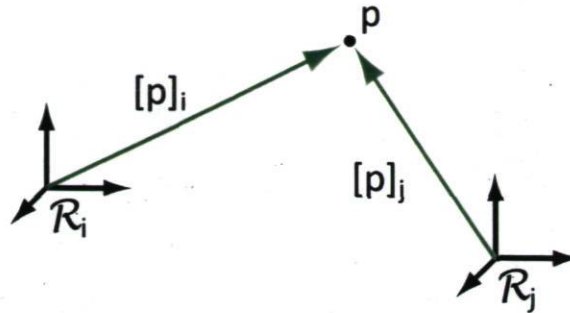


FIGURE 2.5 – Les coordonnées du point p changent selon le repère dans lequel il est exprimé.

En mécanique lagrangienne, on dérive les équations de mouvement d'un système directement à partir des coordonnées généralisées. Toutefois, même si nous traitons notre problème en mécanique newtonienne, les coordonnées généralisées demeurent un outil de choix pour représenter la configuration d'un personnage. Tout d'abord, elles décrivent une configuration par un nombre minimal de paramètres indépendants, ce qui contribue à simplifier les problèmes d'optimisation que nous poserons sur le mouvement. De plus, comme nous le verrons plus loin, on peut facilement obtenir le mouvement d'un corps rigide en coordonnées spatiales à partir d'un mouvement décrit en coordonnées généralisées.

2.4 Transformations rigides

Une transformation rigide générale est composée d'une translation et d'une rotation. Ces transformations forment une composante essentielle de la formalisation de notre problème. Il convient donc d'y consacrer une section avant de s'avancer davantage dans la description du problème et de la solution proposée.

Nous représentons le personnage par une arborescence de corps rigides. Comme nous le verrons plus tard, un repère local sera associé à chacun de ces corps rigides. Les transformations rigides nous serviront donc à prendre une valeur connue dans un repère et à l'exprimer par rapport à un autre repère. En effet, tel qu'illustré à la figure 2.5, les coordonnées d'un point p prendront des valeurs différentes selon le repère dans lequel le point est exprimé. Lorsque nécessaire, nous utiliserons donc la notation $[p]_i$ pour indiquer que le point p est exprimé dans le repère i .

Cette section introduit d'abord la notion de changement de repères en coordon-

nées cartésiennes, afin de présenter le concept dans son expression la plus simple. Nous présentons ensuite les changements de repères en coordonnées homogènes puis en coordonnées spatiales.

2.4.1 Coordonnées cartésiennes

En coordonnées cartésiennes, la transformation rigide d'un vecteur passe par deux opérations : la multiplication du vecteur par une matrice de rotation et l'addition d'un vecteur de translation. La figure 2.6 présente ce concept et montre que la transformation permettant d'exprimer un vecteur dans un nouveau repère correspond à l'inverse de la transformation qui aligne le repère d'origine sur le nouveau repère.

La transformation permettant d'exprimer un vecteur \vec{p} dans un nouveau repère prend la forme suivante

$$[\vec{p}]_j = {}^jR_i[\vec{p}]_i + {}^j\vec{t}_i, \quad (2.9)$$

où i et j sont respectivement les repères d'origine et de destination et où jR_i et ${}^j\vec{t}_i$ sont respectivement la matrice de rotation et le vecteur de translation de cette transformation.

Le vecteur de translation ${}^j\vec{t}_i$ est un vecteur à 3 dimensions. Il est égal à l'expression de l'origine du repère i dans le repère j

$${}^j\vec{t}_i = [\vec{O}_i]_j. \quad (2.10)$$

En haut à droite de la figure 2.6, on peut facilement voir que le vecteur $-\vec{\delta}$ correspond à l'origine du repère 1 exprimée dans le repère 3.

La matrice de rotation jR_i est une matrice 3x3 qui permet d'exprimer le vecteur $[\vec{p}]_i$ dans un repère ayant la même orientation que le repère j . Cette matrice peut être construite de nombreuses façons, mais nous n'aborderons ici que l'approche des angles d'Euler³. Au sens large, les angles d'Euler permettent de représenter toute rotation par la composition de trois rotations élémentaires. Une rotation élémentaire est une rotation d'un angle θ autour d'un axe du repère. Les matrices de rotation élémentaires

3. Nous utilisons la notion d'angles d'Euler au sens large, sans nous limiter à une convention spécifique quant aux axes de rotation utilisés et à l'ordre selon lequel les rotations sont appliquées

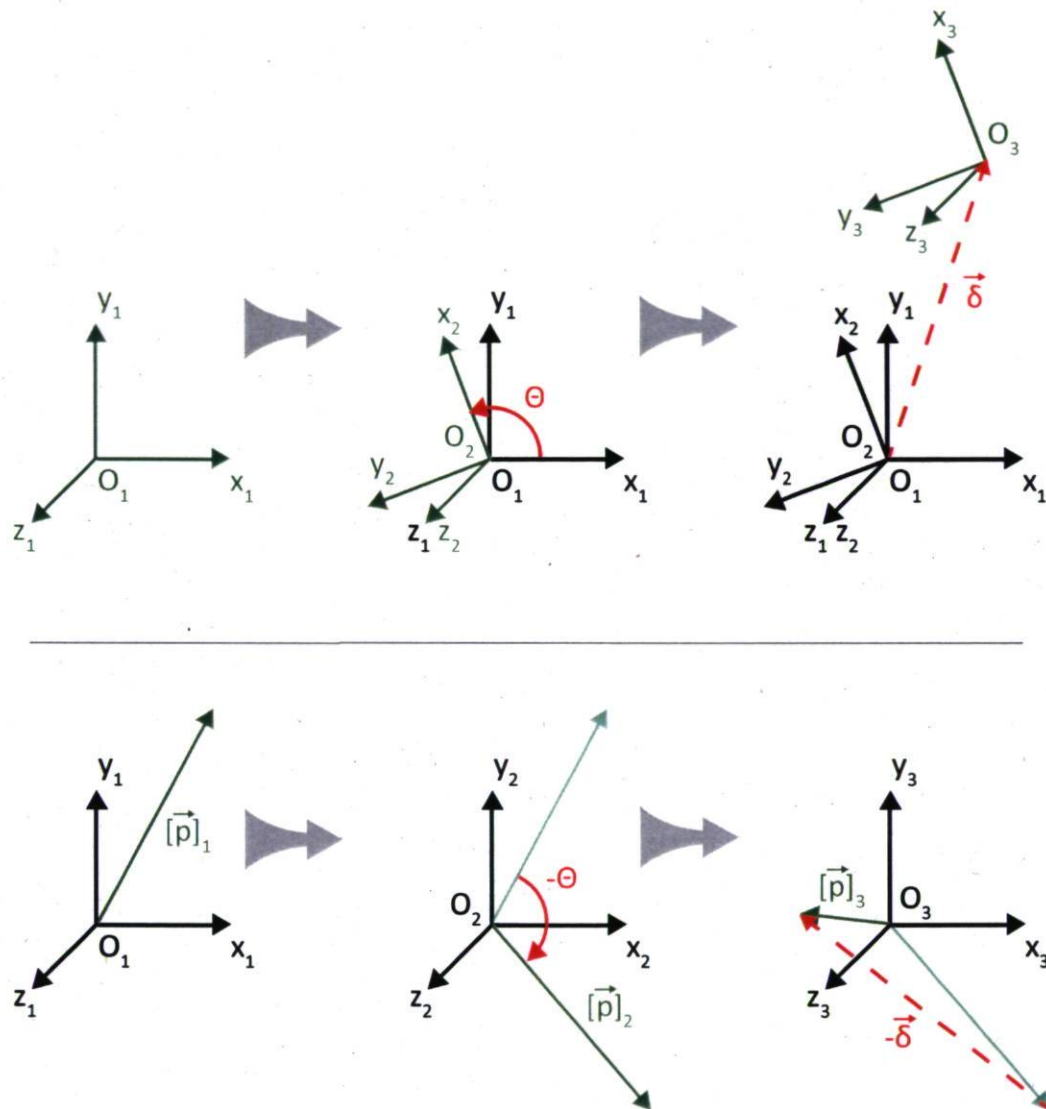


FIGURE 2.6 – En haut, le repère 1 subit une rotation d'angle θ autour de l'axe z_1 puis une translation $\vec{\delta}$ pour former le repère 3. En bas, un vecteur exprimé dans le repère 1 subit une rotation d'angle $-\theta$ autour de l'axe z_1 , suite à quoi on lui additionne $-\vec{\delta}$ afin d'obtenir son expression dans le repère 3. Soulignons que le vecteur de translation $\vec{\delta}$ a une orientation différente dans l'image étant donné que cette image est orientée selon le repère 3.

sont les suivantes :

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{pmatrix} \quad (2.11)$$

$$R_y(\theta) = \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \quad (2.12)$$

$$R_z(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.13)$$

Notons que nous avons $R_x(\theta)^{-1} = R_x(\theta)^t = R_x(-\theta)$ et qu'il en est de même pour $R_y(\theta)$ et $R_z(\theta)$. Cela signifie que l'inverse d'une matrice de rotation est égal à sa transposée et à la rotation inverse. La composition de ces rotations se fait par la multiplication des matrices de la droite vers la gauche. Ainsi, pour représenter une rotation d'angle θ_x autour de l'axe des x suivie d'une rotation d'angle θ_y autour de l'axe des y puis d'une rotation d'angle θ_z autour de l'axe des z , nous effectuons la multiplication suivante :

$$R_{xyz}(\theta_x, \theta_y, \theta_z) = R_z(\theta_z)R_y(\theta_y)R_x(\theta_x).$$

À titre informatif, la transformation rigide inverse de celle présentée à l'équation peut se développer de la façon suivante :

$$\begin{aligned} {}^j R_i [\vec{p}]_i + {}^j \vec{t}_i &= [\vec{p}]_j \\ [\vec{p}]_i &= {}^j R_i^t ([\vec{p}]_j - {}^j \vec{t}_i) \\ [\vec{p}]_i &= {}^j R_i^t [\vec{p}]_j - {}^j R_i^t {}^j \vec{t}_i \\ [\vec{p}]_i &= {}^i R_j [\vec{p}]_j + {}^i \vec{t}_j. \end{aligned}$$

Nous désirons ici souligner l'égalité ${}^i \vec{t}_j = -{}^j R_i^t {}^j \vec{t}_i$ que nous reverrons en coordonnées homogènes.

2.4.2 Coordonnées homogènes

L'expression de vecteurs en coordonnées homogènes présente pour nous un avantage intéressant par rapport à la représentation cartésienne, soit la possibilité d'exprimer une transformation rigide par une seule matrice. Ainsi, en coordonnées homogènes l'équation 2.4.1 devient :

$$[\vec{p}]_j = {}^j \tilde{W}_i [\vec{p}]_i, \quad (2.14)$$

avec la matrice homogène ${}^j\tilde{W}_i$ définie comme :

$${}^j\tilde{W}_i = \begin{pmatrix} & {}^jR_i & & {}^j\vec{t}_i \\ & & & \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.15)$$

Cette représentation plus succincte d'une transformation rigide est particulièrement utile pour représenter la composition de multiples transformations rigides. En effet, à l'image des rotations, la multiplication de deux matrices de transformations rigides homogènes produit une matrice correspondant à l'application successive de ces deux transformations rigides. Sous forme d'équation, nous avons donc :

$${}^k\tilde{W}_i = {}^k\tilde{W}_j {}^j\tilde{W}_i. \quad (2.16)$$

Finalement, la transformation inverse d'une matrice ${}^j\tilde{W}_i$ est donnée par :

$${}^j\tilde{W}_i^{-1} = {}^i\tilde{W}_j = \begin{pmatrix} & {}^iR_j & & {}^i\vec{t}_j \\ & & & \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} & {}^jR_i^t & & -{}^jR_i^t {}^j\vec{t}_i \\ & & & \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.17)$$

2.4.3 Coordonnées spatiales

Le changement de repère de vecteurs spatiaux se fait également par un produit matriciel. Pour un vecteur spatial de mouvement $\hat{m} \in \mathbb{M}^6$ et un vecteur spatial de force $\hat{f} \in \mathbb{F}^6$, le changement de repère s'exprime respectivement par l'équation

$$[\hat{m}]_j = {}^j\hat{X}_i [\hat{m}]_i \quad (2.18)$$

et

$$[\hat{f}]_j = {}^j\hat{X}_i^* [\hat{f}]_i. \quad (2.19)$$

La forme de la matrice de transformation rigide utilisée dépend du type de vecteurs spatiaux auquel elle s'applique, c'est pourquoi nous distinguons la matrice de transformation de forces spatiales par un astérisque. Les matrices de transformations spatiales peuvent s'exprimer en termes de composantes cartésiennes comme suit :

$${}^j\hat{X}_i = \begin{pmatrix} & {}^jR_i & & 0 \\ & ({}^j\vec{t}_i \times) {}^jR_i & & {}^jR_i \\ & & & \\ 0 & & & \end{pmatrix} \quad (2.20)$$

$${}^j\hat{X}_i^* = \begin{pmatrix} & {}^jR_i & & ({}^j\vec{t}_i \times) {}^jR_i \\ & & & \\ 0 & & & {}^jR_i \\ & & & \end{pmatrix}. \quad (2.21)$$

Les transformations inverses peuvent s'obtenir par les équations suivantes :

$${}^i\hat{X}_j = ({}^j\hat{X}_i^*)^t \quad (2.22)$$

$${}^i\hat{X}_j^* = ({}^j\hat{X}_i)^t. \quad (2.23)$$

Puisqu'elles peuvent toutes être exprimées en terme d'une matrice jR_i et d'un vecteur ${}^j\vec{t}_i$, il est trivial de construire une matrice spatiale ${}^j\hat{X}_i$ ou ${}^j\hat{X}_i^*$ à partir d'une matrice homogène ${}^j\tilde{W}_i$ et vice versa. Il est également trivial de passer d'une matrice de transformation de mouvement spatial ${}^j\hat{X}_i$ à une matrice de transformation de force spatiale ${}^j\hat{X}_i^*$ et vice versa.

Tout comme les matrices de transformation homogènes, les matrices de transformation spatiales peuvent être composées pour former une nouvelle matrice de transformation représentant l'application successive des deux transformations :

$${}^k\hat{X}_i = {}^k\hat{X}_j {}^j\hat{X}_i \quad (2.24)$$

$${}^k\hat{X}_i^* = {}^k\hat{X}_j^* {}^j\hat{X}_i^*. \quad (2.25)$$

Chapitre 3

Modélisation

Les éléments principaux de l'approche proposée sont le personnage, le mouvement, l'environnement de synthèse ainsi que la dynamique des systèmes multicorps. Ce chapitre vise à présenter chacun de ces éléments et à en donner une description suffisamment détaillée pour permettre au lecteur d'atteindre une compréhension approfondie des chapitres suivants.

3.1 Modélisation du personnage

Dans le présent ouvrage, un personnage est l'équivalent virtuel d'un être humain ou, par extension, d'un animal vertébré en général. Ce personnage est représenté par un système multicorps, soit un ensemble de corps rigides connectés par des articulations selon un certain graphe. Soulignons que nos algorithmes ne considèrent que les systèmes multicorps dont le graphe des connexions ne comporte aucun cycle.

3.1.1 Corps rigide

Un corps rigide est un objet physique théorique qui nous sert à représenter simplement des objets réels dont la déformation est négligeable. Le corps rigide est défini par une géométrie, une masse et un tenseur d'inertie. Pour un corps rigide donné, la géométrie et la masse sont constants. Le tenseur d'inertie sera constant s'il est exprimé par rapport à un repère attaché au solide, ce qui sera toujours le cas dans le cadre de cet ouvrage.

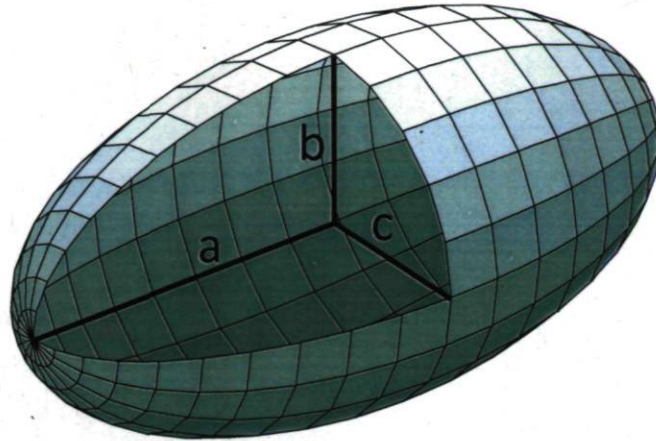


FIGURE 3.1 – Illustration d'un ellipsoïde de demi-axes a , b et c . Dans cet exemple, les demi-axes b et c sont égaux et inférieurs à a , il s'agit donc d'un ellipsoïde prolate.

Aux fins de simplicité, nous représentons les segments du corps humain par des ellipsoïdes prolates de densité uniforme (figure 3.1). La longueur d'un segment est estimée à l'aide des données de capture de mouvement. La masse et la densité du segment sont obtenues en consultant les tables d'anthropométrie de Winter [87] à partir de la masse totale du personnage. Étant donné la longueur l , la masse m et la densité ρ d'un segment, les demi-axes de l'ellipsoïde prolate correspondants sont :

$$a = l/2 \quad (3.1)$$

$$b = \sqrt{\frac{3m}{4\rho\pi a}} \quad (3.2)$$

$$c = b \quad (3.3)$$

Comme nous le verrons plus loin, nous implémentons une gestion des collisions fortement simplifiée. L'interpénétration des segments du corps est permise¹ et les contacts entre les pieds et le sol ne sont vérifiés que sur un petit nombre de points. La représentation ellipsoïdale d'un segment nous sert donc principalement à estimer le tenseur d'inertie de ce segment en son centre de masse par l'équation suivante :

$$I_C = \begin{pmatrix} m\frac{b^2+c^2}{5} & 0 & 0 \\ 0 & m\frac{a^2+c^2}{5} & 0 \\ 0 & 0 & m\frac{a^2+b^2}{5} \end{pmatrix}. \quad (3.4)$$

1. Étant donné que la représentation par ellipsoïdes n'est qu'une approximation de la géométrie d'une personne, empêcher la pénétration des segments du corps pourrait s'avérer être trop restrictif dans la reproduction de mouvements réalistes.

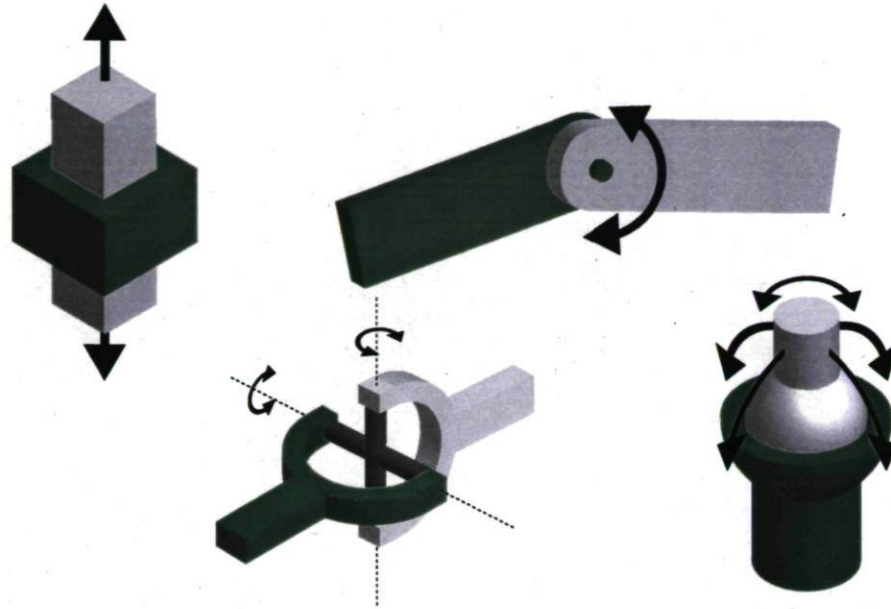


FIGURE 3.2 – Quelques exemples d'articulations couramment utilisées pour modéliser des systèmes mécaniques et biomécaniques. Le corps rigide vert représente le parent, par rapport auquel l'enfant se déplace selon les DDLs de l'articulation. De gauche à droite et de haut en bas nous présentons une articulation prismatique (1 DDL de translation), une articulation rotoïde (1 DDL de rotation), un joint de cardan (2 DDLs de rotation) et une rotule (3 DDLs de rotation).

3.1.2 Articulations

Une articulation sert à lier un corps rigide *parent* (ou *prédécesseur*) à un corps rigide *enfant* (ou *successeur*). Il représente une articulation biomécanique du personnage. L'articulation pose une contrainte sur le mouvement du corps successeur relativement au mouvement du corps prédécesseur. Selon le type d'articulation, ce mouvement relatif pourra s'effectuer sur 0 à 3 DDLs de translation et 0 à 3 DDLs de rotation. La figure 3.2 en présente quelques exemples courants.

Dans le cadre de nos travaux, nous considérons uniquement deux types d'articulations, soient prismatique et rotoïde. L'articulation prismatique permet un mouvement de translation le long d'un axe alors que l'articulation rotoïde permet un mouvement de rotation autour d'un axe. Nous représentons les articulations ayant plus d'un DDL par une séquence d'articulations à 1 DDL reliées par des corps rigides de dimension et de masse nulle. À titre d'exemple, un joint de cardan est représenté par une série de 2 articulations rotoïdes dont les axes de rotation sont perpendiculaires, tel qu'illustré par la figure 3.3. L'avantage principal de grouper des articulations à 1 DDL pour représen-

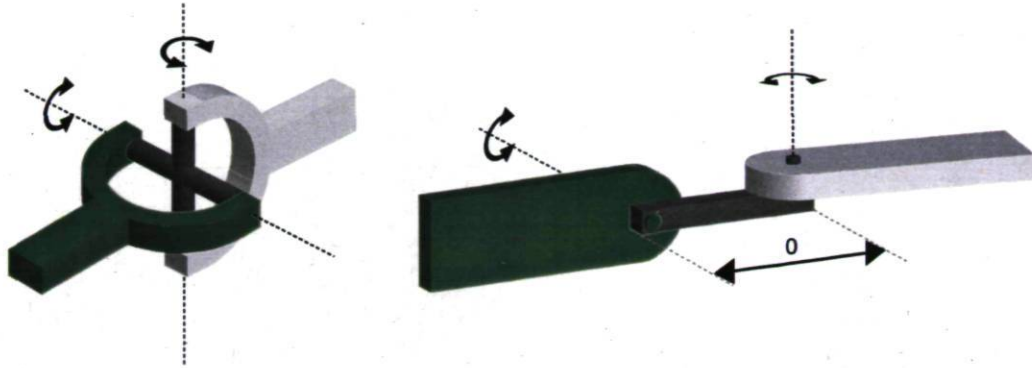


FIGURE 3.3 – Dans notre implémentation, un joint de cardan (à gauche) est représenté par deux articulations rotoïdes dont les axes de rotation sont orthogonaux. Le corps rigide entre ces deux articulations est de longueur et de masse nulle (à droite).

ter des articulations plus complexes est qu'il en résulte un modèle plus simple, ce qui permet de simplifier significativement les algorithmes utilisés.

Lorsque 3 articulations rotoïdes sont utilisées pour représenter une articulation de type *rotule*, il faut toutefois porter une attention particulière aux singularités qui peuvent survenir. En effet, dans un système de 3 articulations rotoïdes, l'orthogonalité des 3 axes de rotation n'est pas préservée dans le cas général. Plus précisément, l'angle donné à la deuxième articulation rotoïde déterminera si les axes de rotation de la première et troisième articulation sont perpendiculaires, colinéaires ou quelque part entre les deux. Lorsque ces deux axes sont colinéaires, on atteint la singularité connue sous le nom de *blocage de cardan* (figure 3.4)². Sous le paradigme des contraintes spatiotemporelles, cette singularité vient directement affecter notre capacité à résoudre le problème d'optimisation numérique, il vaut donc mieux tenter de l'éviter. Puisque les mouvements traités ne sont pas de grande amplitude, il est généralement facile de prévenir ce type de problème en orientant judicieusement les repères locaux de chaque articulation.

3.1.3 Système multicorps

Un système multicorps est un ensemble de corps rigides liés par des articulations. Plus particulièrement, puisque nous traitons de mouvements humains, le système multicorps forme une arborescence. À la base de cette arborescence se trouve une racine

2. Dans un contexte plus général, cette singularité est atteinte lorsque les trois axes de rotation sont coplanaires.

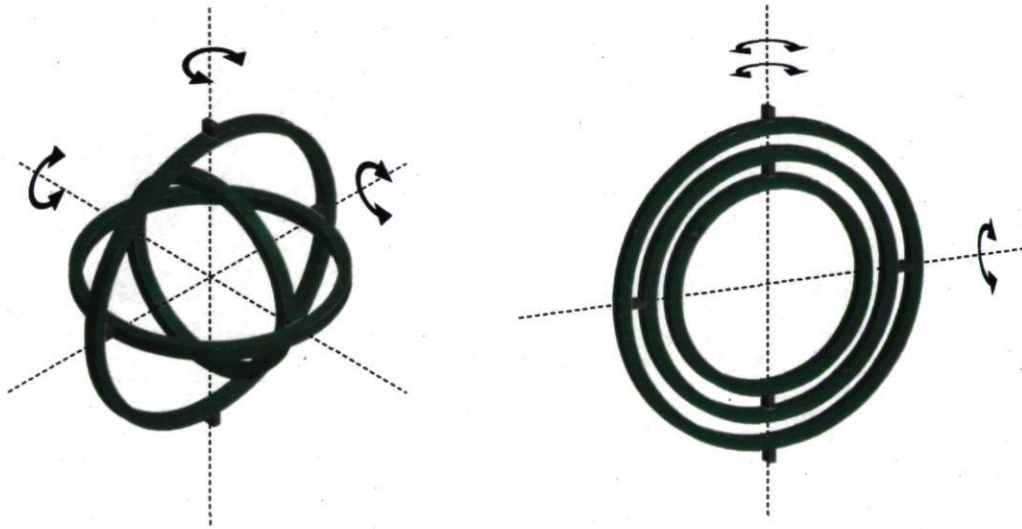


FIGURE 3.4 – Trois articulations rotoïdes imbriquées sont utilisées pour modéliser une articulation de type rotule. Dans leur configuration initiale (à gauche), les axes de rotation des articulations sont orthogonaux. Lorsqu'un blocage de cardan est atteint (à droite), les axes de l'articulation externe et de l'articulation interne sont colinéaires. Dans un contexte d'optimisation, cela signifie que les valeurs d'angle données aux articulations externe et interne s'additionnent. On peut donc attribuer une valeur arbitrairement grande à l'angle de l'articulation externe puisque l'angle de l'articulation interne peut toujours compenser pour cette valeur. Une infinité de ces combinaisons d'angles donnent donc une même orientation au corps enfant, un comportement qui peut nuire à la convergence de l'algorithme d'optimisation.

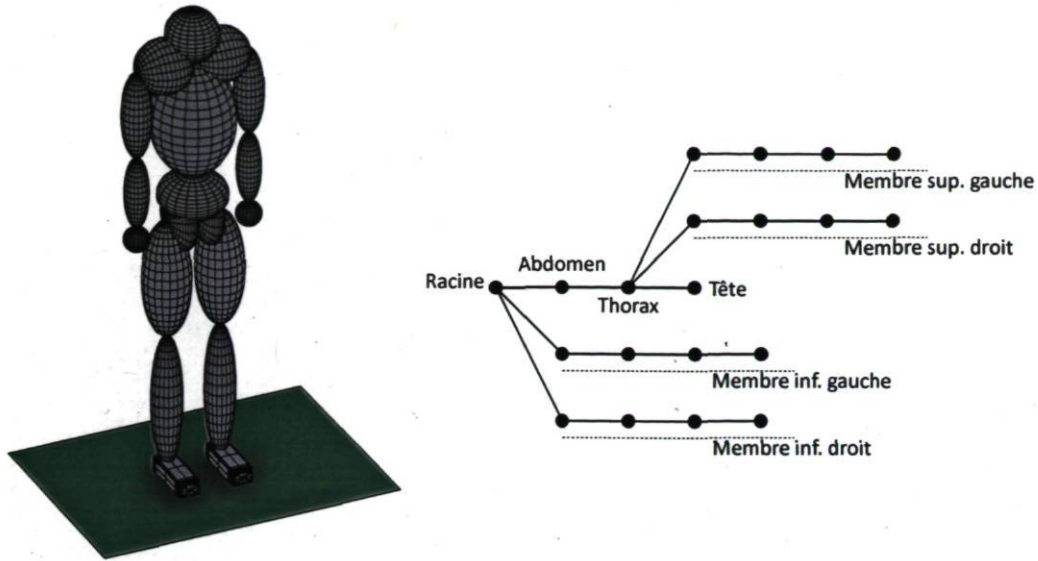


FIGURE 3.5 – Un exemple de système multicorps (à gauche) et le graphe de ses connexions (à droite). Les points du graphe représentent les corps rigides et les liens représentent les articulations.

à laquelle se rattachent des corps rigides ayant de 0 à n enfants. À l'exception de la racine, chaque corps rigide a un corps parent.

La racine est un corps rigide de masse et dimension nulles attachée à l'origine de l'environnement par une articulation à 6 DDLs³. C'est cette racine qui permet au personnage de se déplacer librement dans l'environnement, tant en rotation qu'en translation. L'emplacement de la racine sur le modèle du personnage est un choix arbitraire, mais dans la littérature en animation, elle est généralement placée au centre des hanches. La figure 3.5 présente un exemple de système multicorps et de son graphe de connexions.

Indices

Afin de simplifier les équations, chaque corps rigide est identifié d'un indice entier i . Ω est l'ensemble de tous ces indices. $\lambda(i)$ identifie le parent du corps i et $\phi(i)$ donne l'ensemble, possiblement nul, de ses enfants. $\varpi(i)$ donne l'ensemble de tous les ancêtres du corps i , inclusivement, alors que $\varphi(i)$ donne l'ensemble de tous ses descendants, inclusivement. La figure 3.6 présente des exemples de ces ensembles.

3. Dans notre implémentation, la racine est composée de 3 articulations prismatiques et de 3 articulations rotoïdes.

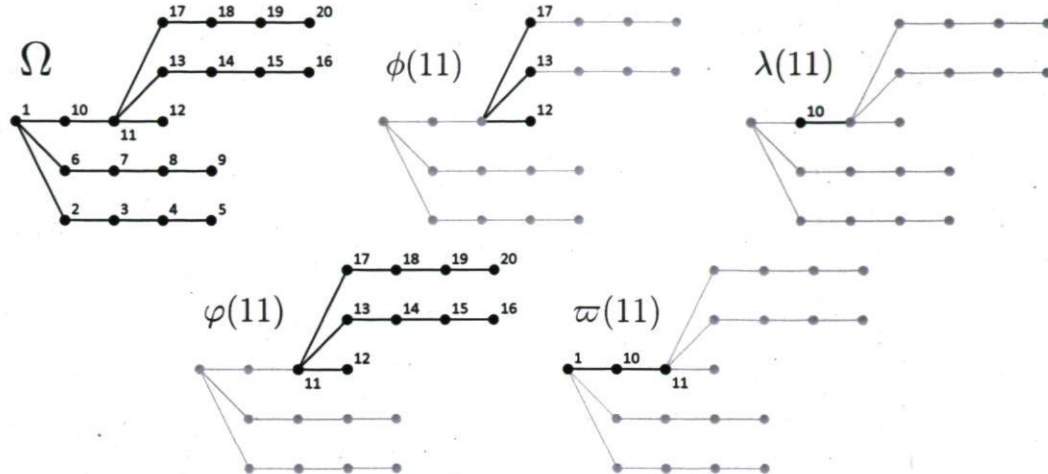


FIGURE 3.6 – Illustration du graphe des connexions d’un système multicorps et des ensembles qui s’y rattachent. De gauche à droite et de haut en bas nous présentons Ω , l’ensemble de tous les corps rigides, $\phi(11)$, l’ensemble des enfants du corps 11, $\lambda(11)$, le parent du corps 11, $\varphi(11)$, l’ensemble des descendants du corps 11 et $\psi(11)$, l’ensemble des ancêtres du corps 11. Rappelons que selon notre définition, l’ensemble des descendants et l’ensemble des ancêtres d’un corps i incluent le corps i . Bien que cela ne soit pas standard en théorie des graphes, cette définition nous permet de simplifier quelques équations.

Dans notre implémentation, nous nous assurons que les indices des corps parents sont plus petits que ceux des enfants ($\lambda(i) < i, \forall i \in \Omega$). Puisque plusieurs des algorithmes utilisés sont formulés sous forme récursive, parcourir les indices en ordre croissant, ou décroissant, nous donne un moyen simple de nous assurer que le calcul a déjà été effectué sur le corps parent avant d’être effectué sur le corps enfant, ou vice versa.

Notons qu’un indice i peut servir à identifier tant un corps rigide que l’articulation qui lie le corps i au corps parent $\lambda(i)$. Nous parlerons alors soit du corps i ou de l’articulation i .

Points d’attache

L’articulation i a un point d’attache sur le corps $\lambda(i)$ et un autre point d’attache sur le corps i . Ces points d’attache se déplacent de façon rigide avec les corps impliqués et, dans nos exemples, ils sont généralement situés aux extrémités de ces corps rigides. Toutefois, tel qu’illustré à la figure 3.7, ils peuvent se situer n’importe où par rapport à ces corps rigides.

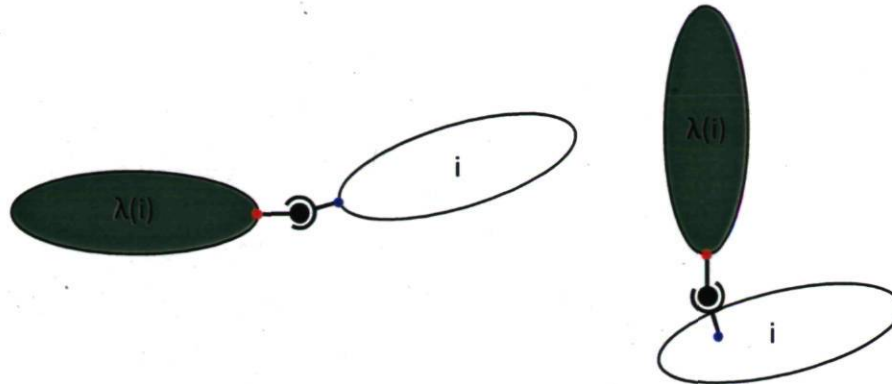


FIGURE 3.7 – L'articulation i est attachée au corps $\lambda(i)$ et au corps i . Le point rouge et le point bleu représentent le point d'attache sur le corps $\lambda(i)$ et sur le corps i respectivement. Notons que dans le cas d'une articulation rotoïde, ces deux points d'attache sont superposés puisque l'articulation a une longueur nulle. L'orientation des repères qui y sont attachés sera toutefois différente dans le cas général.

Repères locaux

Nous attribuons à chaque corps rigide un repère local dans \mathbb{R}^3 . Nous l'identifions également par l'indice du corps rigide correspondant. Le repère i répond toujours aux critères suivants :

- son origine est au point d'attache entre l'articulation i et le corps rigide i ;
- son axe z est aligné avec l'axe de l'articulation i ;
- son axe x est orienté vers le centre de masse du corps rigide i lorsque la masse et la longueur du corps i ne sont pas nulles.

Tel que mentionné dans la section 2.1, nous utilisons la notation $[\vec{p}]_i$ pour indiquer que le point \vec{p} est exprimé dans le repère i . Toutefois, puisque cette notation peut alourdir significativement certaines équations, nous considérerons qu'une valeur relative à un corps rigide i , telle que sa vitesse angulaire $\vec{\omega}_i$ est exprimée dans son repère local sauf si nous spécifions explicitement qu'il en est autrement. Cela revient à dire que nous admettons l'équivalence des notations $[\vec{v}_i]_i$ et \vec{v}_i . L'expression de cette valeur dans un autre repère sera toutefois toujours spécifiée explicitement par la forme $[\vec{v}_i]_j$.

Nous définissons également un second repère local en lien avec l'articulation i , soit le repère intermédiaire i' . Le repère intermédiaire i' répond toujours aux critères suivants :

- son origine est au point d'attache entre l'articulation i et le corps rigide $\lambda(i)$;
- son axe z est aligné avec l'axe de l'articulation i .

Le repère intermédiaire est orienté de façon à ce que l'axe de rotation (ou de translation) de l'articulation i corresponde à l'axe z du repère intermédiaire.

Tel qu'illustré à la figure 3.8, le repère intermédiaire i' est lié au repère $\lambda(i)$ par une transformation rigide qui demeure constante tout au long du mouvement. Cette transformation est définie dans le modèle biomécanique. Le repère local i est lié au repère intermédiaire i' par une transformation rigide qui est strictement fonction de l'articulation i .

Le repère intermédiaire i' permet donc de scinder en deux la transformation rigide liant le repère local $\lambda(i)$ au repère local i en offrant d'une part la partie constante de cette transformation et d'autre part la partie issue de la configuration de l'articulation i . Cela devient particulièrement utile lorsque les dérivées temporelles de ces transformations rigides doivent être établies.

Nous reviendrons sur ces repères dans la section 3.3 alors que nous aborderons la paramétrisation du mouvement.

3.1.4 Contact avec le sol

Afin de réduire le nombre de paramètres des animations, nous considérons dans notre implémentation que seuls les pieds du personnage peuvent entrer en contact avec le sol. Ces contacts s'expriment de deux façons, soit par des points de contact et par une surface de contact.

De plus, afin de simplifier la modélisation du contact entre les pieds et le sol et des contraintes qui s'y rattachent, un pied est considéré comme un seul corps rigide. Il n'y a donc pas d'articulation au niveau des orteils. Cela peut sembler très restrictif face à la synthèse de mouvements de marche réalistes, mais comme nous le verrons plus tard, nous permettons une certaine pénétration du pied dans le sol. La surface de contact qui en résulte, soit l'intersection entre la boîte englobante du pied et le sol, permet de bien imiter la surface de contact réelle du pied même lorsque seule la plante du pied est en contact avec le sol.

Points de contact

Les points de contact sont des points liés au corps rigide représentant chaque pied. Ils sont utilisés pour imposer les contraintes de temps de contact, de non-pénétration du sol et de frottement. En termes simples, les contraintes de temps de contact et de non-pénétration assurent respectivement que chaque point de contact est sur le sol lorsqu'il

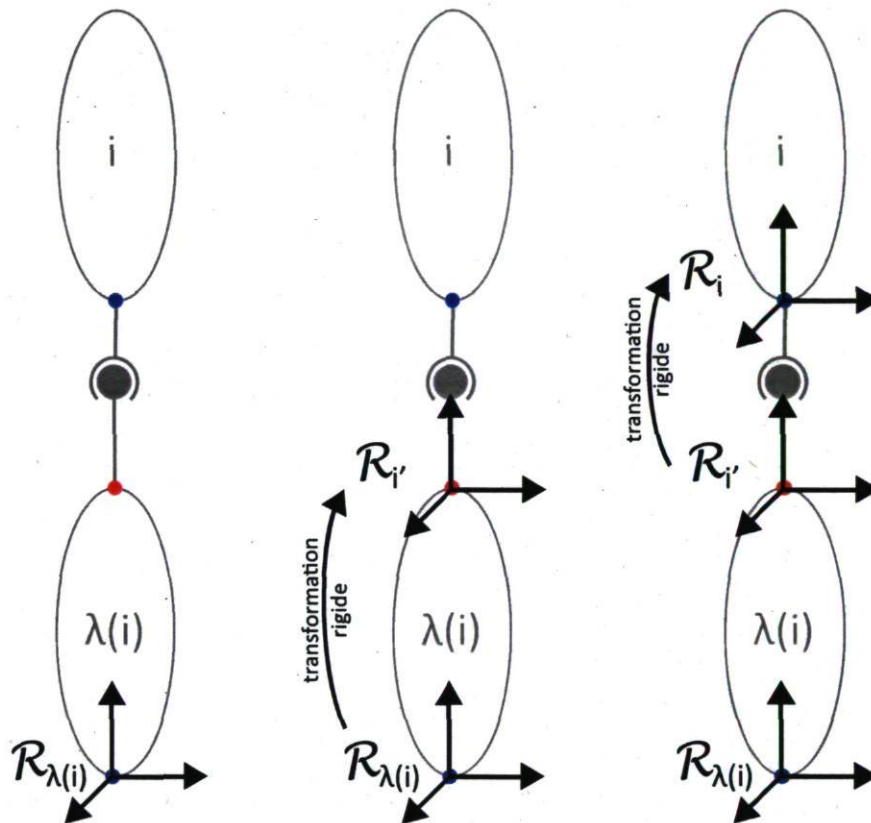


FIGURE 3.8 – Relation entre le repère $\lambda(i)$ et le repère i . Une transformation rigide est appliquée au repère $\lambda(i)$ pour obtenir le repère intermédiaire i' , qui correspond à l'orientation de l'articulation i et à son point d'attache par rapport au corps $\lambda(i)$. Le repère i' est ensuite modifié par l'articulation i , ce qui correspond à une seconde transformation rigide, pour obtenir le repère i .



FIGURE 3.9 – Illustration de deux points de contact sur le pied d'un personnage. Ces points permettent notamment de contrôler différentes parties du pied de façon indépendante, particulièrement quant aux temps de contact. Ainsi, il est possible de permettre au talon de quitter le sol tout en forçant la pointe du pied à rester au sol.

doit être en contact, qu'il est strictement au-dessus du sol autrement (figure 3.9). La contrainte de frottement assure qu'un point en contact ne bouge pas d'une trame à l'autre de l'animation. Ces contraintes seront décrites plus en détail dans le chapitre 5. Tel qu'illustré à la figure 3.9, nous utiliserons dans le cadre de cet ouvrage deux points de contact par pied, un au talon et un à la pointe du pied.

Surface de contact

La surface de contact est définie comme l'intersection du sol et de la boîte englobante du pied, alignée selon les axes du pied⁴. La surface de contact est utilisée pour imposer des contraintes sur les forces de contact en s'assurant que le centre de pression d'une force de contact est contenu dans la surface de contact.

Afin d'obtenir des surfaces de contact correspondant à celles obtenues par la déformation du pied et du soulier lors du contact avec le sol, les points de contact sont placés légèrement à l'intérieur de la boîte englobante. Cela permet donc une certaine pénétration de la boîte englobante dans le sol. La figure 3.10 illustre cette situation.

