



HAL
open science

Identification de modèles physiques et de contrôleurs en animation

Bogdan Stanciulescu

► **To cite this version:**

Bogdan Stanciulescu. Identification de modèles physiques et de contrôleurs en animation. Synthèse d'image et réalité virtuelle [cs.GR]. Institut national polytechnique de Grenoble, 2002. Français. NNT: . tel-01246219

HAL Id: tel-01246219

<https://hal.science/tel-01246219>

Submitted on 18 Dec 2015

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

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

AVANT PROPOS

Ce travail de thèse a fait partie d'une coopération entre l'Ecole Nationale Supérieure de Techniques Avancées, Paris, et le laboratoire ACROE/ICA (Informatique et Création Artistique) de l'Institut National Polytechnique de Grenoble avec le soutien de la Délégation Générale pour l'Armement, Direction des Systèmes et de la Prospective, Service des Recherches et Etudes Amont.

Il a été co-encadré par le Laboratoire d'Électronique et d'Informatique d'ENSTA (LEI) et le Laboratoire ACROE/ICA. Il sera impossible de citer toutes les personnes impliquées ici. J'espérerais donc que tous ceux qui ne sont pas cités me pardonnent.

Je tiens d'abord à remercier M. Jean LOUCHET, Ingénieur en chef de l'Armement, qui a eu l'ouverture d'esprit pour me proposer ce sujet de thèse et cet encadrement. Il est impossible pour moi d'imaginer ma vie en France sans ses conseils, sa camaraderie et sa porte toujours ouverte. Ces qualités sont également celles de ses collègues de Grenoble, Mme Annie LUCIANI et M. Jean-Loup FLORENS, M. Claude CADOZ de l'ACROE et je regrette de ne pas avoir passé plus de temps dans leur équipe pour mieux profiter de notre collaboration.

Je tiens à remercier M. Alain SIBILLE et M. André PICAUD, les responsables du Laboratoire d'Electronique et d'Informatique de l'ENSTA, de m'avoir aidé à m'intégrer dans leur équipe.

Je tiens également à remercier tous les gens du Laboratoire d'Electronique et d'Informatique (LEI) qui ont su être près de moi, particulièrement dans les moments les plus difficiles de ma thèse.

Pour leurs divers conseils, j'aimerais remercier M. Philippe GUERMEUR, Mme Jacqueline DARROZES, M. Omar HAMMAMI, et tous ceux qui m'ont accueilli chaleureusement pendant ce séjour, intéressant et fructueux.

Je remercie tous les étudiants en doctorat et D.E.A. de l'équipe avec qui j'ai partagé beaucoup de moments, dans un laboratoire scientifiquement riche et dans un cadre très multiculturel.

Enfin, je n'en serais pas ici sans le soutien de ma famille, mes très chers Rasvan, mon frère, ma mère Elena et surtout mon père Constantin. Je leur dédie cette thèse. Je n'oublie pas le soutien pendant toutes ces années de Corina, mon amie, et celui de ma belle-sœur Ligia avec sa grande générosité.

RESUME

Les modèles à base de particules et de liaisons sont bien adaptés aux simulations du mouvement sur ordinateur, notamment en synthèse d'images animées. Une limitation pratique provient de l'utilisation de liaisons purement visco-élastiques qui restreint la classe des mouvements autorisés aux comportements passifs et énergétiquement dissipatifs. Un autre point dur de la mise en oeuvre de ces modèles est la méthode de construction et d'identification garantissant le réalisme des comportements synthétisés.

L'objet principal de cette thèse est d'élargir le domaine d'applicabilité des modèles masses-liaisons par l'introduction de la motricité. Pour cela, nous avons introduit deux variantes de liaisons actives ("muscles.") dont nous étudions les caractéristiques, leurs liens avec le TGR (Transducteur Gestuel Rétroactif) de l'ACROE et la proprioception musculaire.

Le premier type de muscle se caractérise par un système de contrôle central conçu comme un système physique indépendant, et une structure de quadripôle. Ce concept est testé dans une application de suivi de cible par un mobile, où une boucle visio-motrice connecte un mobile au contrôleur central.

Le deuxième type de muscle utilise un contrôle local neuronal.

L'identification des systèmes masses-liaisons contenant des muscles de l'un ou l'autre type s'effectue par un algorithme d'évolution artificielle qui optimise les paramètres de contrôle ainsi que les paramètres mécaniques. Elle consiste à ajuster le modèle en minimisant une distance mathématique entre le comportement de référence et le comportement synthétisé par le modèle; des caractéristiques auto-adaptatives permettent de limiter la part d'intervention de l'utilisateur dans l'identification via l'automatisation du choix de certains paramètres de la stratégie d'évolution. La méthode a été testée sur des données cinématiques de synthèse et des données réelles.

SPECIALITE : Imagerie, Vision et Robotique

MOTS-CLES : Synthèse d'images, Animation par ordinateur, Analyse du mouvement, Modèles physiques, Physique algorithmique, Motricité, Identification de modèles, stratégies d'évolution, réseaux de neurones

INTITULE ET ADRESSE DU LABORATOIRE : Laboratoire d'Electronique et d'Informatique de l'Ecole Nationale Supérieure de Techniques Avancées (32 Boulevard Victor, 75739 Paris), en coopération avec l'INPG/ACROE (46 Avenue Felix Viallet, 38000 Grenoble) et le soutien de la DGA (Direction des Systèmes et de la Prospective, Service des Recherches et Etudes Amont).

PLAN DE LA THESE

AVANT PROPOS	2
RESUME	3
CHAPITRE I	6
INTRODUCTION	6
CHAPITRE II	9
LES MODELES PHYSIQUES EN ANIMATION	9
2.1. <i>Modèles cinématiques et modèles dynamiques (physiques)</i>	9
2.1.1. Modèles descriptifs ou cinématiques.....	10
2.1.2. Modèles générateurs ou dynamiques.....	12
2.1.3. Modèles du solide rigide	13
2.1.4. Modèles de solide déformable.....	16
2.2. <i>Cordis-Anima</i>	17
2.3. <i>Animation de créatures par modèle physique</i>	24
CHAPITRE III	27
IDENTIFIER UN MODELE A PARTIR DE LA CINEMATIQUE : LA TECHNIQUE EVOLUTIONNAIRE.....	27
3.1. <i>Introduction aux méthodes évolutionnaires</i>	29
3.2. <i>Stratégie d'évolution: un paramétrage adapté aux grandeurs physiques</i>	29
3.2.1. Fonctions de coût multiples ; coût global à long et court terme, coût local.....	33
3.3. <i>L'approche parisienne</i>	38
3.4. <i>Auto-adaptation</i>	41
CHAPITRE IV	48
APPLICATIONS ET LIMITES	48
4.1. <i>Identification générale et résultats au bruit sur des données synthétiques</i>	48
4.2. <i>Tissus</i>	49
4.3. <i>Mécanique des fluides</i>	50
4.4. <i>Cheval</i>	51
CONCLUSION: NECESSITE D'ENRICHIR LA MODELISATION	52
CHAPITRE V	53
INTRODUCTION DE LA MOTRICITE DANS CORDIS	53
5.1. <i>Modèle élémentaire de locomotion proprioceptive à base de modèle physique. Principes</i>	53
5.2. <i>Contrôle local ou contrôle central ?</i>	55

5.3. <i>Le problème de l'injection d'énergie</i>	56
5.4. <i>Muscle à contrôle central</i>	57
5.4.1. Transducteur gestuel rétroactif (TGR). Muscle.....	57
5.4.2. Contrôle musculaire par couplage de liaisons	59
5.4.3. Le quadripôle et les transmetteurs	60
5.5. <i>Couplage mutuel de liaisons par contrôle neuronal local</i>	63
5.5.1. Modélisation du muscle à contrôle local. Représentation du muscle à contrôle local.....	64
CHAPITRE VI	68
RESULTATS EXPERIMENTAUX.....	68
6.1. <i>Véhicule utilisant les muscles à contrôle central</i>	68
6.1.1. Identification des paramètres du têtard	68
6.1.2. Introduction d'une commande externe: La Vision. Modèle de boucle Visio - motrice	69
6.2. <i>Muscle à contrôle local ; Simulation et résultats</i>	79
6.2.1 Trajectoire Artificielle.....	79
6.2.2. Trajectoire réelle	82
CONCLUSIONS ET PERSPECTIVES	86
ANNEXE 1	89
A1.1 FONCTIONNEMENT DE L'ALGORITHME D'IDENTIFICATION.....	89
A1.2 PRESENTATION DETAILLEE DE L'ALGORITHME.....	96
<i>Identification évolutive des paramètres</i>	96
ANNEXE 2	99
COMPLEXITE DE CALCULS	99
BIBLIOGRAPHIE	102
ABSTRACT	107

CHAPITRE I

Introduction

Le réalisme du mouvement est un des problèmes le plus important dans la synthèse d'images animées. Le modèle de mouvement, qui est le cœur de la modélisation, peut s'appuyer sur plusieurs types d'approches permettant aux modélisateurs d'implémenter le mouvement ; ces approches peuvent être d'inspiration géométrique ou d'inspiration physique. Elles peuvent aussi permettre à l'utilisateur d'interagir avec le modèle en temps réel, par des capacités de communication qu'elles comportent, ou elles peuvent disposer d'une certaine autonomie de fonctionnement au cas où l'on ne s'intéresse qu'au résultat final de la simulation. Les canaux de communication entre le modèle et l'homme peuvent être de nature visuelle, auditive, gestuelle. Le but de chaque approche que le modèle incarne est de rendre le mouvement le plus réaliste possible. Pour cela une voie d'exploitation est l'utilisation des données expérimentales provenant des images réelles, et de les utiliser pour construire des modèles capables d'engendrer des comportements similaires.

A cette fin nous utiliserons les approches physiques, discrètes, basées sur des systèmes de particules. Ces approches se prêtent à modéliser une classe large de problèmes, parmi lesquelles les plus connus sont les processus physiques et les interactions homme-machine. Ces modèles sont souvent utilisés dans la synthèse d'images pour des simulations des objets articulés, déformables, par des systèmes particules - liaisons. Le but est de trouver une méthode par laquelle nous pourrions construire des modèles particules - liaisons à partir des séquences d'images.

Le problème de vision qui se pose est donc d'extraire un modèle physique à partir d'une cinématique, c'est-à-dire de trouver les bons paramètres de ce modèle. Ce modèle, une fois trouvé est utilisé ensuite à reconstituer, par synthèse d'images, des images proches des originales. En ce sens différentes difficultés surgissent, lorsqu'on essaie de modéliser des données réelles, le plus souvent liées aux données qu'on veut reproduire. Cela nous amènera à

construire des différentes primitives, parmi lesquelles la notion de « muscle » et d'injection d'énergie auront un lieu principal.

Dans le deuxième chapitre nous présenterons un tour d'horizon du domaine dans lequel nous travaillons, en commençant avec la description des modèles cinématiques et des modèles dynamiques, ensuite nous passons aux modèles du solide rigide et du solide déformable, terminant le chapitre par l'introduction du milieu de modélisation discret Cordis-Anima, basé sur le concept des particules – liaisons.

Dans le troisième chapitre nous présentons la manière dont nous pouvons identifier un modèle à partir d'une cinématique donnée, nous décrivons la technique évolutionnaire, nous présentons les particularités d'implémentation de cette technique au modèle physique, aussi que l'auto adaptation de paramètres de l'algorithme d'identification.

Dans le quatrième chapitre nous présentons des applications aux phénomènes physiques et les limites de cette méthode d'identification, limites dues aux primitives de modélisation utilisées auparavant.

Dans le cinquième chapitre nous décrivons la manière dont nous introduisons la motricité et le contrôle en Cordis-Anima, nous présenterons l'architecture du « muscle à contrôle central » et le concept de « muscle à contrôle local », avec leurs fonctionnements. Nous étudions aussi les conditions qui font la motricité possible, l'architecture et le fonctionnement des dispositifs à retour d'effort (TGR) qui apportent la notion de proprioceptivité aux muscles. Un problème fondamental discuté dans ce chapitre est le contrôle des systèmes physiques par couplage de liaisons, les dispositifs de couplage pour le muscle à contrôle central, les quadripôles et les transmetteurs, ensuite le couplage des muscles à contrôle local, dénommé couplage mutuel, réalisé à l'aide des unités de commande neuronales.

Dans le sixième chapitre nous présentons les résultats des avancées théoriques que nous avons proposés au chapitre antérieur, résultats centrés sur l'identification des paramètres des muscles et liaisons pour des données synthétiques et réelles, et aussi une application de contrôle par

système physique d'un mobile, dont nous profitons pour introduire le contrôle par boucle visio - motrice.

En dernier un chapitre de conclusion clôture ce travail.

CHAPITRE II

Les Modèles Physiques en Animation

2.1. Modèles cinématiques et modèles dynamiques (physiques)

Le rôle de la modélisation est déterminant dans le domaine de l'infographie. Elle est une démarche de représentation des connaissances sur une scène dans l'analyse d'images, sous une forme abstraite, mathématique : le modèle. La synthèse d'images est le processus inverse, elle utilise le modèle à partir duquel elle fabrique des images. Dans le domaine de l'animation le mouvement en tant que succession des images est le centre d'intérêt du modelleur. Le but de la modélisation en animation est de rendre le mouvement plus réaliste et aussi de fournir des outils faciles à manipuler.

Dans la modélisation des phénomènes physiques deux niveaux d'interprétation existent : le niveau *cinématique* et le niveau *cinétique*. Au niveau cinématique le mouvement des objets est décrit sans prendre en compte les causes qui l'engendrent; la description du mouvement est faite en termes de positions, vitesses et/ou accélérations, qui ne sont que les variables directement *observables* par un observateur externe au phénomène. Au niveau cinétique, le mouvement et ses paramètres cinématiques sont calculés en tant que conséquences directes de l'action des forces, couples et les valeurs de masses que les objets possèdent. Les deux niveaux de modélisation étudiés ensemble forment la *dynamique*, branche de la physique qui modélise les objets en mouvement. Avec ses considérations physiques, on distingue deux types de démarches possibles :

- Les modèles descriptifs : dans cette démarche, les modèles proposés essaient de reproduire exactement les effets cinématiques (positions, vitesse, trajectoires). Ces effets peuvent être parfois très difficiles à modéliser, ce qui rend cette approche très laborieuse et très dépendante de l'application.
- Les modèles générateurs : ces modèles traitent les causes qui produisent les effets en essayant d'établir des lois de comportement. La puissance de cette approche est qu'elle descend un niveau phénoménologique (en étudiant les causes qui produisent les effets) en

dessous de l'approche descriptive. Cet avantage est parfois éclipsé par des difficultés dans le contrôle des paramètres du modèle, qui doivent être ajustés d'une façon indirecte, en regardant toujours les effets produits. Une fois les paramètres statiques (masses, constantes physiques, etc.) et dynamiques (raideurs, amortissements, etc.) du modèle établis, ainsi que les conditions initiales (positions et vitesses), le mouvement apparaît comme une conséquence temporelle de ceux-ci. Nous pouvons donc affirmer que le mouvement est créé par l'évolution temporelle du modèle même.

Les modèles descriptifs sont très employés quand il s'agit de modéliser avec une précision élevée le mouvement mais ils deviennent rapidement inutilisables quand la complexité du modèle croît (en termes de nombre de paramètres). Les modèles générateurs résolvent ce problème des modèles descriptifs, mais leur précision à reproduire le mouvement initial est parfois moins élevée que celle des méthodes descriptives.

2.1.1. Modèles descriptifs ou cinématiques

Les modèles descriptifs sont souvent appelés cinématiques, selon les propriétés physiques qu'ils impliquent. Ces modèles se réfèrent uniquement aux propriétés cinématiques des objets, précisément les positions et les vitesses des points constituants.

Les modèles cinématiques sont les premières approches utilisées dans le domaine de modélisation en animation. Ces approches trouvent leurs origines dans le travail antérieur que les gens ont développé durant l'époque des créations des dessins animés. On compte deux classes de modèles cinématiques [Luc81-85] :

- Animation par manipulation : l'animation est fabriquée à partir d'un objet manipulé attentivement par l'animateur, ensuite il est filmé.
- Animation par composition : cette technique est en fait le dessin animé où le réalisateur dessine les formes successives que l'objet prend, après la séquence réalisée est enregistrée sur un film.

Cette dernière méthode a été parmi les premières utilisées dans l'animation par ordinateur, grâce à sa méthodologie qui remplace les techniques du dessin classique par celles de l'ordinateur. Le travail consiste à créer une animation à partir de quelques images - clefs que le réalisateur fournit. La tâche de l'ordinateur est de générer une séquence d'images par l'application des diverses techniques d'interpolation (linéaires, polynomiales, splines...) sur l'ensemble de degrés

de liberté de l'objet. Cette méthode réduit le travail de l'animateur en éliminant le processus difficile de l'intervalliste. Cependant elle souffre d'une manque de réalisme et la ressemblance avec le mouvement réel reste assez faible.

D'autres techniques plus complexes ont été développées afin de surmonter cet inconvénient, notamment les techniques utilisant les fonctions d'évolution temporelle. L'objet est composé d'une forme de base à laquelle s'ajoute un opérateur de transformation géométrique. Cette approche économise beaucoup de temps de travail processeur, sa rapidité étant significative par rapport aux anciennes techniques, on peut aussi remarquer que l'objet est décomposé de point de vue morphologique en parties qui peuvent, si nécessaire, être animées séparément. Les mieux connues parmi ces techniques sont les *déformations* à forme libre (FFD : Free-Form Deformations) et ses extensions EFFD (Extended Free-Form Deformations) ou pour le contrôle et l'animation des objet déformables AFFD (Animated Free-Form Deformations) [BD95].

D'autres méthodes plus sophistiquées ont été développées, parmi lesquelles un lieu très important revient au *morphing*. Cette méthode permet de construire une transformation d'un objet en autre par déformation continue d'un maillage initial vers un maillage final, en passant, le cas échéant, par des maillages intermédiaires. Cette technique s'apprête à des interventions interactives de l'animateur qui à tout moment peut changer l'emplacement et la densité des 'mailles' afin de favoriser les zones les plus soumises aux déformations par rapport à d'autres moins affectées.

Une des applications les plus souvent rencontrées en *morphing* est la déformation ou le mouvement des visages.

Les méthodes descriptives (cinématiques) sont encore aujourd'hui les méthodes les plus employées dans la synthèse d'images animées. Le progrès réalisé par les machines, combiné à la puissance des logiciels spécialisés (CAO, graphiques) permet aux utilisateurs d'obtenir des effets d'un réalisme élevé. Il faut mentionner aussi le progrès apporté par les techniques de rendu (anglais : « rendering ») qui calculent également les propriétés optiques des surfaces des objets représentés.

2.1.2. Modèles générateurs ou dynamiques

Par rapport aux modèles descriptifs qui décrivent l'objet en fonction de sa représentation visuelle aux yeux de l'observateur extérieur, les modèles générateurs supposent l'existence d'un comportement intrinsèque des objets, d'où les effets visuels, engendrés par l'interaction objet - milieu extérieur, émergent. Dans cette approche les objets sont souvent nommés *acteurs* à cause de leurs facultés de communiquer avec l'environnement et aussi à cause de leur autonomie d'action au sein du milieu extérieur.

Le rôle de l'animateur, dans ce cadre, est de créer des mondes artificiels et laisser les acteurs y évoluer en engendrant des cinématiques diverses. Le comportement des acteurs est basé sur un système de règles; un système de règles établit une correspondance entre l'état des paramètres du milieu dans lequel l'acteur se trouve et les actions que celui-ci entreprend sur ce milieu. L'autonomie de l'acteur se manifeste donc par le fait qu'une fois un système de règles établi, il suit ses propres règles sans aucune intervention de la part de l'animateur. L'animateur peut intervenir sur le fonctionnement du système seulement de manière indirecte, modifiant la table de règles auxquelles l'acteur doit obéir. Les résultats de cette intervention (sur le système de règles) sont évalués par les effets produits sur la cinématique complète des acteurs. Ces résultats servent comme une *mesure qualitative* du travail de l'animateur. On remarque *le feedback indirect*, parce que l'animateur agit sur les 'causes' du mouvement, notamment les systèmes des règles, et récoltant les 'effets' de ces actions, la cinématique.

Une autre catégorie de modèles est celle représentée par les modèles physiques. L'idée qui constitue la base de cette approche est que le mouvement de synthèse est d'autant plus réaliste s'il intègre les connaissances que l'animateur a sur son environnement. Ces connaissances transposées en mathématiques sont la mécanique analytique et la physique en général [Sch84] [Jah84,LC84].

Les modèles physiques peuvent être classifiés selon les différentes méthodes (mécaniques ou analytiques) qu'ils appliquent dans le domaine de l'informatique graphique. Ainsi, du point de vue de la représentation physique des objets simulés, on distingue des modèles du solide continu ou des modèles discrets nommés aussi particuliers. Du point de vue des propriétés des

matériaux modélisés, on distingue les modèles du solide rigide (non-déformable) et des modèles du solide déformable.

2.1.3. Modèles du solide rigide

Ces modèles font partie des premières approches parmi les modèles physiques étudiés. Leur but est de décrire la dynamique des systèmes dont la forme est fixée. Ceci dit, ils ne s'intéressent pas aux déformations, par conséquent aux matériaux, mais seulement à la dynamique des composants et de l'ensemble. Les composants sont supposés, donc, parfaitement rigides. Le fondement de cette approche est la mécanique *newtonienne* et ses trois principes : principe d'inertie, principe fondamental, et principe d'action – réaction. La loi fondamentale de la dynamique s'applique aux centres de gravité des composants de l'objet, composants qui sont convenablement choisis en fonction de la topologie de l'objet et de la répartition spatiale de ces degrés de libertés.

Avec ces considérations introductives, d'ordre qualitatif, sur les idées qui ont généré l'étude de ces modèles, la voie est maintenant ouverte au formalisme mathématique.

La loi qui décrit les translations s'exprime de la manière suivante :

$$\sum F_{ext} = \frac{d\bar{P}}{dt} = M * \ddot{\bar{r}} \quad (2.1)$$

où

$\ddot{\bar{r}}$ représente l'accélération du centre de gravité du composant;

M est la masse totale du composant;

\bar{P} est l'impulsion ou la quantité de mouvement du centre de gravité du composant.

A son tour le mouvement de rotation est décrit par l'équation :

$$\frac{d\bar{L}}{dt} = \sum \bar{N} \quad (2.2)$$

avec

$\vec{L} = I * \vec{\omega}$ le moment cinétique du composant ;

$\vec{N} = \vec{r} \times \vec{F}_{ext}$ le moment de la force appliquée au composant ;

les deux grandeurs physiques calculées par rapport au même repère, et où :

I est le tenseur du moment d'inertie ;

$\vec{\omega}$ est la vitesse angulaire du composant définie par rapport au même repère.

Le tenseur moment d'inertie a comme représentation analytique une forme bilinéaire qui peut être représentée par une matrice. La forme de ce tenseur s'exprime par rapport à un système de référence, qui convenablement choisi peut simplifier la formulation des équations (vectorielles) du mouvement.

$$\sum \vec{N} = I * \frac{d\vec{\omega}}{dt} + \vec{\omega} \times (I \wedge \vec{\omega}) \quad (2.3)$$

En utilisant les symétries de l'objet nous pouvons toujours arriver à diagonaliser la forme bilinéaire tensorielle ce qui nous permet de découpler l'équation vectorielle en trois équations scalaires qui sont les équations d'Euler [LL69] :

$$I_x * \frac{d\omega_x}{dt} + (I_z - I_y) * \omega_z \omega_y = N_x \quad (2.4)$$

$$I_y * \frac{d\omega_y}{dt} + (I_x - I_z) * \omega_x \omega_z = N_y \quad (2.5)$$

$$I_z * \frac{d\omega_z}{dt} + (I_y - I_x) * \omega_y \omega_x = N_z \quad (2.6)$$

Les variables I_x, I_y, I_z , sont les moments principaux d'inertie de l'objet. Donc le problème se résume à la résolution d'un système d'équations différentielles du premier ordre. De nombreuses variantes de résolution discrète des systèmes de ce genre ont été employées, comme Runge-Kutta ou Gauss-Seidel.

Une autre approche souvent utilisée dans la modélisation du solide rigide s'appuie sur la mécanique *lagrangienne*. Ces équations dérivent du principe de *d'Alembert* et elles se réfèrent au travail mécanique et à l'énergie. Dans le cadre de la mécanique lagrangienne *l'état de l'objet* est représenté sous forme vectorielle dans l'espace des phases. Par phase on comprend

l'ensemble des *coordonnées généralisées* correspondant aux degrés de libertés du système et leurs dérivées par rapport au temps, dénommées *vitesses généralisées*. Soit l le nombre de degrés de libertés que le système comporte, la phase du système à un moment t est $(q_1, \dots, q_l, \dot{q}_1, \dots, \dot{q}_l)$.

Les équations de Lagrange sont :

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} = 0, \quad j = 1, \dots, l \quad (2.7)$$

où la fonction L est définie par:

$$L = T - U = \sum_{j=1,l} \frac{1}{2} m_j \dot{q}_j^2 - \sum_{j=1,l} \left(- \frac{\partial U}{\partial q_j} \right) = L(q_1, \dots, q_l, \dot{q}_1, \dots, \dot{q}_l) \quad (2.8)$$

et s'appelle le *Lagrangien* du système. Comme nous le constatons, elle est la différence entre l'énergie cinétique et l'énergie potentielle du système. A l'INRIA (Rennes) un système d'animation basé sur l'approche lagrangienne a été conçu afin de simuler le solide rigide [Arn88].

Une autre approche qui vient en tant qu'extension du solide rigide est constituée par les modèles des objets articulés. Cette approche introduit la notion de *liaison*, dont la fonction est de réduire la capacité de mouvement des composants de l'objet. Du point de vue physique elle limite le nombre de degrés de liberté de l'objet. Dans le cas le plus simple ces liaisons peuvent être holonomes, quoique cela limite parfois le réalisme de la modélisation, mais cela permet de réduire le nombre de paramètres et des équations à résoudre en éliminant les degrés de liberté correspondant aux liaisons.

Différents travaux ont été menés dans le cadre du solide rigide articulé, on peut citer ceux de Wilhelms & Barsky [WB95], Armstrong & Green [AG85], McKenna & Zeltzer [MZ90].

Le contrôle de l'animateur dans ces approches présentées ci-dessus s'effectue de manière directe. Il s'agit d'appliquer de définir des forces externes ou internes et des couples qui sont appliquées au système. L'insuffisance de ces modèles est qu'on ne peut jamais garantir que ces

forces ou couples conduisent le système à rendre la cinématique désirée par l'animateur. De plus, un utilisateur non –avisé ne peut pas manipuler avec une bonne précision de tels modèles.

Les dernières approches dans le domaine du solide rigide que nous présentons ici sont celles basées sur les méthodes de dynamique inverse [BB88]. Dans une inspiration robotique, il s'agit de calculer les forces et les couples externes alors qu'on connaît les positions et les vitesses internes de l'objet aussi que les contraintes existantes. Une faiblesse qui surgit immédiatement de cette technique est qu'elle devient vite difficile et lourde quand la complexité des objets grandit.

2.1.4. Modèles de solide déformable

Une deuxième classe de modèles étudiés en animation est celle qui s'inspire directement de la mécanique du solide déformable. Dans cette approche, la modélisation devient plus complexe que dans l'approche du solide rigide. Pourtant, la grande difficulté qui demeure est le coût élevé des calculs qu'un tel modèle engendre.

Les matériaux simulés sont censés avoir des caractéristiques visco-élastiques, continues et réparties en trois dimensions ou sur une variété bidimensionnelle, dans un espace 3D. Cette formulation mathématique est générale quel que soit l'objet modélisé. Qu'il s'agisse de modéliser un gaz ou un solide ou un objet liquide, le même formalisme mathématique est appliqué. Ce formalisme s'appuie sur les équations décrivant les forces nécessaires à appliquer à l'objet pour déformer sa surface ou modifier son volume. La théorie de l'élasticité nous donne le moyen par lequel ces équations peuvent être réduites à un système d'équations différentielles, dans le cas le plus général non-linéaires.

Les travaux de Terzopoulos, Platt et Fleischer [TPBF867, TF88] ont posé les bases de la modélisation de l'élasticité tridimensionnelle avec les avantages et les inconvénients qui en découlent. Leur démarche consiste dans la discrétisation de la matière qui compose l'objet en établissant un maillage géométrique de la forme de l'objet ou une triangulation du domaine, qui se traduit ensuite par une résolution par éléments finis.

2.2. Cordis-Anima

Les modèles à base de particules ont été utilisés à partir des années soixante dix dans les domaines de la simulation et de la synthèse d'images dans le but de simuler une large variété de phénomènes physiques comme par exemple les explosions, les flammes, les fumées. Leur champ d'applications s'est étendu vers des domaines comme l'analyse du mouvement, la modélisation du mouvement et l'animation par l'ordinateur. Du point de vue physique ils se classifient en :

- modèles cinématiques qui traitent le mouvement au niveau de sa trajectoire qui est complètement définie par l'ensemble des positions et des vitesses ;
- modèles dynamiques, qui traitent le mouvement au niveau des causes qui l'ont produit.

Ces modèles tiennent compte des forces, potentiels d'interaction et des autre grandeurs scalaires (masses par exemple) qui affectent le mouvement.

