HAL
open science

Une approche CAO pour la préconception des mécanismes plans générateurs de trajectoire : RéalisMe

Adrian Vasiliu

To cite this version:

Adrian Vasiliu. Une approche CAO pour la préconception des mécanismes plans générateurs de trajectoire : RéalisMe. Sciences de l'ingénieur [physics]. Ecole Centrale Paris, 1997. Français. NNT : . tel-00393842

HAL Id: tel-00393842

https://theses.hal.science/tel-00393842v1

Submitted on 10 Jun 2009

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.

CENTRALE
PARIS

ECOLE CENTRALE PARIS
THESE DE DOCTORAT

Spécialité
Productique - CAO - Mécanique

Présentée par
Adrian Alexandru VASILIU

le 7 février 1997
pour l'obtention du
GRADE DE DOCTEUR

Une approche CAO pour la préconception des mécanismes plans générateurs de trajectoire : REALISME

Composition du jury :

Président	Jorge ANGELES	Professeur, Université McGill, Montréal
Rapporteurs	Patrick CHEDMAIL	Professeur, Ecole Centrale de Nantes
	Claude BOHATIER	Professeur, Université de Montpellier II
Examinateur	Jean-Pierre MERLET	Directeur de recherche, INRIA, Sophia Antipolis
Directeur de thèse	Jean-Claude BOCQUET	Professeur, Ecole Centrale Paris
Co-directeur de thèse	Bernard YANNOU	Maître de Conférences, Ecole Centrale Paris

Laboratoire Productique Logistique
Ecole Centrale Paris
Grande Voie des Vignes
92 925 Châtenay-Malabry Cedex

Table des matières

Remerciements **xvii**

1 Introduction **1**
1.1 Domaine 1
1.2 Objectifs 3
1.3 REALISME : une plateforme intégrée de préconception 4
1.4 Organisation 5

A Etat de l'art et analyse de la problématique **7**

2 Préconception de mécanismes plans : état de l'art **9**
2.1 Modèles de mécanismes plans 9
2.1.1 Modélisation par graphes 10
2.1.2 Modélisation par groupes d'Assur 12
2.2 Analyse cinématique 14
2.2.1 Méthodes spécifiques à un type de mécanisme 14
2.2.2 Méthodes générales 14
2.2.3 Approches basées sur les contraintes géométriques 16
2.2.4 Approches hybrides 17
2.2.5 Logiciels commercialisés 17
2.3 Synthèse morphologique 18
2.3.1 Approches basées sur les graphes 18
2.3.2 Systèmes experts et à base de connaissances 20
2.4 Synthèse dimensionnelle (cinématique) 20
2.4.1 Méthodes graphiques 20
2.4.2 Méthodes analytiques 21
2.4.3 Méthodes d'optimisation 22
2.4.4 Logiciels 27
2.4.5 Systèmes experts et à base de connaissances 28
2.4.6 Théorie de la courbure de trajectoire 28
2.4.7 Catalogues de trajectoires 29
2.5 L'apport de l'Intelligence Artificielle 30
2.5.1 Cinématique qualitative 30
2.5.2 Raisonnement à base de cas 32
2.5.3 Réseaux neuronaux artificiels 33

3 Problématique, typologie des approches, approches adoptées **37**
3.1 Cahier des charges fonctionnel de la thèse 37
3.2 Analyse versus synthèse 42
3.3 Analyser 43
3.4 Concevoir de manière incrémentale 44
3.5 Synthétiser par composition de briques fonctionnelles 44
3.6 Synthétiser à partir d'une compilation de la connaissance 46
3.6.1 Principe 46
3.6.2 Le remplissage de la base de cas 47
3.6.3 Raisonnement à base de cas élémentaires 48
3.6.4 Raisonnement à base de familles de cas 48
3.6.5 Raisonnement à base de cas par réseaux neuronaux 49
3.7 Synthétiser par optimisation 50
3.7.1 Optimisation dimensionnelle exacte (l'approche classique) 50
3.7.2 Optimisation dimensionnelle basée sur la cinématique qualitative 51
3.8 Conclusion 52

B Modèles de cinématique pour la préconception **53**

4 Modèle systémique des mécanismes plans : les groupes multipolaires **55**
4.1 L'approche systémique 55
4.1.1 Sous-systèmes élémentaires : les groupes multipolaires 55
4.1.2 La construction modulaire des mécanismes plans 58
4.2 Fonctions de transfert en cinématique 65
4.2.1 Exemple : la diade RRR 65
4.2.2 Exemple : le moto-élément 68
4.3 Fonctions de transfert en dynamique inverse 68
4.3.1 Exemple : la diade RRR 69
4.4 Simulation cinématique 69
4.4.1 Algorithme 70
4.4.2 Avantages 70
4.5 Simulation dynamique inverse 72
4.6 Modélisation informatique 72
4.7 Conclusion 73

5 Modèle qualitatif de cinématique **75**
5.1 Introduction 75
5.2 La trajectoire qualitative 76
5.3 Utilisation de la décomposition systémique multipolaire 78
5.4 La simulation cinématique qualitative 79
5.4.1 Une approche basée sur les Centres Instantanés de Rotation 79
5.4.2 Courbure des arcs qualitatifs, base et roulante 81
5.4.3 Propagation d'arcs de cercle le long du mécanisme 81
5.5 Vers une compréhension qualitative du mécanisme 83
5.6 Comparaison avec l'approche de Shrobe 84
5.7 Conclusion 84

C Méthodologies de préconception 85

6 Conception incrémentale : cycles modification/analyse 87
6.1 Introduction . . . 87
6.2 Construction et modification du mécanisme . . . 87
6.2.1 Morphologie . . . 87
6.2.2 Dimensions . . . 89
6.3 Spécification de la trajectoire désirée . . . 91
6.3.1 Spécification avec indépendance du temps . . . 91
6.3.2 Spécification avec dépendance du temps . . . 92
6.4 Simulation cinématique . . . 93
6.4.1 Lois de mouvement des moteurs . . . 93
6.4.2 Paramètres de la simulation . . . 93
6.4.3 Réduction du temps de mise en équation et des calculs itératifs . . . 94
6.4.4 Identification des causes des blocages et des positions singulières . . . 94
6.5 « Chiffrer » la distance entre les trajectoires désirée et générée . . . 96
6.5.1 Distance exacte . . . 96
6.5.2 Distance qualitative . . . 97

7 Un outil pour la synthèse morphologique : la représentation de graphes 99
7.1 Les graphes et la conception de mécanismes . . . 99
7.2 Visualisation des graphes : le problème du dépliage . . . 100
7.2.1 Notions utiles de la théorie des graphes . . . 100
7.2.2 L'approche numérique . . . 102
7.2.3 L'approche topologique de Chieng et Hoeltzel . . . 102
7.3 Notre approche topologique pour l'optimisation . . . 103
7.3.1 Notre fonction coût . . . 103
7.3.2 Modifications topologiques élémentaires . . . 104
7.3.3 Algorithme d'optimisation topologique . . . 106
7.3.4 Algorithmes d'affichage . . . 108
7.3.5 Prise en compte des particularités des graphes . . . 110
7.4 Exemples . . . 111
7.5 Conclusion et débouchés . . . 111

8 Synthèse dimensionnelle : l'approche de l'optimisation 115
8.1 Introduction . . . 115
8.2 Approche d'optimisation dimensionnelle exacte . . . 115
8.2.1 Fonction objectif . . . 115
8.2.2 Variables d'optimisation . . . 118
8.2.3 Contraintes d'environnement . . . 120
8.2.4 Algorithme d'optimisation . . . 120
8.2.5 Accélération de la convergence : le recalage géométrique . . . 121
8.2.6 Exemple . . . 122
8.3 Approche d'optimisation dimensionnelle qualitative . . . 124
8.3.1 Fonction objectif . . . 124
8.3.2 Variables d'optimisation . . . 126
8.3.3 Algorithmes d'optimisation . . . 126
8.3.4 Recalage géométrique . . . 126
8.4 Conclusion . . . 126

9 Synthèse par capitalisation de cas : l'approche par réseaux neuronaux artificiels 129
9.1 Introduction . . . 129
9.2 Une approche par approximation de fonction . . . 131
9.3 Modélisation et indexation des cas . . . 133
9.3.1 Définition du cas . . . 133
9.3.2 Méthode d'indexation des cas . . . 133
9.4 Indexation des trajectoires . . . 134
9.4.1 Prise en compte de la forme : l'approche de Kota et Chiou . . . 134
9.4.2 L'approche de McGarva . . . 135
9.4.3 L'analyse en série de Fourier (calcul des coefficients) . . . 136
9.4.4 Normalisation des coefficients de Fourier représentant les trajectoires . . . 138
9.5 Stockage des morphologies et des dimensions . . . 144
9.5.1 Morphologies . . . 144
9.5.2 Dimensions . . . 144
9.6 Création de la base de cas . . . 146
9.6.1 Méthode de génération aléatoire . . . 146
9.6.2 Le réseau neuronal . . . 148
9.6.3 Intégration du réseau neuronal dans la plateforme de préconception . . . 151
9.6.4 Comparaison avec la méthode de McGarva . . . 151
9.7 Utilisation de la base de cas . . . 152
9.7.1 Méthode . . . 152
9.7.2 Description du cas : spécification de la forme de trajectoire désirée . . . 152
9.7.3 Indexation du cas . . . 153
9.7.4 Obtention de la solution . . . 153
9.8 Procédures de validation . . . 155
9.8.1 Validation pour une solution exacte connue . . . 156
9.8.2 Validation pour une borne minimum de l'erreur connue . . . 159
9.9 Conclusion . . . 160

10 Synthèse basée sur les cas : l'approche par trajectoires qualitatives 163
10.1 Introduction . . . 163
10.2 Modélisation et indexation des cas . . . 164
10.2.1 Définition du cas qualitatif . . . 164
10.2.2 Méthode d'indexation . . . 166
10.3 Indexation des trajectoires . . . 166
10.3.1 Paramétrage des arcs . . . 166
10.3.2 Normalisation de la trajectoire qualitative . . . 167
10.3.3 Détermination de la clé qualitative . . . 174
10.4 Stockage de la morphologie et des dimensions . . . 175
10.4.1 Morphologie . . . 175
10.4.2 Dimensions . . . 177
10.5 Construction de la base de cas . . . 177
10.6 Utilisation de la base de cas . . . 180
10.6.1 Méthode . . . 180
10.6.2 Description du cas : spécification de la forme de trajectoire désirée . . . 180
10.6.3 Calcul de l'index . . . 180
10.6.4 Obtention de la solution . . . 180
10.7 Conclusion . . . 181

D Conclusion et annexes 185

11 Conclusions et perspectives 187
11.1 Contributions de cette thèse 187
11.1.1 Modèles de cinématique 187
11.1.2 Méthodes de synthèse 188
11.2 Domaines d'application 191
11.3 Recherches futures 192

A Glossaire de termes et de concepts 195

B Le raisonnement à base de cas 199
B.1 Structure générique d'un système de Raisonnement à base de cas 199
B.2 Domaines d'applications 200
B.3 CBR en conception 201
B.3.1 Rôle du CBR en Conception 201
B.3.2 Fonctionnalités des systèmes de CBR en Conception 201
B.4 Sources d'information 201

C Présentation sommaire des réseaux neuronaux artificiels 203
C.1 Structure d'un RNA 204
C.2 Apprentissage 206
C.3 Approximation de fonction et généralisation 207
C.4 Déficiences de la rétropropagation standard 207
C.5 L'algorithme d'apprentissage *Rprop* 208

Bibliographie 209

Index 224

Résumé 225

Liste des tableaux

3.1 Comparaison des trois activités de conception : l'analyse, la conception incrémentale et la synthèse. 40
3.2 Croissance du nombre de chaînes cinématiques distinctes avec le nombre de solides composant le mécanisme. 43

4.1 Combinaisons possibles du nombre de solides et de liaisons dans les groupes multipolaires passifs. 57
4.2 Temps de résolution en position (en secondes, 360 pas de calcul). 71
4.3 Les trois types de graphes utilisés en REALISME. 72
4.4 Comparaison de REALISME avec les logiciels ADAMS et SDS. 73

7.1 Comparaison entre l'approche numérique, l'approche topologique de Chieng et Hoeltzel et l'approche topologique de REALISME 112

9.1 Coefficients de Fourier avant normalisation. 143
9.2 Coefficients de Fourier après normalisation. 143
9.3 Paramètres de normalisation pour la trajectoire générée. 154
9.4 Paramètres optimaux de recalage pour les trajectoires modélisées par des coefficients de Fourier. 155
9.5 Comparaison de la synthèse par catalogue de trajectoires exactes et par base de cas utilisant l'analyse de Fourier et l'interpolation avec un réseau neuronal. 161
9.6 Comparaison des deux approches de synthèse selon la taille de la base de cas. . . . 162

10.1 Transformations par symétrie pour chaque arc. 172
10.2 Génération de cas qualitatifs à partir d'une trajectoire exacte. 178
10.3 Recherche des cas les plus proches dans la base de cas qualitatifs. 180
10.4 Paramètres de normalisation d'une trajectoire qualitative. 181
10.5 Paramètres optimaux de recalage pour les trajectoires qualitatives. 181
10.6 Comparaison de la synthèse par CBR utilisant l'analyse harmonique et les réseaux neuronaux avec la synthèse par base de cas qualitatifs. 183

11.1 Tableau récapitulatif des domaines d'application des différentes méthodes utilisées en REALISME. 191

Table des figures

1.1 Exemples industriels d'utilisation du mécanisme quatre barres comme générateur de trajectoire. 2
1.2 Exemple de schéma cinématique d'un mécanisme articulé. 3

3.1 La « bête à cornes » du travail de thèse. 38
3.2 Détermination des fonctions de service d'une méthodologie de préconception des générateurs de trajectoire (par la « pieuvre » de la méthode *APTE*). 39
3.3 Analyse fonctionnelle (diagramme *FAST*) de la préconception des mécanismes plans. 41

4.1 Exemple de groupe multipolaire (la *diade*). 56
4.2 Différents types de groupes multipolaires et leurs symboles systémiques. 58
4.3 Trois types de groupes à deux pôles d'entrée (diade). 58
4.4 Un mécanisme complexe : schéma cinématique, graphe associé et schéma multipolaire. 60
4.5 Le mécanisme quatre barres : schéma cinématique, graphe associé et schéma multipolaire. 63
4.6 a) Notations pour la diade RRT. b) Organisation des données. 65
4.7 Notations pour le moto-élément (groupe actif formé par une manivelle). 68
4.8 Notations pour la diade RRR (calcul en dynamique inverse). 69
4.9 Comparaison des temps de simulation pour une rotation complète de la manivelle. 71
4.10 Modélisation objet des groupes multipolaires. 73

5.1 Exemple de mécanisme articulé. 76
5.2 Approximation d'une trajectoire désirée exacte par une trajectoire désirée qualitative. 77
5.3 Correspondance entre les extrémités des arcs et segments de la trajectoire désirée et les positions de la manivelle. Détermination des angles de la manivelle. 77
5.4 Décomposition multipolaire du mécanisme de la Figure 5.1. 78
5.5 Méthode de calcul du vecteur vitesse du point effecteur. Base et roulante. 79
5.6 (a) Graphe de déduction causale des CIR manquants à partir des CIR connus. (b) Trace formelle des déductions séquentielles des CIR, en termes d'intersections de droites, que l'on peut représenter graphiquement en (c) 80
5.7 (a) La trajectoire qualitative désirée. (b) Positions remarquables de la manivelle. c) Approximation linéaire (c). 81
5.8 (a) La courbe de l'angle du corps 5 en fonction de l'angle de la manivelle présente un palier. (b) Représentation graphique des propriétés dimensionnelles qui concourent à cette fonction, déduites automatiquement. 82
5.9 Principe général de la génération d'explications qualitatives. 83

6.1 L'éditeur de schémas cinématiques. 88
6.2 Etapes des construction incrémentale de la morphologie d'un mécanisme complexe par rajout de groupes multipollaires. 90

6.3 Panneaux d'édition des paramètres dimensionnels. 91
6.4 L'éditeur de trajectoires désirées. 92
6.5 Panneau de commande de la simulation cinématique. 93
6.6 Editeur de la loi de mouvement des moteurs. 94
6.7 Position singulière en cours de simulation d'un mécanisme *quatre barres*. 95
6.8 Blocage en cours de simulation d'un mécanisme *quatre barres*. 95
6.9 Panneau permettant l'analyse des résultats de la simulation. 96
6.10 Représentation superposée des trajectoires exactes désirée et générée. 97
6.11 Représentation superposée des trajectoires qualitatives désirée et générée. 98

7.1 Une représentation à trois niveaux pour la *conception* à partir de spécifications et pour la *reconception* à partir d'un mécanisme existant. 99
7.2 Les trois positions relatives des cycles adjacents. 103
7.3 Exemples de soustractions de cycles. 104
7.4 Configuration impossible de cycles adjacents appartenant à une base fondamentale. 105
7.5 Exemple de soustraction de cycles adjacents. 106
7.6 Filtrage des sommets préalable à l'optimisation. 110
7.7 a) : Panneau de REALISME permettant le contrôle des paramètres de l'algorithme d'optimisation et des procédures d'affichage. b) : Courbe de convergence pour le dépliage du graphe de la Figure 7.8. 111
7.8 Trois représentations différentes d'un même graphe : procédure d'affichage par défaut (a), première procédure de régularisation (b) et deuxième procédure de régularisation (c). 112

8.1 Détermination d'une trajectorie générée ouverte lorsqu'une simulation complète n'est pas possible. 118
8.2 Panneau de REALISME permettant d'effectuer toutes les opérations nécessaires pour l'optimisation dimensionnelle. 123
8.3 Panneau pour la spécification des paramètres de l'algorithme d'optimisation (gradient conjugué). 124
8.4 Principe de l'optimisation qualitative. 125
8.5 Les trois points caractéristiques sur les éléments de trajectoire qualitative. 125

9.1 Fonction exprimant les formes de trajectoire en fonction des morphologies et des dimensions. 131
9.2 Fonction inverse : la morphologie et les dimensions en fonction de la forme de trajectoire. 131
9.3 Structure générale de la base de cas utilisant l'approximation, par un réseau neuronal artificiel (RNA), de la fonction reliant les dimensions du mécanisme aux formes de trajectoires générées. 133
9.4 Méthodologie d'indexation des cas. 134
9.5 Approche de synthèse de forme par comparaison directe des trajectoires générée et désirée, décrites par une suite de points. 135
9.6 Comparaison indirecte des trajectoires en termes de forme, par l'intermédiaire d'une analyse en série de Fourier. 136
9.7 Premier exemple d'approximation par les cinq premières harmoniques. 139
9.8 Deuxième exemple d'approximation par les cinq premières harmoniques. 139
9.9 Troisième exemple d'approximation par les cinq premières harmoniques. 139
9.10 L'ellipse correspondant à la première harmonique. 140

9.11 Exemple de trajectoire exacte (a) et trajectoire normalisée à partir des cinq premières harmoniques (b). 142
9.12 Mécanisme *quatre barres*. La liaison motrice est en A et le point effecteur est le point E. 145
9.13 Panneau de REALISME permettant la génération des cas pour la méthode de synthèse par réseaux neuronaux. 147
9.14 Méthodologie de génération de cas par tirage aléatoire des paramètres dimensionnels 147
9.15 Structure du réseau neuronal. 149
9.16 Courbe de convergence de l'algorithme d'apprentissage. 150
9.17 Méthodologie d'utilisation du réseau neuronal artificiel. 152
9.18 Méthodologie de recalage géométrique. 154
9.19 Exemple de recalage géométrique de deux trajectoires exactes (à gauche) par la normalisation des deux trajectoires correspondant aux cinq premières harmoniques (à droite). 155
9.20 Procédure de validation avec garantie de l'existence d'une solution exacte. 156
9.21 Interpolation des couplages entre les dimensions du mécanisme et les formes de trajectoire. 157
9.22 Exemples de solutions synthétisées pour sixtrajectoires désirées tirées aléatoirement. La trajectoire désirée et la trajectoire synthétisée sont représentées superposées. . . 158
9.23 Méthodologie de validation avec recalage géométrique. 159
9.24 Trajectoire désirée obtenue par perturbation d'une trajectoire faisable 160
9.25 Procédure de validation sans garantie d'existence d'une solution. 160

10.1 Echelle des niveaux de représentation des trajectoires générées par un mécanisme. 165
10.2 Méthodologie d'indexation des cas qualitatifs. 165
10.3 Paramétrage d'un arc i de la trajectoire qualitative. 166
10.4 Vecteurs propres pour la trajectoire. 170
10.5 Rotation d'un arc de la trajectoire qualitative. 171
10.6 Exemples de positions d'un arc, correspondant à un calcul différent de la distance maximale par rapport à l'origine. 172
10.7 Exemple de normalisation d'une trajectoire qualitative (les vecteurs propres sont également représentés). 174
10.8 Exemples de trajectoire qualitative (en haut), trajectoire qualitative normalisée (au milieu) et clé qualitative (en bas). 176
10.9 Trajectoire exacte dont l'approximation qualitative est donnée dans la Figure 10.10. 178
10.10 Exemple d'approximation de la même trajectoire exacte (en haut) pour trois précisions différentes. 179
10.11 Représentation superposée de deux trajectoires exactes (à gauche) et des trajectoires qualitatives correspondantes après recalage géométrique (à droite). 182

B.1 Organigramme d'un système de CBR typique. 200

C.1 Circulation des signaux à traveurs un neurone. 204
C.2 Structure générique d'un neurone artificiel. 204
C.3 Structure générique d'un RNA de type *feedforward*. 205

Liste des algorithmes

4.1 Algorithme de détermination du schéma multipolaire. 61
4.2 Procédure d'identification des diades. 62
4.3 Détermination de la relation structurale (procédure Détermine-rel-struct). 63
4.4 Détermination du schéma multipolaire (procédure Place-voisins). 64
7.1 Optimisation topologique par transformations successives d'une base de cycles indépendants. 107
7.2 Détermination des n couples de cycles les plus intéressants. 108
7.3 Algorithme d'affichage par défaut. 109
9.1 Procédure permettant de savoir s'il faut effectuer une rotation supplémentaire. . . 141
9.2 Génération aléatoire de cas pour la constitution d'une base de cas. 148
9.3 Détermination des coefficients de Fourier d'après les points de contrôle de splines spécifiés par l'utilisateur. 153
10.1 Algorithme pour la normalisation d'une trajectoire qualitative. 167
10.2 Procédure de création des cas qualitatifs. 178

Remerciements

Je dois à beaucoup de professeurs, collègues et amis la réussite de ma thèse. Durant mes études d'ingénieur à Bucarest, le regretté professeur Christian Pelecudi et Ion Simionescu, forts de leur grande science dans le domaine, m'ont initié à l'étude des mécanismes articulés. Constantin Stancescu et Cornel Pârvu m'ont accueilli chaleureusement dans leur laboratoire de l'Institut Polytechnique de Bucarest, où j'ai pris goût à la CAO...

Y'a-t-il au monde un deuxième Jean-Claude Bocquet? Comme tous ceux qui le connaissent, je suis persuadé que non... Le laboratoire Productique Logistique est sa création et la convivialité qui y règne lui est due en grande partie. Je lui suis reconnaissant d'avoir accepté d'être mon directeur de thèse, mais aussi de m'avoir proposé de suivre, comme année préparatoire, le DEA de Génie des Systèmes Industriels dirigé par Pierre Dejax.

Du premier jusqu'au dernier jour, Bernard Yannou a été à la fois l'encadrant rigoureux et le compagnon de recherche dont j'avais besoin. Son vif esprit critique a parfois censuré mes enthousiasmes injustifiés. Ses idées et propositions, ses travaux sur la cinématique qualitative ont constamment relancé mes propres recherches. *Last but not least*, l'amitié et la confiance qu'il m'a montrées ont été un stimulent permanent.

Pascal Laurent, gardien du temple mathématique au bénéfice de nombreux chercheurs de l'Ecole Centrale, a mis souvent à contribution, à mon profit, son esprit de géométrie, mais aussi celui de finesse...

Mounib Mekhilef m'a donné de précieux conseils et même des morceaux de code intégrés dans REALISME (optimisation par gradient et par recuit, courbes splines).

Mon ami de Roumanie, Andrei Sebe, chercheur infatigable... dans ses heures de loisirs, a contribué à cette thèse par l'image contagieuse de sa passion pour la recherche, mais également par un algorithme, des conseils et des encouragements.

Dans une phase incipiente de cette thèse, Christian Goualou a eu l'amabilité de mettre à mon service sa grande compétence dans la théorie des mécanismes.

Je ne peux m'empêcher d'évoquer ici toute l'amitié que j'ai pour mes collègues Safia Kedad, Sophie Canet-Magnoux, Hakan Celikoglu, Bénachir Medjdoub et bien d'autres. Je me souviendrai aussi de la gentillesse des secrétaires du laboratoire, Anne Prévot, Annie Chahin, Sylvie Guillemain et Agnès Poisson.

Le fardeau des ennuis informatiques s'est subitement allégé lorsque Prébarane Louvois est arrivé au laboratoire pour prendre en charge la maintenance informatique.

Enfin, je remercie Jorge Angeles, Patrick Chedmail, Claude Bohatier et Jean-Pierre Merlet de m'avoir fait l'honneur d'examiner les résultats de mes travaux en participant au jury de thèse.

J'exprime également ma gratitude à l'Association des Centraliens, qui m'a accordé son soutien financier durant les premiers mois de ma thèse.

Quant à mon épouse Gina, les sentiments que j'éprouve pour elle dépassent largement les limites de ce qui peut être dit ici...

Chapitre 1

Introduction

1.1 Domaine

Les objets techniques étudiés dans cette thèse sont les *mécanismes plans articulés* (*linkages*), c'est-à-dire les mécanismes en chaînes cinématiques fermées, à topologie fixe [1], composés de solides rigides connectés par des liaisons élémentaires parfaites, de type pivot et glissière. Cette classe de mécanismes est représentative d'une partie relativement importante des mécanismes utilisés dans l'industrie manufacturière (automobiles, grues, machines agricoles, machines outils, moteurs, mécanismes de préhension en robotique, etc).

A l'intérieur de cette catégorie de mécanismes, nous étudions plus particulièrement les *générateurs de trajectoire*. Pour cette classe de mécanismes, la fonction la plus importante du cahier des charges fonctionnel est de nature cinématique ; elle porte sur la trajectoire parcourue par un point particulier du mécanisme, dit *point effecteur*. Ainsi, la spécification peut porter soit sur un certain nombre de points de passage obligés, soit sur des caractéristiques de formes sur tout ou partie de la trajectoire (arcs de cercle, segment de droite, forme ovale, etc). De plus, à ces points peuvent être associés ou pas des moments précis de temps (*dépendance du temps* ou *indépendance du temps*). Trois exemples industriels de générateurs de trajectoire sont donnés dans la Figure 1.1.

Les hypothèses liées au caractère parfait des liaisons et à la rigidité des solides, ainsi que l'absence d'une étude dynamique, sont généralement compatibles avec le positionnement de cette thèse dans l'étape de *préconception* (ou *conception préliminaire*). Durant cette étape, les mécanismes sont représentés par des schémas cinématiques dimensionnés. Dans la Figure 1.2 est donné un exemple de représentation graphique d'un schéma cinématique, utilisant des symboles qui seront employés tout au long de cette thèse.

Moins complexes à étudier que les mécanismes spatiaux, les mécanismes plans articulés sont surtout moins complexes à fabriquer et, généralement, plus robustes et moins coûteux en fonctionnement. Ils sont mieux adaptés industriellement que les robots (chaîne cinématique ouverte) à des tâches spécifiques. Ceci explique leur présence massive dans l'industrie, où ils répondent à une grande partie des besoins [2].

---

1. C'est-à-dire : les liaisons sont toujours les mêmes dans le temps.

2. Une proportion même approximative entre les mécanismes plans et spatiaux est difficile à établir. Faut-il prendre en compte le nombre de morphologies distinctes utilisées ou le nombre total de mécanismes utilisés dans chaque catégorie ? Faut-il pondérer ces nombres par l'importance économique et technique de chaque mécanisme, exprimée par exemple par son coût ? De plus, il faudrait prendre en compte l'évolution dans le temps et identifier la tendance actuelle. Seule une ample enquête détaillée sur le terrain pourrait fournir une estimation. En effet, le décompte dans les catalogues de mécanismes existants, d'ailleurs la plupart déjà anciens, ne reflète pas forcément d'une façon fidèle la réalité industrielle.

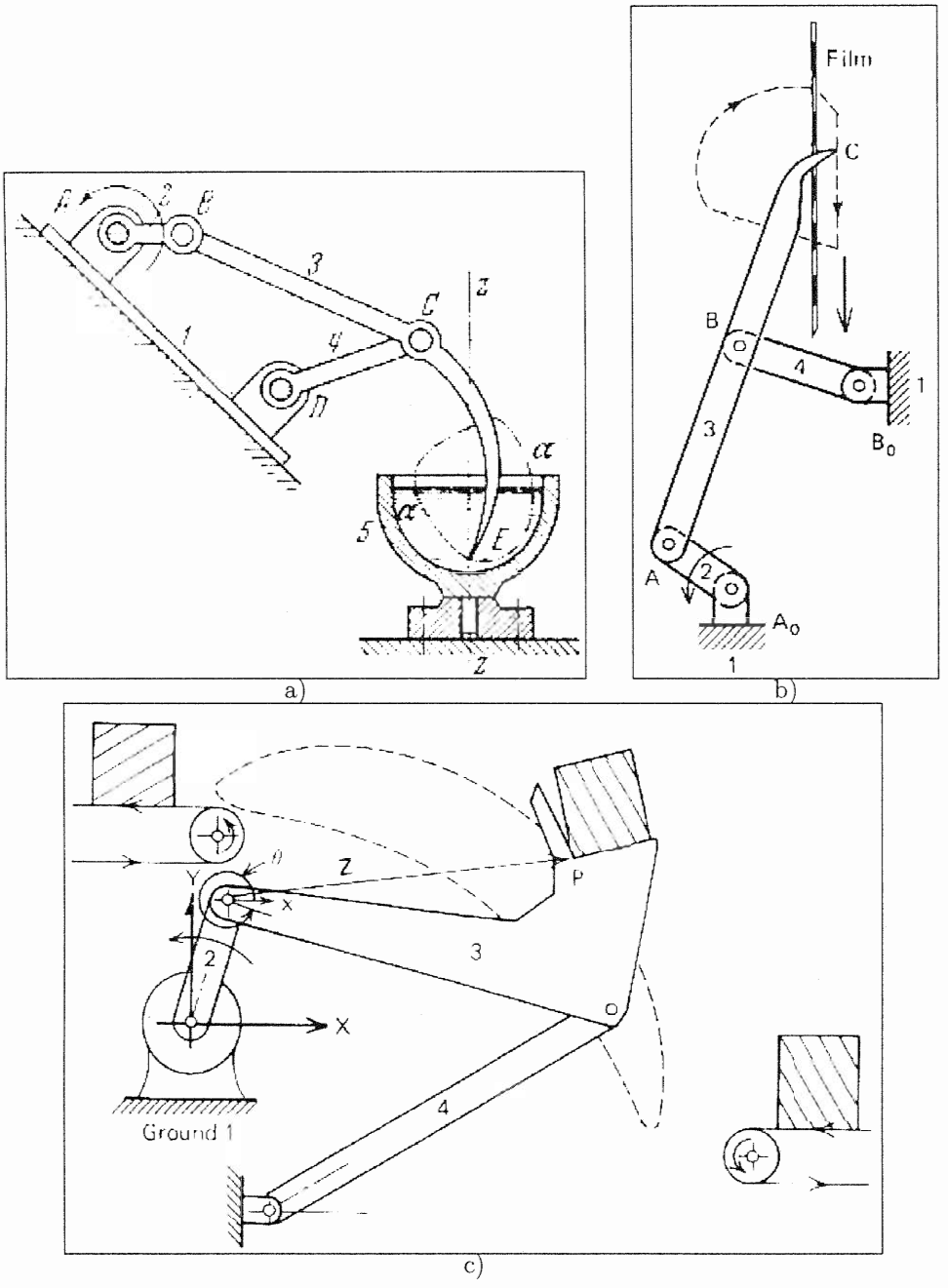


FIG. 1.1 – *Exemples industriels d'utilisation du mécanisme quatre barres comme générateurs de trajectoire. a) : Mécanisme du pétrin mélangeur ([Art75b]). b) : Mécanisme indexeur de pellicule dans un projecteur cinématographique ([SE84]). c) : Mécanisme pour machine d'assemblage ([SE84]). Le premier mécanisme doit réaliser une trajectoire avec indépendance du temps, les deux auires avec dépendance du temps.*

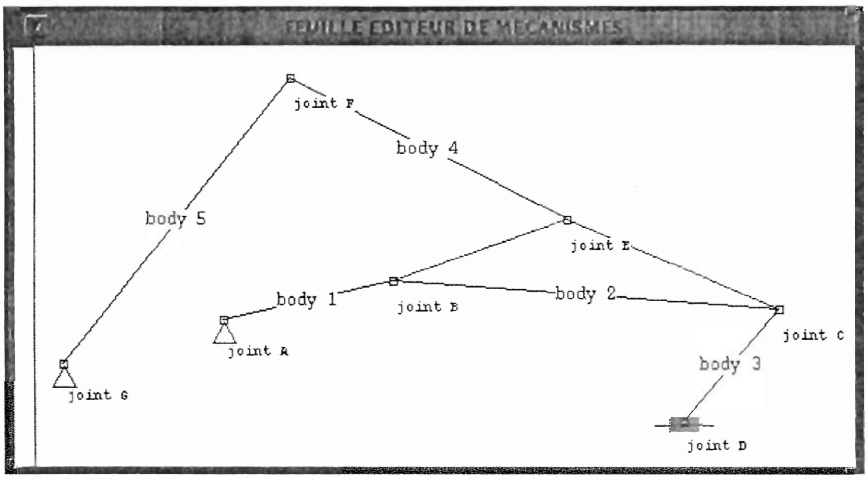


FIG. 1.2 – *Représentation graphique du schéma cinématique d'un mécanisme articulé composé d'un système bielle-manivelle (corps 1, 2, et 3) et d'une diade (corps 4 et 5). La liaison A est une pivot liée au bâti, pilotée par un moteur de rotation ; la liaison D est une glissière liée au bâti ; les autres liaisons sont des pivots.*

Parmi les mécanismes plans, les générateurs de trajectoire ne représentent qu'une fraction qu'il est difficile d'estimer, mais qui est probablement relativement réduite en pourcentage. Compte tenu, cependant, du grand nombre total de mécanismes, le problème de la préconception des générateurs de trajectoire conserve son importance.

1.2 Objectifs

L'étude des mécanismes plans articulés a suscité un grand intérêt de la part des mécaniciens depuis plus d'un siècle. De multiples aspects théoriques et pratiques ont fait l'objet de très nombreux livres, thèses, articles et communications (cf. l'état de l'art, chap. 2). Les recherches se poursuivent actuellement, ce qui est attesté par le nombre important de publications qui voient le jour. Ces recherches visent soit à l'amélioration des méthodes de conception déjà connues, soit au développement de nouvelles méthodes, adaptées à des objectifs de conception plus divers, multi-critères ou à des objectifs typiques de préconception (synthèse de solutions, spécifications qualitatives...).

L'avancée de la théorie des mécanismes et du calcul numérique, même si des progrès restent à faire, permet aujourd'hui de disposer d'outils assez performants d'*analyse* de mécanismes. Souvent basés sur des approches théoriques de la mécanique générale, ces outils sont caractérisés par le souci de généralité (mécanismes spatiaux, éléments flexibles, frottements, jeux, etc) et par la complexité inhérente à leur puissance. Cependant, en phase de préconception – où il s'agit de « dégrossir » le problème en dimensionnant approximativement les mécanismes potentiels – cette généralité et cette richesse ne sont pas toujours nécessaires et risquent de nuire à la simplicité d'emploi et à l'efficacité. De plus, la littérature ne fournit pas de solutions satisfaisantes pour certains types de problèmes de *synthèse*. Or, le besoin existe, car dans la pratique la synthèse repose souvent sur la méthode « essai-erreur ». On peut même estimer que la disponibilité de méthodes de synthèse puissantes et flexibles rendrait inutile, dans certains cas, l'emploi de robots, avec un gain de coût évident[3].

3. Le mécanismes articulés peuvent avoir également une certaine flexibilité, si des réglages de position pour les liaisons fixes ou de longueur pour les barres sont prévus.

La synthèse présente une difficulté beaucoup plus grande que l'analyse. Une première raison de cette difficulté est, comme souvent dans la conception de produits, l'explosion combinatoire du nombre de solutions potentielles. Une deuxième difficulté est spécifique aux produits mécaniques : l'absence de relation bijective entre *structure* et *fonction* cinématique, autrement dit l'impossibilité d'assigner une fonction cinématique précise à un composant structurel. La même barre ou liaison pivot, par exemple, peut appartenir à des mécanismes remplissant des fonctions très différentes. Enfin, une troisième source de difficulté est liée à l'extrême sensibilité par rapport aux valeurs des paramètres : une petite variation de longueur de barre peut complètement modifier le comportement cinématique et dynamique d'un mécanisme.

Pour les générateurs de trajectoire, la plus importante limitation des logiciels et des méthodes de synthèse actuels est l'obligation de définir la fonction du mécanisme, c'est-à-dire la trajectoire désirée, par une liste de points de passage, de plus souvent en nombre très réduit. Aussi, un des objectifs principaux de cette thèse est la mise au point de méthodes s'affranchissant de cette limitation et permettant au concepteur de spécifier la trajectoire désirée en termes approximatifs ou *qualitatifs*, d'une part, portant sur la *forme* de tout ou partie de la trajectoire (trajectoire indexée dans le temps ou pas), d'autre part.

D'une manière plus générale, l'objectif de cette thèse est la définition de méthodes d'aide à la conception des générateurs de trajectoire présentant les caractéristiques suivantes :

- adaptées au *besoin d'analyse* en phase de préconception (permettre de déterminer d'une manière rapide et simple le comportement de différentes alternatives de conception),
- apportant une contribution à la satisfaction du *besoin d'aide à la conception* (pour éviter d'explorer des solutions insatisfaisantes du point de vue cinématique),
- *capitalisant* les résultats des conceptions antérieures,
- orientées vers une *compréhension* de la structure,
- prenant en compte une spécification de la trajectoire désirée en termes de *forme*,
- permettant de prendre en compte une spécification de trajectoire *qualitative*.

Dans cette thèse, nous partons d'un constat qui apparaît à plusieurs reprises dans la littérature : la multitude de formulations possibles du cahier des charges fonctionnel rend très difficile et en pratique même impossible la mise au point d'une méthode de synthèse *générale*, unique. L'attitude adéquate consiste alors à rechercher les approches les plus adéquates pour chaque classe de problèmes de conception ou, si possible, couvrant un certain nombre d'entre elles.

Cette thèse s'inscrit donc dans le domaine des *méthodes de préconception*. Il ne s'agit pas d'une thèse de *théorie des mécanismes*, car les choix théoriques et méthodologiques ont été faits à partir de besoins spécifiques à la préconception. Il nous a paru plus utile de mettre davantage l'accent sur l'adéquation entre, d'une part, les modèles ou les méthodes et, d'autre part les besoins spécifiques (cf. l'étude du point de vue fonctionnel dans le chap. 3).

1.3 RealisMe : une plateforme intégrée de préconception

Cette diversité de problématiques de conception nous a déterminé d'étudier et de développer un ensemble de méthodes, dont la complémentarité et l'intégration sont illustrées par la plate-

forme de préconception REALISME[4]. Dans cette plateforme, conçue pour être évolutive (grâce à la programmation orientée objets), sont implémentés :

- un éditeur graphique de mécanismes (à main levée),
- un module de cinématique, incluant une modélisation systémique modulaire et un modèle de cinématique qualitative,
- un module d'optimisation dimensionnelle : traditionnelle et qualitative,
- un module de synthèse par capitalisation de connaissances, sorte de « compilation » d'un catalogue de trajectoires, avec utilisation de la modélisation qualitative, d'une part, et des réseaux neuronaux artificiels, d'autre part
- un module de représentation dépliée de graphes de mécanismes, utilisant une méthode originale.

Cet ensemble de modules et de méthodes apportent chacun une contribution partielle et complémentaire à la réalisation de l'objectif de cette thèse[5].

Le développement informatique de REALISME a été réalisé essentiellement à l'aide des logiciels de la société *Ilog* :

- *Le-Lisp* : langage Lisp orienté objets (développé par l'*INRIA*),
- *Aïda* : librairie de développement d'interfaces graphiques,
- *Grapher* : librairie de gestion de graphes.

Pour des raisons d'efficacité, les parties nécessitant du calcul intensif font appel à des routines programmées en langage *C*. Le développement des réseaux neuronaux artificiels a été réalisé grâce au logiciel *SNNS* de l'Université de Stuttgart. L'essentiel du module de cinématique qualitative a été conçu et programmé par Bernard Yannou.

La plateforme REALISME, développée sur une station de travail *Sun Sparc 10*, est portable sur toute machine supportant un compilateur *C* et les logiciels *Le-Lisp/Aïda*.

1.4 Organisation

Le mémoire de thèse se divise en quatre parties. La première partie comprend un état de l'art général pour le domaine de la conception des mécanismes articulés plans (chap. 2) et une analyse globale et conceptuelle des besoins spécifiques à la préconception des générateurs de trajectoire, ainsi qu'une typologie générale des approches de préconception (chap. 3).

La deuxième partie présente deux modèles de cinématique jouant un rôle important dans cette thèse : le modèle systémique par décomposition multipolaire des mécanismes (chap. 4) et le modèle de cinématique qualitative (chap. 5). La troisième partie porte sur les méthodologies de préconception : la conception incrémentale par des cycles d'analyse et modification (chap. 6), une approche de dépliage des graphes utile pour la synthèse morphologique (chap. 7), l'approche de synthèse dimensionnelle par l'optimisation (chap. 8), la synthèse par capitalisation de cas utilisant les réseaux neuronaux artificiels (chap. 9) et la synthèse par capitalisation de cas par l'utilisation des trajectoires qualitatives (chap. 10).

4. Comme *Réalisation de mécanismes*, mais aussi comme « réalisme », en référence au parti pris pragmatique des approches mises en œuvre.

5. Une partie des liens multiples unissant ces différents modules seront explicités dans la Figure 3.3, p. 41.

La dernière partie de la thèse contient les conclusions de ce travail et les perspectives pour les recherches futures (chap. 11).

En annexe, sont donnés un glossaire de termes et de concepts fréquement utilisés dans cette thèse[6] (annexe A), une présentation de l'approche de *Raisonnement à base de cas* (annexe B) et une présentation des *réseaux neuronaux artificiels* (annexe C).

La thèse s'achève par la bibliographie et l'index.

6. Ce glossaire comprend des définitions relativement informelles, destinées à expliquer rapidement l'essentiel des termes et concepts. Ces définitions étant données, dans la thèse, au fur et à mesure, le lecteur pourra néanmoins recourir au glossaire en cas de besoin. Dans le texte, il n'y a pas de renvoi systématique vers ces définitions.

Première partie

Etat de l'art et analyse de la problématique

Chapitre 2

Préconception de mécanismes plans : état de l'art

Le présent état de l'art ne se propose pas de couvrir l'ensemble des travaux existants dans le domaine de la préconception des mécanismes plans articulés. Loin d'être exhaustive, cette bibliographie a essentiellement comme but de structurer, de classifier une partie des approches les plus liées aux travaux de cette thèse. En présentant ce cadre, les approches que nous avons adoptées ou développées dans cette thèse pourront être positionnées par rapport aux recherches antérieures, à la fois dans leurs particularités et dans leur continuité.

2.1 Modèles de mécanismes plans

Les mécanismes réels sont des objets physiques. Comme pour tout objet technique, leur étude et leur conception nécessitent l'emploi de modèles. Les usages possibles de ces modèles au sein des méthodes d'analyse et de synthèse sont multiples : l'identification des typologies de mécanismes, la structuration des données morphologiques et dimensionnelles en vue de l'informatisation. Un rôle important est, également, celui de permettre la visualisation des mécanismes même à des stades initiaux de la conception, lorsque seulement une partie des paramètres ont été définis. Ils sont donc indispensables pour l'interactivité entre le concepteur et les méthodes de conception.

Le modèle de *Hartenberg et Denavit* [HD64, p. 347–355], par exemple, comportant une notation symbolique[1] basée sur des matrices de transformation, est bien adapté pour les mécanismes spatiaux et comprend l'information dimensionnelle nécessaire pour l'analyse cinématique et dynamique. Cependant, son caractère très général le rend moins efficace pour les mécanismes plans. De plus, il est difficile de séparer l'information portant sur la morphologie de celle portant sur les dimensions (cf. *Olson* [OER85]). Lorsque l'objectif final est la mise au point de méthodes de préconception[2], le critère prioritaire de choix des modèles est moins en rapport avec leurs qualités théoriques et générales qu'avec leurs avantages par rapport aux méthodes de conception.

Parmi les nombreux modèles de mécanismes proposés, nous étudierons en particulier la modélisation par graphes et par groupes d'Assur. La modélisation par graphes a été beaucoup utilisée (voir §2.1.1), surtout pour la synthèse morphologique. Le russe L. Assur, dont les travaux ont été repris et étendus par différents autres chercheurs, a établi un système de classification des mécanismes plans basé sur une décomposition en sous-chaînes cinématiques (*groupes d'Assur*). Des détails sur ce dernier modèle sont donnés au §2.1.2. L'extension *systémique multipolaire* de ce

1. Le premier modèle symbolique pour les mécanismes appartient probablement à Fr. Reuleaux.
2. Pour une définition de la préconception, voir §1.2, p. 4.

modèle fera également l'objet du chapitre 4. A la fin de ce chapitre, une nouvelle génération de modèles, qualitatifs, d'intelligence artificielle, ou adaptés à une automatisation de la modélisation et de la conception, sera présentée au §2.5.

2.1.1 Modélisation par graphes

La modélisation par graphes des chaînes cinématiques fermées consiste le plus souvent à représenter chaque corps (solide, pièce) par un nœud du graphe et chaque liaison par une arête[3]. Aux entités du graphe peuvent être associées des attributs correspondant aux types de liaisons ou d'éléments (bâti, pièce menante ou menée). Ces graphes sont *connexes* (sinon il représenterait plusieurs systèmes indépendants) et *sans extrémités arborescentes* (sur chaque pièce il existe au moins deux liaisons, sinon il s'agirait de robotique). Le modèle par graphes contient toute l'information concernant la morphologie du mécanisme, abstraction faite des dimensions. Néanmoins, les dimensions peuvent aussi être stockées comme attributs des entités du graphe.

L'analogie naturelle entre les chaînes cinématiques fermées et les graphes a été depuis longtemps mise en évidence. Les premiers travaux utilisant cette idée sont dûs à F. Crossley et à F. Freudenstein, en 1964 (d'après *Olson et al.* [OER85] et *Harary et Yan* [HY90]). Depuis, de nombreuses approches, dont une partie seront mentionnées ci-après, ont exploité ce modèle.

La modélisation par graphes a les caractéristiques suivantes :

- facilement *représentable informatiquement* et *manipulable interactivement* ;
- niveau de représentation topologique intéressant en préconception, lorsque les dimensions n'ont pas encore été définies ; le *découplage de la complexité* entre l'aspect morphologique et l'aspect dimensionnel permet d'envisager des *modèles hiérarchiques* pour la synthèse, comportant une étape morphologique suivie par une étape dimensionnelle (cf. également §7.1) ;
- *généricité* (la modélisation par graphes est « *domain-independent* »).

Les graphes de mécanismes peuvent être planaires ou non planaires[4], mais cela ne semble pas avoir d'incidence sur les propriétés des mécanismes correspondants (cf. *Siestrunck* [Sie73]).

Parmi les nombreux travaux basés sur la modélisation par graphes, on peut distinguer trois familles :

- L'étude des propriétés et des caractéristiques propres à la modélisation par graphes. Par exemple, *Siestrunck* [Sie73] donne une présentation générale de la modélisation par graphes et de son utilisation pour les calculs de l'indice de mobilité et du nombre de mobilités cinématiques (degrés de liberté) du mécanisme, ainsi que pour l'énumération des différentes morphologies. *Harary et Yan* [HY90] étudient, du point de vue de la théorie des graphes, les propriétés de la modélisation par graphes et hypergraphes des chaînes cinématiques, en particulier la bijectivité de la relation entre graphes et chaînes. Ils proposent également des algorithmes simples de construction du graphe unique d'une chaîne et de formation d'une chaîne unique à partir du graphe.
- L'énumération automatique de tous les graphes (morphologies) de chaînes cinématiques et de mécanismes répondant à des critères d'ordre morphologique (voir §2.3.1.1).

3. Ce modèle est préférable, pour l'automatisation ou l'aide à la conception, au modèle classique de *graphe structurel*, où chaque élément est représenté par un polygone ayant un nombre de sommets égal au nombre de liaisons adjacentes à l'élément. Entre autres, sa représentation graphique automatique est plus délicate, à cause de la présence de polygones (cf. *Siestrunck* [Sie73]).

4. Un graphe non planaire ne peut pas être représenté dans le plan sans entrecroisement d'arêtes. (Voir aussi §2.1.1.1 et chap. 7.)

– L'analyse du type de mobilité à partir du graphe (voir §2.3.1.2).

Le plus souvent, les travaux utilisant les graphes visent ou débouchent sur une méthode de synthèse morphologique (voir §2.3.1).

2.1.1.1 La représentation des graphes et le problème du dépliage

Le plus souvent, la modélisation par graphes est utilisée dans les méthodes d'énumération automatique de morphologies. Les solutions ainsi générées doivent être évaluées, ce qui nécessite aussi une représentation graphique, afin de permettre l'interaction avec le concepteur. Le problème de la représentation graphique des graphes de mécanismes peut être formulé ainsi : trouver un jeu de coordonnées des nœuds assurant une bonne lisibilité, en particulier en minimisant le nombre d'entrecroisement d'arêtes. Nous appelons cette opération *dépliage*. Ci-après sont présentés quelques travaux dans le domaine. (Dans le chapitre 7, notre méthode originale de dépliage sera présentée.)

Tests de planarité Le problème de la recherche d'un critère efficace indiquant si un graphe est planaire ou non planaire a depuis longtemps préoccupé les spécialistes de la théorie des graphes. En effet, un algorithme utilisant le critère classique donné par le théorème de Kuratowski explose très rapidement en combinatoire. Ce critère stipule qu'un graphe est non planaire si et seulement si il contient comme sous-graphe au moins un des deux graphes dits de Kuratowksi[5].

Dans une étude très étendue et rigoureuse parue en 1963, *Tutte* [Tut63] explore un certain nombre d'aspects théoriques liés à la planarité des graphes. En particulier, il démontre l'équivalence entre la propriété de planarité et l'existence d'une base de faces du graphe. Cette propriété sera exploitée par la suite dans d'autres travaux sur le dépliage des graphes, y compris pour l'algorithme de dépliage développé dans REALISME (voir chap. 7).

Hopcroft et Tarjan [HT74] décrivent un algorithme permettant de déterminer si un graphe est planaire ou non. L'algorithme utilise une *recherche en profondeur d'abord* et a une complexité en temps et en mémoire de $O(n)$, où n est le nombre de nœuds.

Représentation des graphes planaires L'approche proposée par *Chieng et Hoeltzel* [CH90] pour la représentation des graphes structurels planaires convertit le problème numérique à variables continues – coordonnées des nœuds – dans un problème topologique à variables discrètes. Pour cela, la représentation du graphe est modélisée par les relations topologiques d'intériorité et d'extériorité entres les cycles du graphe (indépendants plus un). Une énumération combinatoire crée toutes les variantes de positionnements relatifs des cycles. Ensuite, un test de planarité spécifique est appliqué pour identifier une configuration permettant une représentation planaire. Finalement, le choix des coordonnées des nœuds est réalisé à partir de la solution topologique précédemment identifiée[6].

Belfiore et Pennestri [BP94] proposent un algorithme non itératif pour la représentation des graphes structurels planaires, utilisant les barycentres des éléments.

Chrobak et Payne [CP95] présentent un algorithme de complexité linéaire pour représenter sur une grille les graphes planaires.

Nievergelt et Deo [ND95] étudient la représentation des graphes planaires comportant des arêtes élastiques, déformables.

5. Ces deux graphes comportent cinq, respectivement six nœuds, tous reliés entre eux par des arcs (connectivité totale).

6. Cette approche est à la base de notre propre algorithme de dépliage des graphes, implémenté dans REALISME. Pour les détails, voir chap. 7.

Représentation des graphes non planaires Selon la plupart des chercheurs, une représentation lisible des graphes non planaires doit respecter les conditions suivantes :

- arêtes représentées par des segments de droite,
- nombre minimal d'entrecroisements d'arêtes,
- cycle extérieur représenté par un polygone convexe.

La représentation dans le plan d'un graphe non planaire en minimisant le nombre d'entrecroisements d'arêtes est connue comme un problème NP-complet, qui ne peut être traité que par des heuristiques.

Watanabe [Wat89] présente une heuristique de minimisation du nombre d'entrecroisements pour une représentation sur une grille. L'algorithme comporte très peu de retours-arrière. De préférence, les nœuds reliés par des arêtes sont placés en proximité (densité uniforme). L'algorithme combine une heuristique de choix du prochain nœud à placer et une heuristique de choix du placement sur la grille pour le nœud choisi.

Kamada et Kawai [KK89] modélisent un système de ressorts entre chaque paire de nœuds, ressorts dont la longueur correspond à la longueur du plus court chemin les reliant (mesurée en nombre de nœuds). Le placement optimal des nœuds est ensuite obtenu par la minimisation de l'énergie du système de ressorts.

Fang et Freudenstein [FF90] présente une approche à trois niveaux : combinatoire, géométrique et ornemental, pour la représentation des graphes structurels de mécanismes. L'approche utilise la contraction des éléments polygonaux.

Bertolazzi et al. [BDBL95] utilisent une approche adaptive, comportant plusieurs algorithmes de représentation et une stratégie de choix de l'algorithme le plus approprié.

Dans un article récent, *Mauskar et Krishnamurty* [MK96b] utilisent le même type d'approche que *Chieng et Hoeltzel* [CH90] (voir §2.1.1.1), avec un algorithme différent. Le résultats sont annoncés meilleurs.

Une application des algorithmes génétiques pour la minimisation du nombre d'entrecroisements, présentée dans *Mäkinen et Sieranta* [MS94], est limitée à la représentation des graphes bipartis.

2.1.2 Modélisation par groupes d'Assur

Le cinématicien russe L. Assur formulait, dans ses travaux entrepris entre 1913 et 1918 à Saint-Petersbourg, un principe de construction des mécanismes par adjonctions successives de chaînes cinématiques définies par certaines propriétés structurales (voir *Artobolevski* [Art75b, p. 47–49]). L'idée fondamentale est de considérer les mécanismes comme un assemblage de « composants » cinématiques standards, appelés ultérieurement *groupes d'Assur*.

L'approche – connue également sous le nom d'*approche modulaire*[7] – a été développée par les mécaniciens soviétiques, notamment *Artobolevski*[8] [Art75b]. *Manolescu* [Man68] l'a utilisée à des fins de classification des mécanismes et des chaînes cinématiques. L'application à l'analyse cinématique a été étudiée par *Crossley et Seshachar* [CS71] et par *Bogdan et Huncher* [BH67]. Les recherches de *Verho* [Ver73] ont porté sur l'extension du concept de *groupe d'Assur* pour inclure des groupes à mobilité non nulle, comportant autant de liaisons motrices que de degrés de liberté.

Il existe deux variantes principales de l'approche par groupes d'Assur. Dans la première, les groupes sont définis comme des chaînes cinématique *fermées*. Ainsi, *Galletti* [Gal86] définit les

7. Un certain flou terminologique subsiste dans la littérature entre l'*approche modulaire* et l'*approche par composants*. Nous préférons réserver le nom d'*approche par composants* à celle visant à identifier des vecteurs ou des multi-vecteurs (cf. par exemple *Olson et al.* [OER87], *Sardain* [Sar93]).

8. Auteur du célèbre catalogue de mécanismes [Art75a].

chaînes cinématiques d'Assur (CCA) comme des chaînes cinématiques *fermées* de mobilité $m = 0$ (structures), à partir desquelles il est impossible d'obtenir d'autres chaînes cinématiques de mobilité nulle par suppression d'un ou de plusieurs éléments. Une définition équivalente est donnée par *Innocenti* [Inn94] : il définit une CCA comme une structure ne contenant pas d'autre structure. Autrement dit, une CCA composée de n pièces représente l'arrangement minimal des n pièces pour lequel, en connaissant la position d'une pièce, on peut déterminer la position des $n-1$ autres pièces.

La deuxième variante d'approches par groupes d'Assur décompose le mécanisme en sous-chaînes cinématique *ouvertes*. Ainsi, *Verho* [Ver73] définit le groupe comme étant une partie d'un mécanisme, dont la mobilité, quand les liaisons extérieures sont immobilisées, est égale au nombre de liaisons motrices contenues dans le groupe et ne pouvant pas être divisé en parties plus petites vérifiant le même critère. *Artobolevski* [Art75b, p. 48] définit les groupes d'Assur comme « *une chaîne cinématique ouverte qui* [d'une part] *a le degré de mobilité nul par rapport aux éléments qui forment des couples cinématiques avec les parties libres de ses éléments et qui* [d'autre part] *ne peut être décomposée en des chaînes plus élémentaires possédant aussi un degré de mobilité nul* ».

Dijksman et Timmermans [DT94] décrivent les groupes d'Assur, d'une façon moins précise, comme « *une partie d'une chaîne fermée qui ne peut être enlevée sans modifier la mobilité ou la fermeture de la chaîne* ». Cette définition donne une condition nécessaire, mais non suffisante, puisqu'elle ne pose pas la condition qu'un groupe ne soit pas décomposable en d'autres groupes.

Un développement dans le cadre de la théorie des systèmes a été apporté par l'école roumaine de théorie des mécanismes, notamment *Pelecudi et Simionescu* [PDS82], [PS82], [PS86a], [PS86b]. Cette approche – l'*approche systémique multipolaire* – est développée dans le chapitre 4.

Kinzel et Chang [KC84] rappellent que la plupart des mécanismes sont composés d'une série de composants cinématiques standards, dont l'analyse cinématique peut être faite sans calcul itératif en identifiant l'ordre dans lequel ils doivent être analysés et en utilisant les *équations explicites* pour chaque composant. Le nombre de mécanismes pouvant être analysés cinématiquement par l'approche modulaire est élargi, grâce à une extension de la technique d'inversion de Goodman. Cependant, aucune automatisation de la décomposition modulaire n'est prévue (l'utilisateur doit « déterminer mentalement » l'ordre de résolution cinématique). Nous verrons par la suite (chap. 4) que l'approche systémique multipolaire permet d'obtenir cette décomposition en équations explicites, accélérant ainsi l'analyse cinématique.