Certains chercheurs définissent les contacts sans surface en utilisant un minimum de quatre points de contact par pied [62, 49]. Ces points de contacts se trouvent à délimiter la surface inférieure du pied. Sous cette représentation, une force de contact est donnée comme la somme de forces appliquées à chacun de ces points. L'utilisation de la surface de contact conjointement avec des points de contact nous permet toutefois d'utiliser un plus petit nombre de points de contact, puisqu'ils ne servent qu'au respect des contraintes cinématiques. Par exemple, deux points de contact sont suffisants pour différencier les temps de contact du talon et de la pointe du pied. De plus, la surface de contact permet d'exprimer une force de contact en seulement six paramètres, par

4. La boîte englobante du pied est la boîte englobante de l'ellipsoïde représentant le pied.

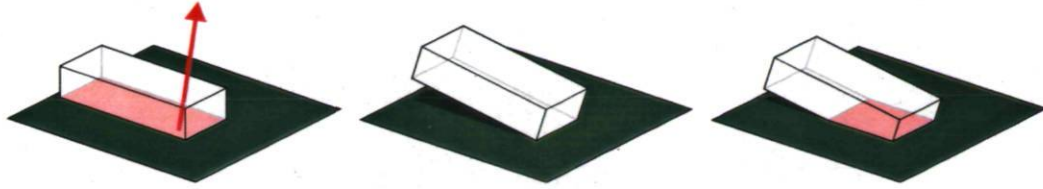


FIGURE 3.10 – Illustration de surfaces de contact entre un pied et le sol. Dans la figure de gauche, le talon et la pointe du pied sont en contact avec le sol et toute force de contact dont le centre de pression est contenu dans la surface rouge est valide. Dans les deux autres figures, seule la pointe du pied est en contact avec le sol. La surface de contact est limitée à un segment de droite lorsque la boîte englobante ne peut pénétrer dans le sol (au centre), alors qu'elle a une plus grande superficie lorsque la pénétration est permise (à droite). C'est le positionnement des points de contact qui détermine à quel point la boîte englobante peut pénétrer dans le sol.

rapport à trois paramètres par point de contact lorsque la surface de contact n'est pas utilisée. Un moins grand nombre de paramètres facilite le processus d'optimisation numérique.

3.2 Environnement de synthèse

L'environnement de synthèse détermine les conditions dans lesquelles le mouvement doit s'effectuer. L'environnement est composé d'un repère inertiel, d'une géométrie et de quelques constantes.

Le repère inertiel \mathcal{R}_0 est fixé dans l'environnement de synthèse. C'est relativement à ce repère que l'environnement et le mouvement sont décrits. Son origine est au point $(0, 0, 0)^t$ de l'espace \mathbb{R}^3 . De façon générale nous considérons que l'axe z est orienté vers le haut.

La géométrie de l'environnement sert principalement à indiquer où se situe le sol dans l'espace tridimensionnel. Dans notre implémentation, le sol est donné par l'équation d'un plan tel que $z = 0$ mais il pourrait en principe être représenté par n'importe quelle surface décrite dans un espace tridimensionnel.

Les constantes associées à l'environnement permettent de calculer la dynamique du personnage. Ces constantes sont le vecteur d'accélération gravitationnelle \vec{g} et le coefficient de frottement statique du sol. Le coefficient de frottement du sol sert à

définir le cône de frottement, soit le rapport maximum admis par le frottement entre les composantes parallèle et normale d'une force de contact au sol. Si une force de contact n'est pas contenue dans ce cône, il en résulterait un glissement. Le coefficient de frottement est donc utilisé dans les contraintes posées sur les forces de contact pour assurer que ces forces soient contenues dans le cône de frottement.

3.3 Paramétrisation du mouvement

Afin d'opérer sur des mouvements, il convient d'établir un ensemble de paramètres permettant de les décrire adéquatement. Dans cet ouvrage, nous ferons appel à trois paramétrisations distinctes, chacune ayant ses avantages propres.

3.3.1 Coordonnées généralisées

Un mouvement du personnage sera décrit par une matrice \mathbf{Q} dont la colonne t , soit le vecteur \mathbf{q}_t , donne la configuration du personnage au temps t en *coordonnées généralisées*. La valeur de l'élément $q_{i,t}$ correspond à la valeur prise par la i^e coordonnée articulaire du personnage au temps t . Sous l'hypothèse que toutes ses coordonnées articulaires sont indépendantes, donc que la variation de valeur d'une coordonnée n'affecte pas une autre coordonnée, les vecteurs \mathbf{q} décrivent chaque configuration du personnage avec un nombre minimum de paramètres.

$$\mathbf{Q} = \left(\mathbf{q}_1 \quad \mathbf{q}_2 \quad \dots \quad \mathbf{q}_T \right) = \begin{pmatrix} q_{1,1} & q_{1,2} & \dots & q_{1,T} \\ q_{2,1} & q_{2,2} & \dots & q_{2,T} \\ \vdots & \vdots & \ddots & \vdots \\ q_{n,1} & q_{n,2} & \dots & q_{n,T} \end{pmatrix} \quad (3.5)$$

Les coordonnées généralisées nous permettent aussi de décrire les forces généralisées appliquées à chacune des articulations (efforts articulaires). La force généralisée appliquée à la coordonnée articulaire i au temps t est donnée par l'élément $\tau_{i,t}$. Cette force généralisée représente la force appliquée par l'articulation i (attachée au corps rigide $\lambda(i)$) sur le corps rigide i . Le vecteur $\boldsymbol{\tau}_t$ regroupe les efforts articulaires du personnage au temps t et forme la colonne t de la matrice $\boldsymbol{\Gamma}$.

$$\Gamma = \begin{pmatrix} \tau_1 & \tau_2 & \dots & \tau_T \end{pmatrix} = \begin{pmatrix} \tau_{1,1} & \tau_{1,2} & \dots & \tau_{1,T} \\ \tau_{2,1} & \tau_{2,2} & \dots & \tau_{2,T} \\ \vdots & \vdots & \ddots & \vdots \\ \tau_{n,1} & \tau_{n,2} & \dots & \tau_{n,T} \end{pmatrix} \quad (3.6)$$

Différences finies

Les différentes configurations \mathbf{q}_t représentent en fait des échantillons réguliers d'un mouvement continu. Ainsi, la ligne i de la matrice \mathbf{Q} peut être vue comme les échantillons d'un signal continu donnant la valeur de la i^e coordonnée articulaire en fonction du temps.

Puisque nous manipulons un signal échantillonné, nous pouvons utiliser les différences finies pour en déterminer les premières et secondes dérivées temporelles. Nous définissons ces dérivées par les différences centrées suivantes :

$$\dot{q}_{i,t} = \frac{q_{i,t+1} - q_{i,t-1}}{2\Delta_t} \quad (3.7)$$

$$\ddot{q}_{i,t} = \frac{q_{i,t+1} - 2q_{i,t} + q_{i,t-1}}{\Delta_t^2} \quad (3.8)$$

où Δ_t est l'intervalle de temps entre deux échantillons. Ces équations correspondent à la première et la seconde dérivée temporelle évaluée au temps t de la courbe quadratique passant par les points $q_{i,t-1}$, $q_{i,t}$ et $q_{i,t+1}$.

Puisque le signal continu sous-jacent nous est inconnu, nous utilisons les différences finies pour *définir* ces dérivées et non seulement pour les *estimer*. Nous posons donc que le signal i représenté par les échantillons $q_{i,t}$ a les dérivées temporelles susmentionnées.

Ces dérivées seront utilisées à maintes occasions afin de décrire la cinématique des corps rigides et du système multicorps dans son ensemble.

3.3.2 Coordonnées cartésiennes et coordonnées homogènes

Si les coordonnées généralisées nous permettent d'exprimer une configuration à l'aide d'un nombre minimal de paramètres, elles ne permettent pas directement la reconstruction de cette configuration dans un espace euclidien. À titre d'exemple, un vecteur \mathbf{q}_t décrivant une configuration du personnage ne nous permet pas de connaître directement

la position et l'orientation du repère local du pied droit dans l'espace \mathbb{R}^3 . À cette fin, il nous faut convertir la configuration en coordonnées cartésiennes à l'aide du vecteur \mathbf{q}_t et du modèle du personnage. Ce dernier nous permet de connaître la longueur de chaque corps rigide et l'orientation de chaque articulation.

Afin de reconstruire la configuration du personnage dans \mathbb{R}^3 , nous devons pouvoir exprimer dans le repère inertiel des valeurs qui sont connues dans des repères locaux à l'aide d'une transformation rigide. Tel que nous l'avons vu au chapitre précédent, les matrices de changement de repère homogènes et spatiales peuvent être composées (équations 2.16, 2.24 et 2.25). Cette propriété nous permet d'obtenir la matrice de transformation d'un repère i au repère inertiel \mathcal{R}_0 par la composition de matrices *locales* de la forme ${}^{\lambda(i)}\tilde{W}_i$ puisque nous avons l'égalité suivante :

$${}^0\tilde{W}_i = {}^0\tilde{W}_1 \dots {}^{\lambda(\lambda(i))}\tilde{W}_{\lambda(i)} {}^{\lambda(i)}\tilde{W}_i \quad (3.9)$$

Comme nous l'avons présenté dans la section 3.1.3, la transformation entre le repère $\lambda(i)$ et le repère i dépend de deux éléments, soient :

- la position et l'orientation de l'articulation i sur le corps $\lambda(i)$
- et l'état de l'articulation i .

En d'autres termes, nous pouvons décomposer la matrice ${}^{\lambda(i)}\tilde{W}_i$ en deux matrices en passant par le repère intermédiaire i' :

$${}^{\lambda(i)}\tilde{W}_i = {}^{\lambda(i)}\tilde{W}_{i'} {}^{i'}\tilde{W}_i. \quad (3.10)$$

La matrice ${}^{\lambda(i)}\tilde{W}_{i'}$ sert donc à identifier la position et l'orientation du point d'attache de l'articulation i sur le corps $\lambda(i)$. Puisque le point d'attache bouge de façon rigide avec le corps $\lambda(i)$, cette matrice est une matrice de translation constante dans le temps. Elle dépend seulement du personnage et non de sa configuration. Ce faisant, les n matrices ${}^{\lambda(i)}\tilde{W}_{i'}$ font partie du modèle du personnage.

Par contre, la matrice ${}^{i'}\tilde{W}_i$ est une matrice de rotation ou de translation qui varie en fonction du temps et dépend de l'état de l'articulation i . Si l'articulation i est prismatique, alors la matrice ${}^{i'}\tilde{W}_i$ aura la forme :

$${}^{i'}\tilde{W}_i = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_i \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

où q_i est la i^e coordonnée généralisée de la configuration. Si l'articulation i est rotoïde,

cette matrice aura la forme suivante :

$${}^{i'}\tilde{W}_i = \begin{pmatrix} \cos(q_i) & -\sin(q_i) & 0 & 0 \\ \sin(q_i) & \cos(q_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Avec ces matrices, nous sommes en mesure de construire les matrices ${}^0\tilde{W}_i$ pour tout $i \in \Omega$ avec l'équation réursive suivante :

$${}^0\tilde{W}_i = {}^0\tilde{W}_{\lambda(i)} \lambda^{(i)} \tilde{W}_{i'} {}^{i'}\tilde{W}_i. \quad (3.11)$$

Nous sommes ensuite en mesure de dessiner la configuration du personnage dans l'espace \mathbb{R}^3 et de connaître la position de certains points, tels que les points de contact, par rapport au repère inertiel par l'équation :

$$[\tilde{p}]_0 = {}^0\tilde{W}_i [\tilde{p}]_i.$$

3.3.3 Coordonnées spatiales

Les coordonnées homogènes nous servent à décrire la configuration du personnage. Les coordonnées spatiales nous servent quant à elles à décrire la cinématique et la cinétique du personnage.

Puisque nous utilisons ici aussi les repères locaux de chaque corps rigide, nous devons ici aussi définir les matrices de changement de repère. Celles-ci ont les mêmes propriétés que leurs homologues homogènes. Notamment, la matrice ${}^i\hat{X}_{\lambda(i)}$ peut être décomposée par rapport au repère intermédiaire i'

$${}^i\hat{X}_{\lambda(i)} = {}^i\hat{X}_{i'} {}^{i'}\hat{X}_{\lambda(i)}, \quad (3.12)$$

où la matrice ${}^{i'}\hat{X}_{\lambda(i)}$ est constante et dépend du modèle du personnage et où la matrice ${}^i\hat{X}_{i'}$ dépend de l'état de l'articulation i . Il en va de même pour les matrices de changement de repère des forces spatiales.

Pour une articulation prismatique les matrices ${}^i\hat{X}_{i'}$ et ${}^i\hat{X}_{i'}^*$ auront la forme :

$${}^i\hat{X}_{i'} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & q_i & 0 & 1 & 0 & 0 \\ -q_i & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.13)$$

$${}^i\hat{X}_{i'}^* = \begin{pmatrix} 1 & 0 & 0 & 0 & q_i & 0 \\ 0 & 1 & 0 & -q_i & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.14)$$

Pour une articulation rotoïde, ces matrices auront la forme

$${}^i\hat{X}_{i'} = {}^i\hat{X}_{i'}^* = \begin{pmatrix} \cos(q_i) & \sin(q_i) & 0 & 0 & 0 & 0 \\ -\sin(q_i) & \cos(q_i) & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos(q_i) & \sin(q_i) & 0 \\ 0 & 0 & 0 & -\sin(q_i) & \cos(q_i) & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.15)$$

Ces matrices peuvent également être composées pour obtenir les transformations entre le repère inertiel et n'importe quel repère local⁵ :

$${}^i\hat{X}_0 = {}^i\hat{X}_{i'} {}^{i'}\hat{X}_{\lambda(i)}^{\lambda(i)} \hat{X}_0 \quad (3.16)$$

$${}^i\hat{X}_0^* = {}^i\hat{X}_{i'}^* {}^{\lambda(i)}\hat{X}_{\lambda(i)}^* \hat{X}_0^* \quad (3.17)$$

Sous-espace de mouvement

Un sous-espace est décrit par une base dont les vecteurs ne couvrent pas toutes les dimensions de l'espace dans lequel ils sont contenus. À titre d'exemple, deux vecteurs orthonormaux décrivent un sous-espace dans \mathbb{R}^3 , soit un plan dans un espace à 3 dimensions.

5. Nous présentons les transformations spatiales du repère inertiel vers les repères locaux étant donné l'utilisation que nous en ferons dans notre algorithme de dynamique inverse. Par opposition, nous avons présenté les transformations homogènes des repères locaux vers le repère inertiel puisqu'elles nous servent ultimement à exprimer des points par rapport au repère inertiel. Dans tous les cas, l'inverse des matrices de transformation homogènes et spatiales est facilement calculé à l'aide des formules données dans la section 2.4

On associe à chaque articulation i un sous-espace de mouvement S_i . S_i est une matrice dont les colonnes sont les vecteurs formant la base du sous-espace. Ces vecteurs correspondent aux axes de l'articulation i dans \mathbb{M}^6 et \mathbb{F}^6 . Puisque nous n'utilisons que des articulations à 1 DDL, les matrices S_i seront en fait composées d'une seule colonne, soit

$$S_{prismatique} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (3.18)$$

pour les articulations prismatiques et

$$S_{rotoïde} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (3.19)$$

pour les articulations rotôïdes. Ces matrices correspondent aux contraintes que nous avons posées sur les repères locaux, soit que l'axe z d'un repère i coïncide avec l'axe de l'articulation i .

Les matrices de sous-espaces de mouvements permettent de convertir une quantité cinématique généralisée en quantité spatiale. Pour une articulation i de vitesse \dot{q}_i et d'accélération \ddot{q}_i , la vitesse relative du corps rigide i par rapport au corps $\lambda(i)$ est donnée par la vitesse spatiale :

$$\hat{v}_{Ji} = S_i \dot{q}_i. \quad (3.20)$$

Exprimée dans le repère local, la vitesse spatiale absolue du corps rigide i est donnée par l'équation récursive suivante :

$$\hat{v}_i = {}^i \tilde{X}_{\lambda(i)} \hat{v}_{\lambda(i)} + \hat{v}_{Ji}. \quad (3.21)$$

L'accélération spatiale est définie comme la première dérivée temporelle de la vitesse spatiale. L'accélération relative du corps rigide i par rapport au corps $\lambda(i)$ est

$$\hat{a}_{Ji} = S_i \ddot{q}_i + \hat{v}_i \hat{\times} \hat{v}_{Ji} \quad (3.22)$$

ce qui correspond à la dérivée temporelle de l'équation 3.20, alors que son accélération absolue est donnée par :

$$\hat{a}_i = {}^i \tilde{X}_{\lambda(i)} \hat{a}_{\lambda(i)} + \hat{a}_{Ji}. \quad (3.23)$$

Les matrices de sous-espaces de mouvements permettent également de convertir une force spatiale en force généralisée. Ainsi, si une force spatiale \hat{f}_i doit être appliquée au corps rigide i , on sait que l'articulation i doit appliquer un effort articulaire $\tau_i = S_i^t \hat{f}_i$. Les autres composantes de la force sont en quelques sortes *absorbées* par les contraintes de l'articulation i , soient les axes selon lesquels l'articulation i ne peut pas bouger.

3.4 Dynamique des systèmes multicorps

Dans la section précédente, nous avons déjà défini la vitesse et l'accélération spatiale d'un corps rigide faisant partie d'un système multicorps. Afin de compléter notre description de la dynamique d'un système multicorps, nous devons maintenant nous attaquer aux quantités cinétiques, soient la quantité de mouvement et les forces spatiales. Avant de procéder, il faut toutefois définir l'inertie spatiale d'un corps rigide.

3.4.1 Inertie spatiale

L'équation 3.4 définit I_{Ci} , soit le tenseur d'inertie en coordonnées cartésiennes d'un ellipsoïde de demi-axes a_i , b_i et c_i . Rappelons que cet ellipsoïde représente le corps rigide i , qui a une longueur l_i et une masse m_i . L'équivalent en coordonnées spatiales est donné par :

$$\hat{I}_{Ci} = \begin{pmatrix} I_{Ci} & \mathbf{0} \\ \mathbf{0} & m_i \mathbf{1} \end{pmatrix}, \quad (3.24)$$

où $\mathbf{0}$ et $\mathbf{1}$ sont respectivement les matrices nulle et identité de dimension 3×3 . En définissant le vecteur cartésien $\vec{a}_i = (a_i, 0, 0)^t$, nous pouvons exprimer le tenseur d'inertie spatiale du corps i par rapport à son repère local par l'équation suivante :

$$\hat{I}_i = \begin{pmatrix} I_{Ci} + m_i \vec{a}_i \times (\vec{a}_i \times)^t & m_i \vec{a}_i \times \\ m_i (\vec{a}_i \times)^t & m_i \mathbf{1} \end{pmatrix}. \quad (3.25)$$

3.4.2 Équations de mouvement

La quantité de mouvement spatiale d'un corps rigide i est égale au produit entre l'inertie spatiale du corps rigide et sa vitesse :

$$\hat{e}_i = \hat{I}_i \hat{v}_i \quad (3.26)$$

et sa dérivée temporelle est donnée par

$$\dot{\hat{e}}_i = \hat{I}_i \hat{a}_i + \hat{v}_i \hat{\times}^* \hat{I}_i \hat{v}_i. \quad (3.27)$$

Par extension de la deuxième loi du mouvement de Newton, la variation de la quantité de mouvement d'un corps rigide doit être égale à la somme des forces qui sont appliquées à ce corps rigide. Si nous considérons uniquement le corps rigide i sans égard au reste du système multicorps, nous avons l'équation de mouvement suivante :

$$\hat{f}_i + \hat{f}_{ext,i} = \hat{I}_i \hat{a}_i + \hat{v}_i \hat{\times}^* \hat{I}_i \hat{v}_i, \quad (3.28)$$

où \hat{f}_i est la force appliquée par l'articulation i et $\hat{f}_{ext,i}$ est la somme des forces externes appliquées à ce corps rigide. Notons que tous les termes de cette équation sont exprimés par rapport au repère i , incluant la somme des forces externes.

Lorsque nous considérons que le corps rigide i est attaché à d'autres corps rigides dans un système multicorps, nous devons également considérer la quantité de mouvement des descendants de i dans l'algorithme de dynamique inverse ce qui nous donne :

$$\hat{f}_i + \hat{f}_{ext,i} = \sum_{j \in \phi(i)} {}^i \hat{X}_j^* \left(\hat{I}_j \hat{a}_j + \hat{v}_j \hat{\times}^* \hat{I}_j \hat{v}_j \right), \quad (3.29)$$

3.4.3 Dynamique inverse

Lorsque les équations de mouvements sont respectées pour un mouvement donné, celui-ci est considéré comme physiquement plausible. Il s'agit là d'une contrainte de premier plan en animation basée sur la physique. Différentes approches d'animation basées sur la physique assurent le respect des équations de mouvement de différentes façons. Plus particulièrement, les approches basées sur les contrôleurs utilisent des algorithmes de dynamique directe alors que les approches de contraintes spatio-temporelles utilisent généralement des algorithmes de dynamique inverse.

À titre de rappel, en dynamique directe on fixe les forces, soit le membre de gauche de l'équation 3.29, pour ensuite résoudre un système d'équations différentielles pour obtenir les valeurs de vitesse \hat{v}_i et d'accélération \hat{a}_i qui respectent l'équation de mouvement. En dynamique inverse, on fixe plutôt la cinématique du mouvement, soit le membre de droite de l'équation 3.29, ce qui nous permet de calculer les forces ayant produit ce mouvement. Si les forces externes sont connues, nous utilisons plutôt l'équation récursive suivante pour déterminer les forces internes du personnage :

$$\hat{f}_i = \hat{I}_i \hat{a}_i + \hat{v}_i \hat{\times}^* \hat{I}_i \hat{v}_i - \hat{f}_{ext,i} + \sum_{j \in \phi(i)} {}^i \hat{X}_j^* \hat{f}_j. \quad (3.30)$$

Les approches de contraintes spatio-temporelles ajustent parallèlement les paramètres de mouvement et de force. Les équations de mouvement sont alors vues comme une contrainte dans le processus d'optimisation : on doit s'assurer que les paramètres de force et de mouvement respectent l'équation 3.29. En ce sens, il serait également possible d'utiliser un algorithme de dynamique directe pour vérifier le respect de cette contrainte. En effet, dans ce cas, les algorithmes de dynamique directe et de dynamique inverse servent tous à mettre en relation les forces et le mouvement. Les algorithmes de dynamique inverse sont toutefois largement préférés étant donné qu'ils sont plus simples à implémenter et à dériver. Ce dernier critère est particulièrement important en optimisation numérique.

Algorithme de Newton-Euler récursif

Afin de calculer la dynamique inverse d'un mouvement \mathbf{Q} , nous utilisons l'algorithme de Newton-Euler récursif dont la complexité est de l'ordre de $\mathcal{O}(n)$. Cet algorithme parcourt l'arborescence du système multicorps à deux reprises. Dans un premier temps, un parcours de la racine vers les feuilles permet d'obtenir les vitesses et accélérations absolues de chaque corps rigide. Dans un deuxième temps, un parcours des feuilles vers la racine permet de calculer le taux de variation de la quantité de mouvement de chaque corps rigide. Pour une configuration \mathbf{q}_t de dérivées $\dot{\mathbf{q}}_t$ et $\ddot{\mathbf{q}}_t$, l'algorithme de Newton-Euler récursif est :

Entrées : $model, \mathbf{q}_t, \dot{\mathbf{q}}_t, \ddot{\mathbf{q}}_t, \hat{f}_{ext,1}, \dots, \hat{f}_{ext,n}$
Sorties : τ_t, \hat{f}_0
 $\hat{v}_0 = \hat{0}$
 $\hat{a}_0 = -\hat{g}$
 $\hat{f}_0 = \hat{0}$
pour $i = 1$ **à** n **faire**
 ${}^i\hat{X}_{i'} = jointTr(model, \mathbf{q}_t, i)$
 ${}^i\hat{X}_{\lambda(i)} = {}^i\hat{X}_{i'} {}^{i'}\hat{X}_{\lambda(i)}$
 $\hat{v}_i = {}^i\hat{X}_{\lambda(i)} \hat{v}_{\lambda(i)} + S_i \dot{q}_{i,t}$
 $\hat{a}_i = {}^i\hat{X}_{\lambda(i)} \hat{a}_{\lambda(i)} + S_i \ddot{q}_{i,t} + \hat{v}_i \times S_i \dot{q}_{i,t}$
 $\hat{f}_i = \hat{I}_i \hat{a}_i + \hat{v}_i \times \hat{I}_i \hat{v}_i - \hat{f}_{ext,i}$
fin
pour $i = n$ **à** 1 **faire**
 $\tau_{i,t} = S_i^t \hat{f}_i$
 $\hat{f}_{\lambda(i)} = \hat{f}_{\lambda(i)} + {}^{\lambda(i)}\hat{X}_i^* \hat{f}_i$
fin

Algorithme 1: Algorithme Newton-Euler récursif

Dans cet algorithme, *model* est le modèle du personnage et la fonction *jointTr* est simplement une fonction qui retourne la matrice de transformation associée à l'articulation i en fonction du type d'articulation, une information contenue dans *model*, et de la i^e valeur du vecteur \mathbf{q}_t . Elle réfère donc à l'équation 3.13 si l'articulation i est prismatique et à l'équation 3.15 si elle est rotoïde. Le vecteur spatial \hat{g} est le vecteur spatial d'accélération gravitationnelle :

$$\hat{g} = \begin{pmatrix} \vec{0} \\ \vec{g} \end{pmatrix}.$$

Il est soustrait de l'accélération pour faire état qu'en présence d'un champ gravitationnel, un corps doit subir une force de direction opposée à l'accélération gravitationnelle afin de demeurer *au repos*.

L'algorithme 1 retourne à la fois τ_t , les forces internes du personnage en coordonnées généralisées, et \hat{f}_0 , qui représente les forces internes devant être appliquées à la racine du personnage. Connaître le vecteur \hat{f}_0 est utile pour implémenter une approche similaire à celle de Fang et Pollard [20, 21], qui ne requiert pas la connaissance explicite des forces internes du personnage.

Chapitre 4

Estimation des forces de contact

Lors de la marche, et de bien d'autres mouvements, le personnage exerce une force contre le sol par l'action de ses muscles et par l'effet de l'accélération gravitationnelle sur sa masse. En contrepartie, tel que postulé par le principe des actions réciproques de Newton, le personnage subit du sol une force de réaction de même amplitude et de sens opposé.

Cette force qui est appliquée au personnage par le sol est généralement appelée force de réaction du sol. Elle est évidemment nulle lorsqu'il n'y a pas de contact entre le pied et le sol. Notons que la composante normale de la force de réaction du sol, soit la projection du vecteur de force sur le vecteur normal du sol, ne peut être négative. En d'autres termes, la force de réaction du sol ne peut pas tirer le personnage en direction du sol, elle ne peut que le pousser.

Le contact entre le pied et le sol ne se fait généralement pas en un seul point, mais sur une surface. On peut donc voir la force de réaction du sol comme l'intégrale sur la surface des forces appliquées par le sol (figure 4.1). Bien que la force en chaque point de cette surface ne contienne aucun couple, le changement de repère nécessaire à l'intégration de ces forces peut introduire des couples dans le vecteur de la force de réaction résultante (figure 4.2).

La force de réaction du sol est généralement évaluée au centre de pression, soit le point de la surface de contact où le vecteur de force de réaction ne comporte aucun couple selon les axes parallèles au sol. Le seul couple pouvant y avoir une valeur non nulle agit autour de l'axe perpendiculaire au sol. Cependant, dans nos équations, il sera souvent plus simple d'exprimer cette force par rapport au repère inertiel (figure 4.3).



FIGURE 4.1 – Le vecteur de force de réaction du sol correspond à l'intégrale des forces appliquées par le sol à chaque point de la surface de contact.

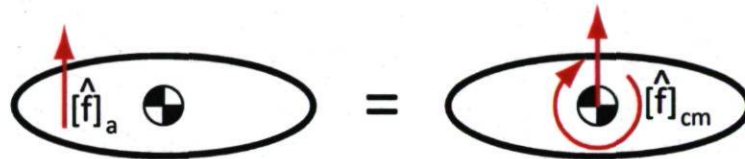


FIGURE 4.2 – Une force appliquée au point \vec{a} affecte le corps rigide de façon identique à une force et un couple appliqués au centre de masse. Cela correspond au changement de repère de la force f , du point \vec{a} au centre de masse du corps rigide.

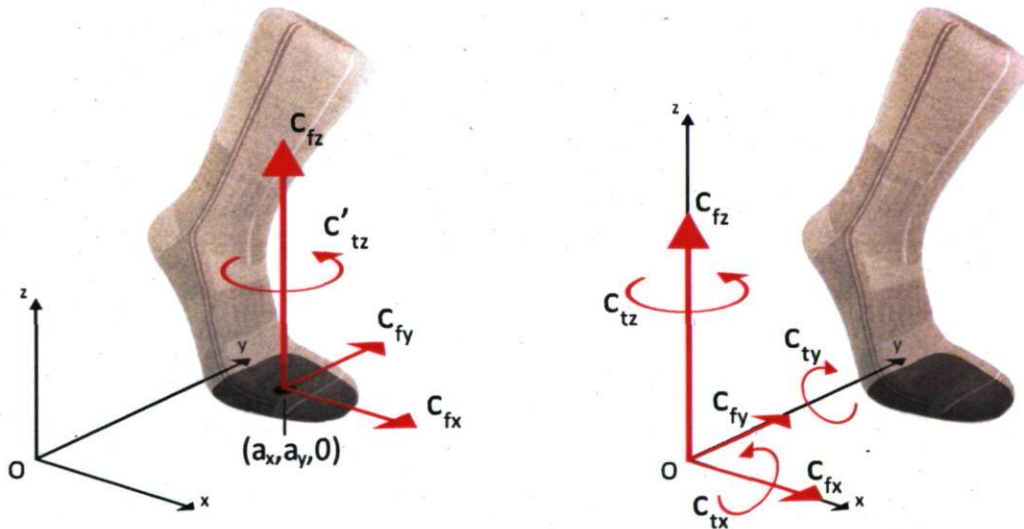


FIGURE 4.3 – (à gauche) Vecteur spatial de force de réaction du sol exprimé par rapport au centre de pression \vec{a} , où les couples en x et en y sont nuls par définition du centre de pression. (à droite) Le même vecteur, exprimé par rapport au repère inertiel, comporte généralement des couples non nuls autour des trois axes.

4.1 Problématique

Sous notre hypothèse de mouvement bipède, nous considérons qu'il n'y a que deux forces de réaction du sol possibles, la force sous le pied droit \hat{c}_d et celle sous le pied gauche \hat{c}_g . La somme de ces deux forces nous donne \hat{c}

$$\hat{c} = [\hat{c}_d]_0 + [\hat{c}_g]_0. \quad (4.1)$$

Comme nous le verrons dans la prochaine section, les équations de dynamique inverse nous permettent de connaître \hat{c} à partir du modèle biomécanique du personnage et de sa cinématique. Le problème que nous allons traiter est donc celui de trouver la valeur des vecteurs \hat{c}_d et \hat{c}_g en connaissant \hat{c} .

Les indices d et g sont utilisés tout au long de ce chapitre. Ils réfèrent aux indices des corps rigides du pied droit et du pied gauche respectivement. À titre d'exemple, \hat{I}_d est le tenseur d'inertie spatiale du pied droit exprimé dans son repère local et, de la même façon, ${}^g\hat{X}_0^*$ est la matrice de transformation spatiale utilisée pour exprimer dans le repère local du pied gauche une force exprimée par rapport au repère inertiel.

4.1.1 Somme des forces de réaction du sol

Rappelons que pour une accélération spatiale \hat{a}_i , une vitesse spatiale \hat{v}_i et une inertie spatiale \hat{I}_i , le vecteur de force spatiale à l'articulation i est donné par l'équation récursive suivante :

$$\hat{f}_i = \hat{I}_i \hat{a}_i + \hat{v}_i \hat{\times}^* \hat{I}_i \hat{v}_i - \hat{f}_{exti} + \sum_{j \in \phi(i)} {}^i\hat{X}_j^* \hat{f}_j. \quad (4.2)$$

où \hat{f}_{exti} est la somme des forces externes agissant sur le corps rigide i . Nous pouvons réécrire l'équation 4.2 en isolant les forces externes :

$$\hat{f}_i = \hat{h}_i - \sum_{j \in \phi(i)} {}^i\hat{X}_j^* \hat{f}_{extj}, \quad (4.3)$$

où \hat{h}_i est un vecteur fictif de forces en absence de forces externes

$$\hat{h}_i = \hat{I}_i \hat{a}_i + \hat{v}_i \hat{\times}^* \hat{I}_i \hat{v}_i + \sum_{j \in \phi(i)} {}^i\hat{X}_j^* \hat{h}_j. \quad (4.4)$$

Dans notre environnement de synthèse, les forces de réaction du sol sont les seules

forces externes agissant sur le personnage¹. La somme des forces de réaction du sol \hat{c} est donc égale à la somme des forces externes :

$$\hat{c} = \sum_i {}^0\hat{X}_i^* \hat{f}_{ext_i}, \quad (4.5)$$

De plus, nous savons que le personnage ne peut appliquer de force selon les 6 DDLs de sa racine puisque ce ne sont pas de véritables articulations, mais plutôt des DDLs fictifs qui relient le personnage au repère inertiel. Les vecteurs de forces doivent donc être nuls pour les 6 premières articulations du personnage

$$\hat{f}_i = \hat{0}, i \in \{1, 2, 3, 4, 5, 6\}.$$

Cela peut être simplifié par

$$\hat{f}_0 = \hat{0} \quad (4.6)$$

en se rappelant que les 6 articulations de la racine sont attachées à des corps rigides fictifs ayant une matrice d'inertie nulle et auxquels on ne peut appliquer directement une force externe. Les vecteurs de force spatiale de ces 6 articulations sont donc toujours identiques, à une transformation spatiale près. En relation à l'équation 4.3, nous devons donc avoir

$$\hat{f}_0 = \hat{h}_0 - \hat{c} = 0,$$

ce qui nous donne finalement la valeur du vecteur \hat{c} (évalué au repère inertiel) :

$$\hat{c} = \hat{h}_0. \quad (4.7)$$

4.2 Approche d'optimisation imbriquée

Soit le problème de fixer \hat{c}_d et \hat{c}_g dans l'équation 4.1. En support simple, le problème est trivial. Une des deux forces de contact doit être nulle et, par conséquent, l'autre force est égale à \hat{c} .

En support double toutefois, il existe une infinité de solutions à l'équation 4.1. Cependant, bien qu'une infinité de paires de forces donnent la même somme \hat{c} (figure 4.4), chacune de ces paires est le résultat d'un effort interne différent. Nous pouvons nous appuyer sur cette connaissance pour poser un critère qui nous permettra de comparer ces solutions et, éventuellement, d'en désigner une qui est préférable à toutes les autres.

1. Au lieu de traiter la gravité comme une force externe, nous soustrayons l'accélération gravitationnelle à l'accélération effective de la racine.



FIGURE 4.4 – Trois paires de forces différentes dont la somme est égale. Bien qu'elles soient issues de différents efforts musculaires du personnage, leur effet net sur la variation de la quantité de mouvement et du moment angulaire du personnage est identique.

Sans supposer que les mouvements humains naturels sont strictement optimaux d'un point de vue énergétique, nous posons l'hypothèse qu'un humain normalement constitué adoptera une démarche ayant une certaine efficacité énergétique. Cette hypothèse nous permet de trouver rapidement une solution qui s'approche beaucoup des forces mesurées sur un sujet humain, tel qu'illustré par les résultats présentés dans la section 4.3.

Afin d'alléger les équations tout au long de ce chapitre, nous remplaçons le terme \hat{c}_d par le terme \hat{x} .

$$\hat{x} = \begin{pmatrix} x_{\tau x} \\ x_{\tau y} \\ x_{\tau z} \\ x_{f x} \\ x_{f y} \\ x_{f z} \end{pmatrix} = [\hat{c}_d]_0.$$

Nous nous intéressons uniquement à la force sous le pied droit puisque la force sous le pied gauche peut être déduite par $[\hat{c}_g]_0 = \hat{c} - \hat{x}$.

L'hypothèse décrite plus tôt quant à l'efficacité énergétique du mouvement nous amène à chercher une force de réaction \hat{x} minimisant les efforts articulaires du personnage. Plus précisément, nous cherchons la valeur de \hat{x} minimisant la somme des efforts articulaires au carré :

$$F(\hat{x}) = \sum_{i \in \Omega} \left(S_i^t \hat{f}_i \right)^t \left(S_i^t \hat{f}_i \right) = \sum_{i \in \Omega} \hat{f}_i^t S_i \hat{f}_i, \quad (4.8)$$

où $S_i = S_i S_i^t$ et où S_i est la base du sous-espace de mouvement de l'articulation i , tel

que décrit dans la section 3.3.3. L'équation 4.8 exprime la somme des carrés des forces généralisées en coordonnées spatiales.

En plus de minimiser la fonction de coût F , la force de réaction \hat{x} doit être telle que les centres de pression de \hat{x} et $(\hat{c} - \hat{x})$ se trouvent dans les limites de la surface de contact du pied droit et gauche respectivement. Nous faisons alors face à un problème d'optimisation avec contraintes. Dans le cas général, il n'existe pas de solution analytique à ce problème et nous devons le résoudre par optimisation numérique.

Tous les problèmes d'optimisation ne sont pas égaux. À titre d'exemple, les algorithmes d'optimisation traitant des problèmes de programmation linéaire ou de programmation quadratique sont généralement beaucoup plus performants que les algorithmes d'optimisation d'ordre supérieur. Afin de profiter de cette performance accrue, nous allons formuler notre problème sous une forme proche de la programmation quadratique.

La programmation quadratique consiste à optimiser une fonction de coût quadratique soumise à des contraintes linéaires. Par des manipulations algébriques, nous pouvons formuler notre fonction de coût comme une fonction quadratique. Nos contraintes ne sont cependant pas linéaires.

Comme nous le verrons dans les prochaines sections, fixer le paramètre x_{fz} à une valeur z nous donne un sous-problème de programmation quadratique, ayant une fonction de coût quadratique et des contraintes linéaires. Nous proposons d'utiliser ce fait dans une approche d'optimisation imbriquée. La boucle interne de l'algorithme prend une valeur fixe pour x_{fz} et trouve l'optimum de ce problème de programmation quadratique. La boucle externe considère la boucle interne comme une fonction prenant une valeur z en entrée et retournant l'évaluation de la fonction de coût pour le vecteur \hat{x} optimal sous la contrainte $x_{fz} = z$. La boucle externe applique un algorithme de *recherche linéaire* à cette fonction par des appels successifs à la boucle interne pour trouver z^* , la valeur optimale de z , et éventuellement \hat{x}^* , la valeur optimale de \hat{x} .

4.2.1 Fonction de coût quadratique

En considérant que les seules forces externes agissant sur le système multicorps sont les deux forces de contact, et que nous additionnons l'accélération gravitationnelle à l'accélération de la racine du système multicorps, nous pouvons réécrire l'équation 4.2

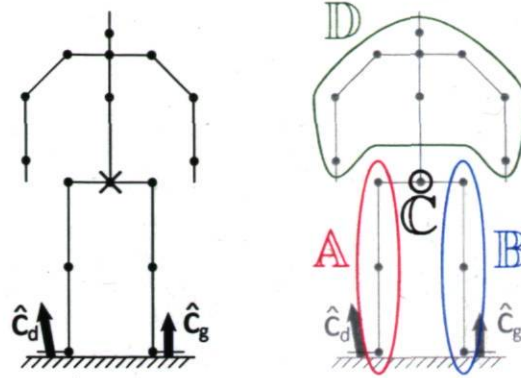


FIGURE 4.5 – Illustration du partitionnement des différentes articulations d'un modèle biomécanique. La racine du modèle est marquée d'un \times dans l'image de gauche. L'ensemble \mathbb{C} contient les articulations qui sont à la fois ancêtres du pied droit et du pied gauche, les ensembles \mathbb{A} et \mathbb{B} contiennent les articulations qui sont uniquement ancêtres du pied droit et du pied gauche respectivement et l'ensemble \mathbb{D} contient toutes les autres articulations du modèle biomécanique.

de la façon suivante :

$$\hat{f}_i = \begin{cases} \hat{I}_i \hat{a}_i + \hat{v}_i \hat{x}^* \hat{I}_i \hat{v}_i - {}^i \hat{X}_G^* \hat{x} + \sum_{j \in \phi(i)} {}^i \hat{X}_j^* \hat{f}_j & , \text{ si } i = d \\ \hat{I}_i \hat{a}_i + \hat{v}_i \hat{x}^* \hat{I}_i \hat{v}_i - {}^i \hat{X}_G^* (\hat{c} - \hat{x}) + \sum_{j \in \phi(i)} {}^i \hat{X}_j^* \hat{f}_j & , \text{ si } i = g \\ \hat{I}_i \hat{a}_i + \hat{v}_i \hat{x}^* \hat{I}_i \hat{v}_i + \sum_{j \in \phi(i)} {}^i \hat{X}_j^* \hat{f}_j & , \text{ sinon} \end{cases} \quad (4.9)$$

où d et g sont les indices du pied droit et du pied gauche respectivement.

Introduisons les ensembles \mathbb{A} , \mathbb{B} , \mathbb{C} et \mathbb{D} :

$$\mathbb{C} = \varpi(d) \cap \varpi(g) \quad (4.10)$$

$$\mathbb{A} = \varpi(d) \setminus \mathbb{C} \quad (4.11)$$

$$\mathbb{B} = \varpi(g) \setminus \mathbb{C} \quad (4.12)$$

$$\mathbb{D} = \Omega \setminus (\mathbb{A} \cup \mathbb{B} \cup \mathbb{C}), \quad (4.13)$$

où Ω est l'ensemble de tous les indices des articulations du système multicorps. La figure 4.5 illustre ce à quoi correspond chacun de ces ensembles. Ces ensembles forment une partition de l'ensemble Ω puisqu'ils sont mutuellement exclusifs et qu'ils couvrent la totalité des articulations du modèle.

Ce partitionnement nous permet d'évacuer \hat{x} de la portion récursive de l'équation

4.9 :

$$\hat{f}_i = \begin{cases} \hat{h}_i - {}^i\hat{X}_0^* \hat{x}, & i \in \mathbb{A} \\ \hat{h}_i - {}^i\hat{X}_0^* (\hat{c} - \hat{x}), & i \in \mathbb{B} \\ \hat{h}_i - {}^i\hat{X}_0^* \hat{c}, & i \in \mathbb{C} \\ \hat{h}_i, & i \in \mathbb{D} \end{cases} \quad (4.14)$$

ce qui facilite grandement sa manipulation.