Dans la modélisation physique, nous disposons d'un grand nombre de variables reliées par les équations qui décrivent les lois physiques. Ces variables sont donc interdépendantes unes des autres, ce qui nous oblige à faire une réflexion sur le choix qu'on a à faire. Les phénomènes physiques s'expriment par deux grandes familles de variables : intensives et extensives. Les variables extensives sont les variables cinématiques qui décrivent le phénomène et qui lui appartiennent. Les variables intensives sont des variables dynamiques (forces, couples, etc.) et comme leur nom suggère, elles décrivent les causes du phénomène. Ces deux classes des variables sont complémentaires ou duales. Chaque variable appartenant à une des classes possède sa variable conjuguée dans l'autre classe. De nombreux exemples provenant des différentes branches de la physique peuvent être cités :

- En mécanique : la force qui est une variable intensive a pour variable extensive conjuguée la position,
- En thermodynamique : la pression a pour variable extensive conjuguée le volume,
- En électricité : nous trouvons l'intensité électrique en tant que variable intensive conjuguée avec la tension électrique en tant que variable extensive.

Dans ce chapitre, nous présentons le formalisme de modélisation physique Cordis-Anima. Ce formalisme a été développé par ACROE et il s'agit d'un ensemble de primitives qui permettent de construire une large variété de systèmes physiques. Il existe deux primitives principales sur

lesquelles s'appuie ce formalisme, dénommées atomes M et L. Tous les autres objets construits par Cordis-Anima sont des structures utilisant ces atomes.

Ces atomes peuvent être décrits comme des « boîtes noires » qui comportent une entrée et une sortie. Le point M reçoit comme grandeur d'entrée une force et la convertit en position à la sortie. Le point L réalise la transformation complémentaire recevant une position en entrée et fournissant une force en sortie.

Le point M assure le passage d'une variable intensive (force) à une variable extensive (position) alors que le point L assure le passage d'une variable extensive (position) à la variable intensive conjuguée (force).

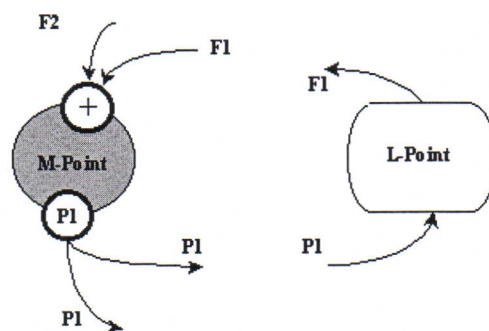


Figure 2.1.
Les primitives Cordis-Anima

Maintenant, nous allons traiter le problème d'interconnexion entre plusieurs points M et L afin de pouvoir construire des modèles physiques d'une complexité élevée. Nous nous apercevons que ce problème d'interconnexion n'est pas simple puisque ces primitives ne nous permettent pas de réaliser toutes les topologies envisageables. Dans le cas le plus simple d'une connexion entre deux points M, il nous faut deux points L pour arriver à faire communiquer les deux points M alors qu'un seul n'était pas suffisant.

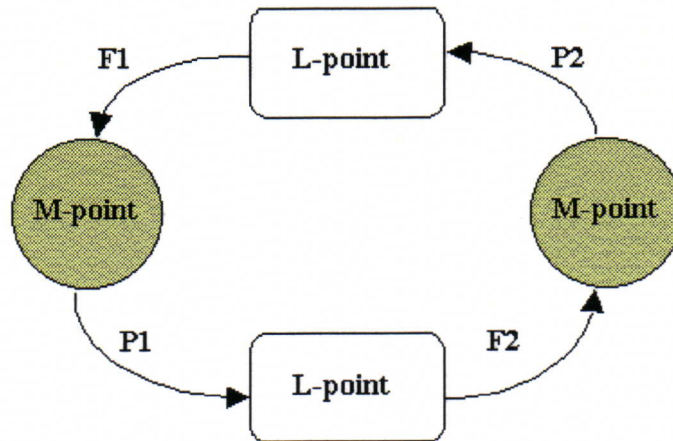


Figure 2.2.
Interconnexion de deux M-points

La modélisation physique nous oblige à nous assurer que le principe d'action - réaction reste valable dans le cas d'un tel réseau alors que mathématiquement ça s'exprime par :

$$\vec{F}_1 = -\vec{F}_2 \quad (2.9)$$

Avec cette condition et pour résoudre le problème de la connexion, nous définissons l'atome, la « liaison <LIA> » étant un module qui est composé de deux points L indissociables. Nous allons dorénavant appeler le point M par le syntagme « atome <MAT> » et nous utiliserons les atomes <LIA> en tant que primitives modélisant la transformation d'une position en force, à place des points L. L'atome <MAT> correspond dans la modélisation physique à une particule tandis que le atome <LIA> correspond à une interaction bilatérale entre deux atomes <MAT>, comme s'est le cas de l'interaction élastique. On remarque que cette évolution respecte le principe d'action - réaction qui demeure fondamental dans le domaine de la modélisation physique. Les atomes <MAT> et <LIA> sont des composants complémentaires pas seulement de point de vue mathématique mais aussi du point de vue phénoménologique. L'atome <MAT> modélise le comportement d'une particule, il caractérise la dynamique du phénomène en tenant compte de ses propres variables tandis que l'atome <LIA> caractérise les corrélations des deux phénomènes, appelées en termes physiques l'interaction. Cette interaction reste en Cordis-Anima une interaction bilatérale, mais comme nous allons montrer plus tard dans un autre chapitre, nous pouvons utiliser des interactions d'ordre supérieur, notamment dans les modèles à base de potentiels.

Sous forme physique, ces considérations phénoménologiques s'expriment de la manière suivante :

$$F_{\text{totale}} + F_{\text{interaction}} = 0 \quad (2.10)$$

où

$$F_{\text{totale}} = m \frac{d^2 x}{dt^2} \quad (2.11)$$

$$F_{\text{interaction}} = F_{\text{élastique}} + F_{\text{viscosité}} \quad (2.12)$$

Ces deux types d'interactions ont été choisies grâce à leur représentativité pour deux grandes catégories d'interactions physiques. La première, l'interaction élastique, est une interaction qui conserve l'énergie et qui peut modéliser les champs conservatifs et la deuxième interaction modélise les interaction avec perte d'énergie, qui correspondent à des champs dissipatifs.

Nous modélisons le comportement des différents objets physiques dans ce cadre.

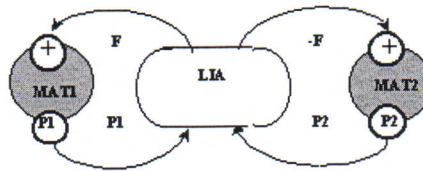


Figure 2.3.
Modules <MAT> and <LIA>

Analysons maintenant ce formalisme de modélisation du point de vue des systèmes dynamiques. La fonction de transfert assurée par l'atome <MAT> est $\sum \vec{F} = m\ddot{x}$ qui est ni plus ni moins que le deuxième principe de la dynamique. Le paramètre m représente la masse inertielle de l'atome <MAT> ou de la particule.

Un atome <LIA> a une fonction de transfert plus complexe, exprimée sous la forme suivante :

$$\left(\vec{F}, -\vec{F} \right) = K(x_1 - x_2) + Z \left(\frac{dx_1}{dt} - \frac{dx_2}{dt} \right) \quad (2.13)$$

Analysons maintenant l'expression ci-dessus. Le premier terme correspond à une dépendance linéaire de la force calculée par la distance relative de deux atomes <MAT> tandis que le

deuxième terme correspond à une dépendance de la force par la vitesse relative des deux atomes <MAT>. Le paramètre K est l'analogie de la constante élastique tandis que le paramètre Z est l'analogie de la viscosité. Nous remarquons facilement que si nous remplaçons la force dans le membre gauche de l'équation, nous obtenons une équation différentielle d'ordre deux. Cette équation nous fait penser aux systèmes dynamiques, qui souvent s'expriment sous une forme analogue.

$$m \frac{d^2 x}{dt^2} - K(x_1 - x_2) - Z \left(\frac{dx_1}{dt} - \frac{dx_2}{dt} \right) = 0 \quad (2.14)$$

Le formalisme Cordis – Anima nous permet de modéliser tous les systèmes dynamiques de deuxième ordre d'une manière connexionniste. Au lieu de résoudre l'équation différentielle qui modélise un système d'ordre deux, nous décomposons son comportement complexe dans un réseau de primitives comportementales simples.

Pour montrer que le formalisme Cordis – Anima est suffisamment riche mathématiquement pour modéliser tout système physique, nous allons maintenant démontrer que n'importe quel phénomène physique peut être décrit par une équation différentielle d'ordre deux [L01b].

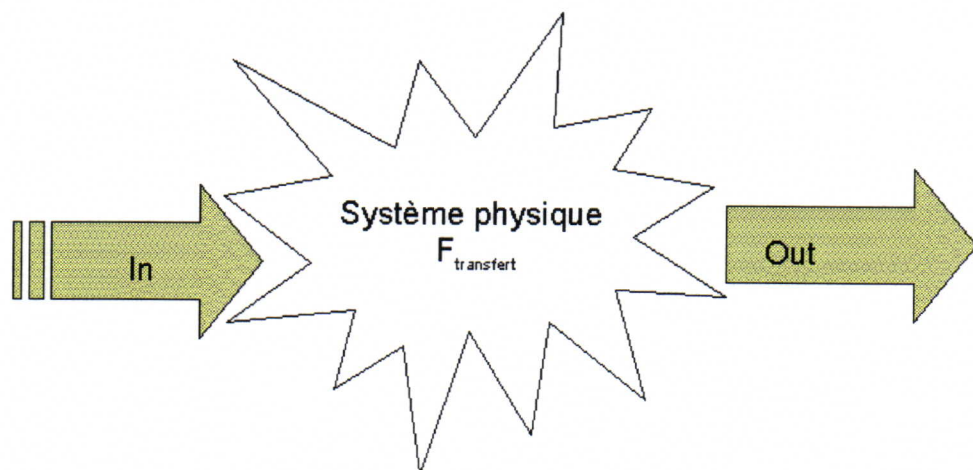


Figure 2.4
Système physique vu en tant que système dynamique

La fonction de transfert peut être écrite sous la forme suivante :

$$OUT(t) = F_{\text{transfert}} [IN(t)] \quad (2.15)$$

Dans le cas le plus général la sortie est un vecteur de paramètres qui dépendent de l'entrée. L'entrée elle-même peut être un vecteur mais pour la simplicité du raisonnement on va considérer une entrée scalaire (faisons l'hypothèse que les autres paramètres physiques en entrée sont figés) et l'on va se référer à une seule composante du vecteur sortie, la généralisation étant ensuite triviale.

Appliquant la transformée de Laplace des deux côtés de l'équation, nous obtenons la relation suivante :

$$O(s) = \frac{P^m(s)}{Q^n(s)} I(s) \quad (2.16)$$

La transformée de Laplace de la fonction de transfert est un rapport de deux polynômes aux coefficients constants de degrés m et n .

Supposons que nous sommes dans le cas où $m > n$. Par une décomposition en fractions rationnelles à coefficients réels on obtient :

$$\frac{P^m(s)}{Q^n(s)} = R_0^{m-n}(s) + \sum_{i,j} \frac{R_i^1(s)}{R_j^2(s)} + \sum_{k,l} \frac{R_k^0(s)}{R_l^1} \quad (2.17)$$

Dans cette relation les $R_j^{\text{degré}}$ sont des polynômes, l'indice supérieur indique le degré du polynôme tandis que l'indice inférieur est une indexation. Nous remarquons qu'après la décomposition nous avons trois termes :

- le premier est un polynôme de degré $(m - n)$;
- le deuxième est une somme de fractions qui ont pour numérateurs des polynômes de degré 1 et pour dénominateurs des polynômes de degré 2 ;
- le troisième terme est une somme de rapports des constantes aux numérateurs et de polynômes de degré un aux dénominateurs.

Maintenant, revenons sur les conditions physiques. Une hypothèse qu'on peut toujours considérer comme vraie est l'hypothèse du système physique réel. Cette hypothèse suppose qu'un système réel ne peut jamais donner une réponse infinie en sortie, indifféremment des valeurs des signaux en entrée et de ses paramètres internes. Cette considération est de nature physique et elle est basée sur l'observation des phénomènes naturels.

Pour exprimer cette hypothèse sous une forme de condition mathématique, nous analysons en détail l'expression antérieure ; si les polynômes $R_0^{m-n}(s)$ sont non-nuls, nous risquons d'obtenir une réponse infinie au cas limite où s va vers l'infini. Or, dans cette situation, l'hypothèse initiale est contredite.

Nous devons donc conclure que la seule condition mathématique qui nous reste est d'imposer aux polynômes $R_0^{m-n}(s)$ d'être nuls partout. Cela est équivalent à dire que le cas $m > n$ n'apparaît jamais en réalité. Le seul cas pouvant réellement se produire est donc $m < n$. Avec ces considérations, le problème est donc restreint à la résolution des polynômes d'ordre 2, qui reste l'ordre le plus élevé dans l'expression.

$$R_j^2(s) = 0 \quad (2.18)$$

En passant dans le domaine temps, ceci est équivalent à une équation différentielle d'ordre 2:

$$m \frac{d^2 x}{dt^2} + z \frac{dx}{dt} + kx = 0 \quad (2.19)$$

L'atome <MAT> contient l'opérateur différentiel d'ordre deux $\frac{d^2}{dt^2}$ et l'atome <LIA> contient

les opérateurs $\frac{d}{dt}, \frac{d^0}{dt^0}$.

Donc ces trois opérateurs différentiels dont nous avons besoin pour modéliser les phénomènes physiques sont présents en Cordis-Anima par l'intermédiaire des atomes <MAT> et <LIA>.

Un exemple de réseaux Cordis-Anima formé par trois particules est représenté dans la figure suivante.

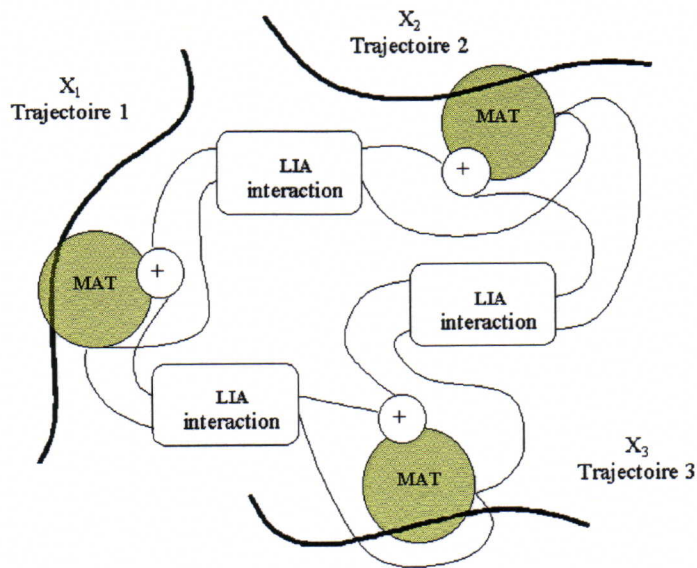


Figure 2.5
Exemple de réseau Cordis-Anima

Dans ce cas, nous avons fait l'hypothèse que le système de particules est isolé par rapport au monde extérieur. Les interactions internes seules sont présentes. Nous observons bien que l'atome <MAT> calcule uniquement des paramètres liés à sa trajectoire, c'est-à-dire des paramètres propres à son comportement, tandis que les <LIA> remplissent le rôle d'interaction entre les différents comportements individuels.

2.3 Animation de créatures par modèle physique

Dans le domaine de l'animation par modèle physique les travaux de van de Panne et ses collaborateurs s'inscrivent parmi les premiers [PKF94]. Leur travail se concentre sur la proposition d'une méthode d'animation des objets articulés. Ils ont utilisé pour la commande de ces articulations des réseaux cycliques de poses. Une pose indique la configuration interne qu'on désire. Elle signifie donc la forme de la créature que nous essayons d'animer. Les graphes de contrôle par poses obligent les contrôleurs à produire du mouvement périodique. Une des avancées importante de leurs travaux est l'interaction permanente de l'objet avec le sol, plus généralement parlant l'environnement qui l'entoure, contact qui mène à une dynamique stable

du mouvement, d'où le nom suggestif de ces objets 'wind-up toys'. Les contrôleurs sont créés aléatoirement et après ils sont adaptés en fonction du comportement escompté.

Dans la même logique de modélisation nous retrouvons les travaux de Terzopoulos et son équipe [TT] sur la construction des poissons artificiels, basé sur du modèle physique. Dans ce cas nous avons à faire avec des systèmes masses-ressorts qui constituent le squelette des animats. Certains ressorts ont été modifiés afin de pouvoir jouer le rôle des muscles producteurs de motricité. La motricité est produite par modification instantanée de la longueur de repos des ressorts – muscles, ce qui équivaut à un apport d'énergie potentielle. La manière dont cela s'exécute est gérée par des contrôleurs. Les schémas de contrôle sont complexes et ils comprennent aussi des caractéristiques comportementales spécifiques (prédateurs, accouplement, évation, pacifiste, etc.). L'interaction avec l'environnement se fait par l'action des résultantes de la pression de l'eau (milieu extérieur) sur les masses qui constituent le squelette. L'enjeu de cette interaction très importante est la capacité du poisson de se déplacer dans son milieu.

Karl Sims [S94] a fait appel aux méthodes d'évolution artificielle dérivées des algorithmes génétiques pour la construction des créatures virtuelles ayant de comportements intéressants. Précisément il a décidé d'évoluer la structure des créatures ainsi que leurs contrôleurs, constitués des réseaux de neurones. Les comportements –cible visés comme la nage, la marche, le saut se constituent en critères d'évaluation pour les techniques d'optimisation génétiques. Intéressant à remarquer dans ces travaux l'émergence des divers comportements qui seraient difficile à construire par des méthodes classiques, qui ne sont pas basées sur l'évolution artificielle. Chaque neurone de commande dans le réseau qui constitue le contrôleur gère un degré de liberté d'une articulation de la créature. Les neurones peuvent implémenter des différentes fonctions mathématiques soumises elles aussi à l'évolution.

Enfin les travaux de Komosinski et Ulatowski [KU99] sur le design d'un système de simulation des créatures artificielles dans des conditions similaires avec celle qu'on trouve dans le milieu naturel. Ces créatures possèdent un contrôleur formé par un réseau de neurones et leur squelette est formé par des articulations et bras rigides. Le réseau des neurones est évolué dans le même temps que la structure mécanique de la créature. Les critères d'évolution des créatures peuvent être plusieurs, comme par exemple de maximiser la vitesse moyenne horizontale par terre, de maximiser la vitesse moyenne horizontale en eau, de maximiser l'énergie potentielle, etc. Le

simulateur Framsticks offre la possibilité d'implémenter plusieurs types de forces, comme la gravitation, la friction, l'amortissement, l'élasticité dans les articulations, la réaction du sol sur l'objet. Les neurones qui constituent le réseau de commande possèdent trois paramètres dans le génotype :

- Force : désigne la vitesse de changement vers la valeur calculée en entrée
- Inertie : désigne la capacité du neurone de garder plus long temps la tendance de changement de son état interne
- Sigmoides : coefficient qui désigne la forme de la fonction de transfert du neurone

Les neurones ont comme entré plusieurs paramètres qui sont donnés par un système de récepteurs (le toucher, l'équilibre et la localisation d'énergie).

Un neurone qui joue le rôle d'un muscle peut changer l'orientation relative d'un bras d'une articulation par rapport à l'autre.

Plusieurs langages de description génétique sont utilisés afin de pouvoir spécifier les architectures et les comportements des créatures à évoluer.

Ce simulateur implémente aussi des différentes voies de survie des créatures, passive ou agressives, il offre aussi la possibilité de spécialisation des membres d'un organisme ou de spécialisation des individus au sein d'une population.

La simulation de l'évolution comporte une évaluation multicritère, un codage d'un individu ou d'une population sur un génotype, des règles de changement de descriptions des créatures par mutation et croisement, durée de vie d'individus, simulation d'évolution spontanée (sans critère) utilisant comme critère la durée de vie.

CHAPITRE III

Identifier un modèle à partir de la cinématique : la technique évolutionnaire

Dans le cadre de la mathématique et de l'ingénierie, on étudie depuis des décennies les techniques d'optimisation et d'identification de paramètres, ainsi que les problèmes inverses où à partir de données, on essaie de reproduire le système qui les a produites. Nous avons constaté que les méthodes d'optimisation classiques s'appuient sur des propriétés de régularité des fonctions à optimiser. Malheureusement, dans la plupart de cas réels, on sait peu de choses sur les propriétés de ces fonctions, ce qui rend souvent inutilisables les méthodes classiques. Ceci dit, des méthodes moins «orthodoxes» ont été développées ces dernières années, basées sur des techniques nouvelles caractéristiques de l'intelligence artificielle, avec souvent une inspiration biologique.

Nicolas Szilas [S95], dans son travail de thèse, a essayé de résoudre le problème de l'identification des paramètres de systèmes Cordis-Anima par l'intermédiaire de réseaux de neurones. Il a modélisé les réseaux masses-ressorts par analogie avec les réseaux de neurones récurrents. Les algorithmes d'entraînement des poids des connexions sont devenus, par analogie, des modalités de trouver les valeurs des paramètres mécaniques du réseau masses-ressorts. Le succès de cette méthode s'est pourtant limité à l'optimisation de paramètres des systèmes masses-ressorts simples : dès que la taille du système augmente, appliquer cette méthode par construction du réseau neuronal équivalent devient de plus en plus difficile. Une autre faiblesse de cette méthode est le fait que les réseaux de neurones sont bien adaptés quand il s'agit de mémoriser des formes ou dans la reconnaissance des formes, mais le sont moins dans les tâches d'optimisation, dont notre problème d'identification est un cas particulier. D'autres techniques nous sont apparues, donc, plus appropriées à résoudre cette tâche.

Une première technique qui peut être envisagée est le *recuit simulé*. Cet algorithme s'appuie sur le parcours de l'espace des paramètres d'une manière aléatoire afin de minimiser la fonction de coût qui mesure la performance de chaque jeu de paramètres. L'origine de cet algorithme se

trouve dans la mécanique statistique classique, la fonction de coût étant assimilée à l'énergie potentielle du système. La configuration d'équilibre en physique correspond à l'état d'énergie potentielle minimale, but de la recherche aléatoire. Par analogie, un paramètre équivalent à la température est nécessaire pour régler la *capacité d'exploration* de l'algorithme. Dans les systèmes physiques la température peut être considérée comme une mesure du désordre combinée avec l'énergie cinétique des constituants. Dans le cas du recuit simulé, le paramètre T contrôle le caractère aléatoire de la recherche ainsi que la capacité du système à *remonter les pentes* (l'équivalent de l'énergie cinétique) de la fonction coût et de sortir des minima locaux. L'algorithme fonctionne de la manière suivante : à chaque pas de temps, le nouveau vecteur état x_{n+1} est calculé à partir de l'ancien vecteur x_n par l'intermédiaire d'un bruit gaussien de variance donnée. Le nouveau vecteur x_{n+1} est retenu si la valeur de la fonction de coût est $f(x_{n+1}) < f(x_n)$; sinon la probabilité de transition de l'état x_n à l'état supérieur x_{n+1} est calculée selon la formule $e^{\left(\frac{f(x_{n+1})-f(x_n)}{T}\right)}$. Cette règle permet l'exploration de l'espace du problème à optimiser vers des valeurs plus élevées de la fonction de coût, permettant au système de s'échapper des minima locaux. Le paramètre qui pilote cette exploration T est l'équivalent de la température. Le contrôle de la convergence de l'algorithme se fait par la décroissance lente (exponentielle) de la température (de manière analogue au procédé de recuit des métaux, en métallurgie).

Les limitations que cette méthode comporte sont reliées d'abord à la vitesse de convergence. La décroissance exponentielle est un processus très lent. Le compromis souvent fait de s'éloigner de ce schéma au profit d'une plus grande vitesse, risque de jeter le système dans un minimum local. Une autre limitation de cette méthode est l'impossibilité de changer la fonction de coût pendant que l'algorithme tourne ; chaque fois que l'on change la fonction de coût, même partiellement, le recuit simulé doit être recommencé à zéro.

Une autre limitation est reliée aux objets complexes ou aux objets dans lesquels les paramètres prennent des valeurs extrêmes. Avec la taille de l'objet la proportion des liaisons mal identifiées croît rapidement, rendant l'algorithme inefficace.

3.1. Introduction aux méthodes évolutionnaires

Les stratégies évolutionnaires sont des méthodes stochastiques d'optimisation de même inspiration que les algorithmes génétiques [G89]. Les deux classes d'algorithmes sont d'inspiration biologique, et tentent de reproduire mathématiquement les mécanismes d'évolution des espèces, tels qu'ils sont décrits par les théories darwiniennes. Tandis que les algorithmes génétiques travaillent sur des individus ayant des valeurs booléennes, les stratégies évolutionnaires utilisent des valeurs réelles, ce qui les rendent plus appropriées pour la modélisation physique.

Comme nous l'avons vu ci-dessus, les autres techniques d'optimisation non-classiques présentent différents désavantages reliés, dans le cas des réseaux neuronaux, aux limitations du nombre des configurations de systèmes physiques que nous pouvons optimiser, et dans le deuxième cas (celui du recuit simulé) à des limitations importantes reliées à la vitesse de convergence. Par rapport aux techniques mentionnées au-dessus, les stratégies évolutionnaires comportent quelques qualités essentielles parmi lesquelles :

- la vitesse de convergence par rapport aux autres techniques heuristiques (dans la plupart des cas réels, elles convergent plus vite que le recuit simulé);
- la scalabilité : les stratégies évolutionnaires se dimensionnent facilement en fonction de la taille du problème à optimiser, (contrairement à ce qui se passe dans le cas des réseaux des neurones).

3.2. Stratégie d'évolution: un paramétrage adapté aux grandeurs physiques

Le fonctionnement d'une stratégie évolutionnaire s'appuie sur quelques notions simples dérivées, comme nous l'avons déjà souligné, des mécanismes darwiniens. D'abord nous rencontrons les notions *d'individu* et de *population*. Le premier est en fait un vecteur aux composants réels (par différence avec les algorithmes génétiques où les composants sont binaires) qui codifie une solution candidate pour notre problème. Une collection d'individus (donc des solutions candidates) constitue une population. Dans le cas particulier auquel nous nous référons, celui des systèmes particules – liaisons, un individu représente un modèle dont les paramètres sont codifiés sous une forme vectorielle : le contenu de gènes. Cette population

d'individus est donc une population de candidats modèles qui subissent l'action de divers opérateurs mathématiques afin de choisir et d'affiner l'individu (modèle) le plus capable d'engendrer un comportement proche de celui escompté.

Maintenant, nous allons décrire la manière dont une stratégie évolutionnaire se construit, avec ses étapes ainsi que les pas à suivre quand nous définissons un tel algorithme. La définition d'une stratégie évolutionnaire comporte donc [Lu93]:

- Un problème à optimiser exprimé sous la forme mathématique d'une fonction à optimiser.
- Un codage des solutions possible du problème sous une forme vectorielle (chromosome).
- Des opérateurs génétiques qui agissent sur les individus et sur la population dans son ensemble, parmi lesquels on trouve le croisement, la mutation, la sélection, etc.
- Divers paramètres de contrôle de l'algorithme comme par exemple la taille de la population, les mutabilités, le nombre de générations...

Dans la situation particulière des systèmes particules –liaisons, un individu est un modèle entier contenant des particules reliées par des liaisons de divers types ayant différents paramètres. Ces modèles subissent donc l'action des mutations et des croisements, les plus performants étant favorisés par rapport aux plus faibles, selon les règles de l'évolution naturelle. Le but est d'obtenir un modèle qui soit capable d'engendrer le comportement le plus proche de celui envisagé, c'est à dire d'optimiser la fonction de coût, dont la valeur nous donne une mesure de cette ressemblance.

La principale raison de choisir une telle stratégie évolutionnaire a été la propriété de continuité des espaces physiques dans lesquels nous devons développer les modèles physiques. Un modèle physique est codifié dans un individu sous la forme d'une chaîne de valeurs réelles (ses paramètres de définition) appelée *génotype*. Le génotype donc est le codage d'un modèle. Le pendant d'un individu dans l'espace des modèles s'appelle *phénotype*. Dans ce cadre de la modélisation physique, il vaut mieux percevoir les stratégies évolutionnaires tantôt comme outil d'amélioration de solutions des problèmes, tantôt comme outils d'optimisation. Cette précision est importante, parce que notre effort d'automatiser le processus de construction de modèles à l'aide des stratégies évolutionnaires dépend de certains choix qu'on est obligé de faire, malgré

notre désir d'aller vers des procédures à 100% automatiques. Comme nous le verrons plus tard, les stratégies évolutionnaires résolvent une bonne partie du travail, auparavant fait « au jugé », mais nous aurons encore des décisions à prendre, comme par exemple le nombre de particules, leur poids total, etc. Dans ces circonstances, où les conditions initiales sont choisies par des raisons externes à l'algorithme, le mot « amélioration » d'un système particules – liaisons paraît plus approprié pour décrire l'action réelle de la stratégie évolutionnaire.

Par rapport aux autres algorithmes, les stratégies évolutionnaires sont munies d'un opérateur nommé *croisement* dont le rôle est de combiner des individus bien cotés par la fonction d'évaluation.