Suh et Radcliffe [SR78, p. 21] font l'observation que la plupart des mécanismes plans sont un assemblage de quelques composants standards. Cependant, leur attention va directement à l'exploitation de cette observation pour l'analyse cinématique et aucune étude supplémentaire n'est faite sur les principes de cette décomposition. Les équations simplifiées pour l'analyse des cinq groupes les plus simples sont déterminées.

Au début, la décomposition en groupes était effectuée « manuellement » par l'utilisateur. Les premiers logiciels basés sur cette approche demandaient à l'utilisateur de préciser la manière de décomposer le mécanisme. Plusieurs chercheurs ont ensuite proposé des procédures de décomposition automatique. On peut citer celle de *Romdhane* [Rom95] pour l'identification des *diades* (groupes d'Assur composés de deux pièces reliées par une liaison) par une recherche des plus courts chemins dans le graphe orienté du mécanisme. Cette méthode est implémentée dans le logiciel d'analyse *KAMEL*. D'autres méthodes sont proposées par *Lauw et al.* [LCK87] et par *M.R. Hansen* [Han96].

Luck et al. [LMR85] présentent le logiciel *GISK* d'analyse de mécanismes, basé sur les groupes d'Assur et ayant également des fonctionnalités limitées de synthèse.

Salem et Manoochehri [SM90] présentent un système d'analyse et de synthèse basé uniquement sur les diades.

Innocenti [Inn94] détermine les solutions explicites pour l'analyse dimensionnelle de la chaîne cinématique d'Assur avec 7 pièces (contenant 4 pièces à trois liaisons chacune, connectées en série).

Une approche modulaire, mais utilisant une autre définition des composants, s'applique aux mécanismes en chaîne cinématique ouverte, en particulier aux bras manipulateurs en robotique (voir par exemple *Manoochehri et Seireg* [MS90]). Une autre extension de l'approche modulaire, proposée par *Nishioka* [Nis96], consiste en une adaptation de l'approche pour une classe de mécanismes spatiaux comportant des cames. Dans cette approche, aux modules plans connus s'ajoutent des modules spatiaux.

L'approche de décomposition en groupes utilisée dans cette thèse est basée sur les concepts systémiques développés dans les travaux de Pelecudi et Simionescu déjà cités. Les détails sur l'approche systémique et son implémentation dans REALISME seront donnés au chap. 4.

2.2 Analyse cinématique

2.2.1 Méthodes spécifiques à un type de mécanisme

Une première classe de méthodes d'analyse cinématique[9] comprend les méthodes développées spécifiquement pour une seule morphologie. La connaissance préalable de la morphologie permet d'incorporer dans la méthode un certain nombre de calculs symboliques qui accélèrent la simulation par l'utilisation de solutions explicites. Cependant, l'absence de généralité et la complexité très vite croissante des calculs symboliques nécessaires diminuent l'intérêt de ces méthodes.

Par exemple, *Bahgat et Osman* [BO92] utilisent l'élimination gaussienne pour établir les équations paramétriques de la trajectoire décrite, en fonction des paramètres dimensionnels indépendants, pour un mécanisme à huit barres avec liaisons de type pivot et glissière.

2.2.2 Méthodes générales

On distingue couramment (cf. par exemple *Hansen* [Han92]) deux classes de méthodes générales d'analyse cinématique :

1. méthodes basées sur une bibliothèque de liaisons standard,

2. méthodes basées sur une bibliothèque de chaînes cinématiques standard (l'approche modulaire).

2.2.2.1 Méthodes basées sur les liaisons

Le premier groupe de méthodes utilise comme données le nombre de pièces et le type des liaisons entre les pièces. Ce sont des méthodes bien adaptées aux mécanismes spatiaux, mais moins pour les mécanismes plans (vitesse réduite, problèmes de convergence...). Néanmoins, des progrès récents dans l'étude des systèmes multicorps ont permis l'amélioration de cette formulation dans des logiciels comme *ADAMS* ou *DADS*. Dans ces logiciels, l'analyse cinématique est réalisée par intégration numérique, à partir de conditions initiales, des équations de mouvement (Newton-Euler ou Lagrange) (cf. [SJ93]). Le livre de *Schiehlen* [Sch90] contient une comparaison des différents logiciels basés sur cette approche. Certains progrès ont été faits dans des logiciels plus récents, comme *Applied Motion* (moins d'équations, intégration plus robuste).

Un état de l'art de ces méthodes est donné par *Bawab et Kirzel* [BK95]. Différentes méthodes utilisées pour établir et résoudre les équations sont présentées ci-après.

9. Une définition de l'anayse cinématique est été donnée à la page 195.

Equations de fermeture de boucle L'approche basée sur les *multi-éléments contraints*, utilisée par les logiciels *DADS* et *ADAMS*, implique la définition, pour chaque élément, d'un repère local, dont la position et l'orientation dans le plan sont décrites par trois coordonnées généralisées (cf. *Sohoni et Haug* [SH82]). A chaque liaison correspond une contrainte entre deux pièces ; on obtient ainsi, pour chaque boucle cinématique, un système d'équations non linéaires faiblement couplées. Le système est résolu grâce à une méthode d'optimisation par projection du gradient.

Le logiciel *DADS* utilise des matrices de transformation modélisant les contraintes de liaison et de rigidité des éléments (cf. *Chieng et Hoeltzel* [CH88]). L'écriture des équations de fermetures des boucles cinématiques correspond au produit ordonné des matrices de transformation. La résolution des équations nécessite une estimation de la solution par résolution d'un système linéaire. Les principaux avantages sont le caractère très général de la méthode et la non-accumulation des erreurs. En revanche, la complexité est relativement grande (en $O(n^3b)$, b = nombre d'itérations) et il est difficile d'obtenir une solution en proximité d'une singularité.

Une méthode semblable est utilisée dans les logiciels *IMP*, *DYMAC*, *DAMS*. *Conti et al.* [CCM92] proposent une approche unifiée pour la prise en compte des contacts intermittents et la détection des interférences entre les pièces.

Méthode des nombres complexes Les logiciels comme *ADAMS, DRAM, LINCAGES*, utilisent la méthode des nombres complexes pour modéliser la position des liaisons et l'orientation des pièces. Dans le cas des petits déplacements, les mouvements (considérés comme superpositions de déplacement linéarisés) peuvent être propagés aux éléments adjacents. La somme des petits déplacements relatifs dans une boucle cinématique est nulle. Cette méthode et robuste et la complexité algorithmique est acceptable (en n^3, la même que celle de l'élimination gaussienne). Cependant, la définition des petits déplacements dépend du problème et les erreurs dues à l'approximation linéaire s'accumulent d'étape en étape dans la procédure itérative d'analyse (cf. [CH88]).

Les nombres complexes sont également utilisés pour l'analyse cinématique (mais aussi pour la synthèse) par *Erdman et Sandor* [ES84].

Méthodes algébriques matricielles Le mouvement d'une pièce rigide peut être considéré comme la somme d'une rotation et d'un déplacement linéaire d'un point de référence arbitraire fixé sur la pièce. Les paramètres de cette rotation et du déplacement linéaire sont rangés par *Suh et Radcliffe* [SR78] dans des matrices de déplacement.

Les matrices de Hartenberg-Denavit [HD64] ont été utilisées pour l'analyse cinématique des mécanismes plans et spatiaux. La méthode utilise l'écriture d'une série de transformations de coordonnées le long des boucles cinématiques.

Conti et al. [CCM92] développent une approche numérique unifiée (liaisons à contact permanent ou intermittent), basée sur les matrices de liaison et sur l'expression des contraintes de liaisons.

Les matrices ont été également utilisées pour l'établissement des équations de synthèse dimensionnelle (cf. par exemple *Mariappan et Krishnamurty* [MK96a]).

2.2.2.2 Méthodes basées sur les modules

Les méthodes basées sur les équations de boucles cinématiques et les méthodes matricielles, présentées ci-avant, ont l'avantage d'être générales (mécanismes plans et spatiaux). Néanmoins, elles ont l'inconvénient d'exiger une résolution numérique itérative pour les équations de position. Elles sont donc relativement lentes et leur convergence n'est pas assurée, car elle dépend fortement du choix de la solution initiale. Au prix d'une moindre généralité, ces désavantages sont éliminés par le deuxième groupe de méthodes, utilisant une bibliothèque de modules ou composants dont

l'assemblage forme le mécanisme à analyser. Les données sont le nombre et le type de ces modules ainsi que leur mode de connexion et leurs dimensions physiques.

Un des premiers logiciels de simulation cinématique basés sur les groupes d'Assur a été *LINKPAC*, développé par *Suh et Radcliffe* (voir [SR78]). Dans *LINKPAC*, un nombre très réduit de types de groupes, dont les équations explicites d'analyse sont prédéterminées, est pris en considération. Lorsque le mécanisme ne se prête pas à une solution explicite, la résolution est effectuée par l'algorithme de Newton-Raphson. Ce logiciel, comme d'autres logiciels basés sur la même approche, sont constitués d'un ensemble de routines informatiques correspondant à l'analyse de chaque groupe. L'analyse du mécanisme est obtenue par des appels à ces routines dans un ordre établi le plus souvent par l'utilisateur.

Chieng et Hoeltzel [CH88] présentent une méthode d'analyse basée sur l'identification symbolique de boucles cinématiques standard et l'utilisation d'une solution prédéterminée séparément pour chacune de ces boucles. A partir de la matrice de connectivité des éléments, un ensemble canonique de boucles indépendantes est identifié, ayant un minimum d'éléments partagés par plusieurs boucles (un ensemble de *faces* au sens de la théorie des graphes). Cependant, cet ensemble n'est pas unique et il n'y a pas de justification rigoureuse de la méthode. Enfin, la résolution des équations est réalisée en rattachant les boucles cinématiques à certaines classes dont les solutions sont connues a priori.

Enderton [End90] présente un logiciel d'analyse basé sur les groupes d'Assur (diades uniquement). Le logiciel décrit par *Salem et Manoochehri* [SM90] ne prend en compte que deux types de groupes d'Assur et exige une décomposition manuelle en groupes. En revanche, ce logiciel comporte certaines fonctionnalités pour la synthèse.

Mlynarski [Mly96] présente une méthode permettant de décomposer les groupes d'Assur de classe supérieure en groupes de classe inférieure, dont l'analyse admet des solutions explicites. La méthode est basée sur une disjonction de certaines liaisons (par introduction de liaisons supplémentaires). Ensuite, l'analyse consiste à chercher (par optimisation) une position du mécanisme ainsi transformé afin que les liaisons disjointes soient connectées de nouveau. Cette méthode implique l'introduction de variables supplémentaires dont le choix dépend de la morphologie du mécanisme.

Le logiciel présenté par *M.R. Hansen* [Han92] [Han96], basé sur l'approche modulaire, est très complet. Il comporte une décomposition automatique en groupes et une extension de l'approche modulaire à l'analyse dynamique et statique.

2.2.3 Approches basées sur les contraintes géométriques

Le logiciel *TLA*, programmé comme sur-couche d'un système de gestion de contraintes géométriques, génère, par des techniques d'Intelligence Artificielle, une procédure d'assemblage permettant de calculer les variables dépendantes en fonction des variables indépendantes (cf. le commentaire de *Sacks et Joskowicz* [SJ93] au livre de *Kramer* [Kra92]). Cet assemblage inclut le calcul explicite à chaque fois que c'est possible et utilise, le cas échéant, des méthodes itératives. L'algorithme de *propagation de contraintes* parcourt de manière séquentielle les boucles indépendantes en calculant un maximum de variables dépendantes par propagation des valeurs des variables dans les contraintes de liaison. Des techniques d'intelligence artificielle sont utilisées pour naviguer dans la combinatoire, lorsque plusieurs « chemins » de propagation sont possibles. Le désavantage est qu'il n'y a pas de modèle paramétrique ; il faut donc refaire tous les calculs à chaque fois, ce qui réduit un peu le gain d'efficacité. Une comparaison avec *ADAMS* a été faite, *TLA* s'avérant 5 à 100 fois plus rapide. La complexité est de l'ordre de $O(n)$. La comparaison de la vitesse avec la résolution par Newton-Raphson et avec la méthode de continuation de Morgan indique un gain d'un ordre de grandeur. Le principal avantage est donc la vitesse, importante pour l'animation rapide. Cette

approche tend – comme l'approche modulaire par groupes d'Assur – à minimiser les résolutions itératives de systèmes d'équations, mais elle est moins rigoureuse, moins systématique et générale.

2.2.4 Approches hybrides

A partir des travaux déjà cités d'*Enderton* [End90] et de ceux de Kramer, *Brunkhart* [Bru94] a développé le logiciel *MechEdit*. Ce logiciel comporte un solveur de contraintes géométriques utilisant une approche hybride : les solutions explicites pour les groupes d'Assur sont utilisées pour réduire la complexité des systèmes d'équations et la méthode de Newton-Raphson est employée pour les variables restantes.

Bawab et Kinzel [BK95] présentent une comparaison des avantages relatifs des différentes méthodes d'analyse cinématiques, en relevant la meilleure efficacité de l'approche modulaire dans le cas des mécanismes complexes, grâce à l'utilisation de solutions explicites des équations de position. Ceci élimine le plus souvent tout calcul numérique itératif et améliore la convergence autour des singularités. Cependant, la méthode proposée par ces auteurs ne retient que l'idée de grouper les équations de façon à obtenir des sous-systèmes pour lesquels une solution explicite est connue. Le cas échéant, une résolution itérative (Newton-Raphson) est utilisée. Cette méthode est basée sur l'inspection d'une matrice d'incidence spécifique, exprimant l'apparition des variables dans les différentes équations.

Konkar95 se propose d'améliorer la rapidité de l'analyse cinématique, pour les besoins de la conception itérative, par la réutilisation de résultats intermédiaires. La méthode de simulation utilise une modélisation algébrique par des torseurs (*screw mechanics*).

Sakamoto et al. [SOF94] utilisent l'arrangement des éléments sur plusieurs strates parallèles (layers) afin d'éviter les collisions durant le mouvement.

Une des applications bien connues de l'Intelligence Artificielle est le calcul formel. *Garnier et Rideau* [GR91] présentent un système utilisant le logiciel de calcul formel *Maple* comme boîte à outils pour l'établissement du système d'équations algébrico-différentielles représentant le mouvement de systèmes multicorps.

2.2.5 Logiciels commercialisés

D'après les passages en revue de l'état de l'art réalisés par *Erdman* [Erd95] et *Sardain* [Sar93], nous présentons une liste de logiciels commercialisés, comportant des fonctionnalités d'analyse cinématique :

- sur mainframe : *IMP* (*Integrated Mechanisms Program*), *ADAMS* (*Automatic Dynamic Analysis of Mechanical Systems*), *DYMAC* (*Dynamics of MAChinery*), *DADS* (*Dynamic Analysis and Design Systems*)

- sur micro ordinateur et stations de travail : *MICROMECH* et *KADAM* (*Kinematic and Dynamic Analysis of Mechanisms*), *MCADA*, *ANALYTIX*, *Working Model*, *Mechanica-Applied Motion*[10], *MINNSKETCH* (sketcher convivial intégré avec *KADAM*), *Mechanism Design* (intégré avec *I-DEAS*), *Mechanisms* (intégré avec *UNIGRAPHICS*), *Mechanisms* (intégré avec *BRAVO*), *Mécanisme* (intégré avec *EUCLID*), *MSP* (intégré avec *CADDS*), *KIDS* (intégré avec *AutoCAD*), *Cinématique* (intégré dans *CATIA*), *ADAMS*, *DADS*, *SDS*.

Le logiciel *4barres*, dédié à l'analyse cinématique du mécanisme *quatre barres*, a été développé par *Merlet* [Mer95]. Ce logiciel est mis à disposition gratuitement par l'INRIA.

10. Ce logiciel utilise des modèles paramétriques, dont chaque paramètre peut devenir une variable de conception pour l'optimisation. Une analyse de sensibilité peut être effectuée.

2.3 Synthèse morphologique

La synthèse morphologique, c'est-à-dire le choix d'une morphologie de mécanisme, précède traditionnellement la synthèse dimensionnelle et joue un rôle essentiel dans la conception. En effet, la morphologie influe sur les fonctions remplies par le mécanisme et détermine en grande partie son coût. Par exemple, *Freudenstein et Maki* [FM84] effectuent une analyse de la morphologie des pompes à pistons axiaux et cylindrée variable et démontrent que de nouveaux types de pompes, à double fréquence, peuvent être trouvées. La synthèse morphologique créative peut ainsi conduire à d'importantes innovations.

2.3.1 Approches basées sur les graphes

2.3.1.1 Enumération de morphologies

La modélisation par graphes a été largement utilisée pour l'énumération de morphologies.

Mayourian et Freudenstein [MF84] présentent une méthode de création d'un catalogue[11] de chaînes cinématiques, sous contraintes de nombre de degrés de liberté et de barres.

Mruthyunjaya [Mru79] [Mru84] décrit un algorithme d'énumération des morphologies de chaînes cinématiques basé sur la transformation de chaînes binaires. Une première énumération, plus restreinte en combinatoire, concerne des chaînes composées d'éléments binaires avec liaisons multiples. Ces chaînes sont ensuite utilisées pour déterminer toutes les combinaisons possibles de chaînes avec liaisons simples et multiples, en remplaçant successivement les liaisons multiples avec deux liaisons plus simples et en remplaçant les éléments binaires par des éléments polygonaux. Le catalogue créé comprend les chaînes cinématiques jusqu'à 10 barres et trois degrés de liberté.

Sohn et Freudenstein [SF86] utilisent le graphe dual[12] pour développer un algorithme d'énumération automatique des morphologies jusqu'à 11 barres et deux degrés de liberté.

Parmi les chaînes cinématiques énumérées par les méthodes précédentes, beaucoup sont dégénérées (c'est-à-dire ne peuvent pas correspondre à un *mécanisme réel*) ou isomorphes (équivalentes) et doivent en conséquence être éliminées. En ce qui concerne le dernier cas, si *Mayourian et Freudenstein* [MF84] ne mentionnent pas du tout ce problème, *Mruthyunjaya* [Mru79] [Mru84], ainsi que d'autres chercheurs, ont utilisé comme critère pratique de détection de l'isomorphisme les coefficients du polynôme caractéristique de la matrice d'adjacence. Ce critère s'est ensuite avéré faux. Tang et Liu ont été les premiers à proposer, en 1988, l'utilisation d'un réarrangement d'une matrice d'adjacence dans l'ordre du degré des nœuds (*degree code*) comme critère de détection de l'isomorphisme. La méthode, plus simple et plus efficace que celles précédemment utilisées est présenté dans *Tang et Liu* [TL93]. Cette méthode fournit également un formalisme très compact de modélisation des chaînes énumérées.

Tuttle et al. [TPT89a] [TPT89b] utilisent la théorie des groupes finis de symétrie afin d'éliminer les chaînes isomorphes, dans le cadre d'un algorithme d'énumération des chaînes planaires non fractionnées, pour des nombres de barres et de degrés de liberté donnés.

Un autre algorithme d'énumération est proposé par *Hwang* [HH92]. Le catalogue de morphologies créé par l'énumération contient toutes les chaînes non dégénérées et non isomorphes comprenant jusqu'à 13 barres. L'élimination des chaînes isomorphes est basée sur la méthode du *degree code*, déjà mentionnée.

Un critère de détection de l'isomorphisme basé sur les *distances pseudo Hamming* est proposé par *Rao* [RR93].

11. Il s'agit de 35 graphes représentant une large classe de mécanismes plans et spatiaux comportant jusqu'à six pièces (barres).

12. Les nœuds du graphe dual représentent les cycles du graphe conventionnel (décrit au §2.1.1) et les arêtes relient les nœuds représentant des cycles partageant des arêtes.

Liu et Yu [LY95] étudient l'énumération des variantes de chaînes cinématiques en considérant l'introduction de moteurs. Un catalogue est établi contenant 262 mécanismes pour des liaisons pivots et un maximum de huit barres. La détection des isomorphismes est réalisée par la méthode du *degree code*.

Dijksman et Timmermans [DT94] décrivent un algorithme d'énumération des chaînes cinématiques ne contenant pas des groupes d'Assur.

La méthode d'énumération décrite par *Tuttle* [Tut96] couvre les chaînes cinématiques planaires, non fractionnées, comportant jusqu'à six boucles indépendantes et jusqu'à trois degrés de liberté.

2.3.1.2 Analyse de la mobilité

Les propriétés de mobilité sont des propriétés structurales très importantes pour la conception des mécanismes, pour l'étude desquelles la modélisation par graphes est particulièrement utile.

Litvin [Lit80] s'intéresse à la détermination du nombre de degrés de mobilité instantanée.

Baker [Bak81] étudie les mobilités internes et *Davies* [Dav81] utilise une analogie entre les boucles cinématiques dans un mécanisme et les circuits électriques afin d'étudier les mobilités internes du mécanisme.

Mruthyunjaya et Raghavan [MR84] utilisent les structures de données de la modélisation par graphes (matrice d'incidence) pour déterminer les propriétés de mobilité (partielle, fractionnée ou totale) des morphologies.

Agrawal et Rao [AR87] présentent une méthode d'analyse systématique de la mobilité d'une chaîne cinématique.

Gosselin et Angeles [GA87] obtiennent une représentation graphique des différentes régions de mobilité pour les mécanismes articulés plans et sphériques.

Ting [Tin89] étudie les critères de mobilités des mécanismes comportant une seule boucle cinématique.

A partir de la définition d'un ensemble de boucles cinématiques élémentaires, *Liu et Yu* [LY95] décrivent une méthode pour l'identification du type de mobilité des mécanismes à plusieurs degrés de liberté.

2.3.1.3 Utilisation pour l'aide à la synthèse morphologique

La modélisation des mécanismes par graphes a été utilisée, dans la littérature, à des fins de synthèse morphologique.

Olson et al. [OER87] utilisent la modélisation par graphes pour identifier des *composants* (diades, triades et vecteurs libres), afin de générer les équations de synthèse.

Yan [Yan92] présente une méthodologie de conception comprenant une double démarche : généralisation d'un mécanisme existant puis, à partir de la chaîne généralisée, énumération des mécanismes respectant les mêmes contraintes topologiques. Le même type d'approche est présenté par *Hsu* [Hsu92]. La méthodologie de généralisation, basée sur la modélisation par graphes, procède par l'élimination, dans les graphes, des liaisons multiples (reliant plus de deux solides).

Rao [RR93] présente une méthode utilisant les graphes et permettant de comparer deux morphologies caractérisées par le même nombre de barres et de liaisons. Dans un autre article [Rao95], Rao s'attache à classer les morphologies selon des critères fonctionnels (erreur structurale, performances statiques et dynamiques), même si la comparaison exacte des morphologies n'est pas possible en absence des dimensions. Le principe de classement porte essentiellement sur le concept de « compacité » des chaînes cinématiques. Pour cela, la distance entre deux barres est calculée comme étant le plus petit nombre de barres qui les séparent (autrement dit le chemin le plus court dans le graphe correspondant). Les distances ainsi calculées entre tous les couples de barres sont

rangées dans une matrice « distance ». La somme des éléments de cette matrice (*distance value*) est considérée comme un indicateur de « compacité » de la chaîne.

Chew et al. [CSI95] utilisent la modélisation de la morphologie par des graphes dans le cadre d'un système à base de connaissance de synthèse morphologique (voir ci-après).

2.3.2 Systèmes experts et à base de connaissances

Chew et al. [CSI95] proposent un système à base de connaissances utilisant la modélisation par graphes de la morphologie. A partir des spécifications, un ensemble de solutions potentielles est généré. Un filtrage est ensuite effectué, en éliminant les solutions ne répondant pas aux spécifications en utilisant les 85 règles de la base de connaissances.

Toutefois, on doit remarquer que ces approches ne garantissent pas que l'énumération est *complète* et que le filtrage est *cohérent* (total).

2.4 Synthèse dimensionnelle (cinématique)

Par *synthèse dimensionnelle* d'un mécanisme on entend la détermination de ses dimensions, afin que le mécanisme corresponde aux spécifications du cahier des charges. Dans cet état de l'art, nous nous intéressons particulièrement à des spécifications portant sur la trajectoire générée (*génération de trajectoire*).

Les chercheurs s'accordent sur le fait qu'il n'existe, actuellement, aucune méthode de synthèse dimensionnelle générale, répondant à tous les besoins. Nous citons à ce propos *Mariappan et Krishnamurty* [MK96a] : « *Currently no single method, analytical or numerical, exists that is well suited for the dimensional synthesis of general mechanisms.* » et *Kota et Chiou* [KC93] : « *No single method existed that is best suited to all problems.* »

En revanche, il existe une grande diversité de méthodes de synthèse dimensionnelle, chacune comportant des avantages et des désavantages et adaptée à des besoins spécifiques. Ces méthodes sont classifiées en méthodes graphiques, méthodes algébriques (analytiques) et méthodes d'optimisation (cf. par exemple les états de l'art de *Root et Ragsdell* [RR76] et d'*Erdman* [Erd85] [Erd95]).

2.4.1 Méthodes graphiques

Les méthodes graphiques – les seules disponibles jusqu'au milieu des années cinquante – utilisent des constructions géométriques pour la détermination des dimensions du mécanisme. Hélas, ces méthodes s'appliquent seulement à des morphologies préalablement connues et pour des spécifications simples (quelques points de passage pour la génération de trajectoire). Leur utilisation est fortement limitée par leur complexité, par l'imprécision des constructions graphiques et par la difficulté d'être répétées souvent dans un processus de conception itérative.

Dans certains cas, les méthodes graphiques peuvent fournir une solution initiale pertinente pour les méthodes analytiques, en valuant les paramètres constituant des « choix libres ». Dans les années cinquante, Freudenstein et Sandor ont adapté les méthodes graphiques proposées dès 1876 par Burmester pour mettre au point des méthodes de synthèse (cf. *Luck et Modler* [LM94] [LM95]). *Jensen* [Jen84] présente une méthode de synthèse des mécanismes *quatre barres* décrivant une trajectoire symétrique passant par 12 points.

L'équation d'Euler-Savary est souvent utilisée dans les approches de synthèse basées sur des méthodes graphiques. Cette équation met en relation le centre instantané de rotation d'une pièce, un point arbitraire appartenant à la même pièce et le centre instantané de courbure. *Sandor et al.* [SXW90, p. 320–327] présentent une méthode graphique pour la résolution de l'équation d'Euler-Savary.

Eraslan [Era79] décrit une méthode directe de synthèse des approximateurs de lignes droites basée sur deux équations décrivant le centre de courbure de la trajectoire.

Grâce à une « transcription » algébrique de la méthode géométrique des pôles (*pole method*), *Tong et Chiang* [TC92] établissent un système d'équations non linéaires pour la synthèse des générateurs de trajectoires. Le système est résolu par la méthode de Newton-Raphson. *Blechschmidt et Uicker* [BU86] présentent une autre application d'une méthode géométrique et algébrique permettant d'obtenir un système d'équations linéaires pour la synthèse de trajectoire.

2.4.2 Méthodes analytiques

Les méthodes analytiques comportent l'établissement et la résolution d'un système d'équations. A partir des travaux de Freudenstein, la recherche sur la synthèse dimensionnelle a longtemps porté presque uniquement sur les méthodes analytiques. Ces méthodes tirent profit du développement de l'informatique pour fournir des solutions numériques aux problèmes de synthèse complexes et fortement non linéaires, sans nécessiter une solution initiale. Le champ d'application des méthodes analytiques concerne la génération de trajectoire, de mouvement et de fonction (voir les définitions à la page 196).

Pour les générateurs de trajectoire et de mouvement, les méthodes analytiques impliquent la spécification d'un certain nombre de points de passage (*precision points*) appartenant à la trajectoire désirée. Le but de la procédure de synthèse est de déterminer les dimensions d'un mécanisme passant exactement par ces points, le plus souvent avec contraintes de temporalité. L'avantage de ces méthodes est qu'elles fournissent des solutions exactes. En revanche, pour qu'une solution exacte existe, le nombre de points de passage imposés ou de variables dimensionnelles libres doit être limité. Ci-après nous présenterons brièvement deux formulations des méthodes analytiques, basées sur les matrices de déplacement et sur les nombres complexes. D'autres méthodes analytiques ont été proposées, par exemple par *James et Roth* [JR93] [JR94].

2.4.2.1 Méthode des matrices de déplacement

La méthode des matrices de déplacement, proposée en 1967 par Suh et Radcliffe (cf. *Suh et Radcliffe* [SR78], *Simionescu et Cezar* [SC79]), s'applique à la synthèse des générateurs de mouvement. La composition des matrices de déplacement permet d'exprimer analytiquement le mouvement de sortie en fonction de celui d'entrée et d'établir de la sorte les équations de synthèse. A partir de cette méthode, *Sathyadev et Soni* [SS88] développent une méthode de synthèse dont l'objectif est de déterminer les dimensions d'un mécanisme dont une droite tangente appartenant à l'organe effecteur (*coupler*) décrit l'enveloppe tangente de la trajectoire désirée.

A partir de l'approche d'analyse basée sur les *multi-éléments contraints*, utilisée par les logiciels *DADS* et *ADAMS*, et de l'analyse de sensibilité, *Sohoni et Haug* [SH82] développent une méthode de synthèse utilisant les mêmes matrices que celles générées durant l'analyse cinématique et utilise les équations d'état, obtenues en combinant les contraintes générées par les liaisons et les contraintes supplémentaires dues à la présence du/des moteur(s).

2.4.2.2 Méthode des nombres complexes

Les méthodes de synthèse basées sur les nombres complexes suivent une approche analogue aux méthodes basées sur les matrices de déplacement, mais emploient les nombres complexes pour modéliser la position des barres (cf. *Hartenberg et Denavit* [HD64, p. 321]). Ainsi, la position initiale d'une barre k est décrite par un vecteur :

$$\mathbf{Z}_k = Z_k e^{i\theta_1} = Z_k(\cos\theta_1 + i\sin\theta_1) \tag{2.1}$$

où $Z_k = |\mathbf{Z}_k|$ = longueur de la barre entre les liaisons attachées à ses extrémités, $\theta_1 = \arg \mathbf{Z}_k$ = angle du vecteur $\mathbf{Z}_k$ mesuré à partir de l'abscisse d'un repère à orientation fixe solidaire de la barre (cf. *Sandor et Erdman* [SE84, p. 88-92]).

La méthode de synthèse étudie le mouvement de couples d'éléments (ou vecteurs), appelés *diades*[13]. En effet, une grande partie des mécanismes plans sont composés d'une combinaison de diades. Par exemple, le mécanisme quatre barres peut être vu comme l'assemblage de deux diades. L'équation de fermeture de la boucle cinématique est écrite en sommant les vecteurs le long de la boucle, permettant ainsi d'établir les équations de synthèse. En fonction du nombre de points de passages imposés, un certain nombre de paramètres dimensionnels du mécanisme sont choisis par le concepteur, afin de permettre la résolution du système d'équations.

Sandor et Erdman [SE84, p. 88-92] citent l'utilisation de cette méthode pour la synthèse de générateurs de trajectoire passant par deux, trois, quatre et cinq positions, de générateurs de mouvement, de trajectoire et de fonction avec trois positions imposées. Une extension de la méthode permet de synthétiser les mécanismes comportant plusieurs boucles cinématiques indépendantes.

Dans certains travaux, par exemple *Olson et al.* [OER87] ou *Sardain* [Sar93], cette méthode de synthèse est appelée *méthode des composants* (*component-based approach*). Dans *Olson et al.* [OER87], les composants de base sont des vecteurs connectés en série, à un, deux ou trois éléments : vecteurs libres, diades (bi-vecteurs) et triades (tri-vecteurs). Le graphe orienté du mécanisme est utilisé pour l'identification d'un ensemble indépendant de composants et pour générer automatiquement les équations de synthèse. La méthode est générale (mécanismes complexes, multi-boucles), mais le nombre de points de passage imposés est limité à cinq. La méthode des composants est utilisée également par *Sardain* [Sar94] [Sar93].

D'autres travaux basés sur la méthode des nombres complexes sont ceux de *Conte et al.* [CGMS75], *Erdman* [Erd81], *Luck et Modler* [LM95].

Dhingra et Mani [DM93a] [DM93b] proposent une approche basée sur une combinaison de calcul symbolique et de calcul numérique, la *théorie de Burmester*, l'algèbre des nombres complexes et les équations de fermeture de boucles cinématiques. Jusqu'à 11 points de passage peuvent être spécifiés, pour des classes particulières de mécanismes.

2.4.2.3 Elimination des solutions incorrectes

Les méthodes analytiques de synthèse conduisent souvent à des solutions indésirables : mauvais ordre de parcours des points de passage, blocage, bifurcation, angle de transmission trop faible etc. (cf. *Erdman* [Erd95]). En général, leur élimination requiert l'intervention de l'utilisateur.

Toutefois, il existe des méthodes pour éliminer entièrement ou en partie ces défauts (*solution rectification*). Par exemple, *Waldron* [Wal94] présente une méthode géométrique pour éliminer – parmi les solutions obtenues par les méthodes analytiques de synthèse pour trois points de passage imposés – celles comportant des défauts de bifurcation ou d'ordre.

2.4.3 Méthodes d'optimisation

Les méthodes analytiques de synthèse dimensionnelle sont adaptées seulement pour une partie des problèmes de conception réels. Leur grand désavantage est le nombre très réduit de points de passage qui peuvent être prescrits. D'une part, le comportement du mécanisme entre ses points de passage n'est pas pris en considération. D'autre part, la synthèse analytique exacte n'est exacte qu'en théorie, puisqu'il y aura toujours une imprécision due aux erreurs de fabrications, aux jeux dans les liaisons, à la flexibilité des éléments etc. (cf. *Aviles et al.* [AAGdJ85]). De plus, les solutions

13. Le terme de *diade* est également employé par l'approche modulaire (voir chap. 4).

obtenues ne sont pas toujours acceptables (cf. §2.4.2.3) et le concepteur est obligé à adopter des valeurs dimensionnelles arbitraires pour une partie des paramètres.

Les méthodes d'optimisation permettent de spécifier un plus grand nombre de points de passage. L'idée est d'accepter une certaine erreur, c'est-à-dire que le point effecteur ne passe pas exactement par les points de passage spécifiés ; un algorithme d'optimisation est utilisé pour déterminer un ensemble de paramètres dimensionnels minimisant cette erreur. Avec ce type d'approche, le nombre de points de passage n'est plus limité que par la complexité de l'optimisation, qui croît avec le nombre de variables, et, de façon générale, par les difficultés d'assurer la convergence vers une solution acceptable.

En revanche, le plus souvent, les méthodes d'optimisation présentent aussi un certain nombre d'inconvénients :

- convergence relativement lente vers une solution et pas de garantie d'obtenir un résultat satisfaisant,
- difficulté de choisir les valeurs initiales pour les variables de conception,
- aucune formulation du problème d'optimisation n'est assez générale pour être adaptée à tous les types de besoins (cf. cependant *Aviles et al.* [AAGdJ85]).

Il existe un grand nombre d'approches d'optimisation dimensionnelle des mécanismes plans, dont une partie sont présentées par la suite. Plusieurs critères de classification de ces méthodes sont possibles, selon la formulation du problème d'optimisation, le choix des variables de conception, les algorithmes d'optimisation, le domaine d'application, etc.

Ainsi, une partie des approches établissent des équations de synthèse soit à partir des équations de fermeture des boucles cinématiques soit par la détermination d'équations paramétriques du mouvement spécifiques à une morphologie. D'autres approches, plus générales, consistent en un bouclage sur des analyses cinématiques afin de calculer pour chaque valeur du vecteur de conception la valeur de la fonction objectif.

Certaines approches utilisent des contraintes sur les variables de conception pour interdire les solutions correspondant à des mécanismes dont l'assemblage n'est pas possible. D'autres approches préfèrent éviter l'optimisation sous contraintes et incluent le critère d'assemblage dans une fonction pénalité (parfois avec le choix du paramétrage cartésien absolu des dimensions du mécanisme).

Certaines utilisent un calcul explicite des dérivées partielles de la fonction objectif par rapport aux variables, afin d'utiliser des algorithmes d'optimisation plus efficaces. Le plus souvent, ceci est fait au prix d'une moindre généralité, car ce calcul des dérivées est dédié à certaines morphologies. D'autres approches utilisent l'approximation des dérivées par différences finies, au prix d'un plus grand nombre d'évaluations de la fonction objectif.

2.4.3.1 Programmation non linéaire

La *programmation non linéaire* consiste en une minimisation d'une fonction objectif représentant la distance entre les spécifications et les performances obtenues. Pour la génération de trajectoire, par exemple, il s'agit le plus souvent de la somme des carrés des distances entre les points de la trajectoire désirée et ceux de la trajectoire générée. Différents types d'algorithmes d'optimisation peuvent être utilisés. On peut distinguer trois catégories (cf. *Akhras et Angeles* [AA90]) :

- les *méthodes directes*, n'utilisant pas de dérivée. Elles sont faciles à mettre en œuvre, mais lentes. Exemple : *Suh et Radcliffe* [SR78] ;

- les méthodes basées sur le *gradient* (*Newton–Gauss*), utilisant la dérivée première. Elles présentent deux variantes :
  - résolution itérative à partir des équations normales : *Suh et Radcliffe* [SR78],
  - résolution à partir d'une orthogonalisation des équations d'erreur linéarisées (pour la linéarisation, voir par exemple *Minoux* [Min89]) ;
- méthodes *hessiennes* (*Newton-Raphson*), utilisant la dérivée seconde. Ce sont les plus fiables, mais elles sont beaucoup plus coûteuses en calcul.

Parmi les approches d'optimisation, on peut aussi citer :

- *Penalty function approach* : *Alizade et al.* [ASR75] (avec une nouvelle méthode pour l'approximation de la solution initiale) ;
- *Geometric programming approach* : *Rao* [Rao79] ;
- *Sensitivity coefficient method* : *Huang et Roth* [HR93], *Pakes et al.* [PPW79] (avec le critère d'assemblage inclus dans la fonction de pénalité) ;
- *méthodes heuristiques* d'optimisation : *Lee* [Lee76] ;
- l'*approche de l'espace d'état* : *Sohoni et Haug* [SH82] ;
- *Generalized reduced gradient* (*GRG*) : *Suh et Radcliffe* [SR78] ;
- *formulations stochastiques* : *Choubey et Rao* [CR82] (prise en compte dans l'optimisation non seulement de l'erreur structurale, mais aussi des erreurs de fabrication et d'assemblage).

2.4.3.2 La méthode *Selective Precision Synthesis*

Cette méthode est une approche mixte, combinant les avantages des méthodes analytiques exactes et des méthodes d'optimisation. Les chaînes cinématiques sont considérées comme formées de diades (bi-vecteurs). Des voisinages acceptables, dits de précision (*accuracy neighbourhoods*), sont définis autour des points de passage (*precision points*). L'optimisation vise à trouver un mécanisme dont la trajectoire traverse tous les voisinages. Cette méthode, développée par *Kramer et Sandor* [KS75], a été utilisée, entre autres, par *Bajbai et Kramer* [BK85], *Rigelman et Kramer* [RK88]. Dans le logiciel *MOSYS* de *Braune et Lücke* [BL93], la taille des voisinages de précision fait partie des variables de conception pour l'optimisation.

En plus des défauts habituels, qui peuvent apparaître avec les méthodes analytiques de synthèse (voir §2.4.2.3), les mécanismes synthétisés avec cette méthode présentent un défaut spécifique : l'erreur d'assemblage des diades. Cette erreur apparaît lorsque deux diades sont connectées, par exemple, pour former un mécanisme *quatre barres*. *Bajbai et Kramer* [BK85] présentent une méthode de détection et d'élimination des tous ces défauts.

Rigelman et Kramer [RK88] utilisent la *Selective Precision Synthesis* pour minimiser la puissance maximale du moteur en satisfaisant à la fois des contraintes cinématiques et dynamiques. Le tout débouche sur un problème d'optimisation non linéaire résolu par la méthode du gradient généralisé réduit (en anglais *GRG*). L'approche est spécifique au mécanisme *quatre barres* mais prend en compte différents types de spécifications.

2.4.3.3 Optimisation par bouclage sur des analyses

Salem et Manoochehri [SM90] effectuent l'optimisation dimensionnelle par un bouclage sur des analyses cinématiques et dynamiques. L'analyse emploie l'approche modulaire par groupes d'Assur. Le logiciel est programmé dans un environnement *AutoCAD/AutoLISP*. Deux techniques d'optimisation sont utilisées : *directions conjuguées de Powell* et l'exploration aléatoire (*random walk methods*). La fonction objectif est exprimée comme une distance quadratique entre la trajectoire désirée et la trajectoire générée. Les variables de conception sont les longueurs des barres, les angles dans la position initiale et les positions des liaisons liées au bâti. L'optimisation prend en compte des contraintes de mobilités sur les composants et des plages de valeurs acceptables pour les variables de conception. Aucune indication n'est donnée sur la façon dont les contraintes sont intégrées dans la fonction pénalité.

L'approche modulaire est également utilisée par *Cossalter et al.* [CDPS92], avec une formulation différente du problème d'optimisation (des détails sont données au §8.2.1.1).

Pakes et al. [PPW79] réalisent l'optimisation par une recherche aléatoire. L'approche est adoptée pour pallier à l'existence de nombreux minima locaux de la fonction objectif.

On peut également citer les approches d'optimisation par bouclage sur des analyses cinématiques de *Conte et al.* [CGMS75] (avec utilisation de la méthode des nombres complexes pour l'analyse), *Braune et Lücke* [BL93], *Mistree et al.* [MMSM90].

L'approche d'optimisation par bouclage sur des analyses cinématiques, qui est la plus générale, a été adoptée dans cette thèse (cf. chap. 8).

2.4.3.4 Utilisation des équations de fermeture des boucles cinématiques

Dans cette classe de méthodes, à partir des équations de fermeture des boucles cinématiques, un système d'équations non linéaires est formé. Ce système est résolu par des méthodes numériques itératives.

Une difficulté provient de la prise en compte des contraintes d'inégalité, par exemple pour exprimer le fait qu'une longueur de barre doit être positive. En présence de telle contraintes, les solutions ne sont plus continues par rapport aux paramètres indépendants. *Akhras et Angeles* [AA90] éliminent les contraintes d'inégalités par des techniques de séparation des variables (*variable-separation technique*), en découplant les variables de configuration par rapport aux paramètres du mécanisme. L'optimisation est effectuée par un algorithme des moindres carrés sans contrainte, avec utilisation des techniques de continuation pour améliorer la convergence (cf. §2.4.3.8).

Hansen [Han92] utilise une formulation vectorielle pour établir les équations correspondantes à la fermeture des boucles cinématiques. La décomposition du mécanisme en groupes d'Assur permet de formuler des contraintes d'assemblage. Les techniques de continuation (voir §2.4.3.8) sont utilisées.

Bagci et Burke [BB93] utilise l'approche de minimisation des erreurs dans les équations de fermeture des boucles cinématiques. Cette approche conduit souvent à des systèmes algébriques linéaires. En cas de présence de termes non linéaires, la technique de superposition linéaire est utilisée pour factoriser les équations en composants linéaires. Un catalogue de trajectoires est utilisé pour fournir la solution initiale. L'approche s'applique aux mécanismes complexes (multi-boucles). La synthèse de mécanismes complexes, à plusieurs cycles indépendants est traitée également par *Olson et al.* [OER87], *Soni et al.* [SDW88].

D'autres exemples de ce type d'approche existent : *Lee* [Lee76], *Sohoni et Haug* [SH82], *Gosselin et Angeles* [GA88] (spécifique aux manipulateurs parallèles à trois degrés de liberté), *Krishnamurty et Turcic* [KT92] (utilisant les techniques de *nonlinear goal programming*), *Mariappan et Krishnamurty* [MK96a] (utilisation de la notation matricielle et calcul explicite des dérivées partielles).

2.4.3.5 Utilisation des équations paramétriques du mouvement

Dans ce cas, la fonction objectif est construite à partir des équations paramétriques exprimant le mouvement en sortie en fonction des entrées (variables de conception), ce qui n'est possible que pour une classe limitée de mécanismes. Des exemples sont les travaux de *Cleghorn et Fenton* [CF84], *Pugh* [Pug84], *Manoochehri et Seireg* [MS90] (pour les bras manipulateurs), *Dhingra et Rao* [DR91], *Sundar et Shiller* [SS94].

2.4.3.6 Optimisation multicritère et floue

Certaines approches d'optimisation utilisent l'optimisation multicritère et floue afin de prendre en compte simultanément plusieurs critères de conception, notamment des critères combinés de cinématique et de dynamique (inverse). Le plus souvent, ces approches sont basées sur les techniques de *nonlinear goal programming*.

Le *non linear goal programming* a été utilisé par *Krishnamurty et Turcic* [KT92] (problème général de synthèse). *Sandgren* [San90] utilise un arbre d'alternatives de conception (morphologies admissibles). Pour chaque alternative, des spécifications éventuellement différentes et contradictoires sont données, y compris pour celles comportant des incertitudes, des tolérances, des variables discrètes ou floues.

Pugh [Pug84] réalise une optimisation multicritère par la recherche de l'optimum de Pareto sur un ensemble de solutions potentielles générées aléatoirement, avec minimisation de l'erreur structurale et maximisation de l'angle de transmission.

Garret et Hall [GH68] effectue un choix optimal dans un ensemble de solutions générées aléatoirement. Après leur génération, les solutions sont triées par rapport à des critères d'optimalité.

Rao [Rao87] utilise l'*optimisation floue* pour « traduire » des contraintes imprécises, formulées linguistiquement et exprimant des objectifs multiples, en une formulation mathématique utilisant la logique floue (contraintes et fonction objectif floues). L'optimum global est déterminé. La méthode a une applicabilité générale.

Dans l'approche de *Conte et al.* [CGMS75], l'optimisation des critères dynamiques est faite après une première synthèse de trajectoire pour un mécanisme *quatre barres* passant par trois positions imposées, l'optimisation est réalisée sur des critères de dynamique. Afin de ne pas trop dégrader les performances cinématiques, des contraintes d'inégalité sont introduites.

Cleghorn et Fenton [CF84] réalisent une optimisation *avec dépendance du temps* et une minimisation de l'accélération angulaire maximale de l'organe effecteur.

Sukhija et Rao [SR87] utilisent une approche basée sur la théorie de l'information. Les erreurs dans la sortie sont vues comme des variables aléatoires et une fonction d'entropie est définie. La synthèse du générateur de trajectoire revient à minimiser l'entropie maximale. Quatre points de passage peuvent être spécifiés. Mais cette approche nécessite la connaissance de l'équation de la sortie fonction des variables de conception, donc l'approche n'est pas générale.

Dhingra et Rao [DR91] utilisent une formulation générale de programmation multicritère combinant des critères cinématiques et cinétostatiques. Les variables de conception sont les dimensions et les orientations des barres. Pour optimiser les critères cinétostatiques on joue également sur l'équilibrage (*counterweights*). Les solutions obtenues sont garanties contre le défaut de Grashof, le défaut de bifurcation et le défaut d'ordre. Des contraintes vagues et imprécises sont prises en compte dans la définition des objectifs de conception par l'utilisation de la théorie des ensembles flous (à la suite de *Rao* [Rao87]). Un autre exemple d'optimisation combinée est *Rigelman et Kramer* [RK88].

Mistree et al. [MMSM90] utilisent une formulation linguistique des compromis entre les critères cinématiques et dynamiques, afin de les intégrer dans un *DSP* (*Decision Support Problem*). Ces auteurs concluent sur la nécessité d'utiliser la logique floue.

2.4.3.7 Méthodes énergétiques

Les éléments sont considérés flexibles et la fonction d'erreur est basée sur l'énergie élastique accumulée par le mécanisme lorsqu'il est contraint de suivre exactement la trajectoire désirée. Cette formulation permet d'utiliser la méthode pour tous les types de mécanismes et de problèmes de synthèse. Ainsi, *Aviles et al.* [ANAH94] calculent l'énergie par la résolution d'un système non linéaire d'équations d'équilibre statique grâce à la minimisation (sans contrainte) par une méthode de Newton de second ordre. La méthode de synthèse de *Vallejo et al.* [VAHA95] comporte deux étapes : détermination des positions déformées successives du mécanisme et minimisation de la fonction d'erreur. Cette minimisation est réalisée par un algorithme de recherche linéaire et détermination de la direction par une méthode de Newton de second ordre.

Une autre approche de synthèse combinée cinématique et dynamique inverse est celle de *Huang* [HR93]. Il s'agit d'une méthode systématique pour formuler les équations d'équilibre statique (en utilisant le principe du travail virtuel) et pour les résoudre simultanément avec les équations de synthèse cinématique. La prise en compte de la dynamique inverse réduit (encore plus) le nombre de positions qu'on peut prescrire.

2.4.3.8 Méthodes de continuation

L'objectif des méthodes de continuation est d'augmenter la stabilité de l'optimisation et de diminuer sa sensibilité par rapport aux solutions initiales. Dans le cas de la génération de trajectoire, l'idée est de définir une série de trajectoires intermédiaires, déterminées par une interpolation entre la trajectoire générée par le mécanisme courant et la trajectoire désirée. L'optimisation est effectuée séquentiellement, pour chacune de ces trajectoires. (Cf. *Hansen* [Han93] ; une discussion de détail peut être trouvée en *Angeles et al.* [AAA88] et des explications plus sommaires en *Akhras et Angeles* [AA90]. Une bibliographie commentée se trouve aussi dans *Hansen* [Han92].)

Les méthodes de continuation peuvent être vues comme une procédure mathématique de résolution pas à pas de systèmes d'équations non linéaires. En effet, les méthodes comme Newton-Raphson ou Newton-Gauss ont une convergence quadratique seulement dans le voisinage de la solution optimale. L'idée est de faire en sorte de se trouver toujours « près » de la solution. Un ensemble d'équations ayant une solution connue est d'abord établi. Ces équations sont ensuite progressivement approchées des équations réellement à résoudre. La solution du système construit est trouvée d'une manière itérative, à partir d'une solution initiale qui est celle de l'itération précédente.

Des exemples de travaux utilisant ces méthodes sont *Angeles et al.* [AAA88], *Akhras et Angeles* [AA90], *Wampler et al.* [WMS90], M. R. *Hansen* [Han92], J. M. *Hansen* [Han93], *Stärnis et Flugrad* [SF93], *Subbian et Flugrad* [SF94]. Un état de l'art des méthodes de résolutions des systèmes polynomiaux utilisés en cinématique, y compris les méthodes de continuation, est présenté par *Raghavan et Roth* [RR95].

2.4.4 Logiciels

- *KINSYN* [14] (cf. *Erdman* [Erd85]), *RECSYN*, *CADAM*, *SYNTRA*, *LINCAGES* (quatre points de passage, définis en *ICEM DESIGN/DRAFTING*) : Erdman 1983, *MECSYN* : Tang, 1988 :
- *CADOM* et *TADSOL* : synthèse combinée morphologique et dimensionnelle des générateurs de trajectoire et de fonction ;
- *Spacebar* : analyse et synthèse ; unité de base : le mécanisme *quatre barres* ; mais très peu de détails sur la méthode (cf. *Ricci* [Ric93]) ;

14. Développé par Kaufman au MIT, à la fin des années 60, c'est le premier logiciel graphique interactif de conception de mécanismes.

- *AUCASYN* : synthèse de quatre barres pour cinq points de passage par la méthode des moindres carrés (*Leinonen et Paarni* [LP90] ;
- *CADME* : logiciel basé sur l'approche modulaire (*M.R. Hansen* [Han92] [Han96]) ;
- Logiciels commercialisés : *BRAVOMOST* (*Mechanism Optimal Synthesis Tool*), intégré dans *BRAVO*, *DE/Mec* (logiciel autonome). (Cf. *Sardain* [Sar93, p. 35].)

L'article de *Mittelstadt* [MRE85] propose un cahier des charges virtuel et idéal des logiciels intégrés de conception de mécanismes.

2.4.5 Systèmes experts et à base de connaissances

Soni et al. [SDW88] développent un système à base de règles pour aider l'utilisateur à choisir la morphologie du mécanisme. Le choix est effectué sur des critères liés au type de problème de synthèse, au nombre de degrés de liberté, au nombre de liaisons fixes et de barres, au nombre de points de passage spécifiés etc. Le système crée ensuite les équations de synthèse, qui sont résolues numériquement par une correspondance homotopique, sans solution initiale requise.

Hoetlzel et Chieng [HC90b] donnent un état de l'art des méthodes de synthèse utilisant les bases de connaissances. Le point de départ est l'idée de mise en correspondance des *fonctions* et de la *structure*. Cependant, dans le cas des mécanismes une telle correspondance n'est pas biunivoque. Il faut donc essayer de capitaliser plutôt l'expérience des concepteurs et des heuristiques de conception. Deux familles d'approches se distinguent :

- Les approches directes (*forward reasoning*), basées sur l'énumération systématique d'alternatives satisfaisant les buts de conception. Exemple : l'énumération automatique de graphes de mécanismes (logiciel *MECXPERT*). L'avantage de cette approche est l'exhaustivité, tandis que les désavantages sont l'absence de garantie de trouver la solution et des temps de calcul longs.
- Les approches inverses (*backward reasoning*), qui réalisent la synthèse à partir d'une base de données de cas antérieurs, en adaptant le cas le plus proche. Exemple : *Pattern Matching Synthesis* (*PMS*). (Voir les détails au §2.5.3.) L'avantage est la garantie de trouver la solution (si elle fait partie de celles enregistrées) et les temps de conception très courts. En revanche, on est limité par la taille du catalogue de solutions.

Un autre exemple de système expert est donné dans *Yang et al.* [YDDK91], où est décrit le logiciel *DOMES* (*Design of Mechanisms by an Expert System*).

2.4.6 Théorie de la courbure de trajectoire

Lorsque les trajectoires générées doivent présenter des propriétés particulières de courbure, il existe des méthodes de synthèse utilisant la théorie de la courbure de trajectoire (*path curvature theory*). *Watanabe* [Wat92] utilise, pour les trajectoires désirée et générée d'un mécanisme *quatre barres*, l'expression paramétrique (*the natural equation*) de la variation du rayon de courbure en fonction de l'abscisse curviligne. Cette variation est indépendante des translations et des rotations de la trajectoire et sert pour le calcul de l'erreur structurale durant la synthèse. Pour le calcul de la courbure de la trajectoire désirée, on fait passer une spline par les points, tandis que pour la trajectoire générée ce calcul utilise la construction géométrique de Bobillier. L'article étudie également les transformations géométriques optimales permettant de caler la courbe générée sur la courbe désirée. Le facteur d'échelle est calculé par un rapport des longueurs des trajectoires. L'angle de la manivelle qui minimise la valeur absolue de la différence entre les valeurs de la

fonction à la fois pour les équations naturelles et par transformation géométrique de la trajectoire générée afin que la somme des carrés des distances entre les points correspondants soit minimale. Le processus de synthèse utilise une procédure de recherche aléatoire (*direct search using uniform random numbers*).

Eraslan [Era79] décrit une méthode directe de synthèse de mécanisme générateurs de segments de droite, utilisant les équations du centre de courbure de la trajectoire.

Diverses publications (*de Sa et al.* [DSR81], *Gibson et al.* [GMX96]) s'attachent à étudier soit du point de vue géométrique, soit du point de vue algébrique, le propriétés du mouvement dans le plan et les propriétés spécifiques des trajectoires générées par les mécanismes plans. Certaines de ces propriétés ont des applications pour la synthèse.

2.4.7 Catalogues de trajectoires

Depuis longtemps, les catalogues de trajectoires ont été utilisés pour aider le concepteur lors de la synthèse. Ainsi, le catalogue de *Hrones et Nelson* [HN51] contient 7300 trajectoires générées par le point effecteur d'un mécanisme *quatre barres* (cf. *Erdman et Sandor* [ES84, p. 453]). Le catalogue d'Artobolevski [Art75a] est une large compilation de mécanismes utilisés dans différents domaines.

Lorsqu'une méthode de synthèse par optimisation est utilisée, les catalogues peuvent fournir une solution initiale. (Cf. *Bagci et Burke* [BB93] : « *Approximate geometric method of synthesis of multiloop mechanisms using coupler curve atlases has become very popular. In fact it has been the only efficient method of synthesis of multiloop planar mechanisms that guarantees uniform (constant velocity) input rotation.* »)

Zhang et al. [ZNH84] utilise, pour la synthèse des mécanismes à cinq barres avec un engrènement (*geared five-bar linkages*), un catalogue sur papier de trajectoires contenant 732 courbes générées par ordinateur. Afin de limiter l'explosion combinatoire, ce catalogue est limité aux mécanismes à dimensions symétriques, à quatre paramètres dimensionnels et à quatre ratios (*gear ratio*). L'utilisateur choisit visuellement une trajectoire et le mécanisme correspondant sert de solution initiale pour un processus d'optimisation. Les auteurs préconisent une « interpolation visuelle » entre les solutions du catalogue.

Unruh et Krishnaswami [UK95] partent du constat de l'absence d'une méthode de synthèse adaptée à la synthèse de trajectoires pour un nombre infini de points (*infinite point coupler curve synthesis problem*) et proposent d'utiliser un catalogue de trajectoires informatisé. Afin de minimiser la quantité d'information à stocker et de faciliter la comparaison entre deux trajectoires en termes de forme, la modélisation des trajectoires par 10 points de contrôle de B-splines uniformes périodiques est utilisée. La procédure de synthèse consiste en une modélisation similaire de la trajectoire désirée et une comparaison avec le contenu du catalogue. Cette comparaison est basée sur des critères assez discutables et peu fiables, tels que le plus petit et le plus grand angle formé entre deux segments successifs reliant les points de contrôle, la proportion entre la longueur et la largeur du rectangle englobant les points de contrôle, etc.

Kota et al. [KER87] [KER88] utilisent la théorie de la courbure des trajectoires (*path curvature theory*) et un catalogue de mécanismes pour extraire la connaissance nécessaire pour construire un système expert de conception. Le système est adapté à la synthèse des mécanismes de type *dwell* (pour lesquels l'orientation d'une barre varie très peu durant une certaine période), des générateurs d'arcs circulaires et de trajectoires symétriques.

Mayourian et Freudenstein [MF84] décrivent un catalogue de morphologies de mécanisme, les morphologies étant générées par une procédure automatique d'énumération de morphologies sous contraintes comme le nombre de degrés de liberté et le nombre de barres.

Bagci et Burke [BB93] utilisent un catalogue de trajectoires pour choisir la solution initiale dans un processus de minimisation des erreurs dans les équations de fermeture des boucles cinématiques.

Au lieu d'utiliser un catalogue de trajectoires unique et préalablement généré, la méthode de synthèse de trajectoire *avec indépendance du temps* de *Kota et Chiou* [KC93] utilise des plans d'expériences (*experimental design techniques*) afin de choisir la solution initiale pour l'algorithme d'optimisation. Autrement dit, un catalogue est généré pour chaque problème de conception. Pour chaque jeu de dimensions issu du plan d'expériences, la «similarité» de forme entre la trajectoire désirée et la trajectoire simulée est calculée. Cette mesure de similarité fait abstraction de la position, de l'orientation et du point de départ (les détails sur cette métrique de similarité sont donnés au §9.4.1). Cependant, la métrique n'est valable qu'en différentiel par rapport à une trajectoire désirée. Il n'est donc pas possible, avec cette approche, de réaliser une base de cas indépendamment de la trajectoire désirée.