À l'aide du développement présenté dans la section A.1, nous pouvons reformuler la fonction de coût de l'équation 4.8 sous la forme quadratique suivante :

$$F(\hat{x}) = \hat{x}^t \hat{H} \hat{x} + \hat{m}^t \hat{x} + b \quad (4.15)$$

où \hat{H} , \hat{m} et b sont respectivement donnés par :

$$\hat{H} = \sum_{i \in \mathbb{A} \cup \mathbb{B}} {}^i\hat{X}_0^{*t} \mathcal{S}_i {}^i\hat{X}_0^* \quad (4.16)$$

$$\hat{m}^t = 2 \sum_{i \in \mathbb{B}} \left(\hat{h}_i^t \mathcal{S}_i {}^i\hat{X}_0^* - \hat{c}^t {}^i\hat{X}_0^{*t} \mathcal{S}_i {}^i\hat{X}_0^* \right) - 2 \sum_{i \in \mathbb{A}} \hat{h}_i^t \mathcal{S}_i {}^i\hat{X}_0^* \quad (4.17)$$

$$b = \sum_{i \in \mathbb{B} \cup \mathbb{C}} \left(\hat{c}^t {}^i\hat{X}_0^{*t} \mathcal{S}_i {}^i\hat{X}_0^* \hat{c} - 2 \hat{h}_i^t \mathcal{S}_i {}^i\hat{X}_0^* \hat{c} \right) + \sum_{i \in \Omega} \hat{h}_i^t \mathcal{S}_i \hat{h}_i. \quad (4.18)$$

À titre informatif, la dérivée de cette fonction de coût est donnée par :

$$\frac{dF(\hat{x})}{d\hat{x}} = 2\hat{H}\hat{x} + \hat{m}. \quad (4.19)$$

4.2.2 Contraintes linéaires

Dans cette section, nous développons une forme linéaire pour les contraintes des centres de pression pour une valeur de x_{fz} fixe. Rappelons que ces contraintes stipulent que le centre de pression d'une force de réaction du sol doit se trouver à l'intérieur de la surface de contact entre le pied et le sol.

Nous estimons la forme d'un pied par un prisme rectangulaire et le sol par un plan. Afin de représenter la déformation du pied qui entre en contact avec le sol, nous permettons une pénétration de 1 cm du pied dans le sol. La surface de contact nous est donc donnée par l'intersection du plan et du prisme rectangulaire. Comme l'illustre la figure 4.6, cette intersection peut prendre plusieurs formes, de la surface nulle jusqu'à un hexagone.

Un prisme rectangulaire est une approximation grossière de la forme réelle d'un pied, mais cela nous permet d'obtenir très rapidement une estimation raisonnable des

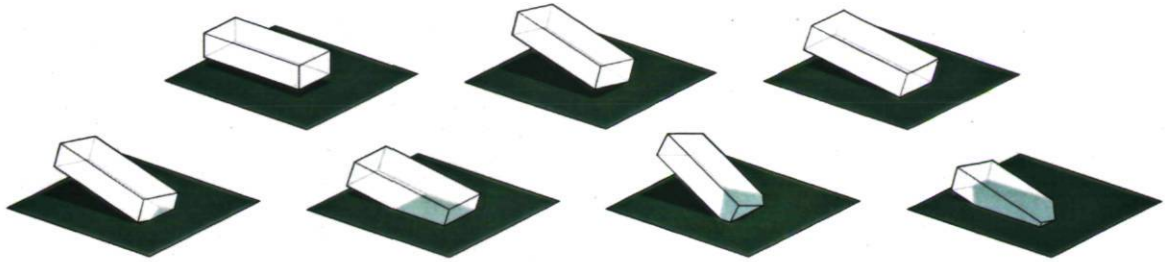


FIGURE 4.6 – Différentes surfaces de contact décrites par l'intersection d'un prisme rectangulaire et d'un plan. De gauche à droite et de haut en bas, on présente la surface nulle, un point, un segment de droite, un triangle, un quadrilatère, un pentagone et un hexagone.

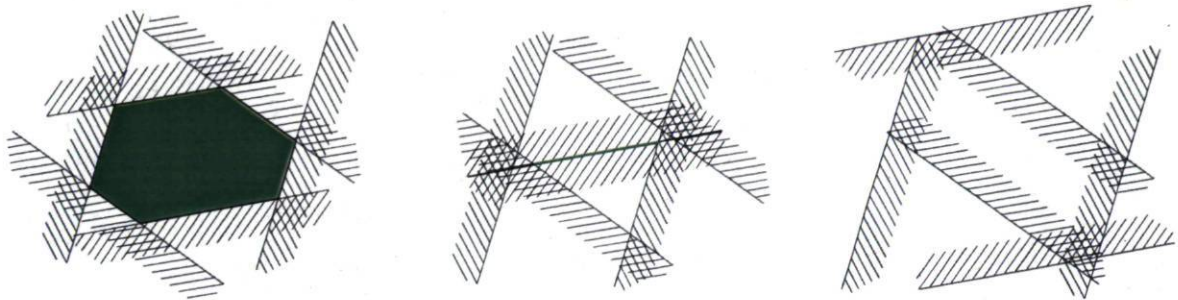


FIGURE 4.7 – L'intersection de 6 demi-plans détermine la surface à l'intérieur de laquelle un centre de pression peut se trouver. La variation des positions relatives de ces demi-plans peut engendrer tous les types de surfaces présentées dans la figure 4.6. Un hexagone, un segment de droite et la surface nulle sont présentés ici en exemples.

forces de réactions du sol. Notons que nous pourrions développer cette même technique en représentant un pied par un polyèdre convexe quelconque. Nous aurions alors une contrainte par face du polyèdre pour chaque pied. Toutefois, pour utiliser un polyèdre concave ou une forme non polyédrique, nous devrions travailler avec des contraintes non linéaires, ce qui ajouterait une certaine complexité au problème d'optimisation.

On peut voir un prisme rectangulaire comme l'intersection de six demi-espaces, où chaque face du prisme est la frontière d'un demi-espace. Lorsqu'ils sont évalués sur un plan, ces demi-espaces produisent des demi-plans. Donc, la surface de contact peut être vue comme l'intersection de six demi-plans (figure 4.7). Notre approche consiste à s'assurer que les centres de pression sont positionnés du bon côté de la frontière de chacun de ces demi-plans.

Afin de développer les équations qui suivent, il faut définir quelques objets mathématiques supplémentaires, notamment les vecteurs \vec{u} , \vec{v} et $\vec{\delta}$, les repères α et β et les scalaires $\alpha_{i,j}$ et $\beta_{i,j}$. Les vecteurs $\vec{u} = (u_x, u_y, 0)^t$ et $\vec{v} = (v_x, v_y, 0)^t$ sont les centres de

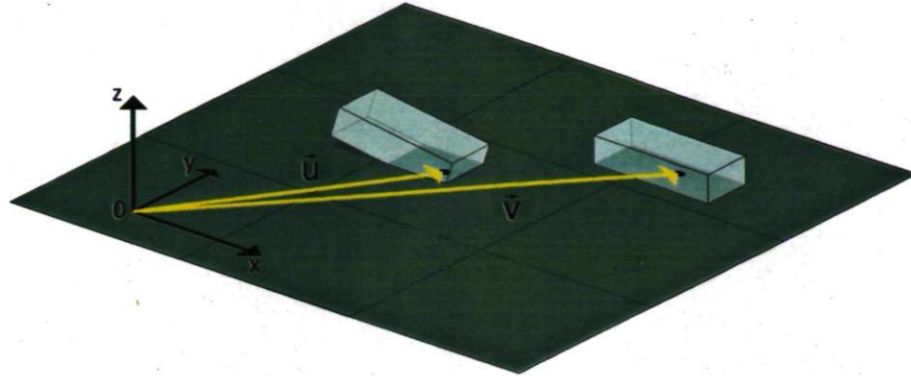


FIGURE 4.8 – Les vecteurs \vec{u} et \vec{v} représentent la position du centre de pression du pied droit et du pied gauche respectivement.

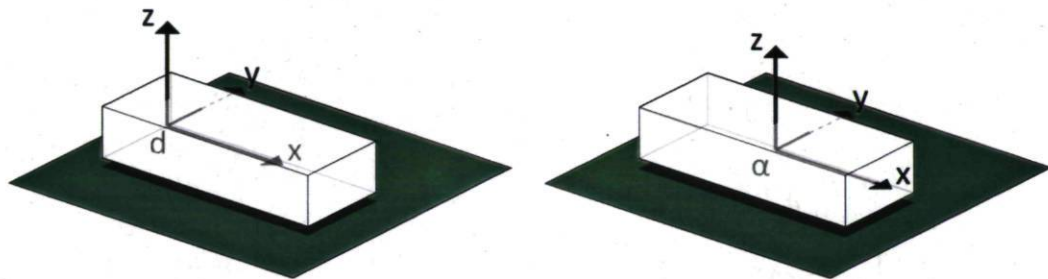


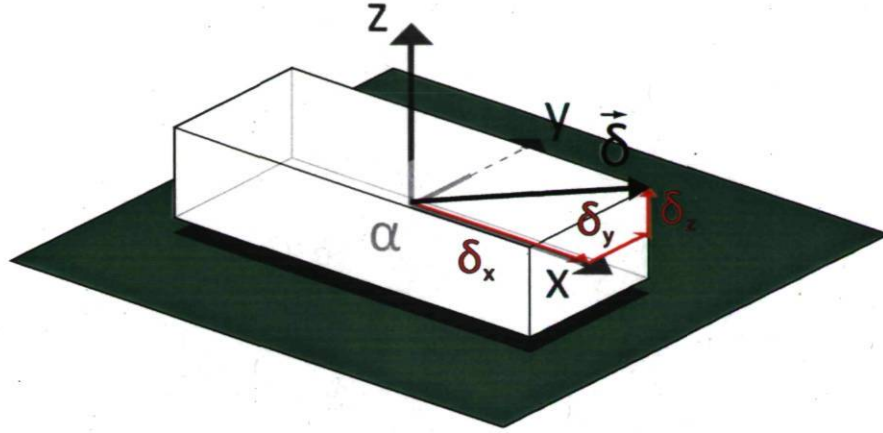
FIGURE 4.9 – Illustration des repères d et α sur le pied droit. Les deux repères sont de même orientation, mais ont une origine distincte. L'origine du repère α est au centre géométrique du corps rigide. L'origine du repère d est centrée sur l'articulation de la cheville. On peut remarquer que la cheville ne se trouve pas à l'extrémité du corps rigide afin de représenter le talon. Pour le pied gauche, le repère d est remplacé par le repère g et le repère α est remplacé par le repère β .

pression du pied droit et du pied gauche respectivement, exprimés par rapport au repère inertiel² (figure 4.8).

Les repères α et β ont les mêmes orientations que les repères d et g respectivement (repères fixés aux chevilles droite et gauche), mais ont leur origine au centre géométrique de chaque pied (figure 4.9). Ils nous servent à délimiter plus facilement la surface de contact entre un pied et le sol.

Le vecteur $\vec{\delta}$ donne la longueur des demi-axes du prisme rectangulaire représentant chaque pied. Pour simplifier la notation, nous considérons que les deux pieds sont de même dimension, bien qu'il soit possible de définir un vecteur $\vec{\delta}$ distinct pour chaque

2. Sans perte de généralité, le sol est considéré être un plan normal à l'axe z du repère inertiel passant par $z = 0$.


 FIGURE 4.10 – Illustration du vecteur $\vec{\delta}$ et de ses composantes δ_x , δ_y et δ_z .

pied. Le vecteur $\vec{\delta}$ est exprimé par rapport aux repères α et β et donne la position du coin du prisme rectangulaire contenu dans l'octant positif de ces repères (figure 4.10). Ses composantes δ_x , δ_y et δ_z donnent respectivement la demi-longueur, la demi-largeur et la demi-hauteur du pied.

Finalement, les scalaires $\alpha_{i,j}$ et $\beta_{i,j}$ sont les éléments des matrices de transformation homogène ${}^{\alpha}\tilde{W}_0$ et ${}^{\beta}\tilde{W}_0$ ³, telles que définies par l'équation :

$${}^{\alpha}\tilde{W}_0 = \begin{pmatrix} {}^{\alpha}R_0 & {}^{\alpha}\vec{t}_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & \alpha_{1,3} & \alpha_{1,4} \\ \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} & \alpha_{2,4} \\ \alpha_{3,1} & \alpha_{3,2} & \alpha_{3,3} & \alpha_{3,4} \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$${}^{\beta}\tilde{W}_0 = \begin{pmatrix} {}^{\beta}R_0 & {}^{\beta}\vec{t}_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \beta_{1,1} & \beta_{1,2} & \beta_{1,3} & \beta_{1,4} \\ \beta_{2,1} & \beta_{2,2} & \beta_{2,3} & \beta_{2,4} \\ \beta_{3,1} & \beta_{3,2} & \beta_{3,3} & \beta_{3,4} \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Chaque demi-plan correspond à une contrainte d'inégalité linéaire sur un centre de pression. Par définition, les deux centres de pression sont au sol ($z=0$). En exprimant les centres de pression par rapport au centre géométrique de chaque pied, nous pouvons évaluer s'ils sont également contenus dans les prismes rectangulaires des pieds. Si c'est le cas, nous aurons les inégalités suivantes

$$-\vec{\delta} \leq {}^{\alpha}R_0\vec{u} + {}^{\alpha}\vec{t}_0 \leq \vec{\delta} \quad (4.20)$$

$$-\vec{\delta} \leq {}^{\beta}R_0\vec{v} + {}^{\beta}\vec{t}_0 \leq \vec{\delta} \quad (4.21)$$

3. Bien que nous n'utilisions pas directement ces matrices de transformation homogènes, référer à leurs éléments nous permet d'alléger la notation.

et les centres de pression seront bel et bien contenus à l'intérieur des surfaces de contact entre les pieds et le sol.

Or, les centres de pression sous les deux pieds sont fonction de \hat{x} . Par définition, les centres de pressions correspondent aux points où l'expression de la force de contact ne comporte aucun couple selon les axes x et y . Ils sont obtenus par les équations suivantes :

$$u_x = -\frac{x_{\tau y}}{x_{fz}} \quad (4.22)$$

$$u_y = \frac{x_{\tau x}}{x_{fz}} \quad (4.23)$$

$$v_x = -\frac{c_{\tau y} - x_{\tau y}}{c_{fz} - x_{fz}} \quad (4.24)$$

$$v_y = \frac{c_{\tau x} - x_{\tau x}}{c_{fz} - x_{fz}} \quad (4.25)$$

En suivant le développement présenté dans la section A.2, nous pouvons reformuler ces contraintes en fonction de \hat{x} :

$$A\hat{x} \leq d(z) \quad (4.26)$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \hat{x} = z \quad (4.27)$$

où A et d sont donnés par :

$$A = \begin{pmatrix} \alpha_{1,2} & -\alpha_{1,1} & 0 & 0 & 0 & 0 \\ \alpha_{2,2} & -\alpha_{2,1} & 0 & 0 & 0 & 0 \\ \alpha_{3,2} & -\alpha_{3,1} & 0 & 0 & 0 & 0 \\ -\alpha_{1,2} & \alpha_{1,1} & 0 & 0 & 0 & 0 \\ -\alpha_{2,2} & \alpha_{2,1} & 0 & 0 & 0 & 0 \\ -\alpha_{3,2} & \alpha_{3,1} & 0 & 0 & 0 & 0 \\ -\beta_{1,2} & \beta_{1,1} & 0 & 0 & 0 & 0 \\ -\beta_{2,2} & \beta_{2,1} & 0 & 0 & 0 & 0 \\ -\beta_{3,2} & \beta_{3,1} & 0 & 0 & 0 & 0 \\ \beta_{1,2} & -\beta_{1,1} & 0 & 0 & 0 & 0 \\ \beta_{2,2} & -\beta_{2,1} & 0 & 0 & 0 & 0 \\ \beta_{3,2} & -\beta_{3,1} & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (4.28)$$

$$d(z) = \begin{pmatrix} z \begin{pmatrix} \vec{\delta} - \alpha \vec{t}_0 \\ \vec{\delta} + \alpha \vec{t}_0 \end{pmatrix} \\ (c_{fz} - z) \begin{pmatrix} \vec{\delta} - \beta \vec{t}_0 \\ \vec{\delta} + \beta \vec{t}_0 \end{pmatrix} + \begin{pmatrix} \beta R_0 \begin{pmatrix} c_{\tau y} \\ -c_{\tau x} \\ 0 \end{pmatrix} \\ \beta R_0 \begin{pmatrix} -c_{\tau y} \\ c_{\tau x} \\ 0 \end{pmatrix} \end{pmatrix} \end{pmatrix}, \quad (4.29)$$

et où z est la valeur à laquelle nous fixons x_{fz} pour ce sous-problème.

4.2.3 Boucle externe

Les sections précédentes nous permettent de définir la boucle interne de notre optimisation imbriquée, que nous représentons sous la forme d'une fonction $G(z)$ dont l'évaluation nécessite la résolution d'un problème de programmation quadratique

$$G(z) = \min_{\hat{x} \in \mathbb{R}^6} F(\hat{x}), \quad \text{sous } \begin{cases} A\hat{x} \leq d(z) \\ x_{fz} = z \end{cases}. \quad (4.30)$$

La boucle externe vise à trouver la valeur z^* minimisant $G(z)$ et, éventuellement, de trouver la valeur du vecteur \hat{x}^* associé à cet optimum $G(z^*)$. Puisqu'il s'agit d'un problème de programmation quadratique, résoudre l'équation 4.30 est relativement simple. En particulier, nous connaissons la forme analytique de tous les éléments de cette équation et nous pouvons facilement obtenir leurs dérivées analytiques.

Trouver le minimum de $G(z)$, et par conséquent la valeur de z^* , correspond toutefois à un problème d'optimisation moins bien défini. Les contraintes introduisent des discontinuités dans cette fonction, qui font de $G(z)$ une fonction non linéaire et qui rendent l'expression d'une dérivée plus difficile. Ces dérivées doivent donc être estimées par différences finies par l'algorithme d'optimisation.

Dans les faits cela ne pose pas un problème majeur, puisqu'il s'agit d'un problème d'optimisation dans un espace à une seule dimension et que les algorithmes traitant ce type de problèmes, les algorithmes de recherche linéaire, peuvent être très performants. De plus, puisque le sol ne peut que pousser le personnage et non le tirer, nous savons que les valeurs de x_{fz} et $(c_{fz} - x_{fz})$ doivent être positives. Ces valeurs ne peuvent être nulles puisque nous serions alors en présence d'un problème de support simple, ce qui se résout de façon analytique. La valeur de z^* est donc comprise dans l'intervalle $]0, c_{fz}[$ et est obtenue en résolvant le problème d'optimisation suivant

$$z^* = \arg \min_{z \in]0, c_{fz}[} G(z). \quad (4.31)$$

Ce problème d'optimisation constitue la boucle externe de notre optimisation imbriquée.

Une fois la valeur de z^* connue, le vecteur de force optimal \hat{x}^* peut être obtenu par l'optimisation suivante :

$$\hat{x}^* = \arg \min_{\hat{x} \in \mathbb{R}^6} F(\hat{x}), \quad \text{sous } \begin{cases} A\hat{x} \leq d(z^*) \\ x_{fz} = z^* \end{cases}. \quad (4.32)$$

Cette valeur peut toutefois être retenue en mémoire lors de la résolution de la boucle externe afin d'éviter de devoir résoudre ce dernier problème d'optimisation.

4.3 Résultats

Afin de tester notre approche d'estimation des forces de réaction du sol, nous l'appliquons à une séquence de marche issue de capture de mouvement et pour laquelle les forces de réaction du sol ont été mesurées à l'aide de plateformes de force. Les séquences de mouvement pour lesquelles nous connaissons à la fois la cinématique et les forces de réaction du sol sont très difficiles à obtenir et nous ne disposons pas de l'équipement nécessaire pour en produire. Nous n'avons réussi à obtenir qu'une seule séquence de la sorte⁴. Notre approche utilise uniquement l'information cinématique de la séquence et

4. La séquence utilisée nous a gracieusement été fournie par monsieur Philippe Corbeil, professeur au département de médecine sociale et préventive de l'Université Laval.

la masse du sujet. L'information des plateformes de force ne sert qu'à comparer nos résultats aux données de forces réelles.

La séquence traitée est une séquence de marche exécutée par un garçon de 23kg où les forces de réaction du sol ont été mesurées par trois plateformes de force. Ces plateformes nous permettent de connaître les forces de réaction lors de deux phases de support double : lorsque le garçon a le pied gauche sur la première plateforme et le pied droit sur la deuxième, et lorsqu'il a le pied gauche sur la troisième plateforme et le pied droit sur la deuxième. L'acquisition de cette séquence s'est faite à un taux d'échantillonnage de 120Hz. La cinématique, donnant la valeur des diverses coordonnées articulaires du personnage, a été obtenue à partir de positions de marqueurs. De plus, les données de cinématique ont été filtrées pour réduire le bruit d'acquisition par un processus qui nous est inconnu. Les données de force n'ont pas été prétraitées.

Nous avons peu d'information sur le sujet ayant effectué le mouvement. Nous savons que sa masse est de 23kg et les marqueurs utilisés lors de la numérisation du mouvement nous ont permis d'estimer la longueur des différentes parties du corps. De plus, la table de Winter [87] nous a permis d'estimer la masse et la densité de chaque partie du corps⁵. Nous avons utilisé ces informations afin de construire un modèle biomécanique à 43 coordonnées articulaires (incluant les 6 articulations de la racine) représentant le garçon. Les corps rigides et les articulations de ce modèle sont tels qu'illustrés à la figure 4.11.

Plusieurs facteurs, tels que le bruit d'acquisition et la simplicité du modèle biomécanique, viennent limiter notre capacité à bien détecter le contact entre un pied et le sol. Puisque l'estimation des temps de contact d'une séquence de capture de mouvement ne fait pas partie de nos objectifs, nous avons utilisé l'information des plateformes de force pour connaître ces temps de contact.

Soulignons que la géométrie de notre modèle biomécanique produit une légère pénétration des pieds dans le sol lorsque le mouvement lui est appliqué. Ceci est notamment dû au fait que nous représentons un pied par un corps rigide, alors qu'en réalité c'est un corps déformable. Cette pénétration est toutefois nécessaire pour estimer la surface de contact entre le pied et le sol. Cela est donc conforme à nos attentes.

5. Bien que la table de Winter [87] ait été établie à partir de données mesurées sur des adultes, l'approximation qu'elle fait du sujet de notre séquence est suffisante pour nos besoins. Une estimation plus précise pourrait toutefois permettre d'améliorer la précision des résultats.

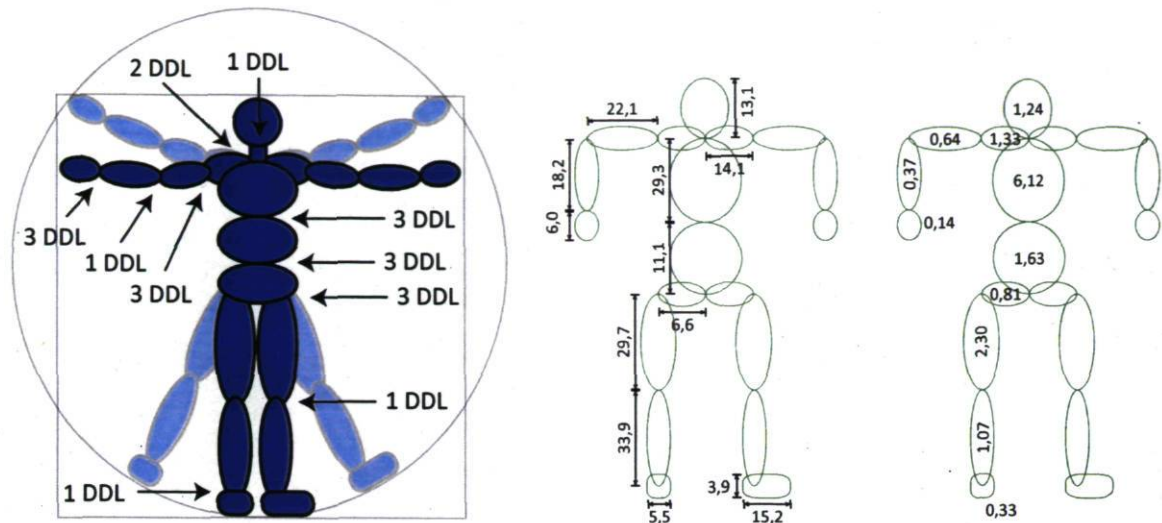


FIGURE 4.11 – Illustration du modèle biomécanique utilisé pour évaluer notre approche d'estimation des forces de réaction du sol. La figure montre les articulations (à gauche), la longueur des segments (au centre) et la masse des segments (à droite) du modèle en question.

4.3.1 Qualité des résultats

Les figures suivantes comparent les résultats obtenus par notre approche aux forces de réaction du sol mesurées par les plateformes de force lors de l'acquisition. Nous utilisons un fond vert pâle pour indiquer une période de support double et un fond jaune pâle pour indiquer une période de support simple, sur le pied droit comme sur le pied gauche. Les figures 4.12 et 4.13 comparent la position des centres de pression mesurés et estimés pour les deux pieds. Les figures 4.14 et 4.15 comparent la mesure et l'estimation des diverses composantes des forces de réaction du sol pour les deux pieds.

Nous nous intéressons plus particulièrement aux périodes de support double, mais les périodes de support simple nous montrent directement les effets du prétraitement des données de cinématique et de la modélisation simplifiée du sujet. En effet, en phase de support simple, la solution peut être connue analytiquement. Avec des données de mouvement idéales et un modèle biomécanique idéal, les forces estimées devraient être identiques aux forces mesurées par les plateformes de force, ce qui n'est visiblement pas le cas.

Malgré les erreurs induites par la modélisation, notre approche est tout de même en mesure de fournir des estimations adéquates des forces de contact. Le tableau 4.3.1 présente une compilation des erreurs des estimations obtenues par notre approche.

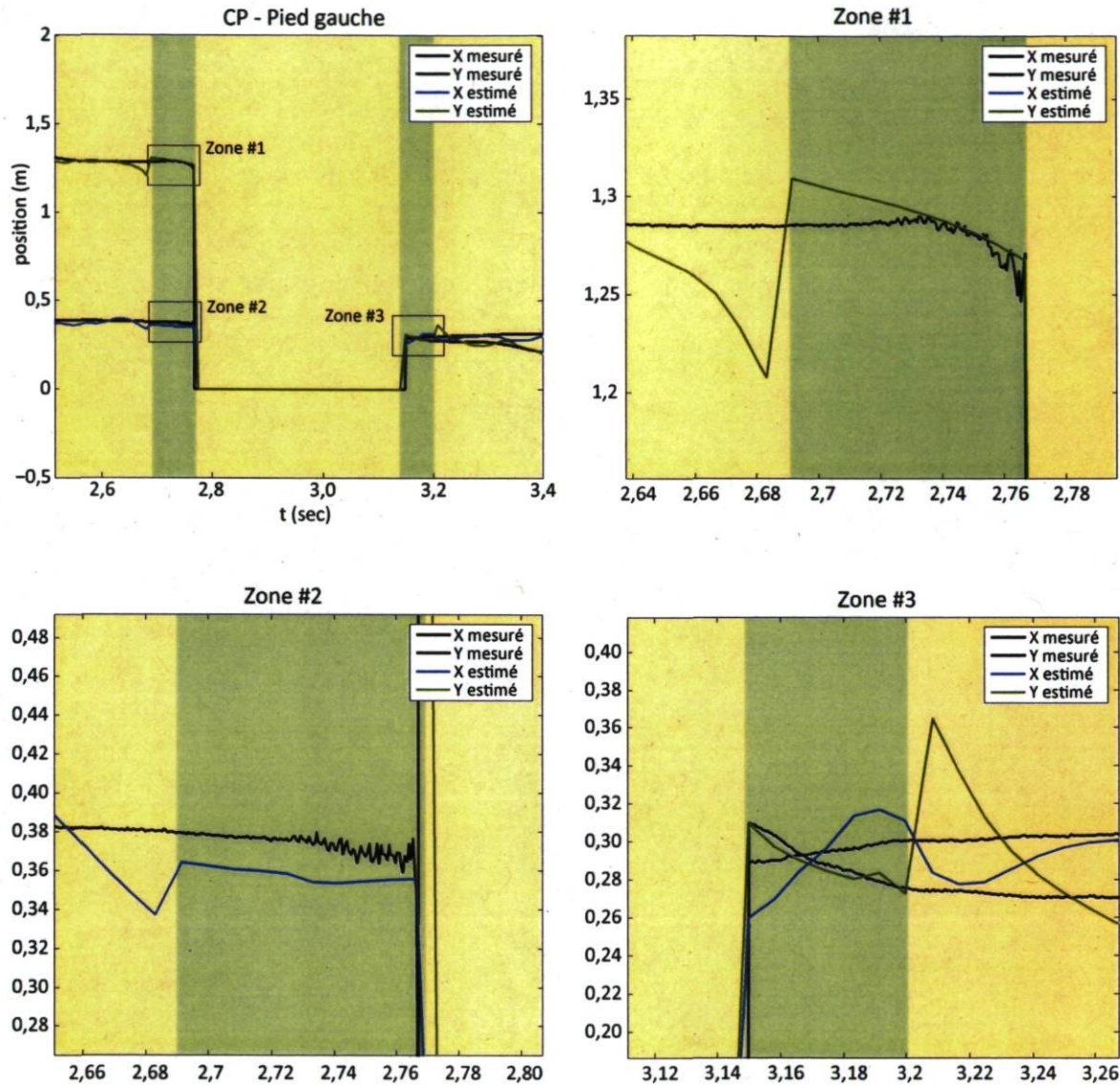


FIGURE 4.12 – Pied gauche - Comparaison entre la position du centre de pression mesurée par les plateformes de force et celle estimée par l'approche proposée. Trois zones d'intérêt sont grossies afin de mieux juger la qualité de l'estimation.

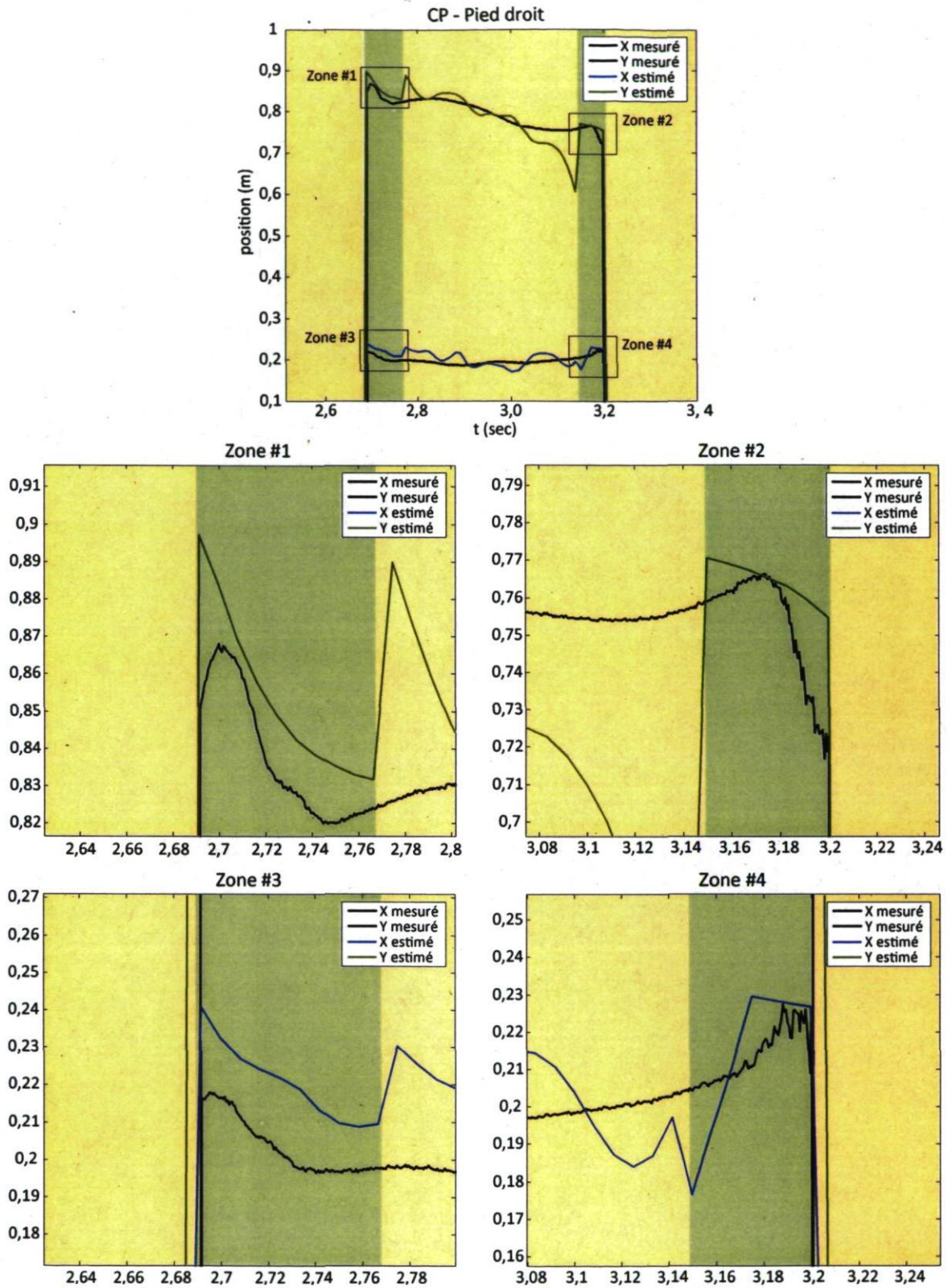


FIGURE 4.13 – Pied droit - Comparaison entre la position du centre de pression mesurée par les plateformes de force et celle estimée par l'approche proposée. Quatre zones d'intérêt sont grossies afin de mieux juger la qualité de l'estimation.

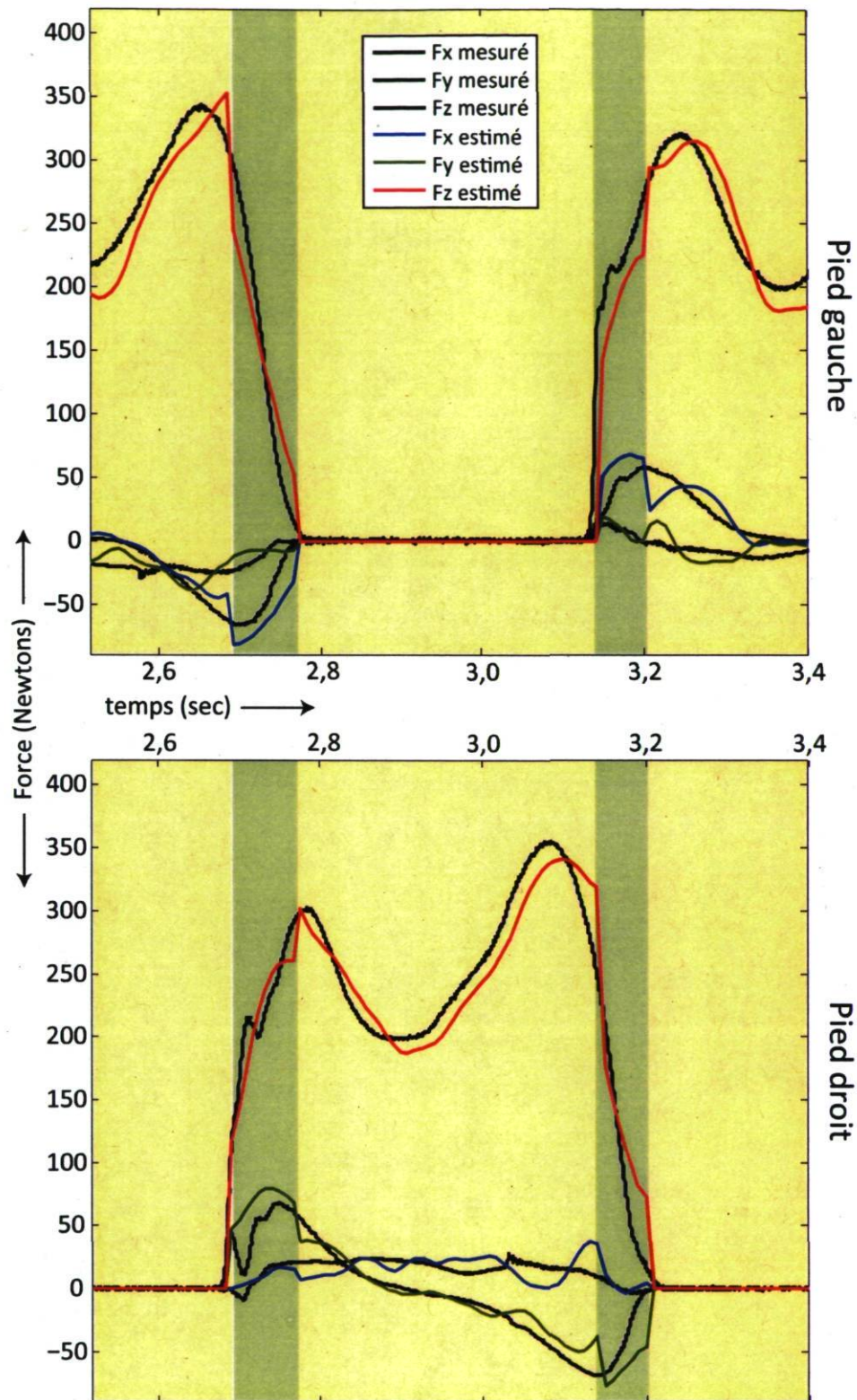


FIGURE 4.14 – Comparaison des forces de réaction du sol mesurées et estimées. Les forces sont évaluées au centre de pression.

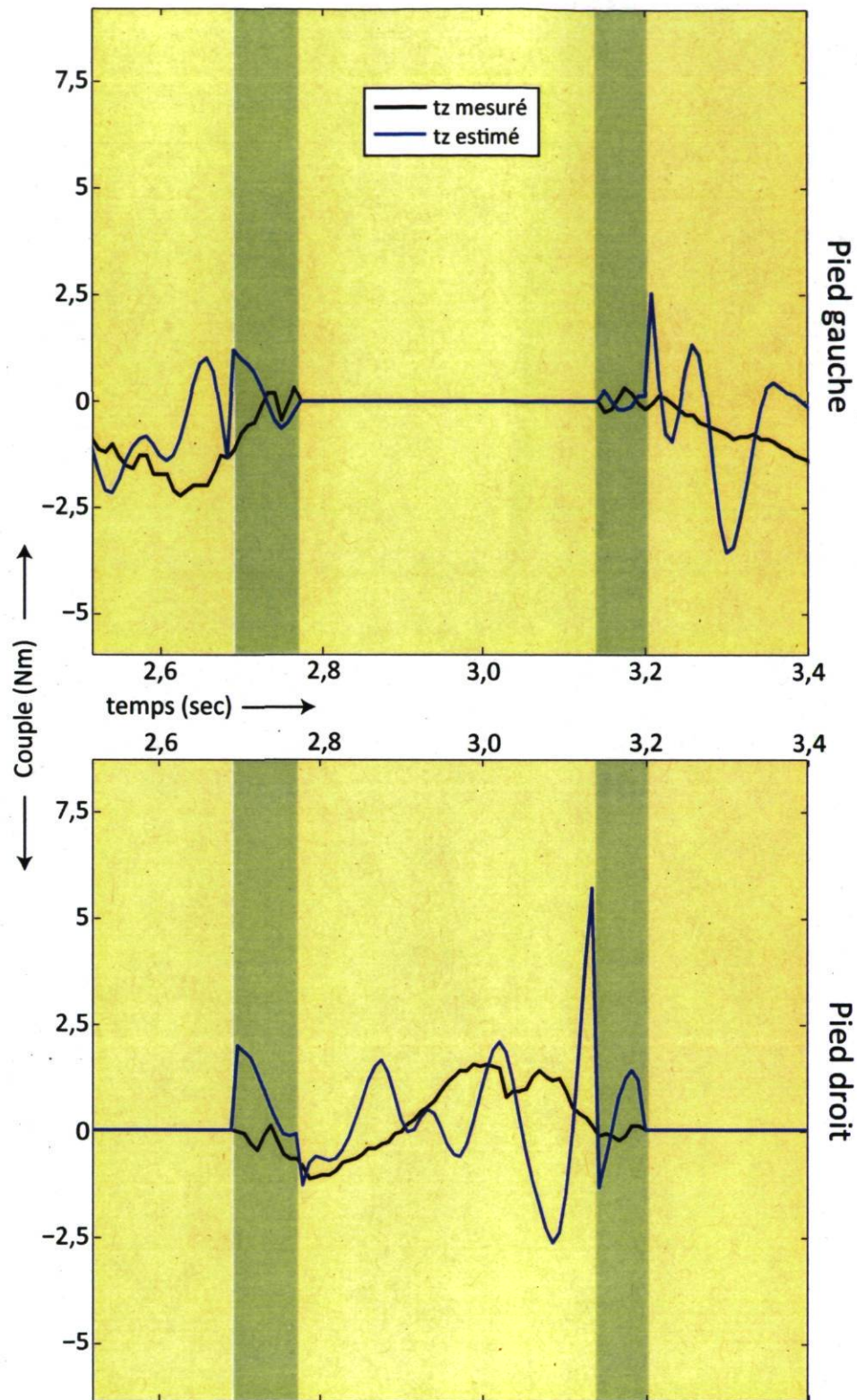


FIGURE 4.15 – Comparaison du couple autour de l'axe z des forces spatiales de réaction du sol mesurées et estimées. Les forces spatiales sont évaluées au centre de pression, les couples en x et y sont donc nuls.

Erreur	RMS	Max	RMS relative	Ren [70]
τ_x (Nm)	31,04	68,80	9%	12%
τ_y (Nm)	9,77	19,63	11%	33%
τ_z (Nm)	8,44	18,74	21%	26%
f_x (N)	5,90	11,42	15%	20%
f_y (N)	23,22	45,57	16%	11%
f_z (N)	33,14	55,73	14%	6%
cp (cm)	2,67	5,14	n/d	n/d

TABLE 4.1 – Erreurs RMS, maximum et RMS relative de notre approche d'estimation des forces de contact par rapport aux forces mesurées par des plateformes de force. La dernière colonne présente à titre comparatif l'erreur RMS relative obtenue par Ren et coll.[70].

Une erreur RMS relative est obtenue en normalisant l'erreur par l'amplitude du signal décrit par la variable en fonction du temps. Donc, pour une variable x et une estimation e , nous avons :

$$RMSE_{rel} = \frac{RMSE}{\frac{1}{2} \left[\left(\max_{0 \leq t \leq T} x_t - \min_{0 \leq t \leq T} x_t \right) + \left(\max_{0 \leq t \leq T} e_t - \min_{0 \leq t \leq T} e_t \right) \right]}. \quad (4.33)$$

Dans l'avant-dernière colonne du tableau 4.3.1, seules les trames de support double ont été utilisées pour déterminer l'amplitude d'un signal donné.

