

## *Thème 2 : Mouvement et contrôle de mouvement*

Contributions de :

N. Courty (VALORIA)

G. Dumont (IRISA)

S. Gibet (VALORIA)

D. Gillard (LERI)

J.-P. Jessel (IRIT)

F. Multon (LPBEM)

## **Première Partie : Cinématique, capture et édition de mouvement**

### **1 Introduction**

L'animation d'humains virtuels est un problème récurrent en synthèse d'images. Deux approches ont été développées au cours des dernières années. La première vise à améliorer les techniques de cinématique inverse et de manipulation d'images clefs pour augmenter le contrôle de l'infographiste sur l'animation. La seconde approche se focalisait sur la génération automatique de mouvements à partir de la simulation des forces animant un corps humain. Bien que donnant des résultats acceptables visuellement, elles atteignaient vite leurs limites, lorsque la complexité de l'humanoïde géré devenait trop importante. En effet un modèle complexe est très difficile à manipuler par des techniques d'images clefs ou de cinématique inverse, même pour un infographiste expérimenté. Et la seconde approche offre dans ce cas des temps de calculs beaucoup trop importants et une maîtrise aléatoire des paramètres.

C'est alors que sont apparus les systèmes de capture de mouvement qui fournissent des animations de grandes qualités. Ils ont offert un moyen confortable de produire des animations réalistes. On les a utilisés au cours de nombreuses occasions comme, par exemple, dans la production d'effets spéciaux au cinéma, dans l'animation de shows télévisés, ou encore dans les jeux vidéo.

On a vite ressenti le besoin de créer des outils permettant d'éditer, de contrôler et d'optimiser les animations ainsi produites. C'est pourquoi nous allons exposer une manière pour le faire qui offre de nombreux avantages et qui utilise un système d'optimisation avec contraintes.

Nous ferons tout d'abord un panorama général des méthodes d'édition du mouvement d'un humain virtuel. Nous approfondirons celles utilisant des contraintes, en détaillant les méthodes d'optimisations utilisées dans ce cas. Nous parlerons ensuite des différentes modifications que l'on peut faire, sur une animation provenant d'un système de capture de mouvement.

### **2 Etat de l'art**

#### ***2.1 Introduction***

L'animation de personnage de synthèse est un domaine de recherche qui s'est développé car le besoin s'est autant fait ressentir dans l'industrie désireuse de plus en plus de réalisme, que pour les chercheurs souhaitant détailler le fonctionnement de l'homme, et valider leurs théories par la simulation. Nous allons exposer les différentes techniques existantes traitant de l'animation de personnage, cela nous permettra de situer l'animation par contrainte par rapport aux autres techniques. Nous allons tout d'abord parler de quelques généralités sur l'animation de personnage, c'est-à-dire, faire une description des structures utilisées, suivie d'un rapide tour d'horizon des différentes manières d'éditer le mouvement. Puis nous aborderons les méthodes de capture de mouvement sans retraitement, suivies des procédés d'analyse et de génération automatique du mouvement. Nous exposerons les différentes méthodes d'optimisation existantes, pour résoudre les problèmes rencontrés. Nous finirons par une étude de l'utilisation de ces contraintes dans l'animation de personnage.

## 2.2 Animation de personnage

Une des méthodes qui est sans doute la plus utilisée repose sur le talent artistique des infographistes. Nous présenterons tout d'abord le modèle de base utilisé pour faciliter l'animation de personnage, qu'est le squelette. Nous parlerons ensuite des facilitées offertes pour le travail de l'animateur, grâce à la cinématique inverse et à l'animation par images clés. Et, pour finir, nous exposerons les avantages, ainsi que les inconvénients offerts par ces techniques.

### 2.2.1 Représentation d'un personnage

Les méthodes d'animation de personnage utilisent, comme beaucoup de techniques d'animation, une version simplifiée de l'objet à animer [1]. En effet il est beaucoup plus simple de manipuler ce qui anime la matière que la matière elle-même. C'est ainsi que l'on laisse le soin à l'ordinateur de calculer les effets complexes des déplacements des muscles et des os sur chaque partie visible de la peau du personnage. Malheureusement, la simulation des muscles est complexe dans l'animation d'un personnage, puisqu'un corps est composé d'un nombre très important de muscles et d'os. Par exemple la marche n'utilise pas moins de 90 muscles, et 55 os. Cette simulation est aussi complexifiée par la difficulté de calculer l'effet des muscles sur la peau en un temps interactif, en fonction de leurs contractions, de leurs positions, de leurs volumes.... C'est pourquoi, au lieu d'essayer de modéliser l'ensemble des systèmes du corps, déclencheur du mouvement du personnage, on simplifie ces phénomènes, en utilisant les éléments à la fois les plus représentatifs du mouvement et les plus facilement manipulables puisque rigides : les os. On les utilisera donc comme référentiel pour le déplacement de la peau en fonction des mouvements du personnage.



*Fig. 1 Représentation d'un squelette simplifié d'un personnage animé. La structure est constituée d'un ensemble de segments, ici représenté en 3D. Pour rendre plus lisible les rotations, les os sont représentés par des octaèdres.*

Même si cette représentation est simplifiée par rapport à la complexité réelle d'un modèle complet d'animation de personnage, elle offre un degré d'erreur suffisamment faible, pour fournir une animation perçue comme réaliste, en temps interactif, voire temps réel. Un personnage est donc représenté par sa structure squelettique (Fig. 1). Ainsi, un os (ou « bone » en anglais) est remplacé par un modèle simplifié, c'est-à-dire par un segment. La simplification du modèle peut aller jusqu'à la diminution du nombre d'os utilisés pour représenter le personnage, ce qui revient à ne plus s'attacher à la réalité anatomique du personnage, mais plutôt à trouver une représentation qui synthétise leurs actions sur le mouvement.

Comme chaque os est relié à un ou plusieurs os par des liaisons, on peut représenter cette structure par un modèle hiérarchique, dont chaque nœud est un os et chaque liaison est un joint qui relie deux os. Il suffit de définir arbitrairement un os comme étant le parent de tous les os, et chaque os aura sa position définie en fonction de la position son os parent (Fig. 2).

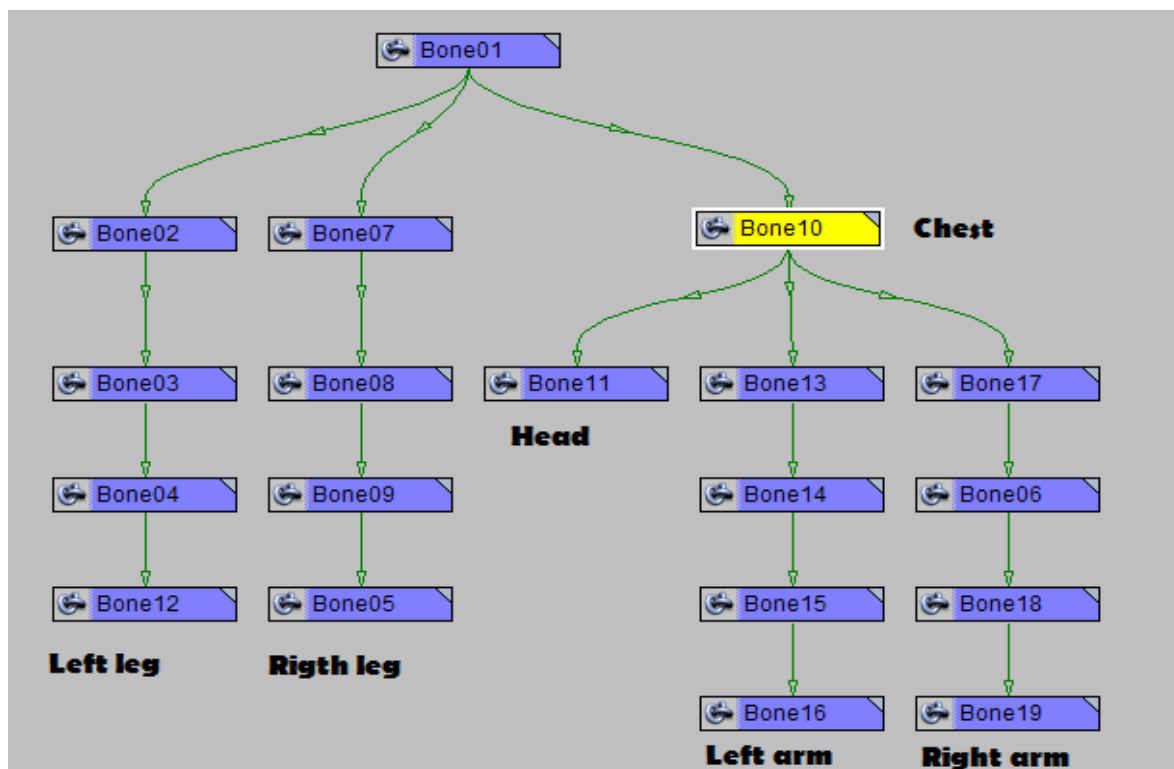


Fig. 2 Représentation schématique de la hiérarchie d'un squelette assez simple. Le bassin est la racine de la hiérarchie.

Pour animer le personnage il suffira de connaître la rotation de chaque os, par rapport à son parent au cours du temps. Le déplacement effectué par chaque nœud en fonction de son parent, est toujours égal à la taille de l'os, et est donc constant, puisqu'on suppose que l'os ne varie pas de longueur au cours du temps. Seule la translation de l'os qui fait office de racine hiérarchique est importante puisqu'elle définit la position générale du personnage dans l'espace de la scène (position absolue).

Cette structure hiérarchique offre plusieurs avantages. Elle conceptualise le squelette par rapport à ses joints, et non plus par rapport à la position des os. En effet, chaque position est donc relative au précédent et non plus de façon absolue dans la scène, et dépend uniquement de la configuration du joint (à part la racine). Mais elle permet aussi d'ajouter facilement des limites d'angles. Ces limites sont liées à la capacité du corps humain à exécuter ou non certaines rotations avec une certaine amplitude maximale. Cela offre donc la possibilité d'inclure plus d'information dans le squelette, en restant facilement contrôlable. Une fois la position des os calculée et contrôlée, on peut habiller ce squelette par une couche de peau.

La position de chaque partie de la peau qui est représentée en synthèse d'image, par des polygones (plus généralement par des triangles), va donc dépendre des os. On utilise pour cela, deux manières de faire dépendre le mouvement de la peau en fonction des os choisis. Ces deux manières sont plus ou moins complexes, et donc choisies en fonction de la puissance de traitement fournie, pour offrir une visualisation en temps réel.

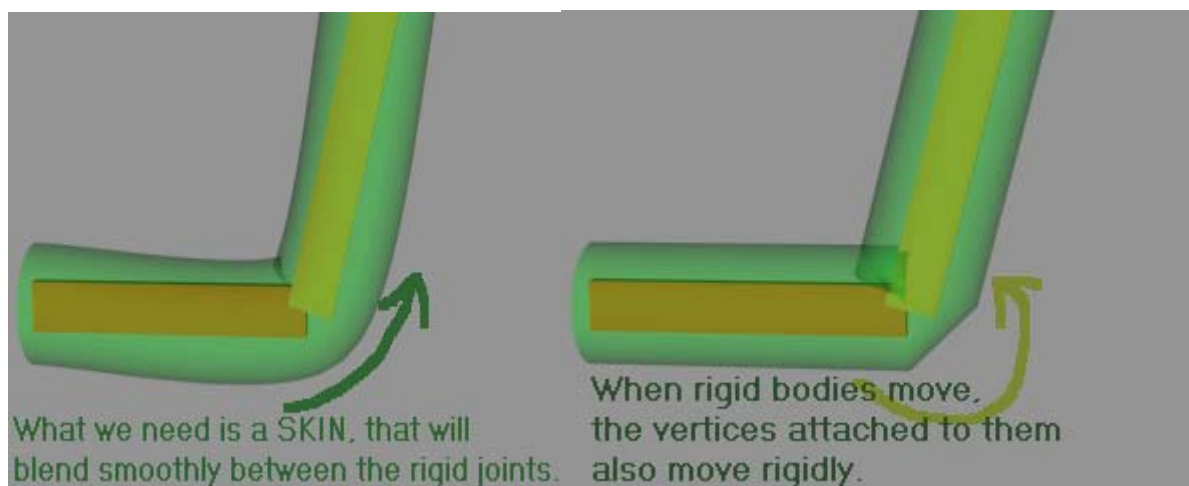


Fig. 3 A droite, la pliure de la peau, influencée par les os, en utilisant la méthode d'influence pondérée. A gauche, le même cas, sans pondération.

La première manière [1] consiste à lier chaque élément de la peau avec un et un seul os. Lorsque l'os se déplacera, la peau qui dépend de cet os se déplacera aussi. Cette méthode offre un rendu tout à fait correct tant que l'articulation n'est pas trop pliée. Dès que l'angle minimal entre les deux segments est trop petit, la jointure présente des problèmes d'interpénétration de la peau (fig. 3). La seconde méthode [2] fait dépendre la position de la peau non plus d'un, mais de plusieurs os. Ainsi chaque os influence de manière pondérée avec les autres os une partie de la peau (fig. 3). Cela demande évidemment plus de puissance de calcul pour pouvoir afficher le personnage en temps réel, mais les déformations sont plus réalistes, et ont moins de problème d'interpénétration.

Cette manière simplifiée de représenter le personnage offre, en plus de l'accélération du traitement, permettant une visualisation temps réel, de simplifier sa manipulation en vue de la modifier ou de lui apporter de nouvelles propriétés.

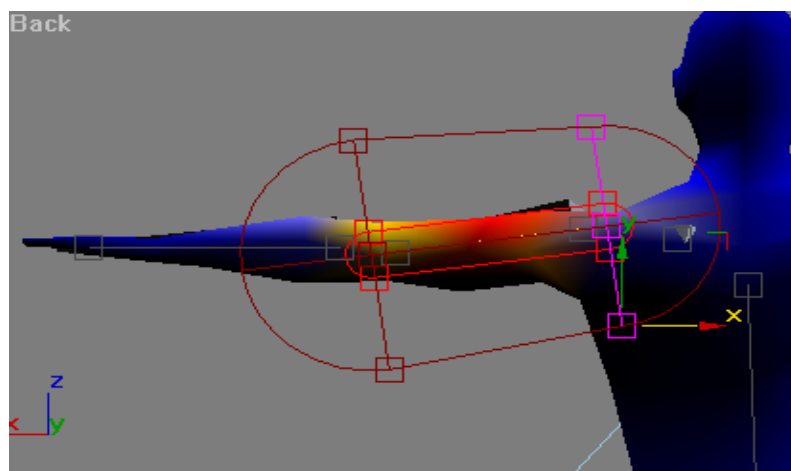


Fig. 4 Un os et sa zone d'influence. Plus la coloration est rouge, plus l'importance de l'influence de l'os sélectionné est importante. La zone d'influence de l'os est ici représentée par une forme ovoïde.

### 2.2.2 Édition de mouvements

Comme un personnage doit pouvoir bouger, il faut bien sûr créer son mouvement. Plusieurs possibilités s'offrent à nous : soit en utilisant un mouvement réel, et mesurer son évolution, pour en faire une copie, soit en analysant le mouvement, pour extraire ses

composantes et ensuite les combiner pour en produire de nouveaux, soit en « dessinant » le mouvement. C'est cette dernière technique, que nous allons aborder dans cette première partie. Nous allons tout d'abord voir comment la méthode d'interpolation par image clef procure une aide précieuse, puis nous parlerons de deux méthodes permettant de décrire le mouvement, que sont la cinématique directe et la cinématique inverse.

### 2.2.2.1 Images clefs

La plus simple des manières, pour modifier un mouvement consiste à modifier ce mouvement « à la main ». Cette méthode est héritée de l'animation 2D traditionnelle, utilisée dans les dessins animés. Un animateur expérimenté dessinait les principales images décrivant le mouvement, et d'autres dessinateurs complétaient l'animation en dessinant les images intermédiaires. Il fut évident d'utiliser l'ordinateur comme moyen de créer ces images intermédiaires et ainsi d'éviter ce travail non essentiel à la description du mouvement comme l'ont utilisé Burtnyk et Wein [3] dès les années soixante-dix. Cette méthode rapportée dans un contexte de travail assisté par ordinateur, consiste à définir la position, l'orientation, ou tout autre paramètre, à des instants donnés. L'ordinateur se chargera de la tâche de compléter les vides dans l'animation, en calculant l'évolution de la valeur des paramètres, entre deux temps pour lesquels, les valeurs ont été précédemment définies par l'animateur.

Pour trouver ces valeurs intermédiaires entre les valeurs définies, on utilise des techniques d'interpolations. La plus simple des techniques est l'interpolation linéaire, qui fait évoluer le paramètre, comme son nom l'indique, de manière linéaire. Mais d'autres techniques, comme les interpolations cubiques (B-Spline, Bézier) un peu plus complexes, offrent par contre des mouvements plus lissés et peuvent permettre de réduire le nombre d'instantanés clefs à définir pour la même qualité d'animation.

L'animation de personnage par image clef a de nombreux avantages, et est pour cela, encore utilisée aujourd'hui de façon très importante dans le monde de la production d'images de synthèse. Elle est très simple d'utilisation, demande peu de puissance de calcul et, étant un héritage de l'animation 2D, est rapidement compréhensible par la plupart des artistes d'animation traditionnelle. Un autre de ses avantages est qu'elle laisse une liberté totale à l'artiste qui peut produire tant des animations réalistes que des animations dans le style des dessins animés.

Par contre le résultat qu'elle produit dépend fortement du talent de l'artiste, puisque c'est lui qui doit imaginer et synthétiser le mouvement, pour ensuite le retranscrire le plus fidèlement possible en 3D. Et la plupart du temps l'artiste a du mal à reproduire un réalisme suffisant. Ceci est en partie dû, au fait qu'il va schématiser les différentes composantes du mouvement, et va donc reproduire le mouvement dans son ensemble, en oubliant certains détails du mouvement, comme les légers tremblements des muscles par exemple. Mais les difficultés que rencontrent un artiste, lors de la création d'un mouvement réaliste, tiennent au fait que le spectateur est un observateur exercé, et qu'il décèle n'importe quelle imperfection. En effet, nous, en tant qu'humain, nous rencontrons et nous voyons en permanence d'autres humains se déplacer, et nous mémorisons donc, inconsciemment, chaque détail du mouvement. Nous pouvons par exemple reconnaître quelqu'un de dos, seulement à la façon dont il marche. Cette importante capacité à reconnaître les mouvements est par conséquent difficile à mettre en défaut. Et produire une illusion suffisamment proche de la réalité est, par conséquent, assez complexe. Pour faciliter cette tâche, il y a d'autres méthodes que l'animation par images clefs, qui permettent de décrire le mouvement.

### 2.2.2.2 Cinématique directe

La cinématique directe consiste à décrire l'évolution du mouvement de la géométrie. Dans notre cas le mouvement s'applique à un personnage formé de pièces articulées par des joints constituant un ensemble hiérarchique. Son mouvement est principalement défini par des rotations, car la longueur de chaque segment est définie et fixe, les angles de rotation seuls, permettent de trouver la position locale des éléments terminaux, en parcourant l'arbre hiérarchique et en accumulant les rotations. Pour animer un modèle, il suffit donc de décrire la manière dont doivent évoluer les différents angles. Il existe de nombreuses manières efficaces et numériquement correctes pour définir ces angles. Sous forme d'équation, cela revient à résoudre :

$$X = f(\theta) ; \text{ avec :}$$

$X = (X_x, X_y, X_z, X_r, X_q, X_y) :$  position terminale d'un élément du squelette

$\theta = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \dots) :$  vecteur des angles d'articulation.

Cette méthode est simple à mettre en œuvre. Cette simplicité du traitement nécessaire à son application permet de manipuler les personnages en temps réel, offrant ainsi une pré visualisation du résultat final très intuitive. Malheureusement sa manipulation devient très complexe lorsque les chaînes hiérarchiques du squelette deviennent trop longues. En effet, la modification d'un angle d'un des joints modifie la position de tous les éléments inférieurs dans la hiérarchie. Pour éviter ce problème, une autre technique est disponible : la cinématique inverse.

### 2.2.2.3 Cinématique inverse

Cette technique consiste à partir plutôt du résultat voulu et à laisser le soin à l'ordinateur de calculer le meilleur mouvement possible des angles pour y parvenir. Cette méthode est utilisée pour contrôler certaines parties d'un squelette, comme par exemple les mains, ou les pieds, Cela simplifie énormément le travail de l'animateur, puisque pour bouger par exemple le bras, il lui suffit de désigner la position dans laquelle la main devra se trouver, et l'ordinateur calcule le reste. D'un point de vue mathématique cela revient à faire l'inverse de la cinématique directe. Au lieu de partir des angles des articulations, et de trouver la position des éléments, on part des positions voulues, pour ensuite reconstruire les angles des rotations nécessaires à ce résultat. Cela se traduit mathématiquement par l'équation suivante, il faut donc trouver la fonction  $f^{-1}$  vérifiant cette équation :

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

Le modèle cinématique inverse est basé sur l'évaluation des variations instantanées de la position des effecteurs terminaux en fonction de celles de chacun des paramètres de liaison. Dans ce cas nous pouvons linéariser, et nous pouvons évaluer la petite variation des angles en fonction de la variation des positions effecteurs terminaux (comme par exemple les mains) à l'aide de l'équation suivante :

$$\Delta X = J \Delta \theta ; \text{ avec :}$$

$J :$  matrice jacobienne carré régulière du système.

$X :$  situation de l'effecteur terminal.

$\theta :$  configuration articulaire du squelette.

Pour trouver  $\Delta \theta$ , on peut exprimer l'équation de cette manière :  $\Delta X = J^{-1} \Delta \theta$

Le problème est que la matrice  $J$  est souvent non inversible, à cause des dimensions différentes de  $X$  et de  $\theta$ . En effet pour une configuration des effecteurs finaux, il peut exister

une infinité de configurations angulaires la vérifiant (redondance). Nous présentons trois techniques classiques pour résoudre cette difficulté.

### *Approches par pseudo-inverse*

La première méthode couramment utilisée en cinématique inverse est celle des inverses généralisés et plus particulièrement des pseudo inverses. Grâce à un algorithme de décomposition en valeurs singulières cette méthode présente l'avantage de fournir une solution de norme minimale lorsqu'elle existe et sinon, une solution optimale au sens des moindres carrés. L'introduction de la pseudo inverse nous permet d'introduire une tâche secondaire, sans gêner la réalisation de la tâche principale, qu'est la cinématique inverse. Cette tâche secondaire correspond à une optimisation qui peut être de tout type. Cela donne l'équation suivante :

$$\Delta\theta = J^+ \Delta X + \alpha (I + J^+ J) \Delta z ; \text{ avec : } J^+ : \text{pseudo inverse de la matrice jacobienne } J$$

$I$  : matrice identité

$\alpha$  : constante de pénalité

$\Delta z$  : contrainte à minimiser qui correspond à la tâche secondaire (par exemple contrainte prenant en compte les angles limites pour minimiser un critère énergétique).