A chaque passage d'une population à une autre, les individus sont évalués et ordonnés à l'intérieur de la population selon les valeurs que la fonction de coût leur donne.

Ceci dit nous allons décrire maintenant les principaux pas dans l'implémentation de notre stratégie évolutionnaire :

Initialisation

C'est la première étape de chaque stratégie évolutionnaire. Un bon début d'algorithme nécessite une bonne répartition des valeurs des paramètres sur l'espace de recherche. Donc nous choisissons une initialisation aléatoire des paramètres des ressorts et amortisseurs. Les longueurs au repos des ressorts seront initialisées autour des valeurs moyennes des longueurs observées. Prenant comme exemple une roue de bicyclette. Pendant que le vélo se déplace la roue, à part du mouvement de translation et rotation qu'elle engendre, subit un mouvement oscillatoire de déformation, à cause des poids qu'elle supporte. Pour identifier les paramètres élastiques et visqueux de la roue, une bonne idée sera donc, d'initialiser les valeurs des rayons autour de moyennes au repos.

Sélection

La sélection est réalisée par un tirage aléatoire d'individus avec des probabilités qui sont proportionnelles aux *rangs* des individus et non aux valeurs de leurs fonctions de coût. Donc les individus qui ont la plus grande probabilité d'être éliminés sont ceux avec un petit rang dans la population. Pour s'assurer que les très bonnes solutions sont conservées, nous gardons les 10%

de meilleurs individus, ce qui fait que le coût du meilleur individu n'augmente jamais d'une génération à la suivante, ce qu'on appelle « élitisme ». Le nombre d'individus est constant d'une population à l'autre, ce qui assure un effectif constant. Le taux de remplacement est d'à peu près 60% d'individus changeant d'une population à la suivante.

Croisement

Le croisement est l'opérateur qui permet à l'algorithme de remplacer les mauvais individus par des nouveaux, qui sont créés à partir du matériel génétique existant dans les meilleurs individus. Deux manières de faire cela sont envisagées ici :

- Croisement uniforme à n -parents : tous les paramètres correspondants à une liaison dans un individu sont remplacés par des valeurs correspondant à un parent choisi au hasard, mais avec un biais vers les individus les plus performants.
- Croisement uniforme à $m \cdot n$ - parents : chaque paramètre d'une liaison est remplacé par la valeur correspondante qui se trouve dans un parent choisi au hasard, avec le même biais que ci-dessus. Par m nous désignons le nombre des paramètres que chaque liaison comporte.

Mutations

La mutation est l'un des principaux opérateurs d'une stratégie évolutionnaire. Son rôle est d'assurer la diversité de la recherche dans l'espace des paramètres, diversité qui serait perdue en appliquant uniquement le croisement. Par la seule combinaison des paramètres, nous risquerions de bloquer l'algorithme dans des points correspondants à des combinaisons des morceaux de chromosomes initiaux.

Les mutations sont appliquées avec une probabilité assez faible sur la population entière (entre 1% et 5%) ce qui nous dit qu'un petit nombre d'individus par génération sont affectés par cet opérateur. Un taux d'application plus important peut favoriser trop la recherche aléatoire au détriment du croisement. Maintenant nous allons exposer la différence que présente la mutation dans un algorithme génétique par rapport à une stratégie évolutionnaire. Celle-ci se manifeste surtout dans l'implémentation algorithmique. Dans le cas des algorithmes génétiques la mutation est un changement de l'état d'un gène qui d'habitude est représenté sur un bit. Dans le

cas des stratégies évolutionnaires la mutation est perçue en tant que « bruit ». Le processus se réalise de la manière suivante : pour chaque paramètre, nous choisissons un individu au hasard parmi les moins performants et on lui modifie le paramètre. L'astuce consiste dans l'application d'un bruit gaussien d'une certaine variance et de moyenne égale à l'ancienne valeur. L'écart type s'appelle *température*, par analogie avec le recuit simulé.

Une variante de mutation complémentaire à celle décrite ci-dessus consiste à introduire des connaissances tirées des données analysées dans le matériel génétique de la population. Cela se fait en tirant au sort une longueur autour de la valeur moyenne observée sur les données initiales. Ce complément de mutation exploite directement les connaissances physiques sur le problème.

Différents tests effectués nous ont suggéré que l'utilisation d'une fonction de coût locale en tant que complément à la fonction de coût globale est un outil assez efficace pour exploiter les spécificités des systèmes particules-liaisons. L'utilisation d'une stratégie évolutionnaire auto-adaptative, comme nous allons le voir dans le paragraphe suivant, permet en revanche, d'automatiser à maximum la stratégie évolutionnaire, en la rendant moins dépendante des connaissances *a priori* qu'on a sur le problème.

3.2.1. Fonctions de coût multiples ; coût global à long et court terme, coût local

L'identification des modèles particules-liaisons se présente comme une résolution de problème inverse. Nous cherchons une représentation mécanique de l'objet qui soit capable d'engendrer le même comportement (trajectoire) que l'objet original, par le biais de la synthèse d'images. Ceci nous oblige à exprimer le problème inverse sous la forme d'une optimisation d'un modèle candidat de façon à minimiser une fonction coût qui mesure la ressemblance entre le comportement original et celui de synthèse.

Donc nous exprimons la fonction de coût par une distance mathématique entre la cinématique produite par le modèle candidat est celle de référence. Le problème inverse consiste donc à trouver le modèle qui a le coût le moins élevé parmi les modèles candidats. Idéalement, le modèle qui se plie à 100% sur l'objet original a une valeur de fonction de coût nulle.

Mathématiquement la fonction de coût peut se définir ainsi :

$$Coût(modèle) = \sum_{temps} \sum_{particules} \left((x_{pred_{t,i}} - x_{réel_{t,i}})^2 + (y_{pred_{t,i}} - y_{réel_{t,i}})^2 + (z_{pred_{t,i}} - z_{réel_{t,i}})^2 \right) \quad (3.1)$$

où :

- $X_{réel_{t,i}}$ est la première coordonnée sur la trajectoire réelle de la particule i , à l'instant t ;
- $X_{pred_{t,i}}$ est la première coordonnée sur la trajectoire de synthèse de la particule i , à l'instant t ; la trajectoire prédite est construite par l'application réitérée du modèle physique aux valeurs des coordonnées ($X_{réel_{t-1,j}}, \dots$) et de vitesses réelles ($V_{réel_{t-1,j}}, \dots$) de toutes les particules à l'instant précédent $t-1$.

Cette fonction de coût présente, malheureusement, quelques désavantages : dans le cas où la trajectoire présente un grand nombre de pas, une faible variation des paramètres du modèle induit à long terme une modification complète de la trajectoire. Le minimum global de la fonction de coût est donc difficile, à atteindre dans ces conditions.

Pour éliminer cette difficulté plusieurs voies sont envisageables :

- La première consiste à privilégier dans l'expression de la fonction de coût l'influence des termes les plus faibles, donc de ceux correspondant aux rapprochements les plus nets de la trajectoire de synthèse à l'originale. Ceci peut s'exprimer sous la forme suivante (apparentée à une moyenne harmonique):

$$Coût(modèle) = \frac{1}{\sum_{temps} \sum_{particules} \left(\frac{1}{(x_{pred_{t,i}} - x_{réel_{t,i}})^2 + (y_{pred_{t,i}} - y_{réel_{t,i}})^2 + (z_{pred_{t,i}} - z_{réel_{t,i}})^2} \right)} \quad (3.2)$$

- La deuxième méthode serait de calculer la fonction coût sur quelques pas de temps seulement en début d'algorithme, puis d'augmenter progressivement le nombre de pas au cours de son déroulement ;
- Une troisième méthode consisterait à calculer chaque point « prédit » par le modèle à partir des coordonnées (positions, vitesses) réelles à l'instant précédant et non pas à partir des points calculés depuis le début de l'algorithme.

Maintenant, nous allons analyser les trois variantes pour voir leurs points forts et faibles. La première solution donne avantage aux ressemblances entre la trajectoire de synthèse et originale, mais elle cache trop les différences. Il suffit d'une bonne ressemblance à un instant t

entre les trajectoires pour que le terme correspondant à celui-ci impose sa valeur dominante aux autres termes, calculés à d'autres instants de temps. Dans le cas limite où l'on a une seule et forte ressemblance entre les deux trajectoires à un certain moment, sur le reste des pas de temps ayant que des grandes différences entre les trajectoires, on peut obtenir une valeur quasiment nulle de la fonction coût, ce qui ne correspondra pas à la réalité.

La deuxième méthode garde le même inconvénient au début, mais celui-ci disparaît progressivement en augmentant le nombre de pas de temps pris en considération. La difficulté qui reste est de trouver une manière d'augmenter le nombre de pas en fonction de la convergence de l'algorithme.

Tenant compte de ce qui précède, Jean Louchet [L94] a proposé une troisième méthode : la fonction de coût est la somme quadratique des écarts entre les points de la trajectoire réelle et les points prédits dans la manière décrite auparavant.

Une faiblesse apparente de cette méthode serait une sensibilité aux bruits affectant la trajectoire réelle, car elle n'est basée que sur des prédictions à court terme. En réalité l'expérience a montré qu'elle reste assez robuste à cet égard, l'avantage que les calculs à court terme apportent à l'algorithme étant décisif pour l'algorithme d'optimisation.

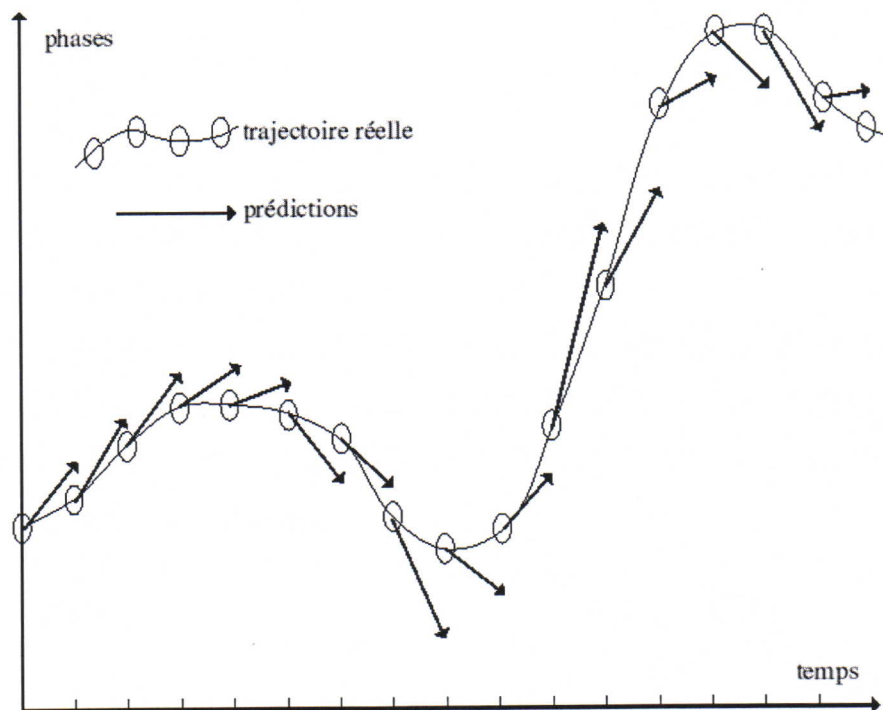


Figure 3.1

Calcul de la fonction de coût à court terme: le coût est obtenu par cumul des écarts entre positions réelles (cercles) et positions prédites à 1 pas de temps (flèches)

Le problème d'optimisation que l'on doit résoudre est donc de trouver dans l'espace des paramètres les bonnes valeurs au repos, raideurs et coefficients d'amortissement pour les différents types de liaisons. Les valeurs des masses des particules et le graphe de connections sont considéré connus.

La forme de cette fonction de coût ne nous permet pas de l'utiliser dans d'autres algorithmes d'optimisation qu'heuristiques. Cette fonction coût n'est pas appropriée à une résolution analytique ni à l'optimisation numérique. Nous avons testé le fonctionnement de cette fonction sur deux méthodes heuristiques, le recuit simulé et sur les stratégies évolutionnaires.

Nous allons maintenant présenter la manière dont nous pouvons exploiter au maximum la topologie des systèmes particules – liaisons, la modalité par laquelle leurs caractéristiques physiques peuvent être spéculées est transformées en qualités algorithmiques, sous la forme des fonctions de coût locales.

L'exploitation des caractéristiques topologiques des systèmes particules – liaisons est possible suite à l'hypothèse, d'inspiration physique, suivante [L94b] :

La position d'une particule à un moment donné t ne dépend que des positions et vitesses des particules voisines (en termes de liaisons) à l'instant $t-1$.

Cette propriété est très intéressante surtout pour les liaisons binaires qui possèdent un voisinage relativement restreint par rapport à d'autres types de liaisons, comme par exemple ternaires ou quaternaires.

La conséquence principale de cette hypothèse est que chaque modification sur une liaison binaire n'induit, à court terme, une modification directe sur les autres particules présentes dans le modèle qui ne font pas partie de son voisinage. Dans l'expression de la fonction de coût n'interviennent que des termes associés à chaque particule, calculés à base de différences à court terme. En modifiant les paramètres d'une liaison, nous ne modifions en fait que les termes correspondant aux particules qui se trouvent à ces extrémités.

Avec cette considération, nous pouvons décomposer la fonction de coût globale en une somme des fonctions de coût locales correspondant à chaque particule et adapter la sélection de la stratégie évolutive en s'appuyant sur les fonctions de coût locales des particules existantes aux

extrémités de cette liaison, éliminant ainsi les influences qui sont dues à la performance de cet individu dans d'autres régions du modèle. Cela signifie inversement que chaque fonction de coût locale, attachée à une particule, fait intervenir seulement les liaisons dont cette particule est une extrémité.

Ceci nous permet donc d'exploiter finement l'information topologique offerte par le modèle et d'obtenir une convergence de l'algorithme quasi-indépendante de la complexité de l'objet à modéliser.

La fonction de coût locale associée à chaque particule s'exprime comme somme temporelle des erreurs de prédiction des positions de cette particule :

$$F_{\text{coût_locale}}(\text{part}, i, \text{modèle}) = \sum_{\text{temps } j} \sum_{\text{voisins de } i} (\text{DistPréd}_{ij} - \text{DistRéelle}_{ij})^2 \quad (3.3)$$

La fonction de coût globale est donc la somme sur l'ensemble de particules, des fonctions de coût locales.

Nous présentons maintenant la manière dont ces innovation modifient le fonctionnement de l'algorithme :

- A chaque génération, toutes les fonctions de coût locales sont évaluées de manière à obtenir autant de critères de classement des individus de la population.
- Pour chacune des particules, nous mettons en œuvre des versions locales des processus de mutation et de croisement, qui ne modifient que les paramètres des liaisons auxquelles la particule courante est connectée. Cela permet au processus évolutif d'opérer localement sur un critère de performance locale. Nous utilisons ici l'hypothèse que la position d'une particule à un instant t ne dépend que des positions et des vitesses de ses voisins à l'instant $t-1$. Si le modèle est localement bon, sa fonction de coût locale aura une valeur basse, même si ailleurs le même modèle était mauvais. Le processus évolutif autour d'une particule donnée est donc quasiment indépendant de son environnement lointain. Cela fait que la convergence de l'algorithme sur des régions se fait indépendamment car les fonctions de coût locales n'interfèrent pas comme c'est le cas dans l'utilisation d'un critère global exprimé par une fonction de coût globale. Le nombre de générations nécessaires pour atteindre une certaine précision ne dépend plus

du nombre des particules et liaisons présentes dans l'ensemble de l'objet mais de la complexité moyenne du voisinage de chaque particule.

- La fonction de coût globale ne sera plus utilisée que dans le cas du processus de sélection où elle sert à protéger les meilleurs 5% individus les plus performants sur l'ensemble du modèle.

La quantité des calculs reste, donc, proportionnelle au nombre des particules et de liaisons et suivant la taille choisie de la population.

Les mécanismes de croisement et de mutation sont modifiés pour n'utiliser que les fonctions de coût qui concernent les particules liées aux liaisons dont les paramètres sont visés par le croisement ou la mutation.

3.3. L'approche Parisienne

Dans ce sous-chapitre nous allons présenter une variante de stratégie évolutionnaire dénommée l'approche parisienne ou co-évolution, qui a été utilisée [M99] dans une application d'identification de masses de particules, dans le même temps avec les paramètres, pour le cas d'un système simple formé de 3 et 4 particules.

L'idée est d'explorer la stratégie d'évolution d'un point de vue particulier inspiré par le paradigme prédateur - proie. Ce paradigme fonctionne de la manière suivante : le prédateur oblige les proies à se développer, développement qui à son tour oblige le prédateur de s'améliorer. Un autre paradigme, plus intéressant pour nous est celui de la symbiose des espèces, qui d'ailleurs est celui le plus utilisé : il présume que le développement d'une espèce est aussi utile à elle-même, mais aussi aux autres espèces. Ce genre d'algorithme est souvent efficace lorsque l'on a faire à un problème à beaucoup d'optima locaux.

Maintenant, nous allons discuter les différences entre les stratégies d'évolution classiques et l'approche Parisienne. Dans l'approche classique, nous exprimons le problème d'optimisation sous la forme d'une fonction de fitness ou de coût, selon l'optimum recherché. Une solution candidate, comme nous l'avons déjà expliqué, qui contient tous les paramètres du problème à

optimiser, représente un individu. Plusieurs individus forment une population qui est soumise aux opérateurs génétiques. Mais, ce que nous intéresse à la fin de l'évolution c'est juste le meilleur individu de toute l'évolution, le reste n'apportant pas d'informations supplémentaires.

Dans l'approche Parisienne, nous codifions une solution candidate sur une population entière ; explicitement, chaque individu représentera une partie de la solution, la solution candidate complète sera l'ensemble des individus. De cette manière toute la population est mieux exploitée.

La voie choisie donc est de séparer le problème en sous-problèmes à optimiser en partageant le chromosome d'individu classique en plusieurs chromosomes de longueurs inférieures.

Chaque morceau de la solution est considéré comme un problème en entier, on lui attache un algorithme évolutif qui, évolue de manière autonome mais qui coopèrent dans le même *écosystème*.

Chaque individu est évalué en fonction de l'amélioration qu'il apporte à la fonction de coût globale, il est mieux évalué si sa contribution fait descendre la valeur de la fonction de coût globale, c'est à dire qu'il coopère bien avec les autres morceaux de solution, ou dévalué si sa contribution est mauvaise, s'il coopère mal.

Nous allons maintenant discuter de la manière dont les systèmes particules-liaisons peuvent être sous-divisés, en tirant avantage de leurs particularités. Pour cela nous regrouperons ensemble les primitives liées de point de vue topologique. Une particule peut être liée à plusieurs liaisons mais une liaison binaire ne peut être liée qu'à deux particules. C'est à cause de ça que nous choisissons le triplet (m, k, l_ϕ) où k, l_ϕ désignent la raideur et la longueur de repos de la liaisons à laquelle il se réfère, m indique la masse de la particule contenue dans le triplet.

Soit un modèle M de N particules dont le codage de ses paramètres se présente sous la forme :

$M = \left(m_0, m_1, \dots, m_N, k_0, \dots, k_{\frac{N(N-1)}{2}}, l_{\phi_0}, \dots, l_{\phi_{\frac{N(N-1)}{2}}} \right)$. Nous partageons ces N^2 paramètres en $N(N-1)$

sous-problèmes.

Par exemple, le sous-problème numéro 0 dans le tableau 3.1 consiste à optimiser les paramètres m_0, k_0, l_{ϕ_0} . Le problème étant partagé en $N(N-1)$ sous-problèmes, nous utilisons alors $N(N-1)$ algorithmes évolutifs.

N° du sous-problème	les paramètres	N° du sous-problème	les paramètres	N° du sous-problème	les paramètres
0	m_0, k_0, l_{ϕ_0}	4	m_1, k_3, l_{ϕ_3}	8	m_2, k_5, l_{ϕ_5}
1	m_0, k_1, l_{ϕ_1}	5	m_1, k_4, l_{ϕ_4}	9	m_3, k_2, l_{ϕ_2}
2	m_0, k_1, l_{ϕ_2}	6	m_2, k_1, l_{ϕ_1}	10	m_3, k_4, l_{ϕ_4}
3	m_1, k_0, l_{ϕ_0}	7	m_2, k_3, l_{ϕ_3}	11	m_3, k_5, l_{ϕ_5}

Tableau 3.1
Décomposition de l'optimisation

En tant que fonction de coût nous considérons la fonction de coût globale qui est calculée à partir de fonctions de coût locales, déjà présentée dans le sous-chapitre précédent, qui nous donnera un indice sur la performance globale du modèle. Mais, la différence dans le cas de l'approche Parisienne est que la fonction de coût d'un individu sera proportionnelle à *l'amélioration* qu'il apporte au système des particules global. L'individu ne sera plus une partie de la solution mais une *modification* de la solution.

Des résultats intéressants ont été obtenus dans le travail de DEA de S. Marukatat [M99], résultats obtenus sur des petits systèmes de particules, de 4, 6, et 10 particules. L'inconvénient majeur de cette approche a été le grand nombre de stratégies d'évolution classiques (au nombre de $N(N-1)$ pour N particules) qui doivent tourner au sein de l'approche Parisienne.

Une autre remarque que nous devons faire et qui manque aux développements actuels de l'approche Parisienne appliquée aux systèmes de particules, se réfère à l'impédance du système. Plus explicitement nous pourrions avoir une infinité de paires de valeurs de masses et de

constantes des liaisons qui donnent la même cinématique. Par contre si nous croisons deux modèles ayant la même fonction de coût, c'est à dire cinématique, nous obtiendrons un modèle mauvais, à cause des grandes différences entre les masses des particules et des différents ordres de grandeurs des raideurs. C'est en effet un problème d'unités, qui en mécanique équivaut à une différence d'impédances. Pour résoudre cela une condition de masse totale fixe du système sera suffisante.

3.4. Auto-adaptation

La première stratégie évolutionnaire a été développée en 1964 par Rechenberg et Schwefel en tant que technique d'optimisation expérimentale. Dans une première phase la stratégie avait seulement deux individus (un père et un descendant) par génération. Cette stratégie, dénommée (1+1)-ES, crée le descendant en utilisant une distribution binomiale avec une espérance 0 et une variance σ^2 qui sont appliquées au parent. De cette manière le descendant devient parent dans la prochaine génération s'il est *meilleur* (plus tard on va expliciter ce que *meilleur* veut dire) que l'ancien parent ; dans le cas contraire l'ancien parent survit. Rechenberg a développé une théorie sur la vitesse de convergence de la (1+1)-ES et il a aussi proposé une règle déterministe selon laquelle la variance de la stratégie peut être adaptée en fonction du taux de succès des mutation. Cette règle a été dénommée la règle 1/5e.

Très vite la (1+1)-ES a été remplacée par des variantes avec $\mu > 1$ parents et $\lambda > 1$ descendants par génération. La première stratégie dans cette série, la $(\mu + 1)$ -ES avec $\mu > 1$ a introduit le concept de recombinaison a fin de fabriquer un descendant à partir de plusieurs parents en recombinaison leurs caractéristiques. Deux autres opérateurs sont appliqués, l'opérateur de mutation et l'opérateur d'évaluation. Le premier opérateur agit sur les gènes modifiant leurs valeurs. L'évaluation est une opération dans laquelle la performance de chaque individu est mesurée en fonction d'un critère exprimé sous une forme mathématique, la fonction de coût. Ce type de stratégie n'a pas été très utilisé mais il a le mérite d'avoir ouvert la transition vers les stratégies modernes d'évolution de type $\left(\begin{matrix} \mu + 1 \\ \mu, 1 \end{matrix} \right)$ -ES [Rec94,Sch95]. Dans le cas de la $(\mu + \lambda)$ -ES, μ parents créent λ descendants par recombinaisons et mutations. Les meilleurs

μ individus parmi les parents plus les descendants sont choisis en tant que parents dans la prochaine génération. Dans le cas de la stratégie (μ, λ) -ES, les meilleurs μ individus parmi seulement les descendants sont gardés ; dans ce cas la condition $\lambda \geq \mu$ est requise.

La stratégie (μ, λ) -ES remplace le processus déterministe de la règle 1/5e par l'auto-adaptabilité des mutations pendant le processus d'évolution [HB92, Sch95].

Les recherches actuellement se concentrent vers le développement des différentes variantes de stratégies évolutionnaires qui exploitent, par exemple, les particularités *hardware* ou qui essayent de résoudre des problèmes de décision à critères multiples, etc.

Nous allons utiliser une notation généralisée pour pouvoir traiter les deux types de stratégies évolutionnaires ensemble. Nous introduisons le coefficient de la durée de vie pour un individu κ .

Les stratégies évolutionnaires contemporaines $(\mu, \kappa, \rho, \lambda)$ -stratégies contiennent quelques caractéristiques qui sont de point de vue théorique ou expérimental, nécessaires à l'algorithme.

Désormais nous considérons un problème d'optimisation n - dimensionnel dont les paramètres varient en continu. La forme sous laquelle le problème s'exprime est la suivante :

$$f^* \equiv f(x^*) = \min \{ f(x) \mid x \in M \subseteq R^n \} \quad (3.4)$$

avec $M = \{ x \in R^n \mid g_j \geq 0 \forall j \in \{1, \dots, q\} \}$ qui est un ensemble de points représentatifs et $g_j : R^n \rightarrow R$ sont des contraintes d'inégalité. Les individus $h' \in I$ de la stratégie évolutionnaire sont des vecteurs dans un espace vectoriel $I = R^n \times S$, où S est un jeu de paramètres propres à la stratégie évolutionnaire qui établissent ses caractéristiques particulières.

La population $P^{(t)} = \{a_1, \dots, a_k\} \in I^k$ représente un ensemble de $k \in \{\mu, \lambda\}$ individus à chaque génération t . $\mu, \lambda \in N$ dénote le nombre de parents et de descendants, respectivement. Les opérateurs génétiques sont introduits par les transformations suivantes :

$$\begin{aligned} \text{rec} : I^\mu &\rightarrow I && \text{recombinaison} \\ \text{mut} : I &\rightarrow I && \text{mutation} \\ \text{sel}_\mu^k : I^k &\rightarrow I^\mu && \text{selection, } k \in \{\lambda, \mu + \lambda\} \end{aligned} \quad (3.5)$$

Une itération de la stratégie évolutionnaire qui fait le passage d'une population $P^{(t)}$ à la suivante $P^{(t+1)}$ peut être représentée par un seul opérateur défini ci-dessous :

$$opt_{ES} : I^\mu \rightarrow I^\mu \quad (3.6)$$

où

$$opt_{ES}(P^{(t)}) = sel_\mu^k \left(\lambda \left\{ mut(rec(P^{(t)})) \right\} \cup Q \right) \quad (3.7)$$

où $Q \in \{P^{(t)}, \emptyset\}$.

Dans le cas (μ, λ) - ES nous avons les conditions $Q = \emptyset, k = \lambda$; pour le cas $(\mu + \lambda)$ - ES $Q = P^{(t)}, k = \mu + \lambda$.

Désormais, on va noter $z \sim N(\zeta, \sigma^2)$ la distribution normale avec la moyenne ζ et la variance σ^2 .

Schwefel a été le premier à utiliser une stratégie évolutionnaire auto- adaptative dans laquelle les taux de mutation peuvent s'adapter dans le même temps que l'algorithme tourne. La solution du problème a été trouvée en regardant de plus près les mécanismes que le génotype naturel possède, notamment celles qui commandent les mutabilités des gènes par l'intermédiaire de différents types d'enzymes. Ces propriétés empruntées aux stratégies évolutionnaires se traduisent par le fait que les variances des mutations deviennent parties intégrantes de l'individu et elles évoluent de la même façon que les variables et elles sont soumises également à toutes les opérations de mutation et croisement que les variables subissent. Ce processus s'appelle *auto-adaptation des paramètres* [Sch77].

La (μ, λ) - ES est équipée d'un jeu de paramètres de contrôle qui représentent une *distribution normale n- dimensionnelle* et servent à la mutation de l'individu :

$$I = R^n \times R_+^{n_\sigma} \times [-\pi, \pi]^{n_\sigma}, \quad (3.8)$$

où nous convenons de noter par $S = R^{n_\sigma} \times [-\pi, \pi]^{n_\sigma}$.

Un individu à la forme suivante $\vec{a} = (\vec{x}, \vec{\sigma}, \vec{\alpha}) \in I$ et il est composé par les éléments suivants :

- $\vec{x} \in R^n$: vecteur des variables objet. C'est le seul qui compte dans l'évaluation de la fonction objectif (ou coût) ;
- $\vec{\sigma} \in R_+^{n_\sigma}$: vecteur des variances ($1 \leq n_\sigma \leq n$) de la distribution normale ;
- $\vec{\alpha} \in [-\pi, \pi]^{n_\alpha}$: vecteur des angles d'inclinaison ($n_\alpha = (n - n_\sigma / 2) \cdot (n_\sigma - 1)$), qui définit les corrélations linéaires des mutations du vecteur des variables objet.

Les paramètres de la stratégie établissent les variances et les covariances de la distribution n-dimensionnelle qui est utilisée pour explorer l'espace des variables.