L'amplitude des erreurs présentées dans le tableau 4.3.1 est non négligeable. Cette précision est néanmoins suffisante pour nos besoins, soit d'obtenir une estimation initiale grossière des forces de réaction du sol. Une modélisation plus précise serait toutefois requise afin d'atteindre une précision suffisante pour des applications où un diagnostic doit être posé à partir de ces signaux, tel qu'en kinésiologie.

Le tableau 4.3.1 présente également une comparaison avec les résultats obtenus par Ren et coll.[70]. Leur approche consiste essentiellement à utiliser une fonction de transfert pour déterminer la répartition, entre les deux pieds, de la somme des forces de contact. Par rapport à la nôtre, cette approche profite d'une certaine simplicité, puisqu'elle offre une solution directe ne nécessitant aucun processus d'optimisation. Toutefois, notre approche à l'avantage de contraindre le centre de pression estimées à l'intérieur de la surface de contact, ce qui est essentielle pour qu'une force soit considérée valide dans notre contexte. De plus, la comparaison des erreurs RMS relatives nous porte à croire que notre approche d'estimation se positionne avantageusement par rapport aux approches existantes en sciences du mouvement, et ce malgré un modèle biomécanique très approximatif.

Discussion

Tel que mentionné plus tôt, les erreurs visibles dans nos figures en phase de support simple sont attribuables à une modélisation biomécanique approximative et possiblement à des erreurs induites dans le prétraitement des données. Ces mêmes facteurs affectent certainement la précision des forces de contacts estimées en phase de support double.

Outre ces facteurs, la fonction de coût utilisée peut aussi contribuer à l'écart entre les forces estimées et mesurées. Bien que la minimisation de l'effort interne du personnage semble être une hypothèse prometteuse, une étude sur un plus grand nombre de sujets pourrait certainement nous permettre de la raffiner.

Ce type d'étude et une modélisation biomécanique plus adéquate nous permettraient de comparer sur des bases plus solides l'approche proposée aux autres approches similaires développées en sciences du mouvement. Dans le cadre de nos recherches, le type de séquences requises pour une telle étude se sont toutefois avérés être une ressource très rare.

4.3.2 Performance

Afin de mieux évaluer la performance de notre approche, nous présentons ici trois formulations distinctes du problème d'estimation des forces de réaction du sol en support double. Il s'agit de trois formulations du même problème et toutes visent à minimiser l'effort interne du personnage. La figure 4.16 compare les temps de calculs requis pour converger avec chacune de ces trois formulations. Ces temps ont été établis sur 100 essais. Les dérivées analytiques des fonctions de coût et des fonctions de contraintes sont fournies aux algorithmes d'optimisation pour les trois formulations, à l'exception de la boucle externe de la formulation imbriquée dont la dérivée analytique est inconnue. Dans nos expérimentations, les trois formulations convergent toujours et elles donnent toutes un résultat identique au millième de Newton (Newton-mètre) près.

Optimisation non linéaire naïve La formulation *non linéaire naïve* consiste à optimiser la fonction de coût sous sa forme initiale (équation 4.8) sous les contraintes non linéaires (équations 4.20 et 4.21)

$$\min_{\hat{x} \in \mathbb{R}^6} \sum_{i \in \Omega} \hat{f}_i^t(\hat{x}) \mathcal{S}_i \hat{f}_i(\hat{x}), \quad \text{sous } \begin{cases} -\vec{\delta} \leq {}^\alpha R_0 \vec{u}(\hat{x}) + {}^\alpha \vec{t}_0 \leq \vec{\delta} \\ -\vec{\delta} \leq {}^\beta R_0 \vec{v}(\hat{x}) + {}^\beta \vec{t}_0 \leq \vec{\delta} \end{cases}.$$

Notre implémentation actuelle résout ce problème à l'aide de la fonction d'optimisation non linéaire *fmincon* de la librairie d'optimisation de Matlab[®].

Optimisation non linéaire/quadratique La formulation *non linéaire/quadratique* consiste à optimiser la fonction de coût sous sa forme quadratique (équation 4.15). Toutefois, comme la formulation naïve, elle utilise les contraintes non linéaires des équations 4.20 et 4.21.

$$\min_{\hat{x} \in \mathbb{R}^6} \hat{x}^t \hat{H} \hat{x} + \hat{m}^t \hat{x} + b, \quad \text{sous} \begin{cases} -\vec{\delta} \leq {}^\alpha R_0 \vec{u}(\hat{x}) + {}^\alpha \vec{t}_0 \leq \vec{\delta} \\ -\vec{\delta} \leq {}^\beta R_0 \vec{v}(\hat{x}) + {}^\beta \vec{t}_0 \leq \vec{\delta} \end{cases}.$$

Étant donné les contraintes non linéaires, la fonction *fmincon* est utilisée pour résoudre ce problème.

Optimisation imbriquée La formulation *imbriquée* est la formulation développée tout au long de ce chapitre, basée sur les équations 4.30 et 4.31. Dans notre implémentation actuelle, l'optimisation est effectuée à l'aide des fonctions *quadprog* et *fminbnd* de la librairie d'optimisation de Matlab[®]. La fonction *quadprog* est appelée pour résoudre le problème de programmation quadratique (QP) de la boucle interne (équation 4.30)

$$G(z) = \min_{\hat{x} \in \mathbb{R}^6} \hat{x}^t \hat{H} \hat{x} + \hat{m}^t \hat{x} + b, \quad \text{sous} \begin{cases} A\hat{x} \leq d(z) \\ x_{fz} = z \end{cases},$$

alors que la fonction *fminbnd* est utilisée pour résoudre la recherche linéaire bornée de la boucle externe (équation 4.31)

$$z^* = \arg \min_{z \in]0, c_{fz}[} G(z).$$

Avec notre implémentation Matlab[®] actuelle, nous pouvons obtenir une vitesse de traitement d'un peu plus de 20fps par *thread* sur un processeur Intel Core i7[®] 920⁶, soit un temps de moins de 50 millisecondes pour résoudre l'équation 4.31. Ce temps d'exécution est dominé largement par la résolution des sous-problèmes de QP, qui se résolvent en un temps de l'ordre de 5 millisecondes et qui sont appelés en moyenne de neuf à dix fois par la boucle externe.

6. Notre implémentation actuelle ne profite pas du parallélisme de ce type de processeur. Toutefois puisque chaque trame d'une séquence peut être traitée indépendamment, il est envisageable d'en produire une implémentation profitant des 8 *threads* de ce processeur pour obtenir une vitesse de traitement d'environ 160fps.

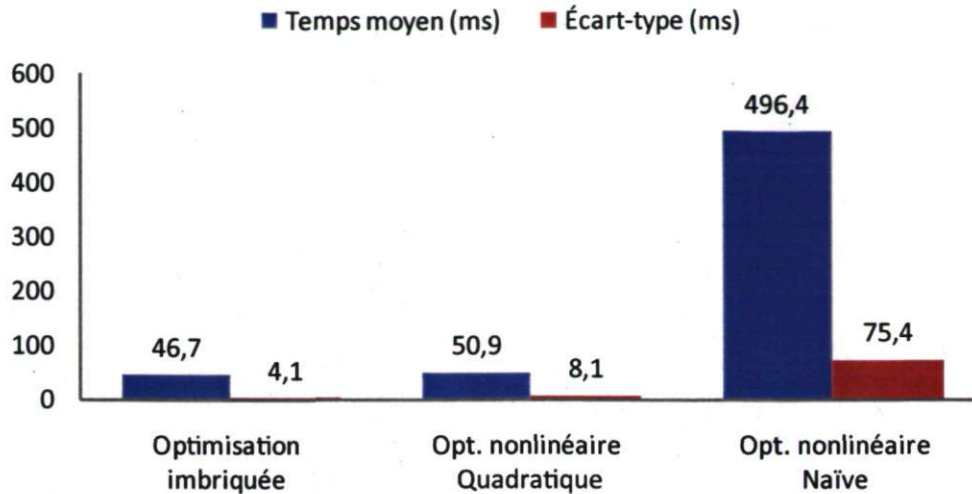


FIGURE 4.16 – Performance comparative sur 100 essais des différentes formulations du problème d’estimation de forces de réaction du sol. Les valeurs données sont les temps moyens, et leurs écarts-types, pour l’estimation des forces de réaction du sol pour une trame de support double de la séquence de marche. Les temps sont donnés en millisecondes et ont été mesurés sur un processeur Intel Core i7 920.

Analyse

Les résultats présentés dans la figure 4.16 montrent bien le gain de performance obtenu par la formulation quadratique de la fonction de coût. Une différence de performance si grande entre la formulation naïve et les deux autres formulations s’explique par le fait que la formulation naïve nécessite le calcul de la dynamique inverse à chaque pas d’optimisation afin de recalculer les différents vecteurs de force \hat{f}_i . Par comparaison, la forme quadratique ne nécessite qu’un seul calcul de la dynamique inverse afin de calculer les constantes \hat{H} , \hat{m} et b .

Nous observons également un gain de performance léger mais significatif de la formulation imbriquée par rapport à la formulation non linéaire/quadratique. Ce gain est imputable à la subdivision du problème initial en sous-problèmes de QP.

Soulignons que certains algorithmes d’optimisation non linéaire, dont l’algorithme de *programmation quadratique séquentielle* (SQP) utilisé par *fmincon*, consistent essentiellement à résoudre une séquence de problèmes de QP approximant localement le problème de programmation non linéaire. Ces algorithmes se trouvent donc également à subdiviser le problème initial en sous-problèmes de QP. La différence principale avec notre approche imbriquée est que les problèmes de QP formés par un algorithme de SQP sont des *approximations* locales du problème initial. Par opposition, le sous-problème

de l'équation 4.30 est une formulation *exacte* du problème initial lorsque nous fixons une variable.

Dans les faits, nous observons que notre formulation imbriquée doit généralement résoudre un plus grand nombre de problèmes de QP que la formulation non linéaire / quadratique, ce qui est dû à l'absence de dérivée analytique pour la boucle externe de la formulation imbriquée. Toutefois, les problèmes de QP de la formulation imbriquée sont mieux posés et se résolvent plus rapidement.

Cette comparaison de performance permet de bien apprécier l'importance des deux aspects distinctifs de l'optimisation imbriquée, soit la formulation de la fonction de coût sous sa forme quadratique et l'optimisation du paramètre x_{fz} dans une boucle externe.

Chapitre 5

Synthèse de mouvements

Ce chapitre présente le coeur de notre approche d'extrapolation de mouvements. Il se divise comme suit.

Dans un premier temps, nous présentons notre implémentation des contraintes spatiotemporelles. Dans cette section, nous nous attardons à bien définir l'ensemble des fonctions de coût et fonctions de contraintes qui composent le problème d'optimisation à résoudre pour faire la synthèse de mouvements plausibles et réalistes. Comme nous le verrons, l'optimisation d'un mouvement est un outil qui sera utilisé à plusieurs occasions pour atteindre différents objectifs dans notre approche.

Dans un deuxième temps, nous introduisons la synthèse d'un mouvement tuteur. Cela consiste à prendre des données issues de capture de mouvement et à faire la synthèse d'un mouvement physiquement plausible le plus proche de ces données. Cette étape est préalable à notre approche d'extrapolation de mouvement puisque le mouvement tuteur en est une composante essentielle. Nous abordons également le problème de simplification du modèle utilisé lors de la capture de mouvement, ce qui consiste à en réduire le nombre de coordonnées articulaires afin de le faire correspondre au personnage utilisé dans la synthèse de mouvements.

Finalement, nous présentons notre approche d'extrapolation de mouvement. Nous introduisons notre approche d'optimisation de mouvement incrémentale et nous expliquons comment notre approche d'extrapolation fait la synthèse de mouvements dans différents environnements à partir d'un tuteur.

Nous concluons le chapitre en présentant un exemple réel d'application de notre approche sur un modèle simplifié de personnage humain. Les différentes étapes de l'ap-

proche et leurs résultats sont présentés, suite à quoi nous proposons une brève discussion sur la portée d'une telle approche d'animation et de ses limites.

Dans ce chapitre nous utiliserons les termes *mouvement tuteur* et *mouvement directeur* pour désigner deux éléments distincts de notre approche. Nous désignons par le terme *mouvement tuteur* le mouvement de base utilisé dans notre approche d'extrapolation de mouvement. À titre d'exemple, un mouvement de marche en ligne droite est utilisé comme mouvement tuteur pour produire des mouvements de marche suivant une trajectoire courbe.

Nous utilisons le terme *mouvement directeur* pour désigner le mouvement qui doit être imité le plus fidèlement possible lors de l'optimisation d'un mouvement. Comme nous le verrons plus loin, la différence entre une solution possible et le mouvement directeur est un terme de la fonction de coût lors d'une telle optimisation. Le mouvement directeur sera obtenu à partir par une transformation appliquée au mouvement tuteur. Cette transformation a pour but d'adapter le mouvement directeur à un nouvel environnement par une approximation rapide. Le mouvement directeur conserve donc un certain réalisme mais ne sera généralement pas plausible. Afin de ne pas trop altérer le réalisme du mouvement, la transformation s'appliquera généralement uniquement à la racine du système multicorps, mais une transformation plus complète pourrait aussi être envisagée.

5.1 Optimisation d'un mouvement

Nous nous inspirons de Fang et Pollard [20, 21] pour implémenter une approche de contraintes spatiotemporelles basée sur la quantité de mouvement globale du personnage. C'est-à-dire que nous considérons exclusivement la dynamique du personnage dans son ensemble, soit la correspondance entre les forces externes qui lui sont appliquées et le taux de variation de la quantité de mouvement du personnage. Nous omettons donc la dynamique *interne* du personnage, soit la correspondance entre les taux de variation des quantités de mouvement des corps rigides et efforts articulaires.

Il est important de comprendre que malgré cette omission, les mouvements obtenus par notre approche demeurent physiquement plausibles. Cela veut simplement dire que les forces internes du personnage sont toujours implicitement fixées de façon à respecter l'équation de dynamique inverse (équation 3.30). Il est donc impossible de poser des contraintes supplémentaires sur ces forces, notamment pour en limiter l'amplitude. Cela ne pose pas problème dans notre approche puisque les mouvements synthétisés sont

similaires au mouvement original et que la marche, peu importe la variation que nous en produisons, ne requiert jamais une force surhumaine. Cela pourrait toutefois devenir plus problématique pour des mouvements plus énergiques, tels que des mouvements acrobatiques par exemple.

Cela signifie également qu'il est impossible d'utiliser les forces internes du personnage dans une éventuelle fonction de coût. À titre d'exemple, nous ne sommes pas en mesure de faire la synthèse d'un mouvement minimisant la somme des forces internes du personnage. Une fois de plus, cela ne pose pas problème puisque, dans le domaine de l'animation basée sur la physique, il est connu que ce type de fonction de coût n'est pas adapté pour les mouvements dont les forces sont relativement modestes [49, 84], ce qui inclut la marche humaine, par opposition à la course par exemple. Une fois de plus, cela pourrait toutefois devenir un handicap si nous voulions faire la synthèse de mouvements plus énergiques.

Du point de vue de l'implémentation, les différences entre une approche de contraintes spatiotemporelles *classique* et une basée sur la quantité de mouvement globale du personnage sont mineures. Bien que nous ne présentons que la seconde, il serait relativement simple d'implémenter les deux approches et d'alterner selon le mouvement traité.

Cela dit, l'avantage principal d'une approche basée sur la quantité de mouvement globale du personnage est qu'elle n'utilise pas explicitement les éléments de Γ , la matrice des forces généralisées, comme paramètres. On réduit donc de nT le nombre de paramètres par rapport à l'approche *classique*, où n est le nombre de coordonnées articulaires du personnage et T est le nombre de trames de l'animation.

De plus, les approches basées sur la quantité de mouvement globale du personnage réduisent aussi le nombre de contraintes servant à imposer le respect des équations de mouvement. En effet, pour une trame d'animation t , la seule équation à respecter est $\hat{f}_{0,t} = \hat{0}$, alors que l'approche *classique* demande une contrainte d'équation de mouvement par coordonnée articulaire. Le nombre de contraintes est donc réduit de $(n - 6)T$.

Nous regroupons tous les paramètres du mouvement dans un vecteur quelconque \mathbf{x} . Ce vecteur contient les éléments de \mathbf{Q} , soit la matrice du mouvement en coordonnées généralisées (équation 3.5), et les vecteurs $\hat{f}_{ext,d,t}$ et $\hat{f}_{ext,g,t}$, soient les forces externes appliquées à chaque temps t sur le pied droit et gauche respectivement. Le vecteur \mathbf{x} a donc $(n + 12)T$ composantes.

Sous le paradigme des contraintes spatiotemporelles, l'animation synthétisée est le résultat d'une optimisation non linéaire soumise à des contraintes non linéaires. Le

problème peut donc être exprimé de la façon suivante :

$$\min_{\mathbf{x}} F(\mathbf{x}) \mid C(\mathbf{x}) \geq 0, C_{eq}(\mathbf{x}) = 0 \quad (5.1)$$

où $F : \mathbb{R}^{(n+12)T} \rightarrow \mathbb{R}$ est la fonction de coût, $C : \mathbb{R}^{(n+12)T} \rightarrow \mathbb{R}^{m_c}$ est la fonction de contraintes d'inégalités et $C_{eq} : \mathbb{R}^{(n+12)T} \rightarrow \mathbb{R}^{m_{ceq}}$ est la fonction de contraintes d'égalités. m_c est le nombre d'équations dans les contraintes d'inégalités et m_{ceq} est le nombre d'équations dans les contraintes d'égalités.

Les fonctions F , C et C_{eq} sont développées dans les sections suivantes. Les fonctions de contraintes sont groupées selon qu'elles s'appliquent à la cinématique, à la cinétique ou à la dynamique du mouvement. Les contraintes cinématiques nous permettent de délimiter quels sont les mouvements admissibles, alors que les contraintes cinétiques nous permettent de délimiter quelles sont les forces de contact admissibles. Les contraintes dynamiques servent à assurer le respect des équations de mouvement.

Soulignons que pour des fins de lisibilité, la dépendance à \mathbf{x} des différents termes des fonctions de coût et de contraintes ne sont pas données explicitement dans la formulation de ces fonctions. Dans ces fonctions, les valeurs ayant une dépendance au vecteur \mathbf{x} sont les matrices de transformations, telles que ${}^i\tilde{W}_{j,t}$, ${}^i\hat{X}_{j,t}$ ou ${}^i\hat{X}_{j,t}^*$, les forces externes $\hat{f}_{ext,i,t}$ et les coordonnées généralisées du mouvement $q_{i,t}$.

5.1.1 Fonctions de coût

Nous cherchons à produire un mouvement qui soit le plus similaire possible à un mouvement directeur tout en respectant les contraintes données. Il faut donc que la fonction de coût exprime la différence entre deux mouvements. À cette fin, nous définissons une mesure de distance entre deux configurations et nous exprimons la différence entre deux mouvements comme la somme des distances de toutes les configurations du mouvement synthétisé avec celles du mouvement directeur. Afin d'éviter la confusion, nous utiliserons le symbole \star en exposant pour identifier les valeurs se rapportant au mouvement directeur.

Plusieurs mesures de distance pourraient être définies pour comparer deux configurations d'un même personnage. En animation, deux mesures sont couramment utilisées à cette fin, soient la somme des différences des coordonnées généralisées en valeur absolues et la somme de distances entre des points du système multicorps. Celles-ci sont présentées à la figure 5.1.

La première mesure est relativement simple, elle consiste à prendre la norme au carré

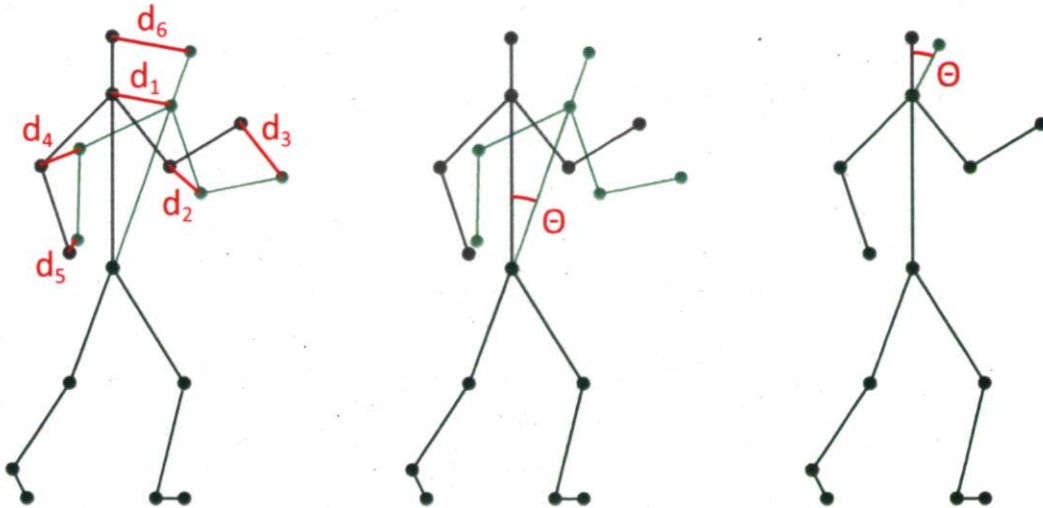


FIGURE 5.1 – Illustration de deux mesures de distances entre deux configurations : une somme de distances euclidiennes entre des marqueurs attachés aux corps rigides (à gauche) et une somme des différences des coordonnées généralisées de chaque configuration (au centre et à droite). Les deux exemples de droite montrent qu'une même différence en coordonnées généralisées peut avoir un impact plus ou moins grand selon l'articulation où cette différence est observée.

de la différence entre le vecteur donnant la configuration du personnage en coordonnées généralisées \mathbf{q}_t et la configuration équivalente dans le mouvement directeur \mathbf{q}_t^* . Cette distance est donnée par $\|\mathbf{q}_t - \mathbf{q}_t^*\|^2$. Comme l'illustre la figure 5.1, cette mesure de distance n'est pas nécessairement adéquate puisqu'elle ne considère pas que la variation de la valeur des différentes coordonnées articulaires du personnage n'ont pas tous le même effet sur la différence perçue entre deux configurations. Il faudrait alors ajouter une pondération à chaque coordonnée articulaire pour faire état de leur impact respectif sur la configuration. Cette pondération peut toutefois s'avérer difficile à établir.

La seconde mesure consiste à positionner des marqueurs sur le personnage et à sommer la distance euclidienne entre la position de chaque marqueur dans la configuration synthétisée et leur position dans la configuration du mouvement directeur. Cette mesure correspond généralement mieux à la différence perçue entre deux configurations, surtout lorsque les distances sont faibles. En positionnant un marqueur de part et d'autre de chaque segment du système multicorps, on se trouve à quantifier la distance entre les segments correspondants¹. Cette mesure de distance est toutefois mal adaptée pour des problèmes tels que le reciblage d'animation, où les dimensions et proportions du personnage diffèrent entre la configuration synthétisée et la configuration du mouve-

1. Ces distances pourraient aussi être pondérées afin de mieux faire état de la longueur relative de chaque segment.

ment directeur. Pour nos besoins, la somme des distances de marqueurs est toutefois préférable.

En fait, nous sommons le carré de ces distances. Cela nous permet de pénaliser plus fortement des configurations où quelques marqueurs seraient très loin de la position désirée. En d'autres termes, nous préférons une configuration où chaque marqueur est relativement proche de la position désirée par rapport à une configuration où la majorité des marqueurs sont exactement à la position désirée mais dont une poignée de marqueurs sont très distants de cette position. Cela correspond à notre hypothèse que la synthèse d'un mouvement dans deux environnements très similaires ne devrait demander que des ajustements mineurs.

Dans notre implémentation, nous utilisons un marqueur par corps rigide, soit le point homogène \tilde{a}_i , qui se trouve à l'extrémité du corps rigide i . Si l_i est la longueur du corps i , nous avons donc $\tilde{a}_i = (l_i, 0, 0, 1)^t$. La position de ce marqueur sur la configuration du mouvement directeur est donné par $\tilde{a}_{i^*} = (l_i, 0, 0, 1)^t$, la différence étant le repère dans lequel il est exprimé. La distance entre deux mouvements est donc donnée par la fonction :

$$F_{dir}(\mathbf{x}) \stackrel{\text{def}}{=} \frac{1}{nT} \sum_{t=1}^T \sum_{i \in \Omega} \left\| {}^0\tilde{W}_{i,t} \tilde{a}_{i,t} - {}^0\tilde{W}_{i^*,t} \tilde{a}_{i^*,t} \right\|^2. \quad (5.2)$$

Selon la situation, il peut arriver qu'un mouvement tende vers des configurations proches de singularités pour les articulations à 3 DDLs de rotation, tels que les hanches ou les épaules du personnage. Comme nous l'avons vu au chapitre 2, la paramétrisation de ces articulations par des angles d'Euler nous assujettit à ce problème de singularités. Puisque cela peut nuire au processus d'optimisation, il peut être intéressant d'ajouter à la fonction de coût un terme pénalisant de telles configurations.²

Tel que présenté dans la section 3.1.2, nous représentons une articulation de type rotule par 3 articulations rotoïdes successives dont les axes de rotation, en position neutre, sont orthonormaux. La singularité, aussi connue sous le nom de blocage de cardan, survient lorsque la deuxième articulation atteint les angles $\pm\pi/2$, $\pm3\pi/2$, $\pm5\pi/2$ et ainsi de suite. Le problème étant causé par l'alignement des axes de rotation de la première et de la troisième articulation rotoïde, nous pouvons considérer que l'amplitude du problème est inversement proportionnel à la perpendicularité de ces deux axes. Il est donc naturel d'utiliser l'angle de la deuxième articulation pour quantifier l'amplitude

2. Dans d'autres contextes, ce problème peut être évité en effectuant un changement de repère lorsque la configuration du système approche une telle singularité. Dans notre contexte, cela pourrait engendrer des discontinuités dans les fonctions de coût et de contraintes et perturber le processus d'optimisation numérique.

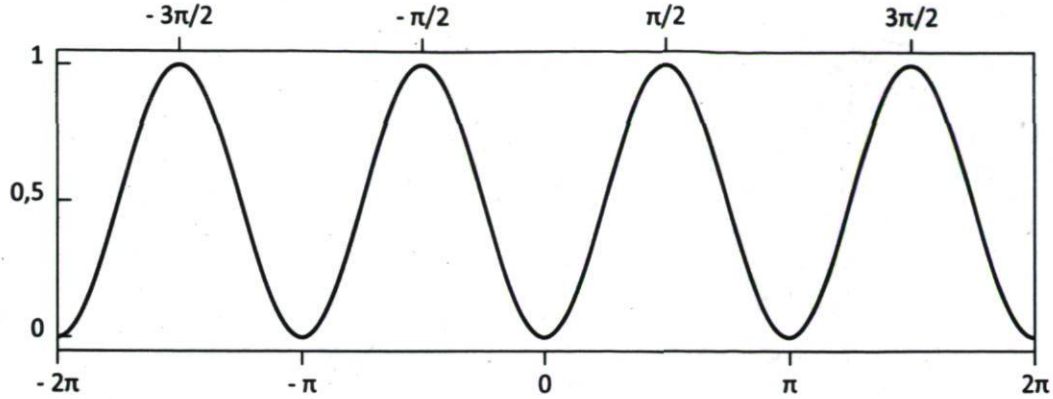


FIGURE 5.2 – La fonction $(1 - \cos(2\theta))/2$ a une valeur de 0 lorsque l'angle θ est un multiple entier de π radians, ce qui correspond à avoir les trois axes de rotation orthogonaux. Par contre, elle a une valeur de 1 pour lorsque θ est à mi-chemin entre les multiples entiers de π radians, ce qui correspond à une singularité pour l'articulation à 3 DDLs de rotation.

du problème.

Soient Ω_{sing} , l'ensemble des indices des articulations susceptibles de provoquer ce type de singularités et n_{sing} , la cardinalité de l'ensemble Ω_{sing} , nous définissons la fonction de coût suivante :

$$F_{sing}(\mathbf{x}) \stackrel{\text{def}}{=} \frac{1}{n_{sing}T} \sum_{t=1}^T \sum_{i \in \Omega_{sing}} \frac{1 - \cos(2q_{i,t})}{2}. \quad (5.3)$$

Cette fonction retourne une valeur entre 0 et 1 indiquant à quel degré le mouvement s'approche de singularités. La figure 5.2 trace la fonction $(1 - \cos(2\theta))/2$ pour aider à mieux comprendre comment un angle $q_{i,t}$ proche d'une singularité est pénalisé par la fonction F_{sing} .

Puisque la fonction de coût d'une optimisation numérique doit ultimement retourner un scalaire, il faut composer les termes F_{dir} et F_{sing} en une seule fonction. La pratique courante est d'additionner les différents termes de la fonction de coût en leur attribuant une certaine pondération η , ce qui nous donne :

$$F(\mathbf{x}) \stackrel{\text{def}}{=} \eta_{dir} F_{dir}(\mathbf{x}) + \eta_{sing} F_{sing}(\mathbf{x}). \quad (5.4)$$

Il est généralement difficile d'ajuster les poids attribués aux différents termes, surtout lorsqu'ils représentent des quantités d'unités différentes, tel que dans notre cas. Nous pouvons toutefois nous appuyer sur le fait que le terme pénalisant les singularités ne donne pas une mesure de la qualité de l'animation synthétisée, il a comme seul objectif de faciliter la convergence du processus d'optimisation. Afin que ce terme n'influence

pas la qualité de l'animation synthétisée, nous utilisons généralement les poids suivants : $\eta_{dir} = 1$ et $\eta_{sing} = 10^{-5}$. Ces poids ont été fixés empiriquement en observant que la valeur maximum du terme F_{sing} est de 1, qu'une valeur de 10^{-5} pour le terme F_{dir} correspond à une distance moyenne d'environ 3 mm par marqueur par trame et qu'une telle distance n'est généralement pas perceptible dans un mouvement de marche. En d'autres mots, étant donné l'importance relative accordée à chacun des termes, les ajustements apportés afin de réduire les problèmes de singularité ne seront pas perceptibles dans le mouvement³.

Dans plusieurs cas, le terme pénalisant les singularités ne s'avère pas nécessaire. Cela est particulièrement vrai pour la procédure d'extrapolation de mouvement itérative, lorsque le mouvement initial est relativement loin de toute singularité. Nous pourrions alors omettre le second terme, ce qui correspond à utiliser les poids suivants : $\eta_{dir} = 1$ et $\eta_{sing} = 0$.

5.1.2 Contraintes cinématiques

Nous posons certaines contraintes cinématiques sur le mouvement afin de mieux le contrôler et d'en assurer un certain réalisme.

Coordonnées articulaires bloquées

Un premier terme nous sert à bloquer certaines coordonnées articulaires du personnage à une valeur donnée. Comme nous le verrons plus loin dans le chapitre, ce terme s'avère utile lorsque nous désirons procéder à un élagage du graphe des connexions du personnage pour ne conserver que les articulations pertinentes au mouvement. Il s'agit d'une contrainte d'égalité posée directement sur les coordonnées généralisées. Soient Ω_{lock} , l'ensemble des indices des articulations bloquées, et κ_i , la valeur à laquelle l'articulation i est bloquée, cette contrainte s'exprime à l'aide de la fonction suivante :

$$C_{lock,i,t}(\mathbf{x}) \stackrel{\text{def}}{=} q_{i,t} - \kappa_i. \quad (5.5)$$

Pour être valide, une solution \mathbf{x} doit respecter les conditions suivantes :

$$C_{lock,i,t}(\mathbf{x}) = 0, \quad \forall i \in \Omega_{lock}, \forall t \in \{1, 2, \dots, T\}, \quad (5.6)$$

c'est à dire qu'une articulation i bloquée doit avoir exactement la valeur κ_i à chaque trame du mouvement.

3. De tels ajustement consistent généralement à faire tourner un corps rigide autour de son axe des x , ce qui permet de modifier l'état des articulations sans toutefois changer la position du marqueur

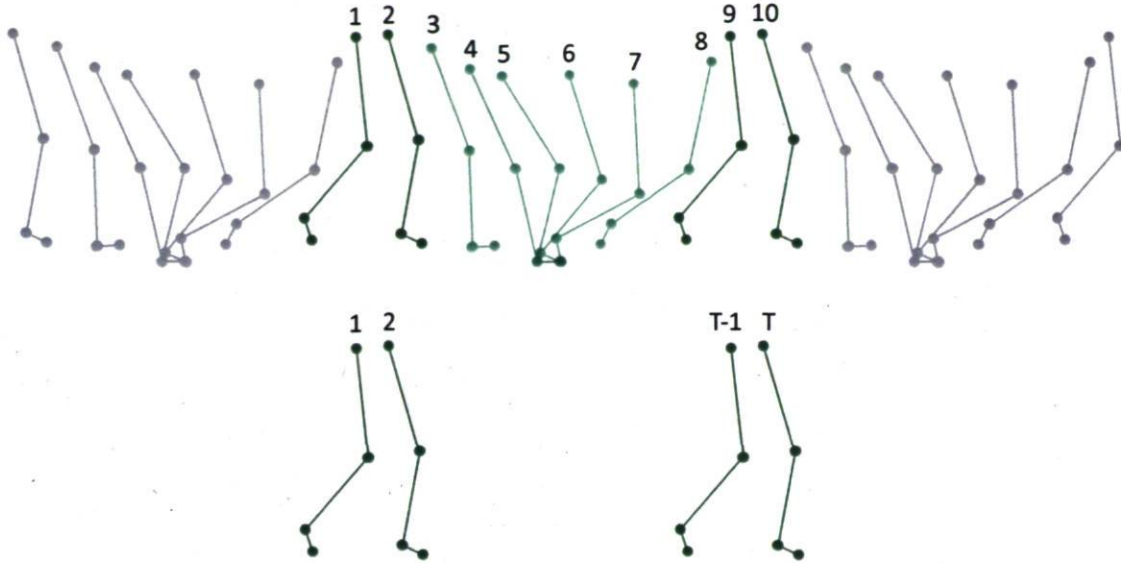


FIGURE 5.3 – Illustration d'un mouvement cyclique pour un personnage simplifié. Le mouvement contient 10 trames qui forment un cycle avec un chevauchement de deux trames. Ce chevauchement sert à valider la continuité entre la fin d'un cycle du mouvement et le début d'un nouveau cycle. Cela se fait par le biais de contraintes cinématiques sur les deux premières et les deux dernières configurations du mouvement. Deux trames suffisent à assurer la continuité du mouvement et de sa première dérivée temporelle.

Mouvement cyclique

Nous pouvons vouloir nous assurer que le mouvement de marche synthétisé soit cyclique. Pour assurer une continuité suffisante entre la fin et le début du mouvement, nous devons nous assurer que la vitesse de translation et de rotation du personnage est identique au début et à la fin du mouvement, que la configuration du personnage soit identique à ces deux temps et que ces configurations aient le même taux de variation. La figure 5.3 illustre la situation.

Nous procédons en posant des contraintes sur les deux premières et les deux dernières trames du mouvement. Notamment, nous posons que la première et l'avant-dernière configurations doivent être identiques à l'exception des 6 coordonnées articulaires de la racine. Il en va de même de la deuxième et de la dernière configuration. Sous forme d'équation, la différence entre ces configurations pour les différentes coordonnées articulaires peut s'exprimer par l'équation suivante :

$$C_{cpos,i}(\mathbf{x}) \stackrel{\text{def}}{=} |q_{i,1} - q_{i,T-1}| + |q_{i,2} - q_{i,T}| \quad (5.7)$$

et la condition à respecter s'exprime par

$$C_{cpos,i}(\mathbf{x}) = 0, \quad \forall i \in \{7, 8, \dots, n\}. \quad (5.8)$$

Puisque nous vérifions l'égalité entre deux paires de configurations consécutives, nous vérifions également que le taux de variation de ces configurations est identique.

Ensuite, pour assurer que la vitesse de translation et de rotation de la racine soit identique au début et à la fin du mouvement, nous comparons les taux de variation des 6 coordonnées articulaires de la racine. Ce taux de variation s'obtient par l'équation suivante :

$$C_{cvel,i}(\mathbf{x}) \stackrel{\text{def}}{=} (q_{i,1} - q_{i,2}) - (q_{i,T-1} - q_{i,T}). \quad (5.9)$$

La contrainte sur les 6 coordonnées articulaires de la racine est alors

$$C_{cvel,i}(\mathbf{x}) = 0, \quad \forall i \in \{1, 2, \dots, 6\}. \quad (5.10)$$

Temps de contact

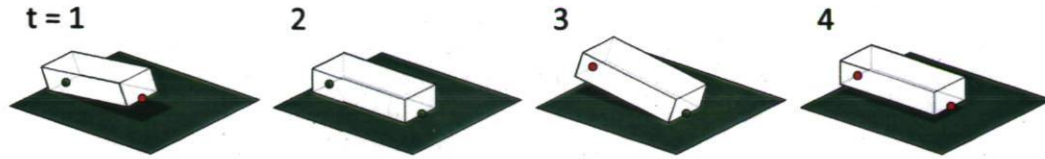
Les contraintes suivantes s'appliquent par rapport aux points de contact du personnage. Nous identifions ces points de contact par des vecteur homogènes \tilde{c}_i . Ces valeurs sont constantes et sont définies dans le modèle du personnage. Le point de contact \tilde{c}_i est attaché à un corps rigide. L'indice de ce corps rigide nous est donné par $\nu(i)$. Nous utilisons généralement deux points de contact par pied dans nos personnages, un au talon et un à la pointe du pied, ce qui nous donne un total de 4 points de contact.

Afin de produire un mouvement de synthèse, il nous faut déterminer les temps de contact. Dans notre approche, définir les temps de contact du mouvement consiste à construire une matrice binaire $A_{m \times T}$, où m est le nombre de points de contact du personnage et T est le nombre de trames du mouvement. Chaque élément de cette matrice est identifiée par un scalaire $\alpha_{i,t}$. Si le point de contact i doit être au sol au temps t , alors nous dirons que le point de contact i est *actif* au temps t et nous aurons $\alpha_{i,t} = 1$. Si, au contraire, ce point ne doit pas être en contact avec le sol, nous dirons que le point de contact i est *inactif* au temps t et nous aurons $\alpha_{i,t} = 0$. La matrice A peut être construite en observant un mouvement issu de capture de mouvement. Un exemple de matrice A est illustré dans la figure 5.4.

La fonction suivante nous permet de connaître la valeur en z de la position du point de contact i au temps t par rapport au repère inertiel :

$$C_{zval,i,t}(\mathbf{x}) \stackrel{\text{def}}{=} (0 \ 0 \ 1 \ 0)^0 \tilde{W}_{\nu(i),t} \tilde{c}_i. \quad (5.11)$$

Tel que mentionné dans la section 3.2, nous utilisons toujours le plan $z = 0$ pour



$$A = \begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & \alpha_{1,3} & \alpha_{1,4} \\ \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} & \alpha_{2,4} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} \beta_{1,1} & \beta_{1,2} & \beta_{1,3} & \beta_{1,4} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 \end{pmatrix}$$

FIGURE 5.4 – Illustration du principe des temps de contact sur un pied ayant deux points de contact. La figure montre quatre configurations distinctes selon que les points de contact sont actifs (en vert) ou inactifs (en rouge). La matrice A donne les temps de contact correspondants à cette animation fictive de quatre trames. La matrice B donne les temps de contact du corps rigide du pied. Si au moins un des points de contact du pied est actif dans une trame donnée, le pied est considéré être en contact avec le sol dans cette trame.

représenter le sol⁴. Ainsi, une valeur nulle de $C_{zval,i,t}(\mathbf{x})$ indique que le point de contact i est sur le sol, une valeur positive indique que le point est au dessus du sol et une valeur négative indique que le point est en dessous du sol. L'évaluation de cette fonction doit donc être nulle pour tout point de contact actif, soit

$$C_{zval,i,t}(\mathbf{x}) = 0, \quad \forall (i,t) \in \{(i,t) | \alpha_{i,t} = 1\}. \quad (5.12)$$

De plus, pour nous assurer qu'un point de contact ne passe jamais sous le sol, il suffit de vérifier que sa position absolue en z est positive lorsqu'il est inactif :

$$C_{zval,i,t}(\mathbf{x}) > 0, \quad \forall (i,t) \in \{(i,t) | \alpha_{i,t} = 0\}. \quad (5.13)$$

Frottement statique

Comme nous le verrons dans la prochaine section, les forces de contact doivent demeurer à l'intérieur du cône de frottement statique. De telles forces ne provoquent pas de glissement du point de contact sur la surface. Au niveau cinématique, nous nous assurons donc que les points de contact ne glissent pas sur le sol lorsqu'ils sont actifs

4. Un plan incliné peut être simulé en appliquant une rotation au vecteur d'accélération gravitationnel

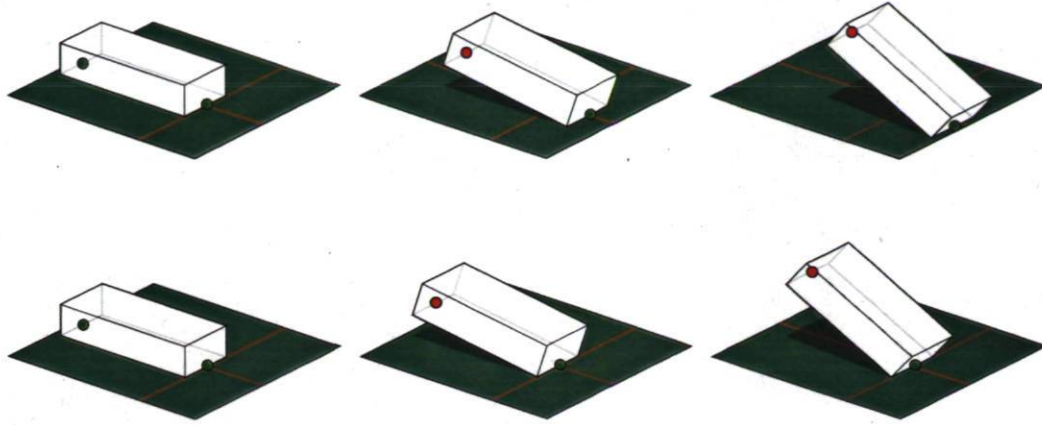


FIGURE 5.5 – Trois trame d’une animation où un point de contact glisse sur le sol alors qu’il est actif (en haut) et trois trames d’une animation où le glissement est corrigé (en bas).

(voir figure 5.5). Nous vérifions cette condition en nous assurant qu’un point de contact ne change pas de position entre deux trames où il est actif.