Nous pouvons donc évaluer les différents angles respectant la position à atteindre de l'effecteur terminal, tout en essayant de minimiser la violation de la contrainte ainsi introduite. Le problème est que cette méthode est assez lourde à calculer, et peut être instable devant certaines singularités [4].

C'est pourquoi d'autres méthodes existent pour résoudre le problème de la cinématique inverse, par exemple la méthode Cyclic Coordinate Descent (CCD) s'appuie sur un principe d'itération de type Jacobi ou Gauss-Seidel. A chaque itération on cherche à minimiser l'erreur de position de l'effecteur final, en ajustant chaque angle. Ainsi on démarre de l'effecteur final pour remonter jusqu'à la racine. Et, pour chaque joint, on cherche l'angle qui permet à l'effecteur final, d'être le plus proche possible de son but. On réitère cette opération jusqu'à ce que l'erreur soit suffisamment faible, ou que le nombre d'itérations maximal soit atteint (dans ce cas, il est fort probable que le but ne peut pas être atteint par l'effecteur).

Cette méthode est rapide et l'on peut y adjoindre des contraintes, comme des angles limites, en empêchant la valeur angulaire de certain joint de dépasser une certaine limite, au cours des itérations. Elle a pour avantage sa simplicité d'implantation, sa rapidité d'exécution dans la majorité des cas, et sa stabilité pour les configurations des singularités, contrairement à la méthode à base de pseudo inverse. Mais elle possède quelques inconvénients, comme par exemple le fait de favoriser des joints finaux, à cause du sens de traitement des joints qui part des effecteurs finaux, pour remonter vers la racine. En effet les angles des joints finaux sont les premiers à être modifiés, et par conséquent, ont plus de chance de subir la rotation la plus importante. Cette méthode possède aussi comme désavantage, de ne pas toujours produire des mouvements naturels. On peut réduire cet effet indésirable en limitant la variation angulaire, lors de chaque itération.

La recherche en animation de méthodes favorisant l'interaction avec l'environnement oriente la recherche de solutions vers des méthodes pouvant s'adapter en temps réel au milieu d'évolution des humanoïdes de synthèse. De manière générale, le contrôle de structures



articulées virtuelles nécessite d'accélérer la résolution de l'algorithme tout en garantissant le respect du réalisme du mouvement généré [5], [6], [8], [9].

### **Approches analytiques**

Des approches analytiques ont été suggérées, qui introduisent des contraintes géométriques sur la structure articulée afin de réduire le nombre de degrés de liberté. Ces méthodes permettent de trouver toutes les solutions au problème de l'inversion. C'est le cas des méthodes d'inversion cinématique qui s'appliquent à des chaînes HAL (Human-Arm-Like). Une revue synthétique de ces méthodes s'appliquant à des chaînes cinématiques contraintes est présentée par Tolani *et al.* [9].

### **Programmation non linéaire**

Un certain nombre d'approches proposent de transformer le problème de cinématique inverse en un problème de programmation non linéaire [10], [11], [12], [14]. A la sortie  $X$  les auteurs associent une fonction potentielle  $P$  exprimant la distance entre la position de l'extrémité de l'effecteur et l'objectif à atteindre.

$$\begin{aligned} P : S &\rightarrow R^+ \\ X &\rightarrow P(X) \in R^+ \end{aligned}$$

Le problème d'inversion est ramené à la minimisation de la fonction  $G(\theta) = P(X)$ . La mise à jour du vecteur d'état  $\theta$  est réalisée par une technique de descente du gradient :

$$\delta\theta = \Phi(\nabla_{\theta}G(\theta)) \quad \Bigg| \quad \text{avec } \nabla_{\theta}G(\theta) = J^T \nabla_X P(X)$$

La commande s'exprime ici à partir de la fonction potentielle  $P$  et de la spécification de buts ou de trajectoires à atteindre (positions cartésiennes, orientations, etc.).

Ce type de méthode permet de résoudre assez efficacement le problème de cinématique inverse, sans calculer explicitement l'inverse de la matrice jacobienne. Par contre, l'optimisation par descente de gradient n'empêche pas l'atteinte de minima locaux. L'approche ne garantit pas non plus le réalisme des mouvements produits.

Exploitées telles quelles, les méthodes de cinématique inverse n'ont pas de pertinence neurophysiologique. De plus elles ne permettent pas de gérer des modifications de l'environnement ni de traiter la variabilité anatomique entre individus.

### **Approches par modèles biologiques**

D'autres approches de résolution du problème de cinématique inverse ont été proposées, qui sont conformes à des hypothèses neuro-physiologiques. Soechting [15] propose une revue de différentes études empiriques exploitables pour le contrôle du mouvement de bras humain. Il propose un algorithme d'inversion cinématique, repris par Koga *et al.* pour la planification du mouvement [16]. Une autre approche aboutit également à une transformation sensori-motrice qui permet de produire des mouvements de bras respectant un certain nombre de lois invariantes des mouvements humains [11], [12], [13]. Cette approche est fondée sur une méthode de descente de gradient associée à l'intégration dans la boucle sensori-motrice de fonctions de contrôle biologiquement plausibles.

### *Approche par algorithmes génétiques*

Une autre méthode pour résoudre un problème de cinématique inverse est une méthode basée sur un principe d'optimisation, par l'utilisation des algorithmes génétiques. En effet cette famille de méthode d'optimisation qui a été utilisée par J. Parker [17] pour résoudre un système de cinématique inverse, offre l'avantage d'être simple à mettre en œuvre, d'être efficace, et de pouvoir s'appliquer à de nombreux types problèmes différents.

Cet algorithme itératif s'appuie sur la métaphore de la sélection naturelle. Cette technique part du principe que le meilleur individu a plus de chance de survivre et de se reproduire. L'algorithme commence par la création d'un groupe d'individus. Chaque individu est une solution au problème, bonne ou mauvaise. À partir de ces individus, en les croisant, ou en faisant muter une partie de leurs gènes (changer une portion aléatoirement de ce qui caractérise la forme de l'individu), on obtient d'autres individus. Il suffit ensuite de mesurer chaque solution représentée par chaque individu et de sélectionner les meilleurs pour recommencer le cycle de mutation et de croisement qui produira une nouvelle génération encore meilleure. Ce processus est répété jusqu'à l'obtention d'une solution suffisamment bonne. Cet algorithme peut s'appliquer à toutes sortes de problèmes, il suffit de définir les opérations de mutation de croisement, et la fonction permettant de mesurer la qualité des solutions, que l'on appelle fonction d'adaptation (*fitness*). Dans notre cas, il faut trouver les valeurs des angles des articulations de la chaîne articulée pour que l'effecteur soit le plus proche possible du but. On prendra donc des individus qui n'ont qu'un seul chromosome correspondant à une configuration des différents angles de la chaîne articulée. Chaque degré de liberté, c'est à dire chaque angle de rotation, sera représenté par un gène du chromosome de l'individu, qui prendra une valeur pour un individu donné.

La fonction de fitness qui sert à évaluer un individu, mesurera la distance de l'effecteur au but final. Plus cette distance sera courte, plus l'individu sera bien noté, et plus il aura des chances de propager ses gènes dans les générations suivantes. On peut exprimer cette fonction tel que :

$$\text{Fitness (« individu »)} = - \text{distance (« but », « effecteur final »)}$$

On peut ajouter à cette formule des contraintes, comme l'évitement de certaines zones.

$$\text{Fitness (« individu »)} = - \text{distance (« but », « effecteur final »)} - \eta_{\text{collision}} * (\text{nbcl}())$$

Avec :  $\text{nbcl}()$  : nombre de collision avec la zone à éviter

$\eta_{\text{collision}}$  : facteur d'importance des collisions dans la fonction de fitness.

De cette manière nous avons tous les éléments nécessaires à l'application de l'algorithme génétique, dans le cas de la cinématique inverse.

D'autres algorithmes moins classiques existent, pour résoudre le problème de la cinématique inverse, comme les algorithmes à base de jacobienne transposée [18], ou comme ceux basées sur les réseaux de neurones.

La cinématique inverse est une méthode d'animation de haut niveau puisqu'il suffit de designer le point à atteindre pour que l'ordinateur calcule la configuration du squelette. Mais elle nécessite toujours un artiste talentueux pour animer un personnage de façon réaliste. C'est pourquoi, au lieu d'essayer de « dessiner » le mouvement, on peut se tourner vers une solution qui capture directement le mouvement sur la réalité.

### **2.2.3 Capture de mouvements**

#### 2.2.3.1 Les systèmes de capture du mouvement

Ici, au lieu de faire appel au talent artistique d'un infographiste, nous pouvons utiliser des systèmes qui vont enregistrer le mouvement d'un acteur, pour ensuite l'exploiter, et l'utiliser lors de la production d'image de synthèse. Ainsi on sera sûr, d'obtenir une animation réaliste. Divers types de méthodes sont employées pour arriver à capturer le mouvement basées sur différentes technologies de repérage d'un point physique dans un espace. Nous allons présenter les principaux types de système, et présenter leurs avantages et leurs inconvénients.

Le type de système le plus courant, utilisé depuis des années dans la création de mouvement de personnages virtuel, est celui basé sur une technologie optique (fig. 5). De nombreuses sociétés comme Elite, Motion Analysis ou Vicon proposent depuis longtemps de tels systèmes. Le principe consiste à appliquer sur une personne, des cibles (pastilles réfléchissantes par exemple) puis à repérer leurs positions dans l'image de plusieurs caméras (infrarouge), pour ensuite en déduire leurs positions dans l'espace, par triangulation. Un des inconvénients de ce système, vient des problèmes d'occultation des capteurs. En effet, pour effectuer la triangulation donnant la position 3D, il faut qu'au minimum 2 caméras puissent voir la pastille (voir souvent plus pour obtenir une bonne précision). On essaie de pallier ce problème en utilisant un très grand nombre de caméras. C'est pourquoi, jusqu'à 16 caméras peuvent être utilisées pour filmer une même scène. Mais dans certains cas, cela n'est pas suffisant, et nécessite une phase de post-traitement, ou il faut retoucher l'animation à la main. Ce problème survient principalement lorsque l'on suit plusieurs personnes en même temps, ou lors de certains mouvements comme les roulades. La calibration pose aussi quelques difficultés. Cette difficulté, qui doit être réglée avant les captures, s'accompagne des réglages et de l'installation lourde de ce type de matériel. Mais cette technologie est encore celle qui est la plus utilisée, car en contrepartie de ces quelques inconvénients, elle présente de nombreux avantages. Par exemple la précision (ou imprécision) obtenue lors d'une capture est stable dans l'espace. Un autre avantage est le fait que le champ de capture couvert, peut être suffisamment important pour répondre à la plupart des besoins. Le dernier domaine d'application qui échappait à cette technologie, était l'utilisation de la capture en temps réel, mais Vicon et Motion Analysis proposent désormais, des systèmes qui en sont capables. C'est sans doute la technologie la plus aboutie, mais elle nécessite encore de gros moyens logistiques (une grande salle, équipements coûteux, ...).

On peut aussi signaler l'existence de systèmes optiques qui reposent sur la lumière infrarouge. La base émet des rayons infrarouges qui balayent la scène. Le capteur détecte instantanément le rayon lumineux, et renvoie les informations nécessaires au calcul de sa position à la base. Ce procédé est utilisé dans les applications les plus exigeantes, comme par exemple dans le milieu médical, car il offre la précision la plus élevée de tous les systèmes optiques. Entre autre, la société Ascension propose un tel système appelé LaserBird.



*Fig. 5. Systèmes de capture du mouvement optique. A gauche les pastilles réfléchissantes sont attachées sur l'acteur, et écartées de la surface de la peau, ce qui peut permettre de réduire les occultations. A droite, elles sont collées sur une combinaison noire, ce qui gêne moins les mouvements. On peut aussi voir, sur l'image de droite, deux acteurs dont les mouvements sont capturés en même temps.*

Le traitement temps réel dans des domaines comme l'animation de personnage interactif au sein d'une émission de télévision, est par contre généralement réservé aux systèmes électromagnétiques (fig. 6). Ce type de système est composé d'un ou plusieurs émetteurs fixes, et d'un ou plusieurs récepteurs disposés sur la personne faisant les mouvements à capturer. En général les émetteurs produisent un champ magnétique, capté par les récepteurs. Les informations sur le champ magnétique captées, sont ensuite analysées, pour en déduire la position 3D des récepteurs. Mais ces données peuvent aussi nous fournir, contrairement aux systèmes optiques, l'orientation de ces capteurs. Grâce aux propriétés des champs magnétiques, il n'y a plus les problèmes d'occultation. La taille réduite de ce type de capteur et l'apparition de système sans câble entre l'acteur et le dispositif de traitement, nous permet d'en placer un grand nombre sur la personne en mouvement. Ainsi ce dispositif est très simple à utiliser, et nécessite que peu de moyens logistiques. Mais cette technique a quelques défauts, puisqu'elle est sensible aux interférences produites par les objets métalliques de l'environnement, et son champ effectif est assez limité. Ceci l'empêche d'enregistrer une capture précise de mouvements trop amples, comme par exemple, une personne qui effectue une longue course. Les deux principaux systèmes commerciaux à base de cette technologie, sont les systèmes Polhemus et Ascension.



Fig. 6 Actrice équipée d'un système électromagnétique d'ASCENSION, sans fil.

Pour des applications d'acteurs virtuels en temps réel, on peut aussi utiliser des systèmes mécaniques. Ils se présentent sous la forme d'une combinaison équipée, en premier lieu, d'un capteur permettant de repérer l'acteur dans l'espace. Assez souvent, ce capteur est un gyroscope qui est un dispositif qui mesure la vitesse et la direction de déplacement. De cette manière, en fixant un point de départ, on peut en déduire le déplacement global de l'acteur. Son mouvement lui, est mesuré par des systèmes mécaniques, qui donnent l'angle de chaque jointure de la personne. Ces systèmes sont peu coûteux, peuvent offrir une très grande zone de capture, mais le squelette mécanique gêne les mouvements, et la précision des captures, ne sont pas aussi bonnes qu'avec un système optique.

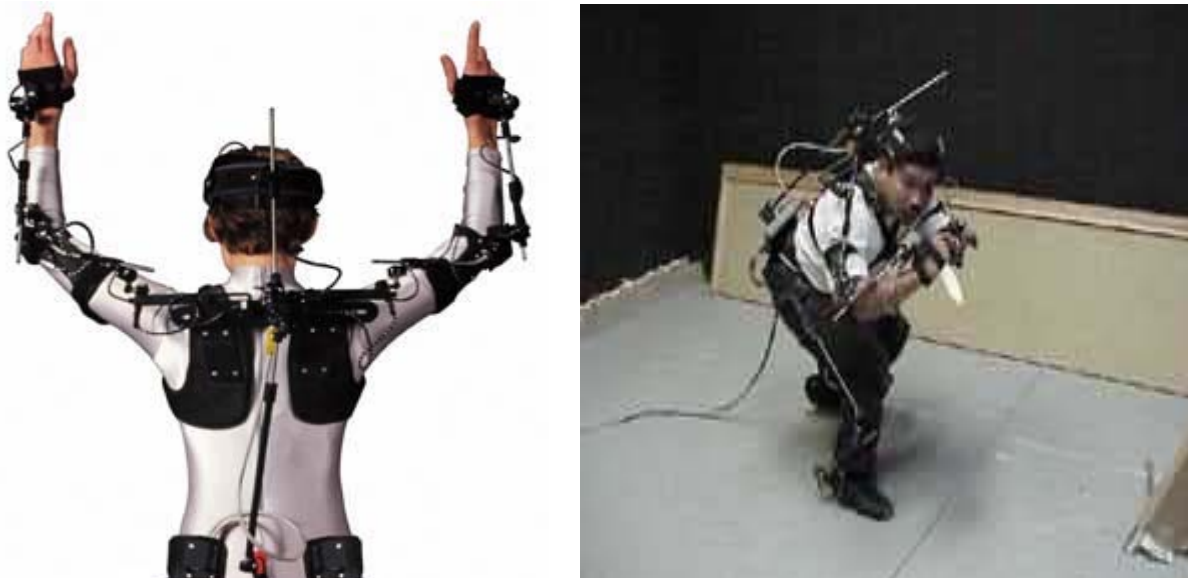


Fig. 7. Acteurs équipés d'un exo-squelette. A droite le system gypsy.

Pour clore ce panorama des techniques de capture de mouvement, il faut signaler les travaux soutenus en analyse d'image (avec une ou plus caméra), qui essaie d'identifier le mouvement de façon non invasive [24] [38]. Mais ces travaux sont encore en phase de recherche.

### 2.2.3.2 Utilisation de base de données de mouvements.

La capture de mouvements présente un intérêt indéniable ; le réalisme des animations obtenues est incomparable avec les méthodes d'animation classiques comme les méthodes cinématiques ou dynamiques. Cependant cette technique présente un défaut qui est la contrepartie directe de sa force : le mouvement capturé est figé dans sa spécificité, et si on tente de le modifier pour l'adapter à une morphologie différente, ou le déformer pour répondre à un objectif donné, le réalisme du mouvement est altéré de manière significative. Différentes méthodes (présentées dans ce document) répondent partiellement à cette lacune et tentent de se servir du mouvement capturé comme d'une base modifiable pour générer des animations particulières. Mais, à partir d'un mouvement donné, le spectre des mouvements que l'on peut obtenir via ces méthodes est relativement limité ; par exemple modifier le style d'un mouvement nécessite des connaissances sur la nature même du mouvement dont on ne dispose pas et qui manquent déjà aux méthodes de synthèse du mouvement. Ainsi, des travaux relativement récents proposent d'utiliser des collections ou base de données de mouvement représentant une large portion des possibilités de mouvements réalisables par un même humanoïde, ou de nombreuses variations sur le même mouvement (parfois les deux). Notons ici que cette variabilité peut s'obtenir dans une seule grosse séquence suffisamment longue pour contenir toutes les informations nécessaires : c'est le cas de graphes de mouvement (*motion graph*) [52], où, par une étude de similarité entre les différentes postures qui constituent le mouvement, on est capable de générer un graphe orienté dont le parcours permet de générer un nouveau mouvement. La difficulté revient ici à pouvoir choisir son chemin dans le graphe relativement à un objectif donné. Généralement ce problème fait référence aux méthodes de programmation dynamique. Lorsqu'on dispose d'une base de données comportant de nombreuses variations sur le même mouvement, plusieurs méthodes ont été développées pour générer de nouveaux mouvements :

- **interpolation basique.** L'interpolation simple entre les mouvements a montré sa force dans le cas de mouvements simples comme le pointage ou atteindre un objet distant [69],
- **décomposition dans le plan comportemental ou fréquentiel.** L'idée est ici d'exprimer l'ensemble de mouvements dans un espace comportemental, stylistique ou émotionnel. Plusieurs méthodes existent : décomposition dans l'espace de Fourier [67], dans un espace de fonction à bases radiales [41] ou en chaînes de Markov cachées [70].
- **Modèles physiques.** L'utilisation d'une base de mouvements marche aussi dans le cas de la simulation dynamique : Van de Panne [68] utilise un ensemble de mouvements simulés par modèles physiques pour générer le mouvement d'une lampe bondissante en sélectionnant relativement à des contraintes décrites par l'utilisateur et l'environnement des transitions dans cet ensemble de mouvements.
- **Modèles statistiques.** Ces méthodes permettent de réaliser une généralisation du mouvement, et de réutiliser le résultat dans un modèle de synthèse. Dans ce cas on en réutilise pas directement le mouvement enregistré. Citons entre autre les méthodes à base d'analyse en composantes principales [71] ou de minimisation d'entropie croisée [70]. La difficulté revient dans ce cas à ne pas délaissier des détails du mouvement qui pourtant contribuent grandement à leur réalisme.
- **Réutilisation du meilleur mouvement.** Dans ce cas, on suppose avoir à disposition de très larges bases de données qui comportent de nombreuses variations des mouvements. La problématique revient alors à classer cette base en

sous-classes (*clustering*) [54], et à décomposer ces mouvements en graphe [66] ou structures probabiliste comme des chaînes de Markov [54]. Certaines problématiques font référence au problème générique d'indexation dans la mise en œuvre de ces recherches.

En conclusion, l'utilisation d'une base de données de mouvements semble être une voie prometteuse permettant de pallier au manque de flexibilité des méthodes basées sur la capture de mouvements, la vraie difficulté résidant cependant toujours dans les espaces de représentation utilisés.

### 2.2.3.3 Discussion

Pour conclure sur la capture de mouvement, nous pouvons dire, que c'est la seule technique qui reproduise le réalisme des mouvements d'une personne d'une façon sûre et fiable. Mais elle a un défaut majeur pour l'édition de mouvement, qui est sa non adaptabilité aux physionomies différentes, entre le personnage virtuel, et le personnage réel capturé. En effet, la capture d'un mouvement effectuée par une personne de très grande taille, ne peut pas être directement utilisé sur un personnage virtuel plus petit. C'est pourquoi, on doit faire appel à des techniques qui vont adapter le mouvement pour faire en sorte que celui-ci soit utilisable sur n'importe quel acteur virtuel. D'autres approches font la démarche inverse, c'est à dire, au lieu d'enregistrer la réalité, on va essayer de simuler dans son ensemble, cette réalité.

### **2.2.4 Génération automatique de mouvements**

Pour cela, il faut revenir à la source du mouvement et essayer de le simuler. Pour qu'un personnage bouge, il faut qu'il exerce des forces sur ses os par l'intermédiaire de ses muscles. Il faut donc simuler l'action de ses muscles et les lier aux os. Il faut représenter toutes les interactions de volume, les forces, les articulations, les torsions ... Ceci est évidemment très complexe et on est obligé de s'appuyer sur des simplifications des modèles employés.

Pour générer les mouvements en voulant se rapprocher le plus possible de la réalité, il faut créer un système de simulation qui respecte les lois physiques simples comme par exemple les lois de la mécanique établies par Isaac Newton. Mais il faut aussi gérer les diverses interactions entre les entités, comme les collisions. Ceci est très complexe, par contre on bénéficie de certains avantages par rapport aux méthodes de cinématique directe et inverse, car certaines contraintes n'ont pas besoin d'être spécifiées, pour être respectées. Par exemple, la non pénétration du pied dans le sol est automatique, car c'est une contrainte qui n'est pas violée si les lois de la physique sont respectées et que les collisions sont détectées correctement [19]. Pour cela nous avons besoin de modéliser un système de simulation le plus correct possible. Il existe deux principaux modèles physiques simple, pour représenter les interactions des éléments et des forces entre elles.