Le nombre des paramètres attachés à une stratégie évolutionnaire peut être modifié par l'utilisateur en fonction de ses sentiments concernant le degré de liberté que la topologie de la fonction d'objectif demande. Du point de vue empirique, une règle qu'on peut établir est que plus l'algorithme est complexe, plus le temps de calcul est grand. Les jeux de paramètres les plus utilisés sont les suivants :

$n_\sigma = 1, n_\alpha = 0$: mutations standard avec un paramètre de mutation qui contrôle la variance pour toutes les mutations des composants du vecteur x

$n_\sigma = n, n_\alpha = 0$: mutation standard avec les $\sigma_1, \dots, \sigma_n$, paramètres de mutation correspondant aux variables x_i du vecteur objet

$n_\sigma = 1, n_\alpha = n \cdot (n - 1) / 2$: mutations corrélées avec une matrice complète de covariance pour chaque individu.

$n_\sigma = 2, n_\alpha = n - 1$: dans une direction de l'espace de recherche, la recherche se déroule avec une variance σ_1^2 tandis que dans toutes les autres directions de l'espace de recherche perpendiculaires à cette direction on utilise la variance σ_2^2 .

Pour expliciter l'idée de mutations corrélées, nous choisissons le cas $n = 2, n_\sigma = 2, n_\alpha = 1$ qui est illustré dans la figure ci-dessous. On doit remarquer que les déviations standard σ_1 et σ_2 établissent la relation des longueurs des axes principaux de l'ellipsoïde, tandis que le paramètre α_{12} représente l'angle de rotation de l'ellipsoïde. Dans le cas général des mutations corrélées,

l'ellipsoïde de mutation peut s'orienter arbitrairement dans l'espace de recherche n -dimensionnel.

Selon la structure générale des individus, nous définissons l'opérateur de mutation de la manière suivante :

$$mut : I \rightarrow I \quad (3.9)$$

$$mut = mu_x \circ (mu_\sigma \times mu_\alpha) \quad (3.10)$$

Cet opérateur est appliqué après la recombinaison sur un individu

$$\hat{a} = (\hat{x}_1, \dots, \hat{x}_n, \hat{\sigma}_1, \dots, \hat{\sigma}_{n_\sigma}, \hat{\alpha}_1, \dots, \hat{\alpha}_{n_\alpha}) \quad (3.11)$$

Il agit d'abord sur la partie du vecteur formée par les paramètres $\hat{\sigma}$ et $\hat{\alpha}$. Le résultat de ces mutations représente le nouveau jeu des paramètres de la stratégie évolutionnaire, qui sont ensuite utilisés pour muter le vecteur \hat{x} .

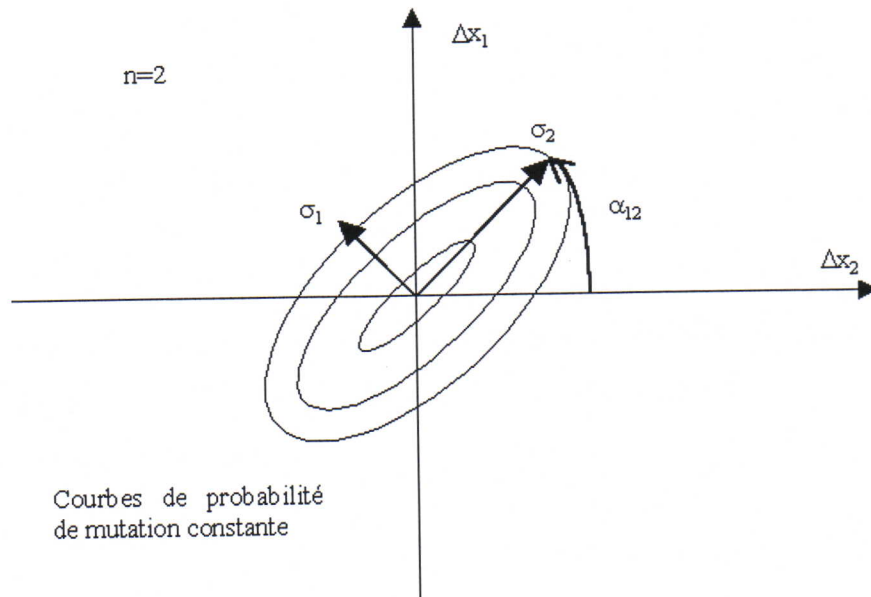


Figure 3.2
Correlations des mutations correspondent aux différents gènes

Les opérateurs de mutations sont définis selon les règles suivantes :

$mu_\sigma : R_+^{n_\sigma} \rightarrow R_+^{n_\sigma}$ qui réalise la mutation sur le vecteur $\hat{\sigma}$

$$mu_{\sigma}(\vec{\sigma}) \equiv (\sigma_1 \exp(z_1 + z_0), \dots, \sigma_{n_{\sigma}} \exp(z_{n_{\sigma}} + z_0)) \equiv \tilde{\sigma} \quad (3.12)$$

où

$$z_0 \sim N(0, \tau_0^2), z_i \sim N(0, \tau^2) \forall i \in \{1, \dots, n_{\sigma}\}. \quad (3.13)$$

Pour éviter que la déviation standard ne devienne zéro, une valeur minimale sera imposée pour tous les σ_i .

$$mu_{\alpha} : R_+^{n_{\alpha}} \rightarrow R_+^{n_{\alpha}} \quad (3.14)$$

$$mu_{\alpha}(\hat{\alpha}) \equiv (\alpha + z_1, \dots, \hat{\alpha}_{n_{\alpha}} + z_{n_{\alpha}}) \equiv \tilde{\alpha} \quad (3.15)$$

où

$z_i \sim N(0, \beta^2) \forall i \in \{1, \dots, n_{\sigma}\}$. Empiriquement, c'est prouvé que $\beta \approx 0.0873 (\approx 5)$ donne des bons résultats. Les angles de rotations sont convertis à leurs équivalents dans un intervalle raisonnable, par exemple $[-\pi, \pi]$, chaque fois quand l'angle de rotation dépasse cet intervalle.

$mu_x : R^n \rightarrow R^n$ réalise la mutation du vecteur des variables x , en utilisant les vecteurs qui ont déjà subi la mutation, $\vec{\sigma}$ et $\vec{\alpha}$:

$$mu_x(\vec{x}) \equiv (\hat{x}_1 + cor_1(\vec{\sigma}, \vec{\alpha}), \dots, \hat{x}_n + cor_n(\vec{\sigma}, \vec{\alpha})) \equiv \tilde{x} \quad (3.16)$$

où

$cor \equiv (cor_1(\vec{\sigma}, \vec{\alpha}), \dots, cor_n(\vec{\sigma}, \vec{\alpha}))$ est un vecteur aléatoire selon la distribution normale. Le vecteur de corrélations peut être calculé par l'intermédiaire de la matrice de rotations selon la formule :

$$cor \vec{r} = T \vec{z} \text{ où } \vec{z} = (z_1, \dots, z_n) \text{ avec } z_i \sim N(0, \bar{\sigma}_i^2) \forall i \in \{1, \dots, n_{\sigma}\} \quad (3.17)$$

et

$$T = \prod_{p=1}^{n_{\sigma}-1} \prod_{q=q+1}^{n_{\sigma}} T_{pq}(\vec{\alpha}_j) \quad (3.18)$$

avec $j = \frac{1}{2}(2n_\sigma - p)(p+1) - 2n_\sigma + q$ [Rud92a]. Les matrices de rotations $T_{pq}(\alpha_j)$ sont des matrices unitaires, exception faite des termes $t_{pp} = t_{qq} = \cos(\alpha_j)$ et $t_{pq} = -t_{qp} = \sin(\alpha_j)$.

L'usage d'une distribution normale logarithmique pour la variation des déviations standard σ_i est motivée par le fait que un processus multiplicatif garanti que les déviations standard restent positives et les petites variations sont favorisées par rapport aux grandes variations [Sch77].

CHAPITRE IV

Applications et limites

4.1. Identification générale et résultats au bruit sur des données synthétiques

Pour tester la performance de l'algorithme de l'identification sur des données synthétique, nous avons utilisé un protocole expérimental qui consiste en bâtir un objet arbitraire et se donner des conditions initiales afin de générer une trajectoire: l'épreuve consiste dans 'l'oubli' des paramètres de l'objet et d'utiliser la stratégie évolutionnaire décrite auparavant pour retrouver ces paramètres de la trajectoire synthétique donnée. Les courbes ci-dessous montrent la convergence de l'algorithme en fonction du nombre de générations, en identifiant un objet rigide qui consiste en 15 masses, 10 types de liaisons différentes (20 paramètres) et 31 liaisons installées, avec des différents niveaux de bruit sur la trajectoire. L'algorithme d'identification utilise 100 images consécutives et une population de 100 individus. Pour tester la résistance au bruit de l'algorithme, nous avons ajouté un bruit Gaussien aléatoire aux données cinématiques avec des différentes déviations standards.

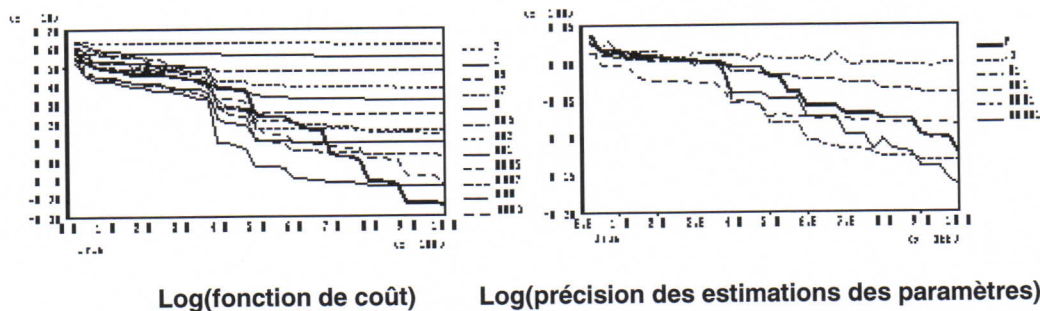


Figure 4.1.
La dépendance du bruit de la fonction de coût

Les niveaux asymptotiques peuvent être interprétés comme une distance entre les données des cinématiques bruitées et la plus proche cinématique physiquement constructible. D'autres tests montrent que la convergence est très bonne après 500 à 1000 générations, indépendamment du

nombre de paramètres devant être identifiés. Dans les résultats typiques de la convergence, montrés ci-dessus, l'erreur moyenne de l'estimation des paramètres après 1000 générations est inférieure à 0.01%, pour les trois bruits les plus bas, considérés.

4.2. Tissus

Provot [P95] a développé un modèle de tissu qui consiste en une maille périodique avec 6 types de ressorts: "ressort structurel" horizontal et vertical entre voisins, "ressort de cisaillement" diagonal et "ressort de flexion de longueur double". Une contrainte de la dynamique inverse autorise un taux de l'allongement maximal τ_c . Le modèle inclut 18 paramètres (6 liaisons de références décrites par 3 paramètres chacun). Grâce à l'utilisation de grands nombres de ressorts identiques dans des états d'élongations différents, nous pourrions utiliser seulement une seule image correspondant à l'état de l'équilibre et une seule autre correspondant à la fin de la séquence synthétique. La convergence est obtenue après approximativement 50 à 100 générations et elle nous donne des résultats en bon accord avec la référence. Le tableau en dessous montre quelques valeurs typiques des 5 paramètres du meilleur individu, dans une population de 100 individus, à la 50e génération, avec 1717 mailles. Les valeurs de la première ligne sont les valeurs de référence (qui ne sont pas connus par l'algorithme):

	longueur0 1	longueur2 3	longueur4 5	raideur01	τ taux d'élongation	coût
<i>référence</i>	0.093750	0.132583	0.187500	4.000000	0.500000	0.000000
value estimée	0.093984	0.132705	0.187083	4.084211	0.497108	0.014619
erreur d'estimation (%)	0.24	0.09	0.27	2.1	0.58	-

Tableau 4.1.
Valeurs typiques obtenues pour 5 paramètres

Le coût 0.0146 correspond à une erreur moyenne sur les positions des particules d'approximatif 5% de la période de la maille, et il donne des différences visuelles difficilement perceptibles par rapport à la séquence originale.

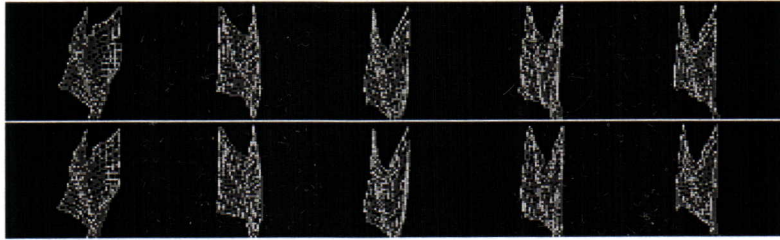


Figure 4.2.
Cinq cadres de l'animation originale et leurs reconstructions
à partir de paramètres identifiés (coût = 0.0146)

4.3. Mécanique des fluides

Les modèles basés sur des particules ont été longtemps utilisés pour simuler des objets fluides comme les nuages ou les fumées. Après Luciani [LHVD95] qui a obtenu des simulations visuellement réalistes des courants de fluides turbulents utilisant les liaisons visqueuses et élastiques conditionnelles, nous avons examiné comment notre algorithme d'identification pourrait reconstruire les paramètres d'un tel modèle de fluide turbulent.

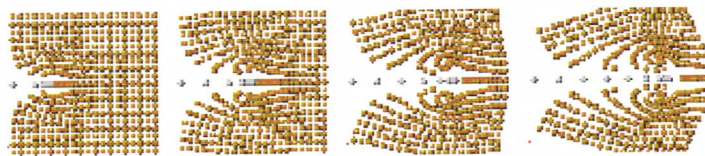


Figure 4.3.
Quatre images d'un jet qui pénètre dans un
fluide (cadres no. 100, 200, 300 et 400 d'une séquence)

La différence principale par rapport au modèle général est le grand nombre de liaisons installées (et par conséquent la charge computationnelle de la fonction de coût) comparé aux objets flexibles standards (approximativement 80000 liaisons en 2-D, exemple d'au-dessus) même si'ils ne sont pas tous activés en même temps. Avec 100 individus et après 1000 générations, l'algorithme donne de très bonnes évaluations des paramètres initiaux, avec la plupart des erreurs de l'estimation en dessous de 1%.

	gas visc	gas visc n,l	visc. dist	jet visc	jet visc n,l,	jet visc. dist	elast	élast. dist
<i>référence</i>	2.5	1.25	1.5	4	2	2	2	1.1
valeur estimée	2.499	1.248	1.500	3.962	2.052	2.000	2.000	1.100
erreur d'estimati on (%)	0.04	0.16	0.0	0.96	2.6	0.0	0.0	0.0

Tableau 4.2.
Résultats expérimentaux d'algorithme de l'identification

4.4. Cheval

Nous avons essayé de tester l'algorithme d'identification de systèmes particules – liaisons utilisant comme problème test, des données cinématiques obtenus d'un cheval de course. Sa silhouette a été relevée en utilisant 35 catadioptrés, a une fréquence d'échantillonnage de 50Hz. Sur chaque patte contenait 5 catadioptrés, le reste de 15 étant utilisés pour le corps et la tête. Nous avons mis en correspondance les points relevés avec les particules d'un modèle physique construit afin de pouvoir simuler le mouvement du cheval. L'expérience nous a montré qu'avec des ressorts, nous obtenons une bonne convergence de l'algorithme d'identification, mais le modèle obtenu à base des valeurs trouvées est fortement instable. En reconstruisant la trajectoire initiale, les premières images essaient de suivre la trajectoire de référence, mais après le modèle explose, les particules s'éloignent les unes des autres. L'introduction des amortisseurs apporte une faible amélioration (2%), et une stabilité supérieure au modèle. Pourtant le mouvement garde une caractéristique désordonnée, ce qui nous suggère que le modèle basé que sur de liaisons visco-élastiques n'est pas suffisamment riche pour nous permettre de simuler les créatures vivantes.

Cet exemple montre deux problèmes non résolus :

1. Les objets très structurés ne se prêtent pas à la même modélisation que les objets à faible degré de d'ordre ; nous ne pouvons pas modéliser avec les même primitives un gaz par exemple et un être vivant ;

2. Les créatures vivantes comportent des propriétés supplémentaires, comme par exemple l'existence des processus chimiques de fabrication d'énergie, à l'intérieur même du système et l'existence du contrôle central ou local par lequel cette énergie est distribuée.

Conclusion: nécessité d'enrichir la modélisation

La conclusion qui s'impose est qu'il est absolument nécessaire d'enrichir les primitives de modélisation lorsqu'il s'agit de phénomènes qui dépassent le cadre de la physique. Même si nous voulons rester dans le paradigme de la modélisation physique, des caractéristiques supplémentaires doivent être ajoutées.

CHAPITRE V

Introduction de la motricité dans Cordis

5.1. Modèle élémentaire de locomotion proprioceptive à base de modèle physique. Principes

Nous commençons ce paragraphe par une réflexion sur les conditions qui permettent l'introduction de la motricité au cadre de modélisation Cordis-Anima. Pour cela nous imaginons le véhicule le plus élémentaire qui pourrait se déplacer par ses propres moyens : un 'muscle' connecté à deux particules qui se trouvent à ses extrémités. Pour que ce système élémentaire soit capable de mouvement plusieurs conditions doivent être satisfaites. La première d'entre elles se réfère à la viscosité du milieu dans lequel le mobile se trouve : elle doit obligatoirement être non-linéaire. Nous remarquons que la locomotion existe toujours seulement si le mobile interagit avec son milieu externe. De plus cette interaction doit contenir une asymétrie quelque part dans sa définition, sinon la motricité n'est pas possible. Supposons que notre mobile évolue dans un environnement homogène, les deux particules connectées aux extrémités du véhicule doivent se trouver au même moment dans des zones de viscosité différentes, sinon le centre de gravité du système construit ne va pas bouger.

Les conditions donc qui doivent être respectées sont :

- La viscosité doit être non-linéaire avec la vitesse relative mobile – environnement ;
- Les particules doivent avoir des vitesses différentes au même moment, ce qui les oblige donc à avoir des masses différentes .

Ces deux conditions nécessaires pour que la motricité soit possible ne sont malheureusement pas suffisantes. La condition supplémentaire est la présence de l'injection de l'énergie et de la manière dont celle-ci se réalise. Cela peut être réalisé par l'utilisation d'un TGR (transducteur gestuel rétroactif) piloté par un signal asymétrique comme par exemple un signal en dent de scie. Un signal symétrique ne produira qu'un mouvement oscillatoire périodique qui passera toujours par les mêmes positions.

Le véhicule le plus élémentaire qu'on peut construire utilisant ces considérations a été dénommé *têtard* à cause de sa similarité avec cette créature.

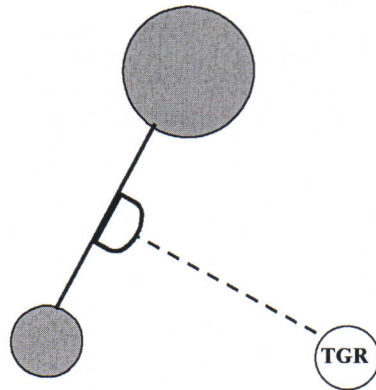


Figure 5.1.
Le Têtard

La structure de ce mobile est très limitée, il peut se déplacer en avant et en arrière sur une direction (univers uni-dimensionnel). Dans un univers multidimensionnel, par collision avec un obstacle il pourrait changer d'orientation en continuant son déplacement sur une nouvelle direction.

Le véhicule le plus élémentaire capable de changer la direction par lui même est constitué par une combinaison de deux têtards. Ce mobile a été nommé la *grenouille*.

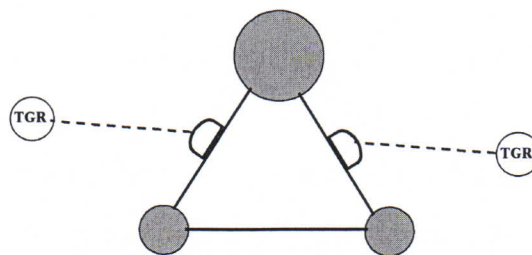


Figure 5.2:
La Grenouille

Si nous appliquons sur les deux TGRs deux signaux excitateurs générés par deux générateurs de pulses triangulaires, nous obtiendrons un véhicule capable de mouvement en absence de toute intervention humaine externe.

Nous allons maintenant appliquer ces considérations théoriques sur un exemple d'application qui est la recherche automatique d'une cible.

L'idée est de faire que le véhicule cherche la cible en lui donnant accès à l'information sur la position de la cible ; ces informations doivent s'inscrire dans le cadre de la modélisation physique (Cordis-Anima). Nous soulignons cette condition importante parce que le but de cette application est de montrer que les capacités de modélisation des modèles particules-liaisons vont au-delà de la modélisation de simples phénomènes physiques. Comme nous allons le remarquer, nous étendrons le champ d'application du *modèle physique particules-liaisons* (Cordis-Anima) vers des domaines comme l'automatique, contrôle, ou la psychologie du mouvement.

Dans l'application que nous nous proposerons donc, la façon la plus naturelle de fournir au mobile des informations sur la cible est de les connecter par un ressort. Dans le paragraphe suivant nous allons analyser ce système en détail.

5.2. Contrôle local ou contrôle central ?

La question fondamentale qui se pose lorsqu'on veut introduire des capacités de contrôle dans un système physique est la manière dont ce contrôle sera lié au système. Il y a deux possibilités de réaliser cela, la première est d'utiliser un contrôleur *central*, extérieur au système qui « observe » son évolution en fonction de quoi il prend des décisions et lui fournit des commandes. La deuxième approche serait l'utilisation d'un contrôleur distribué sur le système physique, donc *local*, qui se charge de diriger un seul composant du système. La seule contrainte que nous imposons aux deux approches est de nature phénoménologique : les deux doivent être compatibles avec la modélisation physique, elles doivent être basées sur des propriétés physiques, et non pas des propriétés d'une autre nature.

Pour cela, le contrôle central sera développé en tant qu'un autre système physique, différent du mobile, tandis que le contrôle local sera introduit par une propriété physique, supplémentaire, qui est le couplage mécanique mutuel des deux oscillateurs [SFLL01].

5.3. Le problème de l'injection d'énergie

Nous allons maintenant discuter de quelques difficultés que nous retrouvons lorsqu'on essaie d'introduire de l'énergie dans les systèmes discrets. D'abord, il faut mentionner que la discrétisation, elle-même introduit une distorsion dans la modélisation. Nous adaptons une théorie (la mécanique) qui s'appuie sur des lois continues sur un support (l'ordinateur) qui a une structure discrète.

Prenons le cas de la trajectoire d'une particule qui se déplace sous l'action d'une force. La discrétisation de cette trajectoire donnera une succession de points dans lesquels notre particule aura un couple position – vitesse bien déterminé. Maintenant, si nous voulons estimer l'accélération au moment t nous avons besoin des deux vitesses aux moments t et $(t-1)$. Cela nous donnera une approximation de l'accélération réelle, en continu, vu le fait que nous avons choisi un pas de temps. La différence entre cette approximation, meilleure lorsque le pas choisi est plus petit, et l'accélération réelle, constitue un résidu d'accélération dû à la discrétisation. Ce résidu correspondant à une particule à un certain moment de temps, multiplié par la masse de la particule donne naissance à une force résiduelle due à la discrétisation. Nous tirons donc la conclusion que la discrétisation induit une force résiduelle qui pompe de l'énergie dans le système. Cela fait que les systèmes de particules qui ne contiennent pas de dissipateurs (amortisseurs) énergétiques divergent au bout d'un nombre de générations de simulation à cause de cet effet. L'amortissement sert, entre autres, à stabiliser les systèmes discrets.

Maintenant, si nous voulons injecter de l'énergie dans le système, nous devons faire attention à la manière dont cela s'exécute ; si l'énergie s'accumule dans le système sous forme d'énergie potentielle plus vite qu'on ne la dépense, le système risque d'exploser après un certain nombre de pas de simulation. C'est à cause de cela que nous devrions envisager d'utiliser une sorte de régime transitoire, dans lequel l'injection se fait vite et le système a le temps de *se relaxer* ensuite.

5.4. Muscle à contrôle central

5.4.1. *Transducteur gestuel rétroactif (TGR). Muscle*

Le concept de muscle à contrôle central a comme point de départ la notion de *proprioceptivité*. Cette notion jugée indispensable dans la définition de muscle artificiel est tirée de la biologie. Dans ce chapitre, elle sera présentée d'une manière qui la rend compatible avec l'environnement de modélisation Cordis-Anima. Nous présenterons aussi les principes de la proprioceptivité et ses modalités d'implémentation utilisant les modèles physiques.

La proprioceptivité est la sensibilité propre aux muscles, os et ligaments. La locomotion est un jeu de fonctions qui font le mouvement possible. Nous définissons la *locomotion proprioceptive* comme le mouvement qui fournit un retour sensitif à son système de locomotion propre. Ce retour se réalise en fonction de son contact avec le milieu externe. Dans d'autres mots on peut dire que le mobile « sent » l'environnement où il évolue. Cette sensation est matérialisée par un retour d'effort de la part de l'environnement vers le mobile. La proprioceptivité est donc un autre canal de communication entre le mobile et l'environnement, mais cette fois à un niveau sensitif.

Un point important de ce travail est de décrire la manière dont un tel concept pourrait s'implémenter à l'aide de modèles physiques.

Le premier pas est de décrire l'introduction d'une action externe dans un environnement Cordis par l'intermédiaire d'un dispositif.

Nous appelons TGR (transducteur gestuel rétroactif) un instrument physique capable de fournir une force comme résultat d'une action (par exemple la variation d'une position). La tâche du TGR est de permettre soit à l'opérateur humain d'interagir directement avec l'objet simulé soit de connecter un objet avec son environnement externe par l'intermédiaire d'un canal gestuel.

L'architecture d'un tel dispositif est présentée dans la figure ci-dessous :

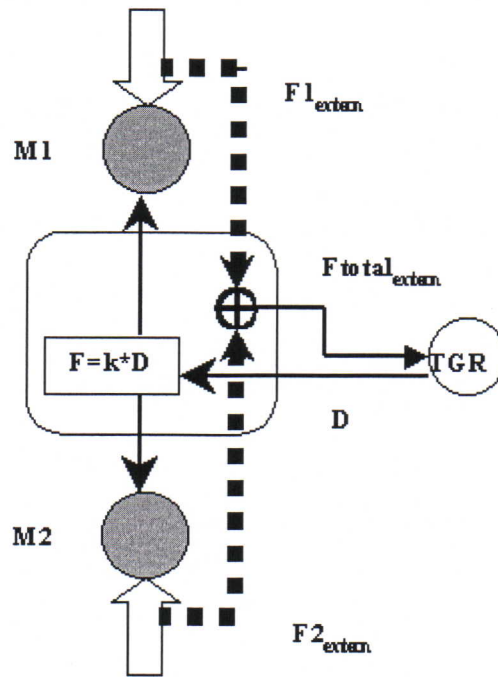


Figure 5.3.
Le transducteur gestuel rétroactif

Elle est définie en accord avec le formalisme de modélisation Cordis-Anima comme du plus petit ensemble de primitives qui peuvent implémenter la proprioceptivité. Le coeur du TGR est constitué d'un bloc fonctionnel appelé *module sensori-moteur*, celui qui assure l'existence de la proprioceptivité.

Le TGR est un dispositif à trois extrémités, deux masses et un point de commande dénommé *tendon*. Le cœur du système est constitué par un module de type <LIA> appelé module sensori-moteur. Son rôle est de calculer et d'appliquer des forces opposées, aux masses M_1 et M_2 connectées à ses extrémités. Cette chose se passe chaque fois qu'on lui fournit une distance en tant que variable de commande. La distance est fournie au système sensori-moteur par le tendon, qui agit comme un élément <MAT>, parce qu'il reçoit une force de la part de l'utilisateur et fournit une distance au système sensori-moteur.

La manière dont le TGR fonctionne est la suivante : l'opérateur actionne sur le tendon (parfois improprement appelé TGR, à cause du fait que c'est la seule partie accessible à l'opérateur, le reste étant un système physique qui tourne en ordinateur) en y appliquant une force de commande. Cette force est convertie par l'atome <MAT> qui est le tendon dans une distance

qui, après est fournie au système sensori-moteur. Ce système de type <LIA>fabrique deux forces visco-élastiques opposées est égales en valeurs, qui se répercutent sur les deux extrémités représentées par les particules M_1 et M_2 . Les particules interagissent avec le milieu dans lequel elles évoluent provoquant une réaction de la part de ce milieu, conforme au principe physique d'action - réaction, toujours respecté par le formalisme Cordis. La résultante vectorielle de toutes ces réactions externes calculée par le système sensori-moteur est renvoyée au tendon. L'opérateur, donc, reçoit la réaction du mouvement qu'il a provoqué dans le système particules - liaisons sous la forme d'un « effort » qui s'oppose à son action.

Le module sensori-moteur est appelé *muscle* à cause de son analogie avec le muscle réel qui commande les forces exercées à ses extrémités par l'intermédiaire d'un signal nerveux.

5.4.2. *Contrôle musculaire par couplage de liaisons*

Un des buts de notre démarche est de munir les systèmes physiques, non seulement avec des capacités motrices, avec des capacités de commande. Comme nous l'avons vu ci-dessus, le muscle agit en présence d'un signal de commande. Ce signal provient d'un monde extérieur au muscle, dans le cas classique c'est l'humain qui commande l'action du muscle. Dans d'autres cas, plus automatisés, un système de commande d'une autre nature, électronique ou informatique par exemple, peut jouer le rôle de l'homme.