L'approche proposée par *McGarva* [MM93] [McG94] utilise l'approximation des trajectoires par les coefficients de Fourier des premières harmoniques (voir les détails au §9.4.3). Une procédure de normalisation permet de s'abstraire des transformations géométriques affines (voir §9.4.4), éliminant ainsi les différences entre des trajectoires dues à des translations, rotations, changements d'échelle, etc. De cette manière, les coefficients normalisés représentent une classe de formes de trajectoires. Un catalogue informatique contenant un grand nombre de trajectoires, codifiées par les coefficients normalisés, est ensuite créé pour différentes dimensions du mécanisme. Pour la synthèse, les coefficients correspondant à la trajectoire désirée permettent de retrouver dans le catalogue les dimensions du mécanisme dont la forme de trajectoire générée est la plus proche de celle de la trajectoire désirée. Ces dimensions constituent ensuite une solution initiale pour l'optimisation dimensionnelle. Cette approche est à la l'origine de notre méthode de synthèse de formes de trajectoires par réseaux neuronaux artificiels (voir chap. 9).

2.5 L'apport de l'Intelligence Artificielle

2.5.1 Cinématique qualitative

La *cinématique qualitative* est un des domaines d'application de la *physique qualitative* (branche de l'Intelligence Artificielle), dont l'objectif est de résoudre certaines classes de problèmes insolubles avec les méthodes traditionnelles. Globalement, l'idée est de pouvoir directement mener une analyse cinématique (dite *qualitative*) sur une classe de mécanismes définie par une morphologie mais dont les dimensions restent non-valuées (intervalles, ordre de grandeurs...), ou éventuellement avec seulement une partie des paramètres valués. En contrepartie d'une moindre précision numérique, des raisonnements plus génériques peuvent ainsi être menés. Une des finalités serait d'exprimer les différentes classes de fonctions pour un mécanisme par des conditions sur ses dimensions, c'est-à-dire de diviser l'espace métrique en classes de fonctions. (Cf. les états de l'art de *Dague et al.* [DY95], *Travé-Massuyès et al.* [TMGY96] et *Yannou et Vasiliu* [YV95], références sur lesquelles s'appuie une partie de cette présentation. L'approche de cinématique qualitative de Yannou a été utilisée dans cette thèse – voir chap. 5, §8.3 et chap. 10.)

Les deux principales approches de cinématique qualitative de la littérature sont celles de *Joskowicz et al.* [JA88] [Jos89] [JS91] [NJ92] [Sac92] [JS93] [SJ93] et de *Faltings et al.* [Fal87] [Fal92] [FS93b] [FS93a] [FNF87] [FNF91]. L'approche de *Joskowicz et al.* étudie les mécanismes spatiaux à axes fixes. Celle de *Faltings et al.* aborde les mécanismes plans formés de deux pièces ayant cha-

cune un degré de liberté, comme roue dentée à cliquet [Fal92] [FS93a] ou mécanismes de horloge [FNF91]. Les principes de ces approches sont relativement similaires :

- les approches sont dédiées aux mécanismes à topologie variable et liaisons avec contact sur des géométries complexes (il s'agit donc d'une classe de mécanismes qui ne font pas l'objet de cette thèse) ;
- l'espace de configuration d'un mécanisme, c'est-à-dire la partie admissible de l'espace de phase des variables de positionnement des pièces[15], est défini par une combinaison d'informations qualitatives (symboliques) et quantitatives (numériques) ;
- une simulation cinématique qualitative consiste en une détermination des transitions d'état, c'est-à-dire des changements de topologies lorsqu'une liaison est établie ou disparaît.

L'approche de *Joskowicz et al.* (notamment *Sacks et Joskowicz* [SJ93]) utilise un partitionnement de l'espace de configuration en *états qualitatifs* (*region diagram* ou *places*) caractérisés par une topologie fixe (liaisons constantes). Un état qualitatif a des frontières qui correspondent à des contacts supplémentaires. Une transition est possible entre deux états s'il existe une frontière commune. En général, les formes complexes de frontière sont linéarisées par morceaux. (*Forbus et al.* [FNF91] considèrent uniquement des formes composées de segments et d'arcs de cercle.)

Le comportement dynamique qualitatif du mécanisme est déterminé en prenant en compte les directions des forces. La propagation des forces dans le mécanisme donne naissance aux mouvements. Un effort amène instantanément une pièce à se translater à vitesse constante ; il n'y a pas d'inertie. Les chocs sont considérés inélastiques et les pièces *collantes* ou *lisses*. Malgré ces hypothèses simplificatrices, une telle simulation est suffisante dans de nombreux cas. Cette méthode élimine la nécessité de détecter les collisions durant la simulation et de résoudre des équations différentielles.

L'espace de configuration du mécanisme représente une sorte de compilation de son comportement. En effet, le système d'équations ne doit pas être modifié à chaque changement de contact comme pour une simulation classique. De plus, des résultats symboliques sont générés, pouvant être comparés à une spécification symbolique du comportement attendu et produisant ainsi des conditions symboliques qui doivent être respectées pour répondre à cette spécification. Ce type d'approche peut servir de base à la compréhension automatique des relations entre la structure et les fonctions et, ainsi, contribuer à l'automatisation partielle de la conception.

Faltings et Sun [FS93a] explorent les relations entre les dimensions des pièces et leurs fonctions cinématiques. A partir d'un mécanisme avec des dimensions initiales, ils proposent une méthode de reconception pour modifier judicieusement les dimensions ou créer des modifications de forme, afin de respecter les fonctions requises et surtout d'éliminer les fonctions cinématiques indésirables. Un langage a été proposé pour représenter qualitativement les fonctions mécaniques par des expressions logiques reliant les forces et mouvements extérieurs avec des prédicats décrivant des comportements. Cette approche est une amélioration des travaux de *Faltings* [Fal92].

Le logiciel *FAMMING* de *Faltings et Sun* [FS96a] utilise cette modélisation qualitative dans le cadre d'une approche de synthèse basée sur le *raisonnement à base de cas*. Ces travaux sont présentés au §2.5.2.

Subramanian et Wang [SW93] proposent une méthode de synthèse de mécanismes spatiaux à topologie fixe et un degré de liberté, limitée aux morphologies pouvant être considérées comme un assemblage arborescent de mécanismes élémentaires avec axes fixes. Les mouvements d'entrée et de sortie de chaque mécanisme élémentaire sont définis par des spécifications cinématiques qualitatives

15. Deux pièces ne pouvant pas être à la même place au même moment, l'espace de configuration est l'espace où les pièces ne se chevauchent pas.

(rotation ou translation, mouvement constant ou alternatif...). Cette approche est une contribution intéressante au problème, encore très peu traité, de la synthèse conjointe morphologique et dimensionnelle, pilotée par les fonctions désirées globales et qualitatives.

Liu et al. [LIA95] partent du constat que, dans l'étape de préconception d'un mécanisme, l'information dimensionnelle numérique n'est pas encore disponible. Le modèle qualitatif, représentant les relations spatiales qualitatives entre les éléments, est déduit à partir du modèle physique du mécanisme (il s'agit essentiellement des mécanismes avec contact intermittent entre les éléments). Une méthode de simulation cinématique adaptée permet de vérifier la présence de fonctions désirées par le concepteur, avant même que les valeurs numériques des paramètres aient été choisies. Cette méthode est fondée sur la recherche des séquences possibles de changements d'états de contact entre les éléments.

Afin de raisonner en termes fonctionnels, *Liu et al.* [LTC96] utilisent pour la synthèse morphologique une modélisation de type qualitatif. Différents modèles de mécanismes existants, classifiés selon plusieurs points de vue fonctionnels, sont conservés dans une bibliothèque. Une heuristique *best-first* (du meilleur voisin) permet de construire, grâce à des règles de transformation, une proposition de solution morphologique répondant aux critères fonctionnels. Les types de fonctions représentés, classés de façon hiérarchique, sont : le type de conversion de mouvement et de vitesse, changement de direction de la vitesse, transformation d'axe, modification de ratio et de direction relative.

Une approche qualitative est utilisée par *Kochev* [Koc92] pour l'"optimisation des masses d'équilibrage, afin de minimiser les efforts maxima dans les liaisons. Cette approche utilise la décomposition en groupes d'Assur et une description qualitative des dépendances des efforts dans les liaisons appartenant aux groupes d'Assur par rapport à différents paramètres de masse. Cette modélisation qualitative contribue à guider la formulation des problèmes d'optimisation.

Les travaux de *Shrobe* [Shr93] sont dédiés aux mécanismes articulés plans à liaisons de type pivots et glissières. Cette approche se propose d'obtenir une *compréhension* des fonctions d'un mécanisme par la détermination d'une explication causale de la forme de la trajectoire du point effecteur. Pour cela, Shrobe mène un raisonnement géométrique, afin de lier de manière causale les caractéristiques de la trajectoire générée aux dimensions du mécanisme. Pour cela, la trajectoire générée – obtenue par une simulation classique – est rendue *qualitative* par une approximation en arcs de cercle et segments. Des détails sur cette approche seront donnés au §5.6.

Dans un esprit qui est proche de l'approche qualitative, les recherches de *Daniel et al.* [DD94] [DL96] portent sur la modélisation déclarative des courbes, une des applications possibles étant la spécification déclarative des trajectoires désirées dans les logiciels de synthèse de mécanismes. La forme des trajectoires est décrite par des termes du langage naturel (qualitatifs), qui sont ensuite « traduits » par une modélisation avec des splines.

2.5.2 Raisonnement à base de cas

Le *raisonnement à base de cas*, discipline de l'Intelligence Artificielle, est présenté dans l'annexe B. En résumé, cette approche regroupe les méthodes d'informatisation de la comparaison d'un problème nouveau avec des cas précédents, afin d'exploiter l'information contenue dans le cas pour l'analyse et la résolution de nouveaux problèmes. Quelques applications à la conception sont présentées ici. (Une approche de type raisonnement à base de cas a été également utilisée dans cette thèse, voir chap. 9 et 10.)

Navinchandra et al. [NSS91] et *Sycara et al.* [SNG+] modélisent les cas antérieurs, ainsi que les spécifications pour les produits à concevoir, par des *graphes d'influence causale* entre les paramètres. (Un exemple d'influence causale est le fait que l'augmentation d'un certain paramètre conduit à la diminution d'un certain autre.) Cette approche est implémentée dans le système *CA-*

DET de conception de schémas de principe pour les systèmes hydromécaniques par l'utilisation des conceptions antérieures (base de cas de 125 produits et composants). A partir d'une spécification fonctionnelle de haut niveau, sous forme de graphe d'influence, on cherche si ce graphe correspond, en totalité ou en partie, à une réalisation antérieure. Si ce n'est pas le cas, on réalise une transformation du graphe de spécification par la *loi série d'influence* (on remplace $A \rightarrow B$ par $A \rightarrow X \rightarrow B$) ou la *loi parallèle d'influence* en faisant intervenir de nouvelles variables. Puis, le processus de reconnaissance de graphe est relancé récursivememnt. L'utilisation de composants des conceptions antérieures laisse quand même place à l'innovation, car parfois certains de ces composants sont en dehors du domaine que le concepteur examine classiquement. Toutefois, la combinatoire étant immense, le système est, en pratique, piloté par le concepteur.

Wood et Agogino [WA96] présentent une nouvelle formulation du raisonnement à base de cas, adaptée à la conception de produits dans un contexte d'ingénierie simultanée. Au lieu d'une approche de conception automatique, il s'agit de privilégier l'interaction entre la base de cas et les concepteurs. Pour cela, la base de cas est accessible aux utilisateurs par une interface de type page *Web*, contenant de l'information hypertexte et multimédia. Des représentations selon différents points de vue et interconnectées sont utilisées. L'utilisateur peut effectuer des recherches *context-sensitive* de cas précédents et lancer des requêtes à niveaux d'abstraction multiples. Il peut également annoter la base de cas. L'indexation des cas est réalisée à la fois en langage naturel (multimédia) et par une base de données relationnelle.

Hua et al. [HFS96] présentent le logiciel *CADRE*, utilisant une approche de raisonnement à base de cas dans le domaine architectural et plus généralement pour le raisonnement géométrique spatial. Ce système emploie une modélisation des principes physiques, de la connaissance spécifique au domaine et des spécifications du cahier des charges dans les termes de la propagation de contraintes arithmétiques sur des variables continues. De cette manière, les fonctions sont reliées à la géométrie du produit. La prise en compte simultanée de contraintes mathématiques issues de différentes vues du produit permet d'intégrer divers critères de conception.

Faltings et Sun [FS96a] ont développé le logiciel *FAMMING* pour la conception innovante de la géométrie des liaisons de type contact ponctuel, utilisées notamment dans des mécanismes d'indexation de position. Le logiciel, basé sur le raisonnement à base de cas, utilise un catalogue de solutions codifiant le lien entre des éléments de structure et les fonctions. En effet, dans le domaine d'application choisi, il est possible de concevoir par assemblage de *briques fonctionnelles* (voir chap. 3). La modélisation des cas est réalisée sous une forme *qualitative* (cf. §2.5.1). L'article présente la formulation qualitative du cahier des charges, l'identification du/des cas précédent(s) le(s) plus proche(s) et l'adaptation de ce cas par une combinaison d'éléments de solutions.

L'approche de conception fonctionnelle décrite dans *Galletti et Giannotti* [GG81] utilise une approche semblable, en plusieurs points, au raisonnement à base de cas. Des informations morphologiques et fonctionnelles pour un certain nombre de mécanismes sont stockées dans un catalogue qui sert, ensuite, à retrouver des solutions potentielles pour le problème de conception. Cependant, la description fonctionnelle est assez pauvre, car il s'agit d'une simple description textuelle, qui ne permet pas, par exemple, la synthèse dimensionnelle de générateurs de trajectoires.

Les travaux sur la synthèse des générateurs de trajectoire par l'utilisation d'un catalogue de trajectoires (voir §2.4.7) s'inscrivent également dans l'optique du raisonnement à base de cas.

2.5.3 Réseaux neuronaux artificiels

Une introduction générale sur les réseaux neuronaux artificiels (RNA) est donnée dans l'annexe C. Ici seront présentées les applications des RNA à la conception de produits en général et des mécanismes en particulier. En effet, après l'essor connu par les applications des RNA à la reconnaissance de caractères et en général dans les problèmes de classification, de nombreux travaux ont

exploré les applications de conception et de modélisation de systèmes complexes. Dans cette thèse, l'approximation de fonction par réseaux neuronaux est utilisée au sein d'une méthode de synthèse basé sur les cas (voir chap. 9).

Un exemple d'application des RNA à la conception de produits est *Kumara et Ham* [KH90]. Les RNA y sont utilisés pour la conception basée sur la réutilisation de l'expérience des concepteurs et des solutions déjà validées. Pour concevoir un nouveau produit, le concepteur associe les fonctionnalités élémentaires connues avec des éléments de solution connus[16], le cas échéant après adaptation. L'apprentissage du réseau est réalisé avec l'algorithme de *rétropropagation* (voir la description au §C.2). Cependant, la représentation des solutions étant réalisée par des matrices de pixels, l'utilisation de cette méthode se limite aux produits pour lesquels les fonctions peuvent être associées à des formes élémentaires bien définies. De plus, le caractère discontinu à la fois de la représentation fonctionnelle et de la représentation structurelle laisse présager que l'utilisation un peu plus complexe d'une base de données relationnelle aurait pu donner le même résultat.

Une matrice de pixels est également utilisée par *Hoeltzel et Chieng* [HC90a]. A partir d'un échantillon de 356 trajectoires, discrétisées dans une matrice de 20×20 pixels, un réseau neuronal est utilisé pour classifier les trajectoires dans 15 catégories, selon des critères de forme. Ensuite, un algorithme permet de déterminer la catégorie à laquelle appartient la trajectoire désirée. De cette manière, des valeurs approximatives des dimensions du mécanisme sont déterminées. Cette solution est ensuite adaptée par une optimisation dimensionnelle classique.

Cook et Chiu [CC95] abordent l'utilisation des RNA, conjointement avec des techniques d'analyse statistique, pour l'estimation des paramètres optimaux de processus dans l'industrie chimique. (Une application à la synthèse de mécanismes peut être envisagée.)

Beaucoup de problèmes de conception peuvent être modélisés comme une approximation de fonction, surtout lorsqu'on est incapable de donner un modèle explicite et on doit se contenter de couples de valeurs des paramètres d'entrée et de sortie du système. De très nombreux travaux ont étudié la capacité des RNA à approximer des fonctions. *Goh* [Goh95] utilise un RNA de type *feedforward* et la rétropropagation afin d'approximer les interactions non linéaires entre des variables dans les systèmes complexes, pour lesquels il n'existe aucun modèle explicite (résistance au chargement des sols en génie civil, capacité de chargement des piles électriques).

Des travaux ont abordé le problème du choix de la *fonction d'activation* (voir la définition dans l'annexe C.1, p. 205). *Dasgupta et Schnitger* [DS93] montrent que la fonction sigmoïde à une capacité d'approximation comparable ou meilleure que certaines classes de fonctions utilisées dans la théorie de l'approximation (fonctions splines et polynomiales). Cependant, la question de savoir si cette capacité d'approximation se traduit par une bonne performance d'apprentissage reste ouverte.

Chen [CC91] analyse la capacité d'approximation de fonctions en $C(\overline{R}^n)$ par un RNA à trois couches avec des fonctions sigmoïdes dans la couche cachée. Dans un autre article, *Chen* [CC] traite de la construction d'un RNA pour l'identification de modèles de systèmes dynamiques.

Des recherches sont consacrées à l'étude de la structure optimale du RNA en fonction du problème d'approximation. *Nabhan et Zomaya* [NZ94] présentent une procédure de type *generate-and-test* pour configurer dynamiquement la structure d'un RNA *feedforward*, en ajoutant ou en effaçant des neurones ou des couches entières.

Murata [Mur94] présente un théorème pouvant servir à estimer la relation quantitative entre la précision d'approximation et le nombre de neurones dans la couche cachée.

Vysniauskas [Vys93] propose une méthodologie d'estimation du nombre optimal de jeux de données d'apprentissage et de couches cachées afin d'obtenir une certaine précision d'approximation. Les RNA *feedforward* multicouches sont une classe d'approximateurs universels du mapping

16. Voir aussi la méthode de conception par *briques fonctionnelles* (chap. 3).

entre deux espaces à dimensions finies. Durant l'apprentissage (par une optimisation des paramètres du réseau), le risque des minima locaux est très faible, grâce à la présence d'un très grand nombre de minima globaux, puisqu'il existe différentes permutations de coefficients synaptiques et changements de signe à valeurs de sortie constantes[17]. L'application est faite sur le problème de l'inversion de la fonction cinématique du robot anthropomorphe *OSCAR-6* (voir ci-après), comportant 6 variables articulaires angulaires (pour cette étude, seules trois variables sont indépendantes). Après apprentissage, le RNA permet de déterminer les déplacements angulaires nécessaires pour atteindre une certaine position dans l'espace de travail du robot.

Le même problème d'inversion cinématique du robot *OSCAR-6* est traité par *Groenewoud* [vhG95], *van der Smagt et Groen* [vdSG95] et *Jansen et al.* [JvdSG95]. Les RNA de type *feed-forward* ne permettent pas d'atteindre le niveau de précision requis pour les problèmes de grande dimension. Un modèle de RNA hiérarchique et multi-résolution est développé. La méthode est basée sur le partitionnement des espaces des variables d'entrée en sous-espaces. Un approximeur est créé pour chaque couple de sous-espaces. Le partitionnement optimal dépend de la répartition des non linéarités et des discontinuités ; il est réalisé par la construction dynamique d'un arbre d'approximeurs.. A la racine de l'arbre se trouve un RNA réalisant une approximation globale. Pour les sous-espaces pour lesquels l'approximation n'est pas assez précise, des nouveaux RNA sont dynamiquement construits. Ainsi, chaque niveau de l'arbre représente un niveau de résolution.

Teo et Sim [TS95] développent une méthodologie de plan d'expériences, à partir des techniques de Taguchi, afin de déterminer les paramètres d'un RNA à structure hiérarchique multi-couche capable de reconnaissance de caractères en faisant abstraction des transformations d'échelle et de rotation.

Heinz [Hei95] décrit un modèle de RNA structuré en arbre, particulièrement adapté pour les tâches de modélisation et de contrôle adaptif. Les paramètres de la structure évoluent pendant l'apprentissage (à partir d'un choix de valeurs initiales). Pendant l'utilisation, la topologie et certains paramètres sont figés, mais d'autres paramètres de sensibilité continuent à s'adapter à l'évolution des données.

17. Pour un RNA avec h neurones cachés sur une seule couche, il existe $h!2^h$ solutions.

Chapitre 3

Problématique, typologie des approches, approches adoptées

Dans le chapitre précédent, nous avons fait un état de l'art des travaux représentatifs, à différents titres, de la préconception des mécanismes articulés plans. Nous allons maintenant présenter d'une façon globale cette problématique de conception, ainsi que l'analyse que nous avons effectuée pour déboucher sur les approches utilisées dans cette thèse. Tandis que les chapitres suivants seront consacrés principalement aux détails techniques, le lecteur trouvera ici leur trame conceptuelle.

3.1 Cahier des charges fonctionnel de la thèse

L'étude scientifique d'un domaine n'est pas tenue a priori de respecter des objectifs d'ordre pratique, de remplir des fonctions prédéfinies. En particulier, le but d'une discipline comme la *théorie des mécanismes* est, globalement, de donner une description exacte et cohérente des lois qui régissent le fonctionnement des objets techniques que sont les mécanismes. Bien évidemment, la théorie des mécanismes débouche souvent sur des applications ; elle leur offre un fondement scientifique. Mais, nous semble-t-il, ce n'est pas directement et uniquement à la théorie des mécanismes que doit faire appel l'ingénieur lorsqu'il recherche les outils et les méthodes les plus appropriés pour résoudre ses problèmes de conception des mécanismes. La sélection des concepts et des théories les plus utiles, ainsi que leur « habillage » au sein d'une méthodologie informatisée interactive, doit faire l'objet d'une étude à part, s'inscrivant dans le domaine de la *théorie de la conception.* En ce sens, cette thèse s'inscrit plutôt dans le domaine de la théorie de la conception que dans celui de la théorie des mécanismes.

Or, le domaine de la conception a ses propres règles et priorités. La principale caractéristique d'une méthodologie de conception est d'être jaugée à l'aune des besoins concrets de l'utilisateur, c'est-à-dire du concepteur. Nous avons décidé, par conséquent ,d'évaluer la pertinence des méthodologies existantes et de celles que nous proposons, au regard des besoins des concepteurs. Pour ce faire, nous avons adopté une démarche d'*Analyse Fonctionnelle du Besoin*[1].

Le sujet de cette thèse est la conception automatique ou semi-automatique des mécanismes articulés plans dont la trajectoire d'un point effecteur correspond à une trajectoire désirée (*path generation problem*). Afin d'étudier cette problématique de conception, nous avons concentré notre analyse sur les besoins auxquels devrait répondre un logiciel de CAO dédié à la préconception des générateurs de trajectoire.

1. Ce domaine a fait l'objet d'une normalisation AFNOR, sous le numéro NF X50-151.

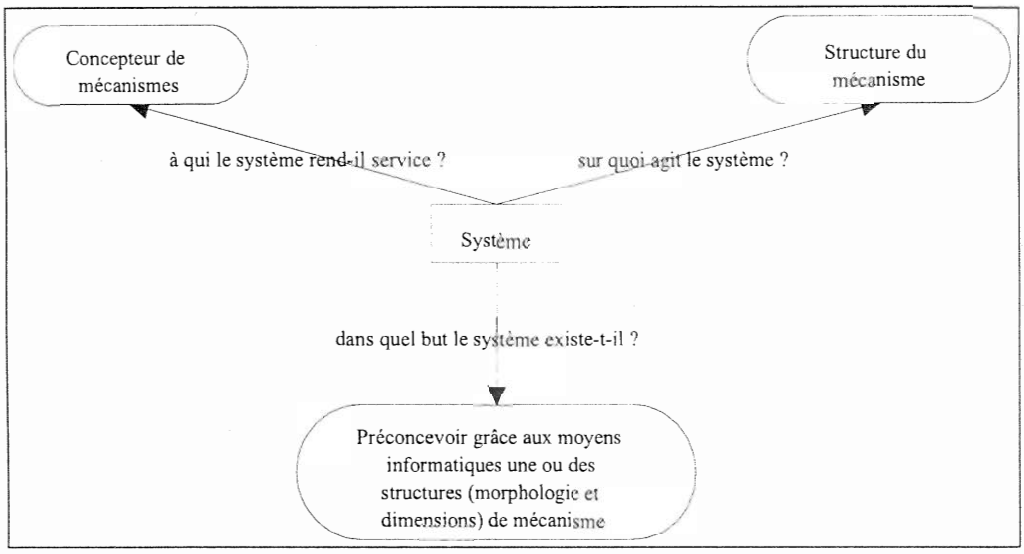


FIG. 3.1 – *La « bête à cornes » du travail de thèse.*

Conformément à la méthodologie d'analyse fonctionnelle, la première étape est la définition du besoin fondamental par l'outil de la « bête à cornes » (Figure 3.1), qui consiste à se poser les questions suivantes :

- *A qui le système rend-il service ?* Réponse : au concepteur de générateurs de trajectoire.
- *Sur quoi agit le système ?* Réponse : sur la structure du mécanisme.
- *Dans quel but le système existe-t-il ?* Réponse : « préconcevoir » grâce aux moyens informatiques une ou des structures (morphologie et dimensions) de mécanisme.

Cette « bête à cornes » s'accompagne d'un *contrôle de validité* du besoin fondamental qui consiste à répondre à trois autres questions, pour s'assurer de l'importance du système et de la pérennité de son besoin :

- *Pourquoi le système existe-t-il ?* Réponse : parce que les générateurs de trajectoire sont nécessaires dans l'industrie.
- *Dans quel but ce besoin existe-t-il ?* Réponses : pour résoudre des nouveaux problèmes de conception ; pour être compétitif (*délai, coût, qualité*) ; pour résoudre des aspects qui étaient jusqu'alors mal pris en compte : synthétiser conjointement la morphologie et les dimensions, prendre en compte au niveau de la spécification des caractéristiques de forme pure de trajectoire plutôt qu'une liste arbitraire de points.
- *Quelles sont les causes d'évolution ou de disparition de ce besoin ?* Réponses : à court terme, apparition d'une nouvelle méthode générale qui résolve les mêmes objectifs ou remplacement économiquement avantageux des générateurs de trajectoire par des robots.
- *Quelles sont les insatisfactions actuelles pour ce type de système ?* Réponses : difficulté de synthétiser une trajectoire passant par un grand nombre de points ou définie par sa *forme*, synthèse morphologique inefficace, manipulation lourde pour l'analyse, mauvais traitement des blocages et des positions singulières en cours de simulation.

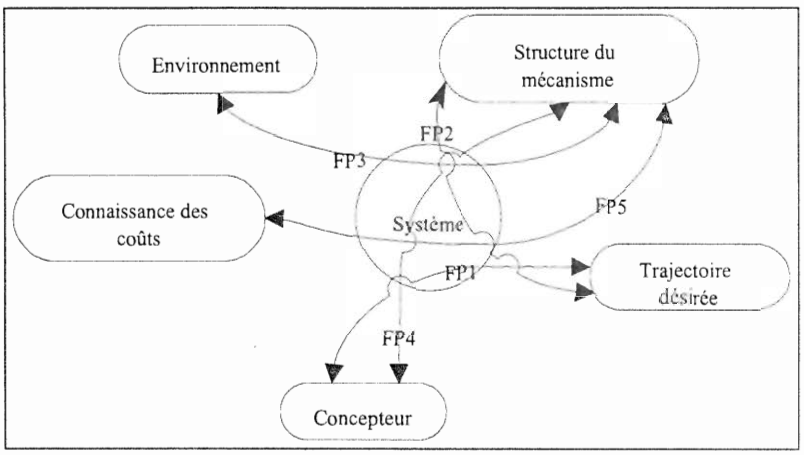


FIG. 3.2 – *Détermination des fonctions de service d'une méthodologie de préconception des générateurs de trajectoire (par la « pieuvre » de la méthode* APTE*).*

La deuxième étape de l'analyse fonctionnelle est l'*identification des fonctions de service* que doit remplir un système de préconception des générateurs de trajectoire. Cela passe par l'établissement de la liste des milieux extérieurs au système qui sont, dans une phase quelconque d'utilisation du système, en rapport avec celui-ci. Nous avons distingué cinq éléments du milieu externe : le concepteur, la trajectoire désirée, la structure du mécanisme conçu, l'environnement de fonctionnement du mécanisme et la connaissance des coûts des mécanismes (prix de revient). Les fonctions de service du système sont des services que rend le système entre deux milieux extérieurs (*fonctions principales*) ou à un milieu seul (*fonctions contraintes*). Nous avons décliné ces fonctions de service par le moyen d'un outil graphique dénommé la « pieuvre », intégrée dans la méthode *APTE* (voir Figure 3.2).

Les fonctions principales sont :

FP1 : Permettre au concepteur de représenter la fonction cinématique désirée.

FP2 : Concevoir la structure d'un ou des mécanismes répondant au mieux à la fonction cinématique désirée.

FP3 : Concevoir un mécanisme qui puisse évoluer dans un environnement donné.

FP4 : Permettre de concevoir rapidement un mécanisme.

FP5 : Avoir une estimation du coût global des mécanismes solutions.

Par rapport aux trois objectifs globaux de tout développement de produit industriel, *Qualité*, *Coût* et *Délai*, les fonctions principales *FP2* et *FP3* contribuent à la *Qualité* du produit, la fonction *FP4* est essentielle pour respecter le *Délai* et la fonction *FP5* est impliquée dans l'objectif de réduction du *Coût*.

Par la suite, chaque méthodologie que nous envisagerons devra être confrontée en regard des fonctions de service. L'adéquation d'un moyen particulier avec les objectifs se fera par l'intermédiaire d'un diagramme *FAST* (*Function Analysis System Technique*) qui établit les liens logiques entre causes (besoins) et moyens (solutions méthodologiques). Un diagramme *FAST* consiste à partir de chaque fonction de service et à étendre de manière récursive le graphe, en se posant de proche en proche pour chaque fonction :

- en allant vers la droite, la question *Comment* ? La réponse doit commencer par *En...* et correspond à un *moyen technique*.

	Analyse	Conception incrémentale	Synthèse
Espace de solution	à un élément (la solution candidate)	réduit (autour d'une solution initiale)	grand
Connaissance préalable de la solution	complète	approximative	aucune
Difficulté	réduite	moyenne	grande
Qualité de la conception	dépend de la solution candidate	dépend de la solution initiale	grande

TAB. 3.1 – *Comparaison des trois activités de conception : l'analyse, la conception incrémentale et la synthèse.*

- en allant vers la gauche, la question *Dans Quel But?* La réponse doit commencer par *Pour...* et correspond à une *cause*.

Ainsi, la Figure 3.3 présente l'analyse fonctionnelle selon le formalisme des diagrammes *FAST* d'un système de conception des générateurs de trajectoire. Ce schéma comprend les fonctions principales ainsi que les principaux *moyens techniques* adoptés ou développés dans cette thèse. Pour des raisons de lisibilité, certains liens n'ont pas été représentés sur ce schéma ; les cases correspondant aux principales contributions de la thèse sont marquées par des doubles traits verticaux.

Trois façons de réaliser le besoin fondamental *FP2*, correspondant à trois activités distinctes, y sont distinguées :

- *analyser* (lorsqu'il existe une solution candidate qu'il faut simplement tester),
- *concevoir de manière incrémentale* (lorsque le concepteur dispose d'une solution initiale qu'il s'agit de modifier ou d'adapter au cahier des charges),
- *synthétiser* (lorsque la recherche de la solution est réalisée à partir du cahier des charges, aucune connaissance sur la solution n'étant disponible au début de la conception).

Dans le Tableau 3.1 nous comparons ces trois activités par rapport à trois critères : la taille de l'espace de solutions considéré, la « quantité » d'informations sur la solution disponibles au début de l'activité et le degré de difficulté.

A proprement parler, l'*analyse* n'est pas une activité de conception, elle est simplement une étape de vérification suivant l'obtention de la solution candidate. Toute l'information sur le mécanisme à tester est disponible au début du processus, ce qui rend l'analyse relativement facile. En revanche, l'analyse n'a qu'un apport indirect à la recherche de solution ; la qualité du résultat final dépend intégralement de la qualité de la solution prise en compte.

La complexité de la *conception incrémentale* est plus grande, puisque le concepteur doit naviguer dans un espace de solutions autour de la solution initiale. Même si la qualité de la conception est toujours influencée par le choix de la solution initiale, la conception incrémentale permet de rechercher un optimum local par rapport au cahier des charges.

Finalement, la *synthèse* consiste en à rechercher la solution dans l'espace complet, défini par tous les choix de conception possibles et limité seulement par les contraintes du cahier des charges ou d'autres contraintes implicites. Evidemment, plus l'espace de solutions est grand, plus complexe est la recherche de la solution, mais en revanche il existe plus de chances de trouver une solution optimale par rapport au cahier des charges.

Dans la suite de ce chapitre, nous allons étudier plus en détail ces trois activités, en détaillant la décomposition fonctionnelle. Il est à remarquer qu'il n'existe pas, dans notre cas, d'autres possibilités pour réaliser la fonction *FP2*. Le premier niveau de décomposition est ainsi le seul

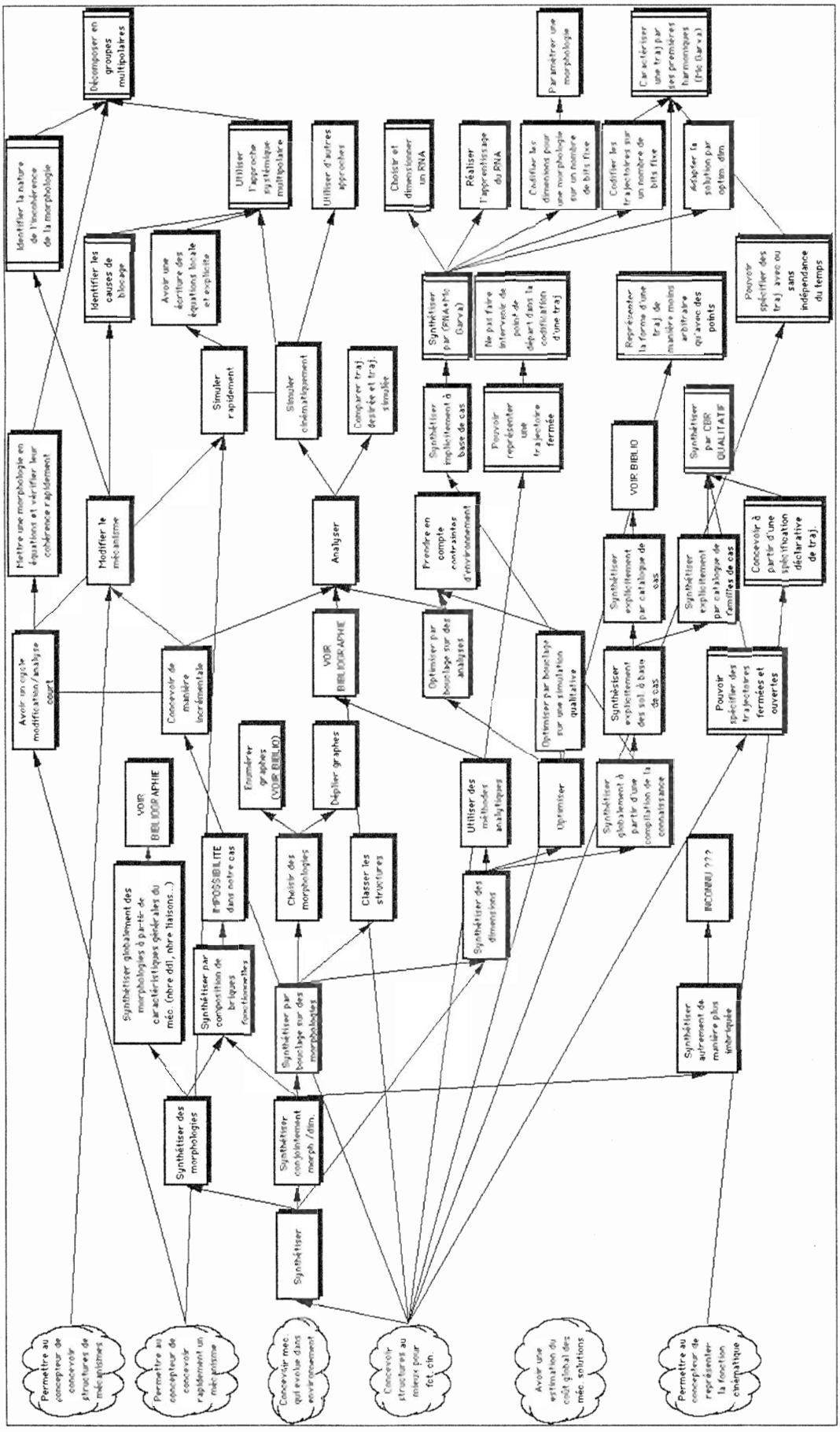


FIG. 3.3 – *Analyse fonctionnelle (diagramme* FAST*) de la préconception des mécanismes plans.*

possible (à des variations de vocabulaire près). En revanche, lorsque la décomposition avance vers les solutions techniques concrètes, le caractère absolu et unique de la décomposition diminue. En effet, la décomposition s'inspire forcément, dans une certaine mesure, des éléments de solutions existants ou connus et des partis pris méthodologiques.

Néanmoins, même si l'on ne peut prétendre à une rationalisation absolue du choix des techniques, ce type de démarche est préférable à une démarche qui ne considérerait que d'une manière subsidiaire l'aspect fonctionnel, puisqu'il permet de garder à l'esprit la cohérence globale entre fonctions et solutions. De plus, même lorsqu'une partie des choix techniques ont été réalisés autrement que par une analyse fonctionnelle (par exemple, disponibilité d'une méthode originale, pressentie efficace), replacer ces techniques dans le cadre d'une analyse fonctionnelle permet, bien qu'a posteriori, de vérifier leur adéquation avec les besoins.

3.2 Analyse versus synthèse

Du point de vue du domaine de recherche qu'est la *conception* en Intelligence Artificielle, un mécanisme est une *structure* physique qu'il faut assembler (*topologie*) et dimensionner (*dimensions*). La trajectoire désirée est la *fonction* comportementale que doit réaliser la structure de la meilleure manière possible. Ce problème de la conception automatique de mécanismes est celui de la *synthèse* d'une structure physique à partir d'une fonction spécifiée sous la forme d'une trajectoire. Le problème peut être représenté par le schéma suivant :

$$\textbf{Fonction} \xrightarrow{\textbf{Synthèse}} \textbf{Structure(topologie, dimensions)}$$

Il est à remarquer que la détermination de la fonction comportementale d'une structure, ou l'*analyse*, se fait par l'intermédiaire d'outils de simulation comportementale, mais chaque simulation est souvent coûteuse en temps de calcul :

$$\textbf{Structure(topologie, dimensions)} \xrightarrow{\textbf{Simulation}} \textbf{Fonction}$$

La synthèse morphologique et dimensionnelle des générateurs de trajectoire présente des difficultés particulières. Pour la synthèse morphologique, nous disposons de peu de connaissances explicites sur le lien entre la morphologie et les trajectoires générées. De plus, on assiste à une explosion combinatoire du nombre de solutions. En effet, la croissance du nombre de morphologies distinctes avec le nombre de solides composant le mécanisme est exponentielle (voir Tableau 3.2, tiré de *Hwang* [HH92]).

En revanche, le plus souvent le concepteur s'intéresse prioritairement aux mécanismes d'une complexité morphologique réduite, dont la conception, la réalisation et l'exploitation sont moins chères. Il reste néanmoins un espace de solutions assez étendu. De plus, la pertinence d'une solution morphologique ne pourra être complètement évaluée qu'après le choix des dimensions.

Quant à la synthèse dimensionnelle, le nombre de variables de conception est, en principe, le nombre minimal de paramètres nécessaires pour définir complètement la géométrie du mécanisme. Ce nombre peut être encore réduit en choisissant a priori les valeurs d'une partie de ces paramètres, mais cela restreint l'étendue de l'espace de solutions et diminue la qualité de la solution finale. Dans le cas où la trajectoire désirée est spécifiée en termes de forme et non pas en termes de points de passage obligatoires (ce type de synthèse fait l'objet des chapitres 9 et 10), l'emploi des méthodes de synthèse décrites dans ces chapitres permet une réduction du nombre de variables sans diminution de l'optimalité de la solution (voir §9.5.2.1 et §10.4.2.1).

Une autre source de difficulté pour la synthèse dimensionnelle est la relation très complexe entre les paramètres dimensionnels et les trajectoires générées. Les trajectoires sont extrêmement sensibles par rapport aux valeurs des paramètres dimensionnels : une petite variation de longueur

Nombre de pièces	Degrés de lib.	Nombre de chaînes cinem. planes
6	1	2
7	2	4
8	1	16
	3	7
9	2	40
	4	10
10	1	230
	3	98
	5	14
11	2	839
	4	189
	6	19

TAB. 3.2 – *Croissance du nombre de chaînes cinématiques distinctes avec le nombre de solides composant le mécanisme.*

de barre peut complètement modifier le comportement cinématique et dynamique d'un mécanisme. La relation entre la fonction – qu'elle soit exprimée par la suite de points composants la trajectoire désirée ou par sa forme – et les dimensions est fortement non linéaire. De plus, certaines zones de l'espace des paramètres ne sont pas faisables, dans le sens où le mécanisme ne décrit pas une trajectoire fermée. Inversement, pour une morphologie fixée, l'existence d'une solution dimensionnelle permettant de générer une trajectoire donnée n'est pas garantie. Une dernière caractéristique est la présence d'un très grand nombre d'optima locaux.

Pour revenir à l'analyse fonctionnelle, la synthèse de mécanismes se divise traditionnellement en une étape de synthèse morphologique et une étape de synthèse dimensionnelle. Comme il a été dit précédemment la synthèse dimensionnelle des générateurs de trajectoire devra nécessairement succéder à une synthèse morphologique préalable. Une démarche de synthèse conjointe morphologique et dimensionnelle ne pourra donc pas consister en une intégration intime des deux étapes. Cependant, il est possible de les intégrer dans une méthode de préconception qui facilite la prise en compte de plusieurs morphologies candidates et la comparaison des performances. Par conséquent, la synthèse morphologique sera effectuée en examinant successivement des morphologies cohérentes dans l'ordre croissant de leur complexité ou coût estimé (cf. Figure 3.3). Quant à la synthèse dimensionnelle, nous avons choisi de nous intéresser à l'optimisation dimensionnelle et non pas aux méthodes analytiques de synthèse. Ces méthodes, largement traitées dans la littérature (cf. l'état de l'art au §2.4.2), consistent à établir et à résoudre – en fonction d'un nombre réduit de points de passage du point effecteur – un système d'équations de synthèse. L'inconvénient majeur de cette classe de méthodes est la limitation drastique du nombre de points de passage. Ce nombre dépend de la morphologie du mécanisme et du nombre de variables dimensionnelles fixées préalablement. Dans les méthodes d'optimisation dimensionnelle, l'objectif est de déterminer les dimensions pour lesquelles le mécanisme génère une trajectoire approximant au mieux la trajectoire désirée. Les deux types d'approches d'optimisation dimensionnelles explorés dans cette thèse font l'objet des sous-chapitres §3.7.1 et §3.7.2.

3.3 Analyser

Le but de l'analyse est de valider l'adéquation d'une solution par rapport au cahier des charges. Dans notre cas, conformément au domaine fixé pour cette thèse, il s'agit de l'analyse cinématique

(éventuellement aussi en dynamique inverse) de mécanismes générateurs de trajectoire dont la morphologie et les dimensions sont connues. Deux étapes sont indispensables : l'obtention de la trajectoire générée par le mécanisme et la comparaison de cette trajectoire avec la trajectoire désirée (cf. Figure 3.3). A son tour, obtenir la trajectoire implique une étape de modélisation du mécanisme, afin de pouvoir le manipuler dans le logiciel de CAO, suivie par l'étape de simulation cinématique. Ensuite, le choix de la méthode des groupes multipolaires (voir chap. 4) impose les étapes : décomposition multipolaire, bouclage sur les configurations successives du mécanisme pour un tour du moteur. La détermination d'une configuration consiste à vérifier toutes les contraintes géométriques pour chaque groupe.

En ce qui concerne le besoin d'analyse, les logiciels actuels sont le plus souvent basés sur des approches théoriques de la mécanique générale. Le souci de généralité (mécanismes 3D, prise en compte des éléments flexibles, des frottements, des jeux etc.) et la complexité inhérente des modèles font de ces logiciels des instruments très puissants. Mais, pour une partie des applications et surtout en phase de préconception, cette généralité et la richesse des résultats fournis ne sont pas toujours nécessaires et risquent de nuire à la simplicité d'emploi et à l'efficacité. Le choix de la méthode multipolaire est une réponse à ce constat (pour les détails, voir §2.1.2 et chap. 4).

3.4 Concevoir de manière incrémentale

Il est très fréquent, dans la pratique de la conception, de partir d'une solution existante et de lui apporter des modifications pour l'adapter à un besoin nouveau. Cela se traduit, pour la conception de mécanismes (cf. Figure 3.3), dans des cycles comportant une phase de modification (dimensionnelle et/ou morphologique) et une phase d'analyse (déjà présentée). Le choix des techniques utilisées sera ensuite guidé par la nécessité de rendre ce cycle le plus court et le plus simple possible. En effet, pouvoir essayer facilement un grand nombre de solutions conduit à une meilleure qualité du mécanisme conçu.

Ainsi, la modification rapide du mécanisme peut être obtenue par l'emploi du schéma cinématique comme support graphique de la modélisation du mécanisme, permettant d'apporter toutes les modifications d'une manière graphique, à la souris, par rajout ou effacement d'éléments et par le déplacement des liaisons.

Afin de rendre le plus rapide possible la simulation cinématique, les techniques choisies devront permettre d'écarter au plus tôt des solutions incohérentes cinématiquement, de réduire au minimum le temps de mise en équations et les calculs itératifs. Plus en détail, ces objectifs peuvent être réalisés par une écriture locale des équations et l'emploi d'équations explicites, la simulation cinématique par la méthode multipolaire étant une réponse appropriée à tous ces critères.

La méthode systémique multipolaire, utilisée au cœur du logiciel de préconception, présente également l'avantage de permettre une exploration maîtrisée et raisonnée des morphologies. (Voir aussi chap. 6.)

3.5 Synthétiser par composition de briques fonctionnelles

Les approches de synthèse structurelle se divisent en deux grandes catégories selon que la connaissance mise en jeu puisse ou pas se décomposer en briques fonctionnelles et que toute structure puisse être la résultante de la composition de plusieurs briques fonctionnelles. Il existe une brique fonctionnelle lorsqu'on est capable d'associer une sous-fonction à une partie de la structure (un composant ou un sous-ensemble de composants). Une brique fonctionnelle se caractérise donc par un couple :

(sous-fonction, ss-ens de composants)

La composition de briques fonctionnelles ($\mathbf{f}_i$,$\mathbf{comp}_i$) consiste en la connexion des sous-ensembles de composants $\mathbf{comp}_i$ par des liaisons en mécanique, des tuyaux en hydraulique, des fils électriques en électronique, etc. D'un assemblage de briques structurelles doit pouvoir émerger une fonction résultante. Il faut donc pouvoir déterminer cette fonction résultante directement à partir des fonctions des briques élémentaires et de la topologie de la connexion :

$$\textbf{fonction-résultante} = F(\mathbf{f}_1, \ldots, \mathbf{f}_n, \textbf{topologie-connexions})$$

Cela est donc beaucoup plus rapide que de déterminer la fonction résultante par simulation (ou analyse) :

$$(\mathbf{comp}_1, \ldots, \mathbf{comp}_n, \textbf{topologie-connexions}) \xrightarrow{\text{Simulation}} \textbf{fonction-résultante}$$

D'autre part, toutes les approches de synthèse structurelle consistent à déterminer l'algorithme de composition incrémentale des briques qui soit le moins « explosif » possible, c'est-à-dire qui trouve en des temps raisonnables une ou des solutions structurelles ou qui prouve qu'il n'en existe pas de simple. Toute l'astuce de ces algorithmes est d'essayer de composer les briques fonctionnelles le moins « à l'aveuglette », en pressentant que la fonction de la structure en cours de conception s'approche de la fonction désirée.

La possibilité d'assembler les briques fonctionnelles repose sur le principe « *no function in structure* » (cf. *de Kleer et Brown* [dKB84]). Ce principe signifie que la connexion des briques fonctionnelles entre elles n'introduit pas une complexité trop importante dans la fonction résultante au point qu'il n'existe pas de connaissance explicite pour la déterminer à partir des fonctions élémentaires et qu'il faille donc simuler le système pour la déterminer. Cette simulation signifierait qu'une bonne part de la fonction résultante proviendrait de la structure de connexion plutôt que des sous-ensembles de composants eux-mêmes ; c'est le sens du principe « *no function in structure*».

L'existence de briques fonctionnelles composables dépend de la nature de la connaissance du domaine et du type de fonction. Elle existe dans le domaine de l'hydraulique, en régime établi, pour des fonctions de type température ou débit ; dans le domaine de l'électronique, il s'agit de fonctions de type intensité ou tension. Il existe toutefois une exception en électronique de puissance où les influences magnétiques d'un composant sur un autre, indépendamment même de la connexion logique des briques structurelles, dépend de la proximité des composants. En électrodynamique, par contre, où on s'intéresse aux phénomènes dynamiques transitoires, les comportements dynamiques des variables aux bornes d'une brique structurelle sont issus de la résolution globale d'un système d'équations différentielles. Par conséquent, le principe « *no function in structure*» n'est pas respecté et la synthèse par composition de briques fonctionnelles n'est donc pas possible.

Il en est bien sur de même en mécanique multi-corps (des mécanismes) où les fonctions dynamiques ne respectent pas le principe « *no function in structure* ». Examinons maintenant le domaine de la cinématique qui nous intéresse le plus. Là, le problème est différent de celui de l'électrodynamique. Le vrai problème est qu'il n'existe, la plupart du temps, même pas de notion de brique fonctionnelle. En effet, il faudrait être capable d'associer dans une brique un ensemble de composants mécaniques à une sous-fonction cinématique désirée du mécanisme.

Dans notre cas de figure, la fonction cinématique étant une trajectoire, il n'est pas possible d'affecter une fonction contributive à la trajectoire finale à un composant mécanique (par exemple une barre ou une liaison) ou à un sous-ensemble de composants[2]. D'autre part, la trajectoire d'un point effecteur est issue de la topologie même du mécanisme car une simulation met en jeu une résolution globale sur le mécanisme ; le principe « *no function in structure* » n'est pas non plus

2. Même en considérant les mécanismes comme formés de parties structurales clairement identifiées grâce à la décomposition en groupes multipolaires (voir chap. 4), le couplage fonctionnel entre les groupes est extrêmement fort.

respecté. Il est donc clair que nous ne pourrons pas faire de synthèse par composition de briques fonctionnelles pour les mécanismes articulés.

En plus du couplage fonctionnel entre les parties structurales, il existe un très fort couplage entre la morphologie et les dimensions du mécanisme. Par conséquent, pour les générateurs de trajectoire, un raisonnement fonctionnel portant sur la morphologie indépendamment des dimensions n'est pas possible[3].

Des exceptions existent toutefois si on se limite à la synthèse de classes de mécanismes qui nous semblent assez restreintes. Le prix à payer pour avoir des briques fonctionnelles et respecter le principe « *no function in structure* » semble être de ne considérer que des classes particulières de mécanismes et de cahiers des charges. Ainsi, une approche de synthèse basée sur le principe « *no function in structure* » a été proposé par *Freudenstein et Maki* [FM79] pour des mécanismes répondant à d'autres types de cahiers des charges. (Les travaux qui reposent sur ce principe proposent souvent comme critère de choix des morphologies le nombre de degrés de liberté, le nombre de circuits indépendants ou d'éléments, le nombre de liaisons fixes, le type de conversion de mouvement, le nombre de points de passage imposés etc. Voir par exemple *Soni et al.* [SDW88], *Chew et al.* [CSI95], *Bohatier et al.* [BGT92], *Yang et al.* [YDDK91], *Kusiak et Szczerbicki* [KS92].)

Le plus souvent, il s'agit des mécanismes élémentaires (briques fonctionnelles) ayant les axes d'entrée et de sortie fixes par rapport au bâti. C'est le cas dans *Subramanian et Wang* [SW93] et *Fauroux et Sartor* [FS96b].

Dans *Subramanian et Wang*, les mécanismes élémentaires (briques) ont un arbre d'entrée et un ou plusieurs arbres de sortie ; ce sont des engrenages, des bielle-manivelle, des pignon-crémaillère, des roues et vis sans fin...

Les fonctions cinématiques sont composées de la nature des mouvements d'entrée et de sortie (rotation ou translation, vitesse constante ou alternative), de leur quantification (valeur de vitesse ou fréquence d'oscillation), de la nature des transformations de mouvement (linéaire, non linéaire).

Fauroux et Sartor ne considèrent que des mécanismes élémentaires à un arbre d'entrée et un arbre de sortie (engrenages intérieurs, extérieurs, renvois d'angle, roues et vis sans fin) et s'intéressent plus spécifiquement à la synthèse de réducteurs. Ils intègrent dans les fonctions des mécanismes leur encombrement et le vecteur relatif de l'arbre de sortie par rapport à l'arbre d'entrée. Ainsi, la synthèse d'un réducteur se fait avec des contraintes d'encombrement donné. Aucune solution satisfaisante n'est donnée dans les deux approches pour limiter l'explosion combinatoire de la composition des briques fonctionnelles.

3.6 Synthétiser à partir d'une compilation de la connaissance

3.6.1 Principe

Dans le cas où la synthèse par composition de briques fonctionnelles est interdite, il ne reste comme solution que d'essayer de connaître de manière globale le lien qui existe entre la structure et la fonction désirée. C'est donc notre cas en cinématique plane pour une fonction trajectoire. On est en droit de penser qu'on a une bonne connaissance de ce lien fonction/structure à partir de la connaissance d'un certain nombre de couples (**Structure**$_i$, **Fonction**$_i$) dénommés cas (de conception). Les fonctions sont, bien entendu, déterminées par simulation (analyse) :

3. Cette conclusion apparaît fort souvent dans la littérature. Cf. par exemple *Rao* [Rao95] : « *Exact comparison of chains even for kinematic or static performance is impossible without knowing the actual dimensions.* ».

$$\begin{array}{c} \textbf{Structure}_1 \xrightarrow{\textbf{Simulation}_1} \textbf{Fonction}_1 \\ \textbf{Structure}_2 \xrightarrow{\textbf{Simulation}_2} \textbf{Fonction}_2 \\ \cdots \\ \textbf{Structure}_n \xrightarrow{\textbf{Simulation}_n} \textbf{Fonction}_n \end{array}$$

Il s'agit en fait de mémoriser un certain nombre de « photographies » de ce qui est faisable et de savoir l'utiliser au mieux. Le remplissage d'une base de cas a des chances d'être extrêmement coûteux. Se pose aussi le problème du stockage informatique du cas :

$$\left.\begin{array}{c} (\textbf{Structure}_1, \textbf{Fonction}_1) \longrightarrow \\ (\textbf{Structure}_2, \textbf{Fonction}_2) \longrightarrow \\ \cdots \\ (\textbf{Structure}_n, \textbf{Fonction}_n) \longrightarrow \end{array}\right\} \text{BASE DE CAS}$$

Une fois la base de cas créée, il faut être capable de synthétiser une structure cohérente apte à répondre à une fonction désirée :

$$\textbf{Fonction} \xrightarrow{\text{BASE DE CAS}} \textbf{Structure}$$

Dans de nombreux domaines où la synthèse directe d'une solution est difficile ou impossible, l'utilisation pour la conception d'un nouveau produit d'une mémoire des produits déjà conçu est une alternative intéressante ou même une pratique courante. Depuis plus de 10 ans, un domaine de l'intelligence artificielle s'est peu à peu délimité, ayant comme objet la formalisation et l'informatisation des méthodes basées sur une mémoire des produits ou de cas antérieurs : le *raisonnement à base de cas (case-based reasoning)*. (Voir une présentation de ce domaine dans l'annexe B). La conception des générateurs de trajectoire se prête particulièrement bien à ce type de démarche, car le nombre de cas disponibles est en principe infini : il suffit de les générer par simulations cinématiques, sans qu'il soit nécessaire de se limiter, comme dans d'autres domaines, à l'enregistrement des produits réellement conçus dans le passé.

Un deuxième facteur qui incite à l'utilisation de cette approche est sa compatibilité avec les pratiques réelles de conception. En effet, pour une partie des problèmes de conception (surtout pour les générateurs de trajectoire avec indépendance du temps), les concepteurs utilisent depuis longtemps des catalogues de trajectoires (cf. l'état de l'art, §2.4.7).

A partir du principe général de l'utilisation d'un ensemble de cas, nous proposons deux méthodes de synthèse de générateurs de trajectoire avec indépendance du temps : la synthèse par interpolation entre les cas et la synthèse par sélection dans une base de cas qualitatifs (cf. Figure 3.3).

3.6.2 Le remplissage de la base de cas

Pour avoir une chance de capter de manière exhaustive la complexité du lien entre la fonction et la structure, il faut être attentif à de nombreux paramètres :

- Il faut considérer un nombre suffisant de cas ; ce nombre dépend de :
  - la dimension de l'espace des structures,
  - la dimension de l'espace des fonctions,
  - la progressivité dans l'évolution des fonctions en rapport avec celles des structures (bonne continuité ou pas).

- Il faut de ce fait, minimiser les dimensions des espaces des structures et des fonctions. Ceci n'est pas toujours très simple. Pour les structures par exemple, il faut tenter de choisir un paramétrage minimal de la structure. De manière à s'abstraire des similitudes dans les couples (**Structure**, **Fonction**), ce qui ne ferait que croître inutilement la base, on est souvent amené à effectuer des opérations de normalisation. Pour une structure par exemple, sa fonction ne dépend pas de sa taille (échelle) ni de sa position.

- Il faut que les cas soient suffisamment représentatifs des fonctions réalisables. Se pose alors le problème de la génération des cas : les structures doivent-elles être générées aléatoirement ou de manière raisonnée? La réponse dépend de la connaissance heuristique du domaine.