Le premier, est le modèle le plus simple. Il est basé sur un système de particules [20] ou masse/ressort. Les objets animés sont représentés par des points matériels. Chaque particule possède une position, une vitesse, une masse, ainsi que des forces exercées sur elle. On peut ainsi calculer sa position, au cours du temps en respectant la gravité ou les force de friction par exemple. Mais pour représenter des forces plus complexes, il faut relier ces particules. Ces liaisons peuvent être de nature élastique ou rigide. Cela suppose que la masse du système est concentrée dans quelques points matériels. Ce modèle est très efficace, généralisable et nécessite peu de puissance de calcul, mais il ne respecte pas la distribution des masses dans un solide et peut être instable.

Pour cela il faut se tourner vers un modèle plus complexe mais plus juste, c'est à dire celui basé sur un système de corps rigides [21]. A partir des lois de Newton, la méthode d'intégration d'Euler permet de calculer la vitesse, et l'accélération angulaire. Cette méthode est plus complexe et plus lourde, mais l'erreur de la simulation physique est plus réduite, ce qui lui confère une plus grande stabilité.

Pour utiliser ces systèmes de simulation physique, il faut bien évidemment définir les objets, et les forces présentes dans la scène. Pour cela on doit s'appuyer sur des recherches dans les domaines de la biomécanique et de la robotique. En effet, les études menées en robotique se sont portées sur des mouvements humains comme la marche, la course, ou même des mouvement de gymnastique. Mais comme aucun robot n'est aussi complexe qu'un être humain, cela a amené à trouver une manière de simplifier ces systèmes mécaniques. Cette simplification peut être réutilisée pour recréer le mouvement d'un homme au sein de la simulation. Les données biomécaniques nous fournissent les informations sur les forces exercées et sur leur variation au cours du temps, pour différents mouvements, par l'intermédiaire par exemple de relevés de l'activité musculaire. Cela permet d'avoir une simulation qui ressemble le plus possible à la réalité.

Mais après avoir défini les forces théoriques nécessaires à la simulation d'un être humain, il faut utiliser des éléments que sont les algorithmes de contrôle (ou contrôleur), pour pouvoir les appliquer à la simulation. On distingue deux types de contrôleurs : adaptatifs et non adaptatifs [19]. Les contrôleurs non adaptatifs reproduisent les forces à exercer, d'après les données biomécaniques, ou d'après un algorithme précis. Leur défaut est que leur performance est très dépendante de la rigueur de l'algorithme utilisé. Certains problèmes sont tellement complexes qu'il est presque impossible de créer un contrôleur totalement adapté. C'est pourquoi, de nouvelles techniques sont apparues basées sur les travaux en intelligence artificielle. Cela part du principe que la configuration d'un contrôleur ne doit pas être figée et doit s'adapter pour bénéficier de meilleures performances. Leurs actions pourront même être mélangées pour fournir d'autres mouvements plus complexes [22]. Cela offre un double avantage : le contrôleur va améliorer ses performances au cours de la simulation, et il va aussi pouvoir rester performant dans des situations imprévues. De nombreuses méthodes existent pour les rendre "intelligent". On peut utiliser des algorithmes évolutifs, ou des réseaux de neurones couplés avec un algorithme génétique par exemple.

Mais même ces contrôleurs adaptatifs possèdent toujours quelques paramètres laissés accessibles à l'animateur, pour pouvoir régler le réalisme de leurs actions. En effet chaque algorithme est créé pour une action particulière, comme le balancement du bassin, ou pour garder l'équilibre. Et l'on doit ajuster ces mouvements à la main, puisque qu'il n'existe pas encore de méthode automatique, sans l'intervention humaine, pour mesurer le réalisme d'un mouvement. Un autre problème vient du fait que chaque contrôleur doit être conçu, pour chaque action différente que doit réaliser l'humain virtuel. En effet, on n'a pas encore trouvé de méthode pour générer ces contrôleurs automatiquement.

Mais cette technique de création de mouvement d'humain, possède des qualités indéniables par rapport à celle vu précédemment. Effectivement, elle permet d'obtenir une animation qui peut s'adapter à son environnement, et qui peut même être interactive, et non prédéfinie, contrairement à ce que l'on obtient en capture du mouvement, ou en animation par images clef. Malheureusement, ce procédé n'a pour l'instant pas réussi à produire de résultats visuellement satisfaisant. C'est en grande partie dû à la complexité du corps humain, et aussi par ce qu'il nécessite encore, presque toujours l'intervention d'un artiste et de son appréciation sur le réalisme du mouvement. Mais au lieu de chercher à synthétiser le mouvement, certains ont cherché à l'analyser, comme ce que l'on fait sur une image.



### 2.2.5 Analyse fréquentielle du mouvement

Au cours de l'animation, les rotations ou les translations des différents éléments du squelette, évoluent au cours du temps. Cette évolution peut être représentée sous forme de courbes (fig. 8). Dans ce cas, il suffit de considérer le mouvement comme un ensemble de courbes, et de l'analyser comme tel avec des méthodes de traitement du signal sur le plan fréquentiel, comme les techniques utilisant la transformation de Fourier rapide ou utilisant les ondelettes. Le but est de séparer et de modifier certaines caractéristiques de la courbe pour changer les actions effectuées par l'humain virtuel [23] [24]. En général les basses fréquences contiennent le mouvement global, tandis que les hautes fréquences contiennent les détails, mais aussi le bruit. Ainsi en modifiant le signal, on peut adoucir ou au contraire, augmenter la nervosité de la marche du personnage [23], ou même le rendre plus ou moins heureux, ou plus ou moins en colère [25]. Mais, l'inconvénient, c'est que pendant le filtrage du signal, certaines contraintes peuvent ne pas être respectées, comme la contrainte de non pénétration du pied dans les sols ou les angles limites. En plus de cela, la compréhension des modifications du signal et ses répercussions sur le mouvement, peut être difficilement anticipé par un artiste.

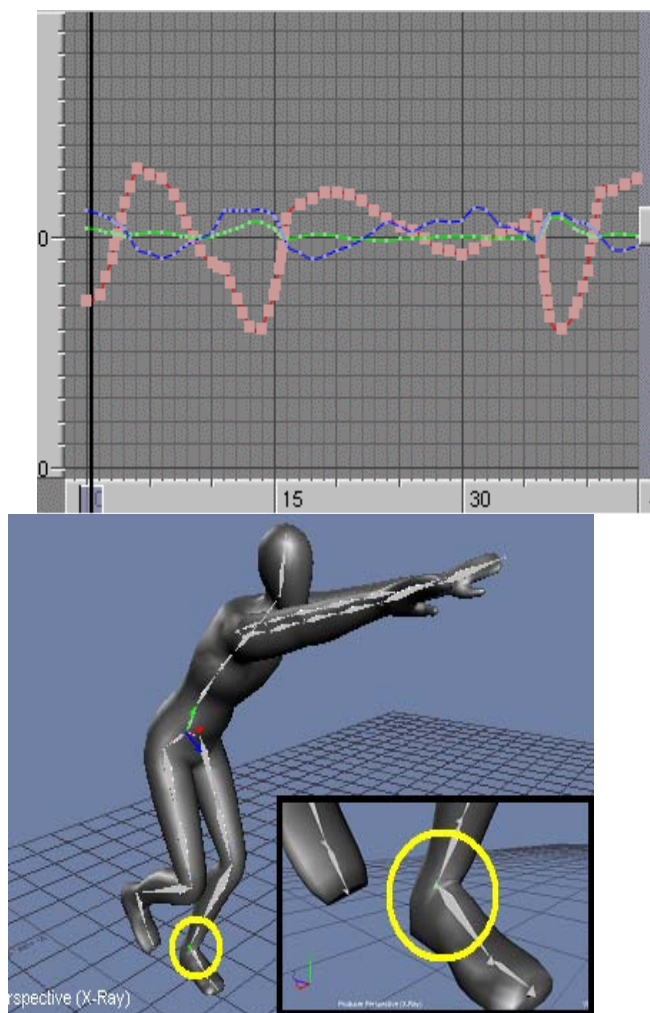


Fig 8 Ces images nous montrent l'évolution des courbes de la cheville d'un personnage qui effectue une roulade. A droite, nous obtenons trois courbes correspondant à l'évolution des rotations suivant les 3 axes (3 degrés de liberté ou DDL) de la cheville du personnage (à droite), sur un total de 81 DDL pour le personnage en entier.

C'est pourquoi l'animation de personnage par contrainte considère, elle aussi, le mouvement comme un signal.

### **2.2.6 Simulation et animation de mains virtuelles.**

La main humaine constitue une partie essentielle de l'outillage biomécanique humain. Outre leurs capacités fonctionnelles, les mains représentent aussi un moyen de communication possédant une expressivité inhérente à la complexité de leur structure. Cette complexité constitue une difficulté pour des animateurs ou dans un cadre de simulation, aussi bien du point de vue de la modélisation que du contrôle moteur. En synthèse d'images, la plupart des travaux sur la main ont eu comme cadre la préhension (grasping dans sa version anglaise) [72] ou le geste, avec comme cadre applicatif la robotique [73], l'interaction homme machine, la langue des signes et la recherche médicale (simulation pré-opératoire) [78]. Il existe aussi toute une gamme de travaux ayant porté sur la modélisation fine de mouvements particuliers (par exemple mouvements instrumentaux [77] ou bien encore des mouvements de nature sportive). Il faut remarquer que l'étude des mouvements de la main peut constituer un domaine d'étude à part entière dans le domaine de l'animation. En effet, même si de nombreuses techniques récentes proposent des méthodes génériques de contrôle de haut niveau de structure de type squelette (méthodes cinématiques, dynamiques ou basées sur le motion capture, détaillées dans le reste de ce document), l'application de ces méthodes dans leurs formes standards échouent à retranscrire la finesse des mouvements de la main [74, 77]. Citons à titre d'exemple les méthodes de cinématique inverse. Particulièrement adaptées pour contraindre des chaînes linéaires, elles ne sont que très peu pratiques pour des chaînes arborescentes comme la main, et il n'est pas possible de considérer de manière indépendante chacun des doigts (une chaîne simple par doigt) car il existe des contraintes de couplage entre les doigts [75]. De plus, dans le cadre de tâches plus complexes comme la préhension d'un objet ou un geste instrumental (accord de guitare), il est relativement peu aisée de définir une tâche particulière ; il faut choisir de manière arbitraire telle ou telle configuration finale de la main alors que l'être humain réalise un rétro contrôle permanent pour ce type de tâches.

Notons finalement que la problématique de l'animation de mains est souvent couplée avec leur modélisation (modèles déformables, éléments finis), la main présentant là encore une problématique de visualisation lui étant propre [76, 78].

## ***2.3. Animation de personnages par contraintes***

### **2.3.1 Introduction**

L'animation d'humains virtuels avec contraintes repose essentiellement sur un problème d'optimisation. De nombreux algorithmes d'optimisation existent pour résoudre des problèmes de natures très différentes. Là où des méthodes analytiques ne trouvent pas de solution, nous pouvons malgré tout en trouver une, convenable, grâce aux méthodes d'optimisation numériques. Elles sont la plupart du temps itératives, et offrent un champ d'application très vaste. Mais elles n'ont pas pour but de trouver une solution exacte, mais plutôt une solution suffisamment bonne, en un temps raisonnable. C'est pourquoi, de tels algorithmes peuvent être utilisés pour modifier une animation. En effet nous retrouvons les principales caractéristiques d'un problème d'optimisation, c'est à dire des contraintes qu'il faut respecter, et une fonction objectif (ou fonction de coût) qu'il faut minimiser, par rapport à des paramètres (ou degrés de liberté). Nous exposerons tout d'abord les principales méthodes

d'optimisation qui peuvent s'appliquer à notre problème. Puis nous parlerons de la façon dont nous allons les appliquer dans notre cas. Et nous finirons par le panorama des possibilités que cela nous offre pour l'édition d'animations de personnage.

### 2.3.2 Méthodes d'optimisation

De nombreuses méthodes d'optimisation ont été développées. Certaines pour un cas très particulier, d'autres, les méta heuristique ont pour vocation de s'appliquer à beaucoup de problèmes différents. Chaque heuristique possède des caractéristiques particulières, comme la place mémoire qu'elle occupe. Mais elles ont aussi des paramètres de contrôle, qu'il faut ajuster dans le but d'améliorer leurs efficacité lors de la résolution d'un problème particulier. Nous allons nous intéresser aux méthodes qui optimisent des paramètres dans un domaine continu, et nous laisserons de côté les problèmes combinatoires, dont le domaine est discret et fini, puisque notre problème en animation de personnage ne manipule pour l'instant, que des nombres réels. Mais avant de décrire les différentes méthodes, nous allons présenter la manière dont on définit un problème d'optimisation.

On peut poser comme fonction que l'on cherche à minimiser (fonction objectif, ou fonction de coût), une fonction  $f$  tel que :

$$F(X) : X \rightarrow \mathbb{R} \quad X : \text{valeurs de l'ensemble des paramètres}$$

On cherchera toujours à minimiser, puisqu'un problème de maximisation peut être transformé en problème de minimisation. La solution  $X$  qui minimisera  $F(X)$  sera la meilleure solution.

Mais il faut tout d'abord définir des contraintes, qui sont des fonctions qui vont restreindre l'espace de recherche. Elles vérifieront si la solution est valide, mais elles ne mesureront pas la qualité de cette solution. Il en existe de deux types :

$$C(X) = 0 : \text{contrainte d'égalité} \quad \text{et} \quad C(X) > 0 : \text{contrainte d'inégalité}$$

Dans un problème de ce type il faut donc restreindre l'espace de recherche en trouvant le domaine des solutions respectant les contraintes. Puis dans ce domaine, il faut chercher la meilleure solution ou une solution s'en rapprochant le plus. Mais pour pouvoir appliquer les heuristiques traditionnelles d'optimisation, il faut transformer ce problème contraint en problème non contraint en rajoutant des paramètres (multiplicateur de Lagrange), ou en ajoutant une fonction de pénalité [27] [28].

On peut exprimer l'équation de cette manière :

$$G(X) = F(X) + \sum_{i \in N_e} w_i * C_i(X)^2 + \sum_{i \in N_i} w_i * (\min(C_i(X), 0))^2$$

Avec :  $w_i$  : poids de la contrainte  $C_i$ .

$N_e$  : indice des contraintes d'égalité

$N_i$  : indice des contraintes d'inégalité

Ainsi nous pouvons utiliser divers algorithmes d'optimisation en plus des algorithmes spécifiques à la résolution de problèmes d'optimisation sous contraintes.

Nous pouvons séparer les heuristiques d'optimisation en deux catégories. Celles qui cherchent à trouver le minimum local et celles qui cherchent à trouver le minimum global. Les méthodes de recherche de l'optimum local sont souvent plus rapides que celles qui font une recherche globale, mais elles ne garantissent pas d'obtenir le meilleur optimum sur

l'ensemble du domaine. Ces méthodes ne s'excluent pas, et peuvent se combiner. Par exemple on peut utiliser une méthode globale pour faire une recherche sur un grand espace de recherche. Cette phase s'appelle la phase de « diversification ». Puis on peut, lors de la phase d'« intensification » faire une recherche locale, sur les zones sélectionnées précédemment, pouvant contenir la meilleure solution. Nous allons tout d'abord présenter quelques méthodes locales, suivies par une présentation des méthodes globales.

### 2.3.2.1 Méthodes locales

Les méthodes locales consistent, à partir d'une solution temporaire minimum quelconque, de trouver les voisins de cette solution.

#### **a) Monte Carlo (recherche aléatoire)**

C'est la plus simple des méthodes stochastiques (non déterministe). Elle consiste à tirer une solution au hasard et à évaluer cette solution par la fonction objectif  $f$ . Si son évaluation est meilleure que la solution courante, alors cette solution devient la meilleure solution courante. On réitère ceci jusqu'à ce que la condition d'arrêt soit vraie.

On peut améliorer cette méthode en tirant un échantillon de  $N$  voisins de la solution en suivant une loi de distribution uniforme. Si le meilleur voisin contenu dans l'échantillon est meilleur que la solution courante, alors on fait comme précédemment. D'autres heuristiques évaluent aussi les voisins de la solution courante, comme la descente de gradient par exemple.

#### **b) Descente de gradient**

Cette méthode [30] s'appuie sur la dérivée de la fonction objectif, ou au moins sur une approximation de celle-ci et éventuellement des dérivées successives comme la méthode de Newton. Elle consiste à chaque itération, à trouver un point voisin, meilleur que la solution courante. Pour trouver ce voisin on se déplace d'un certain pas, avec une certaine direction calculée à partir de la dérivée. On trouve l'optimum global, dans le cas des fonctions objectifs convexes. Sinon elle ne trouve que l'optimum local puisqu'elle ne peut s'échapper d'un optimum local. Seul certains critères imposés sur le pas de déplacement, et sur le choix de la direction à emprunter, permettent de trouver l'optimum global pour des fonctions qui ne sont pas trop irrégulières.

#### **c) Simplex (Nelder-Mead)**

C'est une méthode d'optimisation locale déterministe qui cherche à minimiser directement la fonction objectif  $f$ . Elle a beaucoup d'avantages, puisqu'elle est rapide, utilise peu de mémoire et est assez robuste (résiste au bruit), et facile à programmer. Sa robustesse permet de ne travailler qu'avec une approximation de la fonction objectif, plus rapide à calculer. L'algorithme de Nelder-Mead commence à partir d'un polytope qui est une figure géométrique à  $n+1$  sommets,  $n$  étant la dimension du problème. Ce polytope forme une base, et ces sommets sont ordonnés de telle façon que :

$$f(X_1) < f(X_2) \dots < f(X_n) \text{ avec } X_i \text{ sommet du polytope, et } f \text{ fonction objectif.}$$

Cette figure, par des opérations de réflexion, contraction, expansion, et multicontraction appelée aussi rétrécissement, suivant le cas, va se contracter, se dilater, ou se déplacer. En effet ces opérations donnent des points à tester, qui remplaceront le plus mauvais des sommets du polytope, si la solution représentée par ce point s'avère être meilleur.

Dès que l'écart entre le plus mauvais sommet et le meilleur sommet, est en dessous, d'un certain seuil assez faible, l'algorithme s'arrête. Ainsi l'on obtient une zone approchant

l'optimum local. Mais d'autres algorithmes spécialisés dans la résolution de problèmes à domaine continu non linéaire sous contrainte existent, comme la méthode SQP.

#### **d) SQP**

La méthode SQP (*Sequential Quadratic Programming*) [29] est une méthode d'optimisation que l'on utilise pour résoudre des problèmes contraints. Elle traite les problèmes avec des contraintes d'inégalité non linéaires, des contraintes d'égalité linéaires (en les transformant en inégalité), et des contraintes d'inégalités linéaires. Cet algorithme cherche à satisfaire les contraintes d'abord, et par ce biais déterminer si au moins une solution qui respecte toutes les contraintes, existe. Puis elle cherche à optimiser la fonction objectif. Nous verrons plus en détail cette technique dans la partie application, car nous l'utilisons pour résoudre notre problème.

Nous avons vu les méthodes locales pour résoudre un problème d'optimisation. Les méthodes globales que nous allons voir, vont nous fournir l'optimum global de la fonction sur l'ensemble du domaine de recherche, c'est à dire que l'on va obtenir le meilleur des optima locaux, contrairement aux méthodes locales, qui ne fournissent un tel optimum que dans le cas particulier de fonctions convexes.

#### 2.3.2.2 Méthodes globales

##### **a) Séparation et évaluation progressive "Branch and Bound"**

Cette méthode de recherche opérationnelle déterministe manipule aussi bien des entiers que des réels. Elle effectue une recherche exhaustive d'une ou plusieurs solutions d'un système d'équation/inéquation non linéaire (méthode constructive, complète et exacte) pour un problème CSP numérique à domaine continu. C'est une méthode basée sur l'analyse d'intervalle. On commence par chercher les boîtes englobantes de l'ensemble des solutions du CSP à domaine continu (de dimension égale aux dimensions du problème). Ensuite on choisit une boîte (branch), puis on essaie de la réduire (pruning) pour qu'elle se rapproche le plus possible à l'ensemble des solutions. Trois possibilités s'offrent à nous :

- Il n'y a pas de solutions dans cette boîte, on l'élimine du traitement.
- Le sous domaine encadré par cette boîte englobante, ne contient que des solutions, dans ce cas, on le garde.
- Si on ne sait pas si la boîte englobante contient que des solutions ou non, on divise cette boîte en deux, et on réitère l'algorithme sur ces nouvelles boîtes.

Mais cette méthode doit être utilisée pour des problèmes de taille raisonnable (problème NP-complet dont la complexité ne croît pas proportionnellement à la taille des données traitées) avec une fonction de coût assez simple. En effet on utilise cette fonction de coût pour déterminer si un sous domaine contient ou non les solutions. Cette fonction se réduit la plupart du temps à des opérateurs arithmétiques ou à des fonctions classiques. C'est pourquoi on ne peut pas traiter les problèmes où la fonction est inconnue. C'est à dire lorsque celle-ci se comporte comme une boîte noire : on peut obtenir une évaluation de cette fonction, pour des paramètres donnés, mais son fonctionnement n'est pas, ou difficilement exprimable mathématiquement. Mais il existe d'autres algorithmes qui traitent ce genre de cas, comme la méthode de « recherche tabou ».

##### **b) Recherche tabou**

C'est une méta heuristique présentée pour la première fois par Glover [31] basée sur une idée simple qui consiste à combiner une recherche locale avec une mémoire du parcours.

Grâce à des règles, cette idée simple et efficace offre la possibilité d'éviter les optima locaux, tout en évitant de créer des cycles.

Cet algorithme commence comme un algorithme de recherche local, par une solution quelconque de l'espace des solutions. Ensuite, il examine les voisins, et là où un algorithme de recherche local ne remplace la solution courante que si le voisin améliore cette solution, la recherche tabou prend le meilleur des voisins, même si il offre une solution inférieure à la solution courante. Dans ce cas on a une dégradation de la fonction objectif. Mais cela lui permet de ne pas tomber dans un optimum local et de pouvoir en ressortir. Le problème c'est qu'une fois que l'on a choisi le voisin le moins mauvais, on risque de tomber de nouveau sur cet optimum local, qui est évidemment meilleur que ce mauvais voisin. Donc pour éviter de boucler sur cette valeur, il faut mémoriser le chemin parcouru, et ne pas choisir de nouveau ces solutions. Ces valeurs qui ne doivent pas être sélectionnées une nouvelle fois, sont donc des valeurs *Tabou*, et sont insérées dans une liste. Ainsi une nouvelle solution n'est acceptée que si elle n'est pas dans cette liste.