La fonction suivante nous permet de connaître le vecteur de déplacement d’un point de contact entre deux trames successives :

$$C_{fric,i,t}(\mathbf{x}) \stackrel{\text{def}}{=} {}^0\tilde{W}_{\nu(i),t}\tilde{\mathbf{c}}_i - {}^0\tilde{W}_{\nu(i),t+1}\tilde{\mathbf{c}}_i. \quad (5.14)$$

Pour respecter la contrainte de frottement statique, un point de contact actif dans deux trames consécutives doit avoir un déplacement nul. Cela revient à respecter l’équation suivante :

$$C_{fric,i,t}(\mathbf{x}) = \vec{0}, \quad \forall (i,t) \in \{(i,t) | \alpha_{i,t} = 1 \wedge \alpha_{i,t+1} = 1\}. \quad (5.15)$$

5.1.3 Contraintes cinétiques

Les contraintes cinétiques s’appliquent aux paramètres de force contenus dans le vecteur \mathbf{x} . Puisque nous utilisons une approche basée sur la quantité de mouvement globale du personnage, les forces de contact sont les seuls paramètres de force décrivant le mouvement.

Dans l’expression des contraintes cinétiques, nous utiliserons une matrice binaire $B_{n \times T}$ qui nous indique si un corps rigide doit être en contact avec le sol au temps t . Cette matrice est très similaire à la matrice A définie plus tôt. La différence est que la matrice A indique les temps de contact des *points de contact* alors que la matrice B nous donne les temps de contact des *corps rigides*.

Nous considérons qu'un corps rigide est en contact avec le sol si au moins un de ses points de contact est actif. La matrice B peut donc facilement être construite à partir de la matrice A . L'élément $\beta_{i,t}$ sera égal à 1 seulement si au moins un des éléments $\alpha_{j,t}$ est égal à 1 pour l'ensemble des points j attachés au corps i . L'élément $\beta_{i,t}$ sera nul autrement⁵.

Forces de contact nulles

Lorsque le corps rigide i n'est pas en contact avec le sol, nous devons nous assurer que la force de contact correspondante est nulle.

En utilisant la fonction suivante,

$$C_{f_{ext,i,t}}(\mathbf{x}) \stackrel{\text{def}}{=} \left\| \hat{f}_{ext,i,t} \right\|, \quad (5.16)$$

cela correspond à remplir la contrainte suivante :

$$C_{f_{ext,i,t}}(\mathbf{x}) = 0, \quad \forall (i,t) \in \{(i,t) | \beta_{i,t} = 0\}. \quad (5.17)$$

Cône de frottement

Lorsqu'un corps rigide est en contact avec le sol, la force de contact associée doit être contenue dans le cône de frottement. En d'autres termes, la force de contact ne doit pas être *trop* parallèle à la surface de contact pour être plausible.

Pour vérifier cette condition, nous devons d'abord exprimer les trois dernières composantes d'une force spatiale de contact $\hat{f}_{ext,i,t}$ comme la somme d'une force normale à la surface de contact et d'une force parallèle à la surface de contact : $\vec{f}_{ext,i,t_\perp} + \vec{f}_{ext,i,t_\parallel}$. Soulignons que par rapport au repère intertiel, la composante normale de la force de contact s'exprime par un seul scalaire : $[\vec{f}_{ext,i,t_\perp}]_0 = (0, 0, f_{fz,i,t_\perp})^t$.

Un coefficient de frottement μ détermine l'étendu du cône de frottement en donnant le ratio maximum permis entre la norme de la force parallèle et celle de la force normale : $\|\vec{f}_{ext,i,t_\parallel}\| / \|f_{fz,i,t_\perp}\| \leq \mu$. La figure 5.6 illustre le principe du cône de frottement.

5. Puisque notre implémentation présume que seuls les pieds gauche et droit ont des points de contact, nous n'avons que deux lignes de la matrice B à construire. Toutefois, afin de ne pas alourdir la notation davantage, nous considérons une matrice B de n lignes dont toutes les lignes seront nulles à l'exception des deux lignes correspondant aux pieds. Cela nous permet d'utiliser directement les indices des corps rigides dans nos équations.

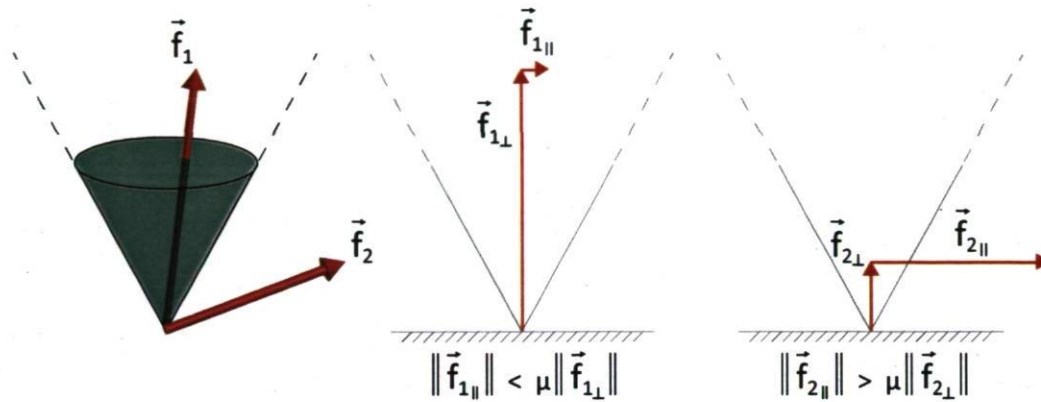


FIGURE 5.6 – Illustration du principe du cône de frottement. La force \vec{f}_1 est contenue dans le cône de frottement puisque la norme de sa composante parallèle au sol est au moins $1/\mu$ fois plus petite que la norme de la composante normale de cette force. Ce n'est pas le cas pour la force \vec{f}_2 qui se trouve à l'extérieur du cône de frottement. Dans cet exemple, le coefficient de frottement est de 0,5. En augmentant la valeur du coefficient de frottement μ , on augmente l'angle solide du cône.

La fonction suivante nous permet de déterminer si une force est contenue ou non à l'intérieur du cône défini par le coefficient μ :

$$C_{\text{cône},i,t}(\mathbf{x}) \stackrel{\text{def}}{=} \mu f_{fz,i,t_\perp} - \|\vec{f}_{\text{ext},i,t_\parallel}\|. \quad (5.18)$$

Le valeur donnée par cette fonction sera négative si les trois dernières composantes de la force spatiale $\hat{f}_{\text{ext},i,t}$ est à l'extérieur du cône. Elle sera positive autrement. Nous pouvons donc poser la contrainte du cône de frottement sur les corps rigides en contact par l'équation suivante :

$$C_{\text{cône},i,t}(\mathbf{x}) \geq 0, \quad \forall (i, t) \in \{(i, t) | \beta_{j,t} = 1\}. \quad (5.19)$$

De plus, il faut s'assurer que la composante normale de la force de contact est positive, autrement le sol pourrait tirer le personnage vers lui. Pour valider cela, nous utilisons la fonction suivante

$$C_{\text{push},i,t}(\mathbf{x}) \stackrel{\text{def}}{=} f_{fz,i,t_\perp} \quad (5.20)$$

à l'aide de laquelle nous posons la condition suivante, pour assurer une force normale non négative pour tous les corps rigides en contact :

$$C_{\text{push},i,t}(\mathbf{x}) \geq 0, \quad \forall (i, t) \in \{(i, t) | \beta_{j,t} = 1\}. \quad (5.21)$$

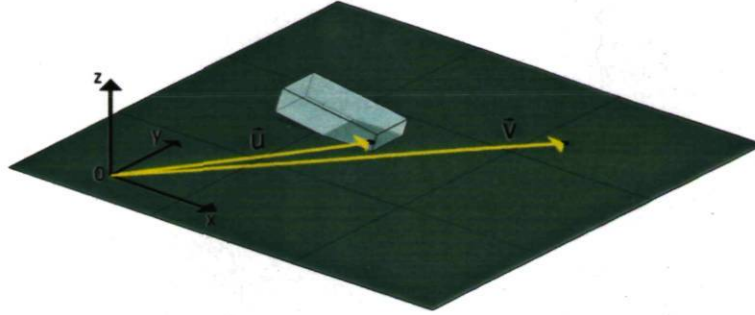


FIGURE 5.7 – Deux forces distinctes ont des centres de pression distincts. Dans l'exemple présenté, le centre de pression \vec{u} correspond à une force de contact valide puisqu'il est contenu dans l'intersection entre le sol et la boîte englobante du corps rigide. Par opposition, le centre de pression \vec{v} ne correspond pas à une force de contact valide. Rappelons qu'une telle pénétration d'un corps rigide dans le sol est possible lorsque les points de contact sont placés à l'intérieur de la boîte englobante et non sur sa surface.

Centre de pression

Lorsqu'un corps rigide est en contact avec le sol, le centre de pression de la force de contact doit être contenu dans la surface de contact entre le sol et le corps rigide. Ces contraintes sont similaires à celles posées par les équations 4.20 et 4.21 de notre approche d'estimation des forces de contact. Nous les reformulons ici sous une forme plus générale.

Soit une force de contact $\hat{f}_{ext,i,t}$ appliquée par le sol sur le corps i au temps t . Si nous exprimons cette force dans le repère inertiel à l'aide de ses composantes : $[\hat{f}_{ext,i,t}]_0 = (f_{\tau x,i,t}, f_{\tau y,i,t}, f_{\tau z,i,t}, f_{fx,i,t}, f_{fy,i,t}, f_{fz,i,t})^t$, nous pouvons exprimer son centre de pression, soit le point où l'Expression de la force de contact ne comporte aucun couple selon les axes x et y , par le vecteur $\vec{u}_{i,t}$:

$$[\vec{u}_{i,t}]_0 = \left(-\frac{f_{\tau y,i,t}}{f_{fz}}, \frac{f_{\tau x,i,t}}{f_{fz}}, 0 \right). \quad (5.22)$$

La contrainte du centre de pression assure que le centre de pression se trouve sur l'intersection du sol et de la boîte englobante du corps rigide. La figure 5.7 illustre ce principe. Soulignons que le centre de pression est, par définition, un point sur le sol.

En introduisant le repère ι , un repère dont l'origine est le centre de masse du corps rigide i et dont les axes sont orientés de façon identique au repère i , il est possible d'exprimer le centre de pression par rapport au centre du corps rigide i : $[\vec{u}_{i,t}]_{\iota} = {}^{\iota}R_0[\vec{u}_{i,t}]_0 + {}^{\iota}\vec{t}_0$.

Nous définissons ensuite le vecteur $\vec{\delta}_i$ qui regroupe les longueurs des demi-axes a_i , b_i et c_i du corps rigide i : $\vec{\delta}_i = (a_i, b_i, c_i)^t$.

Les fonctions suivantes permettent de connaître la position du centre de pression par rapport aux 6 plans qui délimitent la boîte englobante du corps rigide i :

$$C_{copp,i,t}(\mathbf{x}) \stackrel{\text{def}}{=} -[\vec{u}_{i,t}]_l + \vec{\delta}_i, \quad (5.23)$$

$$C_{copn,i,t}(\mathbf{x}) \stackrel{\text{def}}{=} [\vec{u}_{i,t}]_l + \vec{\delta}_i. \quad (5.24)$$

Les valeurs des éléments retournés par ces fonctions seront toutes positives ou nulles seulement si le centre de pression est à l'intérieur de la boîte englobante.

Nous pouvons utiliser ces fonctions pour poser les six contraintes qui nous assurent que le centre de pression de la force $\hat{f}_{ext,i,t}$ est bien contenu dans la boîte englobante du corps rigide i lorsque ce corps est en contact avec le sol :

$$C_{copp,i,t}(\mathbf{x}) \geq \vec{0}, \quad \forall (i,t) \in \{(i,t) | \beta_{i,t} = 1\} \quad (5.25)$$

$$C_{copn,i,t}(\mathbf{x}) \geq \vec{0}, \quad \forall (i,t) \in \{(i,t) | \beta_{i,t} = 1\}. \quad (5.26)$$

5.1.4 Contraintes dynamiques

Les contraintes dynamiques sont à la base des approches de contraintes spatiotemporelles. Elles assurent le respect des équations de mouvement du personnage. Dans le cas d'une approche basée sur la quantité de mouvement globale du personnage, ces contraintes consistent uniquement à assurer que les efforts articulaires de la racine sont nuls. Cela revient à dire que toute variation de la quantité de mouvement globale du personnage doit être le résultat de l'application de forces externes.

La fonction suivante nous donne, pour le temps t , la force interne qui doit être appliquée à la racine pour expliquer son mouvement :

$$C_{eom,t}(\mathbf{x}) \stackrel{\text{def}}{=} \sum_{i=1}^n {}^0\hat{X}_{i,t}^* \dot{\hat{e}}_{i,t} - \sum_{i=1}^n {}^0\hat{X}_{i,t}^* \hat{f}_{ext,i,t} = \hat{f}_{0,t}. \quad (5.27)$$

où $\dot{\hat{e}}_{i,t}$ est le taux de variation de la quantité de mouvement spatiale du corps i au temps t . Afin d'être plausible, cette force doit être nulle à chaque trame de l'animation. Le mouvement doit donc respecter l'équation suivante :

$$C_{eom,t}(\mathbf{x}) = \hat{0}, \quad \forall t \in \{1, 2, \dots, T\}. \quad (5.28)$$

5.1.5 Paramètres du problème d'optimisation

Tel que mentionné plus tôt, l'optimisation d'un mouvement est pour nous un outil de base que nous utilisons à diverses fins dans notre approche d'extrapolation de mouvements. Suite à la description des différents termes qui interviennent dans notre implémentation des contraintes spatiotemporelles, nous pouvons réécrire de façon plus détaillée le problème d'optimisation qui lui est associé :

$$\min_{\mathbf{x}} F(\mathbf{x}) \mid \left\{ \begin{array}{l} C_{lock,i,t}(\mathbf{x}) = 0, \quad \forall i \in \Omega_{lock}, \forall t \in \{1, 2, \dots, T\} \\ C_{cpos,i}(\mathbf{x}) = 0, \quad \forall i \in \{7, 8, \dots, n\} \\ C_{cvel,i}(\mathbf{x}) = 0, \quad \forall i \in \{1, 2, \dots, 6\} \\ C_{zval,i,t}(\mathbf{x}) = 0, \quad \forall (i, t) \in \{(i, t) \mid \alpha_{i,t} = 1\} \\ C_{zval,i,t}(\mathbf{x}) > 0, \quad \forall (i, t) \in \{(i, t) \mid \alpha_{i,t} = 0\} \\ C_{fric,i,t}(\mathbf{x}) = \vec{0}, \quad \forall (i, t) \in \{(i, t) \mid \alpha_{i,t} = 1 \wedge \alpha_{i,t+1} = 1\} \\ C_{fext,i,t}(\mathbf{x}) = 0, \quad \forall (i, t) \in \{(i, t) \mid \beta_{i,t} = 0\} \\ C_{cone,i,t}(\mathbf{x}) \geq 0, \quad \forall (i, t) \in \{(i, t) \mid \beta_{j,t} = 1\} \\ C_{push,i,t}(\mathbf{x}) \geq 0, \quad \forall (i, t) \in \{(i, t) \mid \beta_{j,t} = 1\} \\ C_{copp,i,t}(\mathbf{x}) \geq \vec{0}, \quad \forall (i, t) \in \{(i, t) \mid \beta_{i,t} = 1\} \\ C_{copn,i,t}(\mathbf{x}) \geq \vec{0}, \quad \forall (i, t) \in \{(i, t) \mid \beta_{i,t} = 1\} \\ C_{eom,t}(\mathbf{x}) = \hat{0}, \quad \forall t \in \{1, 2, \dots, T\} \end{array} \right. \quad (5.29)$$

Bien que cette formulation puisse sembler quelque peu aride, il faut simplement en comprendre que la solution à un problème d'optimisation de mouvement doit respecter simultanément l'ensemble des contraintes susmentionnées tout en minimisant la fonction de coût $F(\mathbf{x})$. En fait, toute valeur de \mathbf{x} respectant ces contraintes représente un mouvement plausible. Le réalisme du mouvement, quant à lui, est *atteint* lorsque l'évaluation de la fonction de coût est suffisamment faible, de l'ordre de quelques millimètres par exemple, dans la mesure où le mouvement directeur est lui-même réaliste.

L'évaluation de la fonction de coût et des différentes de contraintes dépend de quelques paramètres. En variant ces paramètres il nous est possible de caractériser différents problèmes auxquels nous faisons face et, éventuellement, de les résoudre par optimisation numérique. Comme nous le verrons plus tard, en utilisant différents paramètres nous pouvons notamment rendre un mouvement plausible, simplifier le graphe des connexions d'un personnage ou extrapoler un mouvement. Ces paramètres sont décrits dans le tableau 5.1.

Dans les problèmes que nous formulons plus loin, certains paramètres demeurent constants : la pondération utilisée y est toujours $\eta_{dir} = 1$ et $\eta_{sing} = 10^{-5}$, les temps de contact correspondent toujours à ceux du mouvement directeur et le mouvement est toujours cyclique. Nous utiliserons donc uniquement les quatre autres paramètres

Paramètres	Description
Personnage	Modèle biomécanique utilisé pour faire la synthèse de mouvement. Sa description comprend notamment le graphe des connexions et la géométrie du personnage, les masses et tenseurs d'inertie des corps rigides qui le composent ainsi que ses points de contact.
Mouvement directeur	Mouvement original duquel le mouvement optimal doit s'approcher le plus possible. Il est composé d'une fréquence d'échantillonnage et de la matrice \mathbf{Q}^* , qui contient la valeurs des différentes coordonnées articulaires en fonction du temps.
Environnement	Caractéristiques de l'environnement de synthèse, données par \vec{g} , le vecteur d'accélération gravitationnel, et μ , le coefficient de frottement entre le sol et les pieds du personnage.
Coordonnées bloquées	Liste des indices des coordonnées articulaires du personnage dont la valeur doit demeurer constante tout au long du mouvement ainsi que les valeurs κ_i que doivent prendre ces coordonnées articulaires.
Pondération	Poids respectifs dans la fonction de coût du terme de distance au mouvement directeur, donné par η_{dir} , et du terme pénalisant les singularités, donné par η_{sing} .
Temps de contact	Description des temps de contact devant être respectés pendant le mouvement. Ils sont fixés par la matrice \mathbf{A} donnant les temps de contact pour les différents points de contact du personnage et, par extension, la matrice \mathbf{B} donnant les temps de contact des différents corps rigides du personnage.
Mouvement cyclique	Indique si le mouvement est cyclique ou non.

TABLE 5.1 – Description des divers paramètres utilisés dans la caractérisation d'un problème d'optimisation de mouvement.

pour définir les différents problèmes d'optimisation de mouvement, soient le personnage utilisé, le mouvement directeur, la description de l'environnement et la description des coordonnées articulaires bloquées.

Outre les paramètres d'un problème, celui-ci est également caractérisé par une valeur initiale, soit la valeur à partir de laquelle l'optimisation est lancée, ainsi que par l'optimum du problème posé. Dans les sections suivantes, nous décrivons chaque problème d'optimisation de mouvement en termes de ses paramètres, de sa valeur initiale et de son optimum.

5.2 Production du mouvement tuteur

L'approche d'extrapolation de mouvements proposée se base sur l'utilisation d'un mouvement tuteur issu de capture de mouvement. L'idée est de l'adapter à différents environnements de synthèse tout en conservant les détails fins de ce mouvement. C'est ainsi que nous sommes en mesure de produire des mouvements réalistes.

Les données de capture de mouvement présentent généralement le réalisme voulu pour servir de tuteur. Elles requièrent toutefois une adaptation afin d'être plausibles en fonction du modèle biomécanique du personnage et de l'environnement de synthèse. En effet, plusieurs facteurs font qu'une capture de mouvement ne produise pas un mouvement plausible dans notre environnement de synthèse, dont les différences entre le modèle du personnage et le sujet ayant effectué le mouvement, les différences entre l'environnement de synthèse et l'environnement de capture et le bruit d'acquisition. Ce problème sera abordé à la section 5.2.2.

Un autre problème associé aux données de capture de mouvement est qu'elles s'appliquent généralement à un personnage articulé ayant un très grand nombre de coordonnées articulaires. Afin de réduire la complexité des problèmes d'optimisation subséquents, il convient donc de réduire le nombre de coordonnées articulaires sans toutefois altérer le mouvement significativement. Avec certaines adaptations, ce problème peut aussi être posé comme un problème d'optimisation de mouvement.

Le processus de production d'un mouvement tuteur à partir de données de capture de mouvement est illustré à la figure 5.8.

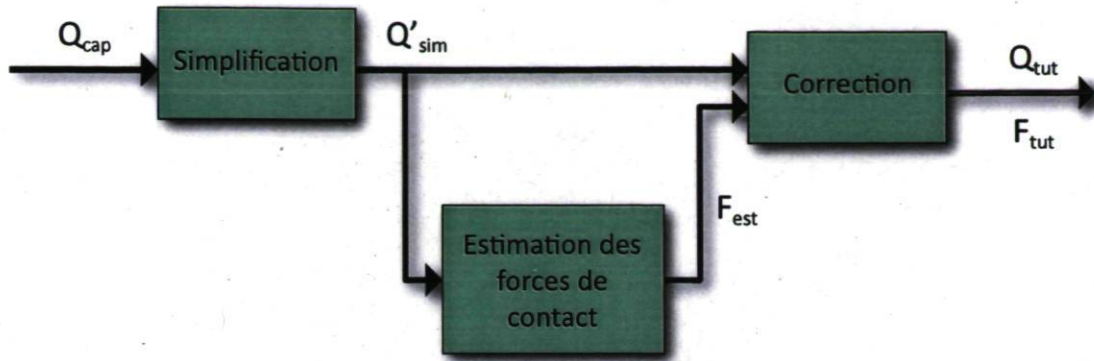


FIGURE 5.8 – Processus de création d'un mouvement tuteur à partir de données de capture de mouvement. Le mouvement capturé Q_{cap} subit une étape de simplification afin de correspondre aux coordonnées articulaires du personnage de synthèse. Les forces de contact du mouvement résultant, soit Q'_{sim} , sont ensuite estimées par la méthode décrite au chapitre 4. Finalement, une correction est appliquée au mouvement simplifié Q'_{sim} et aux forces estimées F_{est} pour produire le tuteur. Le mouvement dynamique défini par Q_{tut} et F_{tut} est un mouvement similaire au mouvement capturé, mais il est plausible pour le personnage de synthèse dans l'environnement de synthèse.

5.2.1 Simplification

La première étape de notre approche consiste à réduire le nombre de coordonnées articulaires impliquées dans le mouvement. Pour ce faire, nous devons d'abord définir deux modèles biomécaniques : le personnage de capture et le personnage de synthèse.

Le personnage de capture est un modèle biomécanique dont le graphe des connexions est identique au modèle articulé utilisé dans les données de capture de mouvement. Ce type de modèle aura typiquement 3 DDLs de rotation à chaque articulation. Les dimensions du personnage sont obtenues à partir des données de capture de mouvement. Les masses des différents membres du personnage sont estimées à l'aide de la table de Winter[87]. Nous considérons deux points de contact par pied, un premier au talon et un second à la pointe du pied. Le personnage de capture a $n + n_{sup}$ coordonnées articulaires.

Le personnage de synthèse est visuellement identique au personnage de capture. Ils ont les mêmes membres et ceux-ci sont de mêmes dimensions et de même masse. Les articulations du personnage de synthèse sont toutefois simplifiées par rapport à celles du personnage de capture. Leurs coordonnées articulaires sont limitées pour mieux correspondre à un personnage humain et, par le fait même, pour simplifier le problème d'optimisation. À titre d'exemple, le genou d'un personnage de capture aura 3 DDLs

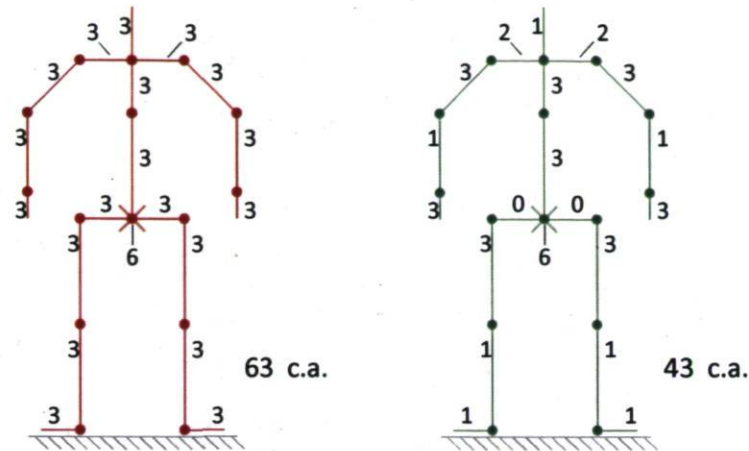


FIGURE 5.9 – Illustration d’un personnage de capture typique (à gauche) et du personnage de synthèse qui lui serait associé (à droite). Pour les deux modèles, la figure donne le nombre de coordonnées articulaires de chaque corps par rapport à son corps parent. Le personnage de capture a 6 DDLs à la racine et 3 DDLs de rotation à chaque articulation, pour un total de 63 coordonnées articulaires. Sur le personnage de synthèse, les articulations sont limitées aux coordonnées articulaires d’un être humain véritable. Les coordonnées articulaires qui ne contribuent pas significativement aux mouvements désirés sont aussi éliminés sur le personnage de synthèse.

de rotation alors que celui du personnage de synthèse en aura un seul. Le personnage de synthèse a n coordonnées articulaires. La figure 5.9 illustre la différence entre les coordonnées articulaires des deux modèles.

Puisque le mouvement capturé \mathbf{Q}_{cap} est issu d’un acteur humain, les coordonnées articulaires superflues du personnage de capture ne comportent que très peu de variation dans ce mouvement. C’est ce qui nous permet de les ignorer sans affecter le mouvement de façon significative. Soulignons également que le personnage de synthèse est construit de façon à faciliter cette procédure, notamment par l’orientation des divers membres du modèle selon les axes naturels des articulations. À titre d’exemple, l’axe z du repère local d’un tibia est aligné avec l’axe de rotation du genou.

Définissons l’ensemble Ω_{sup} comme étant l’ensemble des indices des coordonnées articulaires superflues, soient les coordonnées articulaires qui ne sont pas représentées dans le personnage de synthèse. Le problème de la simplification du personnage consiste donc à rendre constantes les valeurs associées aux coordonnées articulaires dont les indices forment Ω_{sup} dans le mouvement capturé \mathbf{Q}_{cap} . Ce problème peut être posé comme un problème d’optimisation de mouvement.

Ce problème ne correspond pas exactement à celui posé dans l’équation 5.29 puisque,

Réduction du nombre de coordonnées articulaires	
Personnage	Personnage de capture
Mouvement directeur	\mathbf{Q}_{cap}
Environnement	–
Coordonnées bloquées	$\Omega_{lock} = \Omega_{sup},$ $\kappa_i = 0, \forall i \in \Omega_{lock}$
Valeur initiale	$\mathbf{Q}_{init} = \mathbf{Q}_{cap},$ $\mathbf{F}_{init} = -$
Valeur optimale	\mathbf{Q}_{sim}

TABLE 5.2 – Caractérisation du problème de réduction du nombre de degrés de liberté. Puisqu'ils s'agit d'un problème simplifié, certains paramètres ne sont pas requis.

contrairement aux problèmes décrits plus loin, il ne demande pas de produire un mouvement plausible. En fait, nous simplifions le problème au maximum puisqu'il opère sur un grand nombre de coordonnées articulaires et risquerait de ne pas converger autrement. Ainsi, la seule contrainte que le mouvement doit respecter est celle des coordonnées articulaires bloquées. Nous avons donc le problème suivant :

$$\min_{\mathbf{x}} F(\mathbf{x}) \mid C_{lock,i,t}(\mathbf{x}) = 0, \quad \forall i \in \Omega_{sup}, \forall t \in \{1, 2, \dots, T\} \quad (5.30)$$

avec $\kappa_i = 0$ pour toutes les valeurs de i . En pratique, les valeurs des forces de contact sont omises du vecteur \mathbf{x} puisqu'elles n'interviennent pas dans la fonction de coût ou dans la contrainte.

Les paramètres de ce problème sont les suivants. L'optimisation se fait à l'aide du personnage de capture. Le mouvement directeur correspond aux données de capture de mouvement \mathbf{Q}_{cap} . Les coordonnées articulaires bloquées sont données par l'ensemble Ω_{sup} et elles sont fixées à $\kappa_i = 0$.

La valeur initiale du mouvement est \mathbf{Q}_{cap} et l'optimum du problème est \mathbf{Q}_{sim} , un mouvement similaire à \mathbf{Q}_{cap} mais pour lequel les lignes correspondant aux coordonnées articulaires superflues sont nulles. Afin de pouvoir appliquer ce mouvement au personnage de synthèse, nous devons enlever ces lignes de la matrice \mathbf{Q}_{sim} . Nous obtenons alors la matrice \mathbf{Q}'_{sim} qui est de dimension $n \times T$. Le tableau 5.2 résume la caractérisation de ce problème.

5.2.2 Correction

Une fois que le mouvement capturé peut être appliqué au personnage de synthèse, nous pouvons nous attarder à en faire un mouvement plausible. Ce mouvement pourra ensuite servir de tuteur pour l'extrapolation de mouvements subséquente. Nous pouvons poser ce problème sous la forme de l'optimisation de mouvement décrite ci-dessous.

Ce problème utilise le personnage de synthèse et son mouvement directeur est donné par la matrice \mathbf{Q}'_{sim} , soit le mouvement capturé adapté au personnage de synthèse à l'étape précédente. À ce stade, l'environnement de synthèse doit être le plus similaire possible à l'environnement de capture. En supposant que le sol de l'environnement de capture soit horizontal et proche du niveau de la mer, nous utilisons le vecteur d'accélération gravitationnel suivant : $\vec{g} = (0, 0, -9, 81)^t$. Nous fixons le coefficient de frottement statique à $\mu = 1$, ce qui approxime bien le frottement entre un sol de béton et des semelles de caoutchouc. Aucune coordonnée articulaire n'est bloquée pour ce problème.

La valeur initiale utilisée pour résoudre ce problème est composée du mouvement directeur, soit la matrice \mathbf{Q}'_{sim} , ainsi que les forces de contact estimées pour ce mouvement par la méthode que nous avons présenté au chapitre 4.

Les paramètres de ce problème cherchent à reproduire le plus fidèlement possible les conditions présentes lors de la séance de capture de mouvement. De fait, l'optimum du problème devrait être un mouvement similaire aux données de capture de mouvement qui soit plausible pour le personnage de synthèse sous l'environnement donné. Ce mouvement corrigé sera notre mouvement tuteur, que nous identifierons par \mathbf{Q}_{tut} . Les forces de contact issues de ce mouvement sont données par la matrice \mathbf{F}_{tut} . Rappelons que ces deux éléments sont contenus dans la solution \mathbf{x} retenue lors de l'optimisation du mouvement. Le tableau 5.3 résume la caractérisation de ce problème.

5.3 Synthèse de nouveaux mouvements

Tel que mentionné plus tôt, l'approche d'extrapolation de mouvements que nous proposons est basée sur une optimisation de mouvement incrémentale. Cette optimisation incrémentale s'appuie fortement sur le principe d'homotopie entre deux fonctions. Il convient donc de définir ce principe avant de poursuivre.

Correction des données capturées	
Personnage	Personnage de synthèse
Mouvement directeur	Q'_{sim}
Environnement	$\vec{g} = (0, 0, -9, 81)^t$, $\mu = 1$
Coordonnées bloquées	$\Omega_{lock} = \emptyset$
Valeur initiale	$Q_{init} = Q'_{sim}$, $F_{init} = EFC(Q'_{sim})$
Valeur optimale	Q_{tut} , F_{tut}

TABLE 5.3 – Caractérisation du problème de correction de données issues de capture de mouvement. La valeur initiale des forces de contacts est donnée par notre approche d'estimation des forces de contact, identifiée comme la fonction $EFC(Q'_{sim})$.

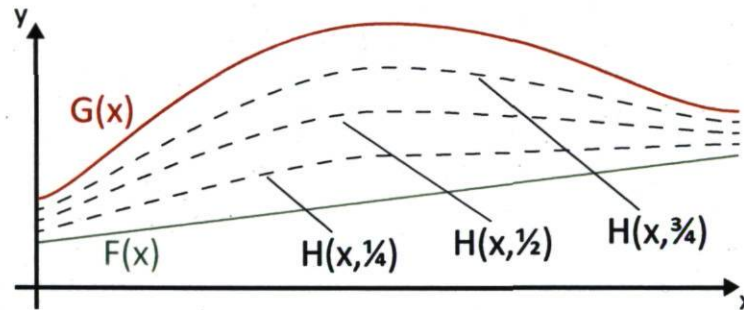


FIGURE 5.10 – Exemple d'une homotopie H définie entre deux fonctions homotopes F et G . Les courbes de niveau de H sont tracées en pointillé.

Deux fonctions sont dites homotopes si une déformation continue, une homotopie, permet de passer de l'une à l'autre. Formellement, une homotopie entre deux fonctions continues $F : X \rightarrow Y$ et $G : X \rightarrow Y$ est définie comme une fonction continue $H : X \times [0, 1] \rightarrow Y$ telle que $H(x, 0) = F(x)$ et $H(x, 1) = G(x)$. Les fonctions F et G sont alors dites *homotopes* d'homotopie H . L'homotopie la plus simple à définir est probablement $H(x, \lambda) = \lambda G(x) + (1 - \lambda)F(x)$. La figure 5.10 illustre un exemple d'homotopie.

Les méthodes de continuation utilisent les homotopies pour résoudre des systèmes d'équations non linéaires. Sommairement, l'idée derrière les méthodes de continuation est relativement simple. On commence par définir un système simplifié qui soit homotope avec le système à résoudre. Dans la mesure où l'on peut résoudre le système simplifié, on peut partir de cette solution et la faire évoluer, en modifiant le système selon l'homotopie, jusqu'à ce que l'on obtienne une solution au système initial. Les méthodes de

continuation sont présentées avec plus de détails dans [60].

L'optimisation de mouvement incrémentale que nous présentons est fondée sur une idée très similaire à celle des méthodes de continuation, soit de partir d'un problème d'optimisation simple et de modifier ce problème graduellement jusqu'à ce que la solution d'un problème d'optimisation de mouvement plus complexe soit obtenue. À la différence des méthodes de continuation, nous opérons sur des problèmes de programmation non linéaires et non sur des systèmes d'équations non linéaires⁶. Soulignons que dans notre approche, le problème d'optimisation *simple* n'est pas intrinsèquement moins complexe qu'un autre, mais il est plus facile à résoudre puisque nous sommes en mesure de fournir une solution initiale plus proche de son optimum.

Nous dirons que deux problèmes d'optimisation de mouvement sont homotopes si leur fonction de coût et leurs fonctions de contraintes sont toutes homotopes. Rappelons qu'un problème d'optimisation de mouvement diffère d'un autre selon les valeurs que prennent leurs divers paramètres (tableau 5.1). Si la différence entre deux problèmes est entièrement attribuée à des paramètres continus, tels que le coefficient de frottement par exemple, alors ces problèmes sont homotopes. En effet, toutes les fonctions qui définissent le problème d'optimisation de mouvement varient de façon continue avec les paramètres continus. Ce n'est toutefois pas le cas pour les paramètres discrets, tels que les matrices de temps de contact.

À titre d'exemple, faisons varier le coefficient de frottement de μ_1 à μ_2 entre deux problèmes. Nous pouvons décrire l'homotopie suivante entre les fonctions de contraintes du cône de frottement de ces deux problèmes :

$$H_{\text{cône},i,t}(\mathbf{x}, \lambda) \stackrel{\text{def}}{=} ((1 - \lambda)\mu_1 + \lambda\mu_2) f_{fz,i,t_\perp} - \left\| \vec{f}_{\text{ext},i,t_\parallel} \right\|.$$

L'exemple que nous présentons ici est simple puisque le coefficient de frottement n'affecte qu'une seule contrainte, mais le principe est essentiellement le même pour l'ensemble des paramètres continus sur l'ensemble des fonctions qui définissent le problème.

Savoir que deux problèmes d'optimisation de mouvement sont homotopes nous permet de passer progressivement de la solution de l'un à la solution de l'autre. Dans le cas général, lorsque le facteur λ passe de 0 à 1, la fonction de coût se déforme progressivement et son minimum se déplace graduellement. D'une façon analogue, les frontières formées par les contraintes d'égalité et d'inégalité se déplacent et se déforment. Ce faisant, l'optimum du problème se déplace progressivement lui aussi.

6. Si notre problème d'optimisation ne comportait que des contraintes d'égalité, nous pourrions utiliser l'approche des multiplicateurs de Lagrange pour le reformuler comme un système d'équations non linéaires.

Toutefois, il peut arriver que les contraintes se referment autour de l'optimum et amènent l'optimisation incrémentale dans une impasse : même si une solution existe pour cette situation, elle est trop loin de la dernière solution connue pour être trouvée efficacement. Nous avons rencontré une telle impasse dans le cadre de nos expérimentations alors que nous tentions d'extrapoler un mouvement de marche sur un plan incliné à plus de 40° , où un sujet humain aurait probablement modifié la stratégie de son mouvement. Il s'agit d'une des limites fondamentale de cette approche.

De plus, il est important de noter que certains problèmes ne possèdent pas de solution même s'ils sont homotopes à un problème pour lequel une solution existe. À titre d'exemple, le personnage ne sera pas en mesure de marcher si le vecteur d'accélération gravitationnelle est inversé, même si ce problème est homotope à celui d'une marche dans des conditions plus normales.

5.3.1 Problème de base

Peu importe le mouvement que nous désirons extrapoler à partir du mouvement tuteur, la version simplifiée du problème d'optimisation de mouvement réfère toujours au même problème. Nous le désignerons *problème de base*.

Le problème de base est un problème trivial puisque sa solution, comme sa valeur initiale, est le mouvement tuteur. Les paramètres du problème de base sont presque identiques à ceux du problème de correction des données de capture de mouvement (tableau 5.3), à l'exception que le mouvement directeur est donné par \mathbf{Q}_{tut} . Le tableau 5.4 définit ce problème. Nous utilisons \mathbf{Q}_0 et \mathbf{F}_0 pour désigner sa solution. L'indice 0 est utilisé puisque ce problème correspond à une valeur de $\lambda = 0$ pour l'homotopie que nous définirons pour extrapoler un mouvement donné.

5.3.2 Optimisation incrémentale

Pour mieux expliquer comment nous utilisons une homotopie afin d'extrapoler un mouvement, nous allons en présenter un exemple en parallèle. Le problème utilisé en exemple sera l'extrapolation d'un mouvement de marche suivant un arc de cercle de $\pi/2$ radians à partir d'un mouvement de marche en ligne droite.

Il nous faut d'abord définir les deux problèmes d'optimisation de mouvement homotopes. Nous parlerons du problème de base et du problème *visé*. Le problème de

Problème de base	
Personnage	Personnage de synthèse
Mouvement directeur	\mathbf{Q}_{tut}
Environnement	$\vec{g} = (0, 0, -9, 81)^t,$ $\mu = 1$
Coordonnées bloquées	$\Omega_{lock} = \emptyset$
Valeur initiale	$\mathbf{Q}_{init} = \mathbf{Q}_{tut},$ $\mathbf{F}_{init} = \mathbf{F}_{tut}$
Valeur optimale	$\mathbf{Q}_0 (= \mathbf{Q}_{tut}),$ $\mathbf{F}_0 (= \mathbf{F}_{tut})$

TABLE 5.4 – Caractérisation de la version simplifiée du problème d'extrapolation de mouvement.

base est donné dans la section précédente. Le problème *visé* est celui dont la solution correspond au mouvement dont on désire faire la synthèse. Dans cet exemple, il s'agit du problème produisant une marche dont la trajectoire décrit un arc de cercle de $\pi/2$ radians.

Dans cet exemple, le mouvement directeur est le seul paramètre qui diffère entre le problème de base et le problème visé. Pour l'instant, supposons l'existence d'une fonction $\Phi(\mathbf{Q}, \theta)$ qui prend un mouvement de trajectoire rectiligne \mathbf{Q} et qui le transforme pour que sa trajectoire décrive un arc de cercle de θ radians. Le mouvement directeur du problème visé est alors $\Phi(\mathbf{Q}_{tut}, \pi/2)$. Ce problème est décrit dans le tableau 5.5. La fonction $\Phi(\mathbf{Q}, \theta)$ sera développée dans la section 5.4.

La valeur initiale donnée dans le tableau 5.5 est composée du mouvement transformé $\Phi(\mathbf{Q}_{tut}, \pi/2)$ et de notre estimation des forces de contact sur ce mouvement. Cette valeur initiale n'est généralement pas suffisamment proche de l'optimum pour en faire un problème simple étant donné que la transformation appliquée produit de nombreux artéfacts dans le mouvement, tels que d'importants glissements de pieds. Ces artéfacts placent la valeur initiale loin du respect des contraintes⁷.

C'est ici qu'intervient l'idée d'une optimisation incrémentale. En posant une multitude de problèmes intermédiaires entre le problème de base et le problème visé, nous sommes en mesure d'utiliser la solution d'une itération comme valeur initiale pour l'itération suivante. Si les incréments sont suffisamment petits, chaque problème inter-

7. Même s'il n'est pas plausible, le mouvement transformé peut servir de mouvement directeur puisqu'il conserve les caractéristiques globales et détaillées du mouvement à synthétiser.

Problème visé - Marche courbe	
Personnage	Personnage de synthèse
Mouvement directeur	$\Phi(\mathbf{Q}_{tut}, \pi/2)$
Environnement	$\vec{g} = (0, 0, -9, 81)^t$, $\mu = 1$
Coordonnées bloquées	$\Omega_{lock} = \emptyset$
Valeur initiale	$\mathbf{Q}_{init} = \Phi(\mathbf{Q}_{tut}, \pi/2)$, $\mathbf{F}_{init} = EFC(\Phi(\mathbf{Q}_{tut}, \pi/2))$
Valeur optimale	\mathbf{Q}_1 , \mathbf{F}_1

TABLE 5.5 – Caractérisation du problème de synthèse d'une marche courbe.

médiaire devient suffisamment simple pour être résolu efficacement.