3.6.3 Raisonnement à base de cas élémentaires

Nous avons déjà évoqué l'existence des catalogues de trajectoires et leur utilisation pour la synthèse des générateurs de trajectoires. Le plus souvent, ces catalogues comportant des exemples de trajectoires et les dimensions du mécanisme correspondant sont édités sur support papier, plus rarement informatisés. Dans le cas du catalogue papier, le concepteur choisit visuellement dans le catalogue une trajectoire semblable à la trajectoire désirée et récupère le mécanisme l'ayant générée. Eventuellement, cette solution pourra être ensuite peaufinée, soit par essai-erreur soit par une optimisation proprement-dite à partir de cette solution initiale. Dans le cas du catalogue informatisé, traditionnellement il s'agit simplement de rendre plus aisée la consultation et l'exploitation du catalogue. (Cf. l'état de l'art, §2.4.7)

Cette démarche de conception peut être assimilée, conceptuellement, à un *raisonnement à base de cas élémentaires*. En effet, les catalogues sont un empilement de cas sans aucune transformation ou indexation des cas, simplement par l'enregistrement des trajectoires issues des simulations cinématiques et représentées par des successions de points. S'ils peuvent, bien entendu, aider le concepteur dans une certaine mesure, ils ne sont certainement pas la seule façon possible de mettre à profit une mémoire de trajectoires pour la conception des générateurs de trajectoire. Les catalogues de trajectoires présentent notamment deux désavantages : l'impossibilité de réaliser directement une interpolation entre les cas et la difficulté de comparer automatiquement les différents cas du point de vue fonctionnel, c'est-à-dire du point de vue de la forme des trajectoires.

3.6.4 Raisonnement à base de familles de cas

A cause de l'extrême variété des trajectoires qu'un même mécanisme peut générer, un bon catalogue de trajectoires doit contenir un très grand nombre de cas. Par conséquent, d'une part, la quantité d'informations à stocker pour chaque cas doit être la plus petite possible. D'autre part, la modélisation choisie pour les cas (autrement dit le type d'informations enregistrées) doit faciliter la recherche de la plus proche trajectoire par rapport à la trajectoire désirée. Pour ces deux raisons, l'utilisation des *cas élémentaires*, c'est-à-dire la modélisation des trajectoires par les coordonnées d'une suite de points, n'est pas optimale.

Dans ce contexte, la modélisation *qualitative* des trajectoires (cf. chap. 5) pour la constitution d'une base de cas est une option intéressante. Cette modélisation qualitative consiste en une approximation des trajectoires en arcs de cercles. En effet, premièrement, l'erreur commise par l'approximation en arcs de cercles est compatible avec le fait que l'utilisation d'un catalogue de trajectoires conduit de toute façon à une solution approximative. D'ailleurs, le concepteur n'a souvent qu'une idée approximative de la trajectoire désirée et il est dommageable de l'obliger à formuler sa requête en termes exacts. Deuxièmement, la trajectoire qualitative est beaucoup plus compacte que la trajectoire exacte, spécifiée par une liste de points. Troisièmement, la trajectoire qualitative élimine une source de complexité inutile pour la synthèse de forme, qui est due à la représentation

(non fonctionnelle dans ce cas) des trajectoires par une liste de points. Il devient plus facile, de cette manière, d'aborder le problème de synthèse le plus difficile : la génération de trajectoire avec indépendance du temps. Ce type de synthèse est difficile à réaliser avec un catalogue informatisé contenant des trajectoires exactes (liste de points). Lorsqu'il n'y a pas de correspondance un-à-un entre les points qui composent la trajectoire désirée et celles du catalogue, le calcul de la similarité entre les trajectoires est très coûteux.

Dans le chap. 10, une méthode originale de constitution et d'utilisation d'une base de cas qualitatifs est présentée, comportant une procédure de *normalisation* des trajectoires qualitatives (cf. §10.3.2) et une indexation moyennant une *clé qualitative* (cf. §10.3.3). Les cas contiennent, pour chaque valeur distincte de clé qualitative, un certain nombre des trajectoires qualitatives normalisées, correspondant à différentes précisions d'approximation en arcs de cercle. Les cas contiennent également le vecteur décrivant les dimensions du mécanisme, ainsi qu'une désignation (symbolique) de la morphologie du mécanisme ayant généré la trajectoire.

L'avantage de cette clé d'indexation à valeurs discrètes est de fournir un critère rationnel de discrétisation des trajectoires, permettant un filtrage des cas à admettre dans la base. Le but est d'avoir un nombre limité de cas pour chaque valeur unique de la clé d'indexation, en constituant ainsi une *base de familles de cas*. Autrement dit, la normalisation et la clé qualitative permettent de mieux répartir les cas, à nombre total constant, sur l'ensemble des formes qualitatives de trajectoires. De plus, ainsi constituée, la base de cas est multi-résolution (fournit des solutions dans une gamme plus large de précisions) et permet au concepteur d'exprimer des requêtes approximatives.

La génération des cas est réalisée par le tirage aléatoire de jeux de dimensions, suivi de la détermination de la clé qualitative d'indexation pour la trajectoire générée, au préalable rendue qualitative et normalisée. La dernière étape est la vérification de l'existence ou non dans la base d'un cas avec une clé similaire. Compte tenu du caractère discret de la clé, cette vérification peut être implémentée d'une manière très efficace. Le tirage aléatoire de dimensions continue jusqu'à ce que l'apparition de nouvelles clés ait une fréquence inférieure à un certain seuil (critère d'arrêt paramétré).

Le domaine d'application de cette méthode est la synthèse des générateurs de formes de trajectoires ouvertes ou fermées, avec indépendance du temps. Des trajectoires générées par différentes morphologies peuvent être utilisées pour constituer la même base de cas (en stockant simplement une information symbolique identifiant la morphologie). La méthode permet, ainsi, la synthèse conjointe morphologique et dimensionnelle.

3.6.5 Raisonnement à base de cas par réseaux neuronaux

La deuxième méthode de synthèse basée sur les cas, développée dans cette thèse, utilise une approche différente, dont la spécificité est liée à la représentation des cas et au mode de constitution de la base de cas. La base de cas qualitatifs, précédemment présentée, permettait une sélection *explicite* du cas le plus proche, par l'utilisation d'un fichier de données classique. Il s'agit maintenant de réaliser une synthèse *implicite* à partir de cas, moyennant un *réseau neuronal artificiel*. (Une présentation des réseaux neuronaux est donnée dans l'annexe C.)

Les cas qualitatifs s'affranchissaient de la représentation des trajectoires par une liste des points – qui est une source non fonctionnelle de complexité – grâce à l'approximation en arcs de cercle. Ici, un résultat équivalent est obtenu par une analyse harmonique des trajectoires. Les trajectoires sont ainsi représentées par un certain nombre de coefficients de Fourier qui, après une normalisation, codifient toute une classe de trajectoires ayant la même forme.

Afin de réaliser, pour une morphologie choisie, la synthèse implicite par un réseau neuronal, les cas sont considérés comme un ensemble d'échantillons ou couples $(x, f(x))$ d'une fonction f dont nous ne disposons pas de l'expression explicite. Ici, x représente la codification des formes de

trajectoires et $f(x)$ représente les dimensions du mécanisme correspondant. Comme d'habitude, l'ensemble de cas (ou d'échantillons) est généré par des simulations cinématiques. Cet ensemble sert ensuite à l'apprentissage d'un réseau neuronal[4], réalisant ainsi l'interpolation des cas. Lors de la synthèse, le réseau permet de calculer très facilement les dimensions d'un mécanisme générant une trajectoire dont la forme est proche de la forme spécifiée par le concepteur.

Une différence par rapport aux autres méthodes basées sur les cas est le fait que, grâce à l'interpolation, cette solution approximative ne correspond plus seulement à des cas utilisés pour la constitution de la base de cas. C'est un avantage important, car, par ailleurs, il est de toute façon impossible de générer suffisamment de cas pour couvrir toutes les formes de trajectoires. Cette méthode concentre dans très peu de données (emmagasinées dans le réseau) virtuellement une infinité de solutions potentielles. En somme, l'utilisation combinée des coefficients de Fourier et du réseau neuronal permet de s'affranchir à la fois du caractère discret de la représentation des trajectoires par une liste de points et du caractère discret de la base de cas.

Cependant, ce problème d'approximation de fonction est très difficile à résoudre. En effet, dans certaines zones de l'espace de solutions (jeux de dimensions du mécanisme), cette fonction n'est pas correctement définie, car la même forme de trajectoire peut être générée par différents jeux de dimensions du même mécanisme. Les cas situés en proximité de ces zones risquent donc de ne pas être interpolés correctement. De plus, certaines formes de trajectoire ne peuvent pas être générées par un mécanisme d'une morphologie donnée et, inversement, certains jeux de dimensions ne permettent pas la génération d'une trajectoire fermée. A cela s'ajoute la non linéarité et la multi-dimensionalité[5].

Par conséquent, des précautions s'imposent lors de la génération des cas, préalable à l'apprentissage du réseau, afin de limiter autant que possible les zones de non-définition de la fonction. Il s'agit de la limitation des intervalles de valeurs des dimensions du mécanisme, à l'intérieur desquels sont générés les cas, et du choix d'une position fixe pour certaines liaisons. Malheureusement, le choix des intervalles de dimensions appropriés, s'il est relativement aisé pour les morphologies les plus simples, devient de plus en plus difficile pour les morphologies complexes.

La solution approximative fournie par le réseau neuronal peut être ensuite utilisée comme solution initiale pour une optimisation dimensionnelle classique (cf. Figure 3.3).

3.7 Synthétiser par optimisation

3.7.1 Optimisation dimensionnelle exacte (l'approche classique)

L'approche d'optimisation dimensionnelle la plus utilisée dans la littérature consiste en un bouclage sur des analyses cinématiques (dont une présentation conceptuelle a été donnée au §3.3). En effet, la fonction objectif – qui est une mesure de la distance entre les trajectoires générées et désirées – ne peut pas être exprimée explicitement en fonction des variables dimensionnelles. L'analyse cinématique permet de déterminer la trajectoire générée pour chaque valeur du vecteur de conception et donc, finalement, de calculer la valeur de la fonction objectif. La modification à apporter à chaque itération au vecteur de conception est déterminée par l'algorithme d'optimisation choisi, le plus souvent de type gradient[6]. Si la méthode de synthèse est dédiée à un cas particulier de mécanisme, les dérivées partielles peuvent parfois être déterminées analytiquement, mais dans le cas général leurs valeurs sont calculées par différences finies.

4. L'apprentissage est un algorithme d'optimisation qui modifie itérativement les paramètres du réseau afin de minimiser l'erreur d'approximation.

5. Pour le mécanisme *quatre barres*, nous utilisons 5 paramètres dimensionnels et 17 paramètres codifiant la forme des trajectoires (coefficients de Fourier normalisés).

6. Nous avons essayé également un algorithme d'optimisation globale (le *recuit simulé*), mais les résultats n'ont pas été encourageants (probablement à cause de la taille du vecteur de conception).

Une des difficultés de l'optimisation dimensionnelle est le traitement des cas où la simulation cinématique à partir des valeurs du vecteur de conception conduit à une trajectoire générée ouverte, à cause des singularités géométriques atteintes. Le problème se pose alors de savoir comment pénaliser intelligemment ces cas, en gardant la continuité de la fonction objectif. La solution originale adoptée consiste en une pénalisation de la fonction objectif par un terme dépendant de la différence de longueur entre la trajectoire désirée (fermée) et la trajectoire générée (ouverte). (Voir les détails au §8.2.1.)

La convergence des méthodes de type gradient dépend, entre autres, de la « qualité » de la solution initiale, utilisée comme point de départ. Les méthodes de *continuation*, présentées dans la littérature (voir §2.4.3.8), tendent à réduire cet inconvénient. Nous proposons une méthode originale d'accélération de la convergence utilisant le *recalage géométrique*. Cette méthode exploite une propriété remarquable des problèmes d'optimisation dimensionnelle des générateurs de trajectoire : la trajectoire générée est géométriquement solidaire du mécanisme, autrement dit toute transformation affine appliquée à la géométrie du mécanisme se répercute de la même manière sur la trajectoire. Après un certain nombre d'itérations de l'optimisation, le vecteur de conception correspond à une trajectoire située à une certaine distance de la trajectoire désirée. La transformation géométrique affine de la trajectoire générée qui minimise la distance entre les deux trajectoires, appliquée au mécanisme (donc au vecteur de conception), permettra de faire un « saut » qui rapproche la solution courante de la solution optimale. Cette idée – très simple mais assez efficace – se décline en différentes variantes, selon le mode de calcul de la transformation affine optimale. A son tour, celui-ci dépend du mode de définition des trajectoires. Ici, pour des trajectoires exactes, le calcul de cette transformation est réalisé par l'écriture d'un système linéaire dont les variables sont les paramètres optimaux de transformation. (Pour les détails, voir §8.2.5.)

Pour les générateurs de trajectoire avec indépendance du temps, une bonne solution initiale pour l'optimisation peut être obtenue avec les méthodes basées sur les cas (voir §3.6).

L'optimisation dimensionnelle exacte sera traitée en détail en §8.2.

3.7.2 Optimisation dimensionnelle basée sur la cinématique qualitative

Toutes les approches classiques d'optimisation dimensionnelle – ainsi que les méthodes d'analyse cinématique qu'elles font intervenir – utilisent la description des trajectoires générées par une suite de points. En effet, l'analyse cinématique détermine, pour chaque position du ou des moteurs, la position du point effecteur. La trajectoire sera donc composée de la liste de ces points, dont le nombre dépendra de la discrétisation plus ou moins fine de la position du moteur.

Une approche de cinématique radicalement différente, utilisée dans cette thèse pour l'optimisation dimensionnelle (cf. §8.3), est la *cinématique qualitative* développée par Yannou (*Yannou et Vasiliu* [YV95] [YVM95]). (Pour les détails sur cette approche de cinématique, voir chap. 5). A partir de considérations sur les centres instantanés de rotation et sur la courbure, la trajectoire générée peut être déterminée sous une forme qualitative, composée par une liste d'arcs de cercle et de segments de droite. La *simulation qualitative* consiste en une propagation des morceaux de trajectoires à travers le mécanisme, à partir du ou des moteurs et jusqu'au point effecteur (cf. Figure 3.3). Cette méthode n'est cependant pas une méthode d'analyse cinématique proprement-dite, puisqu'elle suppose la connaissance de la trajectoire désirée pour effectuer la simulation. Cette méthode est donc spécifique à une optimisation dimensionnelle. Deux cas sont possibles :

- soit le concepteur désire imposer une temporalité bien définie du mouvement du point effecteur en rapport avec celui du moteur, auquel cas cette correspondance est spécifiée par le concepteur ;

- soit aucune condition de temporalité n'est imposée, auquel cas cette correspondance sera déterminée par un jeu de variables supplémentaires, se rajoutant au vecteur de conception utilisé pour l'optimisation.

Bien que l'efficacité des algorithmes d'optimisation soit relativement indépendante du type de simulation cinématique utilisé, l'optimisation dimensionnelle basée sur une cinématique qualitative est à peu près deux fois plus rapide qu'une optimisation basée sur une simulation classique, en contrepartie de la diminution de la précision numérique.

D'une manière générale, les méthodes de synthèse dimensionnelle par l'optimisation permettent de résoudre seulement une partie des problèmes de génération de trajectoire. Le succès de l'utilisation de ces méthodes est néanmoins limité[7] et il reste limité malgré l'amélioration de la vitesse obtenue par l'optimisation basée sur la simulation qualitative. Il s'agit de limites en termes de durée, de convergence, de dépendance par rapport à la solution initiale, mais également des limites quant au type de cahier des charges. Ainsi, la difficulté de l'optimisation augmente linéairement avec le nombre de points de passage sur la trajectoire désirée. Cette augmentation est encore plus rapide par rapport à la taille du vecteur de conception. De plus il est très difficile de résoudre la génération de trajectoire avec indépendance du temps (qui implique l'introduction de nombreuses variables de conception supplémentaires). Toutes ces difficultés trouvent partiellement une solution complémentaire dans les méthodes de synthèse basées sur les cas, précédemment présentées.

3.8 Conclusion

La conception des générateurs de trajectoire reste – malgré des longues décennies de recherches – un domaine ouvert, où l'enjeu intellectuel et scientifique se combine avec l'enjeu industriel. En effet, pour certains types de problèmes de synthèse morphologique et dimensionnelle les solutions existantes ne sont pas satisfaisantes (ou n'existent pas du tout). Par conséquent, la plupart des logiciels spécialisés se limitent à des fonctionnalités d'analyse (cinématique, en dynamique inverse et dynamique) et sont souvent lourds à manipuler, surtout pour des cycles essai-erreur.

Il serait illusoire d'espérer trouver une méthode de synthèse universelle, applicable et efficace dans tous les cas particuliers. Dans cette thèse, nous nous sommes proposés plutôt d'exploiter le concept de plateforme de préconception (ou de « boîte à outils »), dans laquelle le concepteur peut puiser dans un ensemble de méthodes. Sur le plan pratique, le but a été de réaliser, dans le logiciel REALISME, un système de conception intégrant plusieurs modèles et méthodes, afin de résoudre mieux des objectifs de conception spécifiques. Avec le souci, en plus, de garder une ouverture maximale du logiciel, lui permettant d'évoluer et de s'enrichir facilement. Néanmoins, nous sommes conscients que les modèles et les méthodes employés dans cette thèse ne sont pas une réponse définitive et qu'ils restent perfectibles.

7. Ceci explique, d'ailleurs, pourquoi les logiciels courants intègrent rarement l'optimisation dimensionnelle parmi leurs fonctionnalités.

Deuxième partie

Modèles de cinématique pour la préconception

Chapitre 4

Modèle systémique des mécanismes plans : les groupes multipolaires

4.1 L'approche systémique

Dans l'état de l'art (§2.1.2), l'approche modulaire d'étude des mécanismes plans a été brièvement présentée. Si ses origines remontent aux premières décennies du siècle, elle a continué jusqu'à présent à connaître différentes applications, variantes ou extensions, démontrant ainsi son intérêt. Dans le présent chapitre est présentée l'extension *systémique multipolaire* de l'approche modulaire. L'apport de l'approche systémique à l'élaboration de méthodes de préconception utilisées dans REALISME est évoqué également au §3.3 et §3.4, ainsi que dans différents chapitres ultérieurs (chap. 6, 8 et 10).

4.1.1 Sous-systèmes élémentaires : les groupes multipolaires

4.1.1.1 Description des groupes

Pour aboutir à une modélisation systémique, il faut d'abord identifier les frontières du système et des sous-systèmes. Ensuite, il faut étudier les entrées et les sorties circulant à travers les frontières. A partir de la notion de *groupe d'Assur*, étendue par *Verho* [Ver73] aux groupes comportant des liaisons motrices[1], l'approche systémique (*Pelecudi et Simionescu* [PDS82] [PS82] [PS86a] [PS86b]) définit des sous-systèmes élémentaires, appelés *groupes multipolaires*. Les groupes sont des parties de mécanisme (ou modules) dont « l'assemblage » permet de constituer un mécanisme complet, un peu comme dans un jeu de légos. Chaque groupe reçoit des « informations » sous forme de mouvement et d'effort. Ces informations, après transformation par des fonctions de transfert, sont transmises à d'autres groupes.

Des simulations de plusieurs types sont possibles à partir de la décomposition en groupes. Ainsi, l'analyse cinématique est basée sur la considération du mécanisme comme une structure composée de lignes de transmission du mouvement à partir des moteurs, à travers les différents organes et jusqu'à l'organe effecteur. L'analyse en dynamique inverse étudie la transmission des efforts sur les mêmes chemins en sens inverse. Le but de l'approche multipolaire est l'identification de cette structure de circulation des mouvements et des efforts, qui permettra l'emploi de méthodes

1. Voir la définition à la page 195.

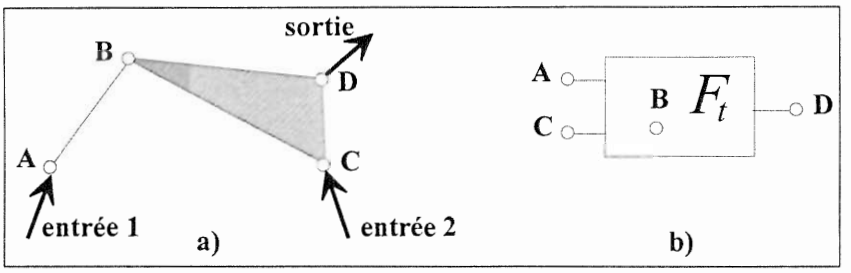


FIG. 4.1 – *Exemple de groupe multipolaire (la* diade*) : schéma cinématique (a) et groupe multipolaire correspondant, représenté par son symbole systémique (b).*

adaptées et efficaces de simulation cinématique et en dynamique inverse, mais également de faciliter la démarche de conception.

Dans un certain sens, les groupes multipolaires sont des *briques fonctionnelles* (cf. la discussion au §3.5, p. 44), reliant un sous-ensemble d'équations à un sous-ensemble de composants mécaniques. Cependant, surtout pour les générateurs de trajectoire, la fonction globale – qui intéresse le concepteur – ne peut pas être déterminée uniquement par la connaissance des *briques*, sans faire intervenir le dimensionnement complet du mécanisme. En effet, les fonctions de transfert attachées aux groupes (sous-ensembles d'équations) ne sont pas du même ordre que la fonction globale (trajectoire). De ce fait, l'utilisation de méthodes de synthèse par composition de *briques fonctionnelles* n'est pas possible. C'est donc par d'autres biais que l'approche multipolaire est utile pour la préconception.

Pour une description plus détaillée des groupes multipolaires, il faut distinguer :

1. les pôles externes d'entrée,
2. les pôles internes,
3. les fonctions de transfert,
4. les pôles externes de sortie.

Les pôles externes d'entrée et de sortie sont des liaisons potentielles[2], permettant la connexion avec les autres groupes. Les pôles internes sont des liaisons effectives entre les éléments d'un même groupe. Un symbole graphique est associé à chaque type de groupe multipolaire (voir un exemple dans la Figure 4.1), sous la forme d'un rectangle avec des « pattes » ou des pôles d'entrée (à gauche), internes (à l'intérieur) et de sortie (à droite)[3]. Les pôles sont soit actifs soit passifs, selon que la liaison cinématique qu'ils représentent est une liaison motrice ou non.

Les pôles externes de sortie permettent de récupérer le mouvement fourni par un groupe situé en amont pour le transmettre à un autre situé en aval. Quant aux fonctions de transfert, représentant la transformation opérée par le groupe entre les entrées et les sorties, elles dépendent de la constitution interne (morphologie, paramètres géométriques invariables) et instantanée du groupe (paramètres cinématiques variables des pôles d'entrée).

4.1.1.2 Classification

Les groupes multipolaires peuvent être classifiés selon le nombre de pôles d'entrée et de pôles internes, et selon leur type (actif ou passif). Puisque le nombre de pôles de sortie dépend seulement

2. Une liaison potentielle est une liaison fictive, représentée sur un seul élément, qui devient une liaison réelle seulement lors de la connexion avec un autre groupe, *par adjonction* d'un deuxième élément.

3. Il faut néanmoins noter que, pour l'analyse en dynamique inverse, le rôle des pôles externes est inversé : les pôles de sortie sont des pôles d'entrée pour les efforts et réciproquement.

n	2	4	6	...
j	3	6	9	...

TAB. 4.1 – *Combinaisons possibles du nombre de solides et de liaisons dans les groupes multipolaires passifs.*

des besoins de connexion des groupes, ce nombre n'est pas une propriété intrinsèque du groupe et ne définit pas son type.

Quelles sont les propriétés structurales qui permettent d'identifier et de classifier les groupes multipolaires? Le partitionnement des mécanismes en groupes doit être réalisé de telle manière que, pour chaque groupe, la connaissance des paramètres cinématiques (position ou orientation, vitesse, accélération) des pôles d'entrée soit suffisante pour la détermination des paramètres des tous les autres éléments du groupe, en particulier des pôles de sortie. Il s'agit donc d'identifier les parties minimales d'un mécanisme respectant ce critère.

Une distinction est faite entre les groupes à mobilité relative $m > 0$, contenant des liaisons motrices, appelés *groupes actifs*, et les groupes à mobilité relative $m = 0$, dépourvus de liaisons motrices et appelés *groupes passifs*. Les *groupes actifs* correspondent aux chaînes cinématiques minimales constituant une source de mouvement[4], et les *groupes passifs* aux chaînes minimales permettant la transmission du mouvement.

Définition 1 *Le groupe multipolaire actif est défini comme une chaîne cinématique de mobilité $m > 0$, obtenue à partir d'une chaîne cinématique fermée, de mobilité m, par enlèvement d'un de ses éléments, et ne pouvant être décomposé en groupes plus simples.*

Définition 2 *Le groupe passif est une chaîne cinématique obtenue à partir d'une chaîne fermée à mobilité $m = 0$ par élimination d'un élément; la mobilité relative reste nulle, mais la mobilité absolue devient $2j$, où j est le nombre de liaisons éliminées*[5]*. Comme pour les groupes actifs, les groupes passifs ne sont pas décomposables en groupes plus simples.*

4.1.1.3 Caractérisation cinématique (groupes passifs)

Pour les groupes passifs, la mobilité relative peut être exprimée par l'équation de Grübler:

$$m = 3n - 2j = 0 \tag{4.1}$$

où n est le nombre d'éléments (hors bâti) et j le nombre de liaisons élémentaires (pivots ou glissières). Puisque m doit être nul, on obtient la condition:

$$j = \frac{3}{2}n \tag{4.2}$$

Puisque $j \in \mathbf{N}$, la condition 4.2 est vérifiée seulement par certaines combinaisons de valeurs de n et j (voir Table 4.1).

Une infinité de groupes peuvent ainsi être énumérés. Toutefois, en pratique, les groupes les plus simples (peu nombreux) sont de loin les plus utilisés. Les groupes les plus fréquemment utilisés, avec leur symbole systémique et leur nom conventionnel, sont représentés dans la Figure 4.2:

- le *zéropôle* (groupe passif comportant uniquement des pôles de sortie; il correspond au bâti),

4. Dans l'approche modulaire utilisée par *Hansen* [Han92], seuls deux types de groupes actifs, appelés *modules d'expansion*, sont identifiés: la manivelle et le moteur de translation. Cependant, il existe des groupes actifs plus complexes (voir par exemple *Pelecudi et Simionescu* [PS82]).

5. On rappelle que, dans cette thèse, seuls les mécanismes composés de liaisons élémentaires (pivots et glissières) sont considérés.

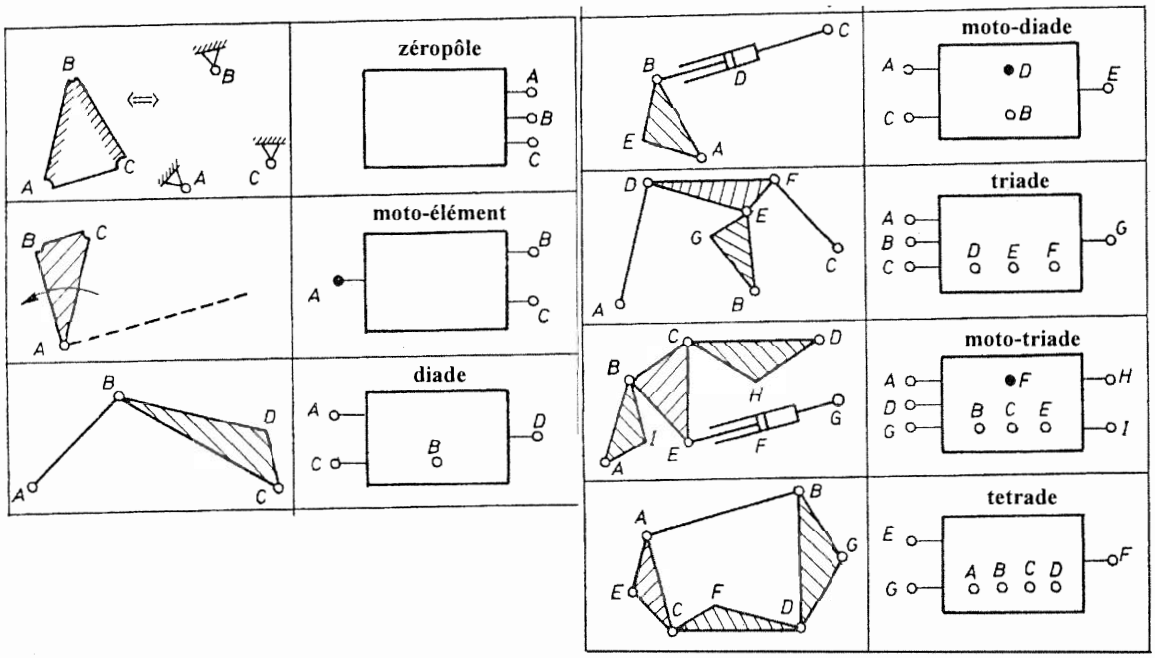


FIG. 4.2 – *Différents types de groupes multipolaires et leurs symboles systémiques. Dans le symbole des groupes actifs, les liaisons motrices sont représentées par un cercle plein. (Cf. [PS86a].)*

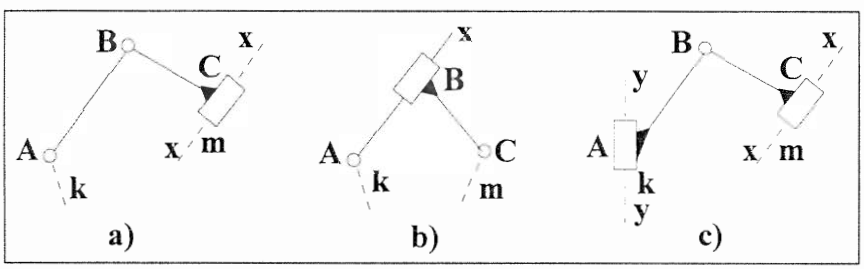


FIG. 4.3 – *Trois types de groupes à deux pôles d'entrée (diade) selon les types des liaisons. a) : diade RRT (rotation-rotation-translation) ; b) diade RTR ; c) diade TRT. (Cf.* Artobolevski *[Art75b].)*

- la *diade*, la *triade* et la *tétrade* (groupes passifs),
- le *moto-élément*, la *moto-diade* et la *moto-tétrade* (groupes actifs).

Un logiciel utilisant seulement ces groupes, autrement dit une bibliothèque très restreinte de modules de base, peut couvrir une grande diversité de mécanismes.

Un dernier critère d'énumération est le type des liaisons représentées par les pôles du groupe. Pour les diades, par exemple, on distingue les diades RRR, RRT, RTR, TRT et RTT (R = rotation, T = translation). (Voir quelques exemples dans la Figure 4.3.)

4.1.2 La construction modulaire des mécanismes plans

Le principe qui est à la base de la construction modulaire des mécanismes plans est le suivant : l'assemblage d'un nombre quelconque de groupes peut être réalisé sans perdre le contrôle du degré

de liberté. Plus exactement, le nombre de degrés de liberté du mécanisme est égal à la somme des mobilités relatives des groupes composant le mécanisme pour une configuration quelconque[6].

En définissant les groupes comme des chaînes cinématiques fermées (voir aussi l'état de l'art, p. 12), *Galletti* [Gal86] propose un processus de construction des mécanismes consistant en une expansion de la structure initiale par remplacement récursif d'élément(s) avec d'autres *chaînes cinématiques d'Assur* (CCA). L'avantage de cette méthode est de permettre la détermination directe, à partir du schéma d'expansion, du type de mobilité (totale, partielle ou fractionnée). Afin d'obtenir des mécanismes, possédant donc au moins un degré de liberté, un ou plusieurs éléments du CCA sont remplacés par une combinaison d'éléments, interconnectés par des liaisons motrices.

Dans l'approche de *Pelecudi et Simionescu* [PS82, p. 17-19], utilisée dans REALISME, un mécanisme est obtenu par la connexion, de différentes manières (en série, en parallèle...), des groupes multipolaires à travers leurs pôles externes. Certaines règles simples sont à observer (cf. [PS82, p. 17–19]) :

- la présence d'un zéropôle est obligatoire,
- au moins un groupe actif doit avoir tous ses pôles d'entrée liés au zéropôle,
- les groupes passifs ne peuvent pas avoir tous les pôles d'entrée connectés au même élément cinématique,
- tous les pôles d'entrée doivent être connectés aux pôles de sortie d'autres groupes,
- le degré de liberté total est égal à la somme des mobilités de chaque groupe.

Ces règles permettent de détecter des morphologies invalides, mais peuvent également aider le concepteur à concevoir une morphologie de façon incrémentale, par rajouts de groupes (cf. §6.2.1).

4.1.2.1 Schéma multipolaire

Les symboles graphiques systémiques des groupes multipolaires (voir la Figure 4.3) servent à la visualisation du *schéma systémique multipolaire*.

Définition 3 *Le schéma multipolaire est un graphe (connexe, non orienté) dont les nœuds sont les groupes multipolaires et les arêtes sont les pôles communs à deux groupes.*

Le schéma multipolaire est une représentation de la structure du mécanisme dans le sens de la théorie des systèmes, modélisant l'enchaînement et le mode de connexion des groupes multipolaires. Ce schéma met en évidence le flux de transmission du mouvement entre les différents groupes du mécanisme.

Pour donner un exemple, la Figure 4.4 présente le schéma cinématique d'un mécanisme complexe, le graphe associé et le schéma multipolaire correspondant. Le schéma multipolaire rend une certaine logique et intelligibilité à la morphologie du mécanisme, pouvant constituer un support durant la préconception (voir aussi le chap. 6).

Le logiciel REALISME comprend une procédure de décomposition automatique des mécanismes en groupes multipolaires[7]. La succession d'étapes de cette procédure (voir Algorithme 4.1) s'inspire

6. Cependant, il peut exister des configurations singulières, pour lesquelles le nombre de degrés de liberté augmente en contrepartie d'un hyperstatisme. Il existe également des positions de blocage lorsqu'on tente d'imposer un mouvement continu au niveau d'une liaison élémentaire. Si ces blocages ne changent pas le nombre de degrés de liberté, il est néanmoins impossible de continuer à imposer le mouvement sur la même liaison.

7. D'autres algorithmes ont été proposés dans la littérature. Celui de Lauw [LCK87] est jugé très lent par Bawab [BK95]. Un autre algorithme est celui de Romdhane [Rom95]. (Voir aussi l'état de l'art, §2.1.2).

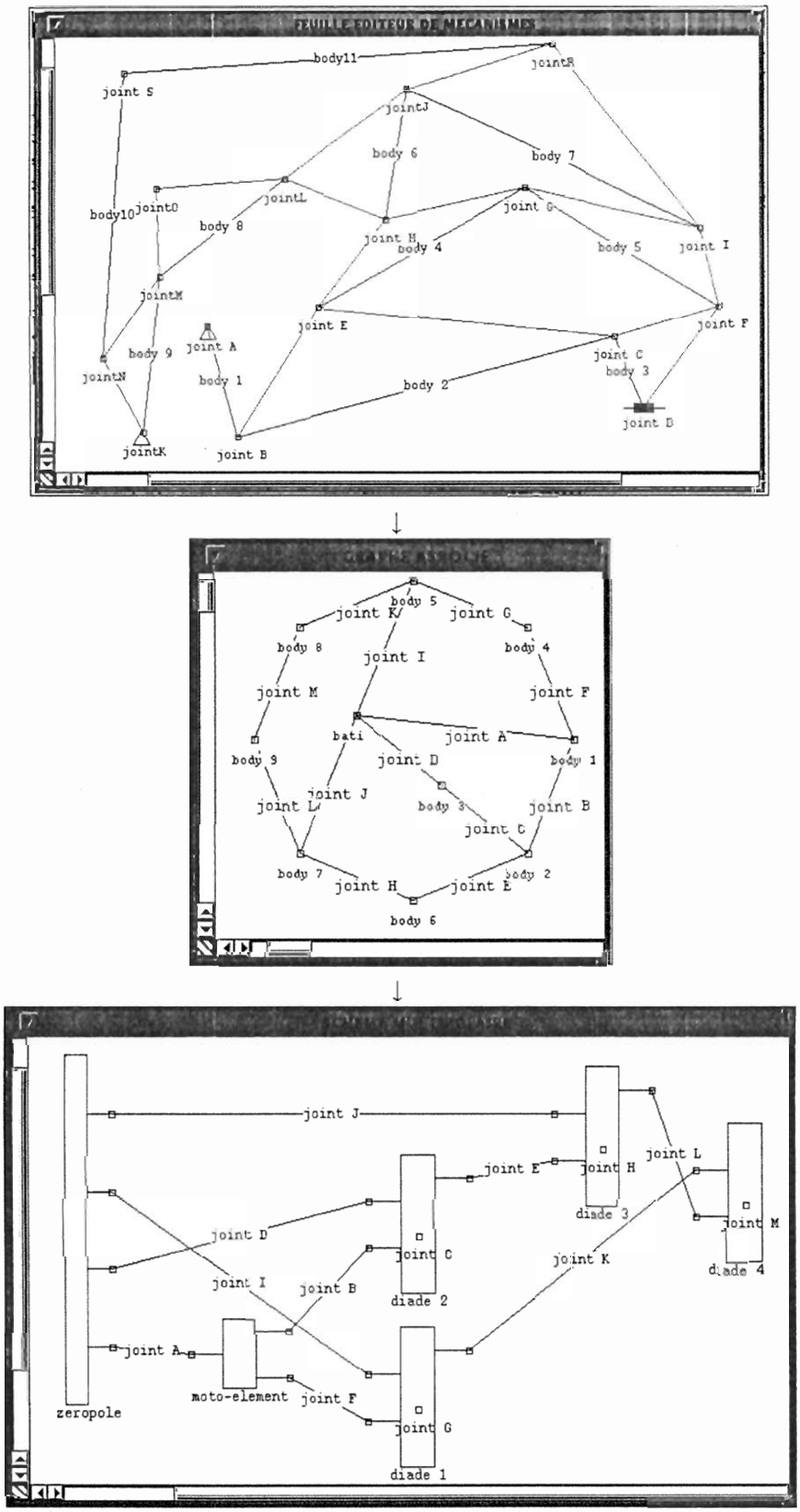


FIG. 4.4 – *Un mécanisme complexe : schéma cinématique, graphe associé et schéma multipolaire.*

d'une méthode de décomposition « manuelle » esquissée par I. Artobolevski[8] et utilise le graphe associé[9] du mécanisme.

Etape 1 Détermination du graphe associé du mécanisme (dans lequel les nœuds correspondent aux solides et les arêtes aux liaisons) ;

Etape 2 Application de la procédure *Recherche-diades* pour l'identification des diades ;

Etape 3 Application de la procédure *Recherche-triades* pour l'identification des triades ;

Etape 4 Application de la procédure *Recherche-tétrades* et ainsi de suite[a].

[a] Nous rappellons que la fréquence d'utilisation des différents types de groupes multipolaires décroît avec leur complexité (les diades sont de loin les plus utilisées). Dans l'énorme majorité des cas, la recherche des groupes peut donc s'arrêter aux groupes les plus simples. C'est cette constatation qui a été utilisée pour l'algorithme de décomposition.

Algorithme 4.1: *Algorithme de détermination du schéma multipolaire.*

La procédure *Recherche-diades* est basée sur une exploration *en profondeur d'abord* dans le graphe associé du mécanisme (voir Algorithme 4.2). A chaque pas de cette exploration, les nœuds courant et précédent du graphe sont examinés. S'ils remplissent un certain nombre de conditions, une diade constituée des deux nœuds est ajoutée à la liste des diades. Les procédures d'identification des autres groupes sont analogues.

Supposons, par exemple, que l'exploration du graphe associé d'un mécanisme *quatre barres* (voir Figure 4.5) démarre avec les nœuds *body 1* et *body 2* (autrement dit, on vérifie si le groupe formé des barres *1* et *2* est une diade). Le test décrit ci-dessus échoue, puisque le nœud *body 1* est adjacent de la liaison motrice *joint A*. Ensuite, le couple de nœuds (*body 2, body 3*) remplit les conditions et est identifié à une diade. Finalement, les couples (*body 3, body 4*) et (*body 4, body 1*) ne sont pas identifiés à des diades car le nœud *body 4* est le bâti.

Au fur et à mesure que les différents groupes sont identifiés, l'information sur leur connectivité, déjà disponible, est utilisée pour organiser les groupes dans une structure de graphe. Une fois que tous les groupes ont été identifiés, il reste à obtenir la représentation graphique de ce graphe (comme dans la Figure 4.4). Du point de vue informatique, la représentation graphique de chaque groupe est une méthode de la classe correspondant au groupe. (Voir les détails d'implémentation au §4.6.) Quant à la position de chaque nœud, c'est-à-dire l'emplacement de chaque groupe, elle sera calculée à l'intérieur de la procédure de détermination de la relation structurale (voir ci-après), de façon à permettre une visualisation lisible du schéma multipolaire.

4.1.2.2 Relation structurale

Le résultat de la détermination du schéma multipolaire est un graphe non orienté dont les nœuds sont les groupes et les arêtes sont les liaisons (pôles) reliant les groupes entre eux. Maintenant,

8. Voici le passage concerné (dans [Art75b, p. 56]) : « *On commence par essayer de séparer du mécanisme les groupes de classe II [diades]. Ce faisant, chaque fois qu'un groupe est séparé, on doit s'assurer que la chaîne cinématique restée possède le même degré de mobilité que le mécanisme primitif et qu'il ne reste aucune partie d'un élément qui ne participe à un couple cinématique. Si la séparation des groupes de classe II s'avère impossible, on passe aux groupes de classe III et l'on essaie de les séparer. Après les groupes de classe III, on entreprend les mêmes tentatives avec les groupes de classe IV, et ainsi de suite. Apres la séparation de tous les groupes il doit rester le bâti et le ou les éléments menants.* »

9. Pour la modélisation des mécanismes par des graphes linéaires, voir §2.1.1.

Procédure *Recherche-diades* (**entrée** : GA ; **sortie** : LD)
% GA = graphe associé du mécanisme
% LD = liste des diades
$LD \leftarrow \emptyset$
Début Parcours en profondeur d'abord des nœuds du GA
$N1 \leftarrow$ nœud courant
$N2 \leftarrow$ nœud précédent
Si (
ni $N1$ ni $N2$ ne sont adjacents à un moteur $\wedge$
ni $N1$ ni $N2$ ne sont le bâti $\wedge$
$N1$ et $N2$ ont chacun deux voisins dans GA [a]
)
Alors
$LD \leftarrow LD + diade(N1, N2)$
Marquer dans GA les nœuds $N1$ et $N2$ et l'arc les reliant
Fin Si
Passer au nœud suivant
Fin Parcours en profondeur d'abord
Fin Procédure

[a] Afin que leur élimination ne modifie pas la mobilité du mécanisme restant.

Algorithme 4.2: *Procédure d'identification des diades.*

il faut déterminer un ordre de parcours de ce graphe, correspondant au flux de transmission du mouvement.

Définition 4 *Par* relation structurale *([PS82]) on entend l'ordre de succession des groupes multipolaires, en partant du bâti vers le dernier groupe, avec la condition que tout groupe doit avoir ses pôles d'entrée connectés à des groupes situés avant lui et tous ses pôles de sortie connectés à des groupes situés après lui.*

Après la détermination de la relation structurale, le schéma multipolaire (par exemple celui de la Figure 4.4) devient un graphe orienté doué d'un ordre de parcours. La relation structurale est utilisée, en sens direct, pour la simulation cinématique (voir chap. 4.4), et en sens inverse, pour l'analyse dynamique inverse (voir §4.5).

La procédure principale de détermination de la relation structurale à partir du schéma multipolaire est présentée dans l'Algorithme 4.3. En même temps, le placement (le choix de la position) des groupes est réalisé. Le *zéropôle*, correspondant au bâti, occupe évidemment la première place dans la relation structurale ; en même temps, une position (coordonnées $x - y$) lui est affectée.

Ensuite, la procédure récursive *Place-groupe* (Algorithme 4.4), appliquée premièrement au zéropôle, examine successivement les groupes connectés en sortie de celui-ci. Parmi ces groupes, ceux dont tous les groupes connectés en entrée ont déjà été placés seront à leur tour placés (respectant ainsi la condition énoncée dans la définition de la relation structurale). Par contre, ceux dont au moins un des groupes connectés en entrée n'a pas encore été placé doivent attendre. C'est seulement lors d'un deuxième examen des mêmes voisins connectés en sortie que la procédure récursive sera relancée pour chaque groupe qui a déjà été placé. Cette démarche en deux passages est nécessaire pour s'assurer que l'appel récursif n'est lancé qu'après avoir placé tous les voisins immédiats

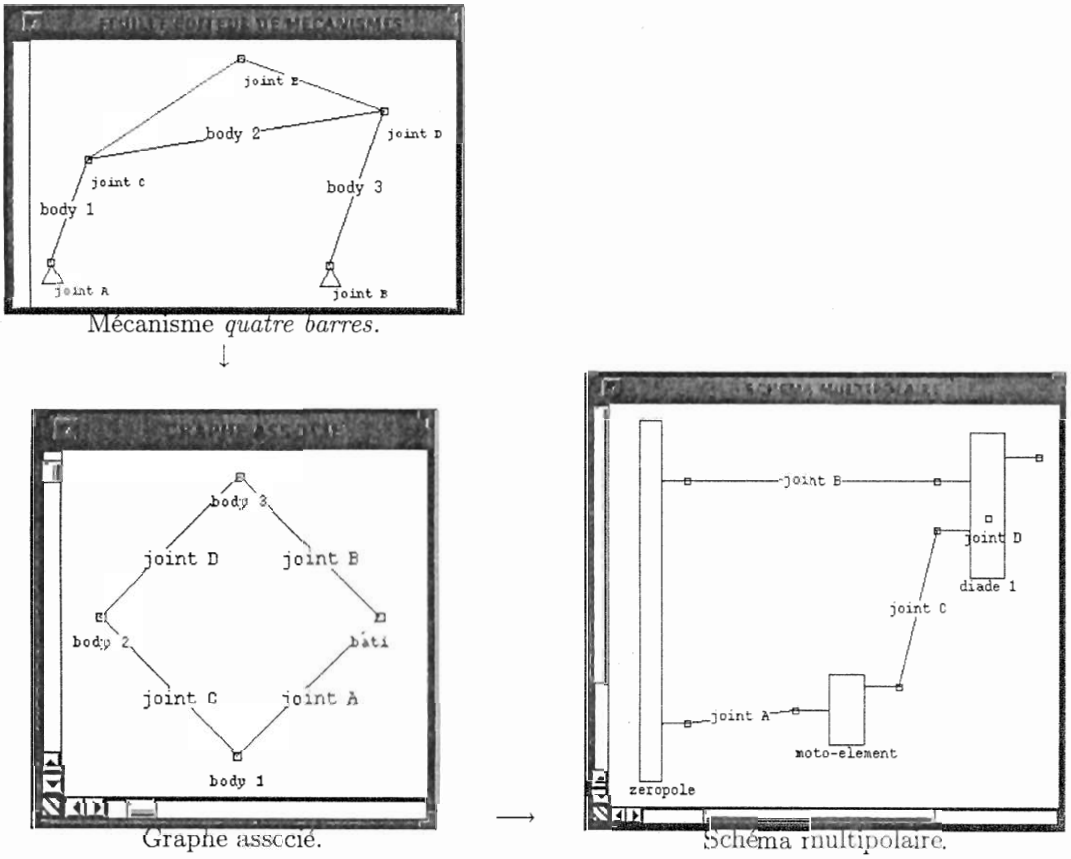


FIG. 4.5 – *Le mécanisme quatre barres : schéma cinématique, graphe associé et schéma multipolaire.*

Procédure *Détermine-rel-struct* (**entrée** : LG ; **sortie** : LOG)
  % LG = liste des groupes (déterminée par l'algorithme présenté au §4.1.2.1)
  % LOG = liste ordonnée des groupes
  $LOG \leftarrow \emptyset$
  $B \leftarrow$ le zéropôle % (groupe correspondant au bâti)
  Place-groupe(B, x_0, y_0) % (x_0, y_0) = position prédéfinie pour B
  Place-voisins(B, LOG)
Fin Procédure

Algorithme 4.3: *Détermination de la relation structurale (procédure Détermine-rel-struct).*

d'un groupe qui peuvent être placés. C'est une sorte de parcours *en largeur d'abord* du schéma multipolaire.

```
Procédure Place-voisins(entrée: GR; entrée/sortie: LOG)
    LVS ← Tous-les-voisins-en-sortie(GR) % liste des groupes en sortie du groupe GR
    Flag ← 1 % pour savoir si au moins un voisin a été placé à l'itération précédente
    % tant qu'il reste au moins un voisin non placé et
    % qu'à l'itération précédente on a placé au moins un...
    Tant que (LVS ≠ ∅ ∧ Flag = 1)
        Flag ← 0
        Pour chaque V ∈ LVS
            LVE ← liste des groupes connectés aux entrées de V
            Si (∀ g ∈ LVE, g ∈ LOG) % si toutes les entrées de V ont été placées
                LVS ← LVS − {V}
                Si (V ∉ LOG) % si V n'est pas encore placé
                    Flag ← 1
                    Place-groupe(V, Position-x(GR) + dx, Position-y(GR))
                                                    % dx = incrément de position
                    LOG ← LOG + V % Rajoute le voisin courant à la fin de LOG
                Fin Si
            Fin Si
        Fin Pour chaque

        Pour chaque V ∈ LVS
            LVE ← liste des groupes connectés aux entrées de V
            Si (∀ g ∈ LVE, g ∈ LOG) % si toutes les entrées de V ont été placées
                LVS = LVS − {V}
                Si (V déjà placé)
                    Place-voisins(V, LOG)
                Fin Si
            Fin Si
        Fin Pour chaque
    Fin Tant que
Fin Procédure
```

Algorithme 4.4: *Détermination du schéma multipolaire (procédure Place-voisins).*

Par exemple, pour le schéma multipolaire de la Figure 4.4, les voisins immédiats du zéropôle sont le groupe actif correspondant à la manivelle (*moto-élément*) et les diades 1, 2 et 3. Seul le groupe actif sera placé immédiatement après le zéropôle, puisque à cet instant ces trois diades n'ont pas tous les groupes en entrée déjà placés. Ensuite, la procédure est relancée à partir du groupe actif. Maintenant, les diades 1 et 2, qui sont ses voisins en sortie, peuvent être placées. Le troisième appel à la procédure permettra le placement de la diade 3 et le quatrième celui de la diade 4. L'ordre dans lequel les groupes sont placés détermine l'ordre d'apparition des groupes dans la relation structurale.

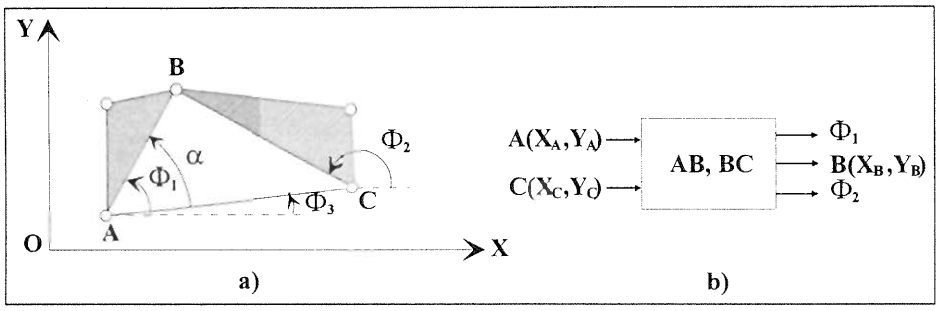


FIG. 4.6 – *a) Notations pour la diade RRT. b) Organisation des données.*

4.2 Fonctions de transfert en cinématique

En tant que sous-systèmes élémentaires du mécanisme, les groupes multipolaires reçoivent, par leurs pôles d'entrée, du mouvement sous forme de positions, de vitesses et d'accélérations. A travers les pôles de sortie, le mouvement est transmis au(x) groupe(s) connecté(s) en sortie, après une transformation selon une fonction de transfert dépendant du type de groupe. Deux exemples de fonction de transfert seront présentés ci-après : la diade RRR (groupe à deux éléments et trois liaisons de rotation) et la manivelle motrice.

Grâce à la connaissance a priori de la morphologie de chaque groupe, les fonctions de transfert correspondantes peuvent être déterminées facilement. De plus, pour la plupart des groupes simples (manivelle motrice, diades), elles se traduisent par des équations explicites donnant les caractéristiques cinématiques des pôles de sortie en fonction de celles des pôles d'entrée. Une décomposition modulaire a ainsi pour effet de réduire au maximum les calculs numériques itératifs durant la simulation cinématique (voir §4.4.2).

4.2.1 Exemple : la diade RRR

4.2.1.1 Positions

Le calcul de la position des éléments du groupe prend en considération la position de ses entrées A et C, supposée connue au moment du calcul (les notations correspondent à la Figure 4.6).

Les données, correspondant à un paramétrage cartésien absolu, peuvent être organisées de la manière suivante :

Données internes : AB, BC – longueurs des deux barres.

Données d'entrées : X_A, Y_A, X_C, Y_C – coordonnées cartésiennes absolues des pôles d'entrée A et C.

Position initiale de la diade : X_B, Y_B – coordonnées approximatives du pôle interne B.

Données de sortie : ϕ_1, ϕ_2 – angles des barres AB et CB, mesurés à partir de l'axe OX ; X_B, Y_B – coordonnées exactes du pôle interne B.

A partir des coordonnées connues des pôles A et C, la distance AC est calculée :

$$AC = \sqrt{(X_C - X_A)^2 + (Y_C - Y_A)^2} \tag{4.3}$$

Le calcul des sorties est possible seulement si les conditions 4.4 et 4.5 sont respectées. Ces deux conditions correspondent aux deux inégalités triangulaires exprimant que la barre AC ne peut pas être plus longue que la somme des deux autres et qu'elle ne peut pas être plus courte que la différence des deux autres.

$$AC < k \cdot (AB + BC) \tag{4.4}$$

$$k \cdot AC > |AB - BC| \tag{4.5}$$

Le coefficient k,avec $k \simeq 0,96$ (valeur empiriquement établie), permet d'éviter de s'approcher trop des positions singulières. Si la condition 4.4 est violée, on a un blocage. Si la condition 4.5 est violée, on a une indétermination de la position suivante (bifurcation) (cf. les Figures 6.7 et 6.8, p. 95). Après le calcul des coordonnées du pôle interne B et des angles des deux barres, les positions de toutes les liaisons attachées (en sortie) aux éléments du groupe sont facilement calculables[10].

Calcul de l'angle ϕ_3 de la barre AC (mesuré à partir de l'axe OX) :

$$\phi_3 = \tan^{-1} \frac{Y_C - Y_A}{X_C - X_A} \tag{4.6}$$

Calcul de l'angle α (entre AC et AB, mesuré à partir de AC) :

$$\alpha = \pm \cos^{-1} \frac{AB^2 + AC^2 - BC^2}{2 \cdot AB \cdot AC} \tag{4.7}$$

Le signe de α dépend du mode d'assemblage dans la position initiale, qui peut être déterminé à l'aide des coordonnées approximatives du pôle B. Comme la simulation est réalisée pour des positions successives de la manivelle (du moteur), les coordonnées exactes du pôle B à la position précédente peuvent être utilisées comme coordonnées approximatives. En effet, au cours d'une simulation, le mode d'assemblage ne change pas (mobilité continue).

Avec la valeur de l'angle α, l'angle ϕ_1 peut être calculé :

$$\phi_1 = \phi_3 + \alpha \tag{4.8}$$

Finalement, les coordonnées exactes du pôle B résultent :

$$\begin{aligned} X_B &= X_A + AB \cdot \cos\phi_1 \\ Y_B &= Y_A + AB \cdot \sin\phi_1 \end{aligned} \tag{4.9}$$

L'angle de la barre BC peut être également calculé :

$$\phi_2 = \tan^{-1} \frac{Y_B - Y_C}{X_B - X_C} \tag{4.10}$$

4.2.1.2 Vitesses

Le calcul des composantes des vitesses s'effectue après le calcul des positions, en utilisant les données et les résultats de celui-ci. Les données supplémentaires (pour le même exemple de la diade RRR) sont les suivantes :

Données d'entrées : $\dot{X}_A$, $\dot{Y}_A$, $\dot{X}_C$, $\dot{Y}_C$ – composantes des vitesses absolues des pôles d'entrée A et C.

Données de sortie : $\dot{\phi}_1$, $\dot{\phi}_2$ – vitesses angulaires absolues des barres AB et CB ; $\dot{X}_B$, $\dot{Y}_B$ – projections de la vitesse du pôle interne B.

Les équations non linéaires de fermeture de chaîne s'écrivent :

$$\begin{cases} AB \cdot \cos\phi_1 - BC \cdot \cos\phi_2 = X_C - X_A \\ AB \cdot \sin\phi_1 - BC \cdot \sin\phi_2 = Y_C - Y_A \end{cases} \tag{4.11}$$

Leur dérivation par rapport au temps conduit au système d'équations linéaires :

10. Pour des raisons d'efficacité, la succession de calculs programmée dans REALISME est légèrement différente. En effet, l'efficacité de la simulation cinématique dépend directement de celle de l'implémentation des fonctions de transfert. Par conséquent, il n'est pas inutile de rechercher l'implémentation la plus adaptée aux particularités du langage de programmation, du système d'exploitation, du matériel etc. Malheureusement, aucune conclusion d'ordre général ne peut être tirée.

$$\begin{cases} -AB \cdot \sin\phi_1 \cdot \dot{\phi}_1 + BC \cdot \sin\phi_2 \cdot \dot{\phi}_2 = \dot{X}_C - \dot{X}_A \\ AB \cdot \cos\phi_1 \cdot \dot{\phi}_1 - BC \cdot \cos\phi_2 \cdot \dot{\phi}_2 = \dot{Y}_C - \dot{Y}_A \end{cases} \tag{4.12}$$

Le système formé par les équations 4.12 étant linéaire en $\dot{\phi}_1$ et $\dot{\phi}_2$, les solutions sont obtenues facilement :

$$\begin{aligned} \dot{\phi}_1 &= -\frac{(\dot{X}_C - \dot{X}_A)(X_B - X_C) + (\dot{Y}_C - \dot{Y}_A)(Y_B - Y_C)}{(Y_B - Y_A)(X_B - X_C) + (Y_B - Y_C)(X_B - X_A)} \\ \dot{\phi}_2 &= -\frac{(\dot{X}_C - \dot{X}_A)(X_B - X_A) + (\dot{Y}_C - \dot{Y}_A)(Y_B - Y_A)}{(Y_B - Y_A)(X_B - X_C) + (Y_B - Y_C)(X_B - X_A)} \end{aligned} \tag{4.13}$$

Ensuite, grâce à la dérivation par rapport au temps des équations 4.9 et en utilisant la valeur obtenue pour $\dot{\phi}_1$, les composantes de la vitesse du point B sont obtenues directement :

$$\begin{aligned} \dot{X}_B &= \dot{X}_A - AB \cdot \sin\phi_1 \cdot \dot{\phi}_1 \\ \dot{Y}_B &= \dot{Y}_A + AB \cdot \cos\phi_1 \cdot \dot{\phi}_1 \end{aligned} \tag{4.14}$$

4.2.1.3 Accélérations

Le calcul des composantes des accélérations s'effectue après le calcul des positions et des vitesses, en utilisant les données et leurs résultats. Les données supplémentaires (pour le même exemple de la diade) sont les suivantes :

Données d'entrée : $\ddot{X}_A$, $\ddot{Y}_A$, $\ddot{X}_C$, $\ddot{Y}_C$ – composantes des accélérations absolues des pôles d'entrée A et C.