Notre démarche est aussi de trouver un système de commande qui remplace l'homme, mais à condition que ce système reste un système physique, de la même nature que le système moteur, appelé par commodité *mobile*. Nous voulons donc construire un *contrôle physique*. Ceci transmettra des signaux de commande au système motrice, *musculaire*, ces signaux étant des paramètres physiques.

Maintenant, notre préoccupation est de définir la modalité dont deux systèmes physiques, un mobile et un contrôleur peuvent être interconnectés de manière que chacun d'entre eux garde son rôle, sans que nous obtenions un autre système physique, plus grand, dans lequel les variables (mécaniques) de commande et motrices deviennent indiscernables.

Pour cela nous faisons l'hypothèse suivante :

Le système physique (mobile) et son système de contrôle (contrôleur) doivent évoluer dans des mondes physiques différents, munis chacun de leur propre paramètre *temps*.

Ceci est une condition qui rend le contrôle physique possible. Le contrôleur et le système contrôlé doivent appartenir à des univers mécaniques différents, autrement le paradigme du contrôle perd sa signification.

Nous pouvons affirmer que deux mondes physiques sont *séparés, si et seulement si*, ils ont des variables temporelles différentes, indépendantes les unes des autres. Dans notre cas, le monde physique dont nous concevons la représentation mentale a son paramètre du temps, pendant que le monde mobile a son propre temps. Cette condition de l'indépendance des temps assure que les processus dans un monde ne sont pas déterminés par l'écoulement du temps seulement dans l'autre monde, mais par l'intermédiaire des variables mécaniques.

Ces considérations nous seront utiles pour la construction des modules d'interconnexion pour la transmission de commandes entre des systèmes physiques, modules qui préservent en même temps leur autonomie.

5.4.3. *Le quadripôle et les transmetteurs*

La signification du quadripôle est de lier ces deux liaisons, en leur permettant de changer des variables mécaniques.

Ce module doit rester complètement compatible avec les principes de la modélisation physique et celui-ci doit rester compatible avec les autres primitives définies. Nous avons pris notre inspiration de la théorie des systèmes électriques et nous avons décidé de construire l'équivalence d'un transformateur en mécanique, que nous appelons le quadripôle.

Le quadripôle, comme son nom le suggère, a quatre points de connexion, divisés en deux paires de deux dipôles.

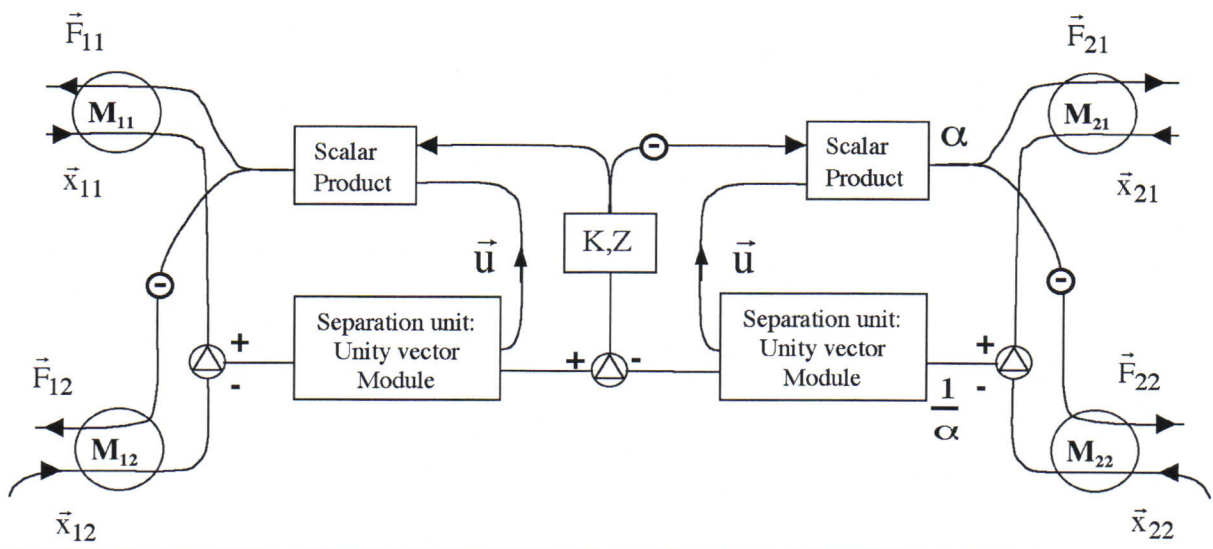


Figure 5.4.
Le Quadripôle

Le quadripôle est un dispositif symétrique qui transfère le travail mécanique d'un dipôle à un autre aussi bien que le transformateur le fait. Ses dipôles évoluent dans des systèmes mécaniques différents, qui échangent de l'énergie entre eux à travers l'intermédiaire du quadripôle. Dans l'absence de la dissipation énergétique, le même travail mécanique doit entrer et sortir les deux connexions dipolaires. Comme le produit scalaire entre la force exercée dans un dipôle et la distance relative entre les particules d'un dipôle est constant, les quadripôles peuvent multiplier la force par un coefficient, dans le même temps diminuant la distance relative par la même valeur, et vice versa. Le fonctionnement du quadripôle est le suivant: les vecteurs des distances relatives aux extrémités, sont décomposés dans les valeurs de leurs modules et leurs vecteurs unitaires (direction et orientation). Les modules sont utilisés pour calculer une force scalaire visco-élastique à travers de l'intermédiaire d'un bloc qui a la constante d'élasticité K et le coefficient de la viscosité Z ; les équations représentées dessous nous montrent le calcul exécuté :

$$F = -k\Delta_t(\text{modules}) - Z[\Delta_t(\text{modules}) - \Delta_{t-1}(\text{modules})] \quad (5.1)$$

$$\Delta_t(\text{modules}) = \text{module}(\text{dipole1}, t) - \text{module}(\text{dipole2}, t) \quad (5.2)$$

La force scalaire obtenue de cette manière, est utilisée pour construire les vecteurs de la force pour chaque dipôle, en multipliant ce scalaire avec les vecteurs de l'unité de chaque dipôle. Cela est rendu par l'intermédiaire des blocs du produit scalaire. Les forces sont après redistribuées sur les particules composant des dipôles.

Maintenant, nous allons étudier deux cas particuliers de quadripôles, qui servent à la transmission des forces ou des positions d'un dipôle à l'autre.

Le *transmetteur de position* est un cas particulier du quadripôle décrit. Il est représenté comme un cas dégénéré du quadripôle, en coupant la branche droite du produit scalaire de la force, et en mettant la condition qu'il reste à zéro.

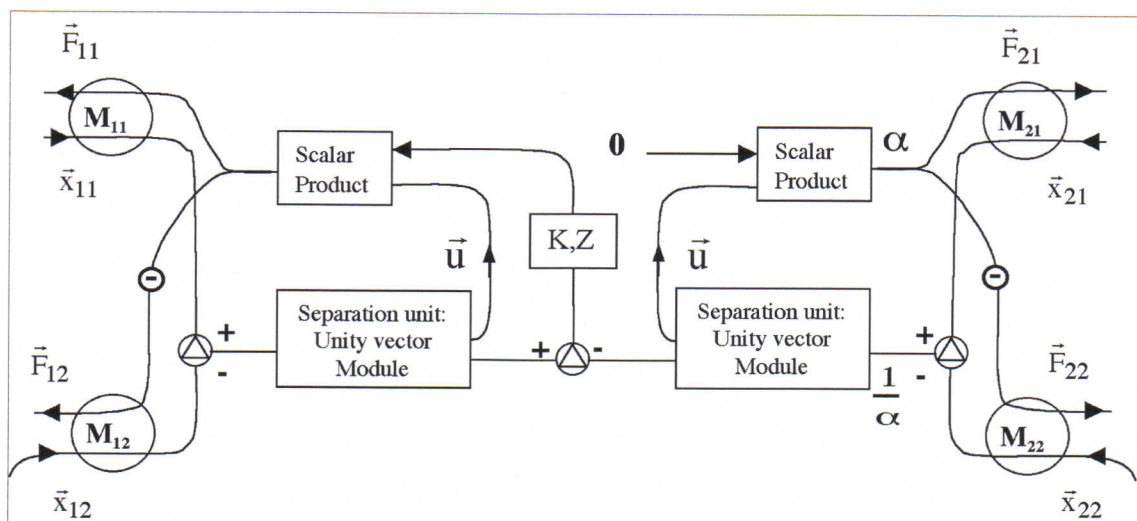


Figure 5.5.
Le transmetteur de position

La condition de force nulle, imposée dans la branche droite, est remplie si et seulement si le module de la distance relative à droite est égal au module de la distance relative à gauche. Cela veut dire que la distance relative du deuxième dipôle doit suivre l'évolution de la distance relative du premier dipôle, ce qui définit son fonctionnement de transmetteur de position. Sa tâche est reproduire la position dans le monde du système de contrôle, en celui du mobile.

Nous remarquons que ce module est complètement compatible avec les considérations de modélisation physiques que nous avons fait.

Un autre cas particulier du quadripôle est le transmetteur de force. Nous le construisons en coupant la branche gauche de la sortie du bloc, exactement comme la figure d'en dessous le montre.

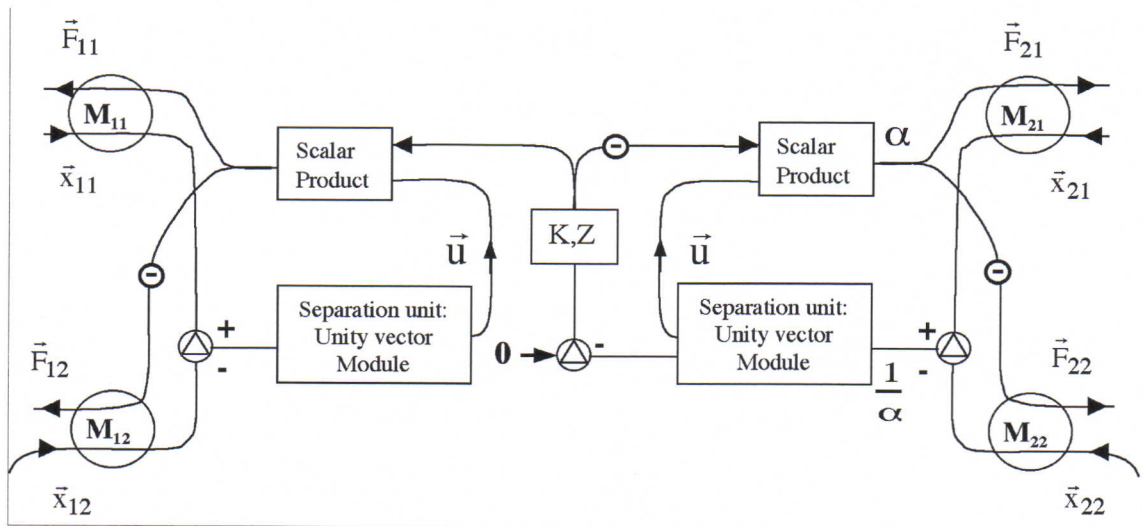


Figure 5.6.
Le transmetteur force

On remarque que mettre à zéro le module gauche, fait que la force calculée dépend uniquement de l'entrée droite. Par conséquent la force dans le dipôle gauche, indépendamment de son propre vecteur de distance relative, suit la force dans du dipôle droite. Ceci est le transmetteur de force et sa tâche est de transférer la force calculée dans le monde du contrôleur aux muscles du monde du mobile. Le transmetteur de force réalise aussi un rôle d'injecteur d'énergie dû au fait qu'il projette la même force sur des distances relatives différentes. De ce point de vue, il peut être considéré comme un élément actif dans la structure Cordis-Anima.

5.5. Couplage mutuel de liaisons par contrôle neuronal local

La liaison entre deux particules différentes était modélisée jusqu'à présent par une interaction visco-élastique, symbolisée par le couple ressort –amortisseur.

Cette approche, inspirée directement de la mécanique, donne de très bons résultats dans la modélisation des phénomènes physiques, mais elle échoue lorsque nous essayons d'identifier le

mouvement d'une créature vivante, surtout de la marche animale. Dans l'équipe ENSTA, Teisserenc [T97] a étudié la marche du cheval et il a conclu que les ressorts ne sont pas suffisants lorsqu'on essaie de modéliser la marche du cheval, donc il a décidé d'introduire les amortisseurs pour rendre le mouvement plus réaliste. Il a obtenu des meilleurs résultats en utilisant ce genre de liaisons, mais seulement lorsqu'il s'agissait de quelques parties particulières tel que jambes, tête, coude.

Un pas supplémentaire dans la modélisation des systèmes particules - liaisons a été l'introduction des ressorts généralisées [DLC93]. Malheureusement, tous ces éléments mentionnés ci-dessus sont des dispositifs passifs, incapables d'injecter de l'énergie dans le système de particules-liaisons. Par conséquent, la nécessité d'introduire des éléments de contrôle actifs s'est révélé comme le prochain pas.

Ces éléments actifs capables produire le mouvement (muscles), doivent réaliser aussi le rôle de coordination du mouvement des différentes parties autonomes.

5.5.1. Modélisation du muscle à contrôle local. Représentation du muscle à contrôle local

Notre démarche de chercher une manière de rendre effectifs les muscles locaux dans le paradigme des systèmes particules – liaisons, nous a apporté plusieurs idées intéressantes qui méritent d'être mentionnées ici.

- La première idée était de développer une analogie d'un milieu laser, comme une collection d'oscillateurs individuels, spatialement distribués, avec la possibilité de les exciter avec une onde passante. Le deuxième choix pourrait être l'étude des oscillateurs couplés, où le forçage et l'amortissement dans le système étaient les deux présents. Cette approche nous a menés à un système des équations différentielles couplées, qui exigent une résolution numérique. Malheureusement le calcul exigé est lourd, dû au grand nombre de variables (force, paramètres de la source).
- Une autre approche pourrait être le contrôle des coefficients d'amortissement, c'est à dire un contrôle sur le flou énergétique dans le réseau des particules-liaisons. Cet effet est très semblable à l'effet transistor, où la barrière de potentiel contrôle le courant entre les

deux autres électrodes. Cette idée exige un système de contrôle supplémentaire capable de décider de l'augmentation ou diminution des coefficients d'amortissement.

- La dernière solution envisagée est d'utiliser des réseaux neuronaux comme un système de commande pour toutes les liaisons dans notre modèle.

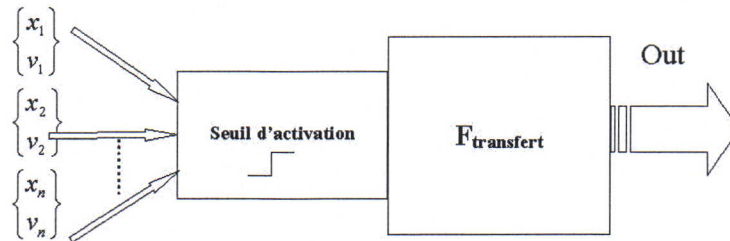


Figure 5.7.
Boîte de commande locale

Pour cela, nous avons proposé d'associer avec chaque liaison un neurone, qui a comme tâche d'observer l'évolution de toutes les autres liaisons qui composent le système mécanique. Le processus d'observation consiste à suivre pour chaque liaison voisine sa vitesse relative et sa position (Figure 5.8.). La sortie du neurone sera une fonction qui dépend de toutes les phases (couples vitesse – position) dans le système. Si la fonction de transfert est activée, son activation est le montant exact d'excitabilité qui sera appliquée sur sa propre liaison, à laquelle il est associé.

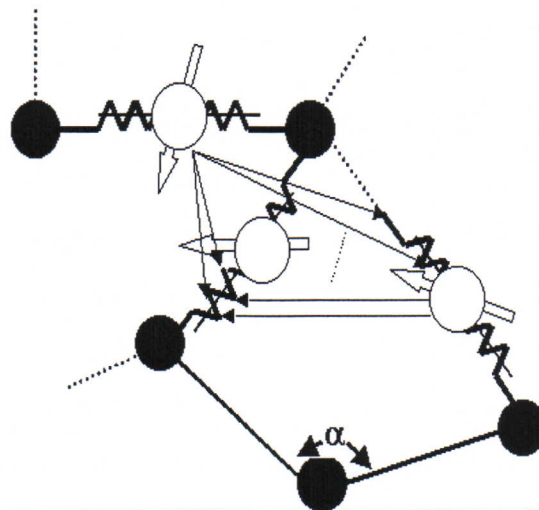


Figure 5.8.
Architecture basée sur des muscles à contrôle local

Cette dernière variante est celle que nous avons adoptée pour ses propriétés intéressantes d'interconnexion [SL00].

On peut se demander où se trouve l'inspiration physique de cette approche, pourquoi elle a attiré notre attention ? La réponse vient de l'historique du développement des réseaux neuronaux. Hopfield a été le premier à s'inspirer de la physique dans le développement des réseaux neuronaux. Il a modélisé le neurone en tant que spin dans un matériau ferromagnétique qui interagit par couplages mutuels avec ses voisins. Par abstraction mais restant totalement compatible avec la théorie ferromagnétique, il a montré la capacité d'un tel réseau de mémoriser des formes, de s'auto-organiser.

Revenant à notre problème de liaisons qui comportent des couplages mutuels, nous avons convenu d'exprimer par un *neurone* l'ensemble des couplages qu'une liaison subit dans un système. Ce mode de représentation est assez intuitive. Dorénavant lorsqu'on va se référer aux neurones associés aux liaisons, nous devons garder en arrière-pensée que le neurone modélise le couplage mutuel qui n'est autre chose qu'une propriété physique des oscillateurs.

Une entrée du neurone est une somme pondérée de toutes les vitesses relatives et position relatives des autres liaisons, exceptées celles de sa propre liaison, à laquelle il est associé.

Chaque neurone a son propre seuil d'activation, en dessous duquel il ne fait rien, et la différence entre la valeur à son entrée et son seuil représente la variable de sa fonction de transfert. Nous avons choisi la tangente hyperbolique en tant que fonction de transfert.

$$Out_{(ij)} = g(h_i) = \tanh\left(\sum_{(lk)} (\omega_{(ij)(lk)} d_{lk} + w_{(ij)(lk)} V_{lk}) - \theta_{(ij)}\right) \quad (5.3)$$

$$\vec{F}_i = \vec{F}_i\left(m_i, \vec{d}(m_i, m_j), \vec{V}(m_i, m_j)\right) + Out_{(ij)}\left(\dots, \vec{d}(m_l, m_k), \vec{V}(m_l, m_k), \dots\right) \quad (5.4)$$

Une manière convenable pour l'implémentation de ces idées est de modifier la structure existante des liaisons binaires et d'ajouter les paramètres supplémentaires exigés par le neurone.

La nouvelle liaison binaire sera l'ancienne ayant en plus tous les poids de connexion, associées avec les autres liaisons, et le seuil du neurone.

Paramètres du muscle à contrôle local:

$$\left\{ (\text{ressort} / \text{amortisseur}), (\vec{\omega}, \vec{w}) \right\} \quad (5.5)$$

La nouvelle liaison binaire agit comme un muscle à contrôle local [SL00].

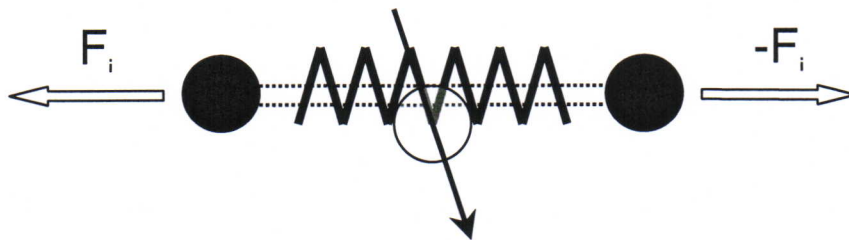


Figure 5.9.
Muscle à contrôle local

Sa structure mécanique est enrichie avec une unité de commande locale, représenté par son neurone associé. Par conséquent, chaque liaison binaire a la capacité de décider quand et suivant les conditions dans lesquelles le système se trouve, il doit injecter de l'énergie sur ses composants mécaniques (ressorts et amortisseurs).

Par la suite, nous appellerons ce nouveau genre de liaisons binaires, « neuro –mécaniques ». Donc, un système à contrôle neural est créé en parallèle avec notre système mécanique, une sorte de sandwich des deux systèmes, un mécanique en un système de commande distribué, d'une manière semblable à celle qui peut être trouvée dans toutes les créatures vivantes primitives.

CHAPITRE VI

Résultats expérimentaux

6.1. Véhicule utilisant les muscles à contrôle central

6.1.1. Identification des paramètres du têtard

Les points <MAT> du têtard interagissent avec l'environnement par une viscosité non-linéaire, caractéristique qui fait la locomotion possible, comme nous l'avons déjà expliqué dans le chapitre 4, section 4.1. Nous considérons la dépendance suivante de la viscosité avec vitesse:

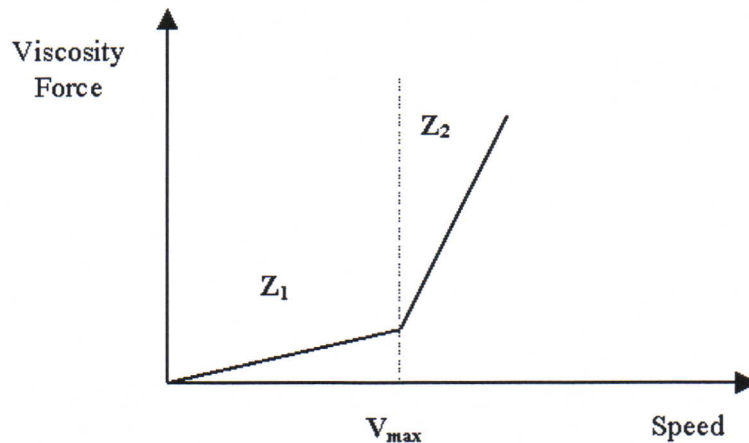


Figure 6.1.
Viscosité non-linéaire

Chaque fois que la masse a une plus grande vitesse, l'incrémentación de la viscosité la fait s'accrocher mieux à l'environnement. Le seuil de la viscosité, V_{max} , désigne la séparation entre les deux zones z_1 , z_2 de la viscosité. Le composant <LIA> du têtard a la raideur k et la longueur du repos L .

Dans le Tableau 6.1 nous présentons quelques résultats de l'identification des paramètres d'un têtard, à partir des données cinématiques. Cette cinématique est constituée par une séquence de 100 images d'un têtard artificiel. La dimension de la population de la stratégie évolutionnaire utilisée dans cette identification est de 1000 individus et le nombre de générations passées pour arriver à ce résultat est 15000.

	V_{max}	$z1$	$z2$	L	k
<i>Paramètres Réels</i>	0.5	0.1	1	1	20
<i>Paramètres Identifiés</i>	0.46	0.10	0.99	1.00	19.99

Tableau 6.1.
Identification des paramètres du têtard

La convergence de la stratégie évolutionnaire est graphiquement représentée dans Figure 6.2 d'en dessous [SFLLO1].

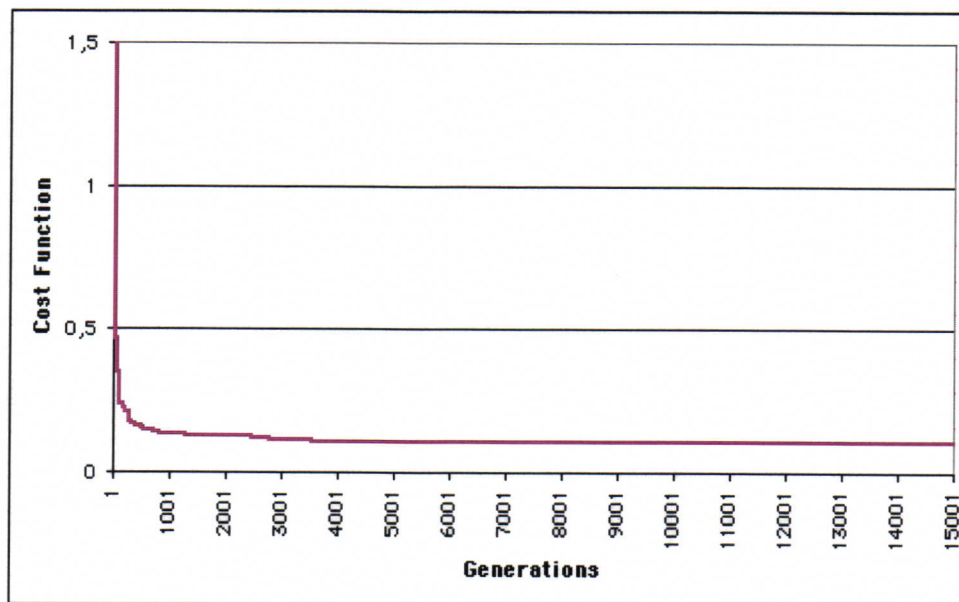


Figure 6.2.
Identification des paramètres du têtard par stratégie évolutionnaire

6.1.2. Introduction d'une commande externe: La Vision. Modèle de boucle Visio - motrice

Le mobile que nous utilisons à titre d'exemple a la structure physique suivante:

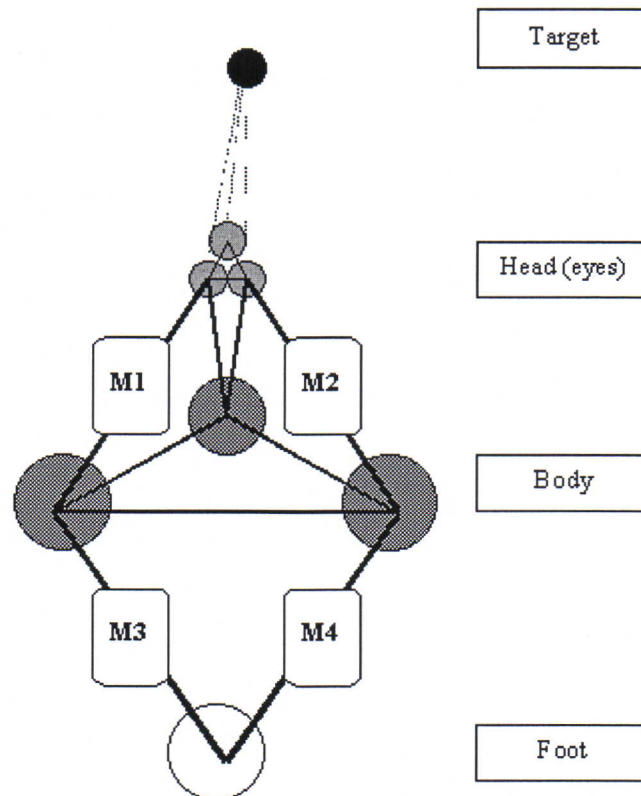


Figure 6.3.
Le mobile (métaphore de la tourelle)

Pour la conception du mobile nous utilisons un paradigme dénommé *la métaphore de la tourelle*. Cette métaphore suppose que notre mobile peut être divisé en niveaux fonctionnels qui travaillent à vitesses différentes. Le premier niveau (dénommé *la tourelle*) a la tâche d'identifier la cible et suivre son mouvement, par conséquent il travaille à une plus grande vitesse, pendant que le niveau bas, de la locomotion, a une petite vitesse et sa tâche est de suivre le mouvement de la tourelle. Dans la Figure 6.3 la tourelle est l'équivalent de la tête, pendant que le niveau locomoteur est l'équivalent du système du formé par l'ensemble corps-pieds.

La tête est composée par trois particules appelées *les yeux*, elles sont utilisées pour un calcul exact de la position relative de la cible par rapport à la position du mobile. Le nombre de trois a été choisi de raisons de précision. Si la cible est loin du mobile, et sa position est co-linéaire avec deux de ses yeux, la présence du troisième oeil nous permet de résoudre trouver (par triangulation) la position exacte de la cible. La tête est connectée au corps à travers deux muscles (M1 et M2) capable la conduire dans la direction de la cible. Il y a aussi deux connections visco-élastiques à une troisième particule dans le squelette du corps.

Le corps du mobile contient trois particules, deux d'entre elles sont les hanches, et la troisième a le rôle simple de stabilisateur de la tête, donc de *cou*. Les hanches sont connectées à une autre particule appelée *le pied* par les deux muscles (M3 et M4). La particule qui compose le pied assure le mécanisme le plus simple pour la propulsion du mobile.

Dans ce chapitre nous introduisons de *l'extéroceptivité* à notre mobile. Ce terme dénote la capacité du mobile de traiter l'information reçue de l'environnement externe. À cette fin le mobile doit présenter les caractéristiques suivantes:

- Il doit contenir un composant moteur, notamment le muscle à contrôle central développé dans le chapitre 4 ;
- Il doit avoir la possibilité de recevoir de l'information sensorielle au sujet de la position relative de la cible;
- Il doit être capable de transformer cette information dans des commandes par l'intermédiaire d'un centre de contrôle.

La boucle visio-motrice est la modalité par laquelle nous traduisons les trois caractéristiques mentionnées au-dessus dans l'architecture du mobile. Une représentation de cette boucle est montrée dans la Figure 6.4.