C'est l'homotopie entre le problème de base et le problème visé qui nous permet de créer un problème intermédiaire. Chaque valeur de λ comprise dans l'intervalle $[0, 1]$ produit un problème distinct. $\lambda = 0$ correspond au problème de base et $\lambda = 1$ correspond au problème visé. Les autres valeurs correspondent à des problèmes intermédiaires. Pour un ensemble ordonné $\Lambda = \{0, \lambda_1, \lambda_2, \dots, \lambda_{m-1}, 1\}$ de valeurs de λ , notre approche d'optimisation de mouvement incrémentale consiste à résoudre en ordre croissant les problèmes intermédiaires correspondant à ces valeurs de λ . Ainsi, le problème lié à λ_i utilise la solution du problème précédent, $\mathbf{Q}_{\lambda_{i-1}}$ et $\mathbf{F}_{\lambda_{i-1}}$, comme valeur initiale.

Pour l'exemple de la marche courbe, un problème intermédiaire aura $\Phi(\mathbf{Q}_{tut}, \lambda_i \pi/2)$ comme mouvement directeur. Ce problème est défini dans le tableau 5.6.

Un autre bon exemple est l'extrapolation d'une marche sur un plan incliné à $\pi/16$ radians à partir d'une marche sur un sol horizontal. Dans cet exemple, le mouvement directeur et l'environnement sont modifiés dans le problème visé. Puisque nous définissons toujours le sol comme étant le plan $z = 0$, nous simulons l'inclinaison du sol par une rotation du vecteur d'accélération gravitationnelle. Si le personnage doit faire face à une pente de θ radians, nous appliquons une rotation de $-\theta$ radians au vecteur d'accélération. Cette rotation doit également être appliquée à la racine du personnage dans chaque configuration du mouvement directeur. Cette transformation du mouvement directeur est donnée par une fonction $\Psi(\mathbf{Q}, \theta)$, qui sera décrite en plus amples détails dans la section 5.4. La figure 5.11 illustre la correspondance entre l'inclinaison du sol et l'inclinaison du vecteur d'accélération gravitationnelle.

Problème intermédiaire (λ_i) - Marche courbe	
Personnage	Personnage de synthèse
Mouvement directeur	$\Phi(\mathbf{Q}_{tut}, \lambda_i \pi/2)$
Environnement	$\vec{g} = (0, 0, -9, 81)^t$, $\mu = 1$
Coordonnées bloquées	$\Omega_{lock} = \emptyset$
Valeur initiale	$\mathbf{Q}_{init} = \mathbf{Q}_{\lambda_{i-1}}$, $\mathbf{F}_{init} = \mathbf{F}_{\lambda_{i-1}}$
Valeur optimale	\mathbf{Q}_{λ_i} , \mathbf{F}_{λ_i}

TABLE 5.6 – Caractérisation du problème intermédiaire lié à λ_i pour l'extrapolation d'une marche courbe.

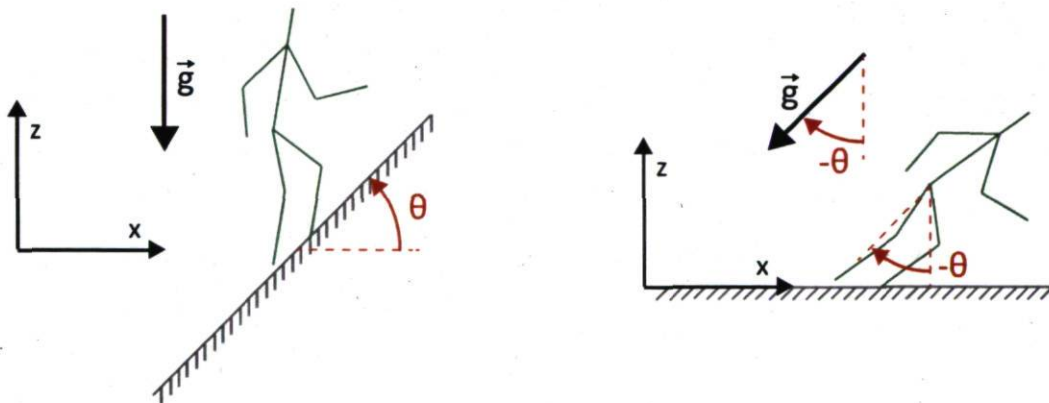


FIGURE 5.11 – À gauche, on illustre un personnage sur un sol incliné à θ radians. Cette situation correspond à l'illustration de droite, où le sol est le plan $z = 0$ et où le vecteur d'accélération gravitationnelle et la configuration du personnage ont subi une rotation de $-\theta$ radians.

Problème visé - Plan incliné	
Personnage	Personnage de synthèse
Mouvement directeur	$\Psi(\mathbf{Q}_{tut}, -\frac{\pi}{16})$
Environnement	$\vec{g} = R_y(-\frac{\pi}{16})(0, 0, -9, 81)^t,$ $\mu = 1$
Coordonnées bloquées	$\Omega_{lock} = \emptyset$
Valeur initiale	$\mathbf{Q}_{init} = \Psi(\mathbf{Q}_{tut}, -\frac{\pi}{16}),$ $\mathbf{F}_{init} = EFC(\Psi(\mathbf{Q}_{tut}, -\frac{\pi}{16}))$
Valeur optimale	$\mathbf{Q}_1,$ \mathbf{F}_1

TABLE 5.7 – Caractérisation du problème de synthèse d'une marche sur un plan incliné.

Pour ce nouvel exemple, le problème visé est décrit dans le tableau 5.7 et les problèmes intermédiaires auront la forme décrite dans le tableau 5.8. Sans perte de généralité, nous présumons que la marche se fait dans la direction de l'axe des x du repère inertiel, la rotation appliquée au vecteur d'accélération gravitationnelle et à la racine du personnage est donc une rotation autour de l'axe y du repère inertiel.

Progression des valeurs de λ

L'ensemble $\Lambda = \{\lambda_0, \lambda_1, \dots, \lambda_m\}$ représente la progression de la valeur du λ à chaque itération de l'optimisation incrémentale. Cette progression n'est pas fixée a priori, elle est plutôt construite au cours du processus d'optimisation. Cela nous permet d'adapter l'incrément en fonction de la complexité d'un problème intermédiaire. Toutefois nous avons toujours $\lambda_0 = 0$ et $\lambda_m = 1$.

Le terme *incrément* désigne ici la différence entre deux valeurs consécutives de λ . Ainsi l'incrément Δ_i est donné par l'équation suivante :

$$\Delta_i = \lambda_i - \lambda_{i-1}. \quad (5.31)$$

L'algorithme que nous utilisons pour fixer ces incréments est relativement simple. Il consiste essentiellement à diviser par deux la valeur d'un incrément tant que le problème intermédiaire correspondant à cet incrément ne peut être résolu en moins de k itérations de l'algorithme d'optimisation (le choix de k sera discuté plus loin). Suite à la résolution d'un problème intermédiaire, le double de l'incrément retenu est utilisé comme valeur initiale du prochain incrément. Malgré sa simplicité, cet algorithme

Problème intermédiaire (λ_i) - Plan incliné	
Personnage	Personnage de synthèse
Mouvement directeur	$\Psi(\mathbf{Q}_{tut}, -\frac{\lambda_i \pi}{16})$
Environnement	$\vec{g} = R_y(-\frac{\lambda_i \pi}{16})(0, 0, -9, 81)^t,$ $\mu = 1$
Coordonnées bloquées	$\Omega_{lock} = \emptyset$
Valeur initiale	$\mathbf{Q}_{init} = \mathbf{Q}_{\lambda_{i-1}},$ $\mathbf{F}_{init} = \mathbf{F}_{\lambda_{i-1}}$
Valeur optimale	$\mathbf{Q}_{\lambda_i},$ \mathbf{F}_{λ_i}

TABLE 5.8 – Caractérisation du problème intermédiaire lié à λ_i pour l'extrapolation d'une marche sur un plan incliné.

permet néanmoins d'adapter la taille de l'incrément à la complexité du problème, en procédant par plus petits incréments lorsqu'un problème intermédiaire est complexe et par plus gros incréments lorsque le problème intermédiaire est suffisamment simple.

Programmation non linéaire

Puisqu'elle consiste à résoudre une séquence de problèmes de programmation non linéaires, notre approche d'optimisation incrémentale de mouvement fait appel à un algorithme d'optimisation non linéaire. Pour être suffisamment rapide dans un espace à grande dimensionnalité, l'algorithme d'optimisation choisi doit généralement connaître les dérivées partielles du problème (équation 5.29). Du point de vue de l'implémentation, cela signifie que nous devons être en mesure de donner les dérivées partielles analytiques de la fonction de coût (section 5.1.1) et de toutes les fonctions de contrainte (sections 5.1.2, 5.1.3 et 5.1.4) par rapport aux éléments du vecteur \mathbf{x} . Outre la longueur des équations qui en résulte, dériver ces fonctions ne pose aucune difficulté majeure. Les dérivées partielles ne sont donc pas incluses dans ce document.

Ce type d'algorithme d'optimisation est généralement itératif : une solution initiale est raffinée graduellement jusqu'à l'atteinte d'un optimum, possiblement un minimum local. Afin de se prémunir contre un problème où l'atteinte d'un tel optimum est impossible, ou simplement trop difficile, ces algorithmes utilisent généralement un paramètre indiquant le nombre maximum d'itérations qu'ils peuvent faire pour tenter de résoudre le problème. S'il réussit à converger vers un optimum avant d'avoir atteint le nombre

maximum d'itérations, l'algorithme retourne la solution. Autrement, il doit indiquer que l'optimum n'a pas été atteint.

Nous désignerons par $optnonlin(P, \mathbf{x}_{init}, k)$ l'algorithme d'optimisation non linéaire utilisé, où P est le problème de programmation non linéaire, \mathbf{x}_{init} est la valeur initiale du problème et k est le nombre maximum d'itérations de l'algorithme. Par abus de notation, nous dirons que $optnonlin(P, \mathbf{x}_{init}, k)$ retourne l'ensemble vide lorsqu'il ne réussit pas à converger vers un optimum avant d'avoir atteint k d'itérations. Soulignons que la valeur de k peut varier fortement selon l'algorithme d'optimisation utilisé. Nous utilisons généralement la valeur de k définie par défaut dans l'algorithme utilisé puisque celui-ci est généralement un bon indicateur que problème est facilement soluble ou qu'il demande à être simplifié.

Optimisation incrémentale de mouvement

Dans sa plus simple expression, notre algorithme d'extrapolation de mouvement par optimisation incrémentale se décrit comme suit :

```

Entrées :  $\mathbf{x}_{tut}, H, \Delta_0, k$ 
Sorties :  $\mathbf{x}_1$ 
 $\lambda_0 = 0$ 
 $\mathbf{x}_0 = \mathbf{x}_{tut}$ 
 $\Delta_1 = \Delta_0$ 
 $i = 1$ 
tant que  $\lambda_{i-1} < 1$  faire
  |  $\lambda_i = \lambda_{i-1} + \Delta_i$ 
  |  $P_{\lambda_i} = H(\lambda_i)$ 
  |  $\mathbf{x}_{\lambda_i} = optnonlin(P_{\lambda_i}, \mathbf{x}_{\lambda_{i-1}}, k)$ 
  | si  $\mathbf{x}_{\lambda_i} \neq \emptyset$  alors
  | |  $\Delta_{i+1} = 2\Delta_i$ 
  | |  $i = i + 1$ 
  | sinon
  | |  $\Delta_i = \Delta_i/2$ 
  | fin
fin
retourner  $\mathbf{x}_1$ 

```

Algorithme 2: Algorithme d'extrapolation de mouvement - version simplifiée

Cet algorithme prend en entrée le mouvement tuteur \mathbf{x}_{tut} , l'homotopie H , une valeur

d'incrément initiale Δ_0 et un nombre maximum d'itérations par problème intermédiaire k . L'homotopie H permet de produire le problème intermédiaire correspondant à λ_i par l'équation $P_{\lambda_i} = H(\lambda_i)$. Soulignons que nous utilisons le vecteur \mathbf{x} et non ses composantes \mathbf{Q} et \mathbf{F} pour désigner un mouvement. Ceci nous permet d'exprimer l'algorithme sous une forme plus générale, sans égard au fait que notre implémentation est basée sur la quantité de mouvement globale du personnage et ne requiert donc pas l'inclusion des forces généralisées dans le vecteur \mathbf{x} .

L'algorithme 2 n'est toutefois pas complet. Sous cette forme, si le processus d'optimisation n'arrive pas à résoudre le problème visé, la valeur de Δ_i tendra éventuellement vers 0 et l'algorithme sera coincé dans une boucle infinie. Pour s'assurer que l'algorithme retourne toujours une valeur, soit la solution du dernier problème résolu, il faut poser une borne inférieure ϵ sur la valeur des l'incrément. Nous devons alors ajouter la valeur de λ du dernier problème résolu aux variables de sortie, afin d'indiquer si la solution retournée est celle du problème visé ou celle d'un problème intermédiaire.

De plus, nous devons ajouter un mécanisme supplémentaire pour nous assurer que les valeurs de λ ne dépassent jamais 1. La forme finale de notre algorithme d'extrapolation de mouvement par optimisation incrémentale devient donc :


```

Entrées :  $\mathbf{x}_{tut}$ ,  $H$ ,  $\Delta_0$ ,  $k$ ,  $\epsilon$ 
Sorties :  $\mathbf{x}_\lambda$ ,  $\lambda$ 
 $\lambda_0 = 0$ 
 $\mathbf{x}_0 = \mathbf{x}_{tut}$ 
 $\Delta_1 = \Delta_0$ 
 $i = 1$ 
tant que  $\lambda_{i-1} < 1$  faire
  |  $\lambda_i = \lambda_{i-1} + \Delta_i$ 
  |  $P_{\lambda_i} = H(\lambda_i)$ 
  |  $\mathbf{x}_{\lambda_i} = \text{optnonlin}(P_{\lambda_i}, \mathbf{x}_{\lambda_{i-1}}, k)$ 
  | si  $\mathbf{x}_{\lambda_i} \neq \emptyset$  alors
  | |  $\Delta_{i+1} = 2\Delta_i$ 
  | | si  $\lambda_i + \Delta_{i+1} > 1$  alors
  | | |  $\Delta_{i+1} = 1 - \lambda_i$ 
  | | fin
  | |  $i = i + 1$ 
  | sinon
  | |  $\Delta_i = \Delta_i/2$ 
  | | si  $\Delta_i < \epsilon$  alors
  | | | retourner  $[\mathbf{x}_{\lambda_{i-1}}, \lambda_{i-1}]$ 
  | | fin
  | fin
fin
retourner  $[\mathbf{x}_1, 1]$ 

```

Algorithme 3: Algorithme d'extrapolation de mouvement

Les paramètres Δ_0 et ϵ sont fixés empiriquement et varient d'un problème d'extrapolation à l'autre. La valeur de k est également fixée empiriquement, mais elle dépend uniquement de l'algorithme d'optimisation utilisé. À titre d'exemple, un problème *suffisamment* simple peut généralement être résolu par l'algorithme SNOPT [24] en moins de 10,000 itérations.

5.4 Résultats

Dans cette section, nous présentons l'étude de cas de l'extrapolation de divers mouvements à partir d'un mouvement de marche rectiligne. Nous abordons d'abord la création du mouvement tuteur pour ensuite nous attarder à l'extrapolation de trois mouvements distincts : un cycle de marche sur une pente descendante, un cycle de marche sur une

pente ascendante et un cycle de marche suivant une trajectoire courbe.

Notre implémentation consiste en un ensemble de scripts Matlab[®] faisant appel à des bibliothèques dynamiques (DLL) pour effectuer les opérations qui seraient autrement trop coûteuses en temps d'exécution. Cela nous a permis une certaine souplesse lors de nos expérimentations tout en offrant un niveau de performance intéressant. Il ne s'agit toutefois pas d'une implémentation entièrement optimisée. D'ailleurs, elle n'est pas optimisée pour profiter pleinement du parallélisme offert sur les processeurs récents.

Pour résoudre les problèmes d'optimisation de mouvements (équations 5.29 et 5.30), nous utilisons SNOPT [24], un algorithme d'optimisation non linéaire qui implémente une approche de programmation quadratique séquentielle. Notre approche n'est cependant pas limitée à l'usage de SNOPT ; tout autre algorithme d'optimisation non linéaire pourrait être utilisé pour le remplacer. Il est toutefois avisé d'utiliser un algorithme capable de traiter des problèmes de grande dimensionnalité et de prendre avantage des matrices fortement éparses sous-tendues par ce type de problème d'optimisation.

Cette étude de cas a été menée sur un ordinateur muni d'un processeur Intel[®] Core i7 920. Ce type de processeur offre 8 *threads* de traitement parallèles. Notre implémentation utilise en moyenne environ 2 *threads* lors de nos expérimentations.

Tel que mentionné en introduction (section 1.3), l'implémentation actuelle de notre approche est quelque peu limitée par certains choix et, possiblement, par quelques erreurs de programmation qui sont toutefois difficiles à isoler étant donné la complexité du système. Ce faisant, le modèle de synthèse utilisé dans cette étude de cas est un modèle simplifié de personnage humain, représentant la moitié inférieure d'un garçon. Nous nommons ce personnage *HalfBoy*. Celui-ci est décrit à la figure 5.12.

Le mouvement initial utilisé est la séquence traitée au chapitre 4. Il s'agit d'une séquence de marche rectiligne sur un sol horizontal exécutée par un garçon de 23kg mesurant 1,20m. Les temps de contact ont été déterminés par l'observation du mouvement et de l'information fournie par les plateformes de force. Le mouvement original est échantillonné à une fréquence de 120Hz et comporte plusieurs cycles de marche. Nous l'avons sous-échantillonné à 15Hz et nous en avons conservé un seul cycle. Le mouvement initial utilisé comporte 15 trames, incluant les deux trames de *chevauchement* requises par la contrainte de mouvements cycliques (figure 5.3).

Le modèle de capture utilisé dans ce mouvement comporte 63 coordonnées articulaires et correspond au modèle illustré à gauche de la figure 5.9. En ne conservant que la moitié inférieure de ce modèle, afin de correspondre à *HalfBoy*, celui-ci comporte 30

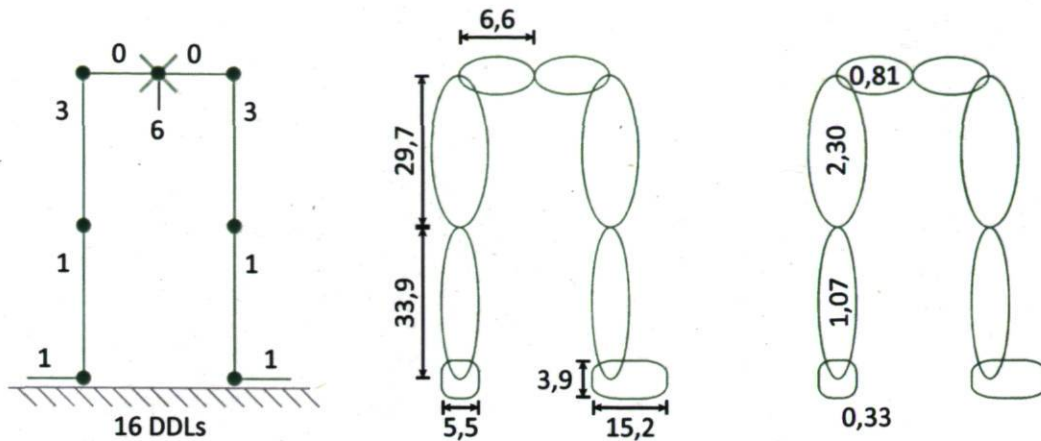


FIGURE 5.12 – Paramètres du personnage *HalfBoy*, qui représente la partie inférieure d'un garçon mesurant 1,20m et pesant 83kg. La figure de gauche indique le nombre de DDLs de chaque corps rigide par rapport à son corps parent. La figure du centre indique la longueur en cm de chaque segment telle que mesurée à partir des données de capture de mouvement. La figure de droite indique la masse de chaque segment en kg.

coordonnées articulaires. Pour cette étude de cas, le mouvement initial comporte donc 15 trames et 30 coordonnées articulaires. Il est illustré à l'aide de quelques trames à la figure 5.13. Les forces de contact de ce mouvement initial ont été estimées à l'aide de l'approche présentée au chapitre 4.

5.4.1 Création du mouvement tuteur

Avant de pouvoir procéder à l'extrapolation de mouvements, nous devons produire le mouvement tuteur. Ce dernier s'obtient en deux étapes : la simplification (5.30) et la correction (5.29).

Simplification

La simplification du mouvement vise à réduire le nombre de coordonnées articulaires actives dans le mouvement. Il s'agit d'un problème relativement simple, dans la mesure où les repères locaux sont orientés selon les axes naturels des articulations du sujet et dans la mesure où nous ne bloquons que les coordonnées articulaires superflues.

La simplification du mouvement initial consiste donc à passer du mouvement de marche sur le modèle de capture à 30 coordonnées articulaires (figure 5.14) et à le re-

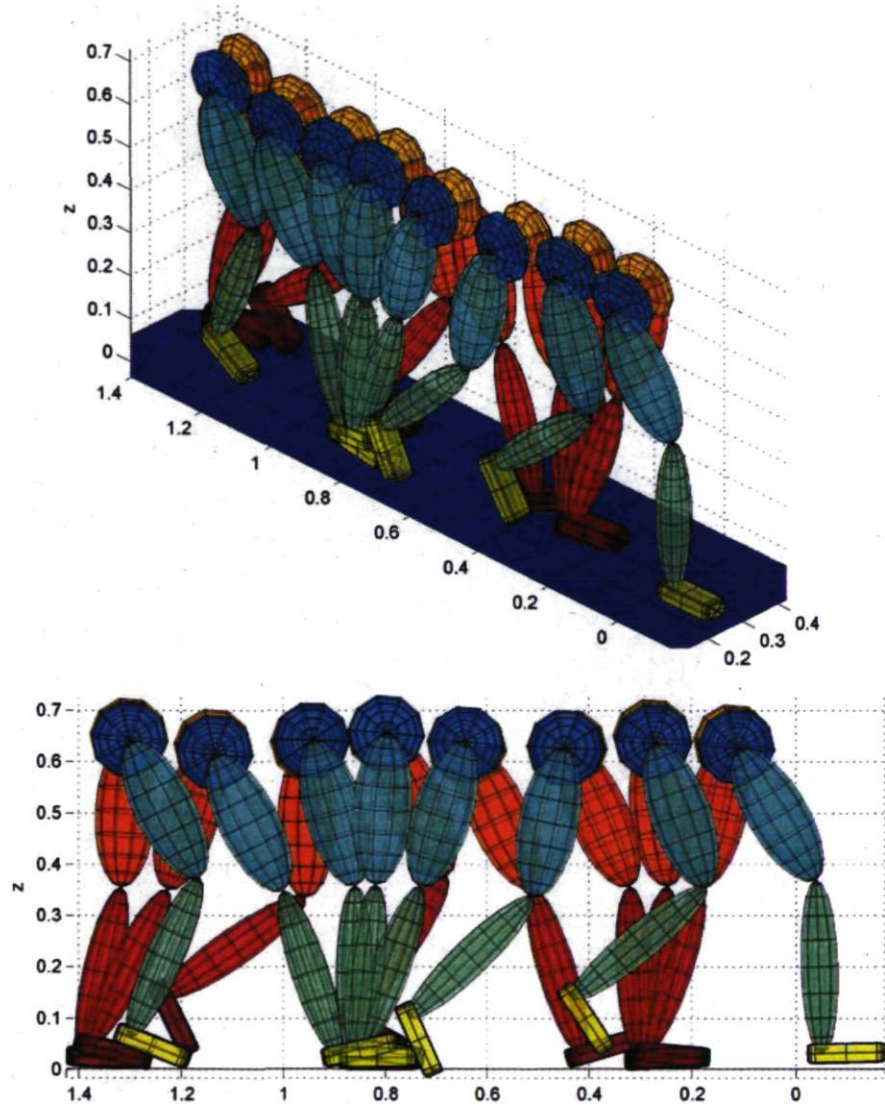


FIGURE 5.13 – Mouvement initial utilisé dans notre étude de cas.

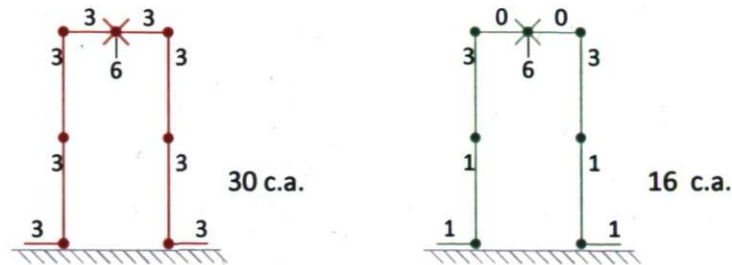


FIGURE 5.14 – Illustration du personnage de capture simplifié (à gauche) et du personnage de synthèse qui lui est associé, *HalfBoy* (à droite). Pour les deux modèles, la figure donne le nombre de DDLs de chaque corps par rapport à son corps parent. Le personnage de capture a 6 DDLs à la racine et 3 DDLs de rotation à chaque articulation, pour un total de 30 coordonnées articulaires. Sur *HalfBoy*, les articulations sont limitées aux coordonnées articulaires d'un être humain véritable et les coordonnées articulaires qui ne contribuent pas significativement aux mouvements désirés sont également éliminées.

produire sur *HalfBoy* (16 coordonnées articulaires). La figure 5.15 présente les résultats de cette simplification. Ces résultats ont nécessité 13,29 secondes de temps de calcul sur la plateforme de calcul décrite ci-dessus.

Tel que la figure 5.15 le montre, les configurations du mouvement simplifié sont presque identiques à celles du mouvement d'origine, à l'exception de la rotation de certains corps rigides autour de leur axe x .

Étant donné la simplicité de ce problème, nous sommes en mesure de le résoudre à l'aide de notre implémentation actuelle même pour des mouvements comportant beaucoup plus de trames et de coordonnées articulaires. À titre informatif, la figure 5.16 présente les résultats que nous avons obtenus pour la simplification d'un cycle de marche de 28 trames (30Hz), passant du personnage de capture entier à 63 coordonnées articulaires à un personnage de synthèse à 43 coordonnées articulaires (figure 5.9). Ces résultats ont nécessité 73 minutes de temps de calcul.

Correction

La correction d'un mouvement consiste à prendre le mouvement *simplifié* et à le rendre physiquement plausible. Ce problème d'optimisation de mouvement est sans conteste le plus complexe de notre approche. En effet, la simplification de mouvement pose un nombre beaucoup plus faible de contraintes et, comme nous le verrons plus loin, la synthèse de nouveaux mouvements s'opère par la résolution d'une multitude de sous problèmes relativement simples.

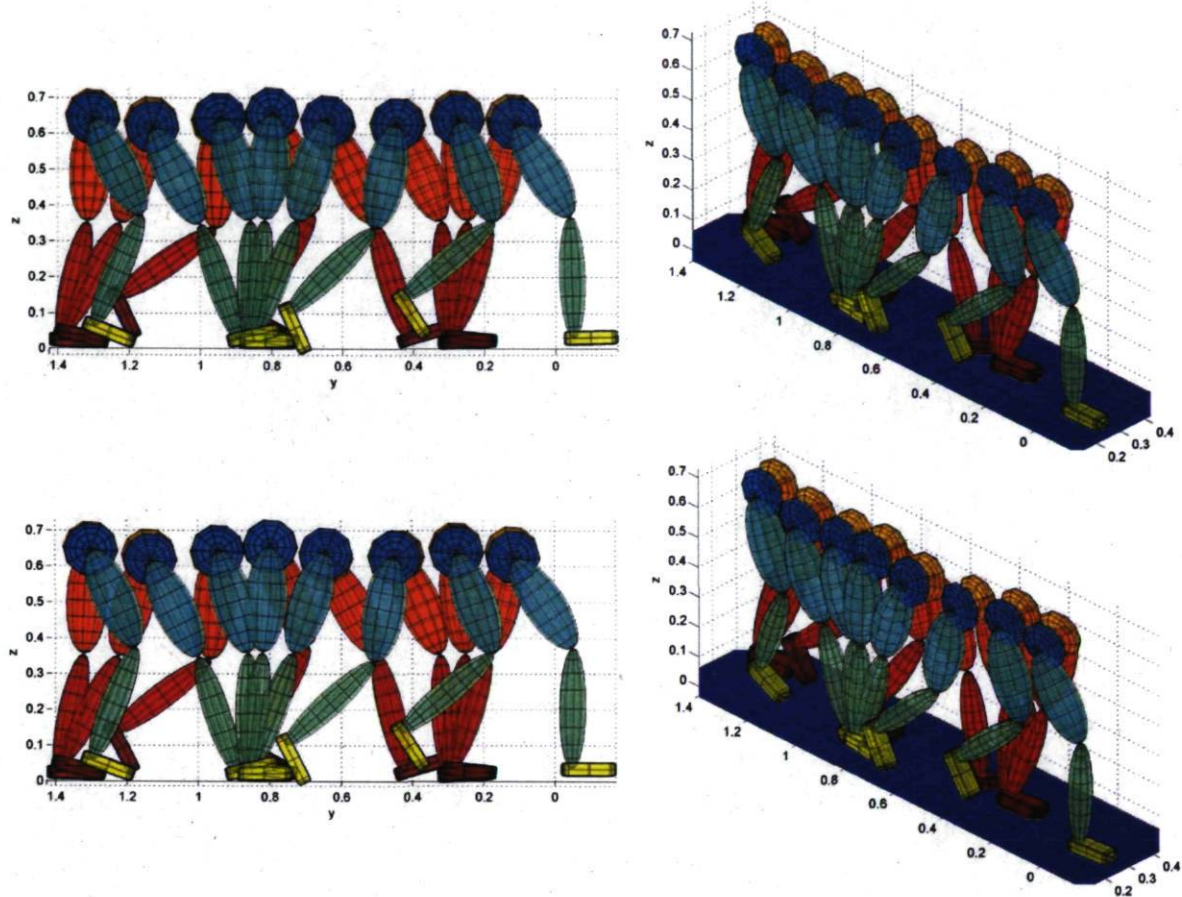


FIGURE 5.15 – Cette figure montre le résultat de l'étape de simplification de données de capture de mouvements. En haut, nous pouvons y voir quelques configurations du mouvement d'origine (30 coordonnées articulaires) de deux points de vue différents et, en bas, nous pouvons voir les configurations correspondantes suite à la simplification (16 coordonnées articulaires).

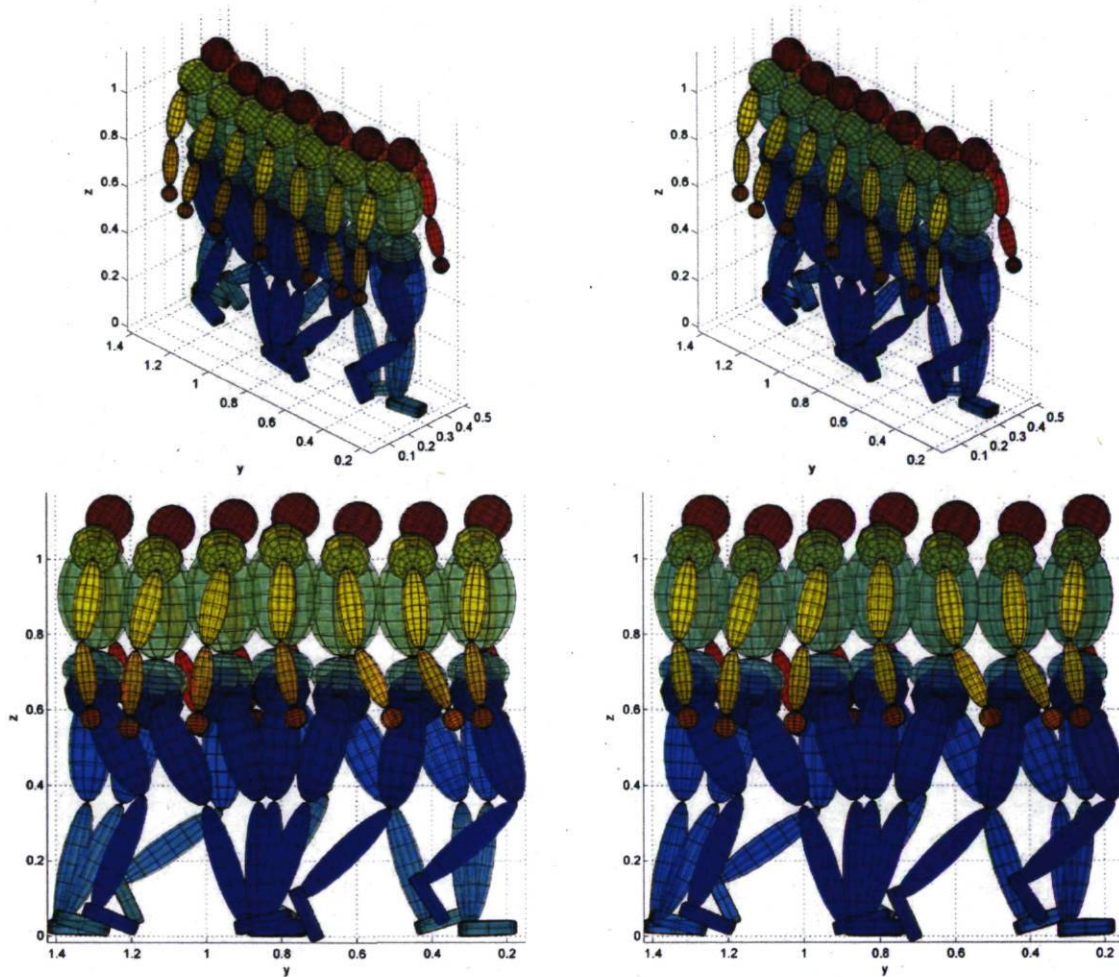


FIGURE 5.16 – Cette figure montre le résultat d’une simplification pour un cycle de marche de 28 trames (30Hz) passant de 63 coordonnées articulaires (à gauche) à 43 coordonnées articulaires (à droite).

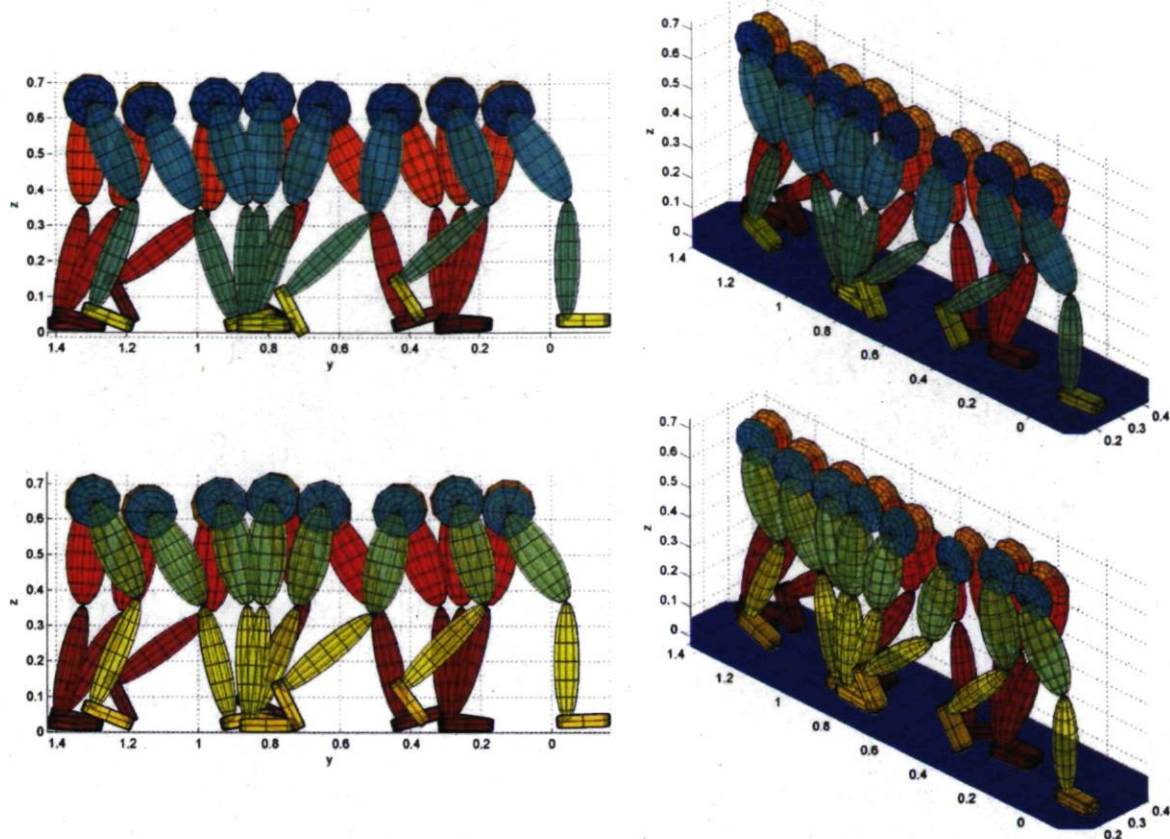


FIGURE 5.17 – Cette figure montre le résultat de l'étape de correction du mouvement simplifié. En haut, nous pouvons y voir quelques configurations du mouvement simplifié de deux points de vue différents et, en bas, nous pouvons voir les configurations correspondantes sur le mouvement tuteur produit par cette étape de correction (Q_{tut}).

Tel qu'indiqué par la table 5.3, le mouvement simplifié est utilisé comme mouvement directeur et, conjointement avec les forces de contact estimées à partir de ce mouvement, comme valeur initiale du problème de correction de mouvement.

La matrice du mouvement tuteur Q_{tut} doit être de dimensions 16×15 et la matrice F_{tut} contenant les forces de contacts correspondantes doit être de dimension 12×15 . Le vecteur x à optimiser contient donc 420 éléments, ce qui fait que l'espace d'optimisation comporte 420 dimensions. L'optimum atteint pour problème correspond au mouvement présenté à la figure 5.17 et aux forces de contact présentées à la figure 5.18. Cette optimisation de mouvement a nécessité 18,3 minutes de temps de calcul.

Ce résultat n'a rien de bien spectaculaire en apparence. Il est fort similaire au mouvement simplifié et, par extension, au mouvement capturé. Toutefois, contrairement à ces autres mouvements, le mouvement tuteur est plausible pour *HalfBoy* dans l'envi-

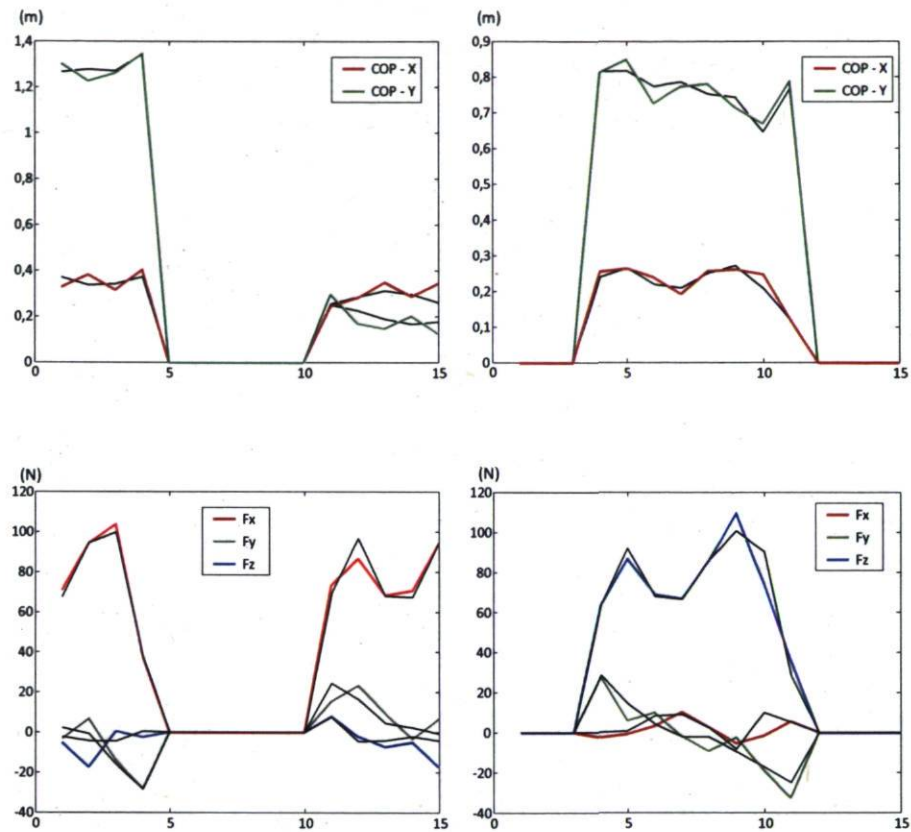


FIGURE 5.18 – Cette figure montre les forces de contact issues de l'étape de correction du mouvement simplifié. Les courbes donnent la position des centres de pression (ligne du haut) et les forces de contact (ligne du bas) pour le pied gauche (colonne de gauche) et le pied droit (colonne de droite) en fonction du temps. Les courbes tracées en noir représentent l'estimation de ces forces telles qu'obtenues par la technique d'estimation des forces de contact présentée au chapitre 4.

ronnement de synthèse et nous connaissons les forces de contact qui correspondent à ce mouvement. Ceci est d'une importance majeure dans notre approche, puisque le mouvement tuteur constitue la solution au problème de base utilisé dans notre approche d'optimisation incrémentale de mouvement (tableau 5.4).

Bien que les phases de support double ne représentent que deux trames dans ce mouvement, l'utilisation de la technique d'estimation des forces de contact (chapitre 4) est essentielle à l'obtention du mouvement tuteur. La figure 5.18 illustre bien la proximité entre les forces estimées par cette technique et les forces de contact à l'optimum du problème de correction de mouvement. Pour ce problème précis, il nous a été impossible de résoudre ce problème sans utiliser cette estimation comme valeur initiale pour les forces de contact.

5.4.2 Extrapolation de mouvements

À partir du mouvement tuteur que nous venons de produire, nous avons utilisé notre approche d'extrapolation de mouvement pour produire trois mouvements différents : une marche descendante sur un plan incliné, une marche ascendante sur un plan incliné et une marche suivant une trajectoire courbe.

Pente descendante

Le problème visé ici correspond à celui défini par le tableau 5.7 et les problèmes intermédiaires correspondent à ceux définis par le tableau 5.8.

Il nous faut toutefois définir la fonction $\Psi(\mathbf{Q}, \theta)$ qui nous permet de transformer le mouvement \mathbf{Q} en appliquant une rotation de θ radians à sa racine autour de l'axe y du repère inertiel. Cette fonction n'a pas une forme algébrique élégante, mais elle peut être décrite comme suit. Pour chaque configuration contenue dans \mathbf{Q} la fonction Ψ applique la rotation θ à la racine autour d'un axe parallèle à l'axe y du repère inertiel. La fonction applique ensuite une translation à la racine du personnage de façon à conserver la distance entre la racine et le sol dans la direction de l'accélération gravitationnelle. La figure 5.19 illustre ces deux étapes.