Données de sortie : $\ddot{\phi}_1$, $\ddot{\phi}_2$ – accélérations angulaires absolues des barres AB et CB ; $\ddot{X}_B$, $\ddot{Y}_B$ – projections de l'accélération du pôle interne B.

Les équations d'accélération sont obtenues par dérivation des équations de vitesse 4.12 et 4.14 par rapport au temps :

$$\begin{aligned} -AB \cdot \sin\phi_1 \cdot \ddot{\phi}_1 + BC \cdot \sin\phi_2 \cdot \ddot{\phi}_2 &= \ddot{X}_C - \ddot{X}_A + AB\cos\phi_1 \cdot \dot{\phi}_1^2 \\ &\quad -BC \cdot \cos\phi_2 \cdot \ddot{\phi}_2^2 && (4.15) \\ AB \cdot \cos\phi_1 \cdot \ddot{\phi}_1 - BC \cdot \cos\phi_2 \cdot \ddot{\phi}_2 &= \ddot{Y}_C - \ddot{Y}_A + AB\sin\phi_1 \cdot \dot{\phi}_1^2 \\ &\quad -BC \cdot \sin\phi_2 \cdot \ddot{\phi}_2^2 && (4.16) \\ \ddot{X}_A - AB \cdot \left(\sin\phi_1 \cdot \ddot{\phi}_1 + \cos\phi_1 \cdot \dot{\phi}_1^2\right) &= \ddot{X}_B && (4.17) \\ \ddot{Y}_A + AB \cdot \left(\cos\phi_1 \cdot \ddot{\phi}_1 - \sin\phi_1 \cdot \dot{\phi}_1^2\right) &= \ddot{Y}_B && (4.18) \end{aligned}$$

Le système formé par les équations 4.15 et 4.16 est linéaire en $\ddot{\phi}_1$ et $\ddot{\phi}_2$. Les solutions sont donc obtenues par les équations 4.19 :

$$\begin{aligned} A &= \begin{bmatrix} -AB \cdot \sin\phi_1 & BC \cdot \sin\phi_2 \\ AB \cdot \cos\phi_1 & -BC \cdot \cos\phi_2 \end{bmatrix} \\ B &= \begin{bmatrix} \ddot{X}_C - \ddot{X}_A + AB\cos\phi_1 \cdot \dot{\phi}_1^2 - BC \cdot \cos\phi_2 \cdot \ddot{\phi}_2^2 \\ \ddot{Y}_C - \ddot{Y}_A + AB\sin\phi_1 \cdot \dot{\phi}_1^2 - BC \cdot \sin\phi_2 \cdot \ddot{\phi}_2^2 \end{bmatrix} \\ \ddot{\phi} &= \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix} = A^{-1}B \end{aligned} \tag{4.19}$$

En remplaçant la valeur obtenue pour $\ddot{\phi}_1$ dans les équations 4.17 et 4.18, les composantes de l'accélération du point B résultent :

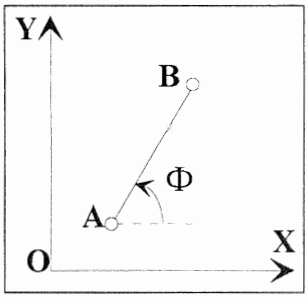


FIG. 4.7 – *Notations pour le moto-élément (groupe actif formé par une manivelle).*

$$\begin{aligned} \ddot{X}_B &= \ddot{X}_A - AB \cdot \left(\sin\phi_1 \cdot \ddot{\phi}_1 + \cos\phi_1 \cdot \dot{\phi}_1^2\right) \\ \ddot{Y}_B &= \ddot{Y}_A + AB \cdot \left(\cos\phi_1 \cdot \ddot{\phi}_1 - \sin\phi_1 \cdot \dot{\phi}_1^2\right) \end{aligned} \tag{4.20}$$

4.2.2 Exemple : le moto-élément

Données internes : AB – longueur de la manivelle (les notations sont indiquées dans la Figure 4.7).
Données d'entrées : X_A, Y_A – coordonnées cartésiennes absolues du pôle d'entrée A ; ϕ, $\dot{\phi}$ et $\ddot{\phi}$ - angle, vitesse angulaire et accélération angulaire de la manivelle (déterminés par la loi de mouvement du moteur).
Données de sortie : X_B, Y_B – coordonnées exactes du pôle de sortie B ; $\dot{X}_B$, $\dot{Y}_B$ – composantes de la vitesse ; $\ddot{X}_B$, $\ddot{Y}_B$ – composantes de l'accélération.

Les calculs sont extrêmement simples. Les coordonnées du pôle de sortie B s'écrivent :

$$\begin{aligned} X_B &= X_A + AB \cdot \cos\phi \\ Y_B &= Y_A + AB \cdot \sin\phi \end{aligned} \tag{4.21}$$

Par dérivation des équations 4.21, on obtient les équations de vitesse :

$$\begin{aligned} \dot{X}_B &= -AB \cdot \sin\phi \cdot \dot{\phi} \\ \dot{Y}_B &= AB \cdot \cos\phi \cdot \dot{\phi} \end{aligned} \tag{4.22}$$

Finalement, par dérivation des équations 4.22, on calcule les composantes de l'accélération :

$$\begin{aligned} \ddot{X}_B &= -AB \cdot (\sin\phi \cdot \ddot{\phi} + \cos\phi \cdot \dot{\phi}^2) \\ \ddot{Y}_B &= AB \cdot (\cos\phi \cdot \ddot{\phi} - \sin\phi \cdot \dot{\phi}^2) \end{aligned} \tag{4.23}$$

4.3 Fonctions de transfert en dynamique inverse

L'analyse en dynamique inverse se propose de déterminer les efforts dans les liaisons cinématiques, pour un mouvement connu (par la simulation cinématique) et pour un ensemble de forces de gravité et de forces extérieures appliquées connu. Au niveau de chaque type de groupe multipolaire, l'expression des efforts en fonction des données d'entrée peut être déterminée a priori, facilitant ainsi les calculs lors de la simulation. A titre indicatif, l'exemple de la diade RRR est présenté ci-après[11].

11. L'objectif de la thèse étant la conception de générateurs de trajectoire, sur des critères purement cinématiques, l'analyse en dynamique inverse n'a pas été implémentée dans REALISME. Néanmoins, tout ce qui est nécessaire, c'est de prévoir une façon d'indiquer les masses et les forces extérieures appliquées, puis d'ajouter les méthodes de calcul des fonctions de transfert aux classes informatiques correspondant aux groupes multipolaires.

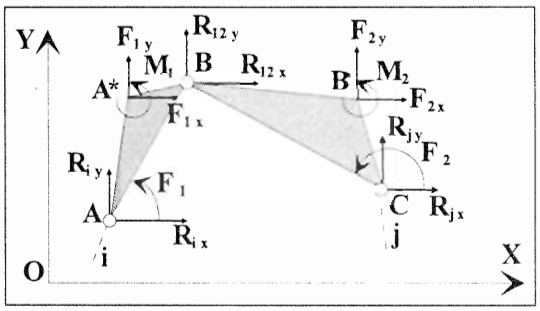


FIG. 4.8 – *Notations pour la diade RRR (calcul en dynamique inverse).*

Les efforts dans les liaisons pivots sont décomposés selon les deux axes du repère cartésien. Dans les liaisons glissières, on calcule l'effort normal sur la droite de translation et le moment. Les éléments des efforts dans chaque liaison sont déterminés par la résolution du système d'équilibre des éléments qui composent les groupes.

4.3.1 Exemple : la diade RRR

Les données peuvent être structurées de la manière suivante (les notations correspondent à la Figure 4.8) :
Données de position : X_A, Y_A, X_B, Y_B, X_C, Y_C – coordonnées des liaisons A, B et C.
Données d'entrée : F_{1_x}, F_{1_y}, F_{2_x}, F_{2_y} – les composantes des forces résultantes F_1 et F_2 sur les éléments 1 et 2 ; M_1 et M_2 – les moments des couples résultants sur les deux éléments ; X_{A^*}, Y_{A^*}, X_{B^*}, Y_{B^*} – coordonnées des points A^* et B^* (points de réduction des systèmes de forces, généralement les centres de gravité des éléments).
Données de sortie : R_{i1_x}, R_{i1_y}, R_{12_x}, R_{12_y}, R_{j1_x}, R_{j1_y} – les composantes des réactions dans les liaisons cinématiques A, B et C.

Les équations de forces s'écrivent en annulant le torseur (F, M) pour chaque solide :

$$\begin{cases} R_{i1_x} + F_{1_x} - R_{12_x} = 0 \\ R_{i1_y} + F_{1_y} - R_{12_y} = 0 \\ R_{i12_x} + F_{2_x} - R_{j2_x} = 0 \\ R_{i1_x} + F_{1_x} - R_{j2_y} = 0 \\ (Y_B - Y_A) \cdot R_{i1_x} - (X_B - X_A) \cdot R_{i1_y} + M_1 + (Y_B - Y_{A^*}) \cdot F_{1_x} - (X_B - X_{A^*}) \cdot F_{1_y} = 0 \\ (Y_B - Y_C) \cdot R_{j2_x} - (X_B - X_C) \cdot R_{j2_y} + M_2 + (Y_B - Y_{B^*}) \cdot F_{2_x} - (X_B - X_{B^*}) \cdot F_{2_y} = 0 \end{cases} \tag{4.24}$$

Le système 4.24 est composé de six équations linéaires par rapport aux six inconnues. Il peut donc être résolu facilement (si le déterminant de la matrice des coefficients est non nul).

4.4 Simulation cinématique

La *simulation cinématique* consiste à déterminer les configurations successives du mécanisme correspondant aux différentes positions des moteurs (cf. *Innocenti* [Inn94]). La simulation cinématique est basée sur l'hypothèse que le mouvement du moteur est connu, les phénomènes dynamiques étant négligés. Pour les générateurs de trajectoire fermée à un seul moteur (ce que nous étudierons essentiellement par la suite), la loi de mouvement du moteur et les phénomènes dynamiques n'ont pas d'influence sur la trajectoire, mais seulement sur la loi de mouvement du point effecteur le long

de la trajectoire. Par conséquent, lorsque la spécification de la trajectoire désirée est indépendante du temps, la simulation cinématique est entièrement satisfaisante. Dans le cas d'une spécification dépendante du temps ou des mécanismes à plusieurs moteurs, la simulation cinématique est une approximation généralement acceptable au début de la conception.

4.4.1 Algorithme

Dans l'approche systémique multipolaire, l'analyse cinématique s'effectue après l'établissement du schéma multipolaire du mécanisme – c'est-à-dire l'identification des différents sous-systèmes (groupes multipolaires) et de leur mode de connexion – et après l'établissement de la relation structurale – c'est-à-dire l'ordre d'enchaînement des groupes à partir du bâti, à travers le(s) moteur(s), vers l'élément effecteur.

Puisque la relation structurale correspond au flux de transmission du mouvement à travers le mécanisme, la simulation cinématique s'effectue d'une manière incrémentale, à partir d'une bibliothèque de fonctions de transfert. Cette bibliothèque contient l'expression formelle des fonctions de transfert préétablie pour chaque type de groupe multipolaire, sous une forme explicite pour les groupes les plus courants (voir §4.2). Pendant le fonctionnement du mécanisme, chaque groupe reçoit du groupe précédent les informations de positions, vitesses et accélérations de ses liaisons d'entrée. Après un calcul interne régi par la fonction de transfert, les paramètres cinématiques des liaisons de sortie sont déterminés et transmis au groupe suivant. Ainsi, une simulation cinématique peut être considérée comme une propagation de mouvement du bâti au groupe terminal (contenant normalement le point effecteur). Les résultats calculés au niveau d'un groupe deviennent ensuite les données d'entrée pour le(s) groupe(s) connecté(s) en sortie. Les calculs se font pour un nombre de pas de temps déterminés par l'utilisateur (voir §6.4.2, p. 93) et en fonction de lois horaires spécifiées pour le(s) moteur(s) (voir §6.4.1, p. 93).

La méthode de simulation s'applique également aux mécanismes à plusieurs degrés de liberté, le concepteur spécifiant les lois horaires pour chaque moteur. Lorsque le mécanisme comprend des groupes plus complexes, pour lesquels la fonction de transfert n'admet pas une forme explicite, la résolution par des méthodes numériques itératives est néanmoins possible. Même dans ce cas, on conserve l'avantage d'une résolution locale, découplée par rapport au reste du mécanisme.

4.4.2 Avantages

4.4.2.1 Rapidité

Ressemblant à la résolution paramétrique de cas dans les modeleurs paramétriques, cette méthode de simulation a l'avantage de repousser au plus tard l'étape numérique et de diminuer le nombre d'itérations de calcul nécessaires. La résolution se fait localement, au niveau de chaque groupe, à travers une sorte de langage interprété qui permet d'obtenir des temps de réponse quasi-instantanés. Ainsi, tandis que les logiciels comme *DADS* ou *ADAMS* comportent une phase de calcul préalable à l'animation graphique, dans REALISME l'image du mécanisme est mise à jour après chaque pas de calcul. Ce parallélisme du calcul et de l'animation est possible grâce à l'efficacité du calcul cinématique.

Une comparaison chiffrée de la vitesse de la simulation cinématique par les différentes méthodes est difficile à faire, puisqu'il faudrait disposer d'une implémentation des méthodes sur la même machine. Néanmoins, dans le tableau 4.2 et dans la Figure 4.9 nous présentons, à titre indicatif, quelques exemples de temps de calcul par rapport à des données rapportées en *Romdhane* [Rom95]. Les trois simulateurs comparés sont : un algorithme de type Newton-Raphson classique (logiciel *DAMS*), l'algorithme basé sur les groupes d'Assur développé par Romdhane et le simulateur de REALISME. Dans l'interprétation de ces résultats il faut tenir compte que REALISME

	Newton-Raphson DAMS	Groupes d'Assur KAMEL	Groupes d'Assur REALISME
quatre barres	112	10	4,4
six barres	140	19	4,6
huit barres	369	non connu	9,2

TAB. 4.2 – *Temps de résolution en position (en secondes, 360 pas de calcul).*

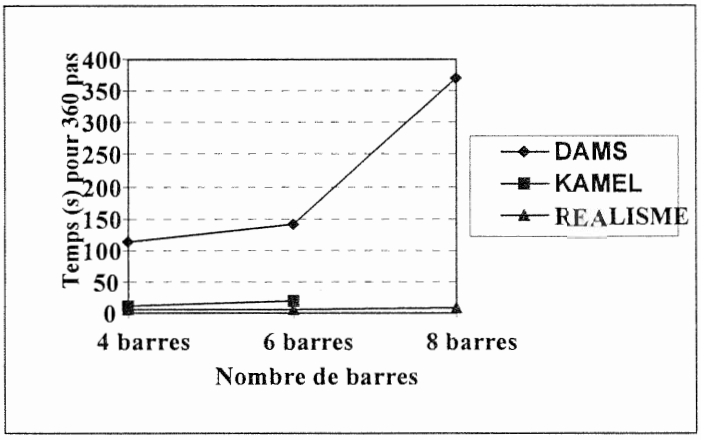


FIG. 4.9 – *Comparaison des temps de simulation pour une rotation complète de la manivelle divisée en 360 pas de calcul.*

est programmé en grande partie dans un langage interprété (Le-Lisp) et tourne sur une station de travail Sun Sparc 10 ; les deux autres logiciels sont programmés en QuickBasic et tournent sur un IBM PS/2.

Il est également à noter que Romdhane signale que, au moins dans cette implémentation, la méthode Newton-Raphson présente des problèmes de convergence lorsque l'incrément entre les positions successives est trop grand. La simulation des mécanismes à six et huit barres a échoué pour un incrément angulaire de la manivelle supérieur à 5°, tandis que dans REALISME elle réussit même pour un incrément de 90°. Dans certains cas, ceci constitue un avantage important de l'approche modulaire par rapport à celles basées sur une résolution itérative (voir aussi §5.3).

Il est important de souligner que le schéma multipolaire et la relation structurale sont uniques pour une même morphologie de mécanisme. Par conséquent, lorsque le concepteur effectue un cycle de modifications dimensionnelles du même mécanisme suivies par des analyses cinématiques, ils ne seront déterminés qu'une seule fois. De ce point de vue, le schéma et la relation structurale jouent le rôle d'une compilation de la morphologie du mécanisme.

4.4.2.2 Identification des causes des blocages et des positions singulières

Une fonctionnalité des plus intéressantes en préconception, spécifique au modèle systémique multipolaire, est la possibilité d'identifier les causes des blocages ou des positions singulières rencontrées en cours de simulation (cf. éq. 4.4 et 4.5). Cela permet au concepteur d'y remédier plus facilement. Cet aspect sera traité plus en détail au §6.4.4.

struct. données / entités du graphe	Schéma cinématique	Graphe associé	Schéma multipolaire
nœuds	liaisons cinématiques	solides (pièces)	groupes multipolaires
arêtes	solides (pièces)	liaisons cinématiques	liaisons cinématiques

TAB. 4.3 – *Les trois types de graphes utilisés en* REALISME.

4.5 Simulation dynamique inverse

L'analyse dynamique s'effectue après l'analyse cinématique et se limite à la dynamique inverse. Les degrés de libertés étant pilotés par des lois horaires imposées, l'objectif de l'analyse est la détermination des efforts et des moments transmissibles, en tenant compte des inerties des barres et des efforts extérieurs appliqués (et connus). Cet objectif est compatible avec la préconception (puisqu'on se propose, le plus souvent, de réaliser un certain mouvement, après quoi on vérifie l'admissibilité des efforts). Les calculs sont faits, évidemment, dans les hypothèses générales énoncées au §1.1.

Comme pour la simulation cinématique, une bibliothèque d'équations formelles, propres à chaque type de groupe, est utilisée (voir §4.3). L'analyse suit le chemin inverse de l'analyse cinématique : elle commence par le dernier groupe (contenant l'élément effecteur) et remonte le calcul des efforts et des moments transmissibles d'une manière incrémentale, pour chaque groupe multipolaire.

4.6 Modélisation informatique

Dans REALISME, trois structures de données principales, implémentées comme des graphes[12], servent à l'analyse cinématique : le schéma cinématique, le graphe associé et le schéma multipolaire. Les entités contenues dans le schéma cinématique et dans le graphe associé sont les liaisons et les barres, tandis que le schéma multipolaire contient des entités à part, les groupes, et des liaisons (voir Tableau 4.3).

La représentation informatique des ces trois entités élémentaires (liaisons, solides et groupes multipolaires) est réalisée grâce à une modélisation objet. Ainsi, pour chaque type d'entité une classe générique est définie. Ensuite, des classes-filles sont définies, afin de modéliser les sous-types de liaisons (pivot, pivot fixé au bâti, glissière, glissière fixée au bâti), de barres (barre simple à deux extrémités ou polygone comportant plusieurs liaisons). Pour les groupes multipolaires, une partie de l'arborescence des sous-classes est donnée dans la Figure 4.10.

Les champs de chaque classe permettent de stocker des informations de différentes natures : numérique, symbolique, logique ou bien des pointeurs vers d'autres entités (coordonnées, longueurs, angles, nom, extrémité de référence, modifiable ou non, etc.). A chaque classe sont également associées différentes méthodes. Ainsi, par exemple, la résolution des équations cinématiques pour un groupe multipolaire, pendant la simulation cinématique, est réalisée par activation d'une méthode héritée de sa classe par l'instance de groupe multipolaire. De même, l'aspect graphique des différentes entités est déterminé par une méthode d'affichage définie pour chaque classe.

L'avantage immédiat de la modélisation orientée objets est la facilité d'extension et de maintenance.

12. La bibliothèque de fonctions Le-Lisp *Grapher*, livrée avec le logiciel Aïda (© Ilog), a été utilisée.

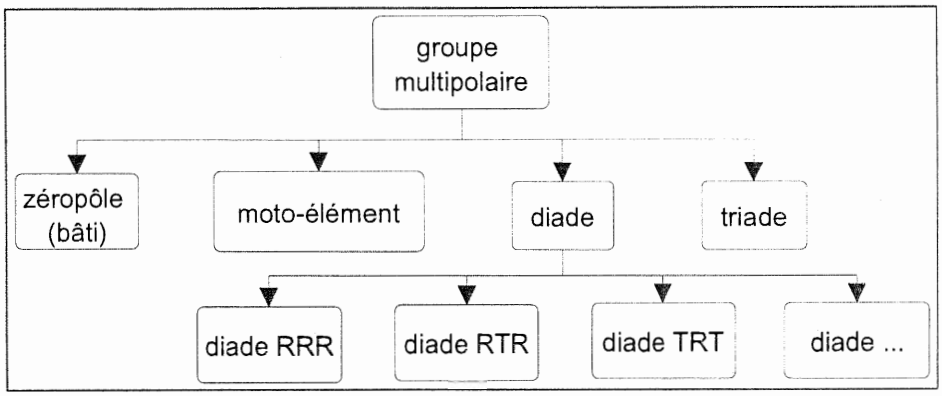


FIG. 4.10 – *Modélisation objet des groupes multipolaires.*

	Compilation morphologie	Mise en équations	Calcul/ Animation	Blocages/Pos. singulières
ADAMS	Détermination hyperstatismes par détermination de rangs de matrices et élimination automatique éq. redondantes	Faite après chaque modification de dimensions	Calcul en bloc, avant animation	Arrêt de la simulation sans explications
SDS			Calcul pendant l'animation (possib. anim. temps réel)	
REALISME	Par détermin. du schéma multip./ relation struct. Info sur lieu où hyperstatisme	Il n'y en a pas. Bibliothèque d'éq. formelles, déjà découplées au maximum	Idem SDS, mais plus rapide	Info. sur le groupe violant une contrainte. Conseils en reconception

TAB. 4.4 – *Comparaison de* REALISME *avec les logiciels ADAMS et SDS.*

4.7 Conclusion

Du point de vue de la théorie des mécanismes, la définition systémique des groupes n'est pas très différente d'autres définitions (par exemple celle de *Verho* [Ver73]). Néanmoins, elle est particulièrement intéressante dans une optique « méthodologies de conception ». En effet, dans son « habillage » systémique, l'approche modulaire permet une représentation plus suggestive, notamment graphique, de la structure des mécanismes et offre un cadre pour appréhender la morphologie et pour concevoir les mécanismes de manière incrémentale, favorisant ainsi l'intégration de ce modèle à des méthodes de préconception. Un deuxième avantage est le lien direct, naturel, avec la méthode d'analyse cinématique, à travers la notion de fonction de transfert, dont les conséquences pratiques sont la rapidité de simulation et l'identification des causes des singularités géométriques.

Le Tableau 4.4 est une comparaison synoptique des fonctionnalités en analyse de REALISME et de deux logiciels réputés du marché, *ADAMS* et *SDS*. Il ne faut cependant pas oublier que les logiciels *ADAMS* et *SDS* sont prévus pour les mécanismes tri-dimensionnels ; la simulation des mécanismes plans est traitée par des algorithmes qui ne leur sont pas spécifiques, comme un cas particulier de simulation tri-dimensionnelle. Il va de soi que la généralité et l'efficacité sont des objectifs contradictoires. En revanche, sous bien d'autres aspects, une maquette comme REALISME ne peut pas rivaliser avec les gros logiciels commercialisés. Dans l'état actuel des choses,

on peut conclure à une certaine complémentarité : tandis que REALISME est plus efficace pour tester rapidement un grand nombre de solutions candidates, *ADAMS* et *SDS* sont, naturellement, meilleurs pour une analyse plus poussée (dynamique, jeux dans les liaisons etc.) une fois qu'une solution approximative a été trouvée.

En ce qui concerne l'analyse, la plupart des caractéristiques de la plateforme REALISME sont semblables à celles d'autres logiciels basés sur une approche modulaire (par groupes d'Assur) (*Suh et Radcliffe* [SR78], *Enderton* [End90], *Salem et Manoochehri* [SM90], *Chieng et Hoeltzel* [CH88], *Młynarski* [Mly96], *M.R. Hansen* [Han92] [Han96]). En plus d'une grande convivialité de l'interface utilisateur – essentielle pour des longs cycles analyse/modification (cf. chap. 6) –, le point fort de REALISME est l'intégration d'un ensemble de méthodes de synthèse (cf. chap. 8, 9 et 10).

Chapitre 5

Modèle qualitatif de cinématique

5.1 Introduction

Traditionnellement, la conception des générateurs de trajectoires, étant motivée par des spécifications cinématiques, comprend une simulation cinématique dont le résultat est une trajectoire indexée dans le temps. Cette trajectoire est représentée généralement par une succession de points ou de positions du point effecteur, correspondant à un bouclage sur des positions du moteur. Une telle trajectoire sera appelée dorénavant une *trajectoire exacte*.

La *cinématique qualitative* développée par Yannou (*Yannou et Vasiliu* [YV95] [YVM95][1]) appartient à la famille des approches *qualitatives* (voir l'état de l'art au §2.5.1 ; cf. également les états de l'art de *Dague et al.* [DY95], *Travé-Massuyès et al.* [TMGY96]). L'idée est d'accepter un « petit sacrifice » quant à la précision numérique, de manière à manipuler des informations plus génériques et à mener un raisonnement de plus haut niveau, plus générique. Pour ce faire, cette cinématique qualitative manipule, à la place des trajectoires exactes, des *trajectoires qualitatives*, composée d'arcs de cercle et de segments.

Du point de vue de la synthèse (morphologique et dimensionnelle), les méthodes de conception existantes sont basées, dans le cas général, sur une connaissance pauvre et mal structurée des interactions entre fonctions, morphologies et dimensions. La cinématique qualitative est une tentative de mieux prendre en compte ces interactions.

Cette cinématique qualitative s'appuie sur des considérations de *Centres Instantanés de Rotation*, des considérations de courbure, ainsi que sur une décomposition systémique multipolaire du mécanisme (cf. chap. 4). Une simulation cinématique qualitative génère directement des trajectoires qualitatives sous la forme d'une liste d'arcs de cercle et de segments de droite. Cela revient à « propager » des morceaux de trajectoire circulaires le long du mécanisme, en partant du moteur jusqu'au point effecteur. Cette propagation permet de générer des *explications qualitatives* sur le fonctionnement du mécanisme, qui sont indépendantes de valeurs particulières des dimensions. Cette fonctionnalité est importante pour aborder le problème de la synthèse, car il peut être envisagé d'inverser ce processus d'explication pour synthétiser directement des classes de mécanismes ayant des propriétés correspondant aux explications.

Esquissée dans *Yannou et Vasiliu* [YV95] [YVM95], cette deuxième voie d'application de la cinématique qualitative n'a pas encore été complètement exploitée ; c'est l'une des perspectives des travaux de Yannou pour les recherches futures et l'une des possibilités de développement de REALISME (cf. aussi §11.3). De plus, pour le problème de la génération de trajectoire il a été montré qu'une optimisation dimensionnelle basée sur une cinématique qualitative est compétitive

1. La présentation de la cinématique qualitative dans ce chapitre est très proche de celles données dans ces publications.

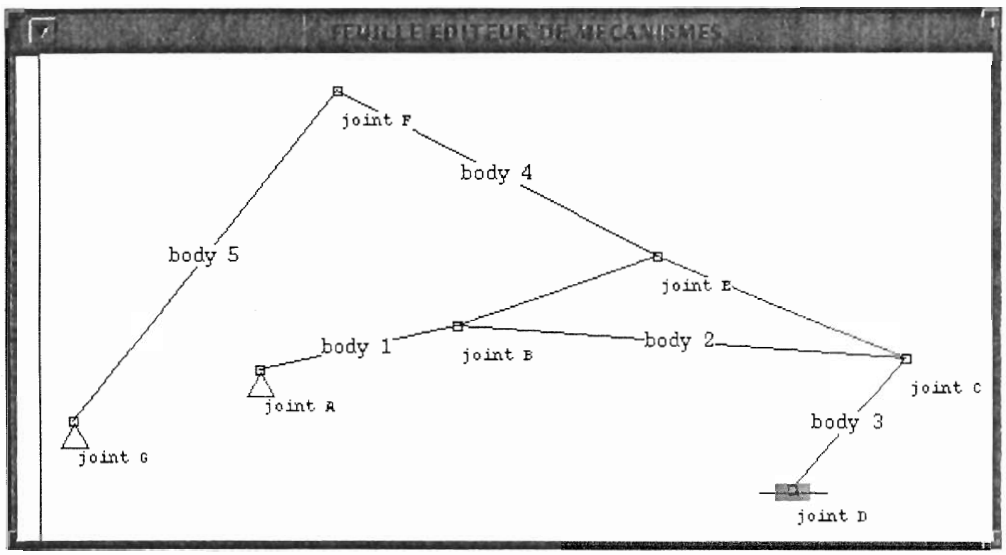


FIG. 5.1 – *Exemple de mécanisme articulé.*

par rapport à une optimisation basée sur une simulation classique. Cette méthode d'optimisation dimensionnelle, dite *qualitative*, a été utilisée dans REALISME (voir §8.3). La notion de trajectoire qualitative ayant montré de nombreuses propriétés en préconception de mécanismes plans et présentant un haut degré de déclarativité, nous l'avons exploitée pour proposer une méthode de synthèse basée sur les cas (cf. chap. 10).

Dans la littérature, seul un article de *Shrobe* [Shr93] utilise une approche relativement proche et applicable à la même classe de mécanismes : mécanismes à liaisons pivots et glissières, à un degré de liberté commandé par une manivelle. Une comparaison des deux approches sera faite au §5.6.

Le mécanisme traité en exemple dans le reste du chapitre est presque identique à celui utilisé par Shrobe. Il s'agit d'un mécanisme articulé (voir Figure 5.1) composé d'un système bielle-manivelle (corps 1, 2 et 3) et d'une diade (corps 4 et 5). La liaison A est une pivot liée au bâti, pilotée par un moteur ; la liaison D est une glissière liée au bâti ; les autres liaisons sont des pivots.

5.2 La trajectoire qualitative

L'entité fondamentale de la cinématique qualitative, aussi bien pour la simulation que pour les explications qualitatives, est la *trajectoire qualitative*. La *simulation cinématique qualitative* (cf. §5.4) s'intéresse à la trajectoire désirée du point effecteur. Pour générer les explications qualitatives, la trajectoire désirée est obtenue, sous la forme d'une liste de points, par une simulation classique pour des dimensions particulières. Dans le cas de l'optimisation, la trajectoire désirée est spécifiée par le concepteur, toujours sous la forme d'une liste de points.

La première opération consiste, dans tous les cas, à approximer cette trajectoire désirée exacte par une trajectoire désirée qualitative sous la forme d'une liste d'arcs de cercle et de segments de droite (voir Figure 5.2), éléments nécessaires à une future simulation cinématique qualitative. Cette approximation a comme unique paramètre une erreur admissible à ne pas dépasser, qui est la distance maximale entre les deux trajectoires. Un panneau de REALISME (voir Figure 8.2, p. 123) permet de commander l'approximation, de choisir l'erreur admissible et de visualiser le résultat.

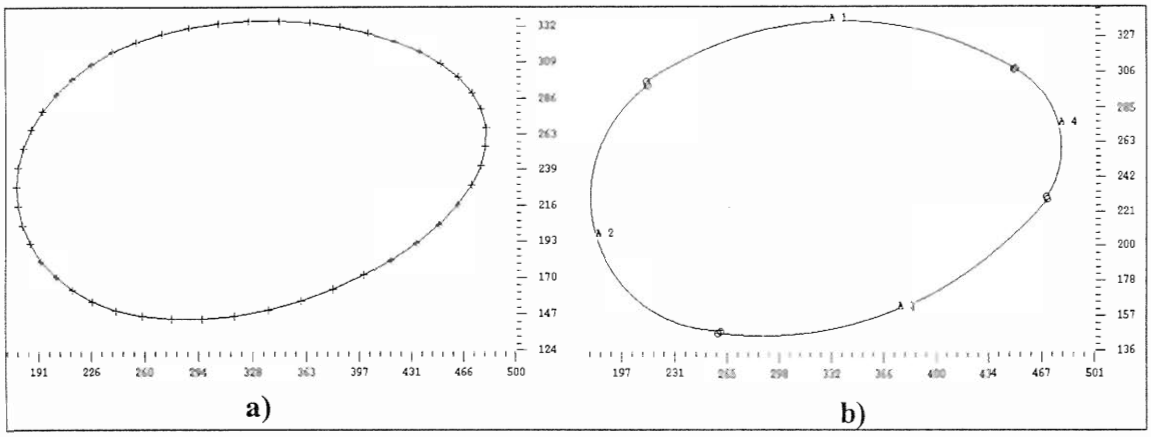


FIG. 5.2 – *Approximation d'une trajectoire désirée* exacte *en (a) (48 points) par une trajectoire désirée* qualitative *en (b) (4 arcs de cercle).*

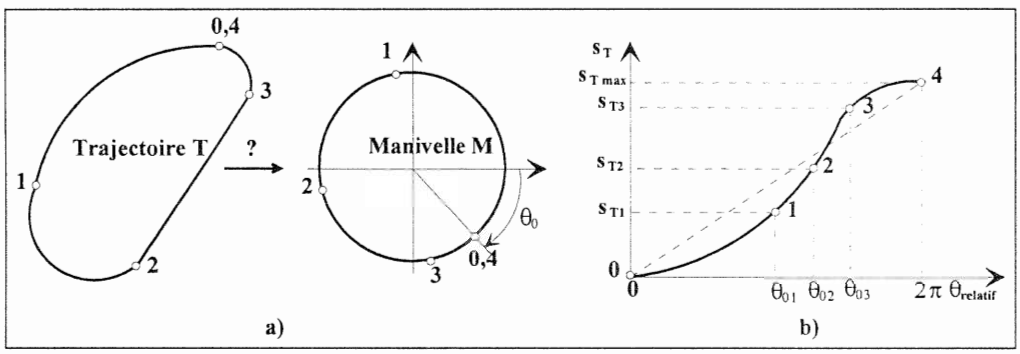


FIG. 5.3 – *(a) Quelle est la correspondance entre les extrémités des arcs et segments de la trajectoire désirée et les positions de la manivelle? (b) La connaissance de la courbe* $s_T = f(\theta_{relatif})$ *permet de déterminer les angles* θ_{0i}.

L'approximation est une minimisation[2], par un algorithme des moindres carrés, du nombre total d'arcs et de segments, tout en respectant l'erreur maximale admissible. A nombre minimal de morceaux qualitatifs de trajectoire, on préfère avoir un maximum de segments, car les segments ont des propriétés qualitatives remarquables. Enfin, le dernier critère est de préférer la solution qui minimise l'erreur.

Il est nécessaire maintenant de faire correspondre les extrémités des arcs et segments de la trajectoire désirée qualitative, notée T par la suite, avec des positions particulières de la manivelle, notée M. Prenons l'exemple de la Figure 5.3.a pour lequel l'approximation a donné un segment et trois arcs. Les quatre positions de la manivelle à déterminer sont paramétrées par l'angle absolu θ_0 pour la position initiale et les trois angles θ_{0i} relativement à cette position.

Dans le cas de la génération d'explications qualitatives, la trajectoire désirée exacte provient d'une simulation classique. En représentant l'abscisse curviligne des points de la trajectoire désirée exacte en fonction de l'angle relatif de la manivelle (voir Figure 5.3.b) et en interpolant linéairement entre les points représentés, on obtient la fonction $s_T = f(\theta_{relatif})$. On détermine ainsi les angles θ_{0i} à partir de la connaissance des abscisses curvilignes s_{T_i} des points extrémités des arcs et segments.

2. L'algorithme utilisé en REALISME a été développé à partir d'un programme écrit par Andrei Sebe (*Institut d'archéologie de l'Académie Roumaine*, Bucarest).

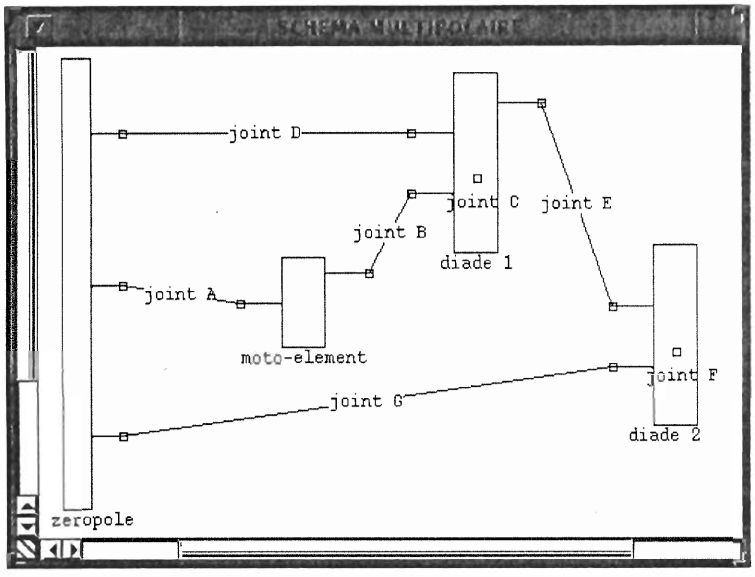


FIG. 5.4 – *Décomposition multipolaire du mécanisme de la Figure 5.1.*

Dans le cas d'une optimisation dimensionnelle, deux cas de génération de trajectoire sont à considérer selon la spécification :

- Le point effecteur doit parcourir la trajectoire désirée avec une temporalité bien définie. Cela revient à imposer $s_T = f(\theta_{relatif})$, c'est-à-dire les θ_{0i} correspondant aux s_{T_i}, ce qui peut avoir un sens pour le concepteur. En effet, si cette courbe est confondue avec la diagonale $((0,0),(2\pi, s_{T_{max}}))$, cela signifie que le parcours de T et la rotation de M se font à des vitesses proportionnelles. En spécifiant une courbe quelconque, le concepteur est à même de définir les zones où le point effecteur a une vitesse curviligne relativement élevée par rapport à la vitesse moyenne définie par la pente de la diagonale, ainsi que les zones de relative accélération et décélération.

- Le point effecteur doit parcourir une trajectoire donnée sans aucune condition de temporalité. Les angles θ_0 et θ_{0i} deviennent alors des variables de conception qui doivent être déterminées au cours du processus d'optimisation. Ceci représente le cas général.

5.3 Utilisation de la décomposition systémique multipolaire

En ce qui concerne la génération d'explications qualitatives du mécanisme (voir §5.5), la décomposition systémique multipolaire (voir chap. 4) identifie automatiquement les *points intéressants* du mécanisme, susceptibles d'intervenir dans ces explications, comme étant les pôles d'entrée et de sortie des groupes multipolaires (voir Figure 5.4), ainsi que le point F pour l'exemple de la Figure 5.1. Pendant la simulation cinématique, on n'utilisera de l'approche multipolaire que la résolution en positions et non celle en vitesses.

La simulation cinématique exacte, préalable à la simulation qualitative, doit être réalisée pour un faible nombre de positions du moteur (correspondant au nombre d'arcs et segments). Or, l'approche multipolaire est plus fiable, lorsque les incréments de position du moteur sont grands, qu'une approche numérique traditionnelle (par résolution numérique de systèmes non linéaires d'équations) (cf. §4.4.2.1). Ceci est un autre avantage du couplage entre la cinématique qualitative et l'approche multipolaire.

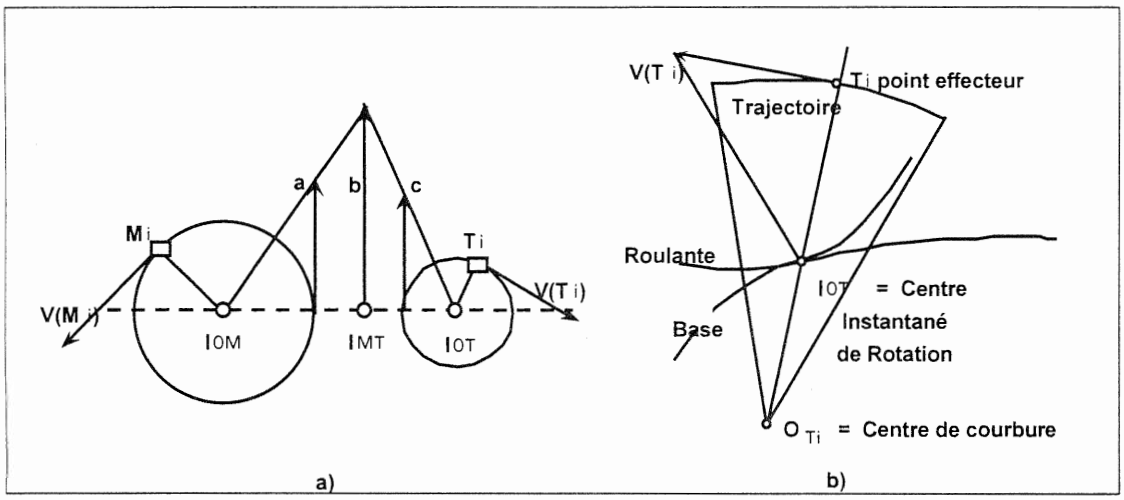


FIG. 5.5 – *(a) Méthode de calcul du vecteur vitesse du point effecteur. Le vecteur $V(M_i)$ et le vecteur* **a** *ont la même norme, il en est de même pour le vecteur $V(T_i)$ et le vecteur* **c**. *Les vecteurs* **a**, **b** *et* **c** *sont orthogonaux à la droite contenant les CIR. (b) Base et roulante.*

5.4 La simulation cinématique qualitative

La simulation cinématique qualitative est le cœur de la méthode de cinématique qualitative. En très peu d'itérations (une pour chaque arc et segment), elle calcule des trajectoires moins précises mais permet d'obtenir un bon temps de calcul (voir §8.3.3) et de produire des explications qualitatives du fonctionnement du mécanisme (voir §5.5).

5.4.1 Une approche basée sur les Centres Instantanés de Rotation

Pour une période de temps infinitésimale, le mouvement relatif entre deux corps i et j est « localement » équivalent à une rotation de centre I_{ij}, appelé *Centre Instantané de Rotation* (CIR) entre les corps i et j. Lorsque i et j sont reliés par une liaison pivot, I_{ij} est constamment confondu avec le centre de cette pivot et, par conséquent, est fixe par rapport à i et j. Une liaison glissière étant équivalente à une liaison pivot dont le centre est « rejeté » à l'infini dans une direction orthogonale à la glissière, le CIR relatif des deux corps i et j reliés par la glissière est lui aussi « rejeté » à l'infini dans la direction D_{ij}. Une propriété remarquable de ces CIR est donnée par le théorème des trois plans glissants :

Théorème 5 *Les CIR relatifs de trois corps i, j, k : I_{ij}, I_{jk} et I_{ik}, sont alignés.*

Considérons maintenant une position particulière de la manivelle pour laquelle la vitesse de rotation de la manivelle est connue. Le CIR entre le bâti 0 et la manivelle M, noté I_{0M}, est confondu avec le centre de la liaison pivot correspondante. La méthode utilisée pour déterminer le vecteur vitesse du point effecteur, relativement au bâti, est expliquée graphiquement dans la Figure 5.5.a ; elle s'appuie sur la connaissance préalable des deux CIR : I_{MT} et I_{0T}, ainsi que sur celle de la position du point effecteur.

Il est intéressant de remarquer que lorsque les points I_{MT} et I_{0M} sont confondus, le point effecteur a localement une vitesse nulle.

Un algorithme a donc été développé, qui fournit la méthode de calcul des CIR manquants à partir des CIR connus et du théorème des trois plans glissants, en minimisant l'intervention de CIR intermédiaires. D'après le théorème des trois plans glissants, deux CIR connus, I_{ij} et I_{jk},

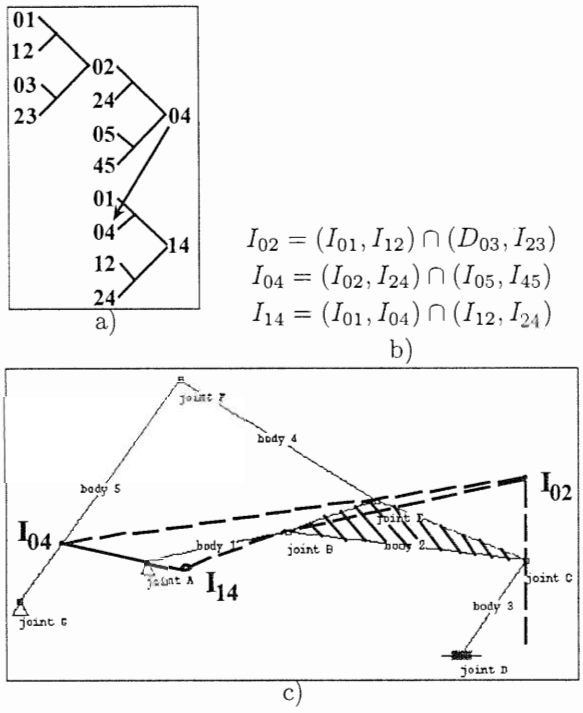


FIG. 5.6 – *(a) Graphe de déduction causale des CIR manquants à partir des CIR connus. (b) Trace formelle des déductions séquentielles des CIR, en termes d'intersections de droites, que l'on peut représenter graphiquement en (c)*

caractérisent une droite passant par I_{ik}, notée ΔI_{ik}. De la même manière, un CIR connu I_{ij} et une direction D_{jk} provenant d'une liaison glissière, caractérisent une droite ΔI_{ik}. Le processus élémentaire de l'algorithme est la déduction d'un CIR I_{ik} à partir de deux droites ΔI_{ik} connues.

Prenons le mécanisme de la Figure 5.1 et supposons que la configuration du mécanisme est connue. Six CIR sur quinze (C_6^2) sont donc déjà connus : I_{01} est en A, I_{12} est en B, I_{23} est en C, I_{24} est en E, I_{45} est en F, I_{05} est en G. Le point effecteur est pris au point E, mais un point intéressant est le point F (voir aussi §5.5). Ces points sont tous les deux solidaires du corps 4. Les deux CIR à déterminer étant I_{MT} et I_{0T}, ils correspondent dans notre exemple à I_{14} et I_{04}. L'algorithme de détermination des CIR manquants va, dans le cas présent, ne faire intervenir qu'un CIR intermédiaire : I_{02}. L'algorithme procède en deux étapes :

1. établissement du graphe causal de déduction des CIR (voir Figure 5.6.a) ;

2. à partir de ce dernier, génération d'une trace formelle des déductions (voir Figure 5.6.b), qui peut être représentée graphiquement sur le mécanisme (voir Figure 5.6.c). Cette trace formelle établit la séquentialité à respecter ; en effet, le calcul d'un CIR à un moment donné doit se faire à partir de CIR ou directions déjà connus.

Cet algorithme n'est appliqué qu'une seule fois pour un type de mécanisme et un point effecteur donnés. La trace formelle des déductions est utilisée pour générer automatiquement la procédure informatique contenant les équations explicites des coordonnées des CIR manquants et de la vitesse du point effecteur (comme il est expliqué dans la Figure 5.5.a). La génération de cette procédure optimisée, ne nécessitant aucun calcul itératif, peut être considérée comme une compilation du mécanisme. Le fait qu'il n'y ait pas de calcul itératif pour déterminer la vitesse du point effecteur est un excellent résultat d'un point de vue rapidité d'exécution ; même un calcul en vitesses par l'approche multipolaire ne peut pas toujours échapper à des calculs itératifs.

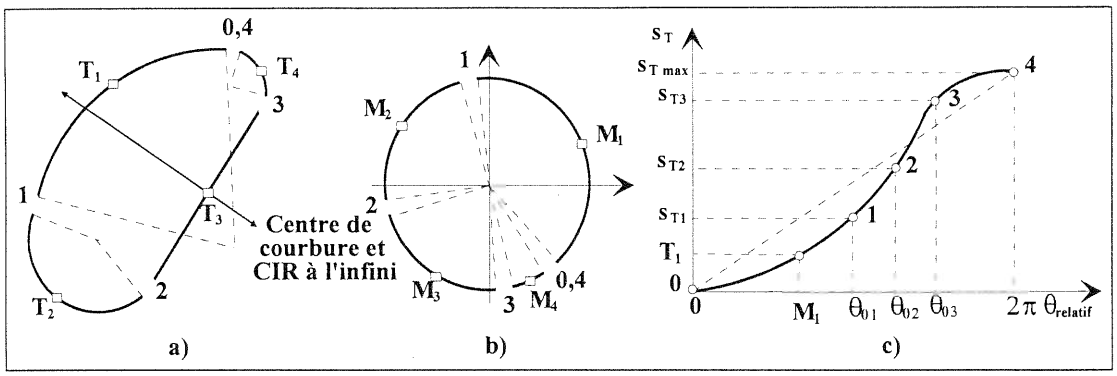


FIG. 5.7 – *(a) La trajectoire qualitative désirée. (b) Les positions remarquables de la manivelle sont déduites de la courbe de la Figure 5.3.b. En disant que les milieux des arcs qualitatifs des deux courbes correspondent temporellement, on commet l'approximation linéaire (c).*

5.4.2 Courbure des arcs qualitatifs, base et roulante

La connaissance de la vitesse du point effecteur n'est pas une fin en soi. Ce qui importe est la détermination du morceau de trajectoire qualitative, centré sur le point effecteur. Le centre et le rayon de courbure de la trajectoire T au point effecteur sont donc à déterminer. Mais le centre de courbure n'est toutefois pas le CIR I_{0T}. En effet, I_{0T} est « localement » le centre de rotation du corps terminal par rapport au bâti mais, l'instant suivant, I_{0T} a bougé. Ce phénomène peut être compris en considérant le lieu des CIR relativement au bâti, à savoir la base, ainsi que le lieu des CIR relativement au corps terminal, à savoir la roulante. Il peut être montré que le roulement sans glissement de la roulante, solidaire du corps terminal, sur la base, décrit parfaitement le mouvement de ce corps terminal. Il est clair sur la Figure 5.5.b que le CIR I_{0T}, le centre de courbure O_{T_i} et le point effecteur T_i sont alignés.

Les problèmes de courbure en cinématique plane ont été largement étudiés par les mathématiciens des dix-huitième et dix-neuvième siècles (cf. *Veldkamp* [Vel63]). Ainsi, le rayon de courbure peut être calculé par l'équation d'Euler-Savary (cf. *Hartenberg et Denavit* [HD64], *Sandor et Erdman* [SE84]). Bien que cette équation puisse être résolue par une méthode graphique, il est difficile d'extraire, de cette manière, des explications suffisamment simples (qualitatives) pour être utiles en conception. C'est la raison pour laquelle une autre méthode a été utilisée pour déterminer le rayon de courbure (voir §5.4.3). Néanmoins, dans l'optique de la génération d'explications qualitatives de haut niveau, il est important de considérer les notions de cercle d'inflexion (lieu des points effecteurs sur le corps terminal, dont la trajectoire présente un segment de droite) et cubique de courbure constante (lieu des points effecteurs dont la trajectoire contient réellement un arc de cercle).

5.4.3 Propagation d'arcs de cercle le long du mécanisme

Considérons maintenant la trajectoire d'un point M de la manivelle à une distance R de I_{0M}. L'approximation qualitative de la trajectoire désirée exacte et la connaissance de la fonction $s_T = f(\theta_{\text{relatif}})$ (voir Figure 5.3.b) ont permis de déterminer les angles θ_0 et θ_{0i} qui vont piloter la simulation qualitative. Ces angles représentent les extrémités des arcs de cercle de la trajectoire du point M. Considérons maintenant les points milieux des arcs de la manivelle, notés M_i (voir Figure 5.7.b) et les points milieux des arcs de la trajectoire désirée qualitative, notés T_i (voir Figure 5.7.a).

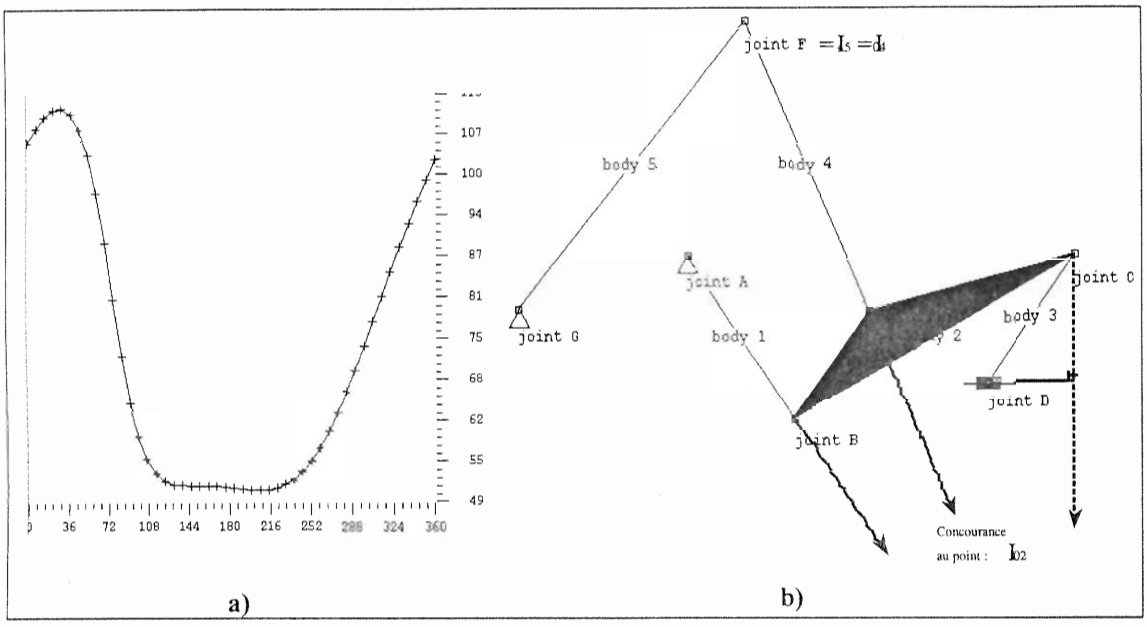


FIG. 5.8 – *(a) La courbe de l'angle du corps 5 en fonction de l'angle de la manivelle présente un palier. (b) Représentation graphique des propriétés dimensionnelles qui concourent à cette fonction, déduites automatiquement.*

En disant que les points M_i correspondent aux points T_i, on commet une approximation linéaire de la fonction : $s_T = f(\theta_{\text{relatif}}) = f(\frac{s_M}{R})$ (voir Figure 5.7.c). Cette approximation signifie qu'on considère le rapport des vitesses curvilignes sur les deux trajectoires constant sur une portion d'arc. Cette approximation linéaire est, en règle générale, tout à fait acceptable.

Pour une simulation cinématique qualitative, à l'inverse, les arcs de cercle de la trajectoire T doivent être calculés à partir des arcs de cercle de la manivelle. Pour chaque position du point de la manivelle en M_i, une résolution en positions est effectuée en parcourant le graphe de décomposition multipolaire du mécanisme ; les positions de T_i et des liaisons sont donc déterminées. Le vecteur vitesse $V(T_i)$ du point effecteur est déterminé par la procédure spécifique générée à partir de la trace formelle de déduction des CIR. Ce vecteur permet de déduire l'orientation de l'arc et sa longueur. En effet, puisque le rapport des vitesses curvilignes a été supposé approximativement constant sur une portion d'arc, la longueur curviligne des deux arcs respecte le même rapport. En définitive, l'amplitude de vitesse est considérée égale à la longueur de l'arc.

Le calcul des vitesses, à partir de la manivelle jusqu'au point effecteur, en passant par des CIR intermédiaires, peut être considéré comme une propagation le long du mécanisme. Puisqu'un vecteur vitesse est quelque peu équivalent à un arc de cercle (excepté pour le rayon de courbure), une simulation cinématique qualitative peut être vue comme une propagation de morceaux de trajectoire circulaires de la manivelle jusqu'au point effecteur.

La méthode choisie pour calculer le rayon de courbure de l'arc au point effecteur consiste à l'obtenir par une deuxième résolution en positions très proche du milieu de l'arc T_i. Il est clair sur la Figure 5.5.b que cette seconde position détermine le centre de courbure car ce dernier est situé sur la droite (I_{0T}, T_i).

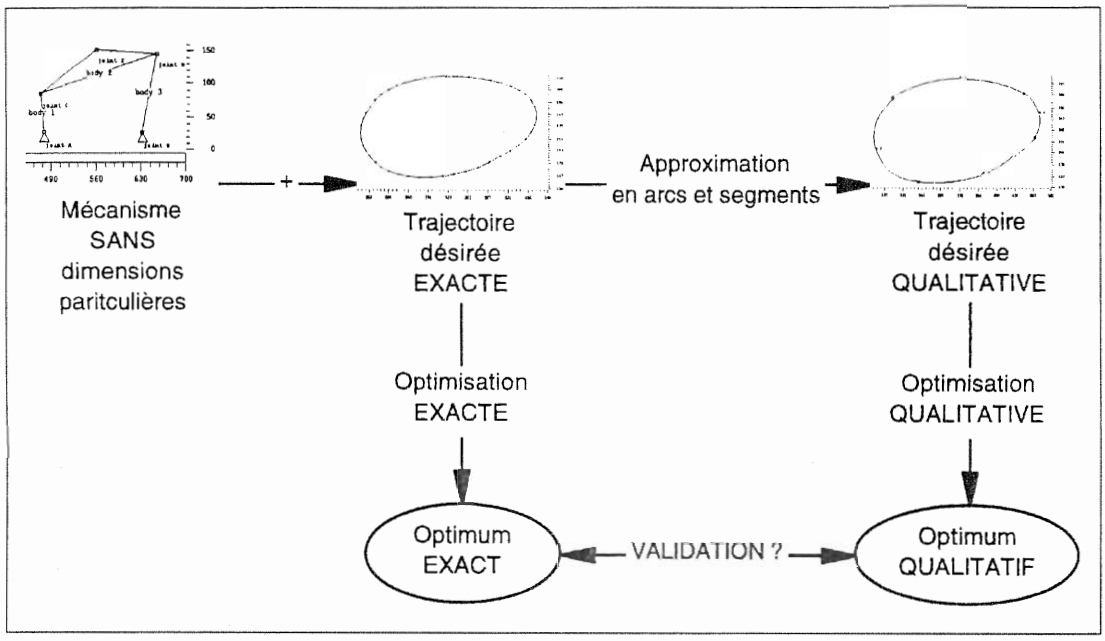


FIG. 5.9 – *Principe général de la génération d'explications qualitatives.*

5.5 Vers une compréhension qualitative du mécanisme

Reprenons le mécanisme de la Figure 5.6.c, dont le point effecteur est E. La trajectoire exacte de E est donnée dans la Figure 5.2.a. Celle-ci est approximée par 4 arcs avec une bonne précision (voir Figure 5.2.b). Ce mécanisme présente un comportement particulier de palier, au niveau de la rotation du corps 5 (voir la Figure 5.8.a), durant une rotation d'environ 140° de la manivelle.