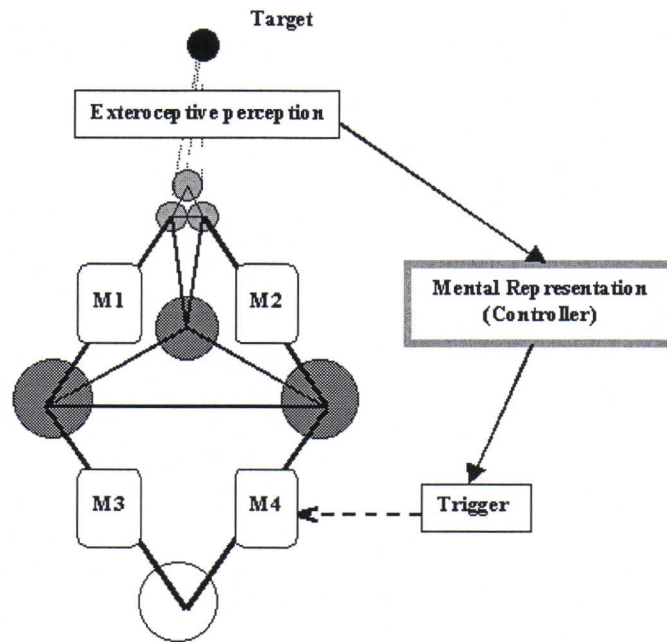


Figure 6.4
Boucle Visio-Motrice

Par l'intermédiaire de l'extéroceptivité du mobile, l'information est traitée par le *module contrôle central* qui, à son tour, commande les muscles. Le module de contrôle central du mobile rend effective la *représentation mentale de l'ensemble mobile-environnement*. Le concept de la représentation mentale du système mobile-environnement se réfère à la description de toutes les variables significantes qui interviennent dans la relation mobile-environnement, étant donnée une tâche (c.-à-d. suivre mobile la cible).

La cadre dans lequel nous travaillons (Cordis-Anima) avec notre engagement à la modélisation physique, nous fournit l'idée de construire un contrôleur central basé sur un système physique. La représentation mentale du système mobile-environnement paraît ici comme un processus physique complexe qui connecte l'espace des variables d'entrée (information sur la cible) et l'espace des variables de sortie (les signaux de commande des muscles).

Tout que ces éléments étant définis, nous construisons la représentation mentale du mobile comme il est montré dans la Figure 6.5:

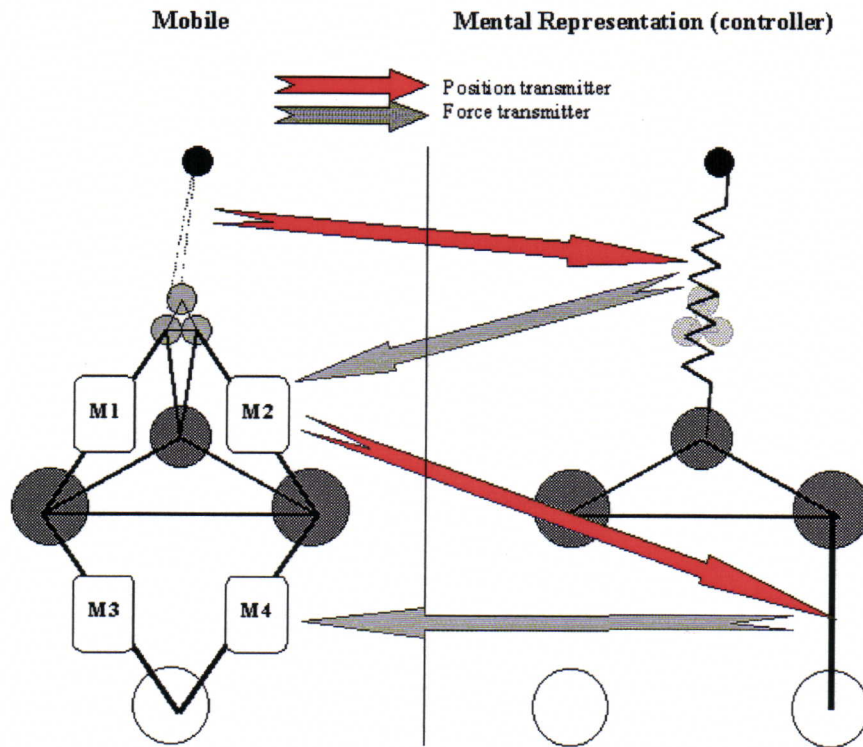


Figure 6.5.
La Représentation Mentale

Comme nous remarquons dans la figure d'en dessus, le contrôleur central (c.-à-d. représentation mentale) reçoit les positions de l'ensemble mobile-environnement par l'intermédiaire des transmetteurs de positions et il calcule, dans sa propre structure, les forces qui seront transmises aux muscles du mobile par le biais des transmetteurs de force. Dans la représentation mentale la relation '*le mobile suit la cible*' est représentée à travers un ressort qui connecte le corps du mobile à la cible. La distance entre le cou du mobile et la cible est calculée utilisant les yeux de la tête du mobile, et elle représente l'entrée pour le dipôle gauche d'un transmetteur de position. Le dipôle droite (sortie) du transmetteur de position est connecté dans la représentation mentale aux particules qui correspondent à la cible virtuelle et au cou virtuel.

La force calculée dans la représentation mentale par le ressort qui connecte les particules est projetée au muscle M2 du mobile à travers un transmetteur de force. Un processus semblable se produit pour la commande du muscle M1. Dans cette manière, la tourelle, dans notre cas particulier la tête du mobile, est guidée pour suivre la position de la cible. Comme nous l'avons spécifié lorsque nous décrivions la métaphore de la tourelle, le corps du mobile suit la tourelle

avec une vitesse de la réaction différente. Pour cela, la distance entre la tête du mobile (tourelle) et la particule droite du corps du mobile est fournie à la représentation mentale par le biais d'un transmetteur de position, qui se connecte à la liaison équivalente au muscle M4. La force calculée ainsi est rendue à l'unité de commande du muscle M4 du mobile. Dans une manière analogue le muscle M3 est commandé à travers la représentation mentale. On remarque qu'une communication à plusieurs niveaux apparaît entre le mobile et sa représentation mentale. En premier lieu, l'information de plus haut de niveau, c'est à dire position de la cible, est passée à la représentation mentale dans un niveau inférieur qui, après l'avoir traitée, la retransmet au mobile, au niveau de la tourelle. De cette manière nous avons fermé la boucle visio-motrice.

D'une manière semblable une autre boucle est utilisée pour commander le niveau de la locomotion en fonction de la place de la tourelle.

Dans les planches suivantes nous présentons des séquences d'images des animations basées sur la grenouille décrite au-dessus. Nous avons représenté dans le même espace 3D le déplacement du mobile qui suit la cible, à gauche dans l'image, à droite nous avons représenté le contrôleur central (représentation mentale) qui commande le déplacement du mobile vers la cible. Dans les images les deux balles singulières sont les représentations de la cible, à gauche nous avons la cible réelle dans l'espace réel du mobile et à droite la cible virtuelle dans l'espace de la représentation mentale. Le déplacement du mobile par rapport à la cible, dans l'espace réel, se traduit dans l'espace de la représentation mentale par un déplacement relatif de la cible virtuelle par rapport à la représentation mentale, qui n'exécute pas un mouvement de translation seulement sa forme change.

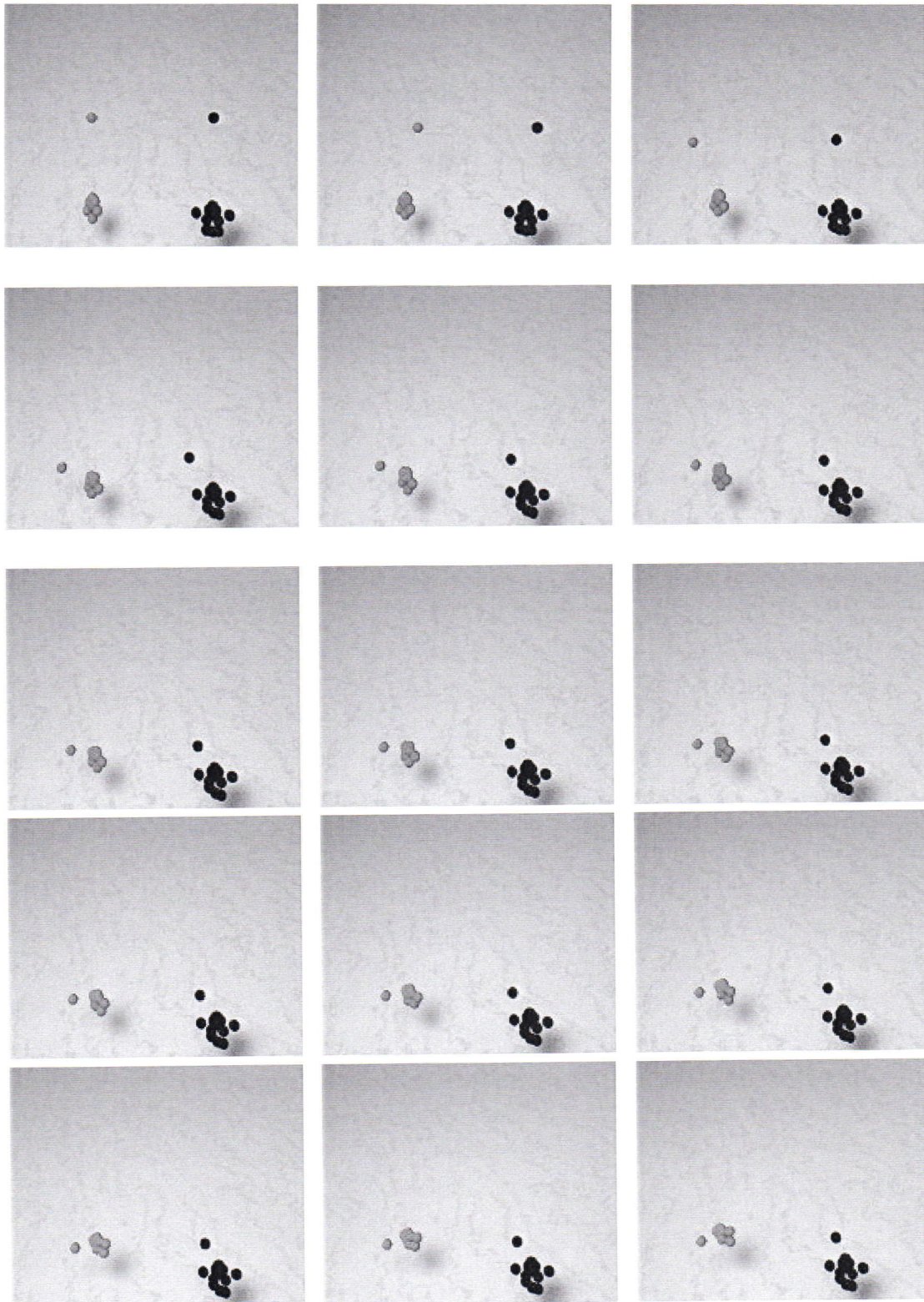


Figure 6.6.
Le mobile- suit-cible (gauche) et sa Représentation Mentale (droite)

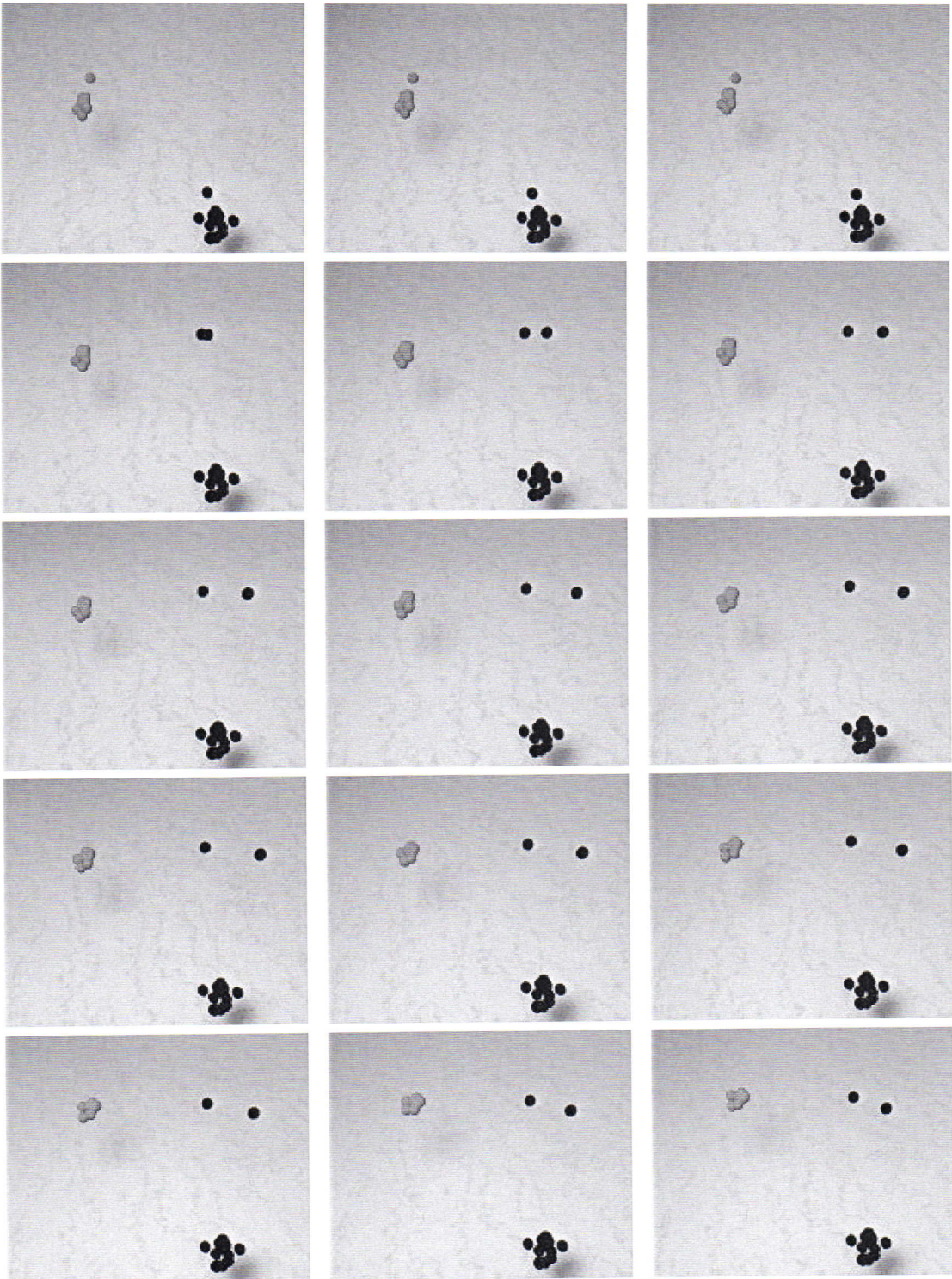


Figure 6.7.
Le mobile- suit-cible (gauche) et sa Représentation Mentale (droite)

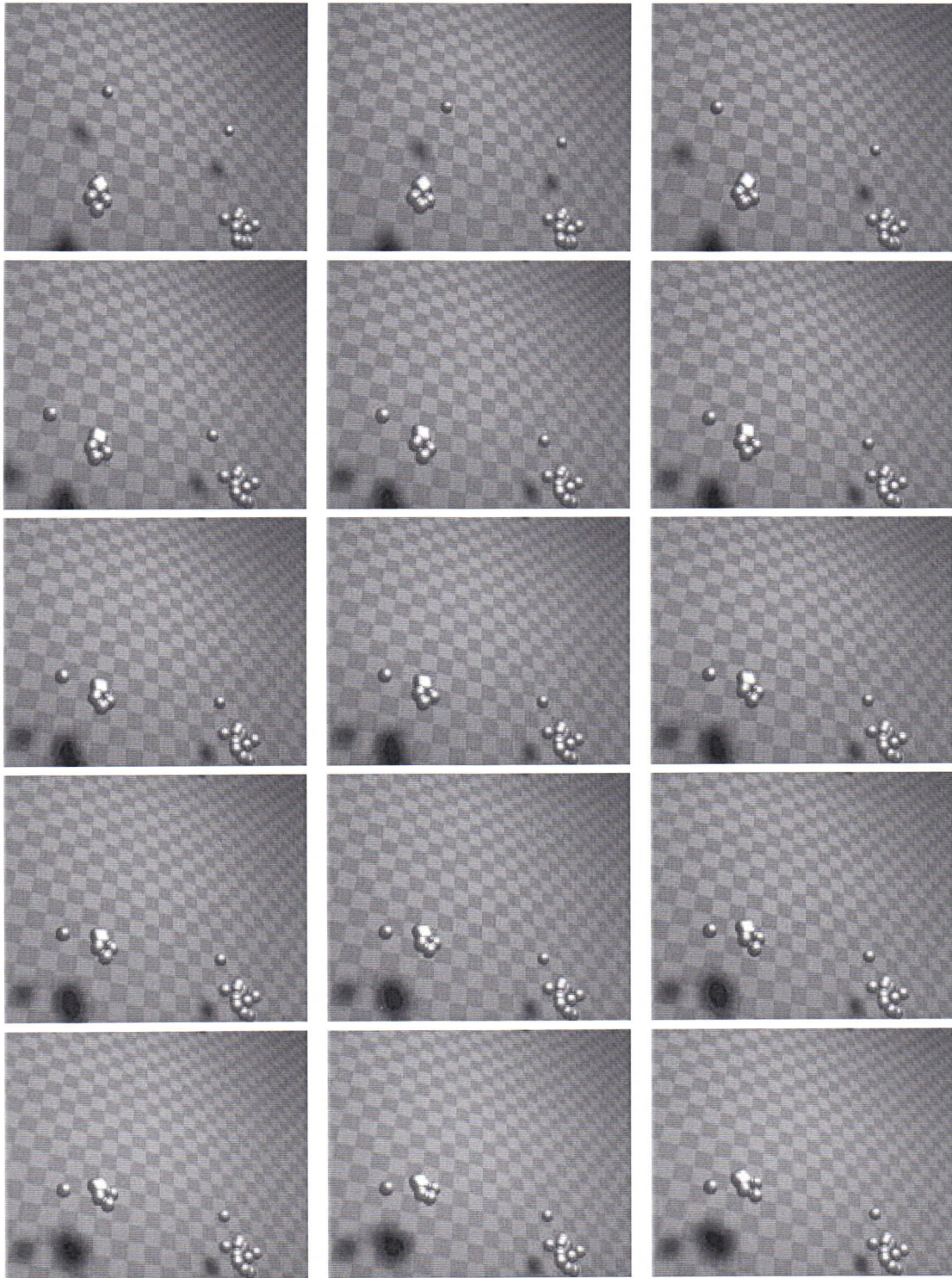


Figure 6.8.
Le mobile- suit-cible (gauche) et sa Représentation Mentale (droite)

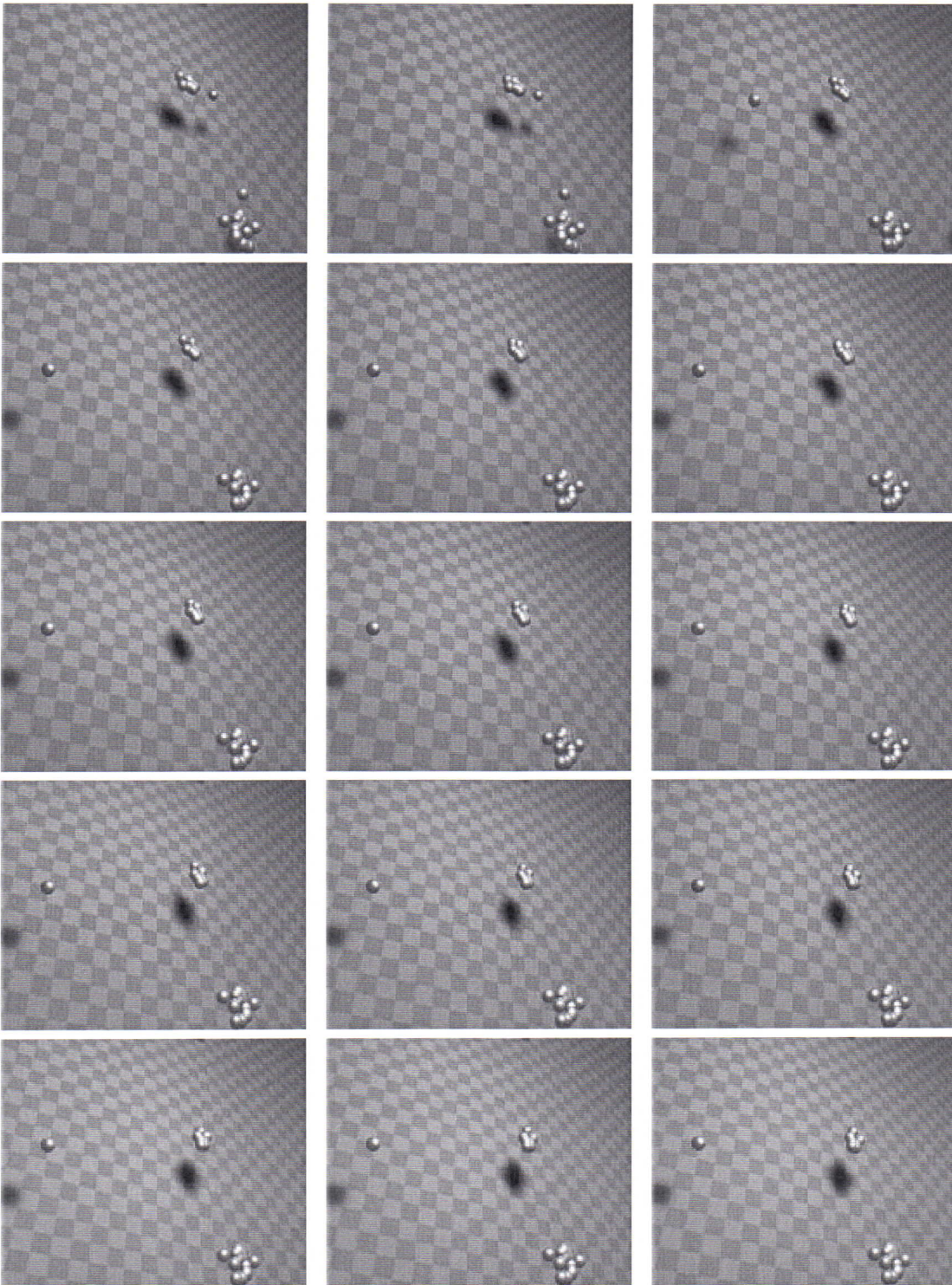


Figure 6.9.
Le mobile- suit-cible (gauche) et sa Représentation Mentale (droite)

6.2. Muscle à contrôle local ; Simulation et résultats

Nous avons testé le modèle de muscle à contrôle local proposé au chapitre précédent sur deux classes de trajectoires [SL00b].

6.2.1 Trajectoire Artificielle

La première classe est représentée par une trajectoire artificiellement produite utilisant un système des particules-liaisons standard dans lequel les liaisons sont purement visco-élastiques, ne contenant aucun contrôleur neural local.

Le test consiste en comparer l'évolution des deux méthodes, une identification standard basée seulement que sur une structure des particules-liaisons classiques, et la nouvelle méthode, basée sur des liaisons à contrôle neural local.

La Figure 6.10 nous montre la différence entre l'évolution des deux méthodes d'identification. La méthode standard converge mieux pendant les premières générations, mais elle est surpassée après quelques deux mille générations du par la nouvelle méthode. Les deux exemples suivants, les Figures 6.11 et Figure 6.12 sont des comparaisons semblables des mêmes méthodes. Nous observons que le résultat est presque le même. La dimension de la population utilisée ici est de 1000 individus.

Analysant les poids des connexions trouvés pour les neurones, nous concluons que l'ajout du système neuronal sur notre structure mécanique est une façon acceptable d'introduire le contrôle sur les liaisons visco-élastiques. Les poids du réseau neuronal ont de petites valeurs autour de zéro, ce qui indique que tous les neurones sont actifs. Une grande valeur pour un ou plusieurs poids suggère qu'un neurone est poussé dans la saturation (sortie ± 1), dû à sa fonction de transfert. L'effet mécanique est que le neurone force constamment une liaison entre deux particules, il pompe le même montant de force à chaque pas de temps.

Le cas opposé où un ou plusieurs des poids sont zéro nous suggère qu'une connexion entre deux liaisons manque, ce qui veut dire que les deux liaisons ont une évolution indépendante de l'une de l'autre, en termes mécaniques les deux liaisons ne sont pas couplées mutuellement.

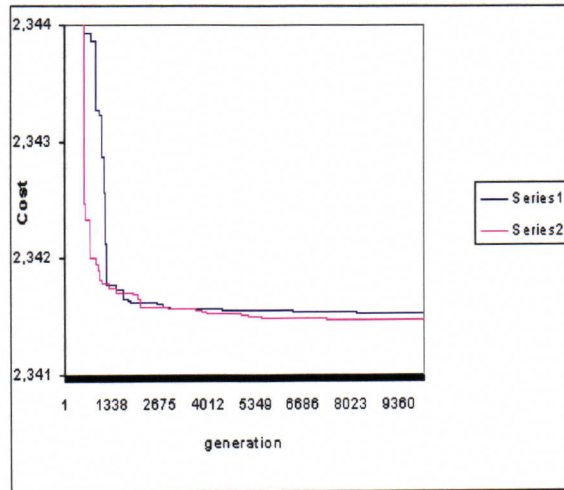


Figure 6.10.
Standard (Série 1) et nouvelle (Série 2) méthodes
(10,000 generations, 1000 individus)

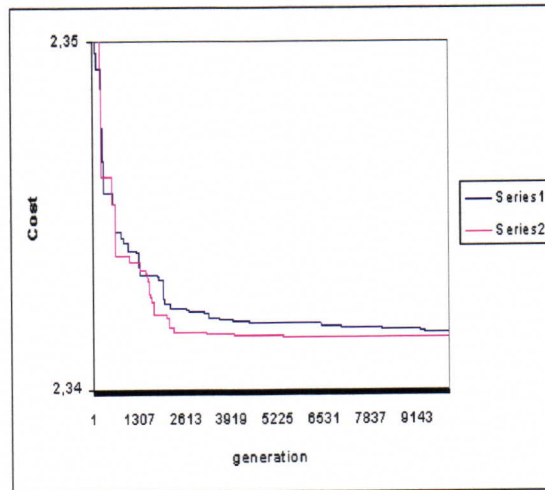


Figure 6.11.
Standard (Série 1) et nouvelle (Série 2) méthodes
(10,000 generations, 1000 individus)

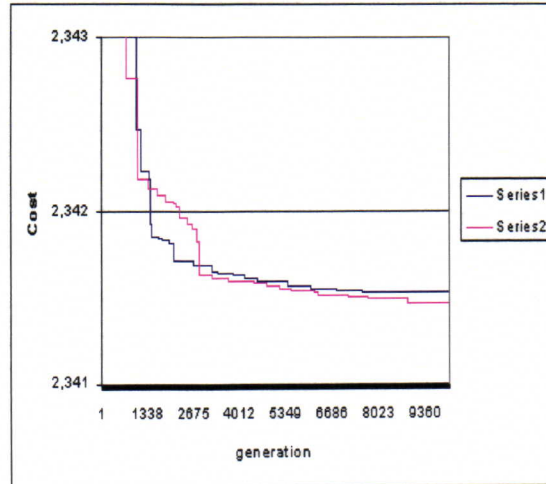


Figure 6.12.
Standard (Série 1) et nouvelle (Série 2) méthodes
(10,000 generations, 1000 individus)

Dans la Figure 6.13 nous avons fait la même comparaison, qui utilise la même taille de population sur 30.000 générations. La même tendance est maintenue: la méthode standard est marche mieux au début, mais la nouvelle méthode est sans aucun doute meilleure sur long terme, pourtant la différence reste petite. Cela peut être expliqué par le fait que notre trajectoire de test a été construite par un système de particules-liaisons standard, visco-élastiques, par conséquent ce n'est pas surprenant que la fonction de coût pour les deux méthodes convergent vers zéro.

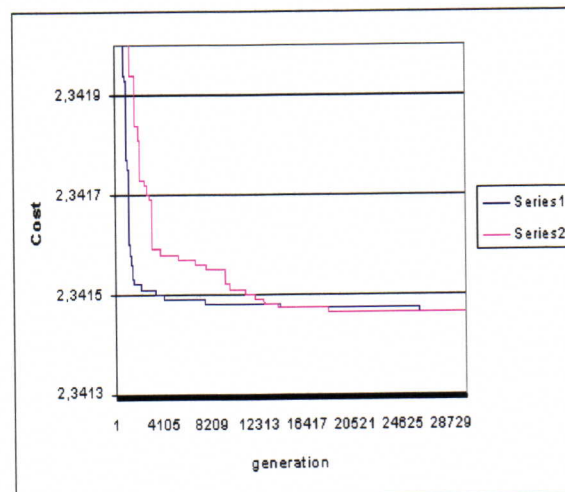


Figure 6.13.
Standard (Série 1) et nouvelle (Série 2) méthodes
(30,000 generations, 1000 individus)

6.2.2. Trajectoire réelle

La deuxième classe des trajectoires sur laquelle nous avons choisi de tester la méthode basée sur des muscles à contrôle local est la classe de trajectoires réelles, notamment la trajectoire de trois points localisée sur un cheval au trot (Figure 6.14¹) [SL00b].