L'inconvénient de cette méthode, c'est que la liste, même si elle est de taille constante et gérée de façon cyclique (la plus ancienne valeur est éliminée et on rajoute la nouvelle valeur en tête de liste), prend beaucoup de place en mémoire. Pour pallier cela, au lieu de mémoriser les valeurs, on va mémoriser les transformations interdites. Par contre cela peut interdire l'évaluation de valeurs non encore visitées. Pour lever cette interdiction on peut se baser sur certains critères (*critère d'aspiration*). Cette amélioration permet de ne pas oublier d'éventuelles solutions, ou de repartir de solutions déjà visitées mais en allant dans une autre direction. Deux autres concepts augmentent l'efficacité de la recherche. Le premier est celui d'*intensification*, qui permet d'explorer plus précisément des zones que l'on soupçonne intéressantes pour la recherche d'une solution. Le second, est le concept de diversification, qui consiste à interrompre la recherche périodiquement, et continuer à partir d'une solution tirée au hasard pour explorer le domaine dans une plus grande largeur.

D'autres méthodes d'optimisation qui se basent aussi sur un système permettant d'éviter de tomber et de rester dans le creux d'un optimum local existant, comme par exemple la méthode du recuit simulé.

### c) Recuit simulé

Le recuit simulé (1983) puise ses origines en thermodynamique par analogie avec le refroidissement lent de la matière en fusion. Il faut refroidir la matière en marquant des paliers suffisants, pour qu'elle puisse atteindre un équilibre thermodynamique. Cette technique est utilisée par exemple pour obtenir de l'acier.

Cette analogie consiste à considérer la fonction  $f$ , comme une fonction d'énergie, et  $x$  un état de la matière, dont  $f(x)$  est l'énergie. On considère un déplacement élémentaire vers un de ses voisins de l'état courant  $x$ , par la fonction  $s()$ . Pour cela on utilise la plupart du temps l'algorithme de Métropolis pour accepter la nouvelle valeur obtenue par perturbation de la solution courante.

Pour cet algorithme méta heuristique, on utilise une valeur qui va régler le critère d'acceptation de la nouvelle valeur, que l'on appelle *Température* ( $T$ ), toujours par analogie à la thermodynamique. En effet lorsque le voisin, tiré aléatoirement, obtient une énergie inférieure à la solution courante (diminution du coût), c'est à dire si :

$$\Delta f = f(s(x)) - f(x) \quad \text{et que} \quad \Delta f < 0$$

Alors, nous remplaçons la solution courante par ce voisin de qualité supérieure.

Par contre si  $\Delta f > 0$ , le voisin est moins bon que la solution courante. Mais comme pour la recherche Tabou, il peut être intéressant de le choisir pour sortir d'un optimum local. Pour cela on va le sélectionner avec une certaine probabilité, qui va dépendre de la température mais aussi de l'écart énergétique avec l'état courant. On exprime cette probabilité par la formule suivante :

$P = e^{(-\Delta f / T)}$ ; avec :	<p>P probabilité d'accepter le nouvel état. e fonction exponentielle. <math>\Delta f</math> différence énergétique entre la solution courante et la solution examinée. T température</p>
------------------------------------	--

Ainsi on peut observer que lorsque la température du système est élevée, la probabilité de choisir n'importe quel variation est proche de 1. Ceci revient à faire une recherche aléatoire dans l'ensemble du domaine d'exploration. On fait ensuite diminuer la température pour converger vers l'optimum global plus rapidement.

L'efficacité de cet algorithme dépend fortement des paramètres de contrôle choisis. On peut changer la valeur de départ de la température, modifier la fonction qui baisse cette température au cours de la recherche, ou encore changer le critère d'arrêt. L'évolution de la température est primordiale. Si elle baisse trop rapidement, la recherche peut être bloquée sur un optimum local, par contre si elle baisse trop lentement, la convergence sera très lente. L'ajustement de ces paramètres de contrôle est aussi très important pour d'autres méthodes d'optimisation, comme celles basées sur les algorithmes génétiques que nous avons vu précédemment, que nous détaillerons plus ici.

#### **d) Les algorithmes génétiques**

Les algorithmes génétiques [32] que nous avons déjà abordés, lors la résolution de la cinématique inverse, font partie des algorithmes stochastiques. Cette méthode répond assez bien au problème d'optimisation avec contraintes qu'il est nécessaire de résoudre lors de l'application de notre technique d'édition et de contrôle d'animation de personnage. Cette méta heuristique s'inspire du fonctionnement de l'évolution des espèces biologiques. En effet ce paradigme darwinien se base sur plusieurs principes :

- La sélection naturelle donne l'avantage aux espèces les plus adaptées
- Variabilité lors du passage enfant ↔ parent
- Objectif est de survivre et de se reproduire
- Adaptation au milieu

On garde le même vocabulaire pour décrire les éléments mathématiques de ce procédé que dans le domaine des études biologiques. Cet algorithme nécessite tout d'abord la création d'une population d'individus : chaque individu représente une solution, et l'on modélise cette solution par un ou plusieurs chromosomes. Les chromosomes possèdent autant de gènes qu'il y a de variable qui définissent la solution. Donc chaque gène est représenté par une valeur (pour note application ça sera la valeur d'un degré de liberté). Pour un individu l'ensemble des valeurs de ces gènes s'appel le génotype. Et ce que l'on évaluera pour juger un individu sera seulement le phénotype qui est la conséquence directe du génotype.

Cet algorithme est itératif et comporte une phase initiale, suivie de trois phases pour chaque itération (fig. 9). Lors de l'initialisation nous générons aléatoirement une population d'individus. Ensuite nous sélectionnons des géniteurs. Ces géniteurs subissent une mutation

ou un croisement (entre deux géniteurs) qui vont créer de cette manière de nouveaux individus ayant un génotype différent. Nous évaluons alors les performances de chaque individu. Dans cette nouvelle population, nous allons choisir suivant leur performance ceux qui survivent pendant la phase de remplacement, en distinguant ou non les parents et les enfants. Ensuite nous recommençons un nouveau cycle en sélectionnant les individus qui pourront se reproduire, en fonction de leurs performances. On reproduit ce cycle jusqu'à ce que la condition d'arrêt soit vraie. Cette condition peut être un temps d'exécution ou un nombre d'itérations maximum, mais peut aussi être fonction de l'importance de l'amélioration de la solution entre chaque itération (si on ne progresse plus, on arrête la recherche). Outre le fait de définir cette condition, on peut bien sûr ajuster beaucoup de paramètres de contrôle, pour adapter l'algorithme génétique au problème traité.

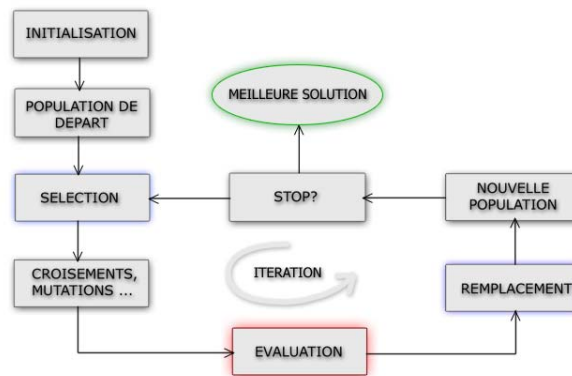


fig. 9 schéma représentant l'algorithme génétique dans son cas général.

Il faut trouver un bon compromis entre l'exploration et l'exploitation :

- L'exploration consiste à élargir la recherche de l'optimum le plus largement possible sur le domaine, pour ne pas manquer l'optimum global.
- L'exploitation est la phase qui utilise déjà les solutions correctes trouvées, pour converger vers la solution finale plus rapidement.

On s'aperçoit vite que si l'on favorise l'exploration, la convergence sera lente, par contre si on privilégie l'exploitation, on risque de ne pas trouver l'optimum global, mais juste un optimum local. Ainsi en ajustant divers paramètres on peut favoriser l'efficacité de l'algorithme au détriment de la qualité de la solution trouvée, ou inversement.

Par exemple le taux (ou nombre) de mutations influence l'exploration. Plus il est élevé, plus les individus auront des chances de ressortir des optima locaux pour continuer l'exploration. En effet une mutation qui est une reproduction asexuée, change la valeur d'un ou plusieurs gènes aléatoirement et engendre ainsi des individus qui peuvent être très différents de leurs parents. L'opérateur de mutation peut lui-même être ajusté pour produire des individus plus ou moins proches de leurs parents et ainsi explorer le domaine d'une façon plus ou moins large. L'opérateur de croisement peut lui aussi influencer le comportement de cet algorithme, mais aura tendance par contre, à augmenter l'exploitation des solutions car il mélange deux individus pour créer la génération suivante. Cependant, les effets du croisement diminuent au fur et à mesure de la convergence, puisque les individus sont de plus en plus semblables au cours de celle-ci.

Un autre paramètre qui influence cet équilibre entre le phénomène de l'exploration et de l'exploitation est la manière de définir l'étape de sélection, ou celle de remplacement. Car ces deux étapes favorisent plus ou moins les meilleurs de la population. Et comme plus on



favorise les meilleurs, moins la diversité génétique est conservée, par conséquent, l'exploration ne se fera pas en largeur mais en profondeur. Mais pour pallier ce problème on peut évidemment faire en sorte de choisir de temps en temps un individu de qualité plus faible. Pour cela, il faut utiliser des systèmes de tirages particuliers, comme par exemple un tirage utilisant la méthode de la roulette pipée, c'est à dire que plus un individu est bon plus il a des chance de se faire sélectionner. Dans ce cas les meilleurs sont favorisés, mais les mauvais ont quand même une chance.

Le dernier élément qui influence la recherche est la taille de la population. Ces paramètres sont très difficiles à ajuster, et peu de règles existent pour automatiser cette tâche. Pour l'instant on utilise la plupart du temps la méthode empirique d'essai/erreur pour optimiser l'application de cet algorithme à un problème particulier.

Il y a d'autres inconvénients. L'algorithme génétique s'apparente à plusieurs recherches de solution avec une méthode locale en parallèle, avec comme originalité l'opérateur de croisement. Malheureusement, cela nécessite une utilisation intensive et volumineuse de la mémoire de stockage des données, par rapport à une seule recherche locale, d'autant plus que la taille de la population peut être de plusieurs centaines d'individus. La sûreté de cet algorithme est elle aussi remise en cause puisqu'on n'a pas encore prouvé sa convergence pour n'importe quel type de domaine. Etant donné que seul son application à la manipulation de valeurs binaire a été prouvée mathématiquement, nous n'avons aucune garantie de la convergence de cette méthode d'optimisation pour d'autres types de variables. Cependant, expérimentalement, les qualités de convergence de cet algorithme, n'ont pas été mises en défaut.

Toutefois, la manière dont il explore le domaine offre de nombreux avantages. En effet, comme la recherche de l'optimum s'effectue par l'exploration du domaine en parallèle et en largeur, cet algorithme possède quelques propriétés intéressantes. Contrairement aux méthodes locales, on ne recherche pas un mais tous les optima locaux. Cela nous donne aussi la possibilité d'hybrider l'algorithme génétique avec d'autres méthodes de recherche locales, pour augmenter son efficacité. Effectivement, si l'on arrête l'algorithme génétique prématurément, nous obtenons les zones à privilégier pour continuer la recherche, on peut alors lancer des méthodes locales, qui sont plus rapides, sur ces mêmes zones. Grâce à ces nombreux avantages, l'algorithme génétique est extrêmement polyvalent.

Après avoir vu les différentes façons de résoudre un problème d'optimisation, par optimisation globale ou locale, nous allons parler de son application concrète à l'animation d'humain virtuel.

### 2.3.3 Animation d'humains virtuels par édition avec contraintes

#### 2.3.3.1 Positionnement

On part du constat que des mouvements réalistes sont la plupart du temps obtenus à partir d'une méthode qui utilise à la base un mouvement déjà très réaliste. L'œil humain étant très exercé à observer des mouvements, il est très difficile de le tromper avec un génération totalement automatiques : les techniques de génération d'un nouveau mouvement à partir de rien, n'ont pour l'instant donné que des résultats peu concluants. Le mouvement de départ bien que limitatif, peut subir de nombreuses modifications pour répondre aux demandes créatives. Ainsi, on peut l'adapter, le modifier, le mélanger avec d'autres, ou l'enrichir de nouvelles propriétés. C'est pourquoi, l'utilisation à la base, de captures de mouvement ou d'autres techniques équivalentes en qualité, est un élément essentiel à l'application des techniques que nous présenterons. Nous expliquerons ici, la démarche suivie pour s'orienter vers l'utilisation de contraintes pour faire évoluer ces mouvements réalistes.

Pour cela, la première idée examinée était de pouvoir mélanger toutes les méthodes vues précédemment, c'est à dire que l'on puisse utiliser conjointement, la capture du mouvement, la cinématique inverse, ou même un moteur physique. La perspective était de pouvoir garder les avantages de chaque méthode lors de l'animation d'un même personnage. Malheureusement ce système mélange des techniques hétérogènes, qui sont difficiles à ajuster une par une, mais qui le sont encore plus lorsqu'elles sont combinées entre-elles. Parfois l'on obtient même l'addition des défauts plutôt que celle des qualités de chacune des techniques mélangées.

L'idée suivante a été d'insérer cette notion de contrainte, pour répondre aux besoins d'édition et de contrôle du mouvement. Mais cette notion est appliquée seulement à chaque trame (image visualisée de l'animation) et donc à un instant donné, et non pas sur l'ensemble du mouvement. Ainsi pour chaque image, nous pouvons nous assurer du respect des contraintes par méthode de cinématique inverse, ou bien sûr par des algorithmes d'optimisation. Comme on part du constat que les modifications que l'on va apporter au mouvement ne sont pas de nature à le changer d'une façon importante, nous pouvons ne pas tenir compte des lois de la dynamique. En effet, l'erreur engendrée par le non respect de ces lois sera sans doute trop faible pour avoir un impact visuel sur l'animation, le mouvement de départ respectant déjà implicitement ces lois.

Pour cela il faut un algorithme spécifique pour chaque modification que l'on veut effectuer. Par exemple, pour animer un personnage qui marche (algorithme de *footstep*), il faut au moins :

- un algorithme qui calcule la position de la racine du squelette
- un algorithme qui calcule la rotation que la racine doit effectuer pour le mouvement de marche
- un algorithme qui calcule la position et l'orientation des pieds (suivie de l'utilisation de la cinématique inverse qui modifie la configuration des jambes en conséquence, pour que le pied puisse atteindre ces pas)
- un dernier qui établit le mouvement du haut du corps.

On s'aperçoit vite que cette approche nécessite des algorithmes dédiés pour chaque problème rencontré. C'est à dire qu'il faut écrire un programme spécifique pour faire des actions, que nous décrirons plus en détail au chapitre suivant, comme par exemple le suivi de

chemin, l'adaptation à de nouveaux squelettes, du mélange de mouvements... La vision de ce système est très locale, puisqu'il traite chaque instant, séparément. C'est pour ça qu'il faut expliciter beaucoup de traitement pour garder une cohérence entre les trames. Ainsi le système ne s'aperçoit qu'au dernier moment de la violation d'une contrainte, et les mesures instantanées qu'il prend pour rectifier cela peuvent apporter une discontinuité dans le mouvement [35]. On peut pallier cela en lissant les déplacements, mais évidemment, cette action peut entraîner la violation d'autres contraintes.

C'est pourquoi, on peut observer quelques incohérences temporelles, et des discontinuités, dans l'évolution de la vitesse ou de l'accélération. Mais une fois que les algorithmes ont été spécifiés pour chaque action, la vitesse de traitement de l'animation est assez bonne. Le fait de tout spécifier permet aussi de connaître exactement les actions produites sur le personnage. Ainsi nous obtenons exactement le mouvement préalablement désiré. Malgré ces quelques avantages, certains se sont penchés sur une solution alternative comme Witkins et Kass [34] ou Gleicher et Litwinowicz [35] pour considérer le problème d'optimisation sur l'ensemble du mouvement.

### 2.3.3.2 « Space Time optimisation »

Effectivement au lieu de considérer le problème d'optimisation juste sur un instant, on peut le traiter dans son ensemble. Ce type d'optimisation est appelée « *spacetime optimisation* » puisqu'elle traite des données spatiales au cours du temps.

Les paramètres à optimiser peuvent être de toute sorte. Tant qu'ils influent sur le mouvement, ils peuvent être pris en compte. Les plus évidents sont les degrés de liberté (*degree of freedom, DOF*) c'est à dire l'angle de chaque joint du squelette, ainsi que la position du *root* (racine), puisqu'ils définissent explicitement la configuration spatiale du personnage. Mais cela peut s'appliquer sur toute sorte de valeur qui évolue au cours du temps, comme le degré d'activation des muscles, la position des pieds ou des mains au cours du temps...

Pour obtenir un mouvement lisse et fluide, nous exprimons l'évolution de ces paramètres à l'aide de courbes, comme des B-Spline ou des courbes de Bézier. Ce lissage offre l'avantage de pouvoir garder les caractéristiques de la courbe d'évolution, tout en utilisant moins de points pour la définir.

L'objectif que l'on va utiliser sur ces paramètres, est de ressembler le plus possible à l'animation d'origine provenant la plupart du temps, d'une capture de mouvement. Puisque l'on suppose que l'on ne va pas créer complètement une nouvelle action exécutée par l'humain virtuel, nous essayons ainsi, de garder le plus possible les caractéristiques réalistes du mouvement initial. Cela se traduit par une fonction objectif qui mesure un degré de ressemblance entre l'animation de départ, provenant de la capture de mouvement, et l'animation finale, modifiée par l'utilisateur.

Les contraintes vont être en rapport direct avec la modification que l'on veut effectuer sur l'animation. Ce sont elles qui vont permettre d'éditer le mouvement. Par exemple, on peut faire en sorte de limiter le glissement des pieds sur le sol, en définissant une contrainte qui va être violée lorsque le pied qui ont touché le sol, bougent horizontalement et pas seulement verticalement. C'est très utile pour faire une transition entre deux mouvements, ou pour changer les proportions du personnage sur lequel est joué l'animation.

On s'aperçoit très vite que l'utilisation de cette approche peut répondre à la plupart des besoins que l'on a pour l'édition de mouvement d'humains virtuels. Effectivement, on

exprime assez facilement les contraintes nécessaires à l'accomplissement des modifications que l'on veut porter sur l'animation.

Si on utilise les courbes d'évolution des paramètres pour modifier le mouvement directement, il se peut que certaines caractéristiques soient effacées. Effectivement certaines contraintes peuvent lisser les courbes et ainsi perdre les petits mouvements de l'animation, essentiels au réalisme du résultat. Pour répondre à ce problème, nous pouvons utiliser une technique qui s'appelle le *motion displacement* [35] [23]. Elle consiste non pas modifier directement la courbe, mais plutôt, à calculer une autre courbe qui représentera le déplacement à effectuer découlant de la contrainte, et venant s'ajouter à la courbe originale. De cette manière nous conservons la forme ainsi que les hautes fréquences du mouvement, permettant de garder le naturel de l'animation.

La méthode de *Space Time optimization* semble cumuler tous les avantages. Premièrement le réalisme du mouvement est assuré grâce à la fonction objectif qui essaie de conserver cette qualité de l'animation de départ, conjointement au respect des contraintes. Mais ce réalisme peut être d'autant plus appuyé en créant spécifiquement des contraintes ajoutant une garantie supplémentaire au respect du réalisme. En effet on peut très bien incorporer le respect des lois de la physique à leur niveau. Ainsi la violation de ces lois correspondra à la violation d'une contrainte [42].

Ensuite le fait de traiter le problème dans sa temporalité permet d'avoir une certaine conscience du mouvement dans son ensemble. De cette façon le système a la possibilité d'anticiper bien avant les actions à exécuter, de manière à trouver le meilleur cheminement précédent son accomplissement. Entre autres, si il on veut faire saisir une balle au personnage virtuel [37], il peut modifier le balancement de son bras et même sa trajectoire, une seconde avant, pour saisir cette balle correctement. Ainsi la plupart des problèmes peuvent se résoudre d'une façon générique et implicite.

On peut ajouter, que, grâce à cette méthode on peut garantir la fluidité du mouvement. Effectivement, il suffit d'utiliser des types de courbes ayant la propriété d'être  $C^2$  continues comme notamment, des B-Spline pour s'assurer de la continuité du mouvement.

Mais on peut en théorie faire, avec ce procédé, des traitements de très haut niveau sur le mouvement. Dans l'absolu, on pourrait même ajouter des caractéristiques émotionnelles, comme par exemple, produire un mouvement de marche triste, enjouée, ou fatiguée. Seule l'imagination, et une autre limite importante à cette technique, restreint son champ d'applications. Cette limite naît dans la manière même de résoudre le problème, c'est à dire en passant par un algorithme d'optimisation. Effectivement l'animation d'un personnage contient énormément de paramètres, par exemple pour un squelette extrêmement simplifié, il faut 20 degrés de liberté (DDL), en prenant 5 points de contrôle par seconde pour définir la courbe d'évolution de ces DDL, pour 5 secondes d'animation il faut pas moins de 500 paramètres à optimiser. A cela s'ajoute le nombre de contraintes pouvant être de plusieurs centaines, pour certains types d'édition que l'on veut effectuer sur l'animation. De cette manière, on peut se retrouver avec des temps de calculs rédhibitoires pour la résolution de ce problème d'optimisation. A cause de cela, on ne peut pas éditer un mouvement en temps interactif et s'apercevoir ou non des erreurs de réglages rapidement.

Un autre inconvénient vient de la difficulté à définir ces contraintes. Même si on peut exprimer toutes sortes de problèmes en les utilisant, cela peut être très complexe à mettre en place. On ne perçoit pas facilement la liaison entre la contrainte exprimée mathématiquement et l'action la plupart du temps géométrique que l'on veut faire exécuter à l'humain virtuel par cette contrainte. On est d'ailleurs souvent obligé de les rectifier ou d'en ajouter de nouvelles pour obtenir exactement le mouvement désiré. Ce manque de perception de ce que va

provoquer une contrainte, peut donc rendre la tâche d'édition assez ardue. Il faut aussi remarquer, que certaines d'entres elles peuvent s'exclure mutuellement. Dans ce cas le système ne peut pas trouver de solutions valables, et il faut alors chercher d'où provient le blocage, ce qui peut être assez complexe.

Mais heureusement le champ d'application de cette méthode est très vaste. Effectivement, les méthodes par contraintes peuvent parfaitement répondre aux principaux types d'édition que l'on peut faire sur un mouvement provenant d'une capture.