Ceci s'explique par le fait que quand le point effecteur parcoure l'arc $A3$ (voir Figure 5.2.b), la longueur de la barre 4 est telle que le point F est constamment au centre de courbure de l'arc $A3$. La méthode de génération d'explication ne va pas seulement décrire le phénomène, mais elle va expliquer pourquoi il apparaît pour ces dimensions particulières du mécanisme. Le point F a été détecté comme étant un point intéressant pour les explications. En essayant de calculer la vitesse du point F, pour la position correspondante au milieu de l'arc $A3$: T_3, un cas particulier est identifié (voir §5.4.1) où I_{45} et I_{04} sont confondus (moyennant une certaine précision numérique). Ainsi, il est prouvé formellement que le corps 5 s'immobilise « localement » pour la position correspondant à T_3. C'est donc une condition nécessaire pour qu'il y ait une immobilisation sur une période de temps (palier).

La seconde condition nécessaire est que le centre de courbure soit confondu avec I_{45} (point F), ce qui est immédiatement vérifié. En ce qui concerne la première condition, la méthode est capable de générer des explications bien plus pertinentes en explicitant pourquoi, dans cette position particulière, les dimensions du mécanisme ont concouru à confondre I_{45} et I_{04}. En effet, d'après le graphe causal de déduction des CIR, nous savons que I_{04} est obtenu par les intersections : $I_{02} = (I_{01}, I_{12}) \cap (D_{03}, I_{23})$ et $I_{04} = (I_{02}, I_{24}) \cap (I_{05}, I_{45})$. Comme I_{45} et I_{04} sont confondus, il est aisé d'en déduire que les trois droites (I_{01}, I_{12}), (I_{45}, I_{24}) et (D_{03}, I_{23}) sont concourantes au point I_{02}. Ce résultat peut être présenté graphiquement au concepteur : « les barres 1, 4 et la droite verticale passant par C sont nécessairement concourantes » (voir Figure 5.8.b).

De cette manière, à partir de dimensions particulières d'un mécanisme de morphologie donnée, une compréhension générale de ce mécanisme a été acquise (voir Figure 5.9). Cette compréhension spécifie des conditions dimensionnelles à respecter pour retrouver la fonction de palier. Cette

généralisation de la connaissance reliant morphologie, dimensions et fonction, à partir d'une simulation spécifique, peut constituer une avancée importante vers un système performant de synthèse conjointe morphologie/dimensions.

5.6 Comparaison avec l'approche de Shrobe

Dans l'approche de Shrobe, il ne s'agit pas d'une nouvelle méthode de cinématique. En effet, Shrobe prend en considération les résultats d'une simulation traditionnelle, c'est-à-dire les trajectoires exactes décrites par les points intéressants du mécanisme. Ces trajectoires sont approximées par des arcs de cercle et des segments. Les points intéressants sont, principalement, les points d'entrée et de sortie des modules constructifs (*building blocks*), comparables aux groupes multipolaires. Ensuite, la méthode de Shrobe détecte certaines similarités entre les propriétés des trajectoires qualitatives, par exemple un intervalle de temps identique pour deux arcs. Une *compréhension* du mécanisme est obtenue par la génération de conjectures portant sur ces propriétés qualitatives. Les explications qualitatives de Shrobe, analogues à celles d'un ingénieur, utilisent deux mécanismes de déduction causale : une décomposition modulaire obtenue par un système de gestion de contraintes géométriques et un processus de calcul des CIR.

Le but de l'approche de Yannou est d'arriver à inverser ce processus d'explication. En effet, le modèle de Shrobe ne comporte pas de connaissance explicite, comme la cinématique basée sur les CIR et la théorie de la courbure. Ainsi, les explications qualitatives obtenues sont hypothétiques et incomplètes. De plus, les conjectures ne sont judicieuses que pour certaines dimensions particulières du mécanisme. De toute façon, il n'est pas nécessaire de recourir à des conjectures, car les relations entre, d'une part, un arc de la trajectoire désirée, l'arc correspondant de la manivelle et les autres points intéressants et, d'autre part, les dimensions particulières du mécanisme peuvent être déterminées à partir de la propagation des CIR et des considérations de courbure.

Pour reprendre l'exemple du mécanisme *dwell* (présentant un palier), le modèle de Shrobe explique l'existence du palier par la rotation du point E sur un arc de cercle autour du point F avec un certain rayon particulier, tandis que le modèle de Yannou identifie comme cause de l'existence de cet arc la concurrence de trois droites et met en relation cette propriété avec les dimensions du mécanisme.

5.7 Conclusion

Le présent chapitre introduit un nouveau modèle de cinématique qualitative pour les mécanismes articulés plans, basé sur une connaissance profonde de la cinématique : considérations de CIR et courbures. Ce modèle revient à propager des morceaux circulaires de trajectoire, de la manivelle jusqu'au point effecteur, grâce à un graphe causal de déduction de CIR. Ce dernier est utilisé pour extraire des explications qualitatives du fonctionnement du mécanisme, à partir d'une simulation avec des dimensions particulières. Cette connaissance reliant la morphologie, les dimensions et les fonctions est essentielle pour aborder la synthèse conjointe morphologie/dimensions. La simulation cinématique qualitative peut être utilisée avantageusement au cœur d'une optimisation dimensionnelle pour le problème de la génération de trajectoire (voir §8.3). La notion de trajectoire qualitative a été également exploitée dans la méthode de synthèse basée sur les cas présentée au chapitre 10. Enfin, étant donnée la souplesse d'utilisation des CIR, le modèle est, en principe, généralisable à plusieurs degrés de liberté, à des moteurs non reliés au bâti, à des moteurs linéaires (embarqués sur des glissières) et à des liaisons de type engrenage.

Troisième partie

Méthodologies de préconception

Chapitre 6

Conception incrémentale : cycles modification/analyse

6.1 Introduction

Couramment, la préconception de mécanismes est un processus itératif comprenant un bouclage sur deux phases :

- définition de la morphologie et des dimensions (définition d'une solution candidate) ;
- analyse cinématique du mécanisme (afin de valider la solution candidate par rapport aux contraintes du cahier des charges).

Malheureusement, les logiciels les plus courants (*ADAMS*, *DADS*, *SDS*...), conçus essentiellement pour l'analyse dynamique des mécanismes tridimensionnels, sont relativement mal adaptés en phase de préconception des mécanismes plans à tâche purement cinématique. Ces logiciels sont d'une utilisation trop lourde et nécessitent un grand nombre de manipulations pour la définition du mécanisme ; souvent, l'analyse comprend une étape de calcul, distincte de l'étape d'animation graphique ou d'affichage des trajectoires. De plus, surtout à proximité des singularités, les algorithmes d'analyse présentent des problèmes de convergence. Ensuite, la modification de la définition initiale du mécanisme est également difficile, surtout pour les modifications de morphologie. Nous pensons que cette lourdeur ne s'explique pas uniquement par des déficiences de l'interface utilisateur de ces logiciels. En partie, elle est inhérente à la complexité des modèles et des méthodes de simulation utilisés.

En phase de préconception, pour la plupart des mécanismes plans, l'emploi dans REALISME du *modèle systémique multipolaire* (présenté au chap. 4) et du *modèle qualitatif* (chap. 5), ainsi que les caractéristiques de l'implémentation informatique de ces modèles et des méthodes d'analyse, permettent de disposer d'une plateforme plus adaptée aux besoins spécifiques de la préconception. Ainsi, le concepteur est incité à explorer un espace de solutions beaucoup plus large et la qualité de la solution finalement retenue sera souvent améliorée.

6.2 Construction et modification du mécanisme

6.2.1 Morphologie

Pour les concepteurs, le moyen le plus habituel et intuitif de représentation de la morphologie est le schéma cinématique. Le logiciel REALISME fournit un éditeur de schémas cinématiques (voir

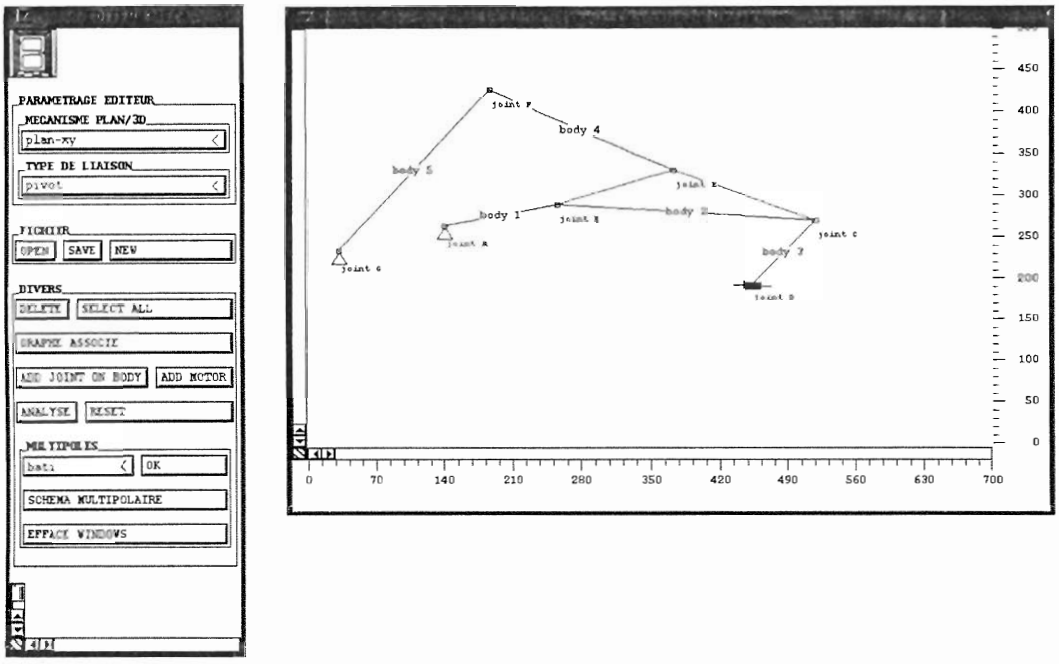


FIG. 6.1 – *L'éditeur de schémas cinématiques. A l'aide des menus et des boutons dans le panneau a), le concepteur construit ou modifie le schéma cinématique dans le panneau b).*

Figure 6.1) permettant la construction « à main levée » du schéma cinématique. Pour déterminer la morphologie, les dimensions présentes dans le schéma n'ont pas de signification : le concepteur « dessine », en quelque sorte, un exemple de schéma cinématique ; la morphologie est déduite automatiquement à partir de cet exemple. Du point de vue informatique, ce schéma est modélisé par une structure de graphe, dont les nœuds correspondent aux liaisons et les arêtes correspondent aux solides du mécanisme. Des symboles graphiques sont associés aux types de liaisons, tandis que les barres sont figurées par des segments reliant les liaisons. Le concepteur choisit le type de liaison dans une bibliothèque de liaisons (pivot, pivot lié au bâti, glissière, glissière liée au bâti), par sélection dans un menu. Ensuite, l'emplacement des liaisons dans le plan est fixé à la souris. La construction d'une barre est réalisée en reliant à la souris les symboles graphiques de deux liaisons situées aux extrémités de la barre.

Si le concepteur maîtrise la modélisation systémique des mécanismes par les groupes multipolaires, la possibilité lui est donnée de travailler directement sur le schéma multipolaire (voir §4.1.2.1), c'est-à-dire un graphe dont les nœuds sont les groupes et les arrêtes sont les liaisons faisant la connexion entre les groupes. (Sans les dimensions, ou plus exactement en lui affectant des dimensions arbitraires à modifier par la suite, le schéma cinématique peut être déduit automatiquement à partir du schéma multipolaire.) La construction directe du schéma multipolaire a plusieurs avantages. D'abord, il est beaucoup plus simple, car il contient beaucoup moins d'éléments que le schéma cinématique. Ensuite, pour des morphologies complexes qui ne sont pas parfaitement connues dès le départ, il est plus facile de mener la construction de la morphologie à l'aide d'une composition par groupes, en rajoutant successivement des groupes, que de tenter de trouver directement l'assemblage de barres souhaité.

Cela ne signifie pas qu'on peut maîtriser ainsi le rapport entre la morphologie et la fonction ou le comportement cinématique du mécanisme. En particulier, on ne peut pas prévoir la conséquence du rajout d'un groupe sur la fonction cinématique (cf. aussi §3.5). Néanmoins, la construction du mécanisme par la connexion des groupes permet une recherche raisonnée dans l'espace des morphologies, notamment en maîtrisant le degré de liberté du mécanisme. A titre d'illustration,

la Figure 6.2 présente la succession d'étapes de construction d'un mécanisme complexe. De plus, lorsqu'on essaye de concevoir un mécanisme permettant la rotation complète de la manivelle, il est possible de procéder incrémentalement, en vérifiant cette propriété pour chaque mécanisme intermédiaire. En effet, en construisant le mécanisme de cette manière, il est possible d'effectuer une simulation cinématique après chaque étape de construction.

A part la vérification *a posteriori*, la méthode systémique peut donc constituer directement un guide dans la construction d'un mécanisme. A partir d'un certain nombre de solides et de liaisons, le nombre de morphologies possibles croît très rapidement. Au lieu d'essayer aveuglement (ou guidé par l'expérience) des morphologies quelconques (dont on s'efforcera ensuite d'estimer la validité morphologique), il est préférable de réaliser le mécanisme progressivement, selon le principe de construction des mécanismes par adjonction de groupes multipolaires.

La morphologie d'un mécanisme complexe comme celui de la Figure 6.2 est difficilement maîtrisable à première vue. Cependant, sa construction incrémentale montre que son noyau est constitué par un simple mécanisme bielle-manivelle ; en sortie de ce mécanisme, d'autres groupes multipolaires à deux solides et trois liaisons (diades) sont connectés successivement. Vue de cette manière, la structure du mécanisme n'apparaît plus comme arbitraire et se voit conférer une certaine logique. En plus, à chaque adjonction de groupe, le respect des conditions morphologiques énoncées précédemment peut permettre d'éviter certaines erreurs et retours-arrière dans le processus de conception.

Une fois construites (ou à n'importe quel moment, d'ailleurs), les morphologies peuvent être sauvegardées dans un fichier. Si besoin est, elles pourront subir des modifications, par élimination ou rajout de solides et de liaisons (ou de groupes). Par exemple, l'élimination d'une barre est obtenue par sélection à la souris suivie de l'emploi de la commande d'effacement dans un menu. Les schémas cinématiques et multipolaires peuvent donc être manipulés dans REALISME de la même manière qu'un dessin dans un logiciel de dessin vectoriel type Windows ou Macintosh.

6.2.2 Dimensions

Comme il a été dit, il est possible de définir la morphologie (dans l'éditeur de schémas cinématiques ou de schémas multipolaires) sans s'intéresser aux dimensions. Mais, en réalité, lors de la création du schéma cinématique, les dimensions sont automatiquement déduites : la définition des dimensions est faite en même temps que celle de la morphologie. Les valeurs des coordonnées des liaisons et des longueurs des barres sont ainsi implicitement déterminées. Néanmoins, toutes ces valeurs numériques peuvent être indiquées, dans l'éditeur de schémas cinématiques, indifféremment à la souris ou par saisie des valeurs précises. Par exemple, un clique avec le bouton de droite de la souris sur une liaison ou sur une barre ouvre le panneau d'édition approprié (voir Figure 6.3). Ces panneaux permettent de saisir les coordonnées exactes des liaisons et de spécifier la longueur et l'angle des barres et la géométrie des barres comportant plus de deux liaisons. Le paramétrage des barres se rapporte à une des extrémités[1], que l'utilisateur peut choisir avec un menu (cf. Figure 6.3.b). En résumé, le concepteur a donc le choix entre une définition précise et une définition approximative, visuelle (ou une combinaison des deux) des dimensions.

Avec ce type d'interface, apporter un changement aux dimensions est donc aussi simple que déplacer une liaison à la souris à une nouvelle position ou saisir de nouvelles coordonnées, longueurs ou inclinaisons de barres. Les modifications sont faites de façon séquentielle, une à la fois, ce qui évite les problèmes de cohérence. L'utilisateur peut à tout moment enregistrer les dimensions courantes ou réutiliser des dimensions (et morphologies) enregistrées précédemment. De plus, lorsque les dimensions du mécanisme sont telles que la simulation pour une rotation complète de la mani-

1. Ce paramétrage de la géométrie n'est évidemment pas le seul possible. En toute généralité, il faudrait pouvoir définir une géométrie paramétrée comme dans les logiciels de type *ProEngineer*.

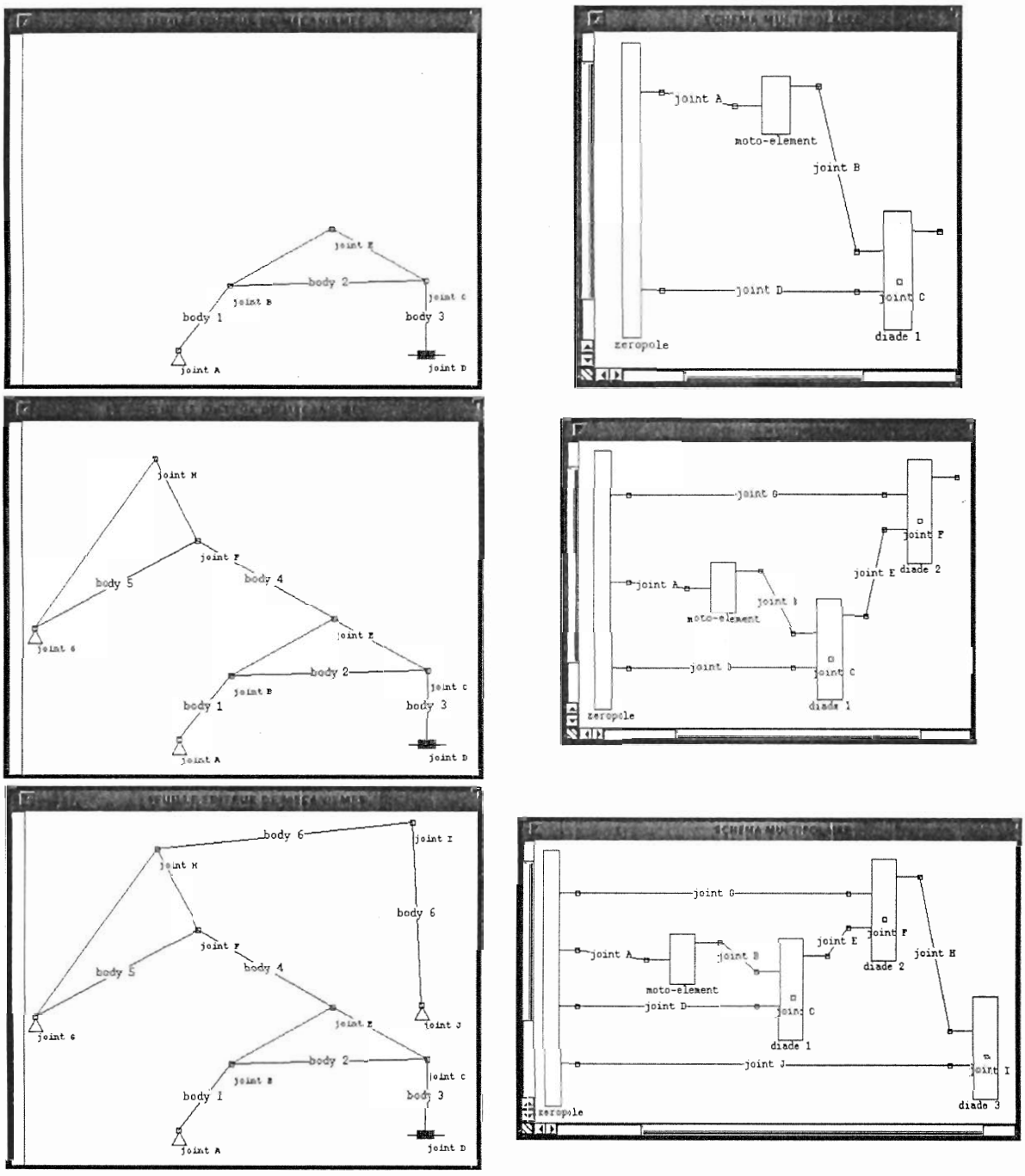


FIG. 6.2 – *Du haut vers le bas, les étapes des construction incrémentale de la morphologie (schéma cinématique, représenté à gauche) d'un mécanisme complexe, par rajout de groupes multipolaires (schéma multipolaire correspondant représenté à droite). A chaque étape, le degré de liberté du mécanisme est égal à un, donc une simulation cinématique est possible.*

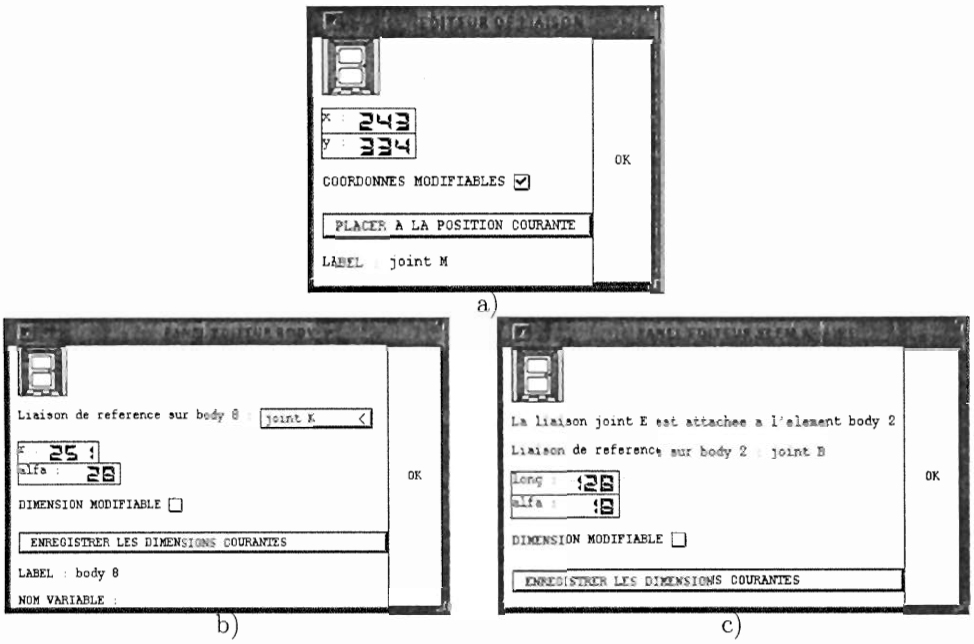


FIG. 6.3 – *Panneaux d'édition des paramètres dimensionnels. a) : Position d'une liaison. b) Dimensions d'une barre. c) Position d'une liaison supplémentaire solidaire avec une barre.*

velle n'est pas possible, les messages de REALISME qui s'affichent automatiquement indiquent les modifications de dimensions susceptibles de permettre la génération d'une trajectoire fermée (voir les détails au §6.4.4, p. 94).

6.3 Spécification de la trajectoire désirée

Pour les générateurs de trajectoire, l'analyse cinématique doit toujours aboutir à la comparaison de la trajectoire générée avec la trajectoire désirée. Il faut donc donner au concepteur les moyens de spécifier la trajectoire désirée. Pour des raisons liées aux types de problèmes de synthèse, aux méthodes de synthèse et aux préférences personnelles du concepteur, plusieurs façons de spécifier cette trajectoire sont possibles. Conformément à l'optique de type « boîte à outils » de REALISME, ces modalités différentes doivent pouvoir coexister dans le même logiciel.

6.3.1 Spécification avec indépendance du temps

S'il n'existe pas de contraintes sur la vitesse de parcours de la trajectoire par le point effecteur, celle-ci peut être spécifiée de nombreuses manières :

- par la spécification précise d'un certain nombre de points appartenant à la trajectoire (mais sans conditions temporelles) ;
- par le dessin qualitatif ou l'esquisse de la forme de la trajectoire.

Dans le premier cas, le concepteur peut écrire un fichier ASCII contenant la liste des coordonnées des points de passage. Ce fichier est ensuite chargé dans REALISME et la trajectoire correspondante peut être visualisée avec l'éditeur de trajectoires (voir Figure 6.4). Dans le deuxième cas, REALISME permet d'utiliser le même éditeur pour dessiner la forme de la trajectoire. Cette opération est réalisée par l'intermédiaire d'une courbe spline. Ainsi, le concepteur place à la souris une succession de points de contrôle d'une courbe spline. Dès que le nombre des points placés est suffisant (4 pour

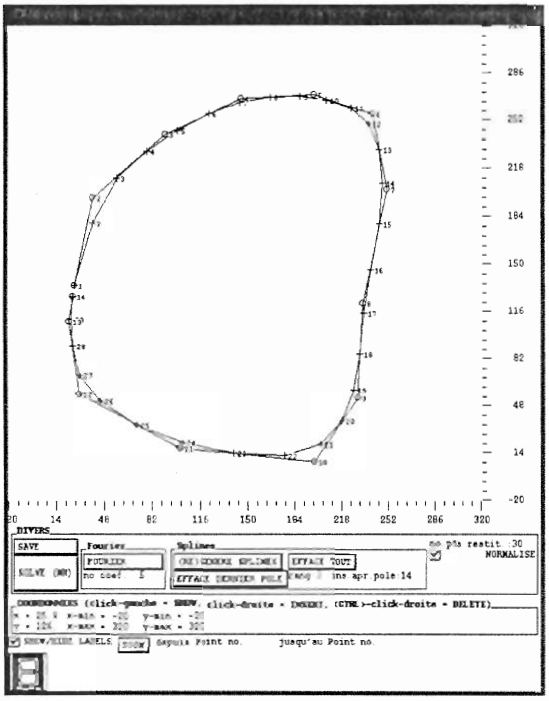


FIG. 6.4 – *L'éditeur de trajectoires désirées.*

des splines de rang 3), la courbe spline est automatiquement visualisée par un ensemble de points équirépartis sur la courbe et dont le nombre est un paramètre modifiable. De cette manière, le concepteur spécifie la trajectoire par manipulation des points de contrôle – l'éditeur permet à tout moment le rajout et l'effacement de points de contrôle – tout en regardant les conséquences sur la courbe spline.

6.3.2 Spécification avec dépendance du temps

La synthèse d'un générateur de trajectoire avec dépendance du temps signifie que le mouvement du point effecteur le long de la trajectoire doit être coordonné en temps avec celui du moteur (mécanismes à un degré de liberté uniquement). Dans ce cas, la manière usuelle de spécifier la trajectoire est d'imposer des points de passage du point effecteur sur la trajectoire en nombre égal au nombre de positions du moteur consécutives et équidistantes. Pour les moteurs de rotation (manivelle), on considère n positions de la manivelle, équiréparties temporellement sur un temps de cycle (dans le cas général cela correspond à une équirépartition angulaire pour une vitesse constate). La spécification de la trajectoire désirée consiste donc à spécifier n points de passage correspondant à ces instants. Il s'agit d'un mode de spécification de la trajectoire très contraignant ; il est très difficile de concevoir un tel mécanisme par essai-erreur, à moins de partir d'une solution initiale proche de l'optimum (et à condition que cet optimum existe...). Il est également possible de spécifier les coordonnées des points de passage à partir de calculs ou de déterminations expérimentales préalables. Dans ce cas, les coordonnées sont transcrites dans un fichier ASCII, qui peut être ensuite chargé dans REALISME. Pour certains tests, il est également possible de charger une trajectoire générée par un mécanisme, obtenue antérieurement par simulation et sauvegardée dans un fichier.

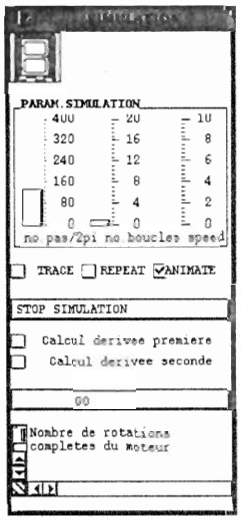


FIG. 6.5 – *Panneau de commande de la simulation cinématique.*

6.4 Simulation cinématique

Une fois terminée la définition morphologique et dimensionnelle du mécanisme, le logiciel REALISME permet d'effectuer la simulation cinématique. Un panneau est prévu à cet effet (voir Figure 6.5).

6.4.1 Lois de mouvement des moteurs

Au préalable, les lois de mouvement du ou des moteurs – dont dépendent les résultats de la simulation – doivent être spécifiées par le concepteur (ou alors des lois par défaut sont utilisées). La procédure est très simple. Un clique souris sur le groupe multipolaire correspondant au moteur (dans la fenêtre d'édition du schéma multipolaire) donne accès à un panneau d'édition de la loi du moteur (voir Figure 6.6). Là, il est possible de spécifier la vitesse (angulaire ou linéaire) du moteur, soit comme une valeur constante, soit comme une expression quelconque fonction du temps (ou, éventuellement, d'autres variables internes du logiciel). Des mouvements alternatifs peuvent donc être spécifiés ; dans ce cas, l'incrément utilisé durant la simulation est directement un incrément de temps.

6.4.2 Paramètres de la simulation

Le panneau de la Figure 6.5 permet également de fixer, à la souris, d'autres paramètres de la simulation :

- nombre de pas de calcul par cycle du moteur (donc incrément de position du moteur) ;
- nombre de boucles de simulation (cycles du moteur) ;
- variables à calculer durant la simulation (par défaut, seulement les positions des liaisons et les angles des barres sont calculés ; si besoin, les dérivées première et seconde peuvent être calculées également) ;
- animation graphique simultanée pendant la simulation ou calculs uniquement ;
- simulation pour un seul cycle du moteur ou répétée jusqu'à une commande explicite d'arrêt.

FIG. 6.6 – *Editeur de la loi de mouvement des moteurs.*

Lorsque la simulation est déclenchée par un clique sur le bouton approprié de ce panneau, deux opérations préalables à la simulation sont exécutées automatiquement et de façon transparente à l'utilisateur : la décomposition en groupes multipolaires (algorithme présenté au §4.1.2.1) et la détermination de la relation structurale (§4.1.2.2). Il est important de noter que ces opérations ne sont faites qu'une seule fois pour la même morphologie, autrement dit, lors d'un changement des dimensions (à une autre itération du processus essai-erreur), elles ne seront pas refaites.

L'animation graphique du mouvement du mécanisme est obtenue « en temps réel », la position du mécanisme étant actualisée après chaque pas de calcul, c'est-à-dire après chaque modification de position du ou des moteurs.

6.4.3 Réduction du temps de mise en équation et des calculs itératifs

La décomposition en groupes multipolaires et la détermination de la relation structurale sont réalisées dans REALISME dans un temps très court (de l'ordre de la seconde). De plus, ces déterminations ne sont effectuées qu'une seule fois pour la même morphologie. Par conséquent, lors des cycles de modifications dimensionnelles, l'analyse cinématique bénéficie de ce facteur commun et il n'est plus nécessaire de refaire la mise en équations.

Ainsi, l'analyse cinématique basée sur l'approche par groupes mutlipolaires a un caractère paramétrique. Cela revient à repousser au plus tard l'étape numérique du calcul et à faire une sorte de compilation formelle de la morphologie du mécanisme.

Pour la plupart des mécanismes plans usuels, la résolution cinématique par cette approche ne requiert pas un calcul numérique itératif. L'efficacité de la simulation est un facteur critique dans la préconception : elle encourage le concepteur à réaliser beaucoup de cycles modification/analyse, autrement dit à explorer un grand nombre de solutions candidates.

6.4.4 Identification des causes des blocages et des positions singulières

Un autre avantage de la résolution locale consiste dans la possibilité d'identifier des causes – et donc des remèdes – au cas où aucune solution numérique valide ne peut être calculée, c'est-à-dire lorsque le mécanisme arrive à un dysfonctionnement (blocage ou position singulière). En effet, les logiciels de type *DADS*[2] (cf. par exemple *Schiehlen* [Sch90]) ne fournissent dans cette situation

2. Le logiciel *MECAmaster*, dans un contexte différent, est capable d'identifier les causes des hyperstatismes et de proposer des solutions.

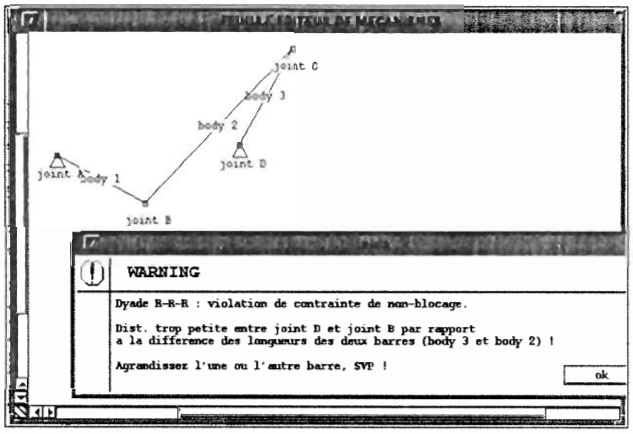


FIG. 6.7 – *Position singulière en cours de simulation d'un mécanisme* quatre barres *(cf. Figure 4.5). Grâce au caractère paramétrique de la simulation, la cause est identifiée et des modifications sont suggérées.*

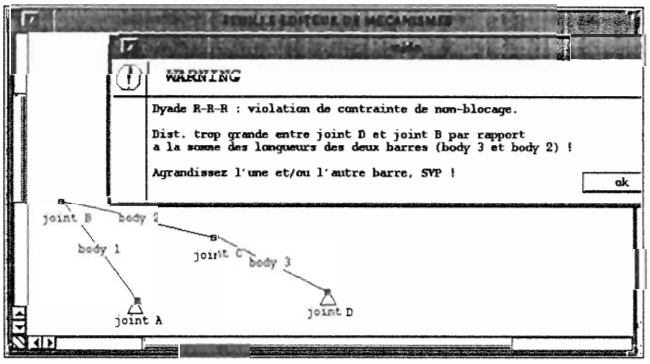


FIG. 6.8 – *Blocage en cours de simulation d'un mécanisme* quatre barres *(cf. Figure 4.5). L'origine du blocage est identifiée et des modifications sont suggérées.*

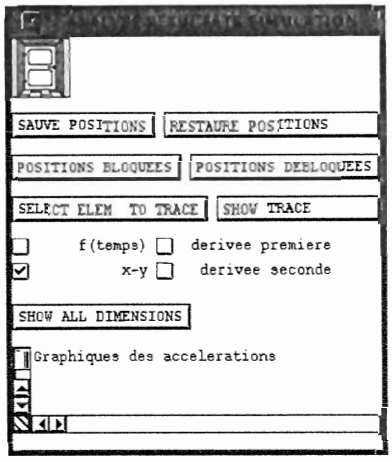


FIG. 6.9 – *Panneau permettant l'analyse des résultats de la simulation.*

aucune information, sauf un message d'erreur. Puisqu'on dispose, dans REALISME, d'un paquet de contraintes géométriques pour chaque type de groupe multipolaire (voir pour la diade éq. 4.4 et 4.5, p. 65), l'absence de solution numérique ne provient plus du non-respect d'une condition mathématique de solvabilité d'un système d'équations, mais de la violation d'une condition géométrique au niveau d'un groupe bien identifié. Les Figures 6.7 et 6.8 présentent deux exemples de message qui s'affiche dans REALISME. Ce message est très précis, c'est-à-dire il indique la contrainte géométrique violée et propose des solutions pour y remédier. Dans les cas les plus simples, ces solutions sont relativement évidentes, mais l'utilité de ces indications augmente pour les mécanismes complexes.

6.5 « Chiffrer » la distance entre les trajectoires désirée et générée

Le but de l'analyse cinématique est de comparer le comportement d'un mécanisme avec les besoins. Dans le cas des mécanismes générateurs de trajectoire, le comportement est représenté par la trajectoire du point effecteur. Après la simulation cinématique, le concepteur peut afficher la trajectoire de différents points du mécanisme (essentiellement centres géométriques de liaisons), soit en $x - y$, soit chacune des coordonnées par rapport au temps (ou paramètres des moteurs, lorsqu'ils sont constants). Le panneau de la Figure 6.9 permet de choisir l'entité pour laquelle on désire visualiser le comportement (trajectoires des liaisons et évolution de l'angle des barres en fonction du paramètre du moteur).

6.5.1 Distance exacte

L'interface de REALISME permet d'afficher en superposé les trajectoires exactes désirée et générée (voir Figure 6.10). Une comparaison visuelle est donc possible. La valeur numérique de la distance entre les deux trajectoires est également calculée et affichée (la procédure de calcul sera donnée plus tard).

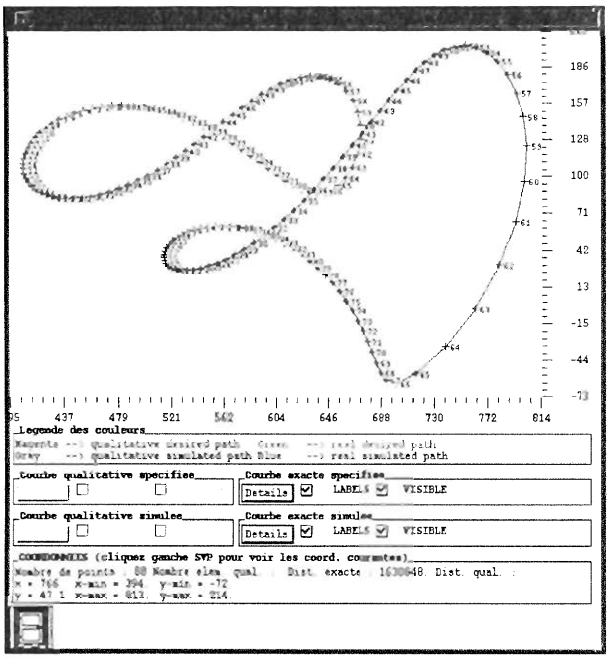


FIG. 6.10 – *Représentation superposée des trajectoires exactes désirée et générée.*

6.5.2 Distance qualitative

La comparaison des trajectoires qualitatives[3] désirée et générée s'effectue de la même manière que pour les trajectoires exactes (voir Figure 6.11).

3. Les trajectoires qualitatives sont une approximation en arcs de cercle d'une trajectoire exacte.

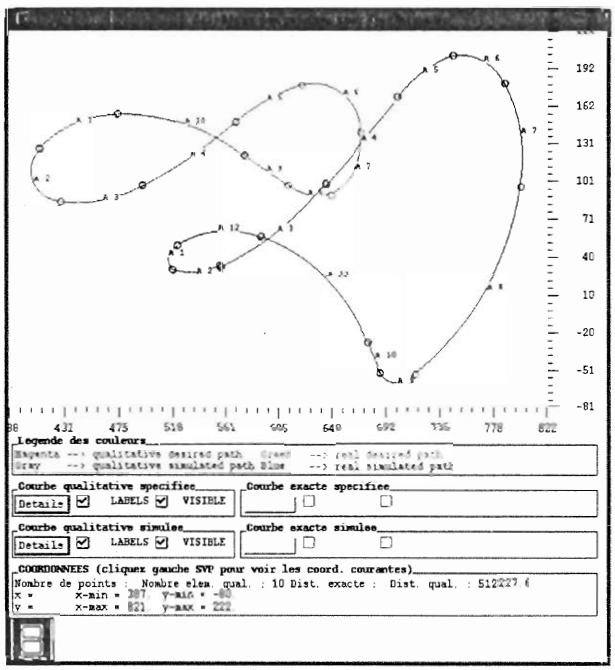


FIG. 6.11 – *Représentation superposée des trajectoires qualitatives désirée et générée.*

Chapitre 7

Un outil pour la synthèse morphologique : la représentation de graphes

7.1 Les graphes et la conception de mécanismes

Un graphe est un modèle expressif pour les mécanismes. Il est une représentation de haut niveau de la connectivité des pièces ; d'autres informations comme le type des liaisons et les dimensions des barres peuvent être stockées comme attributs des entités du graphe. Dans REALISME, nous avons fait usage de ce modèle comme structuration des données et comme support pour certains algorithmes (détermination du schéma multipolaire et de la relation structurale, simulation cinématique). Dans l'état de l'art, nous avons passé en revue la modélisation des mécanismes par des graphes (§2.1.1) et ses applications à la synthèse morphologique (§2.3.1), notamment la génération automatique de graphes associés.

La démarche générale de synthèse peut être résumée par le schéma de la Figure 7.1 qui distingue la conception de la reconception. Pour la conception, il s'agit d'une démarche descendante $a \Rightarrow b \Rightarrow c$: (a) à partir de spécifications du cahier des charges de haut niveau (nombre de degrés de liberté, de pièces, de chaînes cinématiques, etc.), un ensemble de graphes de mécanismes sont énumérés ; (b) l'ensemble de mécanismes correspondant à chaque graphe est généré ; (c) les solutions candidates ainsi obtenues sont évaluées par rapport aux autres critères de conception. Une autre démarche, ascendante puis descendante $c \Rightarrow b \Rightarrow a \Rightarrow b \Rightarrow c$, est utilisée dans le cas de la reconception à partir de la généralisation d'un mécanisme existant (cf. par exemple *Yan* [Yan92], *Hsu* [Hsu92]).

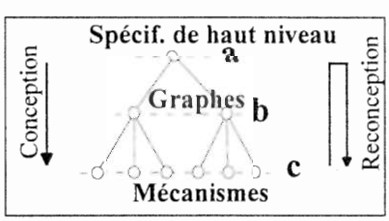


FIG. 7.1 – *Une représentation à trois niveaux pour la* conception *à partir de spécifications et pour la* reconception *à partir d'un mécanisme existant.*

La représentation qualitative et abstraite des mécanismes par des graphes permet au concepteur de mener des raisonnements pour l'analyse structurale et la synthèse. Les concepts, théorèmes et algorithmes de la théorie générale des graphes et les parallèles entre la théorie des graphes et la cinématique peuvent conduire à de nouvelles méthodes de conception (semi)automatique des mécanismes. Par exemple, les coefficients du polynôme caractéristique de la matrice d'adjacence du graphe associé ont des significations physiques et topologiques (*Yan et Hall* [YH81]). Une base de cycles indépendants du graphe correspond à une base de chaînes cinématiques indépendantes. L'utilisation de la modélisation par graphes dans la phase initiale de la conception des mécanismes permet de privilégier dans un premier temps les raisonnements portant sur les spécifications de haut niveau. Les morphologies possibles seront ensuite confrontées aux spécifications plus détaillées.

7.2 Visualisation des graphes : le problème du dépliage

Le plus souvent, les méthodes de conception de mécanismes nécessitent une interaction avec le concepteur. Aussi est-il souhaitable que la modélisation du mécanisme associe la richesse informationnelle – reflétant la fidélité du modèle par rapport à l'objet modélisé – à une transparence par rapport à l'utilisateur. En particulier, le rôle actif du concepteur dans le processus de conception est facilité par une visualisation du modèle d'une manière compréhensible. D'où le problème de la représentation des graphes. Afin d'assurer l'intelligibilité de cette représentation, la contrainte principale est d'éviter les entrecroisements d'arêtes, c'est-à-dire d'obtenir une représentation planaire (dépliée). Une partie des graphes de mécanismes étant non planaires, il faudra minimiser le nombre d'entrecroisements, par une procédure que nous appelons *dépliage du graphe*. Il s'agit d'un problème connu comme étant NP-complet. Le reste du chapitre est consacré à la présentation d'une heuristique originale et efficace de représentation des graphes, mise au point et intégrée dans REALISME.

7.2.1 Notions utiles de la théorie des graphes

La méthode de dépliage que nous proposons utilise des notions et des résultats de la théorie générale des graphes. Nous les présentons brièvement ci-après, en particulier pour les graphes non orientés. (cf. *Berge* [Ber73] [Ber58], *Busacker et Saaty* [BS], *Gondran et Minoux* [GM85], *Christofides* [Chr75], *Derniame* [Der71], *Gaudel et al.* [GSF87], *Kuntzmann* [Kun72], *Livovschi et Georgescu* [LG86], *Mayeda* [May72], *Seshu et Reed* [SR61]).

Un graphe G est défini par un couple (S, A), où S est un ensemble de sommets (ou nœuds) et A est un ensemble de paires de sommets (arêtes ou arcs). La représentation d'un graphe G est dite *dépliée* sur la surface S si le graphe est dessiné sur S sans entrecroisements d'arêtes. Un graphe est *planaire* s'il admet une représentation dépliée dans le plan. Dans le cas contraire, il est *non planaire*.

Un critère de planarité classique est donné par le théorème suivant :

Théorème 6 (Kuratowski) *Un graphe est* non planaire *si et seulement si il contient un sous-graphe homéomorphe*[1] *à l'un des deux graphes non planaires fondamentaux*[2].

1. Deux graphes G et G' sont *homéomorphes* si et seulement si il existe un graphe G'' avec lequel G et G' sont *isomorphes* par le remplacement d'une arête par des *arêtes-série*. Deux graphes G et G' sont *isomorphes* s'il existe une correspondance bijective entre les arêtes de G et de G' respectant les relations d'incidence. Définition des *arêtes-série* : si u_1 et u_2 sont les extrémités de l'arête e_1, v_1 et v_2 les extrémités de l'arête e_2, $u_1 \neq u_2$ et $v_1 \neq v_2$, alors e_1 et e_2 sont des *arêtes-série* si $u_2 = v_1$ et s'il n'y a pas d'autre arête ayant u_2 comme extrémité. (Cf. *Mayeda* [May72].)

2. Il s'agit des graphes *complets* (tous les sommets reliés par une arête) composés de cinq, respectivement six sommets.

Malheureusement, l'utilisation de ce critère n'est pas pratique à cause de son coût exponentiel. *Hopcroft et Tarjan* [HT74] ont proposé un algorithme déterministe pour déterminer si un graphe est planaire ou non, d'une complexité linéaire par rapport au nombre de sommets.

Tous les graphes planaires admettent une représentation dépliée dans le plan qui est telle que toutes les arêtes sont des segments. Une représentation planaire d'un graphe divise le plan dans des surfaces appelées *faces* (ou *régions*)[3]. L'ensemble d'arêtes délimitant une face est appelé sa *frontière*. Il existe toujours une face illimitée qui constitue la *face infinie*. Les autres faces sont appelées *intérieures*.

Une *chaîne* de longueur q est une séquence de q arcs : $L = \{u_1, u_2, \ldots, u_q\}$ telle que chaque arc u_r de la séquence ait une extrémité commune avec l'arc u_{r-1} et l'autre extrémité commune avec l'arc u_{r+1}. Un *cycle* est une chaîne dont les extrémités coïncident. Un *cycle élémentaire* est un cycle ne contenant aucun autre cycle (en parcourant un cycle élémentaire, on ne rencontre pas deux fois le même sommet, sauf l'origine du parcours).

Un nombre de p cycles $\mu^1, \mu^2, \ldots, \mu^p$ sont *dépendants* s'il existe, entre leurs vecteurs associés, une relation vectorielle de la forme :

$$\lambda_1 \vec{\mu}^1 + \lambda_2 \vec{\mu}^2 + \ldots + \lambda_p \vec{\mu}^p = \vec{0} \tag{7.1}$$

(λ_i non tous nuls). Lorsque l'équation 7.1 implique que $\lambda_i = 0$ pour $\forall i = 1, \ldots, p$, les cycles sont dits *indépendants*.

Une *base de cycles* est un ensemble minimal de cycles indépendants $\{\mu^1, \mu^2, \ldots, \mu^p\}$ tel que tout vecteur représentatif d'un cycle μ puisse se mettre sous la forme :

$$\vec{\mu} = \lambda_1 \vec{\mu}^1 + \lambda_2 \vec{\mu}^2 + \ldots + \lambda_p \vec{\mu}^p \tag{7.2}$$

Une *base fondamentale de cycles* par rapport à un *arbre de recouvrement* (cf. la note 9, p. 105) du graphe est l'ensemble des cycles contenant chacun une seule *corde* (cf. la note 7, p. 105).

Le nombre de faces finies est égal à la dimension d'une base de cycles, donnée par la formule d'Euler-Poincaré :

$$s - a + f = 2 \tag{7.3}$$

où s est le nombre de sommets, a le nombre d'arêtes et f le nombre de faces. Dans un graphe planaire, les contours des différentes faces finies constituent une base fondamentale de cycles indépendants. Toute face apparue dans une représentation plane d'un graphe planaire peut devenir une face infinie par une représentation différente du même graphe.

Le nombre minimal d'entrecroisements d'arêtes avec lequel un graphe peut être représenté dans le plan est appelé le *nombre d'entrecroisements (crossing number)* du graphe. La détermination du *nombre d'entrecroisements* d'un graphe quelconque, ainsi que sa représentation dans un plan avec ce nombre minimal d'entrecroisements, est un problème NP-complet.

La visualisation d'un graphe implique l'affectation de coordonnées aux sommets (les arêtes étant représentées par des segments). L'objectif du dépliage des graphes est de trouver une affectation qui minimise le nombre d'entrecroisements. Le dépliage est donc un problème d'*optimisation*.

En pratique, le problème étant NP-complet, seulement des heuristiques peuvent être utilisées. Plusieurs chercheurs ont proposé de telles heuristiques (*Watanabe* [Wat89], *Kamada et Kawai* [KK89], *Fang et Freudenstein* [FF90], *Chieng et Hoeltzel* [CH90], *Makinen et Sieranta* [MS94], *Bertolazzi et al.* [BDBL95], *Mauskar et Krishnamurty* [MK96b] – voir l'état de l'art, §2.1.1.1). Cependant, leurs performances sont difficiles à évaluer. Une chose est néanmoins sûre : la fonctionnalité de dépliage est absente ou donne des résultats décevants dans les logiciels de gestion de graphes du marché (Ilog – *Grapheur*, Rank Xerox – *GraphTalk*).

3. La notion de *face* appartient au graphe topologique (représenté), et non au graphe abstrait.

7.2.2 L'approche numérique

Traditionnellement, une optimisation strictement numérique est utilisée pour le dépliage des graphes. Cette optimisation travaille le plus souvent dans un espace de coordonnées discrétisées, les sommets étant placés sur une grille. A partir d'une affectation initiale aléatoire des coordonnées, l'optimisation tente de réduire le nombre d'entrecroisements, généralement par permutations de sommets. Cette approche n'est malheureusement pas efficace pour plusieurs raisons :

- Pour différentes positions voisines d'un sommet, le nombre d'entrecroisements reste identique. L'espace des coordonnées des nœuds est donc sous-contraint et il est trop vaste par rapport au critère d'optimisation.

- Le calcul de la fonction objectif (nombre d'entrecroisements) à partir des coordonnées des sommets a un coût prohibitif. C'est d'ailleurs la raison pour laquelle la fonctionnelle de minimisation est souvent remplacée par la somme des longueurs des arêtes, au prix d'une moindre qualité des résultats. Pour réduire encore le temps de calcul, à la place de la longueur euclidienne donnée par la norme du vecteur (x, y), $\sqrt{x^2 + y^2}$, on utilise la longueur *de Manhattan* : $|x| + |y|$.

- Le problème présente un grand nombre de minima locaux, car il existe de nombreuses affectations différentes de coordonnées pour le même nombre d'entrecroisements. De plus, certains de ces minima sont insurmontables par le déplacement d'un seul sommet ou par une seule permutation, le déplacement simultané de plusieurs sommets à la fois étant indispensable.

7.2.3 L'approche topologique de Chieng et Hoeltzel

Chieng et Hoeltzel [CH90] ont proposé une approche des plus intéressantes pour synthétiser, pour un graphe planaire, toutes les solutions de dépliage. L'intérêt est que cette synthèse s'effectue dans l'espace topologique des familles de bases fondamentales de cycles plutôt que dans un espace numérique avec tous les défauts que cela représente (voir ci-dessus). Notons que leur approche n'est pas une approche d'optimisation, mais de synthèse qui n'opère que dans le cas des graphes planaires. Nous nous sommes inspirés de cette approche pour proposer un algorithme général d'optimisation du dépliage, pour un graphe quelconque, dans un espace topologique (voir §7.3).

Dans l'approche topologique du dépliage, l'affectation de coordonnées est repoussée à l'étape finale du dépliage, après l'obtention d'une bonne solution topologique. L'entité fondamentale de cette approche est une base fondamentale de cycles du graphe. La justesse du choix des bases fondamentales de cycles comme discriminants à la place des coordonnées est prouvée par une observation pratique : pour une représentation donnée d'un graphe, le nombre d'entrecroisements ne peut être réduit qu'en déplaçant un sommet de l'intérieur d'un cycle vers l'intérieur d'un autre. L'espace topologique ayant une dimension plus petite que l'espace numérique (même discrétisé), un important gain de coût est ainsi obtenu.

Une solution topologique correspond à l'ensemble de relations d'intériorité/extériorité entre les cycles adjacents de la base. Par exemple, pour deux cycles adjacents $C_1 = \{3, 4, 5, 6, 8\}$[4] et $C_2 = \{1, 2, 3, 4, 5, 6, 7\}$ (voir Figure 7.2), trois positions relatives sont possibles : C_1 à l'intérieur de C_2, C_2 à l'intérieur de C_1, et C_1 à l'extérieur de C_2. A chacune de ces positions relatives correspond une représentation dépliée différente.

L'algorithme proposé par Chieng et Hoeltzel consiste en une stratégie *générer et tester*. Pour un graphe abstrait, défini par une de ses bases fondamentales de cycles, toutes les configurations

4. Bien que les cycles soient composés d'arêtes, nous les désignons, pour simplifier l'écriture, par une liste de sommets. Il est sous-entendu qu'ils contiennent aussi l'arête reliant le premier et le dernier sommet de cet ensemble. De plus, il est clair *que ni le* choix du premier élément, ni le sens de parcours du cycle, ne sont pas importants.

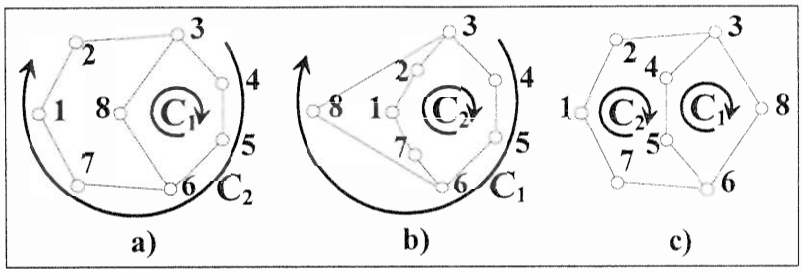


FIG. 7.2 – *Les trois positions relatives pour deux cycles adjacents : a) C_1 à l'intérieur de C_2, notée $(C_1(C_2))$; b) C_2 à l'intérieur de C_1, notée $(C_2(C_1))$; c) C_1 à l'extérieur de C_2, notée $((C_1)(C_2))$.*

topologiques en termes de position relative des cycles sont énumérées. Pour chaque configuration, une procédure spécifique indique si elle peut correspondre à une représentation dépliée. Si une configuration respectant le critère de planarité est trouvée, elle est utilisée pour l'affectation de coordonnées.

L'intérêt majeur de l'approche topologique réside dans l'élimination de la principale source de complexité : la manipulation de coordonnées, remplacées par un espace topologique discret. La solution topologique constitue ensuite l'entrée d'un algorithme déterministe d'affectation de coordonnées. Différents algorithmes d'affichage peuvent ainsi être utilisés pour le même résultat de l'optimisation topologique.

Cependant, l'énumération exhaustive de ces configurations suivie par un test de planarité (algorithme de Chieng et Hoeltzel) a deux inconvénients majeurs :

- le nombre de configurations à énumérer explose avec le nombre de cycles[5] ;
- il est impossible de traiter les graphes non planaires, pour lesquels aucune configuration topologique ne satisfait le test de planarité.

7.3 Notre approche topologique pour l'optimisation

Nous proposons une nouvelle heuristique applicable aux graphes connexes planaires et non planaires (*Vasiliu* [Vas92] [Vas93]). Au lieu d'énumérer et de tester différentes configurations pour une seule et même base fondamentale de cycles, cette heuristique effectue une optimisation, à partir d'une base fondamentale de cycles quelconque, dans l'espace des transformations topologiques possibles de cette base. L'objectif est de trouver si possible une base de *faces*[6] du graphe (cycles situés les uns à côté des autres, comme les pièces d'une mosaïque), sinon une base de cycles minimisant une fonction coût.

7.3.1 Notre fonction coût

Dans l'espace topologique, nous ne pourrons pas estimer directement le nombre d'entrecroisements d'arêtes du graphe. C'est pourquoi, à la place du critère binaire de Chieng et Hoeltzel, indiquant simplement si une configuration correspond à une représentation planaire ou non planaire,

5. 3 configurations pour 2 cycles, 16 pour 3 cycles, 115 pour 4 cycles, 1029 pour 5 cycles, etc.

6. Pour simplifier, nous utilisons le terme de *base de faces* pour désigner une *base de frontières de faces*. Pour comprendre cette notion, il suffit de faire l'analogie entre la base de faces et les contours des pièces composant un puzzle ou une mosaïque (voir aussi §7.2.1).

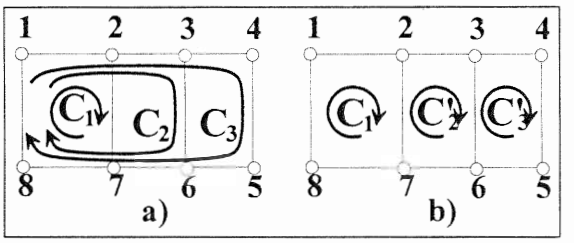


FIG. 7.3 – *Exemple de soustractions de cycles permettant de transformer la base de cycles initiale C_1, C_2, C_3 (a) en la base de faces C_1, C'_2, C'_3 (b).*

il nous faudra une fonctionnelle estimant la mesure dans laquelle une base de cycles correspond à une représentation dépliée. L'idée nous a été suggérée par un théorème de la théorie des graphes :

Théorème 7 (Mac Lane) *Une condition nécessaire et suffisante pour qu'un graphe soit planaire est l'existence d'une base de cycles qui contienne chaque arête précisément deux fois (cf.* Busacker *[BS]).*

La fonction coût à minimiser par l'optimisation sera la somme des occurrences, au-delà du nombre deux, des arêtes dans la base. Autrement dit, si chaque arête a_i apparaît n_i fois dans la base composée de N cycles, alors

$$F = \sum_{i=1}^{N} (\max(n_i, 2) - 2) \tag{7.4}$$

Lorsque cette somme est égale à zéro, la base est une base de faces. On constate effectivement qu'une base de cycles avec un coût réduit correspond à une représentation avec un faible nombre d'entrecroisements. De cette manière, l'algorithme s'applique également aux graphes non planaires.

7.3.2 Modifications topologiques élémentaires

L'optimisation consiste en une succession d'opérations élémentaires de transformation d'une base de cycles. Pour un couple de cycles (faces) adjacents, il est possible de les transformer localement par soustraction (l'opération sera définie plus loin). Des soustractions judicieusement choisies ont pour effet de faire diminuer le nombre d'arêtes partagées par les cycles adjacents et, par ce biais, la valeur de la fonction coût définie précédemment. Le choix de ces soustractions sera fait de manière à obtenir la plus grande réduction possible de la valeur de la fonction coût. Par exemple, supposons que la base de cycles initiale pour le graphe de la Figure 7.3 comprend les cycles $C_1 = \{1,2,7,8\}$, $C_2 = \{1,3,6,7,8\}$ et $C_3 = \{1,2,3,4,5,6,7,8\}$. La configuration topologique correspondante est : C_1 à l'intérieur de C_2, C_2 à l'intérieur de C_3. En appliquant deux opérations de soustraction : $C'_3 = diff(C_3, C_2)$ et $C'_2 = diff(C_2, C_1)$, nous obtenons une base de faces $\{C_1, C'_2, C'_3\}$ (Figure 7.3b).