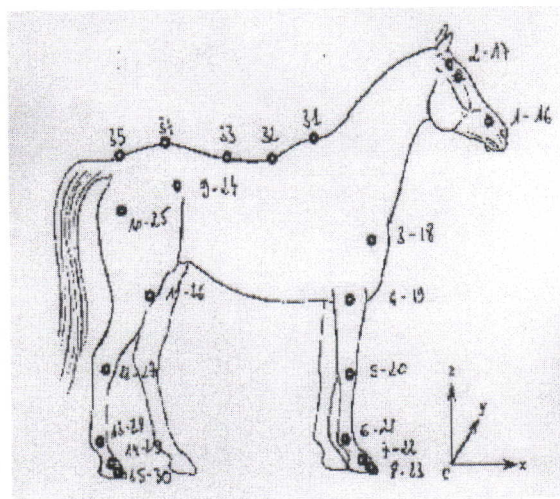


Figure 6.14
Les points choisis pour l'enregistrement de la trajectoire,
localisés sur le corps d'un cheval

Les données de la trajectoire ont été obtenus en plaçant trente-cinq catadioptrés sur le corps d'un cheval, son mouvement a été enregistré à une cadence de 50 Hz [T97]. Il y a 53 cadres contenant nos données. Le premier point est localisé à la bouche, le deuxième point est localisé sur l'extrémité supérieure d'une jambe, et le dernier point représente un pied (Figure 6.15).

¹ Les données du cheval ont été fournies par l'Ecole Vétérinaire, Maisons Alfort et Ecole d'Equitation Nationale, Saumur (France).

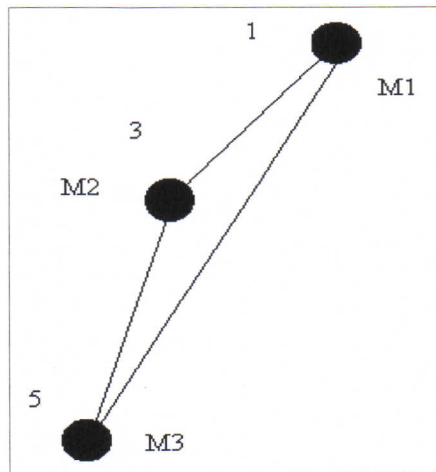


Figure 6.15
Les points choisis pour la comparaison des méthodes

Nous avons utilisé les premiers dix cadres de la trajectoire. Nous modélisons l'interaction des deux-particules utilisant des muscles à contrôle local. La structure de contrôle est composée de neurones à propagation en avant.

Par conséquent nous avons identifié les paramètres dans les deux cas (pur mécanique et systèmes neuro-mécaniques). Ici nous avons trois exemples de comparaison des deux méthodes, sur des différentes populations avec un nombre différent de générations. Dans Figure 6.16 les deux méthodes utilisent une taille de la population de 500 individus sur 50.000 générations.

Notre but est d'approximer au mieux possible les trajectoires. Le premier test était de vérifier l'auto consistance de nos méthodes, en testant un problème soluble à 100%. Comme on peut le remarquer ici, la différence entre les deux évolutions est plus grande dans le cas d'une trajectoire réelle que dans le cas d'une trajectoire de synthèse. Cela nous montre que la nouvelle méthode basée sur le contrôle local marche mieux que l'ancienne méthode utilisant des liaisons classiques. Il est évident que le système du muscle à contrôle local enrichit les capacités de modélisation du système.

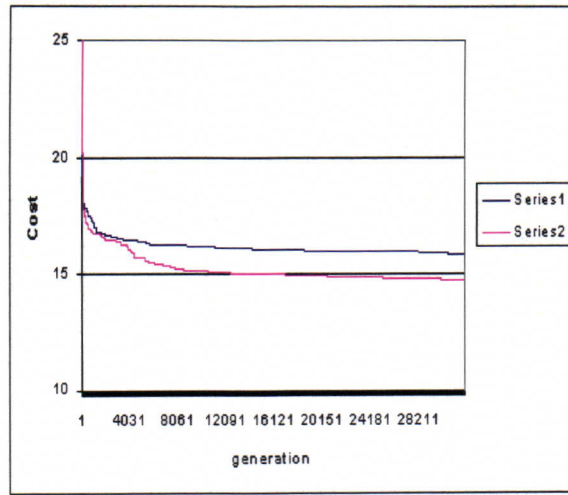


Figure 6.16.
Standard (Série 1) et nouvelle (Série 2) méthodes
(50,000 generations, 500 individus)

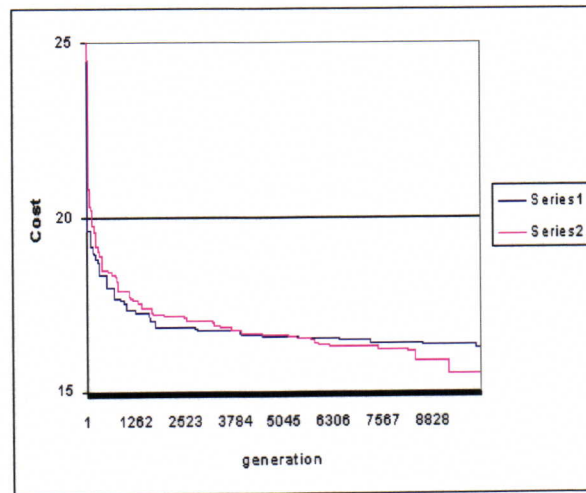


Figure 6.17.
Standard (Série 1) et nouvelle (Série 2) méthodes
(10,000 generations, 200 individus)

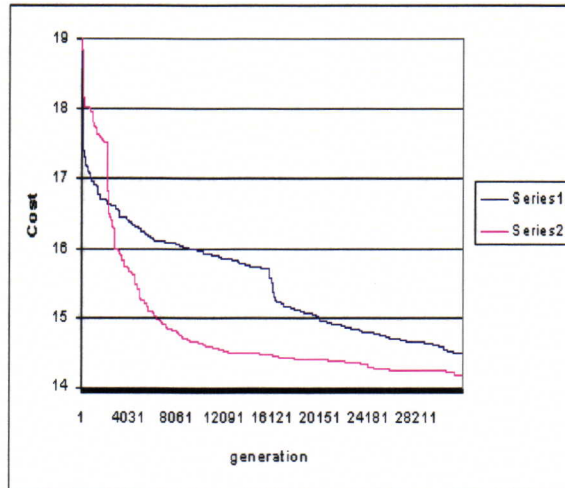


Figure 6.18.
Standard (Série 1) et nouvelle (Série 2) méthodes
(50,000 generations, 100 individus)

Les exemples dans les Figures. 6.17 et 6.18 montrent que la même tendance est maintenue pour les deux méthodes.

CONCLUSIONS ET PERSPECTIVES

Le but de ce travail était d'introduire la motricité dans les systèmes des particules-liaisons, pour qu'on puisse mieux modéliser une plus grande classe de trajectoires physiques.

D'abord nous avons commencé par l'étude des principes qui font la locomotion possible dans le cadre de modélisation Cordis-Anima. Nous avons décrit la manière dont, à l'ACROE, le retour d'effort a été développé et son implémentation physique sous la forme du TGR. Après la présentation de ce dispositif nous avons présenté deux types de mobiles de base, le têtard qui peut effectuer un mouvement uni-dimensionnel et la grenouille, un mobile qui peut explorer deux dimensions. Les deux mobiles contenaient des TGRs en tant que dispositifs moteurs, même si à ce stade nous ne savions pas comment les commander.

Dans la suite donc, nous sommes intéressés à la manière dont le contrôle pourrait être introduit. Deux approches complémentaires se sont démarquées, l'approche du muscle à contrôle central et celle du muscle à contrôle local.

Le muscle à contrôle central, présenté dans la suite de ce travail, a été introduit en définissant d'abord le muscle abstrait et après la notion de proprioceptivité.

Le pas intermédiaire entre la définition du muscle et l'introduction du contrôle central, a été celui de résoudre le problème d'interconnexion des deux liaisons, jugé comme base pour l'introduction du contrôle central. Non pas n'importe quel type de connexion, mais celle qui conserve les propriétés individuelles de chaque système impliqué. Nous avons fait l'hypothèse d'ordre qualitatif qui établit le niveau d'interconnexion selon la distinction système contrôlé – système contrôleur reste valide. Cette hypothèse, qu'on a pris au titre de postulat, qui sera validé après par les résultats, statue la barrière d'interconnexion des deux systèmes au niveau du paramètre temps ; l'interconnexion doit être faite en gardant les paramètres temps locaux inaltérés.

La suite a été constituée par la définition du quadripôle, véritable transformateur mécanique, dispositif de connexion élémentaire entre deux liaisons, qui permet l'échange de l'énergie d'un dipôle à l'autre, sans affecter la circulation locale des grandeurs physiques sur les deux cotés. Avec ces deux cas particuliers, de transmetteurs de positions et de forces nous avons disposé en Cordis-Anima de tous les ingrédients nécessaires à introduire le contrôle.

La suite donc a été de concevoir un système de contrôle basé sur de modèle physique.

Dans le chapitre 6, des résultats, nous avons développé une application qui nous a servi de prétexte pour développer un système de contrôle central.

L'interaction non-linéaire entre les particules et le milieu extérieur nous a obligé de traiter le problème d'identification de ses paramètres. Les résultats ont été plus que satisfaisants, nous avons retrouvé avec une très bonne précision les paramètres qui ont provoqué le déplacement d'un têtard.

La suite a été d'introduire une commande externe par l'intermédiaire d'un système de contrôle physique, construit dans le but de modéliser l'interaction mobile-milieu externe (dans notre cas un cible). Ce système de contrôle a été dénommé aussi « la représentation mentale » du monde mobile-cible. La boucle de connexion a été développée par la vision, en donnant aux mobiles des propriétés extéroceptives.

L'architecture du mobile a été conçue sur plusieurs niveaux hiérarchiques, utilisant une métaphore, notamment la métaphore de la tourelle. Dans cette manière nous avons décomposé la fonction du mobile dans des systèmes autonomes, connectés par des transmetteurs de grandeurs physiques.

Avec ces considérations le mobile a simulé avec succès la tâche de suivre la cible.

Pour introduire le contrôle local, nous avons choisi une approche basée sur des réseaux neuronaux à propagation en avant qui par analogie avec un système nerveux qui s'ajoute à la

structure mécanique exécutant une tâche de coordination de mouvement de ses composants. Du point de vue mécanique, nous pouvons considérer que les connections neuronales entre deux liaisons dans le système, agissent exactement comme un couplage mutuel entre deux oscillateurs. La valeur de l'intensité du couplage et la *coopération* des liaisons sont en effet une modalité de contrôle du réseau des particules-liaisons.

Deux pas ont été considérés pour la validation de notre approche. Tester l'auto consistance de l'algorithme que nous avons choisi sur une trajectoire de la classe de trajectoires qui peuvent complètement être identifiées par les systèmes des particules-liaisons standards. Les résultats obtenus étaient encourageants. La nouvelle approche, basée sur les liaisons neuro-mécaniques se comporte mieux que l'approche classique et converge vers la même valeur de la fonction coût. En outre, nous avons testé notre nouvelle approche sur une trajectoire réelle. Comme nous nous y attendions, la différence entre les deux méthodes était plus évidente dans ce cas, en faveur de l'approche neuro -mécanique.

Dans ce travail, nous avons montré comment il est possible de dépasser une des limitations des systèmes des particules-liaisons conventionnels dans la modélisation des mouvements complexes, comme nous le rencontrons dans les êtres vivants, en introduisant des liaisons à contrôle local neuro-mécaniques. Nous avons montré que la méthode d'identification évolutionnaire développée pour les liaisons conventionnelles visco-élastiques peut être étendue avec succès aux liaisons neuro-mécaniques et peut nous permettre l'utilisation de données cinématiques réelles pour construire le modèle.

Cependant, il est encore nécessaire de connaître les valeurs de masses et le graphe de la structure du système des particules-liaisons, pour permettre à l'algorithme d'identifier les paramètres mécaniques des liaisons et les valeurs de paramètres de contrôle local, neuronaux. La recherche future devrait viser l'identification automatique de masses et structures en utilisant des schémas évolutionnaires plus complexes.

Annexe 1

A1.1 Fonctionnement de l'algorithme d'identification

Nous avons mis comme base de l'identification des paramètres à partir d'une trajectoire donnée un schéma de coopération analyse – synthèse d'images. L'algorithme d'identification analyse les données d'une trajectoire en fonction desquelles il essaie de déterminer les paramètres physiques des composants. Comme mesure de la précision d'identification il utilise la synthèse; il génère des données de synthèse qui se constituent dans une trajectoire de synthèse. Par comparaison entre cette trajectoire et l'originale nous évaluons la performance du modèle ainsi obtenu.

Le logiciel chargé de cette tâche complexe s'appelle T2M (trajectory to model) et il comporte plusieurs fichiers d'entre et de sortie nécessaires à la description des paramètres algorithme et aussi au paramétrage du modèle.

L'inverse de l'identification, la synthèse d'images pure est réalisée par un autre logiciel (complémentaire à T2M) qui s'appelle M2T (model to trajectory) qui, à partir d'un modèle donné décrit dans les fichiers de paramétrage du modèle, génère une animation.

Le schéma présente en dessous montre le fonctionnement de ces deux logiciels.

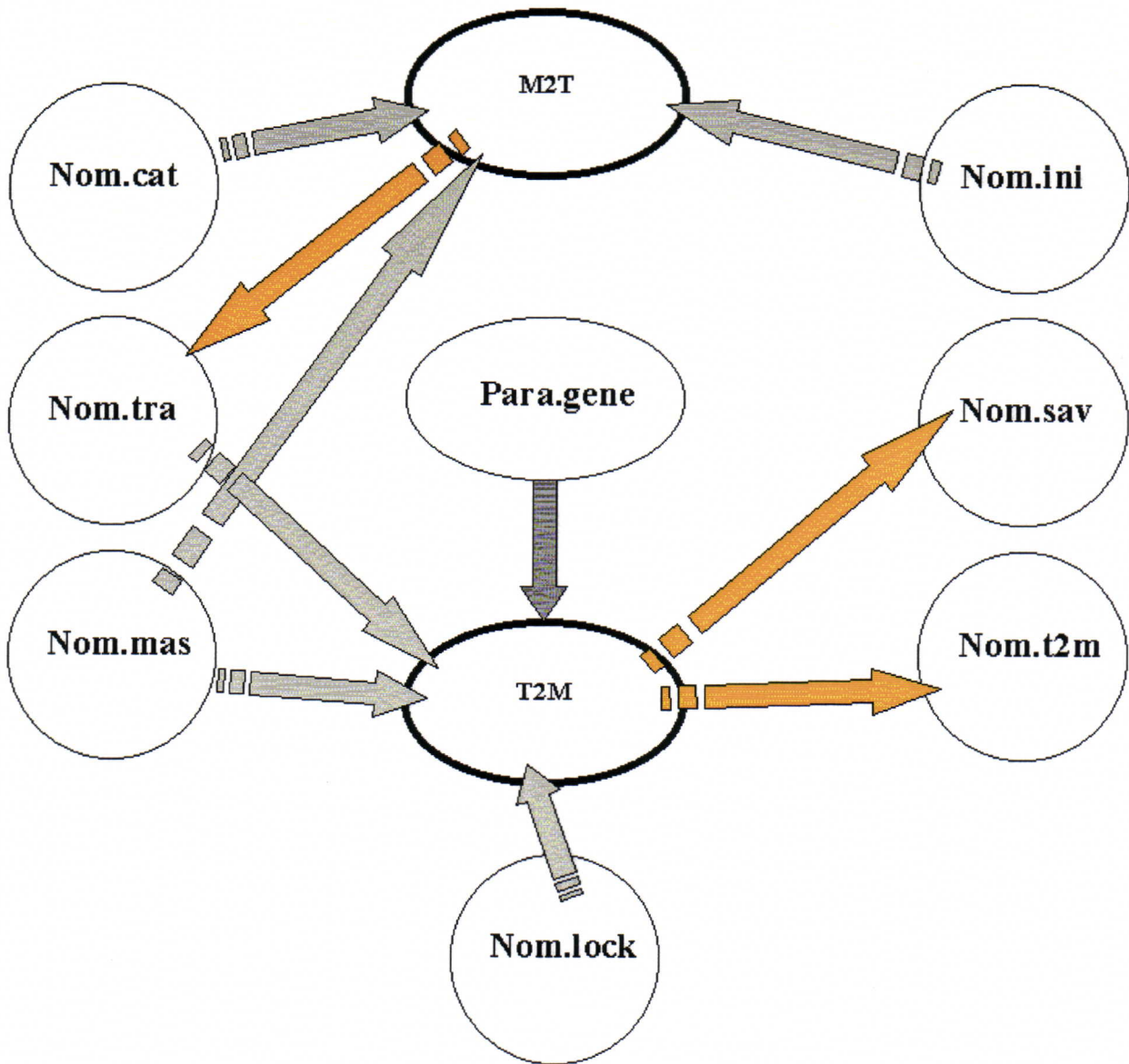


Figure A.1.1. : Schéma des logiciels de création et d'identification de modèles par des stratégies évolutives

Comme nous le remarquons plusieurs fichiers de configuration servent les deux moteurs T2M et M2T.

Le point commun de tous ces fichiers est le 'Nom' en fonction duquel les moteurs T2M et M2T savent quels sont les fichiers associés à un modèle ou trajectoire.

Le fichier Nom.tra est utilisé par le moteur T2M comme fichier d'entre contenant les données de la trajectoire en fonction de laquelle nous identifions le modèle physique. Le moteur M2T fabrique ce fichier lorsqu'il génère une trajectoire à partir d'un modèle donné.

La description de ce fichier et la suivante :

Nombre de particules:3

Nombre de pas de temps:10

Points de l'image 1

1.111100 1.111100 1.111100

-1.081800 -1.081800 -1.081800

1.133300 1.133300 -1.072700

Points de l'image 2

1.234545 1.234545 1.234545

-1.146738 -1.146738 -1.146738

1.305131 1.305131 -1.119604

Points de l'image 3

.....

Pour chaque particule dans chaque images nous précisions ces coordonnées x,y,z.

Le fichier Nom.mas est utilisé par les deux moteurs en tant que fichier d'entre ; il contient la description topologique du modèle, le nombre de particules avec leurs masses, les types des liaisons installées et le nombre de liaisons pour chaque type, leurs positions dans le modèle (entre telle et telle particule).

Nombre de particules:3

1

1

1

Nombre de liaisons unaires-position installees:0

Nombre de liaisons unaires-vitesse installees:0

Nombre de liaisons binaires-position installees:3

0 0 1

1 1 2

2 2 0

Nombre de liaisons binaires-vitesse installees:0

Nombre de liaisons ternaires-position installees:0

Le fichier Nom.cat est peut être le plus spécial des fichiers de configuration ; il contient le catalogue des types de liaisons. Grâce à la symétrie des objets plusieurs liaisons présentent les mêmes caractéristiques ce qui nous permet d'évoluer un seul type de liaison. Ce type sera après répliqué dans tous les endroits où une liaison de ce type intervient. Ce fichier est utilisé par M2T dans le processus de création d'une trajectoire.

Nombre de liaisons unaires-position:0

Nombre de liaisons unaires-vitesse:0

Nombre de liaisons binaires-position:3

1 0.1 0.1 0.1 0 0 0 0 0

2 0.2 0.2 0.2 0 0 0 0 0

3 0.3 0.3 0.3 0 0 0 0 0

Nombre de liaisons binaires-vitesse:0

Nombre de liaisons ternaires-position:0

Le fichier Nom.ini est utilisé par le moteur M2T au cas où la trajectoire que nous voulons générer doit respecter quelques conditions initiales. Ces conditions sont les vitesses initiales et les positions initiales pour chaque particule.

1 1 1 0.1 0.1 0.1

-1 -1 -1 -0.1 -0.1 -0.1

1 1 -1 0.1 0.1 -0.1

Le fichier Nom.sav est le fichier catalogue résultat de l'identification des paramètres par T2M, il représente le catalogue de types des liaisons du modèle.

Nombre de liaisons unaires-position:0

Nombre de liaisons unaires-vitesse:0

Nombre de liaisons binaires-position:3

4.181077 0.043035 1.098245 0.237325 0.205813 -0.079362 0.031257 0.688861 0.636491

10.688819 6.272265 0.772543 0.077617 0.042663 0.065201 0.017958 0.055136 0.490848

4.573826 0.007342 0.468789 0.424568 -0.100529 -0.131444 -0.068969 -0.057744 -0.173257

Nombre de liaisons binaires-vitesse:0

Nombre de liaisons ternaires-position:0

Le fichier Nom.t2m sauvegarde tout le parcours de l'identification. Il mémorise le processus d'identification avec tous les valeurs intermédiaires trouvées.

Le fichier Nom.lock sert à bloquer l'identification sur des éléments du catalogue que nous supposons connus à priori ou identifiés auparavant.

liaisons unaires position

liaisons unaires vitesse

liaisons binaires position

liaisons binaires vitesse

0 x x x 0

1 x x x 0

2 x x x 0

3 x x x 0

4 x x x 0

5 x x x 0

6 x x x 0

7 x x x 0

8 x x x 0

9 x x x 0

10 x x x 0

11 x x x 0

12 x x x 0

13 x x x 0

liaisons ternaires position

Enfin, le fichier Param.gene est utilisé par les deux moteurs M2T et T2M afin de paramétrer la stratégie évolutionnaire utilisée dans le processus d'identification.

generation=1000010

population=2000

pcross=15

pmut=1

reducer=0.01

noise=0

damping=0

Le programme utilise une heuristique pour initialiser les valeurs de longueur au repos (resp. cosinus d'angle au repos) des liaisons binaires (resp. ternaires) à partir des distances moyennes mesurées entre les extrémités de toutes leurs occurrences dans l'objet ("longueur moyenne observée"). Cette heuristique est utilisée principalement à l'initialisation de la population. Cependant l'expérience montre qu'il est utile de la réintroduire en doses très modérées parmi les différents types de mutations possibles.

Bien que l'algorithme impose que la fonction de coût du meilleur individu soit décroissante en fonction des générations, il arrive que l'erreur moyenne d'estimation, elle, remonte alors que la fonction de coût baisse. Cela arrive même lorsque l'identification s'effectue en l'absence de bruit sur les données cinématiques (trajectoires). Cela prouve notamment que l'erreur moyenne d'estimation, mesure "génotypique" de la distance entre deux modèles, n'est pas équivalente à la mesure "phénotypique" de la distance entre les comportements générés par ces deux modèles:

Peu avant la 400^{ème} génération, l'algorithme applique un processus d'optimisation locale par descente de gradient. Cette descente est appliquée au meilleur individu de la génération précédente. Le critère d'arrêt est l'insuffisance du taux de décroissance du coût entre deux pas consécutifs du calcul. Cette procédure est ensuite répétée toutes les 100 générations:

Il importe de ne pas déclencher prématurément cette procédure, car cela introduit dans l'évolution un optimum local très différent de l'optimum recherché, et d'une certaine façon,

leur le mécanisme d'évolution dont il appauvrit la diversité génétique et retarde la convergence. Nous n'avons pas trouvé de critère de déclenchement satisfaisant basé par exemple sur la valeur de fonction de coût atteinte, car celle-ci dépend du niveau de bruit appliqué aux trajectoires, paramètre que l'algorithme ignore. Le critère que nous avons finalement retenu est simplement celui du nombre de générations, qui semble fonctionner de manière indépendante à la fois du bruit, du nombre de particules et du nombre de liaisons à identifier.

A1.2 Présentation détaillée de l'algorithme

Identification évolutive des paramètres

A titre d'illustration, nous donnons ci-dessous de très courts extraits commentés du programme source. Ces extraits correspondent aux liaisons binaires; les manipulations concernant les liaisons ternaires et quaternaires sont identiques avec une complexité légèrement supérieure.

Initialisation

La population initiale est créée de la façon suivante:

A chaque liaison du catalogue, correspondent quatre gènes dans chaque individu:

le premier est la longueur au repos, initialisé à la valeur qui correspond à la distance moyenne mesurée sur l'ensemble des images, entre extrémités des instanciations de cette liaison;

le deuxième est la raideur, initialisée au hasard (approximation d'une répartition gaussienne logarithmique); les deux derniers sont les mutabilités des longueurs et des raideurs, initialisées au hasard de la même manière.

Ce qui précède correspond aux liaisons binaires; les liaisons ternaires et quaternaires sont initialisées de manière analogue.

Dans le cas de liaisons actives nous initialisons aussi les valeurs des poids des connexions entre 0 et 1, mais avec une probabilité décroissante vers 1.

En raison de l'heuristique choisie, les gènes de longueur initiaux sont ainsi uniformes au départ dans la population; la variété viendra par la suite du processus de mutation.

Mutations

Nous faisons coexister plusieurs types de mutations.

Mutation par réinjection:

Pour chaque mutation, on commence par choisir au hasard le type à modifier (c'est-à-dire la référence du catalogue). Ensuite, sont choisis deux individus au hasard: father avec une probabilité équirépartie dans la population, son avec une probabilité équirépartie dans la moitié la moins bien classée de la population. Tous les gènes de son sont effacés et remplacés par des gènes de father, sauf les 4 gènes correspondant au type choisi, qui sont remplacés par des valeurs heuristiques de manière identique à l'initialisation.

Mutation par héritage bruité:

Les individus father et son sont choisis comme précédemment. Les gènes de longueur et de raideur sont hérités de father et perturbés par un bruit d'écart-type égal à la mutabilité correspondante de father.

Un point important est qu'ici les mutabilités sont transmises sous le contrôle du coût global: chaque mutabilité de son est la moyenne entre celle de father et le résultat d'un tirage aléatoire d'écart-type proportionnel au coût de l'individu. Ce mécanisme permet l'adaptation des mutabilités (et donc de la précision de l'algorithme).

Mutation pure:

Ce processus de mutation crée des individus entièrement nouveaux.

Mutation locale:

On choisit une particule au hasard. Les individus father et son sont choisis aléatoirement: son dans la 2ème partie de la population du point de vue du coût global, father avec un biais en faveur des individus ayant la meilleure fonction de coût locale relatif à la particule choisie.

On choisit alors au hasard, une liaison dont la particule choisie soit l'une des extrémités. Les paramètres de cette liaison sur l'individu son sont remplacés par ceux de father, modifiés par

adjonction d'un bruit d'écart-type proportionnel à la mutabilité du paramètre et à la proportion de contribution du coût local par rapport au coût global de l'individu.

Croisement;

On choisit son au hasard dans la 2ème partie de la population, puis on parcourt l'ensemble des types de liaisons. Pour chaque type de liaison, on examine toutes les particules extrémités d'une instanciation de ce type de liaison; pour chacune de ces particules, on tire au hasard un père (father) plutôt performant en termes de fonction de coût locale associée à cette particule. On donne à son comme paramètres du gène correspondant au type courant de liaison, les moyennes des valeurs des paramètres homologues de tous les pères choisis au titre du type de liaison. On a ici une forme de croisement multi-parent.

Sélection

Le processus de sélection est normalement bien individualisé dans les stratégies d'évolution. Ici, il consiste en un tirage aléatoire dans la population, biaisé en faveur des individus les plus performants. En termes de programmation et uniquement pour des raisons de facilité d'écriture, le processus de sélection est scindé et réparti dans les différentes fonctions qui assurent les processus de mutation et de croisement.

Décadence

Cette opération consiste simplement à réduire exponentiellement les mutabilités des individus inchangés d'une génération à l'autre. Typiquement nous choisissons une réduction de 1% par génération. L'effet est antagoniste à celui de la "mutation par héritage bruité" et le complète.

Descente

Il s'agit ici d'une descente de gradients dont l'écriture est longue mais le principe simple (méthode de Powell): tester les dérivées partielles de la fonction de coût globale d'un individu par rapport à chaque paramètre, puis incrémenter le jeu de paramètres de façon à optimiser localement la fonction de coût, et gérer dynamiquement les incréments. Cette procédure est

appliquée toutes les 100 générations, typiquement à partir de la 400ème de manière à laisser à l'évolution pure un nombre de générations suffisant.

Evaluation

Nous avons testé plusieurs méthodes d'évaluation de la fonction de coût. En particulier, une de ces méthodes consiste à calculer la fonction de coût à partir du début de la trajectoire. Outre les inconvénients déjà évoqués concernant la "fonction de coût à long terme" (sensibilité aux conditions initiales...), tout le bénéfice des fonctions de coût locales est perdu car si le nombre N de pas de temps est suffisamment grand, la phase à l'instant t d'une particule dépend des phases de toutes les autres particules et des valeurs de toutes les liaisons à l'instant $t-N$. Nous avons effectué l'expérience, avec mutations et croisements faisant appel uniquement à la fonction de coût globale; les résultats de convergence sont encore inférieurs à ceux que l'on obtient avec une fonction de coût globale calculée à court terme. Nous avons aussi mis le nombre de pas de temps comme paramètre d'entrée de l'algorithme, et mis en évidence le fait que le calcul sur 1 pas de temps est le plus efficace.