### 2.3.4 Les applications possibles

Le but des procédés vus précédemment est de proposer une solution pour répondre aux besoins artistiques ou scientifiques en animation de personnage. Nous allons présenter les différents cas les plus courants d'utilisation des captures de mouvement. Nous commencerons par exposer les techniques visant à combiner ou à adapter l'animation d'un personnage que sont l'adaptation à de nouveaux personnages et le mélange de mouvements. Ensuite nous passerons aux problématiques d'adaptation à l'environnement et d'édition de mouvement, qui changent d'une manière plus importante le mouvement.

#### a) Adaptation à de nouveaux personnages (« Retargeting »)

Cette technique [39] consiste à adapter le mouvement d'une personne à une autre personne de proportion différente. En effet lorsque l'on effectue une capture on obtient les courbes d'évolution des différents angles des joints du squelette. On peut utiliser ces données pour animer un autre squelette. Mais on s'aperçoit que comme les os du squelette sont de taille différente, les différences angulaires enregistrées entre deux images, ne provoquent plus le même déplacement des éléments dans l'espace. Ce phénomène est d'autant plus gênant qu'il peut subvenir dès la première étape d'utilisation de la capture du mouvement, lorsque la personne réelle qui effectue cette capture a un gabarit très différent du personnage imaginaire.

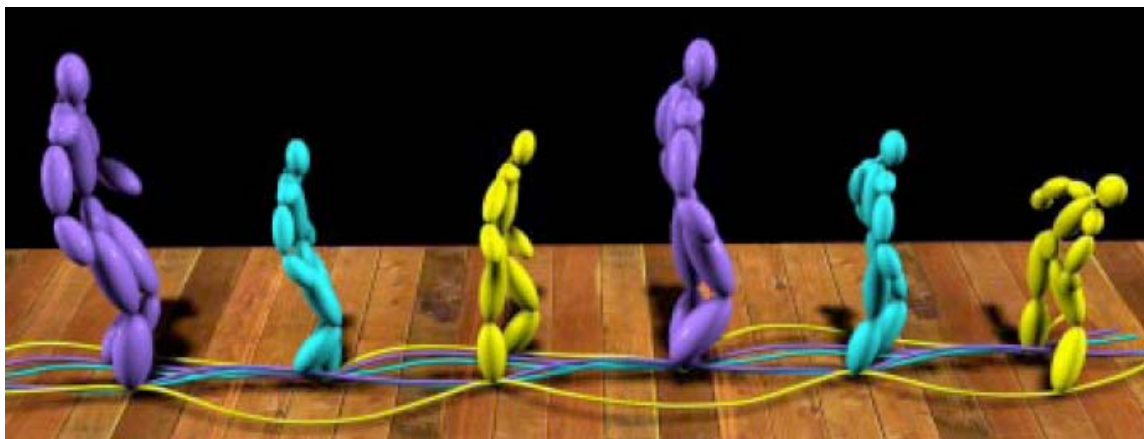


fig. 10 Le personnage mauve est celui d'origine. Le jaune est 40% plus petit et est animé avec le mouvement de marche du personnage original sans rectification. On peut s'apercevoir que les mouvements des pieds sont très différents (lignes jaunes et lignes mauves). Le personnage bleu lui aussi 40% plus petit, a subi une adaptation du mouvement, et suit donc très bien le mouvement de marche d'origine. (Image provenant de la référence [39])

C'est pourquoi on peut observer de nombreuses imperfections dans l'animation. Par exemple, il se peut que les pieds glissent sur le sol lorsque l'on fournit à un personnage l'animation d'un autre personnage plus petit (fig. 10). On peut aussi observer ses bras ou ses jambes qui rentrent dans le corps de l'individu. En plus de cela, un problème de décalage spatial peut subvenir, entre deux personnages (fig. 11), ou entre le personnage et l'environnement. Par exemple s'il saisit un objet, s'il interagit avec un autre personnage ou

s'il frappe une cible, cette animation, appliquée à un squelette plus petit ou plus grand, produira une animation qui ne saisit pas l'objet, qui ne touche pas la personne au même endroit ou qui n'atteint pas la cible [37].



fig. 11 Dans cette image on observe l'adaptation du mouvement de danse de deux personnages. A droite le mouvement original. Au centre, le personnage féminin, cette fois réduit, adapte la position de sa main. A gauche les deux personnages ont la position de leurs mains adaptées à ce nouveau cas. (Image provenant du texte de référence [39])

Mais les techniques d'adaptation de mouvements à de nouveaux personnages permettent d'adapter les mouvements à des squelettes de proportions différentes. De cette manière, on peut véritablement créer des banques de mouvements, totalement indépendantes des personnages. Pour mettre à profit au maximum les données d'une telle banque de mouvement, il existe d'autres techniques comme le mélange de mouvements.

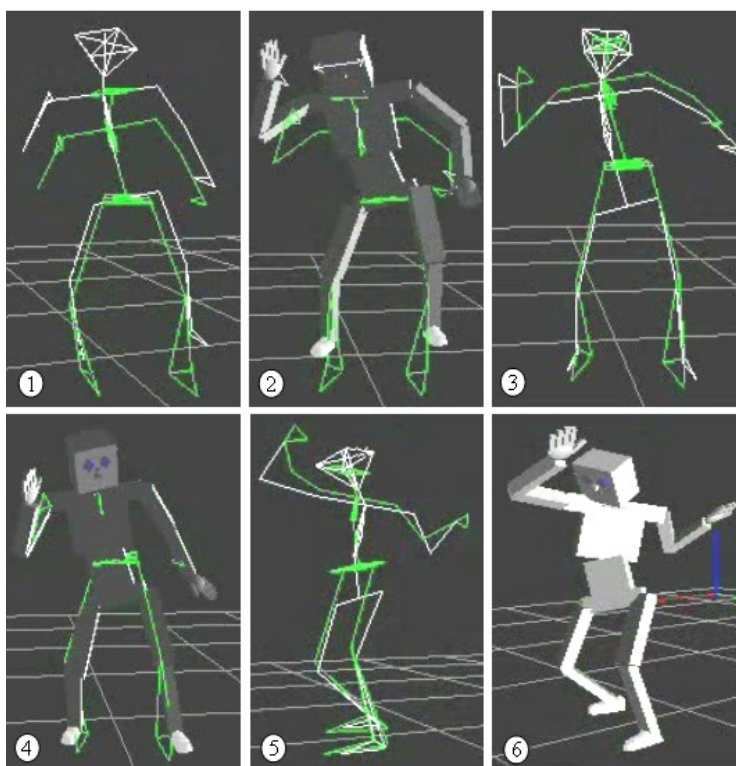


fig 12: adaptation d'un mouvement de dans capturé sur un personnage plus petit (en vert) que le personnage synthétique. On voit ici les différentes phases : 1) répétition des angles sans adaptation avec le squelette en blanc, 2) visualisation sur le personnage synthétique qui ne touche pas le sol, 3) et 4) déplacement des pieds sur le sol, 5) et 6) déplacement des mains du squelette destination pour qu'elles correspondent aux positions capturées [46].

Une autre approche consiste à utiliser un squelette simplifié normalisé [46]. Il suffit alors de contraindre les points du squelette qui doivent respecter l'environnement pour automatiquement adapter la posture aux nouvelles données anatomiques. Les autres points du squelette sont retrouvés par cinématique inverse analytique en travaillant sur des sous-chaînes isolées du squelette.

Quand un mouvement est obtenu sur un squelette simple, il peut en effet être adapté à un plus complexe, comportant un plus grand nombre d'articulations [63]. Dans cette approche, les articulations que l'on retrouve dans les deux squelettes sont directement remplacées et les autres sont calculées par cinématique inverse.

En prenant en compte des paramètres morphologiques, et en utilisant ces méthodes Pronost [66] a adapté des mouvements acquis par capture à un squelette d'hominidé fossile.

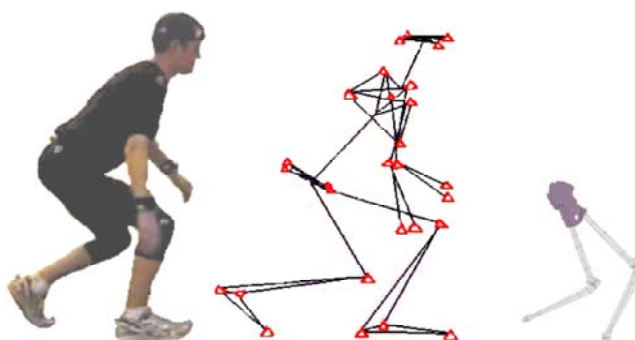


fig.13 De l'homme à l'hominidé fossile : adaptation morphologique de mouvement.

### b) Mélange de mouvements (« Motion blending »)

Si on a un mouvement de marche et un mouvement de saut, n'ayant pas été capturé ensemble mais séparément, on peut vouloir les enchaîner (fig.12). Pour cela il faut faire une transition entre la fin du mouvement de marche et le début du mouvement de saut. On utilise les techniques de mélange de mouvements [40] [41] pour réaliser cette transition.

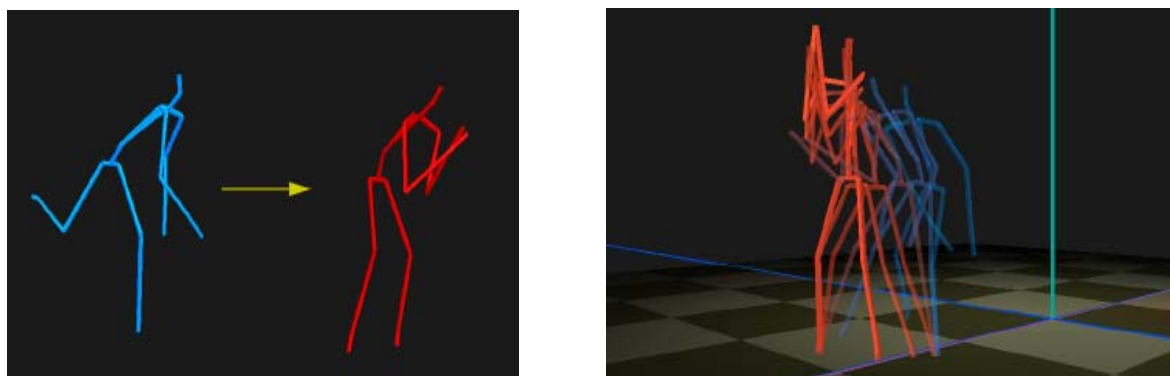


fig.14 A gauche, les deux positions à mélanger. A droite, l'animation de transition entre ces deux positions (Image provenant du texte de référence [41]).

Naturellement, si on fait une simple pondération progressive entre les deux mouvements, nous pouvons obtenir quelques défauts dans l'animation, comme celle qu'on voit dans la figure 16. En effet, ici on ne prend pas en compte les contraintes des pieds sur le sol qu'il faut donc ajouter.

Une autre difficulté vient du fait qu'il faut « caler » les deux animations pour que la fin de la première corresponde le mieux possible avec le début de la seconde. Dans le cas contraire, on risque d'avoir une transition saccadée ou peu naturelle. On peut retrouver ce problème lorsque l'on veut rendre cyclique une animation comme dans la figure 15 (on enchaîne le mouvement avec lui-même). Ces difficultés peuvent encore une fois être surmontées grâce à l'utilisation de contraintes.

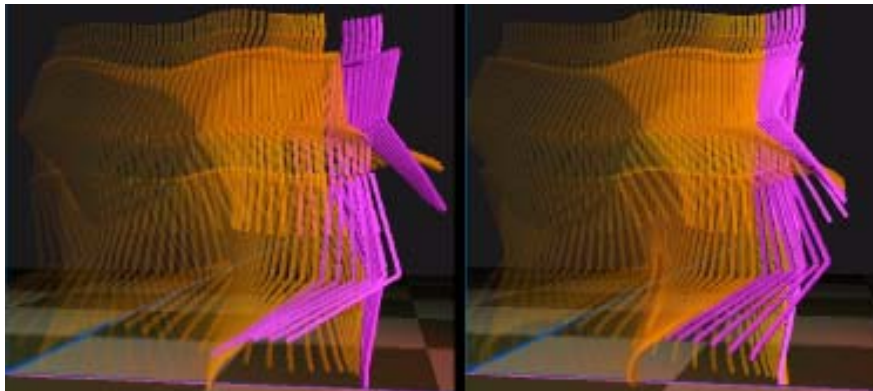


fig.15 A droite mouvement cyclique non ajusté. A gauche le mouvement de marche cyclique corrigé (Image provenant du texte de référence [41]).

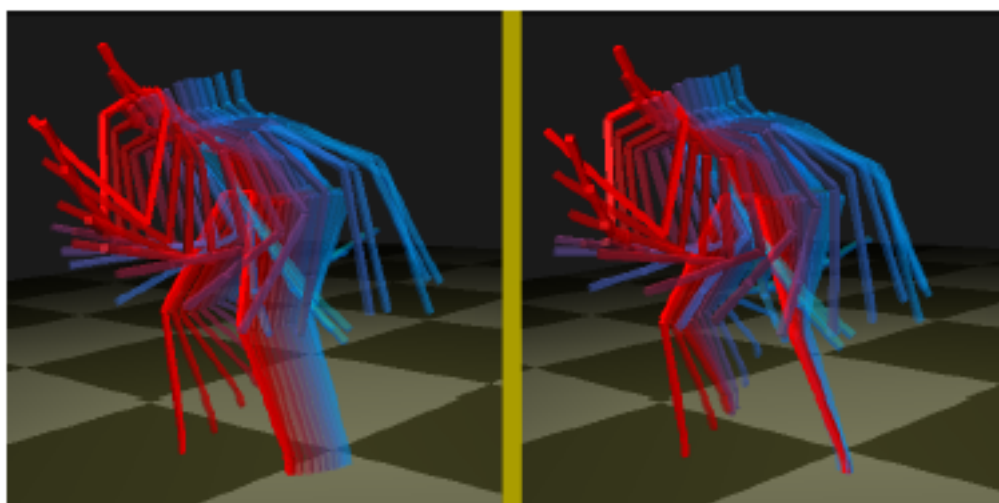


fig. 16 A droite le pied glisse pendant la transition entre les deux positions du squelette. A gauche une contrainte empêche le pied de glisser durant la transition. (Image provenant de la référence [41])

L'un des grands problèmes du mélange de mouvements est d'assurer une correspondance temporelle entre plusieurs mouvements à mélanger. Cette correspondance assure que le mélange de deux mouvements est possible et détermine l'instant le plus approprié pour effectuer le mélange ou la transition [51][52][53][60]. Pour cela, une métrique basée sur le mouvement des segments corporels a été introduite pour sélectionner le meilleur instant de transition [42][46]. Les *Motion graphs* [47][52][54] ont été introduits pour représenter une transition entre deux mouvements compatibles. Deux postures de deux différents mouvements peuvent être connectées dans ce graphe si elles sont compatibles. Ceci implique de calculer à l'avance le graphe de toutes les transitions possibles, à moins de coupler le calcul de distance présenté ci-dessus. Cette technique a été illustrée sur la locomotion humaine sur terrain très complexe impliquant des transitions entre marches différentes, courses, sauts et pas chassés à partir du calcul des empreintes planifiées sur le sol [50][55] et de rapides changements de direction [57]. Cependant, dans ces approches, les



transitions possibles sont déjà identifiées pour résoudre un ensemble de contraintes elles-aussi connues à l'avance.

Pour palier au manque de données dynamiques sur le mouvement obtenu, d'autres approches se basent sur la dynamique (via le contrôle du point où la somme des moments s'annule : *Zero Moment Point*). Il est ainsi possible de tenir compte de l'équilibre pour assurer un minimum de crédibilité aux mouvements obtenus [48][63]. Cependant, ces techniques nécessitent de connaître à l'avance les contraintes à respecter et le mouvement dans sa globalité.

Toutes ces précédentes approches raisonnent sur une sous-séquence entière du mouvement, ce qui est difficile en animation interactive. En effet, l'animation interactive implique l'arrivée de nouveaux mouvements à exécuter à tout moment, sans connaissance préalable des contraintes à respecter. Une alternative est de définir des mouvements élémentaires qui peuvent être mélangés en modifiant des poids en temps réel [49]. Cette technique est proche de celle utilisée dans les jeux vidéo en y ajoutant la possibilité de contrôler une partie du corps uniquement. Les actions sont empilées sur une pile et le mouvement résultat est une somme pondérée de toutes les séquences placées dans la pile. En plus de sa position dans la pile qui donne une première priorité à une action, chacune d'entre elles est associée à une valeur numérique de priorité, pour chaque degré de liberté. Ainsi, un mouvement de marche a une forte priorité sur les jambes et une plus faible sur les bras. Cette technique peut être associée à une représentation normalisée du squelette qui permet de coupler les problèmes de mélange de mouvements et d'adaptation au squelette et à l'environnement [44] (fig. 17). L'avantage de ces deux dernières techniques et qu'elles permettent une gestion en temps réel des mélanges de mouvement, sans devoir pré calculer des séquences entières.

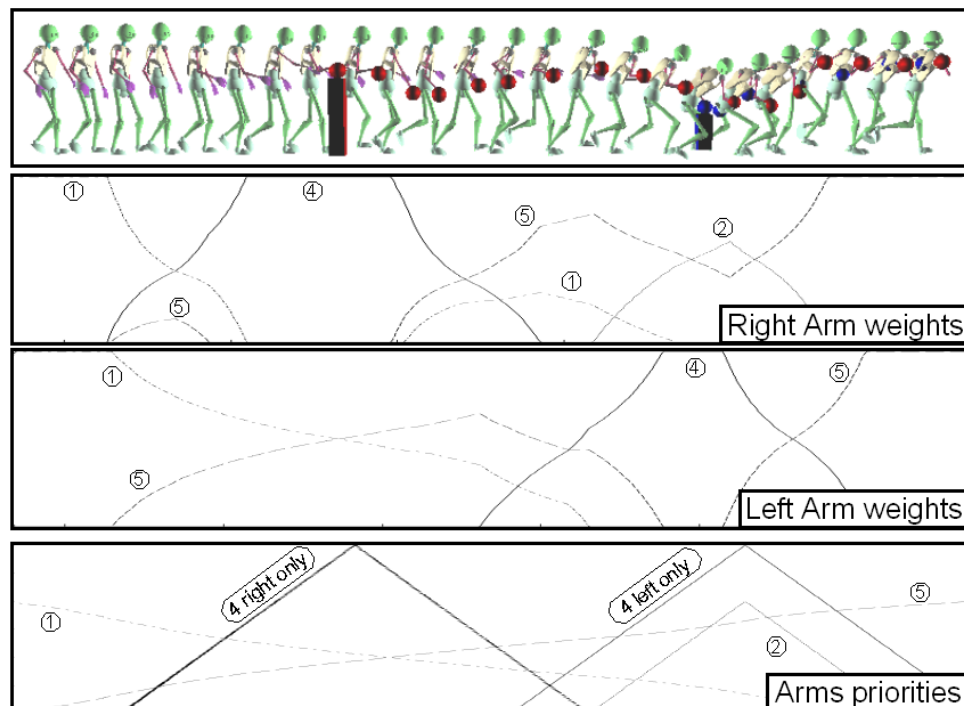


Fig. 17: mélange de 5 mouvements (1: marche normale, 2: s'accroupir, 3: marche « discrète », 4: préhension et 5: course). L'utilisateur ou le module comportemental lance les actions en faisant varier l'état de chaque action et observe en temps réel le mouvement se modifier. Les priorités et les poids sont présentés dans le bas de la figure[44].

Cependant, comme les techniques basées sur de l'optimisation ou de la recherche dans un graphe, il est nécessaire de définir au préalable un ensemble de priorités pour chacun des mouvements et à chaque instant. On ne peut donc pas utiliser directement des données de capture sans une phase manuelle. De plus, la prise en compte de la dynamique est pour l'instant limitée au maintien de l'équilibre et il semble que des améliorations soient à apporter dans cette voie.

Après avoir vu les techniques qui réutilisent les captures de mouvement, nous allons voir celles qui les modifient plus en profondeur.

### **c) Adaptation du mouvement à l'environnement**

Effectivement, les conditions de capture ne sont pas toujours rigoureusement identiques à l'environnement virtuel. Dans ce cas, il faut adapter le mouvement capturé sans pour autant changer sa nature. Notamment, si on veut animer une personne qui monte une pente dans le monde virtuel, il se peut que l'on ne dispose pas du matériel pour construire une pente strictement identique lors de la capture de mouvements. C'est pour cela qu'il faudra mettre en place un algorithme qui va adapter le mouvement à la pente ou à une pente d'inclinaison légèrement différente, en post-traitement.

On doit aussi faire en sorte d'adapter le mouvement lorsque par exemple, il y a des obstacles sur le chemin d'un personnage qui marche. Dans ce cas, le personnage devra contourner ces objets ou marcher dessus. Il faudra donc adapter la trajectoire de marche et changer la position des pieds, pour anticiper ces situations. Ce cas peut être fréquent lors d'animation de foule où l'on ne peut pas faire une capture de mouvement pour chaque personnage sur un sol irrégulier.

Un dernier cas, où l'on peut avoir besoin d'adapter l'animation, c'est lorsque le personnage interagit avec l'environnement. Par exemple si le personnage frappe une balle et si on souhaite faire changer la position de cette balle, il faut changer le déplacement du pied pour qu'il frappe cette nouvelle position. On retrouve ce cas lorsque l'on a deux personnes qui se battent ensemble, et si les personnages virtuels ne sont pas exactement de la même taille.

Pour le cas particulier des systèmes magnétiques qui fournissent directement des angles, Bodenheimer et al. [64] proposent de vérifier que le mouvement rejoué sur un personnage différent est sans collision ni auto-collision. Dans cette technique, le système vérifie que les contraintes géométriques (comme le contact des pieds avec le sol) sont assurées.

Quelque soit le système utilisé, des techniques plus générales ont été proposées à partir de la cinématique inverse pour résoudre des problèmes de type *spacetime constraints* [39]. Le mouvement peut ainsi être adapté à un nouveau squelette tout en respectant les contraintes d'un nouvel environnement. Cependant, comme nous l'avons vu précédemment, ces techniques optimisent un jeu de contraintes définies sur la durée totale du mouvement et posent donc des problèmes pour l'animation interactive.

Avec l'augmentation de la puissance de calculs, la cinématique inverse peut être utilisée sur le corps entier d'un humanoïde pour résoudre ces contraintes en temps réel, au fur et à mesure qu'elles se présentent [59]. A l'inverse des techniques précédentes, en raisonnant uniquement sur l'instant présent, ces techniques doivent engendrer des discontinuités dans certains cas de figure faisant intervenir des contacts. Pour limiter ce problème, les trajectoires peuvent être représentées dans le domaine fréquentiel [61]. Une première approximation du mouvement corrigé est obtenue avec les harmoniques basses puis un raffinement peut être apportées avec les harmoniques plus élevées. Cette approche multirésolution permet de limiter les problèmes de discontinuité en assurant la continuité sur les harmoniques basses.