Les soustractions de cycles sont donc l'étape élémentaire dans notre algorithme d'optimisation. Plus précisément, pour un couple de cycles C_1 et C_2, deux couples de cycles transformés peuvent être générés :

$$(C_1, C_2) \Rightarrow \begin{cases} (C_1,\ C'_2 = diff(C_1, C_2)) \\ (C'_1 = diff(C_1, C_2),\ C_2) \end{cases} \tag{7.5}$$

Les trois couples de cycles – $(C_1, C_2), (C_1, C'_2)$ et (C'_1, C_2) – correspondent aux trois positionnements relatifs de deux cycles adjacents (voir Figure 7.2).

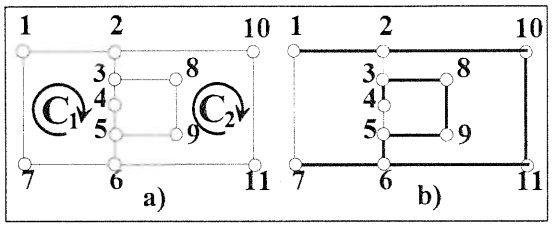


FIG. 7.4 – *a) Configuration impossible : les cycles adjacents* $C_1 = \{1,2,3,4,5,6,7\}$ *et* $C_2 = \{2,3,8,9,5,6,11,10\}$ *ne peuvent pas appartenir à la même base fondamentale de cycles. b) Un exemple d'arbre de recouvrement du même graphe (représenté en traits épais).*

Pour définir l'opération de soustraction de cycles (notée *diff*), qui est propre à notre méthode de dépliage, nous avons établi le lemme suivant :

Lemme 8 *Les sommets et les arêtes communs à deux cycles adjacents appartenant à une base fondamentale forment une séquence contiguë d'arêtes (chaîne).*

Preuve S'il n'y a qu'un seul sommet commun, le problème de la contiguïté ne se pose pas. Examinons le cas où il y a au moins deux sommets communs s_i et s_j. Puisqu'une corde[7] appartient à un seul cycle d'une base fondamentale[8], aucune corde ne peut être parmi les arêtes appartenant à la suite de sommets commune aux cycles C_1 et C_2. Les sommets s_i et s_j appartiennent tous les deux à l'arbre de recouvrement[9] (puisque, par définition, celui-ci passe par tous les sommets). Par conséquent, la seule suite de sommets dans le graphe qui relie s_i à s_j passe uniquement par l'arbre de recouvrement (et pas par une corde). Or, entre deux sommets d'un arbre il n'y a qu'une seule chaîne[10]. Donc cette chaîne est une partie contiguë de la chaîne commune aux deux cycles. Comme ce raisonnement est valable pour n'importe quelle paire de sommets parmi les sommets communs, il est également valable pour la paire formée du premier et du dernier sommet. Donc la chaîne commune est un ensemble contigu de sommets.

De façon intuitive, cette propriété peut être appréhendée sur l'exemple de la Figure 7.4. Une configuration de cycles adjacents (C_1, C_2) appartenant à une base fondamentale, comme celle de la Figure 7.4.a, ne peut pas exister. En effet, pour qu'elle existe, il faudrait qu'il existe un arbre de recouvrement du graphe (dessiné en trait épais dans la Figure 7.4.b) dans lequel le rajout des trois cordes referme trois cycles[11], dont C_1 et C_2. Or, un tel arbre n'existe pas ; en rajoutant la corde $\{1,7\}$ à l'arbre de la Figure 7.4.b, ce n'est pas le cycle C_1 qui est refermé, mais le cycle périphérique $\{1,2,10,11,6,7\}$.

Le lemme 8 permet d'exprimer les deux cycles adjacents de la manière suivante (en faisant abstraction des permutations circulaires) :

$$\begin{aligned} C_1 &= a_1.b.c_1 \\ C_2 &= a_2.b.c_2 \end{aligned} \tag{7.6}$$

7. Pour un arbre de recouvrement a d'un graphe G, une arête de G qui n'est pas en a s'appelle une *corde*. (Pour la notion d'*arbre de recouvrement*, voir la note 9.)

8. C'est le corollaire du théorème suivant : le rajout d'une arête entre deux sommets non adjacents d'un arbre de recouvrement crée un cycle indépendant et un seul.

9. Par définition, l'*arbre de recouvrement* est un sous-graphe connexe d'un graphe connexe qui contient tous les sommets du graphe mais n'a aucun cycle. Tout graphe fini connexe contient au moins un arbre de recouvrement [May72].

10. Démonstration (*Seshu* [SR61]) : s'il n'y avait pas de chaîne, l'arbre ne serait pas connexe ; s'il y en avait deux, ces chemins formeraient un cycle. Or, un arbre ne contient pas de cycle.

11. Cf. la note 8 et la note a dans l'Algorithme 7.1 (p. 107).

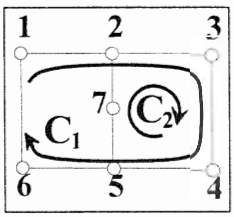


FIG. 7.5 – *Exemple de soustraction de cycles adjacents.*

où a_1, c_1, a_2, c_2 sont des chaînes (séquence contiguës de sommets), b est la chaîne formée par la suite de sommets communs aux cycles C_1 et C_2 et le point (.) est l'opération de concaténation de chaînes[12]. Alors, si on définit l'*inverse* d'une chaîne $\{s_1, s_2, \ldots, s_n\}$ comme étant la chaîne $\{s_n, s_{n-1}, \ldots, s_1\}$, la différence $diff(C_1, C_2)$ peut s'exprimer :

$$diff(C_1, C_2) = a_1.premier(b).inverse(a_2).inverse(c_2).dernier(b).c_1 \tag{7.7}$$

Un deuxième lemme permet de s'assurer de la commutativité de l'opération :

Lemme 9 *A une permutation circulaire et à une inversion du sens de parcours près, l'opération de soustraction des cycles adjacents est commutative :* $diff(C_1, C_2) \Leftrightarrow diff(C_2, C_1)$.

Preuve En utilisant la formule 7.7, on a :

$$diff(C_2, C_1) = a_2.premier(b).inverse(a_1).inverse(c_1).dernier(b).c_2 \tag{7.8}$$

Par ailleurs, pour toute chaîne $c = c_1.c_2. \ldots .c_n$, nous avons :

$$inverse(c) = inverse(c_n).inverse(c_{n-1}). \ldots .inverse(c_1) \tag{7.9}$$

De plus, $inverse(inverse(c)) = c$. Par conséquent, en utilisant les équations 7.8 et 7.9, nous obtenons :

$$inverse(diff(C_2, C_1)) = inverse(c_2).dernier(b).c_1.a_1.premier(b).inverse(a_2) \tag{7.10}$$

En comparant les équations 7.10 et 7.7, on constate que, à une permutation circulaire près, on a :

$$inverse(diff(C_2, C_1) = diff(C_1, C_2)) \tag{7.11}$$

Les opérations $diff(C_2, C_1)$ et $diff(C_1, C_2)$ sont donc équivalentes au sens près.

Pour l'exemple de la Figure 7.5, on a $a_1 = \{1\}$, $b = \{2, 3, 4, 5\}$, $c_1 = \{6\}$, $a_2 = \emptyset$, $c_2 = \{7\}$. En appliquant la formule 7.7, nous obtenons $diff(C_1, C_2) \Leftrightarrow diff(C_2, C_1) \Leftrightarrow \{1, 2, 7, 5, 6\}$.

7.3.3 Algorithme d'optimisation topologique

L'idée fondamentale de l'algorithme est, d'abord, de sélectionner dans la base de cycles un couple de cycles à transformer, puis de choisir la transformation (parmi les deux possibles, voir 7.5) qui rapproche le plus la base de cycles d'une base de faces (selon la fonction coût définie au §7.3.1).

Les étapes de l'optimisation topologique sont présentées dans l'Algorithme 7.1.

12. Soit a_1, soit c_1 peuvent être vides, mais pas les deux puisqu'en ce cas la chaîne formée par les sommets communs se confondrait avec le cycle C_1. Or, cela est impossible, car une chaîne ne peut pas être identique à un cycle. Même raisonnement pour C_2.

Etape 1 Détermination d'une base fondamentale initiale composée de n cycles[a].

Etape 2 Calcul de la fonction de coût (voir équation 7.4). Si elle vaut 0 ou si le nombre d'itérations dépasse la limite fixée, aller à l'étape 5.

Etape 3 Détermination, parmi les $\frac{n(n-1)}{2}$ couples potentiels de cycles, de $(n-1)$ couples de cycles qui ont les plus grandes chances de faire diminuer la fonction coût[b]. Ces couples sont ceux qui ont le plus de sommets en commun. (Voir §7.3.3.1.)

Etape 4 Un seul couple de cycles parmi les n peut être transformé à la fois. La transformation la plus appropriée parmi les $2n$ potentielles est choisie par l'heuristique de la recherche tabou (*tabu search*). (Voir §7.3.3.2.) Ensuite, retour à l'étape 2.

Etape 5 Application d'un algorithme déterministe d'affichage (voir §7.3.4) prenant en compte la base de cycles transformés déterminée précédemment.

[a] Au préalable, un arbre de recouvrement est déterminé (par un algorithme de recherche en profondeur d'abord). Ensuite, en vertu du théorème énoncé dans la note 8 (p. 105), il suffit de trouver le cycle (unique) qui est formé à chaque fois qu'on rajoute à l'arbre une corde.

[b] Le fait de ne considérer, parmi les $\frac{n(n-1)}{2}$ couples potentiels de cycles, que $(n-1)$ couples a comme but, d'une part, de maintenir constant le nombre total de cycles après chaque itération de l'optimisation et, d'autre part, de réduire la taille du voisinage pour l'algortihme de recherche tabou (cf. §7.3.3.2), donc de réduire la complexité de l'optimisation.

Algorithme 7.1: *Optimisation topologique par transformations successives d'une base de cycles indépendants.*

7.3.3.1 Procédure de choix des couples de cycles

La procédure de choix des couples de cycles ayant les plus grandes chances de faire diminuer la fonction coût est donnée dans l'Algorithme 7.2. Cet algorithme est exécuté après chaque itération de l'optimisation, c'est-à-dire après chaque transformation des cycles, mais sans refaire complètement les étapes 1 et 2 (grâce à une procédure économique de mise à jour de la matrice M_d).

7.3.3.2 Heuristique de la recherche tabou

La *recherche tabou* (*tabu search*) est une procédure sous-optimale proposée en 1977 par F. Glover comme une méthode d'optimisation pour les problèmes non linéaires. Elle a été depuis appliquée à des domaines tels que la programmation en nombres entiers, l'affectation quadratique, l'optimisation de circuits électroniques, les problèmes de placement, etc. L'heuristique est adaptée aux problèmes combinatoires, surtout NP-complets, comportant un grand nombre de minima locaux, ce qui est notre cas.

L'heuristique est basée sur l'algorithme du meilleur voisin. Il est donc nécessaire de définir, autour de chaque point de l'espace de solution, un voisinage de points atteignables. Dans notre cas, ce voisinage est défini par un sous-ensemble de transformations potentielles d'une base de cycles ($2(n-1)$ parmi $n(n-1)$ potentielles, cf. l'étape 3 de l'Algorithme 7.1).

L'efficacité de la méthode provient de deux mécanismes spécifiques[13] (cf. *Bland et Dawson* [BD91] :

- La *liste tabou* : une mémoire à court terme des m derniers points visités, utilisée pour éviter de retourner aux mêmes points et pour continuer à avancer dans l'espace de solutions. Cette liste tabou peut être comparée à un serpent se déplaçant sur une surface en cherchant son

13. Il s'agit des deux mécanismes fondamentaux, utilisés pour l'implémentation de la recherche tabou dans REALISME. Dans la littérature, divers autres mécanismes pour améliorer l'heuristique ont été proposés.

Etape 1 Pour la base de cycles courante, construire la matrice triangulaire du graphe dual M_d (dont l'élément a_{ij} est égal au nombre de sommets communs aux cycles C_i et C_j). Initialiser la liste des cycles non encore traités par la liste de tous les cycles.

Etape 2 Pour chaque cycle non encore traité C_i, rechercher (en se référant à la ligne i de M_d), la liste des cycles partageant le plus de sommets avec le cycle C_i. Notons le nombre de sommets communs avec z_i.

Etape 3 Si cette liste contient un seul cycle C_j, enregistrer le couple (C_i, C_j) (s'il n'a pas été enregistré antérieurement). Enlever C_i de la liste des cycles non encore traités.

Etape 4 Si la liste contient plusieurs cycles, appliquer récursivement l'étape 2 pour chacun de ces cycles. Si un cycle C_k parmi eux a le nombre z_k le plus petit par rapport au nombre z des autres cycles de la liste, créer le couple (C_i, C_k) et enlever le cycle C_k de la liste des cycles non encore traités. Si m cycles ($m > 1$) ont le même nombre z plus petit que tous les autres, créer m couples d'une manière aléatoire, sans répétition, parmi les m cycles plus le cycle C_i.[a]

Etape 5 Arrêt si la liste des cycles non encore traités est vide.

[a] Sans entrer dans les détails, le but de l'étape 4 est de permettre d'effectuer les transformations de cycles dans un ordre approprié. Par exemple, pour la Figure 7.3, la transformation $C_3' = diff(C_3, C_2)$ doit être effectuée avant la transformation $C_2' = diff(C_2, C_1)$. Voir *Vasiliu* [Vas92, p. 61–62].

Algorithme 7.2: *Détermination des n couples de cycles les plus intéressants.*

minimum d'altitude sans pouvoir repasser sur sa queue. Si le serpent a une longueur suffisante, il échappe au minima locaux (cuvettes).

- L'*aspiration*, qui inhibe temporairement l'action de la liste tabou et permet ainsi de revenir à des points déjà visités quand cela s'avère utile pour continuer la recherche dans une nouvelle direction. A chaque valeur (discrète) de la fonction coût c est associée une *valeur d'aspiration* qui est le plus petit coût des solutions auxquelles on est arrivé à partir de toutes les solutions de coût c. Cette valeur d'aspiration est remise à jour à chaque itération et permet d'évaluer l'« intérêt» d'une solution. Si le coût de la solution courante est inférieur à la valeur d'aspiration, alors l'éventuel statut tabou de cette solution est ignoré.

Grâce à ces deux mécanismes, la recherche tabou est un compromis entre optimalité et efficacité, notamment avec une bonne capacité à échapper aux minima locaux.

7.3.4 Algorithmes d'affichage

La solution topologique consiste en une base de cycles du graphe. La procédure d'affichage du graphe détermine ensuite une affectation de coordonnées correspondant à la solution topologique. Le gros avantage est la possibilité d'appliquer pour une même solution topologique différentes méthodes d'affichage, qui sont autant de styles ou de standards adaptés à un domaine ou au choix de l'utilisateur.

7.3.4.1 Procédure d'affichage standard

La procédure d'affichage proposée par défaut dans REALISME s'inspire d'un algorithme dédié à la génération de graphes de mécanismes à partir de la matrice du graphe dual (*Sohn et Freudenstein*

[SF86])[14]. Les informations de base sont le nombre et l'identité des arêtes partagées par chaque paire de cycles du graphe. Les étapes principales sont présentées dans l'algorithme 7.3.

Etape 1 Détermination du cycle ayant le plus grand nombre d'arêtes, qui va constituer le cycle périphérique.

Etape 2 Placement des sommets et des arêtes du cycle périphérique (sous la forme d'un polygone régulier).

Etape 3 Détermination, pour les autres cycles, du nombre de sommets ayant déjà été placés lors de l'étape 2.

Etape 4 Dans l'ordre décroissant de ce nombre, placement sur une ligne droite, à distances égales, des sommets nécessaires pour compléter les cycles intérieurs dont les sommets déjà placés forment une chaîne contiguë.

Etape 5 Si tous les cycles ont été représentés, fin de l'algorithme. Sinon, retour à l'étape 3.

Algorithme 7.3: *Algorithme d'affichage par défaut.*

7.3.4.2 Procédures de régularisation

Par la suite, de manière à homogénéiser la densité des sommets et à obtenir une présentation plus plaisante, deux procédures de régularisation peuvent être appliquées. Ces procédures, s'inspirant des algorithmes utilisés pour la régularisation des maillages, recalculent itérativement la position d'un sommet par une moyenne de la position de ses voisins. Il y a une convergence rapide vers un graphe stable. Seul un sous-ensemble de sommets, dont par défaut ceux du cycle périphérique, restent immobiles[15].

Le grand intérêt des procédures de régularisation est qu'elles mettent en évidence les symétries des graphes (lorsqu'elles existent), ce qui est très important dans certains domaines.

La première procédure de régularisation a comme objectif de faire converger chaque sommet vers le centre de gravité de ses voisins (sommets adjacents), c'est-à-dire minimise la somme des carrés des distances jusqu'aux voisins. Pour cela, un algorithme de *surrelaxation* est utilisé :

$$x_i \leftarrow a \cdot \frac{\sum_{k=1}^{n_i} x_k}{n_i} + (1-a) \cdot x_i \tag{7.12}$$

$$y_i \leftarrow a \cdot \frac{\sum_{k=1}^{n_i} y_k}{n_i} + (1-a) \cdot y_i \tag{7.13}$$

où (x_i, y_i) sont les coordonnées du sommet i et (x_k, y_k) les coordonnées de ses n_i voisins. La valeur du coefficient $1 \leq a \leq 2$, empiriquement établie à $1,24$, assure une bonne convergence.

L'objectif de la deuxième procédure est de minimiser la somme des puissances quatrième des distances entre chaque sommet et ses voisins :

14. Dans sa forme d'origine, cet algorithme n'était pas destiné à la représentation des graphes, mais à leur énumération.

15. L'utilisateur peut en spécifier d'autres, en les désignant à la souris.

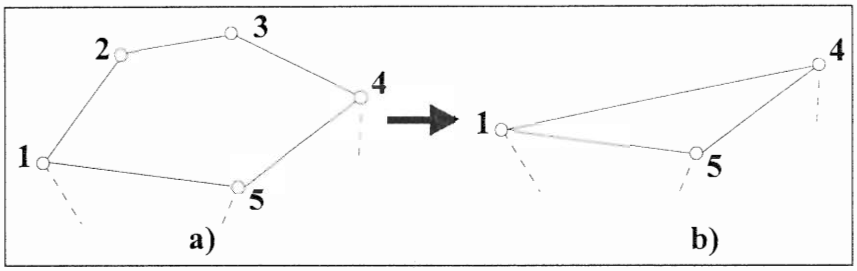


FIG. 7.6 – *Filtrage des sommets préalable à l'optimisation.*

$$x_i \leftarrow x_i + \frac{\sum_{k=1}^{n_i}(x_k - x_i)\left((x_i - x_k)^2 + (y_i - y_k)^2\right)^2}{4\sum_{k=1}^{n_i}\left((x_i - x_k)^2 + (y_i - y_k)^2\right)^2} \tag{7.14}$$

$$y_i \leftarrow y_i + \frac{\sum_{k=1}^{n_i}(y_k - y_i)\left((x_i - x_k)^2 + (y_i - y_k)^2\right)^2}{4\sum_{k=1}^{n_i}\left((x_i - x_k)^2 + (y_i - y_k)^2\right)^2} \tag{7.15}$$

Le calcul des coordonnées est effectué itérativement pour tous les sommets (sauf ceux immobiles) jusqu'à la convergence de l'algorithme (quand le plus grand déplacement de sommet est inférieur à un certain seuil). La convergence est obtenue en général entre 5 et 20 itérations. La différence entre les deux procédures de régularisation concerne le compromis que réalise chacune entre la vitesse de convergence et le degré d'uniformisation de la densité des sommets ; la première procédure est environ deux fois plus rapide, tandis que la deuxième donne des résultats plus esthétiques (cf. Figure 7.8).

En plus de l'utilisation de ces procédures, l'interface graphique de REALISME permet d'apporter à la souris des améliorations par déplacements de sommets, modifications d'échelle, etc.

7.3.5 Prise en compte des particularités des graphes

Pour de gros graphes, on réalise une opération préalable de filtrage du graphe qui le simplifie et augmente la rapidité du dépliage sans affecter en rien le résultat. Ce filtrage consiste en l'élimination des sommets de degré[16] $d \leq 2$ (ou, ce qui revient au même, l'élimination des *arêtes-série* – cf. la note 1, p. 100). Par exemple, pour le graphe de la Figure 7.6, les sommets 2 et 3 sont remplacés par l'arête directe entre les sommets 1 et 4. Ce filtrage ne modifie pas la structure du graphe du point de vue des cycles appartenant à une base fondamentale, mais réduit le nombre de sommets contenus dans les cycles et donc le temps de calcul pour les transformations de cycles. Après l'optimisation, ces sommets seront réintégrés dans le graphe lors de l'affichage.

De même, par souci de généralité, l'algorithme de dépliage permet, dans une certaine mesure, de représenter des graphes non totalement connexes, c'est-à-dire avec des arêtes pendantes ou des sommets isolés. Ce type de parties de graphe est, en réalité, complètement ignoré durant l'optimisation, mais elles sont réintégrées dans le graphe lors de l'affichage, selon des heuristiques de placement adaptées.

16. Le *degré* d'un sommet est le nombre d'arêtes incidentes au sommet.

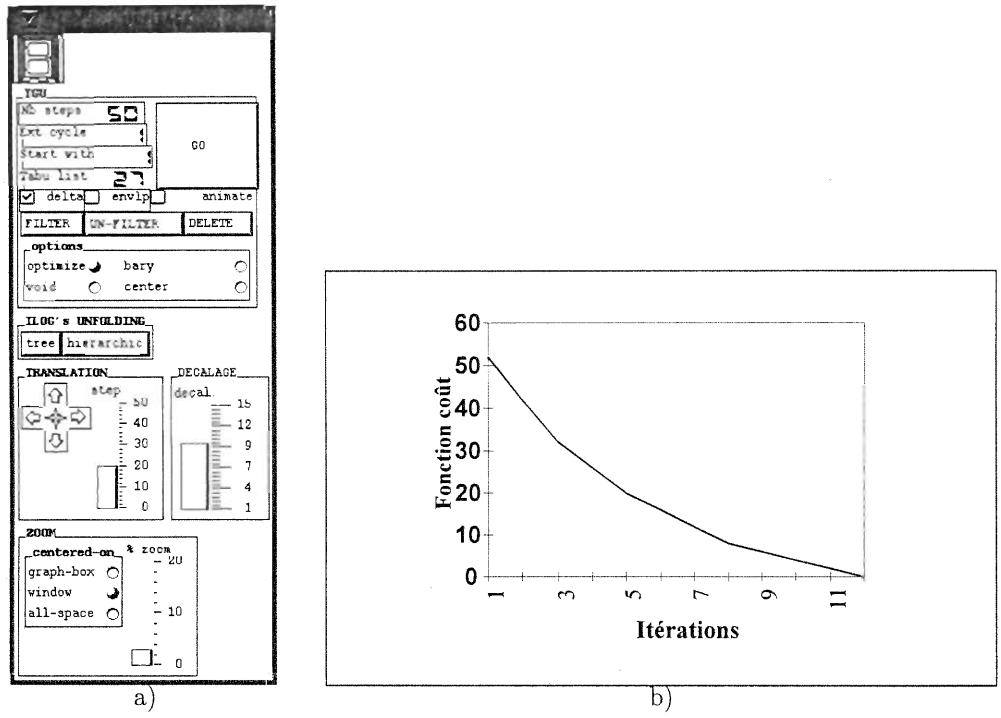


FIG. 7.7 – *a) : Panneau de* REALISME *permettant le contrôle des paramètres de l'algorithme d'optimisation et des procédures d'affichage. b) : Courbe de convergence pour le dépliage du graphe de la Figure 7.8.*

7.4 Exemples

Un exemple de résultat d'utilisation dans REALISME de l'algorithme de dépliage a déjà été donné dans la Figure 4.4, p. 60. En utilisant le bouton approprié dans l'éditeur de schémas cinématiques (Figure 6.1, p. 88), le graphe associé du mécanisme est automatiquement affiché dans sa forme dépliée. L'optimisation topologique et la procédure d'affichage par défaut sont exécutées en moins d'une seconde.

L'utilisation d'une procédure de régularisation peut améliorer la lisibilité du graphe déplié. Pour cela, l'utilisateur a à disposition le panneau de la Figure 7.7.a, qui permet également d'accéder aux paramètres de contrôle des algorithmes d'optimisation et d'affichage (longueur de la liste tabou, nombre maximal d'itérations, choix du cycle périphérique, choix de la procédure d'affichage, translations et changements globaux d'échelle etc.). Un autre exemple de graphe (voir Figure 7.8) permet de comparer les résultats de la procédure d'affichage par défaut et ceux des procédures de régularisation. Pour cet exemple, la courbe de convergence de l'optimisation topologique, représentant l'évolution de la fonction coût (telle qu'elle a été définie au §7.3.1) est présentée dans la Figure 7.7.b.

7.5 Conclusion et débouchés

Une comparaison des principales caractéristiques de la méthode de dépliage de REALISME par rapport aux autres méthodes de la littérature est présentée dans le Tableau 7.1.

Les principales caractéristiques de cette méthode sont la rapidité et la possibilité d'avoir plusieurs présentations différentes quasi instantanées du même graphe, grâce au découplage entre une

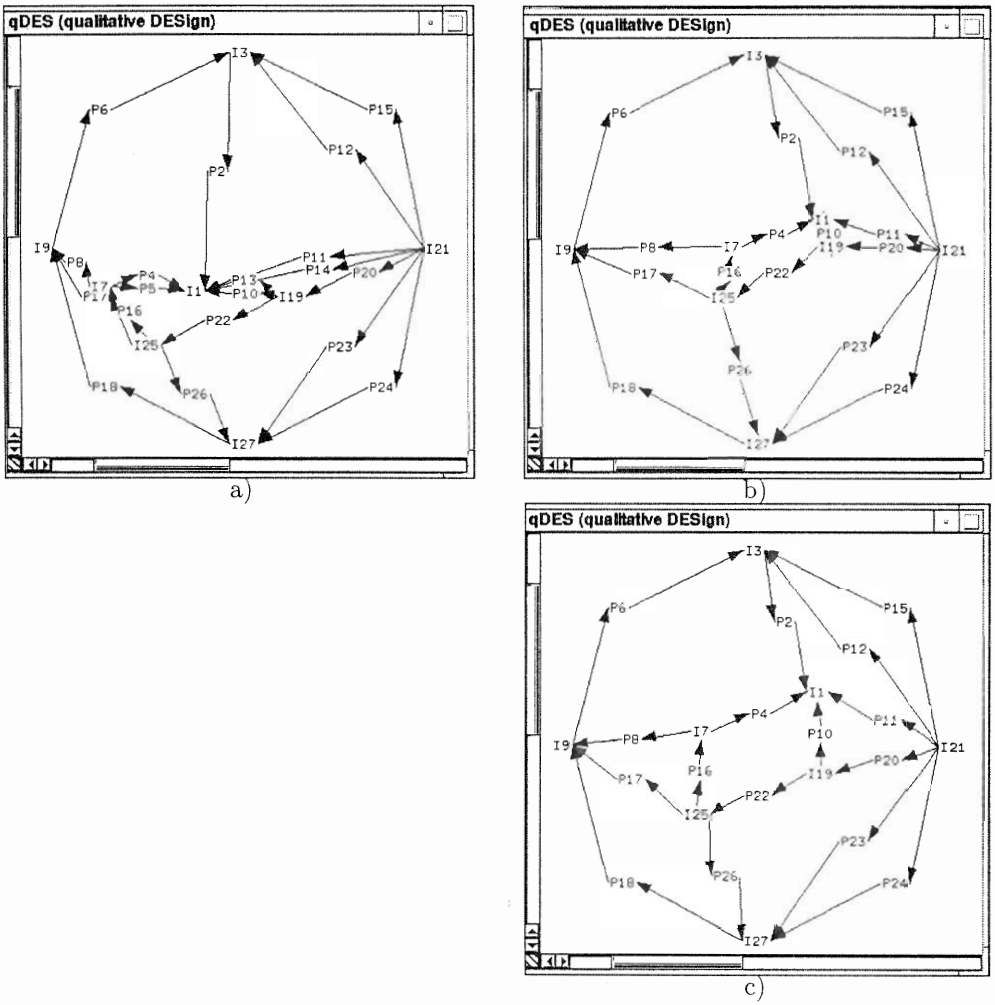


FIG. 7.8 – *Trois représentations différentes d'un même graphe : procédure d'affichage par défaut (a), première procédure de régularisation (b) et deuxième procédure de régularisation (c). Les temps de calcul sont de l'ordre de la seconde.*

	Optimisation numérique	**Approche topologique [CH90]**	**Approche topologique RealisMe**
Solution initiale	Ensemble quelconque de coordonnées	Base fondamentale de cycles	Base fondamentale de cycles
Espace de solutions	*Numérique* : coordonées (discrétisées)	*Topologique* : configurations des cycles	*Topologique* : les bases de cycles
Parcours de l'espace	Permutations de sommets	Enumération de configurations	Soustractions topologiques de cycles adj.
Fonction coût	Exacte : nombre d'entrecroisements	Binaire : configuration valide ou non	Nombre d'arêtes communes aux cycles
Graphes non planaires	Oui	Non	**Oui**
Rapidité	Mauvaise	Aléatoire	**Acceptable**

TAB. 7.1 – *Comparaison entre l'approche numérique, l'approche topologique de Chieng et Hoeltzel et l'approche topologique de* REALISME

optimisation dans l'espace topologique et l'affichage. De plus, la régularisation met en évidence les symétries des graphes.

Nous avons baptisé *TGU* (*Topological Graph Unfolding*) la méthode originale de dépliage, implémentée comme un module de REALISME. Bien que développé pour les graphes de mécanismes, ce module peut évidemment servir dans d'autres domaines. Ainsi, le module *TGU* a été utilisé pour la représentation de graphes d'envisionnement, dans la plateforme *QDES* de conception basée sur la physique qualitative (voir *Yannou* [Yan94] ; cf. Figure 7.8). Il a été également intégré dans le système *ELACT*, dans le domaine de la conception des turbopompes de la fusée Ariane (voir *Cantzler* [Can96]). Finalement, le module *TGU* a été acquis par la société *Ilog* (pour son logiciel de gestion de graphes *Grapheur* – dont la procédure de dépliage de graphes connexes était très déficiente).

Chapitre 8

Synthèse dimensionnelle : l'approche de l'optimisation

8.1 Introduction

Un passage en revue des techniques d'optimisation dimensionnelle des générateurs de trajectoire a été fait au §2.4.3 ; l'analyse de la problématique et des choix méthodologiques adoptés dans ce domaine par cette thèse a été présentée au §3.7.1 et §3.7.2. Le présent chapitre contient le détail des approches d'optimisation dimensionnelle mises en œuvre dans REALISME : l'optimisation dimensionnelle exacte (§8.2) et l'optimisation dimensionnelle qualitative (§8.3).

Par *optimisation dimensionnelle exacte*, nous entendons une optimisation réalisée par bouclage sur des analyses cinématiques traditionnelles, c'est-à-dire qui déterminent pour chaque jeu de dimensions une *trajectoire exacte*, formée par une succession de points.

Par contre, *l'optimisation qualitative* emploie la *simulation qualitative* (cf. chap. 5), dans laquelle les trajectoires générées sont des *trajectoires qualitatives* (arcs de cercles et segments), approximant les trajectoires exactes. Pour les deux approches, seules les trajectoires désirées *fermées* et les mécanismes à un degré de liberté sont considérés. Deux cas sont possibles : la génération de trajectoire avec *indépendance du temps* et avec *dépendance du temps* (voir les définitions à la page 196).

Le même algorithme d'optimisation est utilisé pour l'optimisation exacte et qualitative. Cet algorithme est adapté pour la génération de trajectoire avec dépendance du temps ; nous préciserons néanmoins les adaptations à apporter pour aborder l'indépendance du temps, beaucoup plus complexe en termes d'optimisation.

8.2 Approche d'optimisation dimensionnelle exacte

8.2.1 Fonction objectif

Le but de l'optimisation est la détermination d'un mécanisme remplissant convenablement la fonction cinétique désirée, qui est dans notre cas une *trajectoire désirée exacte* (définie par des points). La procédure de REALISME qui permet au concepteur de spécifier la trajectoire désirée exacte a été présentée en §6.3.

La fonction objectif de l'optimisation doit, principalement, estimer la « la qualité » de chaque mécanisme par rapport au cahier des charges. En notre cas, il s'agit d'une mesure de la distance entre les trajectoires désirée et générée. Cette distance est souvent appelée *erreur structurale*,

puisqu'elle ne dépend que de la structure – morphologie et dimensions – et pas des erreurs de fabrication, jeux dans les liaisons, déformations des pièces etc. L'erreur structurale est donc une composante de l'erreur réelle pendant le fonctionnement du mécanisme. Néanmoins, la prise en considération de la seule erreur structurale est en générale une approximation satisfaisante en phase de préconception.

Lorsque la trajectoire générée est une trajectoire fermée et que la trajectoire spécifiée dépend du temps, la partie de la fonction objectif liée à l'erreur structurale, F_s est calculée dans REALISME, de la manière usuelle dans la littérature, comme la somme des carrés des distances entre les points sur la trajectoire générée (x_i, y_i) et les points sur la trajectoire désirée (x_{d_i}, y_{d_i}), ces points correspondant aux mêmes instants :

$$F_s = \sum_{i=1}^{n} \left[(x_i - x_{d_i})^2 + (y_i - y_{d_i})^2\right] \tag{8.1}$$

où n est le nombre de points composant les deux trajectoires. L'égalité entre le nombre de points formant les deux trajectoires est obtenue par le choix approprié de l'incrément de position du moteur entre deux pas de simulation. Si les points spécifiés sont équirépartis temporellement, les simulations cinématiques seront faites pour des positions du moteur à des intervalles constants de temps. Sinon, ces positions devront être calculées en fonction de la répartition temporelle des points spécifiés (par exemple, pour un moteur avec la loi $s(t)$, les simulations seront faites pour les positions $s(t_i)$, où $t_1, \ldots, t_n$ sont les instants de temps correspondant aux n points spécifiés.

Mais le problème le plus délicat pour l'optimisation dimensionnelle des générateurs de trajectoire est le traitement des cas où la valeur du vecteur de conception ne permet pas une simulation complète (pour un tour complet de la manivelle, par exemple). La trajectoire générée est alors une trajectoire ouverte[1], composée d'un nombre de points inférieur à n. Comment calculer, en ce cas, la fonction distance, tout en conservant une continuité de F_s par rapport aux variables d'optimisation ? Nous présentons d'abord une méthode de la littérature, ensuite la méthode utilisée dans REALISME.

8.2.1.1 Méthode de Cossalter

Utilisant l'approche modulaire par groupes d'Assur, *Cossalter et al.* [CDPS92] définissent un terme de pénalité inclus dans la fonction objectif, mesurant de combien les conditions géométriques d'assemblage au niveau d'un groupe d'Assur sont violées. (Une méthode semblable est utilisée dans *Pakes et al.* [PPW79].) Par exemple, pour la diade RRR (formée des barres AB et BC), les conditions d'assemblage (voir aussi p. 65) sont :

$$\begin{cases} AC \leq (AB + BC) \\ AC \geq |AB - BC| \end{cases} \tag{8.2}$$

Cossalter définit des résidus r (mesurant de combien une des contraintes est violée) :

$$\begin{cases} r = AC - (AB + BC) \\ r = |AB - BC| - AC \end{cases} \tag{8.3}$$

1. Avec certains choix du paramétrage des dimensions, il est possible que pour certaines valeurs du vecteur de conception l'assemblage du mécanisme dans la position initiale soit impossible, donc qu'il n'y ait aucune trajectoire générée. Cependant, avec le paramétrage utilisé en REALISME (voir §8.2.2) l'assemblage dans la position initiale est toujours possible. La trajectoire générée contient donc au moins un point.

Lorsque pendant la simulation aucune violation des conditions d'assemblage n'apparaît, la fonction objectif est l'erreur structurale habituelle (équation 8.1)[2]. Sinon, la fonction objectif devient :

$$F = F_a + F^*, \tag{8.4}$$

$$F_a = w_a \sum_{i=1}^{m} r_i^2 \tag{8.5}$$

où m est le nombre de positions du moteur qui ne permettent pas l'assemblage, w_a est un coefficient et F^* est la valeur de l'erreur pour la portion de trajectoire faisable. Les résidus sont calculés pour le premier groupe d'Assur pour lequel les conditions d'assemblage sont violées. F_a exprime une intégrale quadratique de « combien » la contrainte est violée sur la durée de ce viol.

Cette méthode de Cossalter n'est pas, à notre avis, la meilleure possible, puisque la valeur des résidus r_i dépendra, en partie, du hasard. En effet, pour la même valeur du vecteur de conception, le nombre de positions du moteur pour lesquelles la contrainte d'assemblage est violée, ainsi que les valeurs du résidu, dépendent de la discrétisation choisie pour les positions du moteur. La pénalisation par les résidus (équation 8.5) n'est donc pas une estimation fidèle de la « qualité » du vecteur de conception.

8.2.1.2 Méthode utilisée dans RealisMe

La fonction objectif utilisée dans REALISME comprend la somme pondérée entre l'erreur structurale (équation 8.1) et le carré de la différence des longueurs des trajectoires désirée et générée :

$$F = \frac{v}{n} F_s + (L - L_d)^2 \tag{8.6}$$

où L est la longueur de la trajectoire générée, L_d celle de la trajectoire désirée et v est un coefficient. La division par n de F_s rend la valeur de F indépendante du nombre de points des trajectoires (car F_s contient n fois des carrés de distances) ; l'élévation au carré de $(L - L_d)$ vise l'homogénéisation dimensionnelle (car F_s est un carré de distance) ; une valeur de $v = 100$ s'est avérée convenable (cela signifie qu'on accorde 10 fois moins d'importance à la différence de longueur des trajectoires qu'à la distance moyenne entre les trajectoires).

La prise en considération de la différence de longueur a comme but de compenser la moindre distance entre les trajectoires dans le cas des trajectoires générées ouvertes, due au moindre nombre de points. La fonction F ainsi formulée est une estimation plus continue de l'écart entre la solution désirée et la solution courante.

L'erreur structurale F_s doit être calculée pour l'ensemble des points sur la trajectoire générée, qu'elle soit ouverte ou fermée. Pour cela, lorsqu'une simulation complète n'est pas possible, une deuxième simulation est lancée à partir de la position initiale, avec le sens inverse de rotation de la manivelle (pour les moteurs de rotation). (Voir Figure 8.1.) De cette façon, la totalité de la trajectoire ouverte est déterminée et l'erreur structurale F_s est calculée par la formule :

$$F_s = \sum_{i=1}^{n_1} d_i^2 + \sum_{i=n_1+1}^{n_2} d_i^2 \tag{8.7}$$

$$\text{où } d_i = \sqrt{(x_i - x_{d_i})^2 + (y_i - y_{d_i})^2}$$

Le deuxième terme disparaît lorsque la simulation pour la position $n_1 + 1$ n'est pas possible.

2. L'article de Cossalter traite des générateurs de mouvement. Un terme supplémentaire est donc introduit dans l'équation 8.1 pour tenir compte de l'erreur sur l'angle de l'organe effecteur.

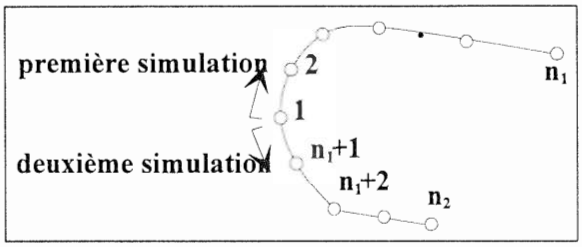


FIG. 8.1 – *Détermination d'une trajectorie générée ouverte lorsqu'une simulation complète n'est pas possible. Une première simulation détermine les points de* 1 *à* n_1 *et une deuxième simulation, en sens invers, détermine les points de* n *à* n_2.

Les longueurs des trajectoires sont calculées comme longueur de la ligne brisée définie par les n_2 points :

$$L = \sum_{i=1}^{n_1-1} d_{i,i+1} + \sum_{i=n_1+1}^{n_2-1} d_{i,i+1} + d_{n_1,n_2} \tag{8.8}$$

$$\text{où } d_{k,p} = \sqrt{(x_k - x_p)^2 + (y_k - y_p)^2}$$

8.2.2 Variables d'optimisation

Une étape importante d'une méthode d'optimisation est l'établissement de la liste des variables d'optimisation (composant le vecteur de conception). Puisque l'optimisation porte seulement sur les dimensions, le choix de la morphologie appartient au concepteur. Ce choix est effectué soit par la construction du mécanisme dans l'éditeur de schémas cinématiques (voir §6.2), soit en chargeant un mécanisme déjà construit et, le cas échéant, en le modifiant. Ensuite, le concepteur doit désigner un point effecteur (pour les détails sur les manipulations dans REALISME, voir §8.2.6).

8.2.2.1 Avec dépendance du temps

En accord avec le choix du paramétrage cartésien absolu pour la méthode de simulation cinématique de REALISME (voir un exemple au §4.2.1.1), les mêmes paramètres dimensionnels sont utilisés comme variables d'optimisation. Par défaut, les coordonnées des n_p liaisons pivots et les orientations des n_g liaisons glissières correspondant à la position initiale du mécanisme constituent le vecteur de conception $\mathbf{X}^t$:

$$\mathbf{X}^t = (x_1, \ldots, x_{n_p}, y_1, \ldots, y_{n_p}, \alpha_1, \ldots, \alpha_{n_g}) \tag{8.9}$$

Le concepteur a néanmoins la possibilité de spécifier un sous-ensemble de ces variables : il est évident que la complexité de l'optimisation est moindre quand il y a moins de variables. De plus, dans certains cas, une partie des paramètres dimensionnels peuvent être imposés pour différentes raisons fonctionnelles (réutilisation partielle d'un mécanisme existant etc.).

Les coordonnées du point effecteur peuvent ne pas être des variables de conception, en étant astreintes à coïncider avec une position particulière sur la trajectoire désirée. Cependant, ceci réduit dans une certaine mesure l'optimalité de la trajectoire synthétisée, puisqu'il n'est pas justifié de privilégier une position particulière du point effecteur. L'élimination de ces variables sera donc faite si on accepte une moindre optimalité comme prix à payer pour une moindre complexité de l'optimisation.

Par rapport aux choix d'un paramétrage relatif, par exemple incluant les longueurs de barres, le paramétrage cartésien absolu a deux avantages :

- le mécanisme est toujours assemblable dans la position initiale, ce qui permet de calculer une trajectoire (ouverte) même si une simulation complète est impossible (voir §8.2.1) ; de cette manière, il n'existe plus de valeur infaisable du vecteur de conception ;
- il n'est plus nécessaire d'ajouter des contraintes supplémentaires pour que les longueurs des barres restent positives.

Puisque les variables d'optimisation sont exactement celles (ou une partie de celles) qui apparaissent dans les expressions formelles des fonctions de transfert au sein des groupes multipolaires (voir §4.2 et §4.4), à chaque itération de l'optimisation la mise à jour du mécanisme - pour une nouvelle valeur du vecteur de conception - est immédiate. Le caractère paramétrique de la simulation cinématique s'étend, ainsi, à l'optimisation dimensionnelle.

8.2.2.2 Avec indépendance du temps

Dans le cas de la synthèse avec indépendance du temps, le point effecteur parcourt la trajectoire sans aucune condition de temporalité. Les positions angulaires de la manivelle θ_i correspondant à chaque point de la trajectoire ne sont donc plus connues. Par conséquent, le vecteur de conception 8.9 est augmenté de n variables θ_i doivent être intégrées un nombre de variables supplémentaires égale au nombre n de points de la trajectoire désirée[3]. Ce sont les angles θ_i de la manivelle correspondant à chaque point i sur la trajectoire. Le vecteur de conception $\mathbf{X}^t$ est alors :

$$\mathbf{X}^t = (x_1, \ldots, x_{n_p}, y_1, \ldots, y_{n_p}, \alpha_1, \ldots, \alpha_{n_g}, \theta_1, \ldots, \theta_n) \tag{8.10}$$

Le nombre de variables de conception augmente donc fortement. De plus, les variables $\theta_1, \ldots, \theta_n$ sont assujetties à des contraintes de précédence, difficiles à prendre en compte :

$$\theta_1 < \theta_2 < \ldots < \theta_n \tag{8.11}$$

Dans ces conditions, le problème d'optimisation devient extrêmement complexe et il n'a pas été abordé dans REALISME. En revanche, d'autres approches – plus à même d'aboutir à une solution – ont été proposées (voir chap. 10 et 9).

Il existe également une autre manière d'éviter l'introduction de variables supplémentaires de l'équation 8.10. Ainsi, la distance entre les deux trajectoires peut être calculée en faisant abstraction de la répartition particulière des points. Par exemple, J. M. Hansen [Han93] utilise l'interpolation par des splines de la trajectoire désirée et, après chaque simulation, de la trajectoire générée. La somme des carrés des distances entre des points équirépartis sur les deux splines est employée comme fonction distance. L'approximation des trajectoires par les premières harmoniques, calculées par une analyse de Fourier et normalisées [MM93] [McG94], peut être également une solution pour s'abstraire de la répartition des points (d'ailleurs employée dans cette thèse dans un autre contexte, voir chap. 9). Malheureusement, ces calculs – qui doivent être faits, au sein de l'optimisation, un grand nombre de fois – sont coûteux. Par conséquent, ces approches ont le désavantage d'être très lentes.

3. En principe, il est possible d'éviter l'introduction de ces variables supplémentaires. Ainsi, la distance entre les deux trajectoires peut être calculée en faisant abstraction de la répartition particulière des points. Ainsi, par exemple J. M. Hansen [Han93] utilise l'interpolation des trajectoires par des splines et calcule la somme des distances entre des points équirépartis sur les splines. L'approximation des trajectoires par les premières harmoniques, calculées par une analyse de Fourier, est également une solution (d'ailleurs employée dans cette thèse dans un autre contexte, voir chap. 9). Malheureusement, ces procédures sont en général trop coûteuses pour permettre l'utilisation au sein de l'optimisation.

8.2.3 Contraintes d'environnement

Dans certains cas, le cahier des charges impose, en plus d'une trajectoire désirée, des contraintes d'encombrement du mécanisme. La plus simple manière est de prescrire une *boîte englobante*[4], sous la forme d'un rectangle avec les côtés parallèles aux axes et dimensionnée par des coordonnées $x_{\min_d}$, $x_{\max_d}$, $y_{\min_d}$ et $y_{\max_d}$. (Au prix d'une plus grande complexité des calculs, il est également possible de considérer une *boîte englobante* séparément pour chaque pièce et des boîtes de formes plus complexes.)

Malheureusement, la *boîte englobante* pour un mécanisme ne peut pas être calculée directement en fonction des variables d'optimisation, qui ne définissent que la position initiale du mécanisme. Pour la calculer, il est donc nécessaire de déterminer, durant la simulation cinématique, les plus grandes et les plus petites valeurs des coordonnées des liaisons (on considère que les liaisons sont les extrémités des pièces). Nous notons $x_{\min} = \min_{i,j} x_{ij}$, $x_{\max}, = \max_{i,j} x_{ij}$, $y_{\min} = \min_{i,j} y_{ij}$ et $y_{\max} = \max_{i,j} y_{ij}$, avec $i = 1, \ldots, n$, n étant le nombre de positions calculées et $j = 1, \ldots, n_l$, n_l étant le nombre de liaisons. Une pénalité $F_{b.e.}$ dépendant du dépassement de la boîte englobante désirée peut être ajoutée à la fonction objectif (équation 8.6) :

$$F_{b.e.} = w_{b.e.}\left(a(x_{\min} - x_{\min_d})^2 + b(x_{\max} - x_{\max_d})^2 + c(y_{\min} - y_{\min_d})^2 + d(y_{\max} - y_{\max_d})^2\right), \text{ avec} \tag{8.12}$$

$$a = \begin{cases} 1 & \text{si} \quad x_{\min} < x_{\min_d} \\ 0 & \text{si} \quad x_{\min} \geq x_{\min_d} \end{cases} \qquad b = \begin{cases} 1 & \text{si} \quad x_{\max} > x_{\max_d} \\ 0 & \text{si} \quad x_{\max} \leq x_{\max_d} \end{cases}$$

$$c = \begin{cases} 1 & \text{si} \quad y_{\min} < y_{\min_d} \\ 0 & \text{si} \quad y_{\min} \geq y_{\min_d} \end{cases} \qquad d = \begin{cases} 1 & \text{si} \quad y_{\max} > y_{\max_d} \\ 0 & \text{si} \quad y_{\max} \leq y_{\max_d} \end{cases} \tag{8.13}$$

La nouvelle fonction objectif F' s'écrit alors :

$$F' = F + w_{b.e.} F_{b.e.} \tag{8.14}$$

où F est la fonction objectif sans contraintes d'encombrement (équation 8.6) et $w_{b.e.}$ est un coefficient.

Suivant un procédé analogue, d'autres contraintes du cahier des charges peuvent, en principe, être intégrées dans la fonction objectif (par exemple longueur minimum et/ou maximum des barres). Cependant, lorsqu'il existe un grand nombre de critères de conception, par exemple des critères d'efforts ou des critères dynamiques, la pondération des différents critères devient difficile et il est souhaitable d'adopter une approche d'optimisation multicritère (voir l'état de l'art, §2.4.3.6).

8.2.4 Algorithme d'optimisation

L'algorithme d'optimisation que nous avons adopté[5] est basé sur celui du gradient conjugué, avec un encadrement du minimum unidimensionnel basé sur le signe du gradient et une recherche du minimum unidimensionnel également par réduction de l'encadrement en regardant le signe du gradient (l'objectif étant d'obtenir un minimum unidimensionnel très précis). Les dérivées partielles sont estimées par différences finies.

4. Pour la prise en compte d'un domaine faisable sous la forme d'un polygone convexe, voir Hansen [Han92, p. 125–138].

5. Je remercie Mounib Mekhilef (*ECP/Université d'Orléans*) pour m'avoir mis à disposition son code d'optimisation.

L'inaccessibilité, dans le cas général, des dérivées partielles[6], la non linéarité du problème, ainsi que les imprécisions numériques des valeurs de la fonction objectif calculées (cf. aussi *Sundar* [SS94]) rendent délicate l'utilisation d'algorithmes rapides (lagrangien, directions conjuguées...). Dans ce contexte, la précision de la méthode de simulation cinématique est très importante. Or, grâce à l'élimination souvent complète des résolutions numériques itératives de systèmes non linéaires, l'approche systémique multipolaire est particulièrement adaptée à une telle méthode d'optimisation[7].

La synthèse des générateurs de trajectoire avec indépendance du temps nécessite la prise en compte, dans le vecteur de conception, d'un certain nombre de variables d'angle soumises à des contraintes de précédence. Ce sont les angles de la manivelle correspondant aux positions du point effecteur. Pour résoudre le problème d'optimisation avec ce type de contraintes, d'autres approches d'optimisation spécifiques doivent être utilisées. Ce type de problème n'a donc pas été traité pour l'instant dans REALISME.

8.2.5 Accélération de la convergence : le recalage géométrique

L'optimisation dimensionnelle des générateurs de trajectoire est caractérisée par une propriété remarquable : la trajectoire générée par un mécanisme est, pour ainsi dire, solidaire du mécanisme, autrement dit toute transformation affine du mécanisme a exactement le même effet sur la trajectoire. Pour donner un exemple simple, si le mécanisme est translaté dans le plan d'un mètre, la trajectoire générée est également translatée d'un mètre.

Cette propriété peut être utilisée pour accélérer la convergence de l'optimisation. A notre connaissance, aucun travail de la littérature n'a suggéré ou exploité cette idée. Après une ou plusieurs itérations de l'optimisation, le mécanisme avec les dimensions D génère la trajectoire T, située à une distance (au sens de l'erreur structurale, équation 8.1) $dist$ de la trajectoire désirée T_d. L'objectif du recalage géométrique est de calculer la transformation géométrique affine du mécanisme qui minimise la distance $dist$. En vertu de la « solidarité » entre mécanismes et trajectoires, cette transformation est calculable en considérant simplement les deux trajectoires T et T_d. Ensuite, l'optimisation reprend à partir d'une position déterminée par les dimensions D transformées, donc d'une position plus proche de l'optimum. L'approche d'optimisation consiste, donc, dans une alternance d'itérations d'optimisation avec une phase de recalage géométrique.

Dans le cas de l'optimisation exacte avec dépendance du temps, les paramètres de cette transformation sont calculés en minimisant la somme des carrés des distances entre les couples de points sur les deux trajectoires. Autrement dit, la suite de points de la trajectoire simulée, après transformation, doit devenir la meilleure approximation, au sens des moindres carrés, de la suite de points de la trajectoire désirée.

Notons x_i, y_i les coordonnées des points sur la trajectoire simulée, x'_i, y'_i les coordonnées des points correspondants sur la trajectoire désirée ($i = 1, \ldots, n$) et a, b, c et d les coefficients de la matrice de transformation. Pour déterminer les valeurs optimales de a, b, c et d, il faut considérer les distances entre chaque couple de points :

$$\underbrace{\begin{pmatrix} x_i & -y_i & 1 & 0 \\ y_i & x_i & 0 & 1 \end{pmatrix}}_{K_i} \underbrace{\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}}_{U} - \underbrace{\begin{pmatrix} x'_i \\ y'_i \end{pmatrix}}_{V_i} \tag{8.15}$$

6. Dans un article récent, *Mariappan et Krishnamurty* [MK96a] présentent une méthode générale pour le calcul des dérivées partielles. Son application permettrait de réduire considérablement le nombre d'évaluations de la fonction et donc d'accélérer l'optimisation.

7. Malheureusement, dans l'implantation informatique concrète de REALISME, une partie de cet avantage est perdue, car le langage *Le-lisp* exécute les calculs en simple précision.

En notant les matrices comme dans l'expression 8.15, le problème de recalage optimal devient un problème de minimisation de la fonctionnelle :

$$\min_U \sum_i \|K_i U - V_i\|^2 \tag{8.16}$$

En développant l'expression 8.16, on obtient :

$$\min_U \left(\sum_i \langle K_i^t K_i U, U \rangle - 2 \langle K_i^t V_i, U \rangle + \langle V_i, V_i \rangle \right) \tag{8.17}$$

En notant $M = \sum_i K_i^t K_i$ et $B = \sum_i K_i^t V_i$, le problème de minimisation revient à la résolution du système linéaire suivant :

$$MU = B \tag{8.18}$$

En développant les termes apparaissant dans 8.17 :

$$K_i^t K_i = \begin{pmatrix} x_i & y_i \\ -y_i & x_i \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_i & -y_i & 1 & 0 \\ y_i & x_i & 0 & 1 \end{pmatrix} = \begin{pmatrix} x_i^2 + y_i^2 & 0 & x_i & y_i \\ 0 & x_i^2 + y_i^2 & -y_i & x_i \\ x_i & -y_i & 1 & 0 \\ y_i & x_i & 0 & 1 \end{pmatrix} \tag{8.19}$$

$$K_i^t V_i = \begin{pmatrix} x_i & y_i \\ -y_i & x_i \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x'_i \\ y'_i \end{pmatrix} = \begin{pmatrix} x_i x'_i + y_i y'_i \\ -y_i x'_i + x_i y'_i \\ x'_i \\ y'_i \end{pmatrix} \tag{8.20}$$

on obtient les expressions suivantes pour M et B :

$$M = \sum_i \begin{pmatrix} x_i^2 + y_i^2 & 0 & x_i & y_i \\ 0 & x_i^2 + y_i^2 & -y_i & x_i \\ x_i & -y_i & 1 & 0 \\ y_i & x_i & 0 & 1 \end{pmatrix} \tag{8.21}$$

$$B = \sum_i \begin{pmatrix} x_i x'_i + y_i y'_i \\ -y_i x'_i + x_i y'_i \\ x'_i \\ y'_i \end{pmatrix} \tag{8.22}$$

Après introduction des expressions 8.21 et 8.22 dans 8.18, il reste à résoudre un système linéaire, dont les solutions seront les scalaires a, b, c et d représentant la transformation optimale.

8.2.6 Exemple

Puisque nous nous limitons à l'optimisation dimensionnelle, le choix de la morphologie appartient au concepteur. Ce choix est effectué soit par la construction du mécanisme dans l'éditeur (voir §6.2), soit en chargeant un mécanisme déjà construit et, le cas échéant, en le modifiant. Ensuite, le concepteur doit désigner un point effecteur (le point E) et les variables de conception, sur lesquelles doit agir l'optimisation, sont indiquées (également par sélection avec la souris). Par défaut, les coordonnées des liaisons pivots et les orientations des liaisons glissières correspondant à la position initiale du mécanisme sont des variables de conception. Pour toutes ces opérations, le concepteur utilise le panneau dédié à l'optimisation (Figure 8.2).

Le concepteur a accès aux paramètres de l'algorithme d'optimisation par l'intermédiaire du panneau de la Figure 8.3.