Annexe 2

Complexité de calculs

L'identification des systèmes particules – liaisons est une tâche qui dépend fortement de la complexité des modèles. Le nombre des liaisons, la longueur des génomes et le nombre des images utilisées à identifier le modèle influencent beaucoup le processus d'identification. Chaque évaluation nécessite une simulation mécanique complète de la séquence d'images, avec tous les calculs inhérents des forces, positions, vitesses, etc.

Nous présentons quelques résultats sur trois expriments [CLL02]:

Séquence de test: Cheval 1. C'est un test sur le cheval décrit dans le chapitre 4.4. Il comporte 35 masses, 53 liaisons binaires (39 ressorts à 4 paramètres chacun et 14 amortisseurs à 2

paramètres chacun), 3 liaisons ternaires (2 paramètres chacune), au total résulte un chromosome d'une longueur de 3840 octets.

Longueur séquence: 53 images.

Génération: 1010.

Population: 1000.

Paramètres : 184.

Evaluation: 73.4 %.

Opérateurs génétiques: 26.6 %.

Séquence de test: Cheval 2. Test sur la même séquence d'images sur le cheval avec 35 masses, qu'avec des liaisons binaires, 39 ressorts à 4 paramètres chacun et 14 amortisseurs à 5 paramètres chacun, au total résulte un chromosome d'une longueur de 4464 octets.

Longueur séquence: 53 images.

Génération: 1010.

Population: 500.

Paramètres : 226.

Evaluation: 69.17 %.

Opérateurs génétiques: 30.83 %.

Malgré la complexité de la fonction de coût la partie réservée aux opérateurs génétiques reste assez importante, environ 30%.

Séquence de test: Muscle. Ce test utilise une séquence de synthèse générée par 3 particules sur 10 images, à l'aide de 2 liaisons actives ou muscles (4 connexions neuronales, 4 paramètres mécaniques).

Longueur séquence: 10 images.

Génération: 1010.

Population: 2000.

Paramètres : 24.

Evaluation: 11.03 %.

Opérateurs génétiques: 88.97 %.

Vue la taille du problème d'identification et la simplicité de la fonction de coût, la partie demandée par le processus d'évaluation descend à 11%.

Bibliographie

- [A88] B. Arnaldi, *Conception d'un noyau d'un système d'animation tridimensionnelle intégrant les lois de la mécanique*, Thèse de Doctorat, Université de Rennes I, juillet 1988.
- [AFS97] A. Agapie, F. Fagarasan, B. Stanculescu, *A Genetic Algorithm for a Fitting Problem*, Conference Proceedings 'The Fifth International Workshop on Software Engineering, Artificial Intelligence, Neural Nets, Genetic Algorithms, Symbolic Algebra and Automatic Calculation', Lausanne, Suisse, September 1996, special issue of 'Nuclear Instruments and Methods in Physics Research 1997.
- [AG85] W. W. Armstrong, M. W. Green, *The dynamics of articulated rigid bodies for purposes of animation*, Proc. of Graphics Interface, 1985 and The Visual Computer, Vol.1., n°5: 231-240, 1985.
- [B90] D. W. Brenner, *Empirical potential for hydrocarbons for use in simulating the chemical vapour deposition of diamond films*, Phys. Rev B, 42, 9458-9471, 1990.
- [B94] William Beaudot, *Le traitement neuronal de l'information dans la rétine des vertébrés: un creuset d'idées pour la vision artificielle*. Thèse de Doctorat, INPG, pp.109-159, décembre 1994.
- [B95] Thomas Bäck, *Evolution Strategies: an alternative evolutionary algorithm*. Journées Evolution Artificielle 95, Brest, septembre 1995.
- [B96] Samuel Boivin, *Recherche et développement d'un algorithme de rendu réaliste dans le cadre d'une coopération analyse-synthèse*, Image'Com96, Bordeaux, mai 1996.
- [BB71] J.J. Batter, F.P. Brooks, Jr., *GROPE-I - A computer display to the sense of feel*, Information Processing, Proc. IFIP Congress 71, 759-763, 1971.
- [BB82] Dana H. Ballard, Christopher M. Brown, *Computer Vision*, Prentice Hall, 1982.
- [BB88] R. R. Barzel, A. Barr, *A modelling system based on dynamic constraints*, Computer Graphics (Siggraph '88), août 1988.
- [BC95] E. Bouvier, E. Cohen, *Simulation of Human Flow with Particle System*, Simulators International XII, 27 (3), April 1995, pp 349-354.
- [BD95] D. Bechmann, N. Dubreuil, *Order-controlled free-form animation*, The Journal of Visualization and Computer Animation, vol.6, 11-32, 1995.
- [BFB94] J. L. Barron, D. J. Fleet, S. S. Beauchemin, *Performance of Optical Flow Techniques*, International Journal of Computer Vision, 12:1, 43-77, 1994.
- [BG96] E. Bouvier, P. Guilloteau, *Crowd simulation in Immersive Space Management*, 3rd Eurographics Workshop on Virtual Environments, Monte Carlo, 19th-20th February 1996.
- [BHW94] D. Breen, D. House, M. Wozny, *Predicting the drape of woven cloth using interacting particles*, Proc. Siggraph 94, Computer Graphics, 365-372, 1994.
- [BL01] A. Boumaza, J. Louchet, *Using Real-time evolution in Robotics*, EVOIASP2001, Artificial Evolution in Image Analysis and Signal Processing, avril 2001, Como, Italie.
- [BM96] Vincent Biragnet, Xavier Mousset, *rapport de stage ENSTA* (dirigé par X. Provot à l'INRIA), mars 1996.
- [BOBK90] F.P. Brooks, M. Ouh-Young, J.J. Batter, J. Kilpatrick, *Project GROPE - Haptic Displays for Scientific Visualization*. Computer Graphics, Vol. 24, N° 4, août 1990.
- [BS95] Thomas Bäck, Hans-Paul Schwefel: *Evolution Strategies I: Variants and their computational implementation*, Genetic Algorithms in Engineering and Computer Science, John Wiley & Sons.
- [C90] Claude Cadoz, *Connaître pour simuler, simuler pour connaître*, Coll. Mod. Phys., Grenoble, 1990.
- [C91] V. Cantoni, M. Ferretti, S. Levialdi, R. Negrini and S. Stefanelli, *Progress in Image Analysis and Processing*, World Scientific, 1991.
- [C94] Claude Cadoz, *Les réalités virtuelles*, Dominos/Flammarion, 1994
- [CB93] D. Cliff and S. Bullock, *Adding "Foveal Vision" to Wilson's Animat*, Pages 49-72, Adaptive Behavior, Volume 2, Number 1, Summer 1993.
- [CP95] J.-P. Cocquerez, S. Philipp et al. *Analyse d'images : filtrage et segmentation*, Masson, 1995
- [CL90] C. Cadoz, A. Luciani, *Modèles comportementaux et transducteurs gestuels rétroactifs*, Coll. Mod. Phys. Grenoble, 1990.
- [CLL02] P. Collet, J. Louchet, E. Lutton, *Issue on the Optimisation of Evolutionary Algorithms Code*, CEC2002, Honolulu, USA, May 12-17, 2002.
- [CLRS99] Pierre Collet, Evelyne Lutton, Frederic Raynal, Marc Schoenauer, *Individual GP: an Alternative Viewpoint for the Resolution of Complex Problems*, GECCO99, Orlando, Florida, July 1999.

- [CLSL00] P. Collet, E. Lutton, M. Schoenauer, J. Louchet, *Take it EASEA*, Parallel Problem Solving from Nature VI, vol 1917, Springer pp 891-901, Paris, September 2000.
- [CSBPSDPA94] J. Cassell, M. Steedman, N.I. Badler, C. Pelachaud, M. Stone, B. Doucville, S. Prevot, B. Achorn, *Modelling the interaction between speech and gesture*, Proceedings of the 16th annual conference of the Cognitive Science Society, Georgia Institute of Technology, Atlanta, USA, 1994 (url: web.mit.fr)
- [CYMT92] M. Carignan, Yang Ying, N. Magnenat-Thalmann, D. Thalmann, *Dressing Animated Synthetic Actors with Complex Deformable Clothes*, Proc. Siggraph 92, Computer Graphics, Vol. 26, n° 2: 99-104, 1992.
- [DLC93] Yves Delnondedieu, Annie Luciani, Claude Cadoz: *Physical Elementary Component for Modelling The Sensory-Motricity: The Primary Muscle*, 4th Eurographics Workshop on Animation and Simulation, pp.193-207, Barcelona September 1993
- [DGC96] Mathieu Desbrun, Marie-Paule Gascuel-Cani *Smoothed Particles: A New Approach for Animated Highly Deformable Bodies*, 7th Eurographics Workshop on Animation and Simulation, Poitiers, France, septembre 1996.
- [DKT95] Oliver Deussen, Leif Kobbelt, Peter Tücke, *Using Simulated Annealing to obtain good nodal approximations of deformable bodies*, Computer Animation and Simulation '95, Proc. of the Eurographics Workshop in Maastricht, The Netherlands, Springer: 30-43, 1995.
- [DP91] T. E. Davis, J. C. Principe, *A simulated annealing-like convergence theory for the simple genetic algorithm*, Proc. 4th Intl. Conf. on Genetic Algorithms: 174-182, 1991.
- [DZ93] P. Dworkin, D. Zeltzer, *A new model for efficient dynamic simulation*, 4th Eurographics Animation and Simulation Workshop: 135-148, 1993.
- [EK95] R.C. Eberhart, J. Kennedy, *A new Optimiser Using Particles Swarm Theory*, Proc. 6th Int. Symposium on Micro Machine and Human Science, Nagoya (Japon), IEEE service Centre, Piscataway, NJ, 39-43, 1995
- [EASEA] (EAsy Specification of Evolutionary Algorithms) home page: <http://www-rocq.inria.fr/EASEA/>
- [EO] (Evolutionary Objects) home page: <http://geneura.ugr.es/~jmerelo/EO.html>
- [Galib] home page: <http://lancet.mit.edu/ga/>
- [F78] J.L. Florens, *Coupleur gestuel rétroactif pour la commande et le contrôle de sons synthétisés*, Thèse INPG, Grenoble, 1978.
- [F83] R. Featherstone, *The calculation of robots using articulated body inertia*, Int. Journal of Robotics Research, 2 (1): 13-30, 1983.
- [FC84] J.L. Florens, C. Cadoz, *Geste instrumental, ordinateur, transducteurs gestuels rétroactifs - 10ème ICMC: 59-61, IRCAM, 1984.*
- [Fo84] J. A. Fodor, *The modularity of mind*, MIT Press, 1984.
- [G66] J.J. Gibson, *The senses considered as perceptual systems*, Boston, Houghton Mifflin Company, 1966.
- [G80] D. B. Gennery, *Modelling the Environment of an Exploring Vehicle by means of Stereo Vision*, PhD thesis, Stanford University, June 1980
- [G89] D. A. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.
- [G95] P. Gaussier. *Autonomous Robots interacting with an unknown world*, Special Issue on Animat Approach to Control, Robotics and Autonomous Systems, 16, 1995.
- [GV89] M.-P. Gascuel, A. Verroust, *Animation à l'aide de la dynamique: état de l'art*. Rapport de recherche du LIENS, 89(5), mai 1989.
- [GVP90] M.-P. Gascuel, A. Verroust, C. Puech *Animation with collisions of deformable articulated bodies*, Eurographics Workshop on Animation & Simulation, septembre 1990.
- [GVP91] M.-P. Gascuel, A. Verroust, C. Puech, *A modelling system for complex deformable bodies suited to animation and collision processing*. J. of Visualization and Computer Animation, 2 (3), août 1991.
- [GW92] R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, Wiley, 1992.
- [H62] P. V. C. Hough, *Method and Means of Recognising Complex Patterns*, U.S. Patent n°3 069 654, 18 December 1962.
- [H75] J. H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor, Univ. of Michigan Press, 1975.
- [H80] R. M. Haralick, *Using Perspective Transformations in Scene Analysis*, Computer Graphics and Image Processing 13, 1980, pp. 191-221.
- [H83] Tom S. Huang, *Image Sequence Processing and Dynamic Scene Analysis*, Springer Verlag, 1983.
- [H97] A. Habibi, *Modélisation d'objets très déformables*, thèse de Doctorat de l'INPG, Grenoble, janvier 1997.
- [HB92] Hoffmeister F. and Bäck T., *Genetic Self-learning*, Proceedings of the 1st European Conference on Artificial Life, 227-235, 1992.

- [HBG92] D.H. House, D.E. Breen, P.H. Getto: *On the Dynamic Simulation of Physically-Based Particle-System Models*, Proceedings of EuroGraphics'92 Workshop on Animation and Simulation, Cambridge England, 5-6 September 1992.
- [HS81] B. K. P. Horn, E. J. Weldon Jr., *Determining Optical Flow*, Artificial Intelligence 17: 185-204, 1981.
- [H91] M. I. Heggie, *Semiclassical interatomic potential for carbon and its application to the self-interstitial in graphite*, J.Phys.: Condens. Matter 3, 3065-3079, 1991.
- [H96] Dave Hutchinson, *Adaptive Refinement for Mass/Spring Simulations*, Eurographics Workshop on Animation & Simulation, 31-45, septembre 1996, Poitiers.
- [K92] J. R. Koza, *Genetic Programming*, MIT Press 1992.
- [KN99] M. Köppen, B. Nickolay, *Design of Image Exploring Agent using Genetic Programming*, Fuzzy Sets and Systems, Special Issue on Softcomputing, 103 (1999) 303-315.
- [KU99] M. Komosinski, S. Ulatowski, *Framsticks: towards a simulation of a nature-like world, creatures and evolution*, Lecture Notes in Artificial Intelligence 1674 pp.261-265, Springer-Verlag
- [IP94] Jim Ivins, John Porrill, *Statistical Snakes: Active Region Models*, British Machine Vision Conference, York, BMVA Press, 377-386, septembre 1994.
- [J73] Gunnar Johansson, *Visual perception of biological motion and a model for its analysis*, Perception and Psychophysics, vol. 14, n°2, 201-211, 1973.
- [J93] Stéphane Jimenez, *Modélisation et simulation d'objets volumiques déformables complexes*, thèse de Doctorat, INPG Grenoble, novembre 1993.
- [J96] Jiang Li, *Identification de modèles d'animation d'écoulements turbulents*, rapport de DEA IARFA, Université Paris6/Institut Blaise Pascal et ENSTA/Laboratoire d'électronique et d'informatique, juillet 1996.
- [JLHTØE92] R. Jones, C. D. Latham, M. I. Heggie, V. Torres, S. Øberg, S. Estreicher, *Contraction of Buckyballs upon Charging*, Phil. Mag., 1992.
- [Jo94] A. Joukhadar, C. Laugier, *Dynamic modelling of rigid and deformable objects for Robotics tasks: motion, deformations and collisions*, Int. Conf. ORIA, Marseille, décembre 1994.
- [Jo95] A. Joukhadar, *Modèles physiques et manipulation dextre*, 3èmes journées du groupe de travail Réalité Virtuelle, Grenoble, février 1995.
- [L01] J. Louchet, *Using an Individual Evolution Strategy for Stereovision*, Genetic Programming and Evolvable Machines Journal, Kluwer (à paraître en février 2001).
- [L01b] A. Luciani, Communication personnelle, Groupe de travail Animation- Simulation, Grenoble, février 2001.
- [L94] J. Louchet, *An Evolutionary Algorithm for Physical Motion Analysis*, British Machine Vision Conference, York, Sep. 1994, 701-710, BMVA Press, 1994.
- [L94b] J. Louchet, *Identification de modèles physiques d'animation*, Journées Infographie Interactive et Intelligence Artificielle, Limoges, pp. 9-20, avril 1994.
- [L95] J. Louchet, *Identification évolutive de modèles physiques pour la synthèse d'images animées*, Evolution Artificielle 94, Toulouse, 20-23 septembre 1994, éd. Cepadues, 1995.
- [L96] J. Louchet, *Self-Adaptive Evolution to identify Structure from Motion*, Image'Com96, Bordeaux, mai 1996.
- [LBCP96] J. Louchet, M. Boccara, D. Crochemore, X. Provot, *Building new tools for synthetic image animation using evolutionary techniques*, Evolution Artificielle/Artificial Evolution 95, Brest, Sep. 95, Springer Lecture Notes on Computer Science, 1996.
- [LC84] A. Luciani, C. Cadoz, *Modélisation et animation gestuelle d'objets - Le système ANIMA*, CESTA - 1er Colloque Image, Biarritz, 1984.
- [LCLSF00] E. Lutton, P. Collet, J. Louchet, M. Sebag, C. Fonlupt, *Evolution Artificielle*, cours polycopié ENSTA, mars 2000.
- [LGL96] J. Louchet, A. Gagalowicz, A. Luciani, *De l'analyse-synthèse d'images à l'identification de modèles physiques particuliers: vers une méthodologie de construction de simulateurs*, Revue Scientifique et Technique de Défense, 119-127, 1997.
- [LFU94] A. Luciani, J. L. Florens, C. Uhl, "Le simulateur temps réel CORDIS-ANIMA", 3ème séminaire Animation-Simulation, Lille, octobre 1994.
- [LJ96] J. Louchet, L. Jiang, *An Identification Tool to build Physical Models for Virtual Reality*, International Workshop on Signal and Image Processing IWISP96, Manchester, novembre 1996, Elsevier NV, pp. 669-672.
- [LP98] W. B. Langdon and R. Poli, *Why Ants are Hard*, Technical Report: CSRP-98-4, January 1998, <http://www.cs.bham.ac.uk/wbl>.
- [LPC95] J. Louchet, X. Provot, D. Crochemore, *Evolutionary identification of cloth animation models*, Eurographics Workshop on Animation and Simulation, Maastricht, septembre 1995.

- [LHVD95] A. Luciani, A. Habibi, A. Vapillon, Y. Duroc, *A Physical Model of Turbulent Fluids*, Eurographics Workshop on Animation and Simulation, Maastricht, 16-29, septembre 1995.
- [LJFCR91] A. Luciani, S. Jimenez, J.-L. Florens, C. Cadoz, O. Raoult, *Computational Physics: a modeller simulator for animated physical objects*, Proc. Eurographics Conference, Wien, Austria, Elsevier, septembre 1991.
- [LM94] Evelyne Lutton, Patrice Martinez, *A Genetic Algorithm for the Detection of 3D Geometric Primitives in Images*, 12th ICPR, Jerusalem, Israel, October 9-13, 1994 / INRIA technical report # 2210.
- [Lo94] Jean Lorenceau *Rôle des informations locales dans la segmentation et l'intégration de mouvements visuels*, journées mathématiques de la vision, Rennes, 10-11 février 1994.
- [LTC94] A. Lanitis, C. C. J. Taylor, T. F. Cootes, *An Automatic Face Identification System Using Flexible Appearance Models*, British Machine Vision Conference, York, BMVA Press, 65-74, 1994.
- [Lu93] E. Lutton, *Etat de l'art des Algorithmes Génétiques*, rapport d'expertise SGDN, décembre 1993.
- [M94] M. Millonas, *Swarms, phase transitions and collective intelligence*, Artificial Life III (ed. C.G Langton), Santa Fe Institute Studies in the Sciences of Complexity, Vol. XVII, Addison Wesley: Reading, MA, 1994.
- [M99] S. Marukatat, *La Construction Automatique de modèle particulière à l'aide d'une stratégie d'évolution*, Rapport stage DEA, LIP6/Université Pierre et Marie Curie, septembre 1999.
- [Ma82] David Marr, *Vision*, Freeman, 1982.
- [MP62] M. J. P. Musgrave, J. A. Pople, Proc. R. Soc. A 268, 474-484, 1962.
- [MZ90] M. McKenna, D. Zeltzer, *Dynamic Simulation of autonomous legged motions*, Computer Graphics (Siggraph 90), 24 (4): 29-38, 1990.
- [N95] F. Neyret, *Animated texels*, Computer Animation and Simulation '95, Proc. of the Eurographics Workshop, Maastricht, Springer, septembre 1995.
- [NBB95] P. Nesi, A. del Bimbo, D. Ben-Tzvi, *A Robust Algorithm for Optical Flow Estimation*, Computer Vision and Image Understanding, vol.62, N°1, July, 59-68, 1995.
- [NM93] J. Thomas Ngo, Joe Marks, *Spacetime constraints revisited*, proc. Siggraph 93, Computer Graphics, 343-350, 1993.
- [OSM93] Agata Opalach-Szwerbel, Steve Maddock, *Implicit surfaces: Appearance, Blending and Consistency*, Fourth Eurographics Animation & Simulation Workshop, Barcelona, septembre 1993.
- [O94] C.W.A.M. van Overveld, *A simple approximation to rigid body dynamics*, The journal of visualisation and computer animation, Vol. 5 n°1, 17-36, Wiley, 1994.
- [OB95] Kees van Overveld, Bart Barenbrug, *All you need is force: a constraint-based approach for rigid body dynamics in computer animation*, Computer Animation and Simulation 95 (Eurographics Workshop on Animation and Simulation), Springer, 80-94, septembre 1995.
- [O'Regan94] Kevin O'Regan, *Invariances et stabilité du monde visuel*, journées mathématiques de la vision, Rennes, 10-11 février 1994.
- [OR81] John O'Rourke, *Motion Detection using Hough technique*, IEEE conference on Pattern Recognition and Image Processing, Dallas 1981, pp. 82-87.
- [OS94] I. Opris, B. Stanculescu, *Continuos-Time Dynamics of Associative Memory*, Conference Proceedings of „National Physics Conference, Sibiu, Romania, September 1994, pp.150.
- [Pe90] Jean Petitot, *Remarques sur la vision*, Revue de synthèse: IVE s. n° 1-2, janvier-juin 1990.
- [P95] X. Provot, *“Deformation Constraints in a Mass-Spring Model to describe Rigid Cloth behaviour”*, Graphics Interface 1995, Québec, avril 1995.
- [PF93] M. van de Panne, E. Fiume, *Sensor-Actuator networks*, Computer Graphics Proceedings, Annual Conf. Series, 1993.
- [PKF94] M; van de Panne, Ryan Kim, Eugene Fiume, *Virtual Wind-up Toys for Animation*, Graphics Interface 1994.
- [R87] Craig Reynolds, *Flocks, Herds and Schools: A Distributed Behavioural Model*, Computer Graphics (Siggraph) Vol. 21, n° 4, 25-34, juillet 1987.
- [R94] I. Rechenberg, *Evolution Strategy*, in J.M. Zurada, R.J. Marks II, C.J. Robinson, Computational Intelligence imitating life, IEEE Press: Piscataway, NJ, 147-159, 1994.
- [Re83] W. T. Reeves, *Particle Systems - A Technique for Modelling a Class of Fuzzy Objects*, Computer Graphics (Siggraph) vol. 17 n° 3, 359-376, juillet 1983.
- [Rec94] Rechenberg I., *Evolutionsstrategie '94*, volume 1 of Werkstatt Bionik und Evolutionstechnik, Frohmann-Holzboog, Stuttgart 1994.
- [Reeves83] W.T.Reeves *Particle Systems- A Technique for Modelling a Class of Fuzzy Objects*, Computer Graphics (Siggraph) vol17 no.3, pp 359-376, juillet 1983.
- [Ri90] J.-C. Risset, *Modèles physiques et perception*, Coll.Mod.Phys, Grenoble, 1990.

- [RL92] G. Roth and M. D. Levine, *Geometric Primitive Extraction using a Genetic Algorithm*, IEEE CVPR Conference, pp. 640-644, 1992.
- [Rud92a] Rudolph G., *On correlated mutations in evolution strategies*, 105-144, Männer and Manderick 1992.
- [S94] Karl Sims, *Evolving Virtual Creatures*, Computer Graphics 1994.
- [S95] Nicolas Szilas, *Apprentissage dans les réseaux récurrents pour la modélisation mécanique et étude de leurs interactions avec l'environnement*, thèse de doctorat, INPG, décembre 1995.
- [S96] B. Stanculescu, *Using Deterministic Neural Networks for Ultrasonic Flaw Classification*, Conference Proceedings 'Fourth European Congress of Intelligent Techniques', Aachen, Germany, Septembre 1996, vol.2 - Data Analysis And Signal Processing, pp. 1505.
- [S99] P.K. Ser, S. Clifford, T. Choy, W.C. Siu, *Genetic Algorithm for the Extraction of Nonanalytic Objects from Multiple Dimensional Parameter Space*, Computer Vision and Image Understanding, vol. 73 no. 1, Academic Press: Orlando, FL, pp. 1-13, 1999.
- [SC93] N. Szilas, C. Cadoz, *Physical Models That Learn*, International Computer Music Conference, Tokyo, 1993.
- [Sch77] Schwefel H.-P., *Numerische Optimierung von Computer-Modellen-mittels der Evolutionsstrategie*, volume 26 of Interdisciplinary Systems Research, Basel 1977.
- [Sch95] Schwefel H.-P., *Evolution and Optimum Seeking*, Sixth-Generation Computer Technology Series, New York 1995.
- [SE97] Ralf Salomon and Peter Eggenberger, *Adaptation on the Evolutionary Time Scale: a Working Hypothesis and Basic Experiments*, Third European Conference on Artificial Evolution, Nîmes, France, October 1997, Springer Lecture Notes on Computer Science no. 1363, pp. 251-262.
- [SFL01] B. Stanculescu, J.L. Florens, J. Louchet, A. Luciani, *Evolving Motricity in Particle-Based Models*, Conference Proceedings ACIVS'2001 – Advanced Concepts For Intelligent Vision Systems, august 2001, Baden-Baden, Allemagne, pp. 6-12.
- [SHARP94] *Rapport d'activité 1994, programme 4: Robotique image et vision, projet Sharp*, 271-275, INRIA 1994.
- [SL00] B. Stanculescu, J. Louchet, *Evolutionary Identification of Active Particle Systems*, WSCG2000, International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media, Plzen, février 2000.
- [SL00b] B. Stanculescu, J. Louchet, *Evolving Physical Models to Understand Motion in Image Sequences*, European Symposium on Intelligent Techniques ESIT'2000, September 14-15, Aachen, Germany.
- [SL99] B. Stanculescu, J. Louchet, *Evolutionary Identification of Active Particle Systems*, AFIG99, Reims.
- [ST92] Szeliski & Tonnesen, *Surface modelling with oriented particle systems*, Computer Graphics, proc. Siggraph'92, 165-194, 1992.
- [T88] J. Tersoff, *New empirical approach for the structure and energy of covalent systems*, Phys.Rev.B, vol 37, n°12, 1988.
- [T88a] J. Tersoff, *Empirical interatomic potential for silicon with improved elastic properties*, Phys.Rev.B, vol 38, n°14, 1988.
- [T90] V. Torres, *Interatomic potentials and defect modelling in semiconductors*, PhD thesis, King's College London, 1990.
- [T97] Patrick Teisserenc: *Identification de modèles d'animation de la locomotion équine*, ENSTA Research Report, 1997.
- [T98] C. K. Tang, G. Medioni, *Integrated Surface, Curve and Junction Inference from Sparse 3-D Data Sets*, Proc. ICCV98, Bombay, IEEE Computer Society Press: Piscataway, NJ., pp. 818-823, 1998.
- [TNM95] D. Tang, J. T. Ngo, J. Marks, *N-body spacetime constraints*, journal of visualization and computer animation, vol.6, 143-154, 1995.
- [TPBF87] D. Terzopoulos, J. Platt, A. Barr, K. Fleischer, *Elastically Deformable Models*, Proc. Siggraph 87, Computer Graphics, Vol. 21, n° 4, 205-214, 1987.
- [TT] X. Tu, D. Terzopoulos, *Artificial Fishes, Physics, Locomotion, Perception, Behaviour*, Computer Graphics 1994.
- [ULF95] C. Uhl, A. Luciani, J.-L. Florens, *Hardware Architecture of a real-time simulator for the Cordis-Anima system*, Computer Graphics International, Leeds, juin 1995.
- [WB95] J. Wilhelms, B. Barsky, *Using Dynamic Analysis to animate articulated bodies as humans and robots*, Graphics Interface'85, 97-104, mai 1985.
- [Z95] M. Zyda, *Using Spread Spectrum Ranging Techniques for Position Tracking in a Virtual Environment*, 2nd IEEE workshop on Networked Realities NR95, Boston, 26-28 octobre 1995.

TITLE: Physical model and controller identification in animation

KEY-WORDS: Image Synthesis, Computer animation, Motion analysis, Physical models, Algorithmical Physics, Motricity, Model identification, Evolutionary strategies, Neural Networks

ABSTRACT

Particle-based mass-bond models are well adapted to motion modelling and image animation. A practical limitation comes from the use of purely visco-elastic bonds which only allow passive, energetically dissipative behaviour. Another difficulty is how to build and adjust such a model in order to ensure realistic synthetic motion.

The main goal of this thesis is to extend the domain of applicability of particle-bond models through the introduction of motricity. To this end two variants of active bonds ("muscles.") have been introduced, their theoretical properties, connections with ACROE's FGT (force-gesture transducer) and proprioception are discussed.

The first type of muscle uses a central control system designed as an independent physical system with a quadripole structure. It is demonstrated through a target-following mobile system, where the controller is connected to the mobile through visual-physical feedback.

The second type of muscle uses a local neural controller.

Identifying a mass-bond system containing either type of muscle is achieved by an evolution strategy which both optimises control parameters and mechanical parameters, through minimisation of a mathematical distance between the reference trajectory and the trajectory predicted by the model. Self-adaptation properties allow reducing the user's involvement into the identification procedure. Experimental test results on synthetic and real-world data are presented.