Une autre approche consiste à utiliser un squelette normalisé pour représenter les successions de postures comprises dans un mouvement [46]. Cette méthode offre l'avantage de résoudre en une passe les problèmes de l'adaptation à l'environnement et à un nouveau squelette. Dans cette technique, on place les points contraints du squelette aux bonnes positions dans l'environnement. Le squelette est ensuite reconstruit géométriquement à partir de ces points et de données anthropométriques (fig. 18). Une fonction de lissage permet d'éviter les discontinuités dans les mouvements.

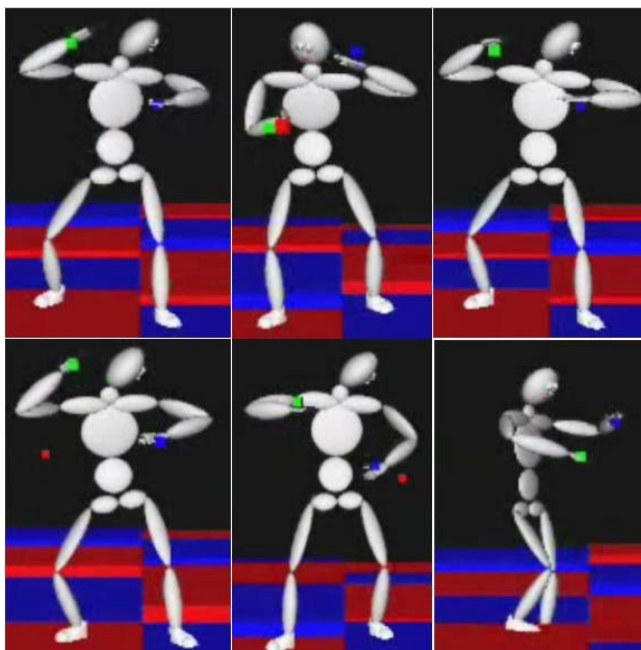


Fig. 18 : adaptation d'un mouvement de danse à l'environnement ; le sol bouge et les cubes indiquent la position où les mains doivent se poser[46].

Mais dans certains cas, ce n'est pas l'environnement qui impose les changements, mais ce sont plus les besoins artistiques.

#### **d) Edition du mouvement**

En effet, la mise en scène de personnage requiert parfois la modification du mouvement capturé original. Pour cela il faut créer des outils offrant une grande richesse de possibilités. La gravité peut être augmentée, pour faire en sorte que le personnage paraisse plus fatigué, et ainsi donner une certaine émotion. De plus on peut interagir avec lui, en le poussant, mais en le forçant à garder l'équilibre malgré tout.

On peut changer les mouvements du personnage plus fondamentalement. Par exemple, on peut le faire changer de trajectoire pour suivre une autre personne ou pour emprunter un chemin [36] sans que les pieds glissent grâce aux contraintes, comme on le voit sur la figure 19. Mais il peut aussi attraper des objets au cours de sa marche, ce qui est une action très différente du mouvement de départ.

Toutes les applications que nous venons de voir peuvent être développées grâce aux méthodes d'optimisation temporelles et spatiales (*Space Time Optimization*). Cela offre déjà de nombreuses façons d'enrichir la palette de possibilité des mouvements originaux. Nous allons exposer l'application que nous avons développée. Celle-ci a pour but de mettre en place une plate-forme d'édition et de contrôle d'humains virtuels, utilisant les méthodes d'optimisation temporelles et spatiales avec contraintes. Il semble que les différents problèmes exposés ci-dessus soient difficiles à découpler. Ainsi, une approche de mélange de mouvements peut produire un mouvement qu'il est très difficile d'adapter à l'environnement

et au personnage. Inversement, que devient un mouvement parfaitement adapté à l'environnement lorsqu'il est mélangé à d'autres. A l'avenir, il semble donc pertinent d'envisager ces problèmes de front pour proposer une technique clé en main allant de la capture du mouvement à son utilisation dans un environnement différent, avec un personnage différent et nécessitant l'utilisation de plusieurs gestes pour effectuer des tâches complexes.

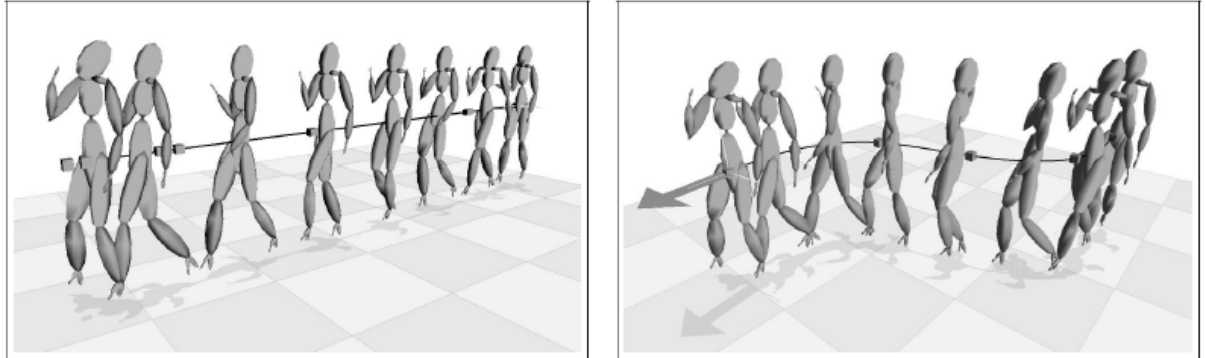


fig. 19 : à droite la marche originale. A gauche la marche originale qui a été modifiée pour suivre le chemin voulu (Image provenant de la référence [36])

Une autre piste de recherche concerne l'utilisation de la dynamique dans ce type de technique. Soit la dynamique peut être utilisée pour contrôler après traitement le résultat, soit il est nécessaire de revoir complètement les approches pour y insérer des contraintes dynamiques. Il semble que ce point soit le moins exploré actuellement.

## Bibliographie :

- [1] Edwin E. Catmull. "A system for computer generated movies". In *Proc. ACM Annual Conf.*, pages 422–431. August 1972.
- [2] N. Magnenat-Thalmann, R. Laperrire, and D. Thalmann. "Joint-dependent local deformations for hand animation and object grasping". In *Proceedings on Graphics Interface '88*, pages 26–33, 1988.
- [3] Burtnyk, N. and Wein, M., "Computer Generated Key Frame Animation". *Journal Of the Society of Motion Picture and Television Engineers* , 1971. 8(3): pp. 149-153.
- [4] Boulic R., Magnenat-Thalmann N. et Thalmann D. Combined direct and inverse kinematic control for articulated figure motion editing. *Computer Graphics Forum*, vol. 2, n°4, 1990.
- [5] Girard M. et Maciejewski A.A. Computational modeling for the computer animation of legged figures. *ACM Computer Graphics*, 19(3), pp. 263-270, 1985.
- [6] Boulic R. et Thalmann D. A global human walking model with real-time kinematic personification. *Visual Computer*, vol. 6, n°6, pp. 344-358, 1992.
- [7] Boulic R., Mas R. et Thalmann D. A robust approach for the center of mass position control with inverse kinetics. *Journal of Computers and Graphics*, vol. 20, n°5, 1996.
- [8] Boulic R., Mas R. et Thalmann D. Complex character positioning based on a compatible flow model of multiple support. *IEEE Transactions on Visualization and Computer Graphics*. pp. 245-261, 1997.

- [9] Tolani D., Goswami A., Badler N. Real-Time Inverse Kinematics Techniques for Anthropomorphic Limbs. *Graphical Models*, vol. 62, pp. 353-388, 2000.
- [10] Gibet S. et Marteau P.F. NonLinear feedback Model of sensori-motor system. *International Conference on Automation, Robotics and Computer Vision*, Singapore, 1992.
- [11] Gibet S., Marteau P.F. A Self-Organized Model for the Control, Planning and Learning of Nonlinear Multi-Dimensional Systems Using a Sensory Feedback, *Journal of Applied Intelligence*, vol.4, pp. 337-349, 1994.
- [12] Gibet S., Lebourque T. Automatic Motion Control, *Eurographics Workshop on Animation and Simulation*, Oslo, 1994.
- [13] Gibet S. Modèles d'analyse synthèse de mouvements, *Habilitation à Diriger des Recherches*, Université de Bretagne Sud, Vannes, 05 juillet 2002.
- [14] Zhao J. et Badler N.: Inverse Kinematics Positioning Using Nonlinear Programming for Highly Articulated Figures. *Transactions of Computer Graphics*, 13(4), pp. 313-336, 1994.
- [15] Soechting J.K. Elements of coordinated arm movements in three-dimensional space. *Perspectives on the coordination of movement*. Ed. S.A. Wallace, Elsevier Science Publishers, Amsterdam, pp. 47-83, 1989.
- [16] Koga, Y., Kondo, K., Kuffner, J., Latombe, J.C. Planning motions with intentions. In *ACM Computer Graphics, Annual conference series*, pp. 395-408, 1994.
- [17] D. Goldberg J. Parker, A. Khoogar. " *Inverse kinematics of redundant robots using genetic algorithms*". *Proceedings of IEEE International Conference on Robotics and Automation*, 1.3:271-276, 1989.
- [18] Bill Baxter "*fast numerical methods for Inverse kinematics*" 2000  
<http://www.cs.unc.edu/~baxter/courses/290/html/img0.htm>
- [19] Nikovski, D. (1994). "*Dynamic simulation methods for animation of legged locomotion*". CS 585 term paper, fin 1994, SIU.
- [20] Witkin Andrew (1991) "*Particle systems dynamics*" SIGGRAPH 91 Tutorial on particle- based system. Section C. ACM Press
- [21] Baraff David (1991) "*Rigid Body Concept dynamics*" SIGGRAPH 91 Tutorial on particle- based system. Section H. ACM Press
- [22] Hodgins, J. K., Wooten, W. L., Brogan, D. C., O'Brien, J. F., 1995. Animating Human Athletics. *Proceedings of Siggraph '95*. In *Computer Graphics*, pp 71-78
- [23] Armin Bruderlin and LanceWilliams. Motion signal processing. *Computer Graphics*, 29(Annual Conference Series) :97-104, 1995.
- [24] Christoph Bregler Jitendra Malik (1997) "Video Motion Capture", in *Image-Based Modeling, Rendering, and Lighting*, SIGGRAPH 99 Course #39
- [25] Kenji Amaya, Armin Bruderlin, and Tom Calvert. (May 1996) "*Emotion from motion*". In *Graphics Interface '96*, pages 222-229
- [26] Paul T. Boggs, Jon W. Tolle (1999) "*Sequential Quadratic Programming for Large-Scale Nonlinear Optimization*"
- [27] Jehee Lee (2000) PhD Thesis "*A Hierarchical Approach to Motion Analysis and Synthesis for Articulated Figures*", KAIST 2000.

- [28] J. C. Platt and A. H. Barr. (August 1988) “*Constraint methods for flexible models*”. *ComputerGraphics (Proceedings of SIGGRAPH 88)*, pages 279–288,
- [29] P.E. Gill, M.A. Saunders, and W. Murray. (San Diego, 1996.) “*SNOPT: An SQP algorithm for large-scale constrained optimization*”. Technical Report NA 96-2, University of California.
- [30] Nicolas Barnier (septembre 1997) “*Optimisation par hybridation d’un algorithme génétique avec la programmation par contrainte*”, rapport de DEA IFP (UPS/INPT)
- [31] Glover F. (1986), “*Future paths for Integer Programming and Links to Artificial Intelligence*”, *Computers and Operations Research*, Volume 13, Issue 5 (May 1986) p533-549
- [32] Marc Schoenauer (Février 2003) “*Projet Fractales*” – INRIA Rocquencourt – France <http://www-rocq.inria.fr/fractales/>
- [33] Michael Gleicher (Mars 22, 2000) “*The Intuitions of Signal Processing (for Motion Editing)*” Graphics Courses Taught at the University of Wisconsin, Madison, appendix of the book *Motion Capture and Motion Editing: Bridging Principle and Practice*, by Jung, Fischer, Gleicher, and Thingvold published A. K. Peters publishers.
- [34] Andrew Witkin, Michael Kass (1988) “*Spacetime Constraints*”. *Proceedings of SIGGRAPH’88*, pages 159-168.
- [35] Michael Gleicher and Peter Litwinowicz. Constraint-based motion adaptation. *The Journal of Visualization and Computer Animation*, 9(2) :65–94, – 1998.
- [36] Michael Gleicher. Motion path editing. In *Symposium on Interactive 3D Graphics*, pages 195–202, 2001.
- [37] Rama Bindiganavale and Norman I. Badler. Motion abstraction and mapping with spatial constraints. *Lecture Notes in Computer Science*, 1537 :70–??, 1998.
- [38] I. Kakadiaris and D. Metaxas. “*Model-Based Estimation of 3D Human Motion with Occlusion Based on Active Multi-Viewpoint Selection*” Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 81-87, San Francisco, CA, June 1996.
- [39] M. Gleicher. (juillet 1998) “*Retargeting motion to new characters*”. *Proceedings of SIGGRAPH 98*, pages 33–42,
- [40] Ahmed, A., F. Mokhtarian, and A. Hilton, “*Parametric Motion Blending through Wavelet Analysis*,” Proc. EuroGraphics Conference, pp. 347-353, Manchester, UK, 2001.
- [41] Charles Rose, Brian Guenter, Bobby Bodenheimer, and Michael F. Cohen. Efficient generation of motion transitions using spacetime constraints. *Computer Graphics*, 30(Annual Conference Series) :147–154, 1996.
- [42] Z. Popovic and A. Witkin. Physically based motion transformation. *Proceedings of ACM SIGGRAPH*, pages 11–20, 1999.
- [43] Thomas Jacobson (21 janvier, 2003) “*Advanced Character Physics*” Lien internet vers l’article: [http://www.gamasutra.com/resource\\_guide/20030121/jacobson\\_04.shtml](http://www.gamasutra.com/resource_guide/20030121/jacobson_04.shtml)
- [44] S. Ménardais, F. Multon, R. Kulpa, B. Arnaldi (2004) Motion blending for real-time animation while accounting for the environment. IEEE CGI, Crète, juin 2004.
- [45] S. Ménardais, F. Multon, R. Kulpa, Bruno Arnaldi (2003) Méthodes d’adaptation et de mélange de mouvements acquis pour l’animation d’humanoïdes, Groupe de Travail Animation/simulation, Grenoble, 2003.

- [46] S. Ménardais (2003) Fusion et Adaptation temps réel de mouvements acquis pour l'animation d'humanoïdes synthétiques, doctorat de l'Université de Rennes 1, janvier 2003.
- [47] Arikan, O., and Forsyth, D. Interactive motion generation from examples. *ACM Transactions on Graphics* 21,3, 2002, 483-490.
- [48] Ashraf, G, and Wong, K.C. Constrained Framespace Interpolation. In *Proceedings of Computer Animation 2001*, IEEE Computer Society Press, 61-72, 2001
- [49] Boulic, R. , Becheiraz, P., Emering, L., and Thalmann , D. Integration of Motion Control Techniques for Virtual Humans and Avatars Realtime Animation. In *Proceedings of ACM International Symposium VRST*, ACM Press / ACM SIGGRAPH, 111-118, 1997
- [50] Choi, M.G., Lee , J. , and Shin , S.Y. Planning Biped Locomotion using Motion Capture Data and Probabilistic Roadmaps. *ACM Transactions on Graphics*, 22, 2, 182-203, 2001
- [51] Gleicher, M., Shin, H. J., Kovar, L., and Jepsen, A. Snap Together Motion: Assembling Run-Time Animation. In *Proceedings of Symposium on Interactive 3D Graphics*, 2003
- [52] Kovar, L., Gleicher, M., and Pighin, F. Motion Graphs *ACM Transactions on Graphics*, 21, 3, 473-482, 2002
- [53] Kovar, L., and Gleicher, M. Flexible Automatic Motion Blending with Registration Curves. In *Proceedings of ACM Symposium on Computer Animation*, ACM Press / ACM SIGGRAPH, 214 – 224, 2003
- [54] Lee, J., Chai, J., Reitsma, PSA., Hodgins, JK, and Pollard, NS. Interactive Control of Avatars Animated with Human Motion Data. In *Proceedings of SIGGRAPH 2002*, ACM Press / ACM SIGGRAPH, J.F. Hughes, Ed., *Computer Graphics Proceedings, Annual Conference Series*, ACM, 491-500, 2002
- [55] Sun, H.C., and Metaxas, D.N. Automating gait generation. In *Proceedings of SIGGRAPH 2001*, ACM Press / ACM SIGGRAPH, E. Fiume, Ed., *Computer Graphics Proceedings, Annual Conference Series*, ACM, 261-269, 2001.
- [56] Tak,S., Song, O.Y., and Ko, O.Y. Spacetime sweeping: an interactive dynamic constraints solver. In *Proceedings of Computer animation 2002*, IEEE Computer Society Press, 261-270, 2002.
- [57] Tsumura, T., Yoshizuka, T., Nojirino, T., and Noma, T. T4 : A motion capture based goal directed realtime responsive locomotion engine. In *Proceedings of Computer Animation 2001*, IEEE Computer Society Press, 52-60, 2001
- [58] Wang, J. and Bodenheimer, B. An evaluation of a Cost Metric for Selecting Transitions between MotionSegments. In *Proceedings of ACM SIGGRAPH Symposium on Computer Animation*, 232-238, 2003
- [59] K. J. Choi & H.S. Ko (2000). Online motion retargetting. *The Journal Of Visualisation And Computer Animation* 2000, vol. 11, no. 5, 223-235, december 2000.
- [60] S. Guo, J. Roberg (1996) A High-Level Mechanism for Human Locomotion Based on Parametric frame space Interpolation. In: *Proceedings of Eurographics Workshop on Computer Animation and Simulation*, 95-107, september 1996.
- [61] J. Lee, SY. Shin (1999) A Hierarchical Approach to Interactive Motion Editing for Human-like Figures. In: *Proceedings of ACM SIGGRAPH*, 39-48, August 1999.

- [62] J.S. Monzani, P. Baerlocher, R. Boulic & D. Thalmann (2000). Using an Intermediate Skeleton and Inverse Kinematics for Motion Retargetting. Eurographics 2000 vol 19, no 3, 2000
- [63] S. Tak, O.Y. Song & H.S. Ko (2002). Spacetime sweeping: an interactive dynamic constraints solver. In: Proceedings of Computer animation 2002, 261-270, June 2002
- [64] Bodenheimer, B., Rose, C., Rosenthal, S., and Pella, J. The process of Motion Capture: Dealing with the data. In Proceedings of Eurographics Workshop on Computer Animation and Simulation, 3-18, 1997
- [65] N. Pronost, G. Dumont et G. Berillon. Interpolation morphologique et posturale pour la simulation de mouvement d'un humanoïde virtuel. 16èmes Journées de l'AFIG, pp 121-128, 2003
- [66] L. Kovar et M. Gleicher. Automated Extraction and Parameterization of Motions in Large Data Sets. Proc. of ACM/Siggraph 2004, 32-40
- [67] M. Unuma, K. Anjyo et R. Takeuchi, R. Fourier Principles for Emotion-based Human Figure Animation, Proc. of ACM/Siggraph 95, 91-96, 1995
- [68] A. Lamouret et M. Van De Panne. Motion Synthesis by example. Eurographics Int. Workshop on Animation and Simulation, EGCAS'1996, 1996
- [69] D. Wiley et JK. Hahn. Interpolation synthesis of Articulated Figure Motion. IEEE Computer Graphics and Applications, 17(6), Novembre, 39-45, 1997
- [70] M. Brand et A. Hertzmann. Style Machines. Proc. of ACM Siggraph 2000, 183-192, 2000
- [71] R. Bowden. Learning Statistical Model of Human Motion. IEEE Workshop on Human Modelling, Analysis and Synthesis, CVPR 2000
- [72] H. Rijkma et M. Girad. Computer Animation of Knowledge-Based Human Grasping. Proc. of ACM Computer Graphics (Siggraph 91), 339-347, 1991
- [73] Y. Koga, K. Kondo, J. Kuffner et JC. Latombe. Planning Motion with Intentions. Proc. of ACM Computer Graphics (Siggraph 94), 395-408, 1994
- [74] L. Moccozet. Hand Modelling and Animation for Virtual Humans. PhD Thesis, U. of Geneva, 1996
- [75] C. Häger-Ross et MH. Schieber. Quantifying the Independence of Hand Finger Movements: Comparison of Digits, Hands and Movement Frequencies. The Journal of Neuroscience, 20(22), 2000.
- [76] I. Albrecht, J. Haber et HP. Seidel. Construction and Animation of Anatomically Based Human Hand Models, In Proc. of ACM/Eurographics SCA 2003, Juillet, San Diego, 2003
- [77] G. ElKhoura et K. Singh. Handrix: Animating the Human Hand, In Proc. of ACM/Eurographics SCA 2003, Juillet, San Diego, 2003
- [78] T. Kurihara et M. Miyata. Modeling Deformable Human Hands From Medical Images, In Proc. of ACM/Eurographics SCA 2004, Aout, Grenoble, 2004



## Deuxième Partie : Dynamique

### 1. Introduction

Les modèles dynamiques permettent d'apporter un réalisme dans les animations par l'utilisation de lois physiques pour la génération du mouvement. Ils permettent d'engendrer des trajectoires pour un nombre quelconques d'acteurs ou de situations grâce à la généralité des équations du mouvement. Par contre le contrôle du mouvement n'est plus directement géré par l'animateur mais par les lois physiques et est donc beaucoup moins intuitif que par des méthodes à base de cinématique. C'est pourquoi de nombreux moteurs d'animation mécanique possèdent en plus des couches logicielles de contrôle du mouvement pour apporter une plus grande souplesse à l'animateur.

En général, ces logiciels ont pour support un système d'équations différentielles du second ordre sous la forme :

$$A(q)\ddot{q} + B(q, \dot{q}) = Q \quad (E 1)$$

Où  $q$ ,  $\dot{q}$  et  $\ddot{q}$  représentent les paramètres de l'animation et leurs dérivées première et seconde,  $A$  la matrice des masses généralisées,  $Q$  le vecteur des efforts imposés au système et  $B(q, \dot{q})$  les autres termes dynamiques (forces d'inertie).