Cette construction de la fonction Ψ est un peu simpliste, notamment parce la longueur d'une foulée augmente avec l'inclinaison et parce qu'elle fait pénétrer les pieds dans le sol. Elle nous permet néanmoins d'illustrer l'extrapolation de mouvement. De

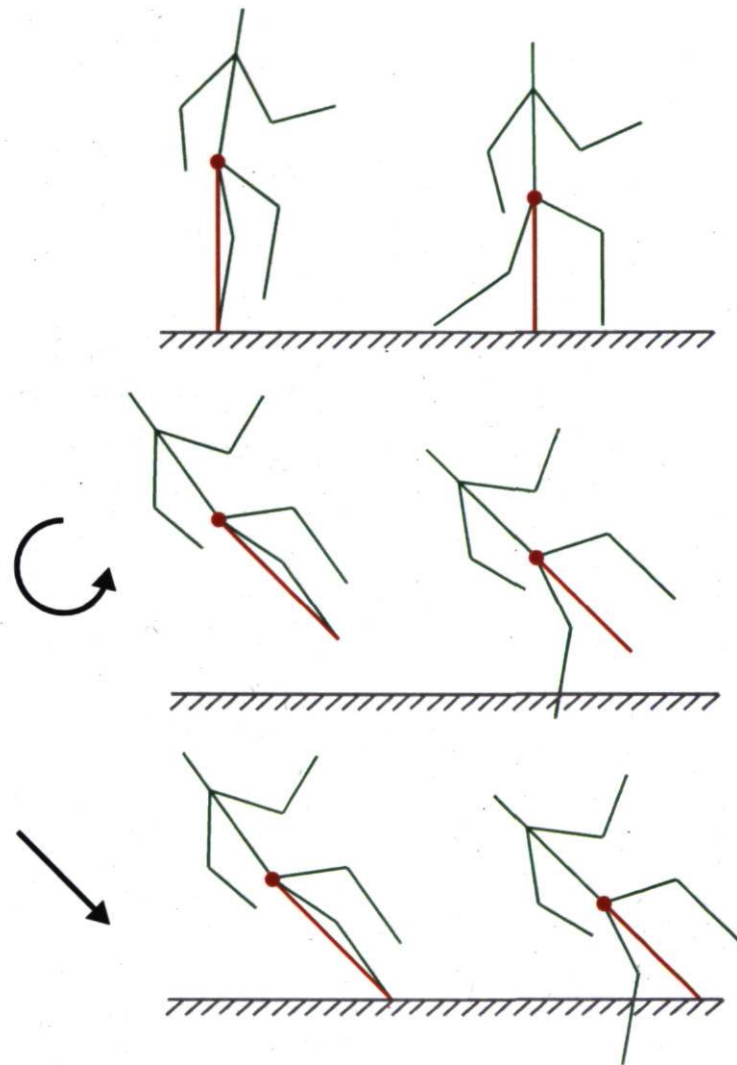


FIGURE 5.19 – Un mouvement pour lequel deux configurations sont illustrées est transformé par une fonction Ψ . La barre rouge représente la hauteur de la racine par rapport au sol. Pour chaque configuration, une rotation est appliquée à la racine suite à quoi une translation est appliquée afin de préserver la hauteur de la racine par rapport au sol selon le nouvel angle. Cette dernière étape aide à assurer que le mouvement directeur soit suffisamment proche du sol pour permettre au personnage d'être en contact avec le sol dans le mouvement extrapolé.

plus, pour des plans de faible inclinaison le mouvement extrapolé avec ce mouvement directeur nous semble suffisamment naturel.

L'inclinaison visée pour le sol est de $\pi/16$ radians. Nous avons fixé l'incrément initial à $\Delta_0 = 0,05$, la borne inférieure sur les incréments à $\epsilon = 0,005$ et le nombre maximum d'itérations internes de SNOPT pour la résolution d'un problème intermédiaire à $k = 10,000$, soit la valeur par défaut suggérée par l'algorithme SNOPT pour un problème de cette taille. La figure 5.20 montre quelques trames du résultat de l'extrapolation de mouvement et la figure 5.21 illustre la progression de cette extrapolation en fonction du temps de calcul.

Dans cet exemple, nous avons limité l'inclinaison de la pente à $\pi/16$ radians, soit $11,25^\circ$. C'est une inclinaison relativement modeste, mais qui permet néanmoins de produire un mouvement significativement différent du mouvement tuteur tout en préservant l'aspect naturel du mouvement.

Au-delà de cette inclinaison, les mouvements extrapolés commencent à perdre de leur réalisme. Cela nous amène à formuler un point important : bien que notre approche permette de produire des mouvements réalistes, tout mouvement issu de cette approche n'est pas forcément réaliste. Il appartient donc à l'utilisateur de juger quel est le seuil à partir duquel le mouvement perd son réalisme.

Bien que subjectif, le seuil au-delà duquel le réalisme du mouvement s'effrite dépend de plusieurs facteurs, dont la nature du mouvement que l'on désire produire et la qualité du mouvement directeur utilisé. Comme nous le verrons avec l'exemple de la marche suivant une trajectoire courbe, un mouvement directeur qui ne comporte pas de défauts majeurs, tels que la pénétration des pieds dans le sol, permet une plus grande amplitude dans l'extrapolation de mouvements.

Pente ascendante

L'exemple de la pente ascendante est presque identique à celui de la pente descendant, à l'exception que la pente visée est d'un angle de $-\pi/16$ radians. Ici aussi avons fixé l'incrément initial à $\Delta_0 = 0,05$, la borne inférieure sur les incréments à $\epsilon = 0,005$ et le nombre maximum d'itérations internes de SNOPT pour la résolution d'un problème intermédiaire à $k = 10,000$. La figure 5.22 montre quelques trames du résultat de l'extrapolation de mouvement et la figure 5.23 illustre la progression de cette extrapolation en fonction du temps de calcul.

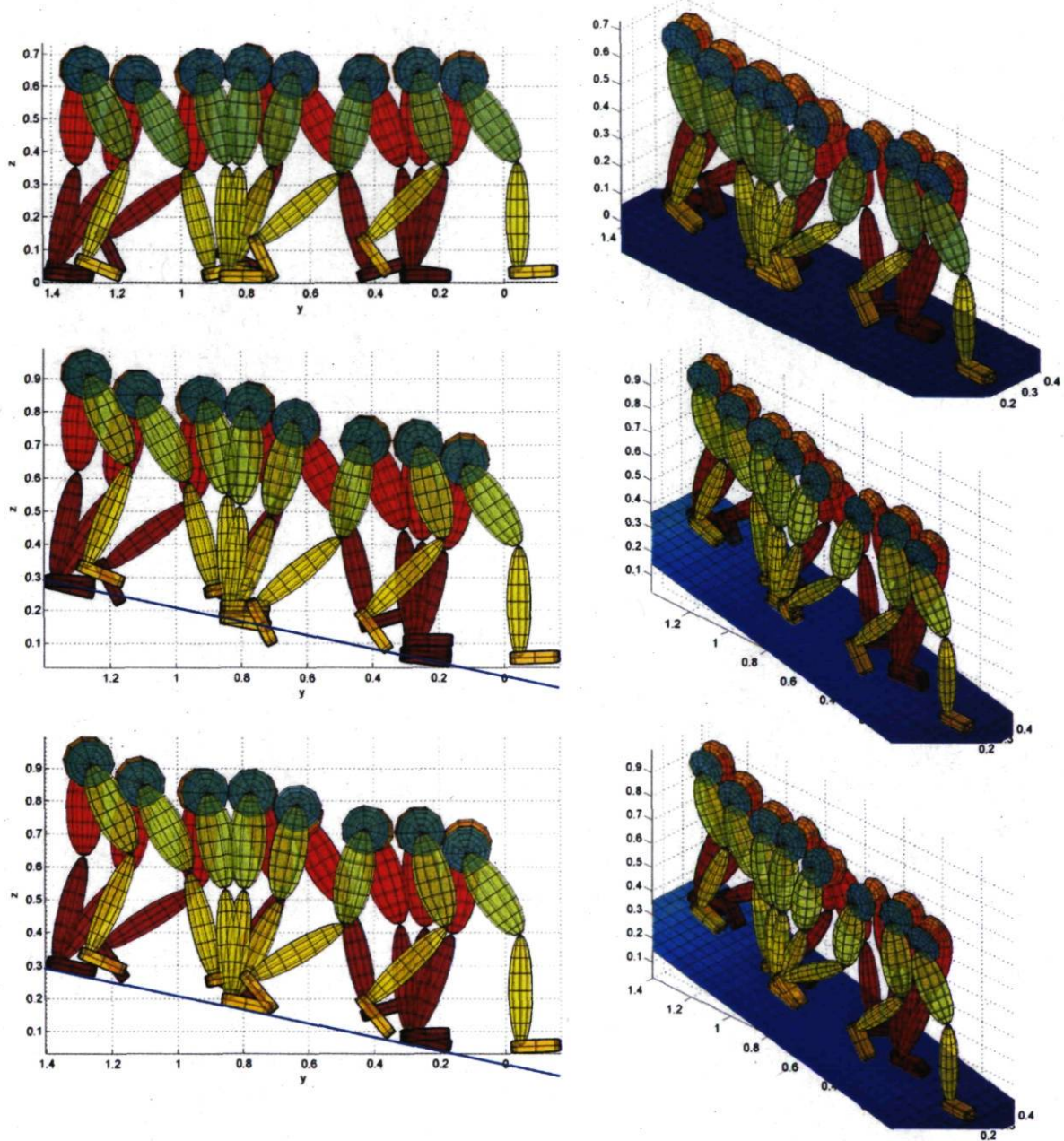


FIGURE 5.20 – En haut, nous présentons quelques trames du mouvement tuteur utilisé pour extrapoler un cycle de marche sur une pente descendante. Le mouvement directeur pour une valeur de $\lambda = 1$ et le mouvement extrapolé sont illustrés au milieu et au bas de la figure respectivement.

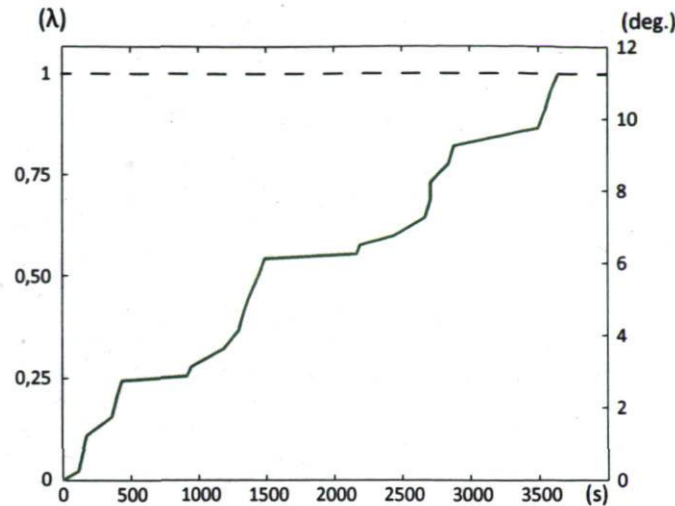


FIGURE 5.21 – Cette courbe montre la progression de l'extrapolation du mouvement en fonction du temps de calcul pour la marche sur une pente descendante. L'axe des abscisses est gradué des valeurs de λ à gauche et des valeurs de l'inclinaison de la pente correspondantes à droite.

Dans cet exemple aussi, nous avons limité l'inclinaison de la pente à $-\pi/16$ radians. Toutefois, contrairement à la marche sur une pente descendante, le mouvement commence à présenter une caractéristique qui ne provient pas du mouvement tuteur, soit un *glissement* du pied au-dessus de la surface le sol avant le contact du talon contre le sol. Bien que le mouvement ait tout de même une apparence naturelle, cette caractéristique du mouvement est introduite par le mouvement directeur, dans lequel le pied passe au travers du sol. Une fois de plus, un choix de mouvement directeur plus judicieux pourrait contrer ce problème.

L'exemple de la pente ascendante nous permet aussi d'illustrer le problème de l'impasse causée par des contraintes qui se referment graduellement sur la solution au cours de l'optimisation incrémentale. Supposons que nous tentons d'obtenir un cycle de marche sur une pente de $-\pi/4$ radians et que nous ne posons pas de borne inférieure sur la valeur des incréments. Le processus d'extrapolation atteindra éventuellement une impasse, un point où aucun progrès n'est possible et où les incréments vont tendre vers 0. La figure 5.24 illustre cette situation.

Trajectoire courbe

Extrapoler un cycle de marche suivant un arc de cercle de $\pi/2$ radians correspond au problème défini par le tableau 5.5 et les problèmes intermédiaires correspondent à

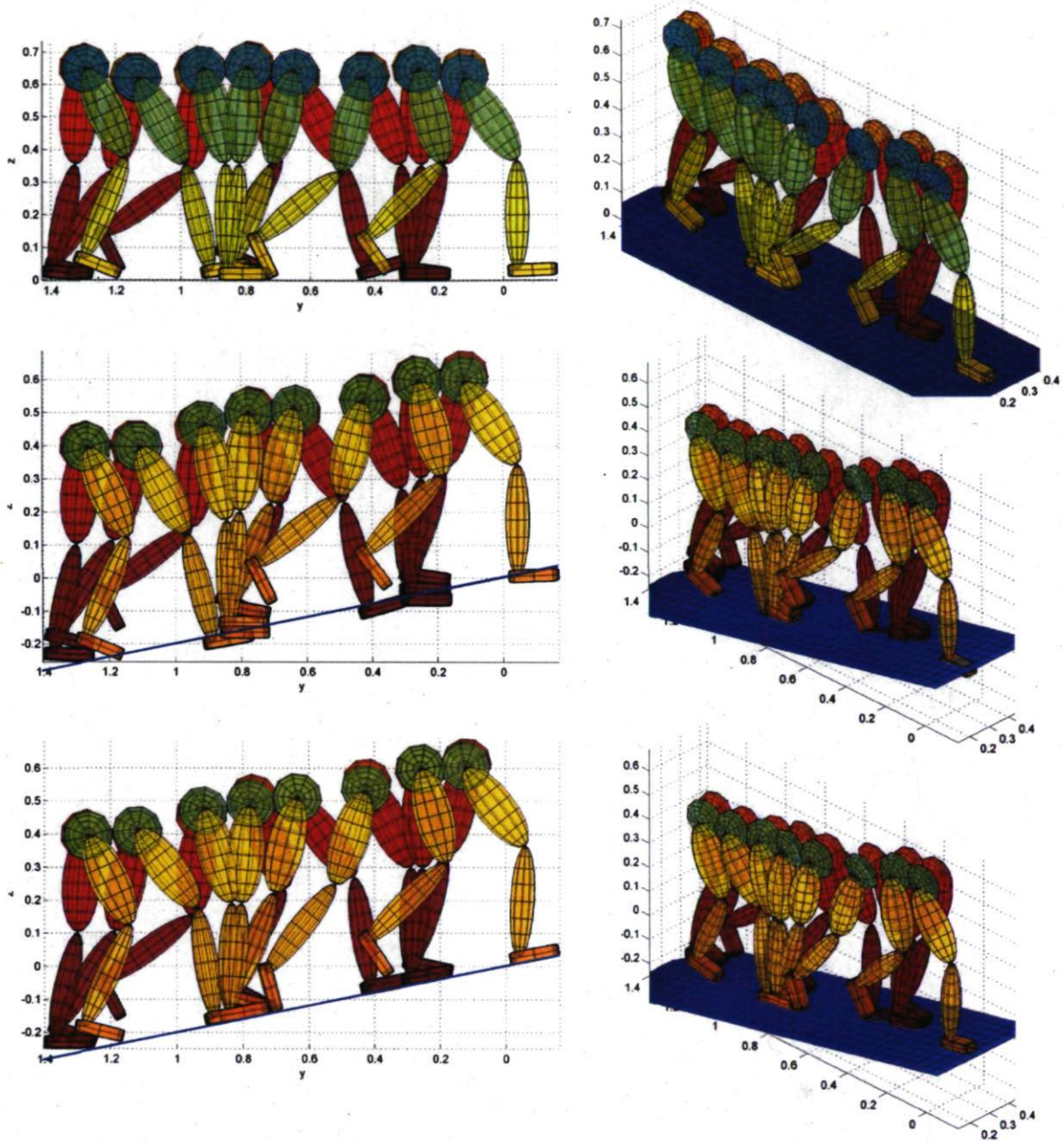


FIGURE 5.22 – En haut, nous présentons quelques trames du mouvement tuteur utilisé pour extrapoler un cycle de marche sur une pente ascendante. Le mouvement directeur pour une valeur de $\lambda = 1$ et le mouvement extrapolé sont illustrés au milieu et au bas de la figure respectivement.

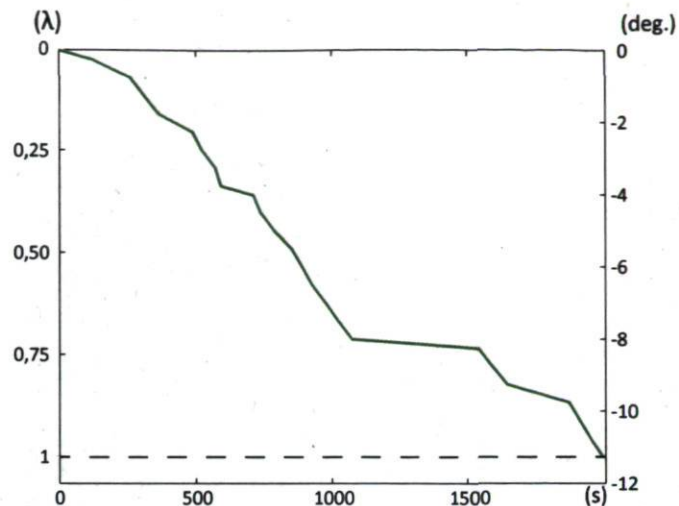


FIGURE 5.23 – Cette courbe montre la progression de l'extrapolation du mouvement en fonction du temps de calcul pour la marche sur une pente ascendante. L'axe des abscisses est gradué des valeurs de λ à gauche et des valeurs de l'inclinaison de la pente correspondantes à droite.

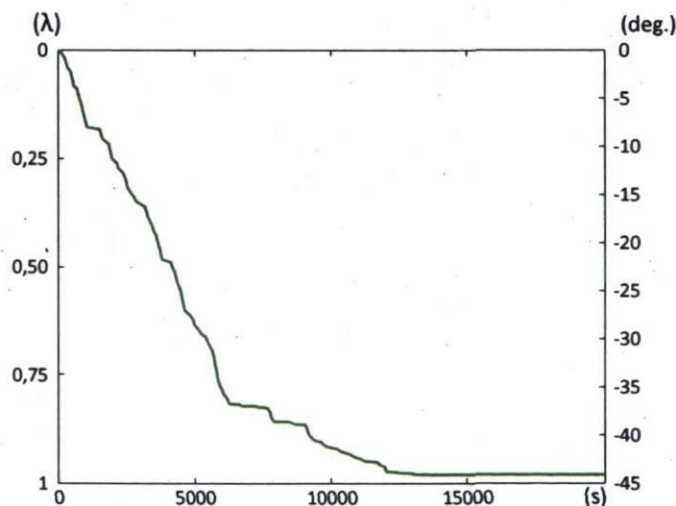


FIGURE 5.24 – Cette courbe montre la progression de l'extrapolation du mouvement en fonction du temps de calcul pour la marche sur une pente ascendante. À un angle proche de $-\pi/4$ radians, l'extrapolation de mouvement rencontre une impasse et la progression n'est plus possible. Sans mécanisme à cet effet, l'algorithme d'extrapolation serait alors coincé dans une boucle sans fin.

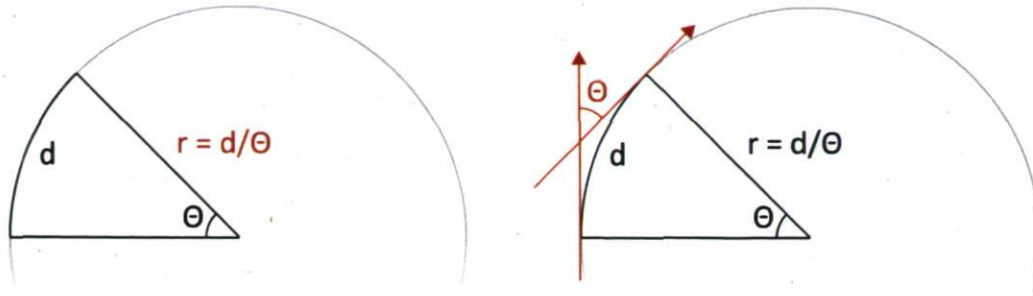


FIGURE 5.25 – À gauche, la figure illustre la relation entre la longueur d'un arc de cercle, l'angle qu'il sous-tend et la longueur du rayon du cercle. À droite, la figure illustre que l'angle θ correspond également à l'angle entre les tangentes aux deux extrémités de l'arc de cercle.

ceux définis par le tableau 5.6.

Nous devons ici définir la fonction $\Phi(\mathbf{Q}, \theta)$ qui nous permet de transformer la trajectoire d'un mouvement \mathbf{Q} en une trajectoire courbe suivant un arc de cercle de θ radians. À l'instar de la fonction Ψ , cette fonction n'a pas une forme algébrique élégante. Elle se décrit mieux avec quelques figures accompagnées d'explications.

Tel qu'illustré à la figure 5.25, un arc de cercle de longueur d sous-tend un angle de θ radians si le rayon du cercle est égal à d/θ . Soulignons également que l'angle entre la tangente aux deux extrémités de l'arc de cercle est également de θ radians (figure 5.25). Cette dernière remarque sera utile pour ajuster l'orientation du personnage le long de cette courbe.

En supposant que le mouvement \mathbf{Q} est cyclique et qu'il suit une trajectoire rectiligne, à vol d'oiseau nous pouvons tracer un segment de droite de longueur d_T entre la racine du personnage à la première configuration et sa racine à l'avant-dernière configuration⁸. La position de la racine à chaque configuration t du mouvement peut alors être paramétrisée par une distance d_t le long de ce segment, ce qui correspond à une longueur d'arc pour un cercle de rayon infini, et par un vecteur sur le plan normal à ce segment de droite donnant la déviation de la racine par rapport au segment de droite (figure 5.26). Notons que la longueur d'arc correspondant à la dernière configuration est supérieure à d_T , mais la même paramétrisation s'applique.

Pour adapter ce mouvement à une trajectoire courbe suivant un arc de cercle de θ

8. Selon notre contrainte sur les mouvements cycliques, la première et l'avant-dernière configurations représentent la même phase dans le cycle

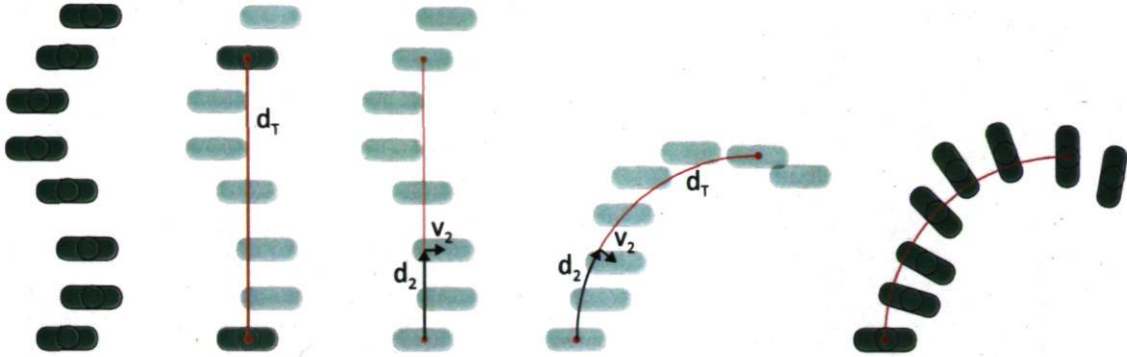


FIGURE 5.26 – À vol d’oiseau, un mouvement pour lequel les configurations illustrées sont transformées par une fonction Φ . De gauche à droite, la figure présente le mouvement original, le segment de droite de longueur d_T , la paramétrisation de la racine d’une configuration, l’application de cette paramétrisation sur l’arc de cercle de longueur d_T et les configurations finales suite à l’ajustement de l’orientation de chaque configuration.

radians, il suffit de remplacer la segment de droite par un arc de cercle de longueur d_T sur un cercle de rayon d/θ . En utilisant la paramétrisation précédente pour la position de la racine sur cet arc de cercle, nous nous trouvons à appliquer la translation nécessaire à chaque configuration pour suivre la trajectoire courbe. Nous appliquons ensuite une rotation autour de la racine selon l’axe z du repère inertiel afin d’orienter le personnage correctement à chaque configuration. L’angle de rotation correspond à la longueur d’arc d_t de cette configuration. Cet angle est donc de $\theta d_t/d_T$ radians. Ce processus est illustré à la figure 5.26.

Cette construction de la fonction Φ est très adéquate pour ce problème. Outre des glissements de pieds sur le sol, elle n’introduit aucun défaut majeur dans le mouvement directeur.

La courbure de la trajectoire visée est de $\pi/2$ radians, ce qui correspond à un quart de tour en un cycle de marche. Nous avons fixé l’incrément initial à $\Delta_0 = 0,005$, la borne inférieure sur les incréments à $\epsilon = 0,0005$ et le nombre maximum d’itérations internes de SNOPT pour la résolution d’un problème intermédiaire à $k = 10,000$. La figure 5.27 montre quelques trames du résultat de l’extrapolation de mouvement et la figure 5.28 illustre la progression de cette extrapolation en fonction du temps de calcul.

Cet exemple montre bien la potentiel de notre approche en produisant des mouvements réalistes significativement différents des données originales de capture de mouvement. Puisque nous passons d’une marche rectiligne à une marche suivant une courbure prononcée, nous pouvons dire que la portée d’extrapolation de cet exemple est significativement plus grande que celle des deux exemples précédents, où nous passons d’un

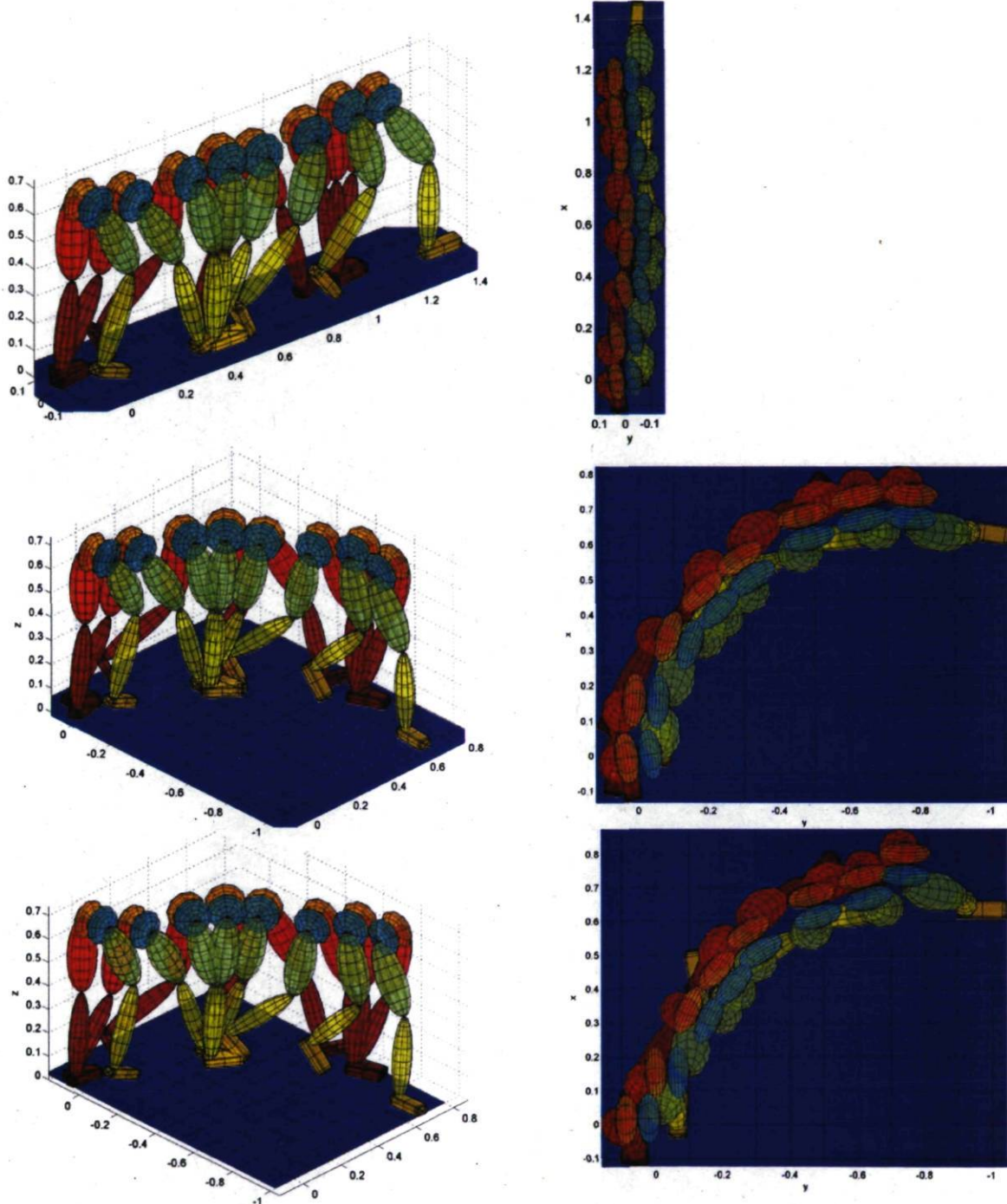


FIGURE 5.27 – En haut, nous présentons quelques trames du mouvement tuteur utilisé pour extrapoler un cycle de marche suivant une trajectoire courbe. Le mouvement directeur pour une valeur de $\lambda = 1$ et le mouvement extrapolé sont illustrés au milieu et au bas de la figure respectivement.

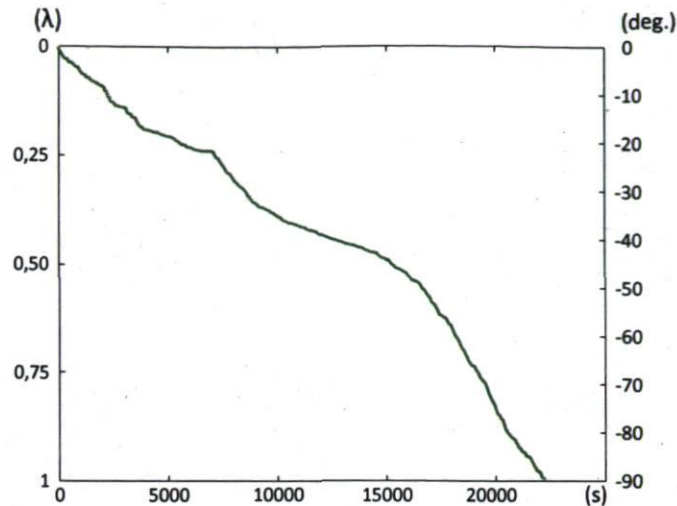


FIGURE 5.28 – Cette courbe montre la progression de l'extrapolation du mouvement en fonction du temps de calcul pour la marche suivant une trajectoire courbe. L'axe des abscisses est gradué des valeurs de λ à gauche et des valeurs des angles de courbure correspondantes à droite.

sol horizontal à une légère pente.

Tel que discuté dans la présentation des exemples précédents, nous avançons que la qualité du mouvement directeur est un facteur majeur dans le potentiel d'extrapolation d'un mouvement. Il convient donc d'y porter une attention particulière avant de se lancer dans l'extrapolation d'un mouvement. Nous sommes également d'avis que certains problèmes d'extrapolation sont intrinsèquement plus simples à résoudre que d'autres.

Ces derniers résultats nous donnent également un bon exemple de l'avantage procuré par une approche incrémentale basée sur l'homotopie entre deux problèmes. En effet, les incréments retenus au long de l'optimisation incrémentale sont très faibles, ce qui nous indique qu'un problème autrement trop complexe peut être remplacé adéquatement par une séquence de problèmes significativement plus simples. Malgré un très grand nombre d'itérations de notre algorithme, le problème est éventuellement résolu.

5.4.3 Discussion

Bien que les résultats présentés aient été obtenus sur un modèle humain simplifié, il ne faut pas confondre les limites de notre approche et les limites de notre implémentation actuelle. Sans croire que cette approche permette l'extrapolation de tout mouvement sans égard à la complexité du personnage, du mouvement ou de l'environnement, nos

résultats sont clairement limités par des facteurs liés à l'implémentation et non par des contraintes propres à l'approche. En effet, les problèmes d'optimisation avec lesquels nous devons composer sont très complexes, notamment à cause des grandes dimensions du problème, ce qui rend le développement du code complexe et l'interprétation des résultats difficile lors de la validation de l'exactitude des modèles.

En effet, toutes les caractéristiques d'un problème de marche bipède sont présentes dans notre étude de cas. Nous traitons un système multicorps bipède qui cherche à imiter des données de capture de mouvement sous un ensemble de contraintes. Utiliser un personnage humain entier dans l'extrapolation de mouvements augmente la taille du problème. Cela n'introduit toutefois pas de nouvelles contraintes ou de nouvelles caractéristiques au problème et nos résultats montrent bien que l'approche proposée permet de résoudre ce type de problème.

En ce sens, l'étude de cas présentée illustre bien la portée de notre approche. Elle permet de produire un grand nombre de variations d'un mouvement donné. Dans la mesure où ces variations ne demandent pas un changement de stratégie dans le mouvement du personnage, celles-ci conservent l'aspect naturel du mouvement. Rappelons que l'extrapolation de mouvements numérisés par optimisation est une contribution originale qui s'inscrit dans l'état de l'art en animation de personnages basée sur la physique.

Un aspect intéressant pour certaines applications est que notre approche d'optimisation incrémentale produit également une multitude de mouvements intermédiaires puisqu'elle est basée sur l'homotopie entre deux problèmes⁹. L'ensemble de ces mouvements intermédiaires sont également des mouvements plausibles et réalistes qui peuvent être utilisés pour animer un personnage. À titre d'exemple, en extrapolant un cycle de marche suivant un arc de cercle de $\pi/2$ radians, nous nous trouvons à produire une multitude de mouvements de marche suivant un arc de cercle de courbure intermédiaire.

De plus, nous pouvons dégager de notre approche l'opération d'*optimisation d'un mouvement*. Cette opération s'avère très versatile dans notre approche puisque le choix de certains paramètres nous permet d'obtenir différents résultats, tels qu'un mouvement simplifié, un mouvement corrigé ou un mouvement extrapolé. Il nous semble évident que cette opération pourrait également être utilisée à d'autres fins dans le domaine de l'animation de personnages basée sur la physique.

9. Nous pouvons d'ailleurs considérer que ces mouvements sont des échantillons d'un continuum de mouvements. En effet, puisque l'homotopie définit un continuum de problèmes d'optimisation de mouvements entre le problème de base et le problème visé, l'optimum de chaque problème intermédiaire fait donc partie d'un continuum.

En analysant les temps de calculs donnés aux figures 5.23, 5.21 et 5.28, il est évident que l'extrapolation de mouvement telle que nous l'avons définie est un problème d'optimisation extrêmement complexe. Une analyse de la non linéarité du problème d'optimisation sous-tendu permettrait certainement de l'exprimer sous une forme plus simple à résoudre, tel que nous l'avons fait pour notre approche d'estimation de forces de contact. Cela montre aussi tout le potentiel de notre approche d'optimisation incrémentale, qui permet de résoudre un type de problème qui serait autrement insoluble.

Chapitre 6

Conclusion

La reproduction adéquate de mouvements humains est un aspect très important dans le domaine de l'animation. Toutefois, elle nécessite souvent la capture du mouvement sur un acteur humain. Malgré la grande qualité des mouvements capturés, il s'agit d'un processus long et coûteux dont le résultat n'a aucune souplesse : le mouvement capturé n'est applicable qu'à un personnage similaire à l'acteur dans un environnement similaire à l'environnement de capture.

Dans le cadre de cet ouvrage, nous avons abordé le problème de la synthèse de mouvements humains plausibles et réalistes. Le mouvement est qualifié de plausible s'il répond aux contraintes de la mécanique classique et il est qualifié de réaliste s'il peut être jugé comme étant d'apparence naturelle. Ces deux aspects ont orienté notre décision quant au choix de l'approche à privilégier.

Afin d'assurer le respect des lois de la mécanique classique dans le mouvement synthétisé, nous avons étudié les différentes approches en animation basée sur la physique. Ces approches peuvent généralement être classées dans l'une de trois catégories : les contraintes spatiotemporelles, les contrôleurs en dynamique directe et le filtrage dynamique. De ces approches, les contraintes spatiotemporelles nous semblent être le paradigme à adopter dans notre solution étant donné le contrôle qu'il nous accorde sur l'animation résultante par le biais de la fonction de coût et des fonctions de contrainte.

Le réalisme du mouvement est une notion difficile à quantifier et son évaluation demeure subjective. Afin de favoriser la synthèse d'un mouvement réaliste, nous exploitons des données de capture de mouvement, lesquelles sont réalistes par construction. Par extrapolation de mouvement sur ces données, nous sommes en mesure de faire la synthèse de nouveaux mouvements tout en conservant les caractéristiques propres au

mouvement initial.

Nous avons donc développé une approche d'extrapolation de mouvement basée sur le paradigme des contraintes spatiotemporelles. La fonction de coût utilisée sert à assurer que le mouvement synthétisé conservera une similarité avec le mouvement à extrapoler alors que les fonctions de contrainte assurent le respect des lois de la mécanique classique. Cette approche répond au problème initial en ce sens qu'elle ajoute une souplesse aux mouvements issus de capture de mouvement en permettant de produire de multiples variations de ces mouvements.

6.1 Réalisations

Dans le cadre de nos travaux, nous avons développé et défini cette approche d'extrapolation de mouvement en nous inspirant de la littérature scientifique sur les contraintes spatiotemporelles. Nous avons également fait une première implémentation de cette approche. Bien que cette implémentation pourrait être améliorée à plusieurs égards, elle nous a néanmoins permis de tester cette idée sur des cas réels et d'en évaluer la portée et les limites.

Notre solution se divise en deux étapes. D'abord, les données de capture de mouvement sont ajustées afin de produire le mouvement tuteur, un mouvement plausible sur le personnage utilisé dans l'environnement de synthèse et pour lequel nous connaissons les forces externes. Ensuite, nous extrapolons divers mouvements à partir de ce mouvement tuteur. Ceux-ci correspondent à l'application du mouvement tuteur dans des conditions différentes. Dans cette approche, l'optimisation d'un mouvement par contraintes spatiotemporelles est un outil dont nous faisons usage à diverses fins.

Les contraintes spatiotemporelles constituent une formulation très élégante d'un problème de synthèse de mouvements. Cependant, les problèmes de programmation non linéaires sous-tendus par celles-ci cachent une grande complexité. Pour un personnage modérément complexe, il n'est généralement pas possible de les résoudre directement par l'application brute d'un algorithme d'optimisation non linéaire. C'est ici qu'interviennent nos deux principales contributions, soient une technique d'estimation des forces de contact (chapitre 4) et une approche d'extrapolation de mouvement progressive (chapitre 5). Ces deux contributions visent un même objectif, celui de simplifier au maximum les problèmes d'optimisation de mouvement rencontrés dans notre approche.

La technique d'estimation des forces de contact est utilisée à l'étape de la création du

tuteur. Elle simplifie le problème de la correction des données de capture de mouvement en donnant une valeur initiale aux paramètres de force de contact. Dans la mesure où les données de capture de mouvement ne demandent pas une trop grande correction, les forces estimées devraient être assez similaires aux forces du mouvement optimal. En fournissant une valeur initiale suffisamment proche de l'optimum, nous simplifions significativement le problème d'optimisation lié à la correction du mouvement.

Notre estimation des forces peut s'appliquer à n'importe quel mouvement qui comporte des phases de support double pour en estimer la répartition des forces de contact. Dans la cadre de notre approche, nous l'appliquons aux données de capture de mouvement puisque celles-ci ne comportent aucune information sur les forces externes impliquées dans le mouvement. La technique consiste à trouver, pour les phases de support double, les forces de réaction du sol minimisant l'effort interne du personnage. Bien que l'hypothèse de l'effort interne minimal ne s'applique généralement pas parfaitement aux mouvements de faible intensité tels que la marche, les forces estimées par cette technique constituent néanmoins une valeur initiale adéquate pour le problème de correction du mouvement. D'ailleurs, les résultats présentés dans la section 4.3 montrent que les forces estimées par cette technique se comparent avantageusement aux forces mesurées par des plateformes de force, un équipement qui n'est généralement pas présent dans les studios de capture de mouvement.

Un aspect central de notre contribution est notre formulation du problème d'estimation de ces forces. Nous avons formulé ce problème comme une optimisation imbriquée, où une boucle externe fait une recherche linéaire sur des sous-problèmes de programmation quadratique. Cette formulation est en tout point équivalente à la formulation *naïve* du problème, mais elle permet d'en atteindre la solution très efficacement.

Notre seconde contribution majeure est notre approche d'extrapolation *incrémentale* de mouvement. Elle intervient évidemment à l'étape d'extrapolation de notre solution. Cette approche divise le problème d'extrapolation de mouvement en plusieurs problèmes intermédiaires. Chaque problème intermédiaire est choisi de façon à être suffisamment simple à résoudre. Ainsi, au lieu de nous buter à un problème d'optimisation de mouvement trop complexe pour nos algorithmes d'optimisation, nous résolvons une séquence de problèmes beaucoup plus simples.

Cette approche est basée sur l'homotopie qui existe entre deux problèmes d'optimisation de mouvement. Un premier problème, le problème de base, est celui pour lequel le mouvement tuteur est l'optimum. Le second problème, le problème visé, est celui dont l'optimum est le mouvement dont nous désirons faire la synthèse. L'homotopie entre ces problèmes représente un continuum de problèmes intermédiaires entre le problème

de base et le problème visé, et les optima de ces problèmes forment également un continuum. Ainsi, en progressant le long de ce continuum par des incréments suffisamment petits, la solution d'un problème intermédiaire peut être utilisée comme valeur initiale pour l'incrément suivant. Chaque problème intermédiaire est donc relativement simple et nous permet éventuellement d'atteindre la solution du problème visé.

6.2 Travaux futurs

Si nos travaux sont originaux, ils constituent également une recherche exploratoire, une première solution proposée dans cette piste de recherche. Ainsi, plusieurs aspects de notre implémentation demandent à être raffinés afin de combler certaines lacunes et d'augmenter la portée de l'approche.