Un tel système est décrit par les objets de la scène affectés de leurs données mécaniques (masse, centre de gravité, matrice d'inertie), les forces appliquées aux objets et éventuellement les contraintes permettant de lier les objets entre eux ou de limiter l'évolution des paramètres. Il est de plus nécessaire de donner l'état initial du système (positions et vitesses, valeurs et dérivées premières des paramètres).

Il existe cependant deux inconvénients majeurs à ce type d'approche :

- la modélisation de l'animation (forces et couples agissant sur les paramètres, contraintes,...) est complexe et peut être difficile à appréhender ;
- les temps de calcul sont très importants.

La dernière classe d'animation consiste à « donner vie » aux acteurs en leur insufflant des actions et réactions vis-à-vis de leur évolution et de leur environnement. Cette approche, appelée animation comportementale, permet d'envisager l'animation d'êtres vivants de façon plus naturelle. En effet, comme le montre Zeltzer [ZELTZER82], il est possible de décomposer l'animation d'un acteur :

- en lui imposant un mouvement général (but à atteindre) ;
- puis en exprimant cette tâche à l'aide d'actions « macroscopiques » définies sur des intervalles de temps donnés (marcher, courir, ... ) ;
- enfin, en décomposant ces « macro actions » en actions élémentaires évaluables par des modèles d'animations classiques (décrits ci-dessus).

De façon générale, la plupart des méthodes se décomposent en quatre points distincts : la base mécanique utilisée pour la création des équations du mouvement ; la formulation des équations ; la gestion des contraintes ; la résolution du système d'équation.

Les différents travaux présentés tendent à optimiser ou améliorer une ou plusieurs de ces étapes. D'autres encore sont spécifiques à des mouvements particuliers comme la marche humaine.

## **2. Base mécanique utilisée pour la création des équations du mouvement**

Deux méthodes principales apparaissent dans la majorité des travaux, ce sont les deux approches classiques de la dynamique. Ces deux approches sont formellement équivalentes. Il s'agit de l'approche basée sur les relations fondamentales de la dynamique, communément appelées équation de « Principe Fondamental de la Dynamique » et de l'approche basée sur une étude énergétique et les équations de Lagranges. :

### ***2.1 les relations fondamentales de la dynamique***

Le système d'équations obtenu est constitué de deux entités. La première ligne exprime le Principe Fondamental de la Dynamique pour la partie correspondant à la translation des objets, la seconde ligne exprime ce même principe pour la partie correspondant à la rotation des solides. Il s'agit donc d'un ensemble de 6 (3+3) équation par solide si le paramétrage utilisé n'est pas réduit en intégrant les équations de liaison.

Ces équation s'écrivent comme suit :

$$\begin{aligned} m \cdot a_M &= F_M \\ I_M \dot{\Omega}_S + \Omega_S \cdot I_M \cdot \Omega_S &= C_M \end{aligned} \tag{E 2}$$

Dans ce système, on note  $m$  la masse du solide  $S$ ,  $a_M$  l'accélération du point  $M$  du solide  $S$ ,  $F_M$  la résultante des forces extérieures appliquées au point  $M$ .  $I_M$  est la matrice d'inertie de  $S$  exprimée en  $M$ ,  $\Omega_S$  est le vecteur vitesse rotation instantanée du solide  $S$  et  $C_M$  la résultante en  $M$  des couples appliqués à  $S$ .

Dans le cas de chaîne cinématiques ouvertes, cette approche permet de créer et résoudre les équations efficacement. Cette résolution est souvent effectuée en mettant en œuvre un algorithme de « Newton-Euler » récursif. Par contre, les boucles cinématiques et les contraintes ne sont pas prises en compte naturellement. Il est donc nécessaire de développer des modules spécifiques pour les intégrer au moteur d'animation [SCHRODER90]. Wilhelms [WILHELMS85] et Armstrong-Green [ARMSTRONG85] génèrent les équations du mouvement à l'aide de ces équations de Newton et d'Euler, cette approche est classique en robotique.

L'algorithme de Armstrong-Green travaille sur un arbre d'objets avec 3 DdL<sup>1</sup> en rotation pour chaque lien entre un père et un fils ; seule la racine de l'arbre possède 6 DdL (3 en rotation et 3 en translation). Chaque lien contient la matrice de changement de repère permettant de passer du repère local au repère racine, notée  $R_{fg}$ , et une autre exprimant le passage du repère local au repère père, notée  $R_{fp}$ . Cette approche est adaptée pour des structure arborescentes.

Ils linéarisent alors les équations d'Euler sous la forme :

$$\dot{\alpha}_p = f(a_p) \tag{E 3}$$

---

<sup>1</sup> Degré de Liberté

$$f_p = g(a_p) \quad (E 4)$$

où  $\dot{\omega}_p$  représente l'accélération angulaire de l'objet  $p$ ,  $a_p$  l'accélération linéaire de  $p$  et  $f_p$  la force de réaction de l'objet  $p$  par rapport à son père.

L'accélération linéaire d'un fils par rapport à l'accélération linéaire et angulaire de son père est alors exprimée par une fonction de la forme :

$$R_{fp} a_f = \omega_p \wedge \omega_p \wedge c_f + a_p - c_f \wedge \dot{\omega} \quad (E 5)$$

où  $\omega_p$  est la vitesse angulaire de  $p$  et  $c_f$  les coordonnées du centre du repère de  $f$  (centre de rotation).

L'algorithme fonctionne alors en deux passes :

- Parcours récursif des objets en partant des fils (extrémités) et en remontant vers la racine. Ce parcours est nécessaire au calcul de paramètres utilisés dans les fonctions  $f$  et  $g$ . Ces paramètres sont calculés récursivement de fils en père et dépendent des forces et couples extérieurs, des éléments inertiels propres aux objets et des couples de réactions entre fils et père ;

- Parcours récursif des objets de la racine vers les fils. Pour la racine, il n'y a pas de force de réaction vers le père donc, d'après (E 4),  $g(a_{p_0}) = 0$  ce qui implique que  $a_{p_0}$  peut être évalué. De même,  $\omega_{p_0}$  peut être calculée à partir de (E 3). Il est ensuite possible de calculer les accélérations linéaires de chaque objet avec (E 5).

Les vitesses, linéaires et angulaires, et les positions sont ensuite calculées à partir de la méthode de résolution d'Euler :  $h(t + \Delta t) = h(t) + \Delta t \cdot \dot{h}(t + \Delta t)$

L'avantage de cette méthode est la faible complexité des calculs en  $O(n)$  où  $n$  est le nombre d'objets. Les inconvénients sont liés à la difficulté de contrôler le mouvement à partir de la description des forces et couples intervenant sur les objets (difficulté liée à l'utilisation de la dynamique) et à la gestion des contraintes (difficulté liée à la méthode récursive).

Hyeongseok [HYEONGSEOK94] utilise ce formalisme pour simuler la marche humaine. Il développe les équations, spécifiquement pour le mouvement des jambes. Elles sont représentées par un système à 97 degrés de liberté (DdL). Il intègre également un module de gestion des boucles qui permet de prendre en compte la phase de mouvement où un pied touche le sol.

En se basant sur ce formalisme et en introduisant des contraintes, [FANG03] propose une méthode d'animation d'humain virtuel. Cette méthode offre un outil efficace permettant de générer des mouvements physiquement cohérents à partir d'états clés définis par l'animateur. Pour ce faire, les équations du mouvement de Newton-Euler sont réécrites afin de permettre de calculer la dérivée de la matrice des couples en temps linéaire. Puis les contraintes particulières sont définies pour chaque type d'action souhaité, comme pour le sol, si l'on ne souhaite pas voir le personnage passer à travers.

## ***2.2 travaux virtuels et le formalisme Lagrangien ;***

Ce formalisme est basé sur une conservation énergétique du système au cours du mouvement et se calcule dans l'espace des paramètres. Les équations de Lagrange (E 6)

créent une relation entre les apports énergétiques des efforts extérieurs  $Q_i$  au cours du mouvement et l'énergie cinétique  $K$  du système.

$$\frac{d}{dt} \frac{\partial K}{\partial \dot{q}_i} - \frac{\partial K}{\partial q_i} = Q_i \quad i \in [1, n] \quad (E 6)$$

Avec  $q_i$  représentant le  $i^{\text{ème}}$  paramètre de positionnement du système.

L'intérêt principal de ce formalisme est de pouvoir intégrer naturellement toutes les interactions entre objets. En effet, les contraintes sont gérées avec la méthode des multiplicateurs de Lagrange et les boucles cinématiques ne sont pas traitées comme des cas particuliers.

Bruderlin [BRUDERLIN84] qui obtient un système du type (E 1) qu'il utilise pour calculer les forces et couples nécessaires à l'obtention d'un mouvement donné. Il applique ce principe pour le calcul des efforts nécessaires à la marche humaine.

Faloutsos [FALOUTSOS95] mixe une méthode à base d'images clés et d'animation dynamique pour l'animation d'objet de type « cartoon ». Le positionnement d'images clés lui permet un meilleur contrôle de l'animation ainsi qu'une facilité de génération d'effets non réalistes de type dessin animé. La dynamique intervient alors pour les étapes intermédiaires aux images clés.

Isaacs-Cohen [ISAACS88] utilisent une formulation basée sur les travaux virtuels de D'Alembert pour créer leur système d'équations du mouvement et une résolution basée sur une méthode proposée par Wittenburg [WITT77]. Leur système admet six DdL possibles pour chaque liaison et s'exprime par la formule :

$A\ddot{q} = B$  avec  $A$  la matrice  $N \times N$  des masses généralisées,  $B$  le vecteur des efforts imposés au système et  $\ddot{q}$  le vecteur des accélérations des DdL réels de rotation ou de translation exprimé en coordonnées généralisées  $q_i$  avec  $i \in [1, N]$ .

Deux options de résolution différentes peuvent être envisagées :

- Soit on connaît  $A$  et  $B$  et l'on cherche à calculer les  $\ddot{q}$  (approche dynamique);
- Soit on cherche à calculer la force à appliquer lorsque l'on connaît tout ou une partie des  $\ddot{q}$  (approche dynamique inverse).

Il est possible d'intégrer des contraintes au système en les posant sous la forme :

$$f_c(q, t) = 0 \quad \text{où } f_c \text{ est une fonction dérivable par rapport à } t \text{ et aux } q_i \quad i \in [1, N]$$

Ces contraintes sont introduites dans le système grâce aux multiplicateurs de Lagrange, une inconnue supplémentaire  $\lambda_c$  est alors associée à chacune d'elles (ce terme est lié à l'intensité de l'effort de maintien de cette contrainte). Le problème est donc ramené à la résolution du système :

$$A' \ddot{q}' = B'$$

où  $A'$  est une matrice de taille  $(M+N) \times (M+N)$  définie par :

$$A' = \begin{bmatrix} A & J^T \\ J & 0 \end{bmatrix} \quad \text{avec } J = \left( \frac{\partial f_c}{\partial q_i} \right)_{c \in [1, M], i \in [1, N]}$$

$$B' = \begin{bmatrix} B \\ -\phi \end{bmatrix} \text{ avec } \phi \text{ terme lié aux réalisations des contraintes et } \ddot{q}' = \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix}.$$

Cette méthode permet de modéliser des scènes complexes et des boucles fermées (grâce aux contraintes). Elle est cependant coûteuse en temps de calcul.

Remion [REMION00] a choisi de développer un moteur basé sur les travaux virtuels avec les équations de Lagrange permettant d'animer directement les coordonnées généralisées, expression naturelle des paramètres de la modélisation. Les articulations peuvent alors avoir un nombre de DdL quelconque, ceux-ci pouvant être simples (rotation, translation), ou complexes comme le « vissage ». L'aspect généraliste du moteur contrebalance les temps de calcul plus importants obtenus par rapport à une méthode basée sur le formalisme Newton/Euler. À noter que certains travaux [HOLLERBACH80, VILLARD95] basés sur les équations de Lagrange obtiennent une complexité identique avec un formalisme de Newton/Euler en  $O(n)$ .

Mendoza et al. [MENDOZA03] présentent quelques problèmes et solutions se basant sur le formalisme Lagrangien. Ils proposent d'une part, une nouvelle méthode paramétrable pour résoudre le calcul des forces dynamiques et de collisions pour des objets rigides et mobiles, et d'autre part, une méthode permettant de découper de manière interactive des objets 3D dans un environnement virtuel. Cette méthode repose sur la résolution d'un problème linéaire complémentaire (LCP) avec un temps de calcul paramétrable.

### ***2.3 Formulation des équations du mouvement***

La majorité des travaux exprime les équations du mouvement sous forme numérique, naturelle en informatique. Cette forme implique des approximations numériques méthodologiques lors du calcul des membres des équations du mouvement qui font intervenir des dérivées.

Il est aussi possible de les exprimer sous forme symbolique [ARNALDI88][ARNALDI89][DUMONT90][ARNALDI94][COZOT96]. Ce mode de représentation est la traduction exacte de la représentation mathématique. Il nécessite des outils de gestion de graphe lourds lors de la création des équations. De plus, les équations proposées ci-dessus n'ont que très rarement une solution analytique (formelle), elle nécessitent un passage au numérique pour la résolution du système. L'intérêt majeur est alors d'avoir une expression formellement exacte des équations du mouvement. Cependant les outils nécessaires à la manipulation symbolique engendrent des développements conséquents et des taux de stockage importants.

Gillard [GILLARD96] a opté pour une création numérique formellement exacte des équations du mouvement qui n'implique pas d'approximation numérique méthodologique. En effet, grâce à un développement manuel des équations de Lagrange, l'algorithme n'effectue pas d'approximation méthodologique au moment de la création du système d'équations. Ce qui permet de garder les avantages de rapidité et d'efficacité du numérique, ainsi que les avantages du symbolique grâce à l'exactitude formelle du système d'équation.

## **4. Contrôle dynamique**

Le contrôle du mouvement d'un système dynamique revient à appliquer des forces et des couples au modèle mécanique représentant le corps articulé de manière à ce que l'évolution du système corresponde au mouvement désiré. Si les forces agissant sur l'objet pour lequel on veut contrôler le mouvement sont spécifiées à chaque pas de simulation, on peut calculer

aisément la position et la vitesse de l'objet en intégrant les équations du mouvement au cours du temps. Ce problème est appelé problème de la valeur initiale, l'état initial étant connu et les états successifs pouvant être calculés à partir des états précédents. L'approche est applicable à des systèmes simples ou passifs, mais la détermination des forces nécessaires pour faire évoluer le système mécanique conformément à une spécification donnée n'est pas triviale. Dans le contexte du contrôle d'un humanoïde, le système à contrôler est hautement non linéaire et adaptatif, et comporte des éléments passifs, caractérisés par une impédance mécanique et des éléments actifs, responsables de l'activation des muscles. De plus, si le système est un corps poly-articulé comportant un excès de d.d.l., il existe une infinité de solutions répondant à la spécification donnée. Pour de tels systèmes, il n'existe pas de méthode générale permettant de concevoir un contrôleur unique capable de piloter les différents paramètres du système. Il est nécessaire de définir des procédures de régularisation pour identifier une solution dans l'ensemble des solutions potentielles. Parmi les solutions proposées, deux directions principales peuvent être retenues : le contrôle par contraintes et les méthodes issues de la théorie du contrôle en automatique.

#### **4.1. Gestion des contraintes**

Les contraintes interviennent de plusieurs manières dans la modélisation de l'animation : contrôler l'évolution des paramètres d'animation (limitation d'angle de rotation, de translation, ...), ou pour lier l'évolution relative de plusieurs paramètres entre eux (vissage, création de boucle cinématique, ...). Leur gestion diffère selon la méthode de création des équations du mouvement. Elles sont en effet naturellement intégrées aux équations de Lagrange par l'ajout de multiplicateurs de Lagrange [WIELAND94] ou par une méthode de pénalisation [ARNALDI88][DUMONT90], alors qu'il faut des développements spécifiques dans le cas des méthodes basées sur Newton-Euler [SCHRODER90][HYEONGSEOK94][KOMURA97] et ce, notamment pour la gestion des boucles cinématiques créées par l'introduction de contraintes.

Certains moteurs d'animation utilisent les contraintes comme la base de la modélisation de l'animation. Gascuel [GASCUEL94] gère les objets à animer comme des entités indépendantes et limite leurs évolutions relatives grâce à des contraintes reliant des points spécifiques des objets à des surfaces d'évolutions permises.

De plus, de nombreux travaux visent à contrôler le mouvement à l'aide de contraintes sur les paramètres d'animation. Les méthodes basées sur la dynamique inverse [VASILO.91] consistent, notamment, à spécifier les trajectoires d'évolution des paramètres et à calculer les efforts nécessaires au maintien de ces contraintes lors de l'animation. Le maintien est alors assuré par la minimisation d'une fonction d'énergie qui peut par exemple représenter une distance entre la position réelle et la position souhaitée. C'est le cas de Barzel et Barr [BARZEL88] qui se basent sur une évaluation itérative de l'écart de réalisation d'une contrainte pour atteindre un minimum acceptable.

Ils partent du problème suivant :

Soit un ensemble d'objets décrits par leurs positions, vitesses, accélérations, soumis à des forces et couples fonctions du temps et un ensemble de contraintes géométriques. Comment trouver les efforts, à chaque instant  $t$ , permettant de satisfaire au mieux les contraintes géométriques.

Ils expriment donc la formule  $D$  à minimiser (car elle atteint rarement zéro) et en déduisent l'équation de retour exponentiel à la réalisation :

$$\ddot{D} + \frac{2}{h}\dot{D} + \frac{1}{h^2}D \quad (\text{E } 7)$$

où  $D$  représente l'écart de réalisation de la contrainte et  $h$  un paramètre temporel permettant de pondérer la vitesse de retour à la réalisation.

Il est alors possible, pour chaque contrainte, d'exprimer son accélération, sa vitesse et sa valeur en fonction des forces, couples, paramètres initiaux imposés au système et des forces « de contrainte » inconnues permettant de la réaliser.

Il suffit ensuite de minimiser le terme (E 7) jusqu'à obtenir un résultat satisfaisant.

Dans le cas de contraintes liées entre elles, il faut alors résoudre un système d'équations du type (E 7).

L'avantage de cette méthode repose sur le contrôle continu du mouvement. Elle nécessite cependant de nombreux calculs ce qui rend son utilisation fastidieuse.

De même, Witkin et Kass [WITKIN88] utilisent une formulation physique fondée sur les équations de Lagrange en coordonnées généralisées et ajoutent la notion de contrainte de positionnement dans le temps. Le but de l'algorithme est de trouver le mouvement utilisant le minimum d'énergie et satisfaisant aux conditions des contraintes. Ils cherchent donc à minimiser une quantité d'énergie du type :

$$R = \int_{t_0}^{t_1} |f(t)|^2 dt \quad \text{où } f \text{ est la force à minimiser au cours du mouvement. Le temps étant}$$

discrétisé, l'expression des vitesses  $\dot{q}$  et accélérations  $\ddot{q}$  au temps  $i$  est obtenue par :

$$\dot{q}_i = \frac{q_i - q_{i-1}}{h} \quad \text{et} \quad \ddot{q}_i = \frac{q_{i+1} - 2q_i + q_{i-1}}{h^2}$$

et l'intégrale à minimiser est remplacée par une somme sur les  $f_i$ , la discrétisation temporelle utilisée est proche d'une intégration par la méthode d'Euler, ou d'une résolution par différences finies

La résolution du système non linéaire est construite à partir d'une méthode dérivée de la méthode de Newton-Raphson. Elle consiste en une résolution par itérations successives à partir d'une solution approchée.

La souplesse du contrôle du mouvement de cette méthode est contrebalancée par des coûts en temps de calcul importants.

[COHEN92], puis [ROSE96], reprennent et améliorent le concept afin de contrôler l'animation de façon interactive.

Pour éviter des problèmes de dérive numérique du système et stabiliser plus facilement celui-ci, il peut être nécessaire d'introduire des termes de retour amorti, visqueux à la réalisation comme dans [BARZEL88, PLATT92].

[BARAFF89] présente un modèle analytique pour trouver la force à appliquer afin d'empêcher deux objets polyédrique de s'interpénétrer. Il poursuit ensuite ses travaux en étendant son système aux objets courbes [BARAFF90]. [DUMONT90] propose un algorithme de résolution de telles interactions pour des objets rigides et des objets déformables.

#### 4.2. Gestion du contrôle en automatique

L'un des objectifs du contrôle en automatique consiste à concevoir des lois de commande de systèmes dynamiques de façon à contrôler l'évolution de ces systèmes par rapport à une trajectoire de référence (consigne). Il s'agit bien également ici de contrôler le mouvement.

La démarche de modélisation comporte plusieurs étapes. La première étape consiste à définir un modèle approché du processus physique à contrôler. Il s'agit ensuite de concevoir une loi de contrôle pour ce système. La dernière étape est l'analyse et la simulation du comportement du système contrôlé.

La qualité de la loi de commande est évaluée à partir de la mesure des temps de réponse, de la stabilité et de la précision.

La théorie du contrôle comporte plusieurs branches d'études. La mieux formalisée et la plus facile à mettre en œuvre est la théorie de la commande linéaire, qui s'applique à des systèmes linéaires représentés dans l'espace d'état par l'équation :

$$\dot{y} = Ay + Bu \quad (\text{E } 8)$$

où  $A$  représente la matrice d'évolution du système et  $B$  la matrice de commande.

En robotique et en animation par ordinateur, une solution fréquemment utilisée est celle qui consiste à répartir un ensemble de contrôleurs sur la structure articulée. Ces contrôleurs, de type Proportionnel-Intégral-Dérivé (PID), prennent en entrée des objectifs à atteindre pour chacune des articulations, sous la forme d'états désirés :  $(q_d, \dot{q}_d)$ , et calculent les couples qui induisent les mouvements désirés :

$$\tau = K_p(q - q_d) - K_d(\dot{q} - \dot{q}_d) - K_i \int (q - q_d) dt \quad (\text{E } 9)$$

Si les méthodes d'étude des systèmes linéaires à partir de la notion de fonction de transfert et de ses représentations (réponse en fréquences, disposition des pôles et zéros) constituent des outils très bien formalisés pour l'analyse et la synthèse des systèmes asservis, il s'avère que ces méthodes ne sont pas adaptées aux systèmes présentant des non linéarités. Or, les systèmes physiques que nous cherchons à contrôler sont par nature fortement non linéaires. De plus, la prise en compte des éléments actifs du système nous incite à nous tourner vers des lois de commande elles-mêmes non linéaires.

Dans le cadre de la conception de contrôleurs de systèmes non linéaires, la plupart des approches tentent de transposer les outils méthodologiques développés pour le contrôle linéaire dans le domaine du contrôle non linéaire. C'est le cas des techniques de linéarisation par retour d'état [RUGH 1991]. Le contrôle robuste étudie le comportement du système en présence de bruits liés à la fois aux erreurs de modélisation et à l'environnement extérieur. Dans ce cas, la conception du contrôleur est guidée par la minimisation de la sensibilité aux perturbations. La théorie du contrôle adaptatif, qui s'applique à des systèmes dont le comportement ou les stratégies de contrôle varient au cours du temps, fait l'objet de nombreux travaux de recherche. Deux formes de contrôle adaptatif sont généralement considérées. La première est basée sur un modèle de référence : un modèle du comportement du système contrôlé est utilisé pour générer les entrées du système, sur la base de la comparaison entre le comportement observé et les prédictions du modèle. Cette stratégie de contrôle exploite des données expérimentales. Le contrôle paramétrique adaptatif consiste quant à lui à commander des systèmes dont la structure dynamique est connue et dont les paramètres varient lentement au cours du temps.

En animation par ordinateur, les contrôleurs PID peuvent être utilisés pour contrôler des systèmes articulés. Couplée à des contrôleurs centralisés, du type automates à états finis, cette méthode est également exploitée pour simuler la locomotion humaine [RAIBERT91],



[VANDEPANNE90], [VANDEPANNE94], [LASZLO96], [HODGINS95], [HODGINS96]. Ces approches permettent la synthèse de mouvements physiquement valides, à partir d'une spécification expressive pour le concepteur. Cependant, la détermination des gains des contrôleurs reste relativement fastidieuse et les mouvements obtenus sont plus proches de ceux des robots que de ceux des humains. De plus, ces techniques nécessitent de segmenter le mouvement en postures clés ou en phases clés pour des mouvements cycliques et ne permettent pas de prendre en compte les modifications de l'environnement. À partir d'un modèle dynamique linéarisé, Nougaret [NOUGARET96] propose d'appliquer des méthodes de l'automatique pour la conception de contrôleurs « clé en main » dédiés à la locomotion dynamique.

Un autre modèle de contrôle d'un bras articulé mécanique a été développé [GIBET92], [GIBET94]. Ce modèle exploite deux boucles adaptatives imbriquées. Une première boucle adaptative traduit un contrôle en position analogue à celle qui s'applique au modèle géométrique [GIBET94a]. Une fonction potentielle exprimant l'écart entre la position de l'extrémité du bras et de la cible est utilisée pour calculer la valeur des déplacements angulaires des articulations à chaque pas de simulation. Le procédé mis en œuvre pour minimiser cette erreur est celui de descente du gradient. Une seconde boucle adaptative, plus rapide que la précédente introduit un contrôle en force dont le rôle est d'assurer un comportement stable du système bouclé et de réaliser des mouvements précis. L'effet de cette rétroaction consiste à appliquer à chaque articulation des impulsions de force rapides dépendant à chaque pas temporel des directions relatives du mouvement et du gradient du signal d'erreur. Les forces internes sont calculées à partir d'une transformation non linéaire  $\Psi$  appliquée aux dérivées successives des déplacements angulaires :  $\delta F = \Psi(\delta q, \dot{q}, \ddot{q})$

## **5 La méthode des éléments finis**

Une autre méthode d'animation physique, basée sur la théorie des éléments finis, existe et offre de bons résultats. Son objectif majeur est de prendre en compte la déformation des objets, elle peut être adaptée à la prise en compte de la déformation des tissus humains et de la peau. Elle repose sur un échantillonnage 2D ou 3D des objets intervenant dans l'animation, on affecte ensuite des paramètres relatifs au matériau à chacun des éléments du maillage, puis pour chaque étape de la simulation, on déduit le déplacement des éléments en fonction des forces appliquées grâce aux lois physiques vues précédemment, Principe Fondamental de la dynamique (Newton-Euler) ou approche Lagrangienne. Cette approche a été décrite par Terzopoulos [TERZOPOULOS87] et mixée avec le formalisme pour des solides rigides [TERZOPOULOS88]. Elle a également été utilisée pour l'interaction d'une main avec une sphère déformable lors d'une saisie dans [GOURET89].

La qualité du résultat est intrinsèquement liée à la qualité du maillage, or le calcul, s'effectuant en chaque élément du maillage, voit sa complexité augmenter proportionnellement à la qualité (finesse) du maillage. Cette méthode présente donc, d'une part le désavantage d'être coûteuse en temps de calcul, et d'autre part de voir ce temps de calcul directement dépendant du résultat souhaité, mais elle a l'avantage d'avoir un formalisme simple et bien défini. Il est à noter que certains travaux ont été effectués pour accélérer les calculs, mais ceux-ci reposent généralement sur une linéarisation et un pré-calcul des efforts dus aux matériaux [COTIN97, PICINBONO01], contraignant ainsi le système à de petits mouvements.

En se basant sur la méthode des éléments finis, McCarthy et al. [MCCARTHY90] ont pu modéliser un genou virtuel à 6 DdL et en animer les différents éléments de manière réaliste. Ils ont ainsi pu comparer leurs résultats à la littérature, trouvant des résultats

concordants, prouvant ainsi, s'il en était besoin, la robustesse de la méthode, ainsi que sa lourdeur en ressources temporelles et mémorielles, puisque le calcul de l'animation a pris 17 heures sur une station SGI Origin 2000 comportant 4 processeurs R10K à 250 MHz. Et 2560 Mo de RAM.

Liesenfeld et al. [LIESENFELD96] utilisent les éléments finis avec les lois lagrangiennes pour animer dynamiquement des corps élastiques.

## **6. Retour sur la résolution du système d'équations**

Toutes les techniques ou approches décrites ici induisent la résolution de systèmes d'équations (souvent différentielles et non linéaires).

Les différentes méthodes de résolution de ce système d'équations du mouvement influent de façon importante sur la qualité du résultats et sur les temps de calculs. Comme dans de nombreux cas en synthèse d'image, il faut trouver un bon compromis entre vitesse d'exécution et qualité du résultat.

Les techniques à base de différences finies, pour la partie différentielle des équations, utilisent directement la fonction d'énergie cinétique du mouvement et l'évaluent plusieurs fois en des points  $(q, \dot{q})$  voisins, afin d'en estimer les dérivées. À l'opposé de cette méthode, qui utilise une approximation méthodologique, on trouve des méthodes basées sur la résolution de système d'équations différentielles. C'est le cas des méthodes qui évaluent les vitesses et positions grâce aux accélérations à une étape donnée et aux vitesses et positions de l'étape précédente. Ainsi que des méthodes dites de prédiction / correction qui utilisent la propriété que les vitesses et les positions varient peu entre deux étapes successives.

L'autre partie des équations, la partie non linéaire, est gérée par des méthodes de type « Newton-Raphson », ou des méthodes basées sur celle-ci. Il s'agit de résolutions itératives qui « approchent » la solution en convergeant.

## **7. Cas particulier : Animation de corps humain**

Enfin, des modèles d'animation spécifiques au corps humain ont aussi été développés. La marche notamment a fait l'objet de nombreuses recherches [BRUDERLIN89, BRUDERLIN96], [McKENNA90], le système mécanique étant connu, les équations peuvent être créées spécifiquement.

A.P. Wieland [WIELAND94] utilise le formalisme Lagrangien avec multiplicateurs de Lagrange pour développer manuellement (avec MAPLE V) les équations du mouvement spécifique à la marche humaine. L'analyse du mouvement et des contraintes à imposer au mouvement des jambes (Données anthropométriques, limitations angulaires des articulations, ...) lui permet alors d'obtenir une simulation réaliste et rapide de la marche.

L'utilisation de données anthropométriques, ainsi que la modélisation de corps humains plus complexes que de simples solides, (prise en compte des muscles, tendon, articulation réaliste, ...) offre de nouvelles possibilités qui permettent d'obtenir des animation réalistes, mais qui implique généralement des coûts en temps de calcul important [THOW HING94]. Komura et Yoshihisa [KOMURA97] ont basé leur modèle d'animation sur les muscles qui régissent le mouvement du corps humain. Les données propres de chaque muscle (force, poids, taille, ...), ainsi que les contraintes qui leurs sont imposées permettent un certain contrôle de l'animation. Le problème principal de ce genre d'approche est d'arriver à exprimer les courbes d'efforts de chaque muscle pour obtenir un mouvement volontaire. Victor Ng Thow Hing [THOW HING94] modélise un corps humain sur la base des muscles

et tendons qui le composent. L'animation est obtenue par leurs actions sur le squelette et des propriétés de déformation et de conservation du volume des muscles au cours du mouvement.

De son côté, Wooten [WOOTEN96] présente un système permettant d'animer un humain virtuel plongeant et nageant à partir de paramètres de hauts niveaux. L'utilisateur peut, par exemple, spécifier la hauteur du saut et le type de nage, l'animation est alors calculée en se basant sur un système d'équations Lagrangien. Sur le même principe, Hodgins [HODGINS98] propose un système capable de calculer des animations réalistes d'humains virtuels lors de pratique sportive de cyclisme, course ou de salto. Ces animations ont ensuite été comparées à des films de sportifs réels et offrent un résultat réaliste. Ces animations profitent du fait que l'animation étant spécifique à une action donnée, il est possible de spécialiser le modèle ou les contraintes afin d'éviter les calculs superflus. Le genou est, par exemple, réduit à une simple articulation à 1 DdL puisque l'animation ne nécessite pas plus de réalisme, et la colonne vertébrale est elle, carrément réduite à un segment. Par ailleurs, le fait de connaître l'animation permet de ne définir que les contraintes nécessaires, comme par exemple le sol dans le cas de la course ou le plongeur pour le plongeur.

## **Bibliographie**

[ARMSTRONG85] Armstrong W., Green M. « The dynamics of articulated rigid bodies for purposes of animation ». *Graphics Interface 85*, 407-415, Montréal, Mai 1985.

[ARNALDI88] Arnaldi B. « Conception du noyau d'un système d'animation de scènes tridimensionnelles intégrant les lois de la mécanique ». PhD Thesis, Université de Rennes I, 1988.

[ARNALDI89] Arnaldi B., Dumont G., Hégron G. « Dynamics and unification of animation control ». *The Visual Computer*, 5, 22-31, 1989.

[ARNALDI94] Arnaldi B. « Modèles physiques pour l'animation ». Document d'habilitation à diriger des recherches, Université de Rennes 1, 1994.

[BARAFF89] Baraff D. « Analytical Methods for Dynamic Simulation of Non-Penetrating Rigid Bodies », *Proc. SIGGRAPH '89*, 223-232, 1989.

[BARAFF90] Baraff D. « Curved surfaces and coherence for non-penetrating rigid body simulation », *Proc. SIGGRAPH '90*, 19-28, 1990.

[BARZEL88] Barzel R., Barr A. « A modeling system based on dynamic constraints ». *Computer Graphics*, 22(4), 179-188, Août 1988.

[BRUDERLIN84] Bruderlin A.W. « Goal-directed, dynamic animation of bipedal locomotion ». PhD Thesis, Simon Fraser University, 1988.

[BRUDERLIN89] Bruderlin A., Calvert T.W. « Goal-directed, dynamic animation of human walking ». *Computer Graphics (Siggraph 89)*, 233-242, 1989.

[BRUDERLIN96] Bruderlin A., Calvert T. « Knowledge-driven, interactive animation of human running ». *Graphics interface'96*, pp213-221, 1996.

[COHEN92] Cohen M.F. « Interactive Spacetime Control for Animation », *Proc. Siggraph'92*, 293-302, 1992.

[COTIN97] Cotin S. « Modèles anatomiques déformables en temps-réel ». Thèse de doctorat. Université de Nice-Sofia Antipolis, 1997.

[COZOT96] Cozot R. « Environnement de simulation de systèmes physiques : modèle et langage ». PhD Thesis, Université de Rennes I, 1996.

- [DUMONT90] Dumont G. «Animation de scènes tridimensionnelles : la mécanique des solides comme modèle de synthèse du mouvement ». PhD Thesis, Université de Rennes I, 1990.
- [FALOUTSOS95] Faloutsos P. « Physics-based animation and control of flexible characters ». PhD Thesis, University of Toronto, 1995.
- [FANG03] Fang A.C. and Pollard N.S. « [Efficient Synthesis of Physically Valid Human Motion](#) ». ACM Transactions on Graphics 22(3) 417-426, SIGGRAPH Proceedings, 2003.
- [GASCUEL94] Gascuel J.D., Gascuel M.P. « Displacement constraints for interactive modeling and animation of articulated structures ». *The Visual Computer*, 10, 191-204, 1994.
- [GIBET92] Gibet S. et Marteau P.F. *Non linear feedback model of sensori-motor systems*. The International Conference on Automation, Robotics and Computer Vision, Singapore, 1992.
- [GIBET94] Gibet S., Marteau P.F. A Self-Organized Model for the Control, Planning and Learning of Nonlinear Multi-Dimensional Systems Using a Sensory Feedback, *Journal of Applied Intelligence*, vol.4, pp. 337-349, 1994.
- [GIBET94a] Gibet S., Lebourque T. Automatic Motion Control, Eurographics Workshop on Animation and Simulation, Oslo, 1994.
- [GILLARD96] Gillard D., Nourrit J.M., Lucas L., Remion Y. « A generic algorithm for the generation of physical movements of solid systems », *Compugraphics'96 proceedings*, 201-210, Paris, Dec. 1996.
- [GOURRET89] Gourret J. P., Magnenat Thalmann N., Thalmann D., « Simulation of Object and Human Skin Deformations in a Grasping task », "SIGGRAPH'89", vol 23, pp.21-30, july 1989.
- [HODGINS95] Hodgins J.K. Animating human athletics. *Proceeding of ACM SIGGRAPH*, Los Angeles, California, août 1995.
- [HODGINS96] Hodgins J.K. Three dimensional human running. *Proceedings of the IEEE conference on Robotics and Automation*, avril 1996.
- [HODGINS98] Hodgins J.K. and Wooten W.L. « Animating Human Athletes ». *Scientific American*, 278(3), 64-69, 1998. <http://www.gvu.gatech.edu/~jessica.hodgins/>
- [HOLLERBACH80] Hollerbach J.M., « A recursive lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity ». *IEEE Transactions on systems, man, and cybernetics*, 730-736, 1980.
- [HYEONGSEOK94] Hyeongseok K. « Kinematic and dynamic techniques for analyzing, predicting, and animating human locomotion ». PhD Thesis, University of Pennsylvania, 1994.
- [ISAACS88] Isaacs P.M., Cohen M.F. « Mixed method for complex kinematic constraints in dynamic figure animation ». *The Visual Computer*, 2(4), 296-305, Decembre 1988.
- [KOMURA97] Komura T. Yoshihisa S. « A muscle-based feed-forward controller of the human body ». *Computer Graphics Forum, Proc.*, 16(3), 165-176, 1997.
- [LASZLO96] Laszlo J., van de Panne M., Fiume E. Limite cycle control and its application to the animation of balancing and walking. *Computer Graphics*, pp. 155-162, août 1996.
- [LIESENFELD96] Liesenfeld R.L.W. and Stolfi J. « Dynamic Animation of Elastic Bodies ». *Anais do IX Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens (SIBGRAPI'96)*, 265-272, 1996.
- [MCCARTHY90] McCarthy A.D., Wilkinson I.D., Hose D.R., Barber D.C., Wood S., Darwent G., Chan D., Bickerstaff D.R. « Musculo-Skeletal Simulation: Finite Element Meshes Derived From Magnetic Resonance Volumes ». *Proc. of the International Society for Magnetic Resonance in Medicine*, May 2002.

[MENDOZA03] Mendoza C., Laugier C., Galizzi O., Faure F. « Dynamic Simulation and 3D Interaction ». Quatrièmes Journées Nationales de Recherche en Robotique (JNRR), Clermont-Ferrand, France, oct. 2003.

[NOUGARET95] Nougaret J.L. et Arnaldi B. Spécification cinématique et simulation dynamique : quelle combinaison optimale pour l'animation ? In : Numéro Spécial « Journées d'informatique graphique », Revue Internationale de CFAO et d'Informatique Graphique, éd. Hermes, 1995.

[NOUGARET96] Nougaret J.L. Contrôle du mouvement pour l'animation. Thèse de Doctorat en informatique, Université de Rennes 1, juin 1996.

[PICINBONO01] Picinbono G. « Modèles géométriques et physiques pour la simulation d'interventions chirurgicales ». Thèse de doctorat. Université de Nice-Sofia Antipolis, 2001.

[PLATT92] Platt J. « A Generalization of Dynamic Constraints ». *Graphical Models and Image Processing*, vol 54 no 6, 516-525, 1992.

[REMION00] Remion Y. « Animation Dynamique : moteur lagrangien généraliste et applications ». Habilitation à Diriger des Recherches, Université de Reims Champagne-Ardenne, 2000.

[RAIBERT91] Animation of dynamic legged locomotion. *Computer Graphics*, pp. 349-358, 1991.

[ROSE96] Rose C. and Guenter B. and Bodenheimer B. and Cohen M.F. « Efficient Generation of Motion Transitions using Spacetime Constraints ». *Computer Graphics*, 30, 147-154, 1996.

[SCHRODER90] Schröder P., Zelter D. « The virtual erector set : dynamic simulation with linear recursive constraint propagation ». *Computer Graphics*, 24, 179-188, 1990.

[TERZOPOULOS87] Terzopoulos D., Platt J., Barr A. et Fleischer K. « Elastically Deformable Models ». *Siggraph '87*, pp 205-214, 1987.

[TERZOPOULOS88] Terzopoulos D. et Witkin A. « Physically based models with rigid and deformable components ». *IEEE Computer Graphic and Applications*, pp41-51, 1988.

[THOW HING94] Thow Hing V. « **A biomechanical musculotendon model for animating articulated objects** ». **PhD Thesis, University of Toronto, 1994.**

[VANDEPANNE90] Van de Panne, M., Fiume, E., Vranesic, Z. Reusable motion synthesis using state-space controllers, *ACM Computer Graphics*, pp. 225-234, 1990.

[VANDEPANNE94] Van de Panne M., Kim R., Fiume E. Virtual wind-up toys for animation. *Graphics Interface*, pp. 208-215, Banff, Alberta, Canada, mai 1994.

[VASILO.91] Vasilonikolidakis N.A., Clapworthy G.J. « Inverse Lagrangian dynamics for animating articulated models ». *The Journal of Visualization and Computer Animation*, 2, 106-113, 1991.

[VILLARD95] Villard D. « D'un formalisme énergétique à la résolution des équations du mouvement : l'usage de la différentiation et des graphes ». Master's Thesis, Université de Rennes 1, 1995.

[WIELAND94] Wieland A.P. « Modeling, simulation, and control of two-legged walking ». PhD Thesis, University of California, Los Angeles, 1994.

[WILHELMS85] Wilhelms J., Barski B. « Using dynamic analysis to animate articulated bodies as humans and robots ». *Graphics Interface 85*, 97-104, Montréal, Mai 1985.

[WITKIN88] Witkin A., Kass M. « Spacetime constraints ». *Computer Graphics*, 22(4), 159-168, Août 1988.

[WITT77] Wittenburg J. « Dynamics of systems of rigid bodies ». Teubner, Stuttgart 1977.

**[WOOTEN96]** Wooten Wayne L. and Hodgins Jessica K. « Animation of Human Diving ». *Computer Graphics Forum*, 15(1), 3-14, 1996.

**[ZELTZER82]** Zeltzer D. « Motor control techniques for figure animation ». *IEEE Computer Graphics and Applications*, 2, 53-59, 1982.

## Troisième Partie : conclusion générale et perspectives

L'objectif de ce document est de dresser un état des lieux des travaux liés à l'animation temps réel d'humains virtuels. La principale difficulté consiste à gérer le compromis entre temps de calcul et réalisme, comme c'est aussi le cas en rendu d'images. D'un côté, il existe des animations basées sur des données cinématiques, par exemple issues de captures du mouvement. Nous avons vu que plusieurs traitements étaient nécessaires pour utiliser ce type de données sur un personnage de taille différente évoluant dans un environnement contraint. On peut alors s'interroger sur l'effet que ces traitements ont sur le réalisme visuel. Les techniques basées sur la gestion d'une vaste base de données autorisant des interpolations entre mouvements modifient moins le geste capturé. On pourrait donc conclure que ce sont elles qui garantissent peut-être le meilleur réalisme pour un coût de calcul moindre. Il est toutefois nécessaire de mettre en œuvre une base de données volumineuse et un prétraitement pour assurer des transitions cohérentes entre les gestes. Plus la base de données est dense et plus l'interpolation linéaire entre mouvements proches peut produire un résultat réaliste. En effet, aucun travail ne démontre qu'un geste peut être issu de la combinaison linéaire de plusieurs autres, sans perte de réalisme, à moins d'échantillonner fortement l'espace des gestes possibles. De plus, il paraît déraisonnable de vouloir créer une base de données universelle et dense de gestes pour être capable de faire face à toutes les situations (prise en compte de forces externes différentes, par exemple).

D'un autre côté, la dynamique pose un cadre plus général qui permet de générer des mouvements capables de s'adapter à un grand nombre d'environnement, sans nécessiter une base de données volumineuse de gestes capturés. Néanmoins, le problème ici est de produire un geste ressemblant à celui qu'aurait effectué un humain dans la même situation, ce qui n'est qu'une solution possible parmi un large espace de mouvements respectant les lois de la mécanique. De plus, un grand nombre de travaux restent à mener, en collaboration avec d'autres disciplines comme la biomécanique et la robotique pour proposer des modélisations plus fines du squelette et pour concevoir de nouveaux types de contrôleurs. D'ailleurs, la simulation dynamique s'impose lorsque l'on souhaite effectuer des simulations « exactes » dans des applications médicales. Par contre, pour l'animation temps réel d'humains virtuels, malgré des travaux récents, il reste difficile de produire des mouvements que l'on pourrait visuellement attribuer à un humain. Dans plusieurs applications, la dynamique est utilisée pour simuler avec succès le comportement passif d'un humain virtuel soumis à des forces externes (crash-tests, dans les jeux vidéos impliquant des impacts...).

Comme nous venons de le voir, ces deux approches posent encore de nombreux problèmes à résoudre pour l'avenir. Il nous semble qu'à l'avenir, en plus de recherches spécialisées dans les deux domaines, il soit intéressant de mener des travaux couplant les deux points de vue. Il faudrait trouver un cadre dans lequel la dynamique viendrait en renfort de la cinématique lorsque les contraintes externes deviennent trop importantes. A l'inverse, on pourrait envisager un renfort de la cinématique pour produire des trajectoires « humaines » en sortie d'un simulateur en guidant les contrôleurs. Ceci démontre un travail pluri-disciplinaire impliquant une mise en commun des outils logiciels. Ce type de travail devra donc certainement s'appuyer sur une mutualisation des moyens provenant de divers laboratoires afin de lever ces verrous technologiques.