Une lacune avouée de notre implémentation est sa représentation des articulations à 3 DDLs par trois articulations rotoïdes, paramétrisée par des angles d'Euler. Cette représentation peut entraîner des problèmes liés aux singularités d'une telle paramétrisation. De plus, étant donné que les coordonnées d'une telle représentation ne sont généralement pas indépendantes, cet espace de représentation des rotations peut être nuisible au processus d'optimisation.

En ce sens, il serait souhaitable de représenter de telles articulations par des articulations sphériques, pour lesquels les coordonnées sont toujours indépendantes. Cela requiert un travail important au sens de l'implémentation, mais celle-ci en serait bonifiée. Il est toutefois difficile d'évaluer le gain réel d'une telle amélioration sur les tâches d'optimisation de mouvements.

Une autre lacune importante de notre approche est notre choix des divers incréments lors de l'extrapolation de mouvement. L'algorithme utilisé a une mémoire très courte. Il cherche toujours à doubler la taille de l'incrément suite au succès de l'optimisation d'un problème intermédiaire et le divise en deux en cas d'échec.

Cet algorithme produit souvent un comportement peu désirable et relativement coûteux en temps de calcul. Supposons que pour un problème donné l'incrément Δ_i idéal, soit le plus gros incrément produisant un problème intermédiaire suffisamment simple, est égal à l'incrément précédent, Δ_{i-1} . L'algorithme tentera d'abord de doubler l'incrément, suite à quoi l'optimisation ne convergera pas dans les délais désirés. L'algorithme, en divisant par deux le dernier incrément tenté, se rabattra ensuite avec succès sur la valeur de Δ_{i-1} . Cette séquence peut se reproduire à maintes reprises lors

de l'extrapolation d'un mouvement et ainsi ajouter un temps de calcul important à la tâche.

Nos expérimentations nous portent à croire que, bien que cet incrément optimal ne soit pas constant tout au long d'un problème d'extrapolation de mouvement donné, il présente néanmoins une cohérence locale. Nous avons d'ailleurs observé à plusieurs reprises le comportement décrit au paragraphe précédent. C'est pourquoi nous sommes d'avis que l'élaboration d'un algorithme un peu plus sophistiqué pour fixer les incréments permettrait potentiellement un gain de performance significatif dans l'extrapolation d'un mouvement. Une telle approche pourrait par exemple fixer l'incrément à partir d'une régression linéaire des n derniers incréments choisis. Cet algorithme devrait également tenter occasionnellement d'augmenter la valeur de l'incrément, sans quoi cette valeur ne pourrait que diminuer au long de l'extrapolation.

Suite aux résultats présentés dans la section 5.4, il nous semble qu'une attention plus particulière devrait être portée aux fonctions produisant les mouvements directeurs des problèmes intermédiaires. Le mouvement directeur est ultimement le mouvement que l'optimisation cherche à imiter. Certaines erreurs dans ces mouvements, telles que la pénétration des pieds dans le sol, semblent nuire fortement à la capacité d'extrapolation de notre approche. C'est un aspect de notre approche qui a été traité rapidement et qui mériterait d'être étudié plus en profondeur. Une étude plus approfondie des recherches en sciences du mouvement, telles que [38, 80, 47, 56, 77], nous permettrait certainement de produire des mouvements directeurs de plus grande qualité.

Les sciences du mouvement pourraient aussi nous permettre de fixer des critères d'évaluation du réalisme du mouvement qui soient objectifs, valables et reproductibles. Ceci pourrait nous permettre d'appuyer l'évaluation subjective de nos résultats. Il serait également intéressant de tester nos approches d'extrapolation de mouvement et d'estimation des forces de contact sur un plus grand nombre de séquences afin de mieux valider ces approches.

Il serait également pertinent d'évaluer l'effet des différents choix que nous avons effectués lors de la modélisation biomécanique du personnage. À titre d'exemple, il serait intéressant de quantifier dans quelle mesure la représentation des membres par des ellipsoïdes affecte la qualité des mouvements extrapolés et la précision des forces de contact estimées par rapport à un modèle plus complexe s'approchant davantage d'un sujet humain.

En analysant les résultats de notre approche d'extrapolation de mouvement, il semble évident que le problème d'optimisation non linéaire que nous posons est très

complexe. Tel que mentionné au chapitre précédent, une analyse de la non linéarité du problème d'optimisation sous-tendu permettrait certainement de l'exprimer sous une forme plus simple à résoudre, tel que nous l'avons fait pour notre approche d'estimation de forces de contact. De plus, il serait intéressant d'étudier la possibilité de borner les fonctions de coût et de contraintes par des fonctions convexes, et de voir dans quelle mesure cela nous permet de simplifier le problème sans nuire à la qualité des résultats.

Finalement, nous croyons qu'il serait fort bénéfique de permettre l'optimisation des paramètres discrets qui composent le mouvement, à l'image de l'approche présentée par Wampler et Popović [84]. Cela permettrait notamment d'ajuster les temps de contact des pieds pendant le processus d'optimisation, ce qui permettrait sans doute d'augmenter la portée de notre approche d'extrapolation de mouvement. Toutefois, il nous semble aussi évident qu'un tel ajout se traduirait par une diminution significative de la performance de notre approche. L'optimisation des temps de contact pourrait donc être utilisée plus sporadiquement.

Bibliographie

- [1] Yeuhi Abe, Marco da Silva, and Jovan Popović. Multiobjective control with frictional contacts. In *SCA '07 : Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 249–258, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.
- [2] Yeuhi Abe, C. Karen Liu, and Zoran Popović. Momentum-based parameterization of dynamic character motion. *Graph. Models*, 68(2) :194–211, 2006.
- [3] Yeuhi Abe and Jovan Popović. Interactive animation of dynamic manipulation. In *SCA '06 : Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, New York, NY, USA, 2006. ACM Press.
- [4] Okan Arikan and David 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, New York, NY, USA, 2002. ACM Press.
- [5] William W. Armstrong and Mark W. Green. The dynamics of articulated rigid bodies for purposes of animation. In *The Visual Computer*, volume 1, pages 231–240. Springer Berlin / Heidelberg, 1985.
- [6] Constantinos A. Balafoutis and Rajnikant V. Patel. *Dynamic analysis of robot manipulators : A cartesian tensor approach*. Kluwer Academic Publishers, 1991.
- [7] Ronen Barzel and Alan H. Barr. A modeling system based on dynamic constraints. In *SIGGRAPH '88 : Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 179–188, New York, NY, USA, 1988. ACM.
- [8] Philippe Beaudoin, Stelian Coros, Michiel van de Panne, and Pierre Poulin. Motion-motif graphs. In *SCA '08 : Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 117–126, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association.
- [9] T.W. Calvert, J. Chapman, and A. Patla. Aspects of the kinematic simulation of human movement. In *IEEE Computer Graphics and Applications*, volume 2, pages 41–50, September 1982.

- [10] Jinxiang Chai and Jessica K. Hodgins. Performance animation from low-dimensional control signals. In *SIGGRAPH '05 : ACM SIGGRAPH 2005 Papers*, pages 686–696, New York, NY, USA, 2005. ACM.
- [11] Jia chi Wu and Zoran Popović. Realistic modeling of bird flight animations. In *ACM Transactions on Graphics*, volume 22, pages 888–895, New York, NY, USA, 2003. ACM Press.
- [12] Byoungwon Choe, Hanook Lee, and Hyeong seok Ko. Performance-driven muscle-based facial animation. *The Journal of Visualization and Computer Animation*, 12 :67–79, 2001.
- [13] Michael F. Cohen. Interactive spacetime control for animation. In *SIGGRAPH '92 : Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 293–302, New York, NY, USA, 1992. ACM Press.
- [14] Marco da Silva, Yeuhi Abe, and Jovan Popović. Interactive simulation of stylized human locomotion. In *SIGGRAPH '08 : ACM SIGGRAPH 2008 papers*, pages 1–10, New York, NY, USA, 2008. ACM.
- [15] Marco da Silva, Frédo Durand, and Jovan Popović. Linear bellman combination for control of character animation. In *SIGGRAPH '09 : ACM SIGGRAPH 2009 papers*, pages 1–10, New York, NY, USA, 2009. ACM.
- [16] Edilson de Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. *ACM Trans. Graph.*, 27(3) :1–10, 2008.
- [17] Petros Faloutsos, Michiel van de Panne, and Demetri 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, New York, NY, USA, 2001. ACM Press.
- [18] Petros Faloutsos, Michiel van de Panne, and Demetri Terzopoulos. The virtual stuntman : dynamic characters with a repertoire of autonomous motor skills. In *Computers and Graphics*, volume 25, pages 933–953. 2001.
- [19] Petros Faloutsos, Michiel van de Panne, and Demetri Terzopoulos. Autonomous reactive control for simulated humanoids. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation, ICRA 2003*, pages 917–924, Taipei, Taiwan, September 2003.
- [20] Anthony C. Fang and Nancy S. Pollard. Efficient synthesis of physically valid human motion. In *ACM Transactions on Graphics*, volume 22, pages 417–426, New York, NY, USA, 2003. ACM Press.
- [21] Anthony C. Fang and Nancy S. Pollard. Efficient computation of optimal, physically valid motion. In *Journal of Robotics Society of Japan*, volume 22. March 2004.

- [22] Roy Featherstone. *Robot Dynamics Algorithms*. Kluwer Academic Publishers, 1987.
- [23] Roy Featherstone. *Rigid Body Dynamics Algorithms*. Springer, 2008.
- [24] Philip E. Gill, Walter Murray, and Michael A. Saunders. Snopt : An sqp algorithm for large-scale constrained optimization. *SIAM Rev.*, 47(1) :99–131, 2005.
- [25] Michael Girard and Anthony 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, New York, NY, USA, 1985. ACM.
- [26] Michael Gleicher. Motion editing with spacetime constraints. In *I3D '97 : Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 139–ff., New York, NY, USA, 1997. ACM.
- [27] Keith Grochow, Steven L. Martin, Aaron Hertzmann, and Zoran Popović. Style-based inverse kinematics. In *ACM Transactions on Graphics*, volume 23, pages 522–531, New York, NY, USA, 2004. ACM Press.
- [28] Radek Grzeszczuk and Demetri Terzopoulos. Automated learning of muscle-actuated locomotion through control abstraction. In *SIGGRAPH '95 : Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 63–70, New York, NY, USA, 1995. ACM Press.
- [29] Radek Grzeszczuk, Demetri Terzopoulos, and Geoffrey Hinton. Neuroanimator : fast neural network emulation and control of physics-based models. In *SIGGRAPH '98 : Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 9–20, New York, NY, USA, 1998. ACM Press.
- [30] Brian Guenter, Cindy Grimm, Daniel Wood, Henrique Malvar, and Fredric Pighin. Making faces. In *SIGGRAPH '98 : Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 55–66, New York, NY, USA, 1998. ACM.
- [31] Rachel Heck and Michael Gleicher. Parametric motion graphs. In *I3D '07 : Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 129–136, New York, NY, USA, 2007. ACM.
- [32] Jessica K. Hodgins, Wayne L. Wooten, David C. Brogan, and James F. O'Brien. Animating human athletics. In *SIGGRAPH '95 : Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 71–78, New York, NY, USA, 1995. ACM Press.
- [33] Eugene Hsu, Marco da Silva, and Jovan Popović. Guided time warping for motion editing. In *SCA '07 : Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 45–52, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

- [34] Eugene Hsu, Sommer Gentry, and Jovan Popović. Example-based control of human motion. In *SCA '04 : Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 69–77, New York, NY, USA, 2004. ACM Press.
- [35] Eugene Hsu, Kari Pulli, and Jovan Popović. Style translation for human motion. In *ACM Transactions on Graphics*, volume 24, pages 1082–1089, New York, NY, USA, 2005. ACM Press.
- [36] Leslie Ikemoto, Okan Arikan, and David Forsyth. Knowing when to put your foot down. In *SI3D '06 : Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 49–53, New York, NY, USA, 2006. ACM Press.
- [37] Paul M. Isaacs and Michael F. Cohen. Controlling dynamic simulation with kinematic constraints. *SIGGRAPH Comput. Graph.*, 21(4) :215–224, 1987.
- [38] Kenji Kawamura, Akihiro Tokuhiko, and Hideo Takechi. Gait analysis of slope walking : a study of step length, stride width, time factors and deviation in the center of pressure. *Acta Med Okayama*, 45(3) :179–184, 1991.
- [39] Kenneth C. Knowlton. A computer technique for producing animated movies. In *AFIPS '64 (Spring) : Proceedings of the April 21-23, 1964, spring joint computer conference*, pages 67–87, New York, NY, USA, 1964. ACM.
- [40] Taku Komura, Howard Leung, and James Kuffner. Animating reactive motions for biped locomotion. In *VRST '04 : Proceedings of the ACM symposium on Virtual reality software and technology*, pages 32–40, New York, NY, USA, 2004. ACM Press.
- [41] Lucas Kovar and Michael Gleicher. Automated extraction and parameterization of motions in large data sets. *ACM Trans. Graph.*, 23(3) :559–568, 2004.
- [42] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. In *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 473–482, New York, NY, USA, 2002. ACM Press.
- [43] Alexis Lamouret and Michiel van de Panne. Motion synthesis by example. In *Proceedings of the Eurographics workshop on Computer animation and simulation '96*, pages 199–212, New York, NY, USA, 1996. Springer-Verlag New York, Inc.
- [44] Joseph Laszlo, Michiel van de Panne, and Eugene Fiume. Limit cycle control and its application to the animation of balancing and walking. In *SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 155–162, New York, NY, USA, 1996. ACM Press.
- [45] Jehee Lee, Jinxiang Chai, Paul S. A. Reitsma, Jessica K. Hodgins, and Nancy S. Pollard. Interactive control of avatars animated with human motion data. In *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 491–500, New York, NY, USA, 2002. ACM Press.

- [46] Sung-Hee Lee and Demetri Terzopoulos. Heads up! : biomechanical modeling and neuromuscular control of the neck. In *ACM Transactions on Graphics*, volume 25, pages 1188–1198, New York, NY, USA, 2006. ACM Press.
- [47] Alain Leroux, Joyce Fung, and Hugues Barbeau. Postural adaptation to walking on inclined surfaces : I. normal strategies. *Gait Posture*, 15(1) :64–74, 2002.
- [48] Yan Li, Tianshu Wang, and Heung-Yeung Shum. Motion texture : a two-level statistical model for character motion synthesis. In *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 465–472, New York, NY, USA, 2002. ACM.
- [49] C. Karen Liu, Aaron Hertzmann, and Zoran Popović. Learning physics-based motion style with nonlinear inverse optimization. In *ACM Transactions on Graphics*, volume 24, pages 1071–1081, New York, NY, USA, 2005. ACM Press.
- [50] C. Karen Liu, Aaron Hertzmann, and Zoran Popović. Composition of complex optimal multi-character motions. In *SCA '06 : Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, New York, NY, USA, 2006. ACM Press.
- [51] Guodong Liu, Jingdan Zhang, Wei Wang, and Leonard McMillan. Human motion estimation from a reduced marker set. In *I3D '06 : Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 35–42, 2006.
- [52] Zicheng Liu, Steven J. Gortler, and Michael F. Cohen. Hierarchical spacetime control. In *SIGGRAPH '94 : Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 35–42, New York, NY, USA, 1994. ACM Press.
- [53] Adriano Macchietto, Victor Zordan, and Christian R. Shelton. Momentum control for balance. *ACM Trans. Graph.*, 28(3) :1–8, 2009.
- [54] Anthony A. Maciejewski and Charles A. Klein. Sam-animation software for simulating articulated motion. *C&G*, 9 :383–391, 1985.
- [55] James McCann, Nancy Pollard, and Siddhartha Srinivasa. Physics-based motion retiming. In *SCA '06 : Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, New York, NY, USA, 2006. ACM Press.
- [56] Andrew Stuart McIntosh, Karen T. Beatty, Leanne N. Dwan, and Deborah R. Vickers. Gait dynamics on an inclined walkway. *Journal of biomechanics*, 39(13) :2491–2502, 2006.
- [57] Uldarico Muico, Yongjoon Lee, Jovan Popović, and Zoran Popović. Contact-aware nonlinear control of dynamic characters. *ACM Transactions on Graphics*, 28(3), 2009.
- [58] Michael Neff and Eugene Fiume. Modeling tension and relaxation for computer animation. In *SCA '02 : Proceedings of the 2002 ACM SIGGRAPH/Eurographics*

- symposium on Computer animation*, pages 81–88, New York, NY, USA, 2002. ACM Press.
- [59] J. Thomas Ngo and Joe Marks. Spacetime constraints revisited. In *SIGGRAPH '93 : Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 343–350, New York, NY, USA, 1993. ACM Press.
- [60] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.
- [61] Sang Il Park and Jessica K. Hodgins. Capturing and animating skin deformation in human motion. In *SIGGRAPH '06 : ACM SIGGRAPH 2006 Papers*, pages 881–889, New York, NY, USA, 2006. ACM.
- [62] Nancy S. Pollard and Paul S. A. Reitsma. Animation of humanlike characters : Dynamic motion filtering with a physically plausible contact model. In *Yale Workshop on Adaptive and Learning Systems*, 2001.
- [63] Jovan Popović, Steven M. Seitz, Michael Erdmann, Zoran Popović, and Andrew Witkin. Interactive manipulation of rigid body simulations. In *SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 209–217, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [64] Zoran Popović. Controlling physics in realistic character animation. In *Communications of the ACM*, volume 43, pages 50–58, New York, NY, USA, 2000. ACM Press.
- [65] Zoran Popović and Andrew Witkin. Physically based motion transformation. In *SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 11–20, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [66] Katherine Pullen and Christoph Bregler. Motion capture assisted animation : texturing and synthesis. In *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 501–508, New York, NY, USA, 2002. ACM.
- [67] Torsten Reil and Phil Husbands. Evolution of central pattern generators for bipedal walking in real-time physics environment. In *IEEE Transactions on Evolutionary Computation*, volume 6, pages 159–168. IEEE Computer Society, Washington, DC, USA, 2002.
- [68] Paul S. A. Reitsma and Nancy S. Pollard. Evaluating motion graphs for character animation. *ACM Trans. Graph.*, 26(4) :18, 2007.
- [69] Cheng Ren, Liming Zhao, and Alla Safonova. Human motion synthesis with optimization-based graphs. In *Computer Graphics Forum (In Proc. of Eurographics 2010, Sweden)*, volume 29, 2010.

- [70] Lei Ren, Richard K. Jones, and David Howard. Whole body inverse dynamics over a complete gait cycle based only on measured kinematics. *Journal of biomechanics*, 41(12) :2750–2759, 2008.
- [71] Charles Rose, Brian Guenter, Bobby Bodenheimer, and Michael F. Cohen. Efficient generation of motion transitions using spacetime constraints. In *SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 147–154, New York, NY, USA, 1996. ACM Press.
- [72] Alla Safonova and Jessica K. Hodgins. Construction and optimal search of interpolated motion graphs. In *SIGGRAPH '07 : ACM SIGGRAPH 2007 papers*, page 106, New York, NY, USA, 2007. ACM.
- [73] Alla Safonova, Jessica K. Hodgins, and Nancy S. Pollard. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. In *ACM Transactions on Graphics*, volume 23, pages 514–521, New York, NY, USA, 2004. ACM Press.
- [74] Ari Shapiro, Fred Pighin, and Petros Faloutsos. Hybrid control for interactive character animation. In *PG '03 : Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, pages 455–461, Washington, DC, USA, 2003. IEEE Computer Society.
- [75] Dana Sharon and Michiel van de Panne. Synthesis of controllers for stylized planar bipedal walking. In *ICRA 2005 : International Conference on Robotics and Automation*, April 2005.
- [76] Hyun Joon Shin and Hyun Seok Oh. Fat graphs : constructing an interactive character with continuous controls. In *SCA '06 : Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 291–298, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.
- [77] Siobhan C. Strike and Matthew J. D. Taylor. The temporal-spatial and ground reaction impulses of turning gait : Is turning symmetrical? *Gait Posture*, 29(4) :597–602, 2009.
- [78] Adnan Sulejmanpašić and Jovan Popović. Adaptation of performed ballistic motion. In *ACM Transactions on Graphics*, volume 24, pages 165–179, New York, NY, USA, 2005. ACM Press.
- [79] Harold C. Sun and Dimitris N. Metaxas. Automating gait generation. In *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 261–270, New York, NY, USA, 2001. ACM Press.
- [80] J. Sun, M. Walters, N. Svensson, and D. Lloyd. The influence of surface slope on human gait characteristics : a study of urban pedestrians walking on an inclined surface. *Ergonomics*, 39(4) :677–692, 1996.

- [81] Mankyu Sung, Lucas Kovar, and Michael Gleicher. Fast and accurate goal-directed motion synthesis for crowds. In *SCA '05 : Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 291–300, New York, NY, USA, 2005. ACM.
- [82] Seyoon Tak and Hyeong-Seok Ko. A physically-based motion retargeting filter. In *ACM Transactions on Graphics*, volume 24, pages 98–117, New York, NY, USA, 2005. ACM Press.
- [83] Michiel van de Panne, Joe Laszlo, Pedro Huang, and Petros Faloutsos. Dynamic human simulation : Towards agile animated characters. In *IEEE International Conference on Robotics and Automation*, April 2000.
- [84] Kevin Wampler and Zoran Popović. Optimal gait and form for animal locomotion. In *SIGGRAPH '09 : ACM SIGGRAPH 2009 papers*, pages 1–8, New York, NY, USA, 2009. ACM.
- [85] Jane P. Wilhelms and Brian A. Barsky. Using dynamic analysis to animate articulated bodies such as humans and robots. In *Proceedings of Graphics Interface '85 on Computer-generated images : the state of the art*, pages 209–229, New York, NY, USA, 1985. Springer-Verlag New York, Inc.
- [86] Lance Williams. Performance-driven facial animation. In *SIGGRAPH '90 : Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 235–242, New York, NY, USA, 1990. ACM.
- [87] David A. Winter. *Biomechanics and Motor Control of Human Movement*. Wiley-Interscience, 2nd edition, 2000.
- [88] Andrew Witkin and Michael Kass. Spacetime constraints. In *SIGGRAPH '88 : Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 159–168, New York, NY, USA, 1988. ACM Press.
- [89] Andrew Witkin and Zoran Popovic. Motion warping. In *SIGGRAPH '95 : Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 105–108, New York, NY, USA, 1995. ACM.
- [90] Po-Feng Yang, Joe Laszlo, and Karan Singh. Layered dynamic control for interactive character swimming. In *SCA '04 : Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 39–47, New York, NY, USA, 2004. ACM Press.
- [91] KangKang Yin, Stelian Coros, Philippe Beaudoin, and Michiel van de Panne. Continuation methods for adapting simulated skills. In *SIGGRAPH '08 : ACM SIGGRAPH 2008 papers*, pages 1–7, New York, NY, USA, 2008. ACM.
- [92] KangKang Yin, Kevin Loken, and Michiel van de Panne. Simbicon : Simple biped locomotion control. *ACM Trans. Graph.*, 26(3) :Article 105, 2007.

- [93] KangKang Yin, Dinesh K. Pai, and Michiel van de Panne. Data-driven interactive balancing behaviors. In *Pacific Graphics 2005*, October 2005.
- [94] Liming Zhao, Aline Normoyle, Sanjeev Khanna, and Alla Safonova. Automatic construction of a minimum size motion graph. In *SCA '09 : Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 27–35, New York, NY, USA, 2009. ACM.
- [95] Liming Zhao and Alla Safonova. Achieving good connectivity in motion graphs. *Graph. Models*, 71(4) :139–152, 2009.
- [96] Victor B. Zordan and Jessica K. Hodgins. Tracking and modifying upper-body human motion data with dynamic simulation. In *In Computer Animation and Simulation '99*, pages 13–22, 1999.
- [97] Victor B. Zordan and Jessica K. Hodgins. Motion capture-driven simulations that hit and react. In *SCA '02 : Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 89–96, New York, NY, USA, 2002. ACM Press.
- [98] Victor B. Zordan, Anna Majkowska, Bill Chiu, and Matthew Fast. Dynamic response for motion capture animation. In *ACM Transactions on Graphics*, volume 24, pages 697–701, New York, NY, USA, 2005. ACM Press.

Annexe A

Développement de la méthode d'estimation des forces de contact

Nous détaillons dans cette annexe le développement des équations permettant de présenter le sous-problème d'estimation des forces de contact comme un problème de programmation quadratique. À titre de rappel, le sous-problème d'estimation des forces de contact est équivalent au problème initial, à l'exception près que la valeur de la variable x_{fz} est fixé à z . Le problème initial est résolu en trouvant la valeur de z pour laquelle le sous-problème donne la solution optimale.

A.1 Équations de \hat{H} , \hat{m} et b

Pour avoir une fonction de coût quadratique, nous devons la manipuler pour obtenir la forme suivante :

$$F(\hat{x}) = \hat{x}^t \hat{H} \hat{x} + \hat{m}^t \hat{x} + b,$$

C'est le problème que nous abordons dans cette section. Notons que, bien que nous développons ici la valeur de b , celle-ci peut être omise à toutes fins pratiques puisqu'elle ne contribue aucunement au processus d'optimisation.

En utilisant les partitions de coordonnées articulaires décrites dans la section 4.2.1, nous pouvons réécrire la fonction de coût de l'équation 4.8 de la façon suivante

$$F(\hat{x}) = \sum_{i \in A} \hat{f}_i^t \mathcal{S}_i \hat{f}_i + \sum_{i \in B} \hat{f}_i^t \mathcal{S}_i \hat{f}_i + \sum_{i \in C} \hat{f}_i^t \mathcal{S}_i \hat{f}_i + \sum_{i \in D} \hat{f}_i^t \mathcal{S}_i \hat{f}_i \quad (\text{A.1})$$

ce qui nous permet de trouver la forme quadratique des sommes de chaque ensemble séparément.

Pour l'ensemble \mathbb{A} nous avons :

$$\begin{aligned}
 \sum_{i \in \mathbb{A}} \hat{f}_i^t \mathcal{S}_i \hat{f}_i &= \sum_{i \in \mathbb{A}} \left[\left(\hat{h}_i - {}^i \hat{X}_0^* \hat{x} \right)^t \mathcal{S}_i \left(\hat{h}_i - {}^i \hat{X}_0^* \hat{x} \right) \right] \\
 &= \sum_{i \in \mathbb{A}} \left[\left(\hat{h}_i^t - \hat{x}^{ti} \hat{X}_0^{*t} \right) \mathcal{S}_i \left(\hat{h}_i - {}^i \hat{X}_0^* \hat{x} \right) \right] \\
 &= \sum_{i \in \mathbb{A}} \left[\hat{x}^{ti} \hat{X}_0^{*t} \mathcal{S}_i {}^i \hat{X}_0^* \hat{x} - \hat{h}_i^t \mathcal{S}_i {}^i \hat{X}_0^* \hat{x} - \hat{x}^{ti} \hat{X}_0^{*t} \mathcal{S}_i \hat{h}_i + \hat{h}_i^t \mathcal{S}_i \hat{h}_i \right] \\
 &= \sum_{i \in \mathbb{A}} \left[\hat{x}^t \left({}^i \hat{X}_0^{*t} \mathcal{S}_i {}^i \hat{X}_0^* \right) \hat{x} - \left(2 \hat{h}_i^t \mathcal{S}_i {}^i \hat{X}_0^* \right) \hat{x} + \left(\hat{h}_i^t \mathcal{S}_i \hat{h}_i \right) \right] \\
 &= \hat{x}^t \sum_{i \in \mathbb{A}} \left({}^i \hat{X}_0^{*t} \mathcal{S}_i {}^i \hat{X}_0^* \right) \hat{x} - 2 \sum_{i \in \mathbb{A}} \left(\hat{h}_i^t \mathcal{S}_i {}^i \hat{X}_0^* \right) \hat{x} + \sum_{i \in \mathbb{A}} \left(\hat{h}_i^t \mathcal{S}_i \hat{h}_i \right) \\
 &= \hat{x}^t \hat{Q}_A \hat{x} + \hat{m}_A^t \hat{x} + b_A
 \end{aligned}$$

En répétant le même processus pour l'ensemble \mathbb{B} nous obtenons une équation un peu plus complexe.

$$\begin{aligned}
 \sum_{i \in \mathbb{B}} \hat{f}_i^t \mathcal{S}_i \hat{f}_i &= \sum_{i \in \mathbb{B}} \left[\left(\hat{h}_i - {}^i \hat{X}_0^* \hat{c} + {}^i \hat{X}_0^* \hat{x} \right)^t \mathcal{S}_i \left(\hat{h}_i - {}^i \hat{X}_0^* \hat{c} + {}^i \hat{X}_0^* \hat{x} \right) \right] \\
 &= \sum_{i \in \mathbb{B}} \left[\left(\hat{h}_i^t - \hat{c}^{ti} \hat{X}_0^{*t} + \hat{x}^{ti} \hat{X}_0^{*t} \right) \mathcal{S}_i \left(\hat{h}_i - {}^i \hat{X}_0^* \hat{c} + {}^i \hat{X}_0^* \hat{x} \right) \right] \\
 &= \sum_{i \in \mathbb{B}} \left(\hat{h}_i^t \mathcal{S}_i \hat{h}_i - 2 \hat{h}_i^t \mathcal{S}_i {}^i \hat{X}_0^* \hat{c} + 2 \hat{h}_i^t \mathcal{S}_i {}^i \hat{X}_0^* \hat{x} + \hat{c}^{ti} \hat{X}_0^{*t} \mathcal{S}_i {}^i \hat{X}_0^* \hat{c} \right. \\
 &\quad \left. - 2 \hat{c}^{ti} \hat{X}_0^{*t} \mathcal{S}_i {}^i \hat{X}_0^* \hat{x} + \hat{x}^{ti} \hat{X}_0^{*t} \mathcal{S}_i {}^i \hat{X}_0^* \hat{x} \right) \\
 &= \hat{x}^t \sum_{i \in \mathbb{B}} \left({}^i \hat{X}_0^{*t} \mathcal{S}_i {}^i \hat{X}_0^* \right) \hat{x} + 2 \sum_{i \in \mathbb{B}} \left(\hat{h}_i^t \mathcal{S}_i {}^i \hat{X}_0^* - \hat{c}^{ti} \hat{X}_0^{*t} \mathcal{S}_i {}^i \hat{X}_0^* \right) \hat{x} \\
 &\quad + \sum_{i \in \mathbb{B}} \left(\hat{h}_i^t \mathcal{S}_i \hat{h}_i + \hat{c}^{ti} \hat{X}_0^{*t} \mathcal{S}_i {}^i \hat{X}_0^* \hat{c} - 2 \hat{h}_i^t \mathcal{S}_i {}^i \hat{X}_0^* \hat{c} \right) \\
 &= \hat{x}^t \hat{Q}_B \hat{x} + \hat{m}_B^t \hat{x} + b_B
 \end{aligned}$$

Pour l'ensemble \mathbb{C} nous avons une forme identique à celle de l'ensemble \mathbb{A} , en sub-

stituant \hat{x} par \hat{c} , mais dont le résultat ne dépend pas du choix de \hat{x} :

$$\begin{aligned} \sum_{i \in C} \hat{f}_i^t \mathcal{S}_i \hat{f}_i &= \sum_{i \in C} \left[\left(\hat{h}_i - {}^i \hat{X}_0^* \hat{c} \right)^t \mathcal{S}_i \left(\hat{h}_i - {}^i \hat{X}_0^* \hat{c} \right) \right] \\ &= \hat{c}^t \sum_{i \in C} \left({}^i \hat{X}_0^{*t} \mathcal{S}_i {}^i \hat{X}_0^* \right) \hat{c} - 2 \sum_{i \in C} \left(\hat{h}_i^t \mathcal{S}_i {}^i \hat{X}_0^* \right) \hat{c} + \sum_{i \in C} \left(\hat{h}_i^t \mathcal{S}_i \hat{h}_i \right) \\ &= b_C \end{aligned}$$

Finalement, pour l'ensemble \mathbb{D} , nous avons simplement :

$$\begin{aligned} \sum_{i \in \mathbb{D}} \hat{f}_i^t \mathcal{S}_i \hat{f}_i &= \sum_{i \in \mathbb{D}} \left(\hat{h}_i^t \mathcal{S}_i \hat{h}_i \right) \\ &= b_D \end{aligned}$$

Pour obtenir \hat{H} , \hat{m} et b , il suffit d'additionner les termes obtenus plus haut. Nous avons donc :

$$\begin{aligned} \hat{H} &= \hat{Q}_A + \hat{Q}_B \\ \hat{m} &= \hat{m}_A + \hat{m}_B \\ b &= b_A + b_B + b_C + b_D \end{aligned}$$

que l'on peut développer et simplifier pour obtenir nos équations finales :

$$\begin{aligned} \hat{H} &= \sum_{i \in A \cup B} {}^i \hat{X}_0^{*t} \mathcal{S}_i {}^i \hat{X}_0^* \\ \hat{m}^t &= 2 \sum_{i \in B} \left(\hat{h}_i^t \mathcal{S}_i {}^i \hat{X}_0^* - \hat{c}^t {}^i \hat{X}_0^{*t} \mathcal{S}_i {}^i \hat{X}_0^* \right) - 2 \sum_{i \in A} \hat{h}_i^t \mathcal{S}_i {}^i \hat{X}_0^* \\ b &= \sum_{i \in B \cup C} \left(\hat{c}^t {}^i \hat{X}_0^{*t} \mathcal{S}_i {}^i \hat{X}_0^* \hat{c} - 2 \hat{h}_i^t \mathcal{S}_i {}^i \hat{X}_0^* \hat{c} \right) + \sum_{i \in \Omega} \hat{h}_i^t \mathcal{S}_i \hat{h}_i \end{aligned}$$

A.2 Équations des contraintes linéaires

Tel que nous l'avons présenté dans la section 4.2.2, les vecteurs \vec{u} et \vec{v} , respectivement les centres de pression sous le pied droit et sous le pied gauche, doivent répondre aux inégalités suivantes

$$-\vec{\delta} \leq {}^\alpha R_0 \vec{u} + {}^\alpha \vec{t}_0 \leq \vec{\delta} \quad (\text{A.2})$$

$$-\vec{\delta} \leq {}^\beta R_0 \vec{v} + {}^\beta \vec{t}_0 \leq \vec{\delta} \quad (\text{A.3})$$

afin d'être contenus à l'intérieur des surfaces de contact entre les pieds et le sol. Dans cette section, nous développons ces contraintes pour en obtenir une formulation linéaire fonction de \hat{x} en utilisant le fait que les centres de pression peuvent s'exprimer de la façon suivante :

$$\vec{u} = \begin{pmatrix} -\frac{x_{\tau y}}{x_{fz}} \\ \frac{x_{\tau x}}{x_{fz}} \\ 0 \end{pmatrix}$$

$$\vec{v} = \begin{pmatrix} -\frac{c_{\tau y} - x_{\tau y}}{c_{fz} - x_{fz}} \\ \frac{c_{\tau x} - x_{\tau x}}{c_{fz} - x_{fz}} \\ 0 \end{pmatrix}$$

et que x_{fz} doit être égal à la constante z .

A.2.1 Pied droit

En substituant \vec{u} par $(-x_{\tau y}/x_{fz}, x_{\tau x}/x_{fz}, 0)^t$ dans l'équation A.2, nous obtenons

$$-\vec{\delta} \leq \alpha R_0 \begin{pmatrix} -x_{\tau y}/x_{fz} \\ x_{\tau x}/x_{fz} \\ 0 \end{pmatrix} + \alpha \vec{t}_0 \leq \vec{\delta}$$

$$-\vec{\delta} - \alpha \vec{t}_0 \leq \alpha R_0 \begin{pmatrix} -x_{\tau y}/z \\ x_{\tau x}/z \\ 0 \end{pmatrix} \leq \vec{\delta} - \alpha \vec{t}_0$$

$$-z (\vec{\delta} + \alpha \vec{t}_0) \leq \alpha R_0 \begin{pmatrix} -x_{\tau y} \\ x_{\tau x} \\ 0 \end{pmatrix} \leq z (\vec{\delta} - \alpha \vec{t}_0)$$

Nous avons donc un total de six contraintes, données par les deux inégalités suivantes :

$$\alpha R_0 \begin{pmatrix} -x_{\tau y} \\ x_{\tau x} \\ 0 \end{pmatrix} \leq z (\vec{\delta} - \alpha \vec{t}_0)$$

$$\alpha R_0 \begin{pmatrix} x_{\tau y} \\ -x_{\tau x} \\ 0 \end{pmatrix} \leq z (\vec{\delta} + \alpha \vec{t}_0)$$

que nous pouvons reformuler de la façon suivante :

$$\begin{pmatrix} \alpha_{1,2} & -\alpha_{1,1} & 0 & 0 & 0 & 0 \\ \alpha_{2,2} & -\alpha_{2,1} & 0 & 0 & 0 & 0 \\ \alpha_{3,2} & -\alpha_{3,1} & 0 & 0 & 0 & 0 \\ -\alpha_{1,2} & \alpha_{1,1} & 0 & 0 & 0 & 0 \\ -\alpha_{2,2} & \alpha_{2,1} & 0 & 0 & 0 & 0 \\ -\alpha_{3,2} & \alpha_{3,1} & 0 & 0 & 0 & 0 \end{pmatrix} \hat{x} \leq z \begin{pmatrix} \vec{\delta} - \alpha \vec{t}_0 \\ \vec{\delta} + \alpha \vec{t}_0 \end{pmatrix}. \quad (\text{A.4})$$

A.2.2 Pied gauche

En substituant \vec{v} par $(-(c_{\tau y} - x_{\tau y})/(c_{fz} - x_{fz}), (c_{\tau x} - x_{\tau x})/(c_{fz} - x_{fz}), 0)^t$ dans l'équation A.3, nous obtenons

$$\begin{aligned} -\vec{\delta} &\leq \beta R_0 \begin{pmatrix} \frac{-c_{\tau y} - x_{\tau y}}{c_{fz} - x_{fz}} \\ \frac{c_{\tau x} - x_{\tau x}}{c_{fz} - x_{fz}} \\ 0 \end{pmatrix} + \beta \vec{t}_0 \leq \vec{\delta} \\ -\vec{\delta} - \beta \vec{t}_0 &\leq \beta R_0 \begin{pmatrix} \frac{-c_{\tau y} - x_{\tau y}}{c_{fz} - z} \\ \frac{c_{\tau x} - x_{\tau x}}{c_{fz} - z} \\ 0 \end{pmatrix} \leq \vec{\delta} - \beta \vec{t}_0 \\ -(c_{fz} - z) (\vec{\delta} + \beta \vec{t}_0) &\leq \beta R_0 \begin{pmatrix} -c_{\tau y} + x_{\tau y} \\ c_{\tau x} - x_{\tau x} \\ 0 \end{pmatrix} \leq (c_{fz} - z) (\vec{\delta} - \beta \vec{t}_0) \\ -(c_{fz} - z) (\vec{\delta} + \beta \vec{t}_0) - k &\leq \beta R_0 \begin{pmatrix} x_{\tau y} \\ -x_{\tau x} \\ 0 \end{pmatrix} \leq (c_{fz} - z) (\vec{\delta} - \beta \vec{t}_0) - k \end{aligned}$$

$$\text{avec } k = \beta R_0 \begin{pmatrix} -c_{\tau y} \\ c_{\tau x} \\ 0 \end{pmatrix}.$$

Les six contraintes du centre de pression sous le pied gauche sont données par les

deux inégalités suivantes :

$$\begin{aligned} {}^\beta R_0 \begin{pmatrix} x_{\tau y} \\ -x_{\tau x} \\ 0 \end{pmatrix} &\leq (c_{fz} - z) (\vec{\delta} - \beta \vec{t}_0) - k \\ {}^\beta R_0 \begin{pmatrix} x_{\tau y} \\ -x_{\tau x} \\ 0 \end{pmatrix} &\leq (c_{fz} - z) (\vec{\delta} + \beta \vec{t}_0) + k \end{aligned}$$

que nous pouvons reformuler de la façon suivante :

$$\begin{pmatrix} -\beta_{1,2} & \beta_{1,1} & 0 & 0 & 0 & 0 \\ -\beta_{2,2} & \beta_{2,1} & 0 & 0 & 0 & 0 \\ -\beta_{3,2} & \beta_{3,1} & 0 & 0 & 0 & 0 \\ \beta_{1,2} & -\beta_{1,1} & 0 & 0 & 0 & 0 \\ \beta_{2,2} & -\beta_{2,1} & 0 & 0 & 0 & 0 \\ \beta_{3,2} & -\beta_{3,1} & 0 & 0 & 0 & 0 \end{pmatrix} \hat{x} \leq (c_{fz} - z) \begin{pmatrix} \vec{\delta} - \beta \vec{t}_0 \\ \vec{\delta} + \beta \vec{t}_0 \end{pmatrix} + \begin{pmatrix} -k \\ k \end{pmatrix}. \quad (\text{A.5})$$

A.2.3 Contraintes combinées

Finalement, nous pouvons regrouper les inégalités A.4 et A.5 en une seule pour obtenir

$$\begin{pmatrix} \alpha_{1,2} & -\alpha_{1,1} & 0 & 0 & 0 & 0 \\ \alpha_{2,2} & -\alpha_{2,1} & 0 & 0 & 0 & 0 \\ \alpha_{3,2} & -\alpha_{3,1} & 0 & 0 & 0 & 0 \\ -\alpha_{1,2} & \alpha_{1,1} & 0 & 0 & 0 & 0 \\ -\alpha_{2,2} & \alpha_{2,1} & 0 & 0 & 0 & 0 \\ -\alpha_{3,2} & \alpha_{3,1} & 0 & 0 & 0 & 0 \\ -\beta_{1,2} & \beta_{1,1} & 0 & 0 & 0 & 0 \\ -\beta_{2,2} & \beta_{2,1} & 0 & 0 & 0 & 0 \\ -\beta_{3,2} & \beta_{3,1} & 0 & 0 & 0 & 0 \\ \beta_{1,2} & -\beta_{1,1} & 0 & 0 & 0 & 0 \\ \beta_{2,2} & -\beta_{2,1} & 0 & 0 & 0 & 0 \\ \beta_{3,2} & -\beta_{3,1} & 0 & 0 & 0 & 0 \end{pmatrix} \hat{x} \leq \begin{pmatrix} z \begin{pmatrix} \vec{\delta} - \alpha \vec{t}_0 \\ \vec{\delta} + \alpha \vec{t}_0 \end{pmatrix} \\ (c_{fz} - z) \begin{pmatrix} \vec{\delta} - \beta \vec{t}_0 \\ \vec{\delta} + \beta \vec{t}_0 \end{pmatrix} + \begin{pmatrix} -k \\ k \end{pmatrix} \end{pmatrix}, \quad (\text{A.6})$$

Avec la contrainte d'égalité $x_{fz} = z$, ces équations constituent les contraintes linéaires de notre sous-problème.