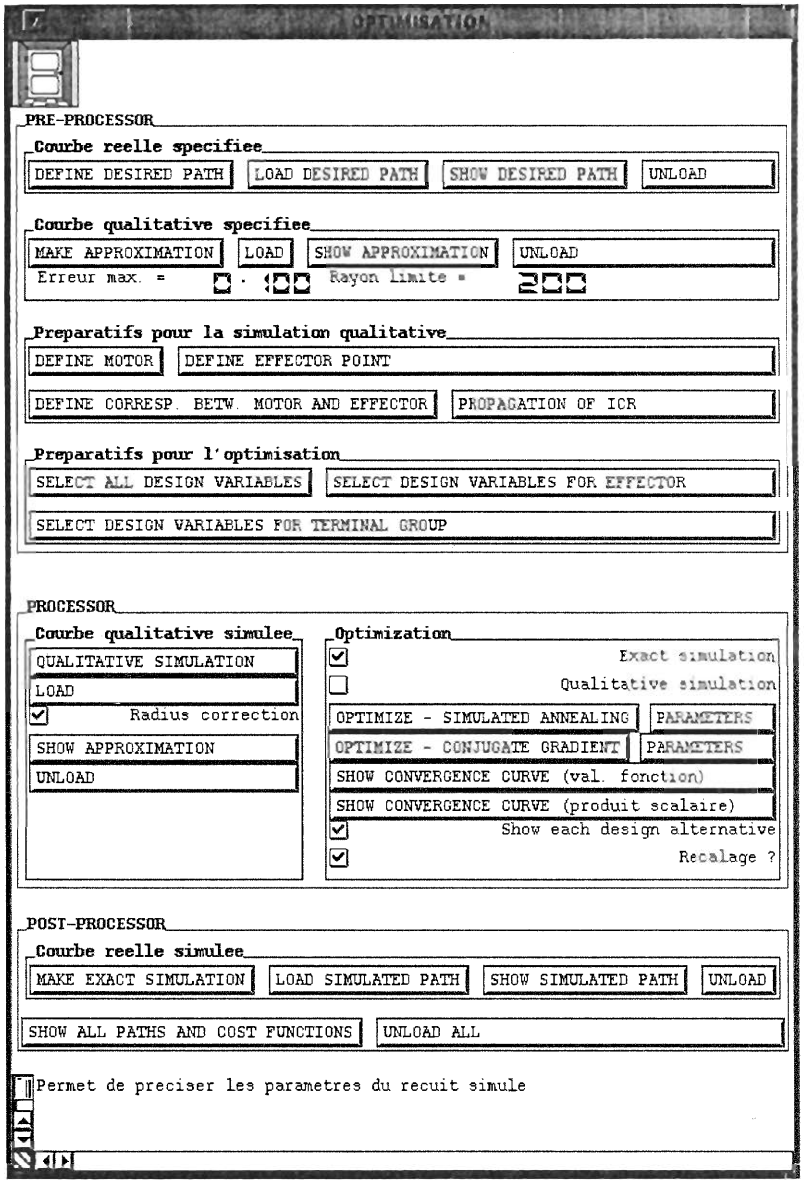


FIG. 8.2 – *Panneau de* REALISME *permettant d'effectuer toutes les opérations nécessaires pour l'optimisation dimensionnelle.*

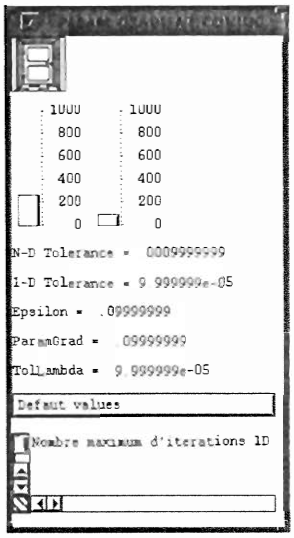


FIG. 8.3 – *Panneau pour la spécification des paramètres de l'algorithme d'optimisation (gradient conjugué).*

8.3 Approche d'optimisation dimensionnelle qualitative

L'*optimisation dimensionnelle qualitative* d'un mécanisme (cf. *Yannou et al.* [YV95] [YVM95]), pour le problème de la génération de trajectoire, consiste à déterminer ses dimensions optimales afin que la trajectoire qualitative[8] qu'il génère par une simulation qualitative[9] soit la plus proche possible de la trajectoire qualitative désirée. La démarche (voir Figure 8.4) consiste à accepter une moindre précision des trajectoires afin de bénéficier de la rapidité supérieure de la simulation qualitative par rapport à une simulation cinématique traditionnelle. De plus, pour la synthèse avec dépendance de temps, le concepteur est à même de spécifier la corrélation entre la vitesse du point effecteur et celle de la manivelle. Le recours à une approximation des trajectoires reste sans influence significative sur la solution trouvée : la seule différence concerne le type de spécification possible est le temps de simulation.

8.3.1 Fonction objectif

Comme pour l'optimisation dimensionnelle exacte, la fonction objectif doit être une évaluation de la distance entre les trajectoires générée et désirée. Cette évaluation est obtenue en comparant directement les arcs et segments qualitatifs deux à deux. Pour cela, trois points sont considérés sur chaque arc ou segment : les deux extrémités, A et C, ainsi que le point milieu B (voir Figure 8.5). La fonction distance F_{dist} est calculée comme la somme pondérée des carrés des distances entre les points correspondants (les points situés sur la trajectoire désirée ont l'indice d ; n_a et n_s sont les nombres d'arcs et de segments) :

$$F_{\text{dist}} = \frac{p_1}{n_a + n_s} \sum_{\text{arcs et segments}} \left(\frac{d^2_{A_d,A}}{2} + d^2_{B_d,B} + \frac{d^2_{C_d,C}}{2} \right) \tag{8.23}$$

$$\text{où } d_{P_1,P_2} = \sqrt{(x_{P_1} - x_{P_2})^2 + (y_{P_1} - y_{P_2})^2} \tag{8.24}$$

8. La trajectoire qualitative est une approximation de la trajectoire désirée exacte, avec une précision donnée, en arcs de cercles et segments. Cf. §5.2.

9. Voir §5.4.

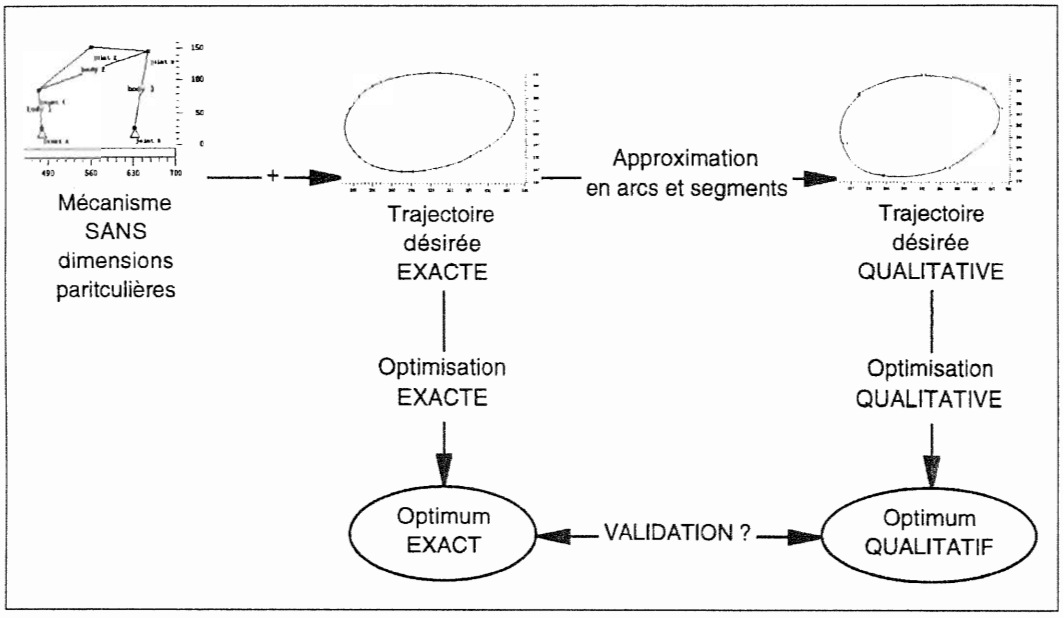


FIG. 8.4 – *Principe de l'optimisation qualitative.*

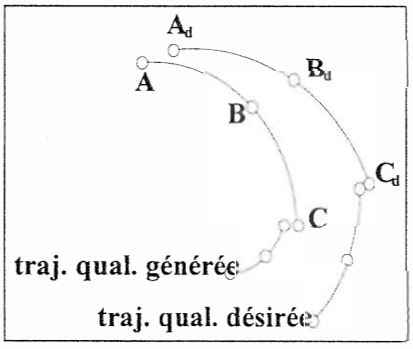


FIG. 8.5 – *Les trois points caractéristiques sur les éléments de trajectoire qualitative : les extrémités A et C, le point milieu B.*

La contribution des points A et B est pondérée par un coefficient $\frac{1}{2}$ car celle-ci est comptée deux fois pour deux arcs (ou segments) consécutifs. p_1 est un autre coefficient de pondération.

La fonction objectif doit pénaliser les mécanismes qui rencontrent des positions de blocage en cours de simulation. Pour ce faire, la même méthode est utilisée que pour l'optimisation exacte, en introduisant une pénalité proportionnelle au carré de la différence des longueurs curvilignes des trajectoires désirée (L_d) et générée (L) :

$$F_{\text{bloc}} = p_2 \left(L_d - L\right)^2 \tag{8.25}$$

Ces longueurs sont calculées par la somme des longueurs de chaque arc et segment :

$$L = \sum_{i=1}^{n_a} \varphi_i r_i + \sum_{i=1}^{n_s} d^2_{A_i,C_i} \tag{8.26}$$

où φ_i et r_i sont l'angle et le rayon des arcs de la trajectoire qualitative et A et C sont les extrémités des segments (pour d, même signification que dans l'équation 8.24).

Lorsqu'en cours de simulation, un blocage est rencontré à cause d'une singularité géométrique, une simulation est relancée en sens inverse jusqu'au prochain blocage, de manière à obtenir une valeur maximale pour L (même méthode que pour l'optimisation exacte, cf. §8.2.1.2).

Il faut noter que par rapport à une optimisation dimensionnelle exacte, le calcul de F_{dist} (éq. 8.23) et de L (éq. 8.26) est moins coûteux, puisque le nombre total d'arcs et de segments $n = n_a + n_s$ est normalement beaucoup plus petit que le nombre de points sur les trajectoires exactes.

Enfin, une pénalité F_{contr} est introduite pour tenir compte d'autres contraintes de conception (comme au §8.2.3). La fonction pénalité finale (globale) est donc :

$$F = F_{\text{dist}} + F_{\text{bloc}} + F_{\text{contr}} \tag{8.27}$$

8.3.2 Variables d'optimisation

8.3.2.1 Avec dépendance du temps

Pour la synthèse avec dépendance du temps, le vecteur de conception est le même que celui utilisé pour l'optimisation dimensionnelle exacte (voir §8.2.2.1).

8.3.2.2 Avec indépendance du temps

Comme pour l'optimisation exacte, la synthèse avec indépendance du temps oblige à augmenter le nombre de variables d'optimisation. Les variables supplémentaires, dans ce cas, sont les angles θ_0, θ_{01},..., θ_{0n} (cf. §5.2) représentant, pour chaque arc et segment qualitatif, l'angle initial de la manivelle (θ_0) et les angles des positions successives relativement à cette position initiale (θ_{0i}). Un avantage certain de l'approche qualitative est que le nombre de variables d'angles (θ_0 et θ_{0i}) est égal au nombre d'arcs (et de segments) qualitatifs. Dans l'optimisation exacte, ce nombre serait égal au nombre de points de la trajectoire désirée exacte.

8.3.3 Algorithmes d'optimisation

L'algorithme d'optimisation utilisé pour l'optimisation dimensionnelle qualitative est le même que celui adopté pour l'optimisation exacte (voir §8.2.4).

Une optimisation basée sur une simulation qualitative s'avère à peu près deux fois plus rapide qu'une optimisation basée sur une simulation classique. Ce résultat respecte le rapport de vitesse des simulations qualitative et classique, car l'efficacité de l'algorithme d'optimisation est relativement indépendante du type de simulation.

8.3.4 Recalage géométrique

De la même manière que pour l'optimisation dimensionnelle exacte (cf. §8.2.5), le recalage géométrique peut être utilisé afin d'accélérer la convergence de l'optimisation qualitative. La procédure spécifique de recalage est identique à celle utilisée au sein de la méthode de synthèse basée sur les familles de trajectoires qualitatives et dont la description détaillée sera donnée au §10.6.4.2.

8.4 Conclusion

L'optimisation en général et l'optimisation dimensionnelle des mécanismes sont des domaines de recherche très vastes. Un grand nombre de techniques d'optimisation ont été utilisées, souvent adaptées à certains cas particuliers. Il est hors du cadre de cette thèse de tenter une comparaison globale des différents modèles et méthodes d'optimisation dimensionnelle des mécanismes proposées dans la littérature. Dans cette thèse, qui vise à constituer une « boîte à outils » pour

la préconception, l'objectif était d'illustrer l'intégration dans une plateforme de conception, au cœur de l'optimisation, des deux modèles de cinématique : l'approche systémique multipolaire et la cinématique qualitative.

La différence entre l'optimisation exacte et qualitative tient essentiellement à la rapidité. En échange d'une moindre précision, l'optimisation qualitative est plus rapide. La différence de vitesse provient de trois sources :

- La simulation qualitative est plus rapide que la simulation traditionnelle.
- Le calcul de la fonction coût est moins coûteux, parce qu'il y a moins d'arcs et de segments que de points sur les trajectoires exactes.
- Pour la synthèse avec indépendance de temps, il faut introduire dans le vecteur de conception moins de variables d'angle supplémentaires.

Chapitre 9

Synthèse par capitalisation de cas : l'approche par réseaux neuronaux artificiels

9.1 Introduction

Les méthodes de synthèse des générateurs de trajectoires basées sur les cas, développées dans REALISME, permettent au concepteur de spécifier une *forme de trajectoire* comme objectif de la conception préliminaire (des exemples industriels correspondant à une telle spécification ont été donnés dans la Figure 1.1, p. 2). Ceci constitue un complément utile par rapport aux méthodes de synthèse analytiques et de synthèse par optimisation, qui partent d'une spécification de trajectoire sous la forme d'une liste de points (trajectoire exacte). Pour les méthodes de synthèse analytiques, le nombre de points de passage imposés (*precision points*) est limité à un faible nombre (typiquement entre 3 et 10), ce qui dans certains cas nuit à la « richesse » de la spécification. Ce nombre de points atteint quelques dizaines pour les méthodes d'optimisation dimensionnelle (cf. chap. 8). Parfois, ce type de spécification correspond au besoin du concepteur, quand on n'a pas d'exigences particulières sur la trajectoire entre ces points. Mais lorsque la vraie fonction désirée est la forme de tout ou partie d'une trajectoire, ces méthodes ne sont pas pilotées par l'expression des fonctions[1]. En fait, à cause de l'outil de conception mal adapté, le concepteur est obligé de « dégrader » la spécification, en traduisant les critères fonctionnels par des critères tels qu'une suite de points (cf. l'état de l'art de la synthèse dimensionnelle au §2.4).

Par rapport à cette situation, les buts des méthodes de synthèse basées sur les cas (présentées dans ce chapitre et dans le chapitre suivant) sont :

- prendre en compte des *spécifications qualitatives* ;
- admettre des spécifications portant sur une trajectoire ouverte ou sur des parties de trajectoire[2] ;
- fournir, le cas échéant, des bonnes solutions initiales pour les méthodes d'optimisation dimensionnelle, afin de pallier ainsi leur sensibilité par rapport aux solutions initiales.

1. Sauf certaines méthodes dédiées à des mécanismes particuliers : générateurs de lignes droites ou d'arcs de cercle, mécanismes *dwell*, etc. (cf. l'état de l'art, §2.4.6).
2. Ce but concerne la méthode de synthèse présentée au chapitre suivant.

Ces méthodes s'inscrivent dans une démarche de type *raisonnement à base de cas* (pour des détails sur cette branche de l'intelligence artificielle, voir l'annexe B). Deux raisons principales justifient l'utilisation de cette approche :

- l'impossibilité ou la difficulté d'appliquer, pour un tel type de cahier des charges, les méthodes de synthèse dimensionnelle connues dans la littérature (graphiques, analytiques, optimisation) ;

- la disponibilité d'un grand nombre de cas représentatifs ; en effet, en un temps raisonnable on peut calculer par simulation cinématique les trajectoires générées par un mécanisme d'une morphologie donnée, pour un grand nombre de jeux de dimensions.

Absence de modèle analytique pour la synthèse dimensionnelle, absence de méthode directe pour la synthèse morphologique et accessibilité d'un grand nombre de *cas*[3] : c'est le contexte par définition qui se prête à l'application d'une approche de type *raisonnement à base de cas*. Le raisonnement à base de cas appliqué à la synthèse de mécanismes permet de partir directement du critère de forme, grâce à l'exploration d'une base rassemblant un grand nombre de cas, codifiés sous une forme appropriée.

Un *cas* de mécanisme (au sens du *raisonnement à base de cas*) a un contenu auquel on peut accéder par un index (cf. §9.3.1). Le contenu représente la structure du mécanisme (dimensions et morphologie) ayant permis de générer une trajectoire qui donnera l'index. Pour des raisons techniques d'implémentation, il faut savoir représenter cet index de trajectoire de manière succincte, voire en un nombre donné de paramètres (dans le cas d'un réseau neuronal artificiel). De la même manière, il faudra réfléchir à la manière de représenter la structure de la manière la plus succincte pour ne pas différencier inutilement des cas identiques dans la base de cas et pour ainsi faire diminuer le plus possible la complexité, la dimension du problème.

Les dimensions du mécanisme peuvent être codifiées simplement, par exemple, par les coordonnées des liaisons dans la position initiale du mécanisme. Par contre, la manière habituelle de décrire les trajectoires comme une suite de points n'est pas adéquate pour représenter la seule forme (avec indépendance du temps) car il n'y a pas de bijectivité entre la liste de points et la trajectoire (une même trajectoire peut être décrite par plusieurs listes de points différentes). Cette représentation conduit, d'une part, à une explosion en termes de taille et, d'autre part, à l'impossibilité de s'abstraire de la répartition particulière des points sur la trajectoire (qui souvent n'est pas un critère fonctionnel et dépend, entre autres, de la discrétisation choisie durant la simulation cinématique). Il serait donc utile de trouver des méthodes plus efficaces pour indexer les trajectoires.

Dans son récent état de l'art sur la conception des mécanismes assistée par ordinateur, *Erdman* [Erd95] considère comme un idéal pour l'avenir l'apparition de logiciels intégrant, entre autres, une « mémoire » ou une « bibliothèque de solutions ». Quelques approches de ce type sont décrites dans la littérature (voir l'état de l'art, §2.4.7, §2.5.3, §2.5.2). Mais il s'agit, souvent, de l'enregistrement des « cas » tels quels, sous forme de catalogue papier ou informatisé, permettant simplement le choix de la solution précédente la plus appropriée. Nous proposons deux solutions inédites, utilisant une codification des cas. Celle exposée dans le présent chapitre utilise une « bibliothèque de solutions » permettant une interpolation, par un réseau neuronal artificiel, entre les solutions enregistrées, codifiées par des coefficients de Fourier (le réseau neuronal jouant le rôle d'un approximeur de fonction). La solution présentée au chapitre suivant emploie une codification des cas par des classes *qualitatives* de trajectoires.

3. Nous entendons ici par *cas* une association entre, d'une part, la morphologie et les dimensions du mécanisme, et d'autre part la trajectoire générée. Cf. §9.3.1.

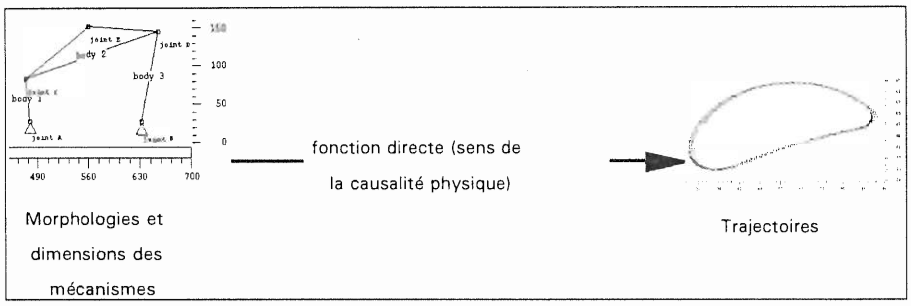


FIG. 9.1 – $T = F(M, D)$ – *Fonction exprimant les formes de trajectoire en fonction des morphologies et des dimensions.*

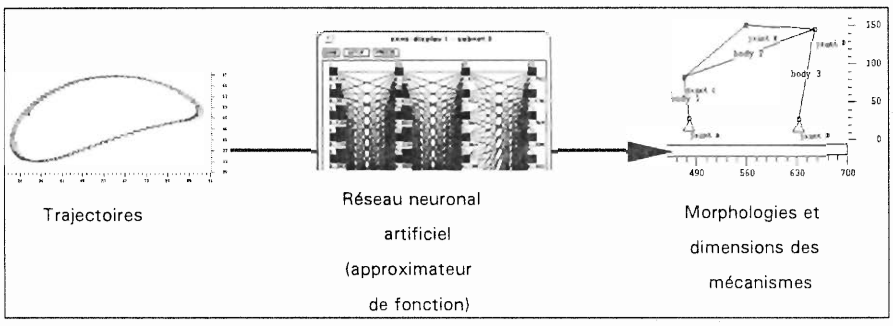


FIG. 9.2 – $(M, D) = F^{-1}(T)$ – *Fonction inverse : la morphologie et les dimensions en fonction de la forme de trajectoire.*

Dans les deux cas, la prise en compte d'une description approximative ou qualitative des trajectoires a comme résultat l'amélioration du rapport entre la taille de la base de cas et la « qualité » des solutions obtenues.

9.2 Une approche par approximation de fonction

A chaque morphologie M et à chaque jeu de dimensions D d'un mécanisme correspond une forme de trajectoire T du point effecteur. Cette trajectoire peut être soit ouverte soit fermée[4]. Il est donc possible de définir une fonction F (voir Figure 9.1), reliant les formes de trajectoire aux morphologies et aux dimensions. La valeur de F est calculable en chaque « point » par une simulation cinématique.

Posé en ces termes, le problème de la synthèse conjointe morphologique et dimensionnelle peut être vu comme une inversion de la fonction F, afin d'utiliser la fonction F^{-1} (voir Figure 9.2) pour déterminer M et D pour une valeur donnée de T. Malheureusement, il y a plusieurs difficultés à cela :

- La fonction F, en toute généralité, n'est pas inversible : la même forme de trajectoire T peut être générée par plusieurs mécanismes de dimensions et de morphologies différentes.
- L'expression explicite de F est soit inconnue, soit trop complexe pour être exploitable[5].

4. Avec un paramétrage des dimensions par les coordonnées cartésienne absolues, le mécanisme est toujours assemblable dans la position initiale. La trajectoire générée contient donc au moins un point. (Cf. la note 1, p. 116.)

5. Pour un simple mécanisme *quatre barres*, la fonction reliant les trajectoires aux dimensions fait intervenir un grand nombre de termes.

- Un mécanisme d'une morphologie donnée ne peut décrire, par son point effecteur, qu'un sous-ensemble de toutes les formes de trajectoire possibles[6], qui sont évidemment en nombre infini. Comme, en pratique, seul un ensemble fini de morphologies est pris en compte, nous n'avons pas la garantie de l'existence d'une solution exacte.

Afin de permettre la définition de cette fonction inverse F^{-1}, il faudra donc s'assurer qu'à une forme de trajectoire correspond un seul jeu de dimensions. Ceci n'est pas possible et n'a pas de sens en règle générale, mais peut être partiellement réalisé par un choix approprié des intervalles de valeurs pour les dimensions (pour une explication plus détaillée, voir §9.5.2). En ce qui concerne le manque de l'expression explicite de la fonction, nous utiliserons une approche de type *raisonnement à base de cas*. Finalement, l'absence – en général – d'une solution exacte nous conduira à traiter le problème général de la synthèse comme un problème d'optimisation : « trouver le mécanisme dont la forme de la trajectoire générée s'approche le plus de celle de la trajectoire désirée ».

La méthode de synthèse de forme de REALISME, utilisant l'approximation de fonction grâce à un réseau neuronal, ainsi que les travaux de J. McGarva [MM93] [McG94] ayant servi de point de départ à notre méthode de codification des trajectoires, sont les seuls, à notre connaissance, à s'attaquer à la synthèse de forme de trajectoire. En effet, les techniques d'optimisation dimensionnelle, décrites au chap. 8, ne permettent que la synthèse d'une trajectoire spécifiée par une suite de points. De plus, les méthodes d'optimisation globale sont lentes et la qualité de la solution des méthodes locales (de type gradient) dépend largement du choix de la solution initiale[7]. La méthode décrite dans ce chapitre permet de déterminer une bonne estimation de la solution, pouvant servir de solution initiale pour l'optimisation dimensionnelle.

La première solution testée a consisté à réaliser l'apprentissage[8], dans le réseau neuronal, de la fonction F (voir Figure 9.1), c'est-à-dire avec les dimensions du mécanisme en entrée du réseau et les formes de trajectoire en sortie. Un algorithme d'inversion (cf. *Zell et al.* [ZMV+95]) devait permettre de déterminer, pour une sortie donnée, l'entrée correspondante. Dans ce cas, le problème de non définition de la fonction est évité, car à un jeu de dimensions correspond une seule forme de trajectoire[9]. Malheureusement, avec ce type de réseau neuronal on ne fait que déplacer le problème : à cause de la nature de la fonction F (non linéaire, non convexe et surtout non injective et non surjective), l'algorithme d'inversion, déjà très instable et peu fiable dans des cas plus simples, ne donne pas de bons résultats.

Nous avons donc choisi l'approximation, par le réseau neuronal, de la fonction inverse F^{-1} (voir Figure 9.2), avec les formes de trajectoire en entrée du réseau et les dimensions en sortie. Le principe de l'approche de synthèse utilisée pour la codification du contenu des cas est présenté dans la Figure 9.3. Les différentes étapes de cette méthode de synthèse seront détaillées dans les sous-chapitres suivants.

6. Par exemple, un mécanisme *quatre barres* génère des sextiques.

7. Néanmoins, les méthodes de continuation (cf. §2.4.3.8) et la méthode de recalage géométrique (cf. §8.2.5 et §8.3.4) peuvent diminuer la sensibilité par rapport au choix de la solution initiale.

8. L'apprentissage d'un réseau neuronal est une méthode pour optimiser les paramètres du réseau afin d'approximer au mieux la relation entre les entrées et les sorties du réseau. Voir les détails dans §C.2 et §9.6.2.3.

9. Un simulateur neuronal peut ainsi être réalisé, mais son intérêt est limité, car nous disposons déjà de méthodes efficaces de simulation cinématique.

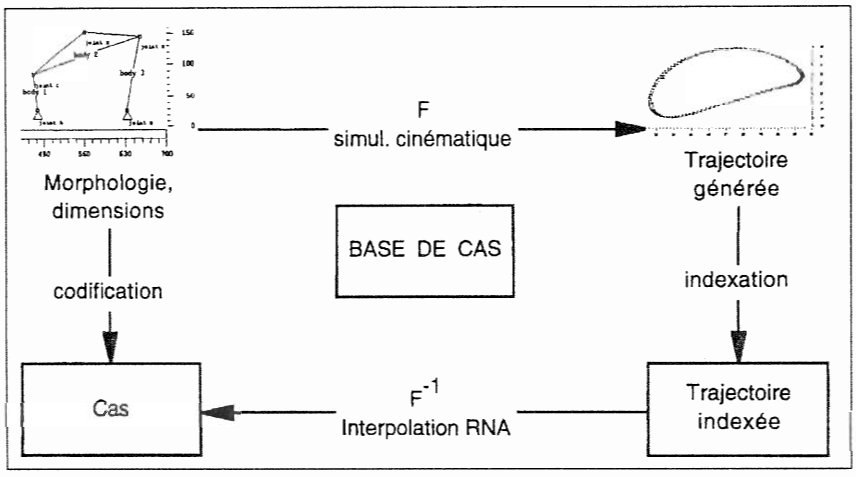


FIG. 9.3 – *Structure générale de la base de cas utilisant l'approximation, par un réseau neuronal artificiel (RNA), de la fonction reliant les dimensions du mécanisme aux formes de trajectoires générées.*

9.3 Modélisation et indexation des cas

9.3.1 Définition du cas

Un *cas* (au sens du *Raisonnement à base de cas*, voir l'annexe B) est défini dans REALISME comme une association entre :

- les dimensions du mécanisme pour une morphologie donnée (voir §9.5.2), et
- la forme des trajectoires correspondantes (voir §9.4).

Afin que la base de cas soit utile pour la conception, le choix des méthodes de représentation du contenu des cas doit respecter certaines contraintes. Ainsi, les cas doivent être *indexés*, d'une façon qui permette au concepteur d'obtenir rapidement une solution pour le problème de synthèse (§9.4). L'index et le contenu d'un cas sont des structures de données qui caractérisent la forme de la trajectoire et la structure d'un mécanisme (morphologie, dimensions, point effecteur). Ces structures de données doivent être *les plus petites possibles* pour réduire le plus possible la dimension du problème, d'où le besoin de *normaliser* les cas (§9.4.4), afin que chaque cas représente une classe entière de cas, équivalents quant à la forme de trajectoire. D'autres contraintes sont imposées par le choix d'un réseau neuronal comme moyen de stocker et d'interpoler les cas : données uniquement numériques (non-symboliques), à valeurs continues (non discrètes), nombre fixe des valeurs numériques stockées dans la même base, adimensionalisation des données (§9.6.2.4).

9.3.2 Méthode d'indexation des cas

Par *indexation des cas* on entend une transformation (ou codification), avant le stockage, des informations qui représentent le cas, afin de réduire sa taille et de faciliter la recherche des cas lors de l'utilisation. (Voir aussi l'annexe B.)

La méthode d'indexation utilisée dans REALISME, présentée dans la Figure 9.4, consiste en une série de transformations appliquées aux trajectoires et aux dimensions des mécanismes. Les détails sur les différentes étapes de la méthode seront présentés dans les sous-chapitres suivants.

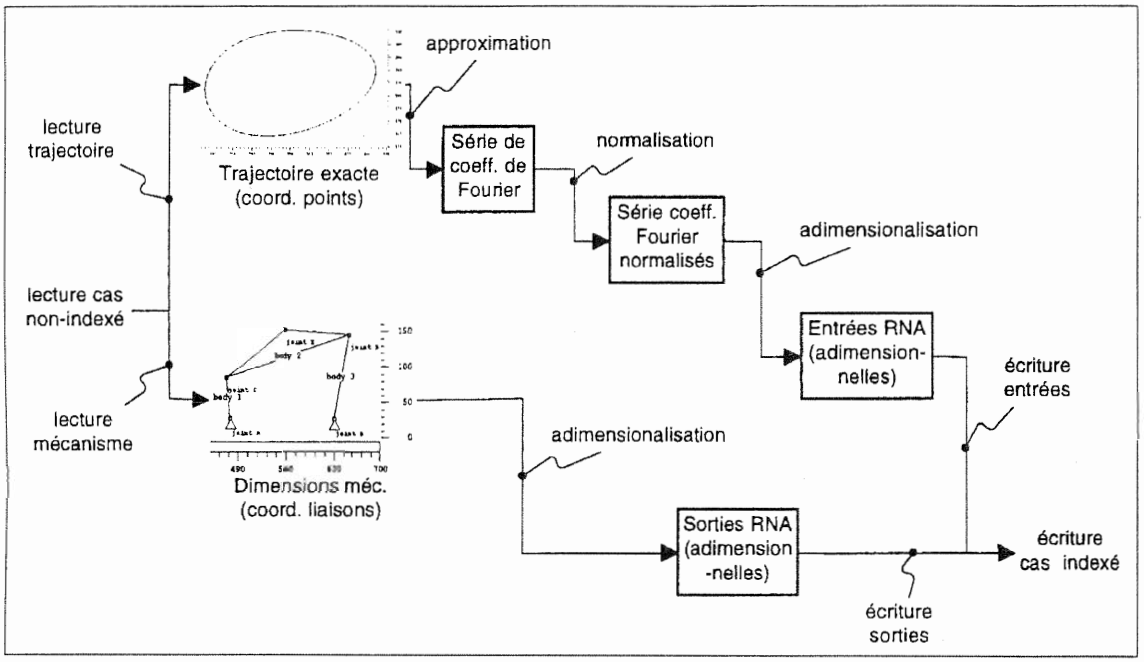


FIG. 9.4 – *Méthodologie d'indexation des cas.*

9.4 Indexation des trajectoires

Compte tenu de l'objectif de réaliser une synthèse sur le critère de la *forme de trajectoire*, il est nécessaire d'adopter une méthode de codification succincte de la forme des trajectoires générées par le point effecteur des mécanismes. Avant de présenter l'approche utilisée dans REALISME, nous étudierons d'abord une approche de la littérature qui s'est avérée inappropriée.

9.4.1 Prise en compte de la forme : l'approche de Kota et Chiou

Une méthode permettant de valuer la forme d'une trajectoire (donnée par des points), indépendamment de la translation, de l'orientation et du point de départ, est présentée par *Kota* [Kot92] et *Kota et Chiou* [KC93]. Afin de réaliser une optimisation dimensionnelle dont l'objectif est la synthèse d'une *forme* de trajectoire, ces articles proposent une méthode de calcul du degré de *déviation* (ou *index de similarité*) entre deux trajectoires (ouvertes ou fermées). Les trajectoires désirée et générée sont en fait des trajectoires exactes (liste de points), mais la distance entre ces trajectoires est calculée en termes de forme et non pas comme une somme de carrée de distances entre les points.

La première étape consiste à choisir un point de départ parmi les points P_i de la trajectoire désirée (arbitrairement quand elle est fermée; une des extrémités quand elle est ouverte). Ensuite, pour chaque point, la longueur paramétrique ($PLENGTH$) et l'angle paramétrique ($PANGLE$) sont calculés. $PLENGTH_i$ représente le ratio entre la longueur du segment (P_0, P_i) et la somme des ces longueurs calculées pour tous les autres points. $PANGLE_i$ est l'angle entre les tangentes à la courbe dans les points P_0 et P_i. En traçant la courbe de $PANGLE$ en fonction de $PLENGTH$ la courbe paramétrique est obtenue. Finalement, la déviation ou l'écart de forme D est calculée par la formule :

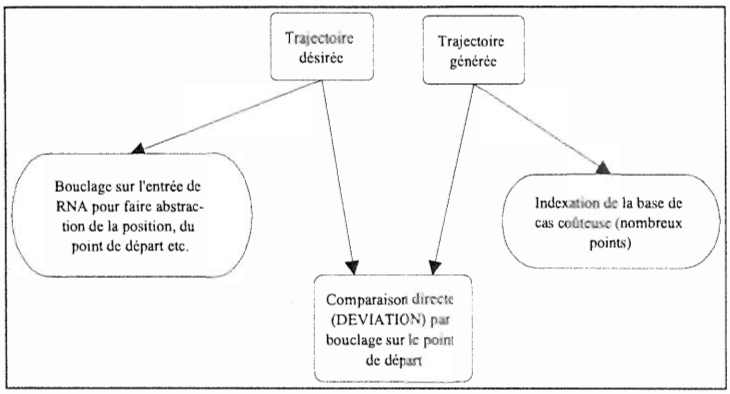


FIG. 9.5 – *Approche de synthèse de forme par comparaison directe des trajectoires générée et désirée, décrites par une suite de points.*

$$D = \sqrt{\sum_{i=1}^{n}(Y_{gi} - Y_{di})^2} \tag{9.1}$$

où Y_{gi} est l'ordonnée de la courbe paramétrique pour la trajectoire générée, Y_{di} est l'ordonnée de la courbe paramétrique pour la trajectoire désirée ; n est le nombre de points.

S'abstraire de la translation et de l'orientation est obtenu naturellement par la prise en compte des rapports des longueurs. Cependant, s'abstraire du point de départ est une opération très coûteuse, étant réalisé par n calculs successifs de la déviation D, en prenant à chaque fois un autre point de départ sur la courbe désirée et en retenant, au final, la valeur la plus petite de D.

L'inconvénient majeur de la méthode de Kota et Chiou est que la métrique donnée par l'équation 9.1 n'est valable que par rapport à une trajectoire désirée unique et ne représente donc pas la forme de la trajectoire en soi. Par conséquent, leur approche de synthèse consiste en une génération d'un ensemble de cas pour chaque trajectoire désirée, par des *plans d'expériences*. Par contre, nous avons besoin d'une méthode de prise en compte de la forme indépendante de la trajectoire désirée. Pour cette raison, la méthode de Kota et Chiou ne peut pas être utilisée pour la création d'un catalogue de formes de trajectoires.

De manière générale, il apparaît que les approches basées sur une comparaison directe de la forme des trajectoires désirée et générée, utilisant une description des trajectoires par une collection de points (voir Figure 9.5), ne sont pas satisfaisantes.

9.4.2 L'approche de McGarva

A cause des raisons exposées précédemment, il est nécessaire d'utiliser une méthode permettant de s'abstraire complètement de la représentation des trajectoires par des points, qui est une source inutile de complexité. Pour cela, nous avons repris le modèle de *McGarva* [MM93] [McG94], consistant à utiliser une analyse harmonique (en série de Fourier) des trajectoires exactes. Conformément

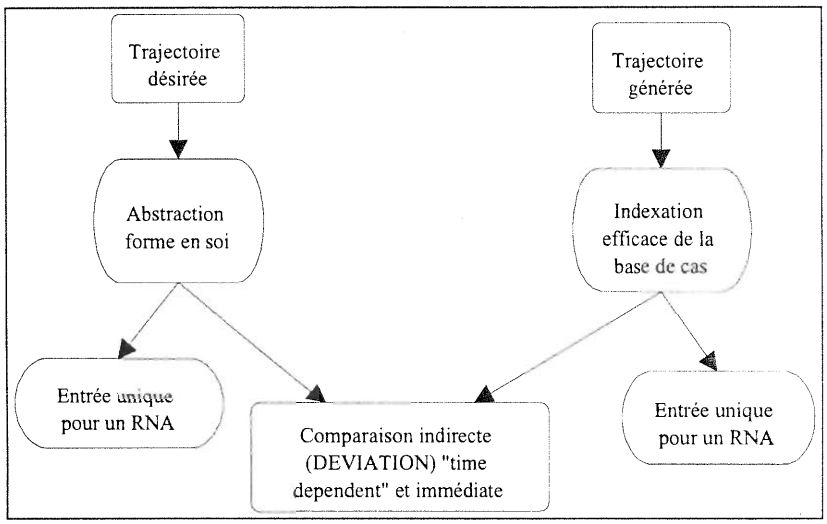


FIG. 9.6 – *Comparaison indirecte des trajectoires en termes de forme, par l'intermédiaire d'une analyse en série de Fourier.*

aux buts généraux de l'indexation des cas (cf. p. 133), les avantages de cette représentation sont les suivants :

- la quantité d'information à stocker diminue fortement, puisque le vecteur des coordonnées des points composant la trajectoire est remplacé, moyennant une approximation, par celui des coefficients de Fourier, moins nombreux ;[10]
- les coefficients de Fourier, grâce à la normalisation (voir §9.4.4), modélisent uniquement la forme, en particulier indépendamment de la répartition particulière des points sur la trajectoire (qui n'est pas, dans ce cas, un critère fonctionnel).

En conclusion, l'approche proposée par McGarva, de prise en compte indirecte de la forme des trajectoires par l'intermédiaire d'une analyse de Fourier (voir Figure 9.6), est beaucoup plus adaptée à l'usage pour la synthèse de trajectoire avec *indépendance du temps*. Les sous-chapitres suivants présentent les détails de la méthode d'indexation de McGarva, adoptée dans REALISME avec quelques modifications.

9.4.3 L'analyse en série de Fourier (calcul des coefficients)

La position d'un point appartenant à une trajectoire peut être modélisée par une fonction complexe, exprimée dans le plan complexe :

$$z(t) = x(t) + iy(t) \tag{9.2}$$

Pour des trajectoires fermées, la fonction est périodique (de période unitaire) : $z(t+1) = z(t)$. Le développement en série de Fourier[11] s'écrit alors :

$$z(t) = \sum_{m=-\infty}^{\infty} a_m \exp(2\pi imt) \tag{9.3}$$

10. En pratique, après normalisation (cf. §9.4.4), 17 valeurs numériques de coefficients sont utilisées, correspondant aux cinq premières harmoniques. Si une précision plus grande est nécessaire, la procédure de création de la base de cas permet très facilement d'augmenter le nombre d'harmoniques prises en compte.

11. Sous réserve que $z(t)$ respecte les conditions de Dirichlet.

Les coefficients complexes de Fourier sont donnés par la formule :

$$a_m = \int_0^1 \exp(-2\pi imt)z(t)dt \tag{9.4}$$

Le coefficient a_0 est appelé fondamental, a_1 et a_{-1} représentent la première harmonique, a_2 et a_{-2} la deuxième harmonique, etc.

Si la fonction $z(t)$ est connue seulement par un ensemble de points z_k $(k = 1, \ldots, N)$ appartenant à la trajectoire, l'intégrale de l'équation 9.4 requiert une approximation numérique. Pour cela, $z(t)$ est paramétrée en termes de longueurs des segments reliant deux points successifs, t_k étant ainsi une approximation de la longueur de la trajectoire entre le premier point et le point courant z_k. En ces conditions, l'approximation numérique de l'intégrale peut être calculée par la formule des trapèzes :

$$a_m = \frac{1}{N}\sum_{k=0}^{N-1}\frac{t_{k+1}-t_k}{2}\left(z_{k+1}\exp(-2\pi imt_{k+1}) + z_k\exp(-2\pi imt_k)\right) \tag{9.5}$$

En développant l'équation 9.5, nous obtenons[12] :

$$\begin{aligned}
\mathrm{Re}(a_m) &= \frac{1}{N}\sum_{k=0}^{N-1}\Big(\frac{t_{k+1}-t_k}{2}(x_{k+1}\cos(-2\pi mt_{k+1}) - y_{k+1}\sin(-2\pi mt_{k+1}) \\
&\quad + x_k\cos(-2\pi mt_k) - y_k\sin(-2\pi mt_k))\Big) && (9.6)\\
\mathrm{Im}(a_m) &= \frac{1}{N}\sum_{k=0}^{N-1}\Big(\frac{t_{k+1}-t_k}{2}(x_{k+1}\sin(-2\pi mt_{k+1}) + y_{k+1}\cos(-2\pi mt_{k+1}) \\
&\quad + x_k\sin(-2\pi mt_k) + y_k\cos(-2\pi mt_k))\Big) && (9.7)
\end{aligned}$$

Afin de calculer les coefficients a_m, il est nécessaire de calculer les valeurs du paramètre t_k pour chaque point (x_i, y_i), $i = 1, \ldots, N$. Une approximation de la longueur totale L de cette trajectoire est le périmètre du polygone constitué par les points :

$$L = \sum_{i=1}^{N-1}\sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} + \sqrt{(x_1 - x_N)^2 + (y_1 - y_N)^2} \tag{9.8}$$

Avec la même approximation, la longueur L_k de la trajectoire entre le premier point et le point k est :

$$L_k = \sum_{i=1}^{k-1}\sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \tag{9.9}$$

Les paramètres t_k peuvent donc être déterminés par le rapport de ces longueurs :

$$t_k = \frac{L_k}{L} \tag{9.10}$$

En utilisant les valeurs de t_k dans les équations 9.6 et 9.7, les valeurs des coefficients complexes de Fourier peuvent être calculées. Seules les cinq premières harmoniques sont utilisées dans REALISME (mais le nombre d'harmoniques retenues peut être modifié par l'utilisateur).

Il est important de remarquer que, avec ce paramétrage de la trajectoire en termes de longueurs, l'information liée à la vitesse de parcours de la trajectoire a été éliminée de la représentation harmonique. C'est un aspect important pour l'application à une méthode de synthèse avec indépendance du temps.

12. Le nombre de points N doit respecter la relation : $N \geqslant 4(M+1) - 5$, où M est le nombre d'harmoniques qu'on veut calculer.

A partir de l'équation 9.3, en effectuant la somme pour les cinq premières harmoniques, il est possible de reconstituer avec une bonne approximation une trajectoire sous la forme d'une liste de points équirépartis (x_k, y_k) :

$$z_k(t) = x_k + iy_k = \sum_{m=-5}^{5} a_m \exp(2\pi i m t_k) \tag{9.11}$$

Cette reconstitution peut être utilisée pour évaluer l'erreur d'approximation commise. Les Figures 9.7, 9.8 et 9.9 présentent trois exemples typiques de trajectoires (sur chaque figure, la trajectoire d'origine et la reconstitution à partir des cinq premières harmoniques sont dessinées en superposé). La conclusion est que l'approximation par les cinq premières harmoniques est un compromis acceptable entre la précision d'approximation et le nombre de coefficients. Comme on pouvait s'attendre à cause du caractère global de l'approximation par l'analyse de Fourier, les erreurs d'approximation les plus grandes sont faites sur les parties de trajectoire présentant des discontinuités ou des changements locaux de forme. Néanmoins, la présence des harmoniques supérieures dans les trajectoires générées semble limitée, ce qui nous autorise à les négliger dans l'approximation.

9.4.4 Normalisation des coefficients de Fourier représentant les trajectoires

Le but de la normalisation est d'obtenir une représentation de la forme de la trajectoire, indépendamment des transformations affines qu'elle peut subir : translation, rotation, échelle, ainsi que de s'abstraire du point de départ et de la direction de parcours. A l'issue de cette procédure, les coefficients de Fourier de la trajectoire normalisée sont représentatifs pour une classe de trajectoires. La procédure de normalisation proposée par McGarva, prend en compte directement les coefficients de Fourier des harmoniques inférieures. La présentation ci-après reprend en partie celle de *McGarva* [MM93] [McG94]. Néanmoins, des différences avec la méthode utilisée dans REALISME seront signalées.

9.4.4.1 Abstraction de la translation

L'harmonique fondamentale a_0 représente la position du centre de gravité de la trajectoire, c'est-à-dire la moyenne des coordonnées des points. (Ceci peut être vérifié d'après l'équation 9.5 pour $m = 0$.) Le critère choisi pour la normalisation en translation est que le du centre de gravité soit située à l'origine du repère cartésien $(0, 0)$. La trajectoire déplacée étant représentée par $z(t) - a_0$, la translation (et donc la normalisation en translation) peut être obtenue simplement en annulant le coefficient de l'harmonique fondamentale :

$$a_0 \leftarrow 0 \tag{9.12}$$

Par conséquent, le nombre de valeurs numériques de coefficients à stocker pour chaque trajectoire est réduit de un.

9.4.4.2 Abstraction du point de départ

En effectuant le changement de variable $s = t + \frac{\beta}{2\pi}$ dans l'équation 9.3, la position du point de départ (pour $s = 0$) correspond à $t = -\frac{\beta}{2\pi}$. (Pour la signification de β, voir §9.4.4.3.) On obtient donc :

$$z(s) = \sum_{m=-\infty}^{\infty} a_m \exp\left(2\pi i m \left(s - \frac{\beta}{2\pi}\right)\right) = \sum_{m=-\infty}^{\infty} a_m \exp\left(\frac{2\pi i m s - i m \beta}{2\pi}\right)$$

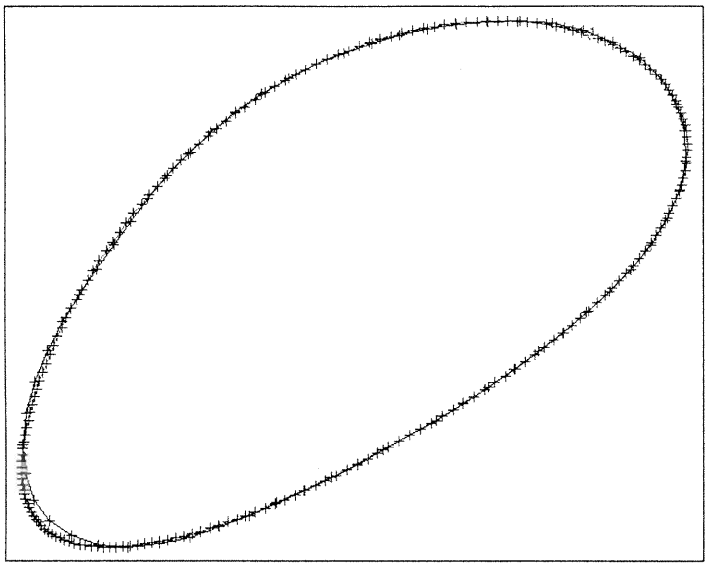


FIG. 9.7 – *Premier exemple d'approximation par les cinq premières harmoniques.*

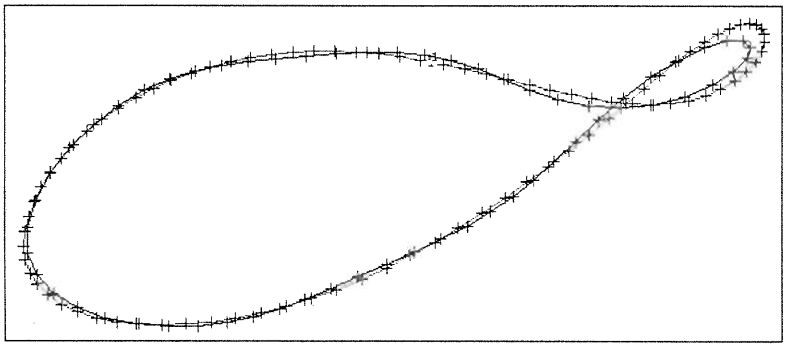


FIG. 9.8 – *Deuxième exemple d'approximation par les cinq premières harmoniques.*

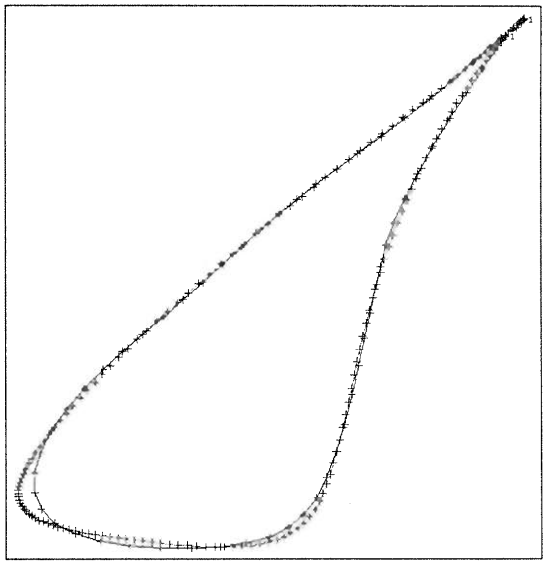


FIG. 9.9 – *Troisième exemple d'approximation par les cinq premières harmoniques.*

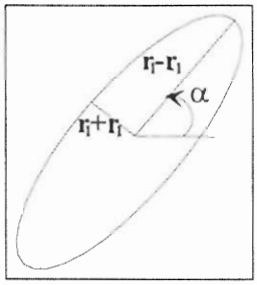


FIG. 9.10 – *L'ellipse correspondant à la première harmonique.*

$$= \sum_{m=-\infty}^{\infty} a_m \exp(2\pi i m s) \exp(-im\beta) \tag{9.13}$$

En comparant les équations 9.3 et 9.13, on constate que l'effet de la normalisation du point de départ peut être obtenue par la transformation :

$$a_m \leftarrow a_m \exp(-im\beta) \tag{9.14}$$

9.4.4.3 Abstraction de la rotation

A partir de l'équation 9.3, le coefficient complexe correspondant à la première harmonique s'écrit :

$$z_1(t) = a_1 \exp(2\pi i t) + a_{-1} \exp(-2\pi i t) \tag{9.15}$$

En écrivant chaque coefficient sous la forme $a_m = r_m \exp(i\Phi_m)$, avec $r_m = |a_m|$ et $\Phi_m = \arg(a_m)$, nous obtenons :

$$\begin{aligned} z_1(t) &= r_1 \exp(2\pi t + \Phi_1) + r_{-1} \exp\left(i(-2\pi t + \Phi_{-1})\right) = \\ &= \exp(i\alpha)\left(r_1 \exp(2\pi t + \beta) + r_{-1} \exp\left(i(-2\pi t - \beta)\right)\right) \end{aligned} \tag{9.16}$$

où

$$\alpha = \frac{\Phi_1 + \Phi_{-1}}{2} \tag{9.17}$$

$$\beta = \frac{\Phi_1 - \Phi_{-1}}{2} \tag{9.18}$$

Finalement, en développant l'équation 9.16, nous obtenons :

$$z_1(t) = \exp(i\alpha)\left((r_1 + r_{-1})\cos(2\pi t + \beta) + i(r_1 - r_{-1})\sin(2\pi t + \beta)\right) \tag{9.19}$$

Le terme entre les parenthèses carrées représente une ellipse centrée sur l'origine et dont le grand demi-axe est $(r_1 + r_{-1})$ et le petit demi-axe est $(r_1 - r_{-1})$ (voir Figure 9.10). L'effet de la multiplication avec $\exp(i\alpha)$ est une rotation de l'ellipse, en sens anti-horaire, d'angle α.

L'axe principal de la trajectoire est orienté par rapport à l'abscisse selon l'angle α[13]. La normalisation consiste en une rotation d'angle $-\alpha$ autour du centre de gravité :

$$a_m \leftarrow a_m \exp(-i\alpha) \tag{9.20}$$

13. Dans *McGarva* [McG94], il y a une erreur dans l'expression de α.

La conséquence est que la trajectoire normalisée est « allongée» avec sa plus grande dimension en horizontale.

Cependant, cette normalisation en rotation, décrite ici conformément à la procédure de McGarva, n'est définie qu'à un angle π près. En effet, en examinant la formule 9.17, on constate que l'angle α est déterminé à π près, les angles Φ_1 et Φ_{-1} étant définis à 2π près. Par conséquent, après la transformation 9.20, nous utilisons dans REALISME une procédure pour déterminer s'il faut éventuellement effectuer une rotation supplémentaire de π. Cette procédure (voir Algorithme 9.1) détermine la première harmonique qui n'est pas symétrique et fournit le critère pour décider si une rotation supplémentaire de π est nécessaire.

```
Procédure Test-rotation-de-π (entrée : a_k, k = 1, ..., M ; sortie : res)
    res ← 0
    eps ← 10^-8
    Pour k variant de 2 à M
        Si (k paire)
            val ← Re(a_k + a_-k)
        Sinon
            val ← Im(a_k + a_-k)
        Fin Si

        Si (abs(val) > eps) % non-nulle à une précision près
            Si (val < 0)
                res ← 1
            Fin si
            Break Pour % sortie de la boucle
        Fin Si
    Fin Pour
Fin Procédure
```

Algorithme 9.1: *Procédure permettant de savoir s'il faut effectuer une rotation supplémentaire d'angle π.*

Si le résultat de cette procédure est $res = 1$, on réalise donc une rotation supplémentaire d'angle π de la trajectoire, ce qui est obtenu simplement en changeant le signe des coefficients :

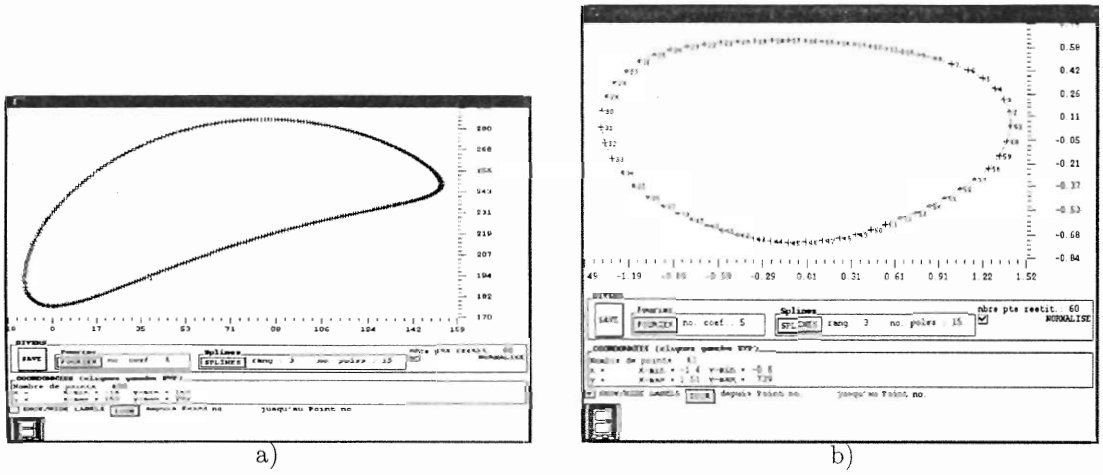
$$a_m \leftarrow -a_m \tag{9.21}$$

Finalement, il faut compenser l'effet de cette dernière rotation sur le point de départ (cf. §9.4.4.2) :

$$a_m \leftarrow a_m \exp(-im\pi) \tag{9.22}$$

9.4.4.4 Abstraction de la direction

Lorsque $r_1 > r_{-1}$, le petit demi-axe de l'ellipse a une longueur positive et l'ellipse est tracée en sens anti-horaire. En ce qui concerne les autres harmoniques, le terme $z_m(t)$ représente une ellipse centrée sur l'origine et dont les axes sont $(r_m + r_{-m})$ et $(r_m - r_{-m})$. Le sens de parcours est anti-horaire lorsque $r_m > r_{-m}$.



a) b)

FIG. 9.11 – *Exemple de trajectoire exacte (a) et trajectoire normalisée à partir des cinq premières harmoniques (b).*

La direction normalisée de parcours de la trajectoire est celle qui correspond au sens anti-horaire pour la première harmonique, c'est-à-dire pour :

$$|a_1| > |a_{-1}| \tag{9.23}$$

Quand cette relation n'est pas respectée, la direction est renversée en remplaçant a_m par a_{-m} et réciproquement pour toutes les valeurs de m.

9.4.4.5 Abstraction de l'échelle

La normalisation de l'échelle prend en considération la longueur du grand demi-axe de l'ellipse correspondant à la première harmonique, c'est-à-dire $|a_1| + |a_{-1}|$. La normalisation est obtenue en divisant tous les coefficients par $|a_1|$ (supposé non nul) :

$$a_m \leftarrow \frac{a_m}{|a_1|} \tag{9.24}$$

Après cette opération, le nombre de coefficients à stocker se réduit encore de deux, car nous avons :

$$r_1 = 1 \tag{9.25}$$

9.4.4.6 Effets sur les coefficients

En combinant l'inéquation 9.23 et l'équation 9.25, on obtient :

$$0 < r_{-1} < 1 \tag{9.26}$$

Après la normalisation, les coefficients a_1 et a_{-1} deviennent réels : $a_1 = 1 + 0i$ et $a_{-1} = r_{-1} + 0i$. Par ailleurs, la normalisation en translation a annulé le coefficient $a_0 = 0 + 0i$. Au total, la normalisation a donc comme résultat, entre autres, de diminuer le nombre de valeurs de coefficients de $(5 \times 2 + 1) \times 2 = 22$ à $17 = 22 - 5$.

m	Re(a_m)	Im(a_m)
-5	0,65	1,26
-4	0,94	-0,34
-3	5,26	0,32
-2	2,88	2,59
-1	-11,51	-57,69
0	63,44	235,91
1	-17,74	9,91
2	-6,13	-0,26
3	-0,41	-0,17
4	0,44	1,14
5	-0,20	0,24

TAB. 9.1 – *Coefficients de Fourier avant normalisation.*

m	Re(a_m)	Im(a_m)
-5	-0,01	0,00
-4	0,00	0,02
-3	0,00	0,00
-2	0,01	0,10
-1	0,34	0,00
0	0,00	0,00
1	1,00	0,00
2	0,00	0,07
3	0,09	0,00
4	0,00	-0,02
5	0,02	0,00

TAB. 9.2 – *Coefficients de Fourier après normalisation.*

9.4.4.7 Exemple

Pour la trajectoire de la Figure 9.11.a, la reconstitution d'une trajectoire à partir des cinq premières harmoniques, après normalisation des coefficients, est donnée dans la Figure 9.11.b. Les coefficients avant normalisation sont donnés dans le Tableau 9.1. Les coefficients normalisés sont donnés dans le Tableau 9.2. Le grand nombre de coefficients nuls s'explique par la symétrie et la régularité de la trajectoire, mais ce n'est pas souvent le cas.

9.5 Stockage des morphologies et des dimensions

9.5.1 Morphologies

La codification de la morphologie d'un mécanisme, c'est-à-dire celle d'un graphe, est assez volumineuse. D'autre part, l'interpolation d'un RNA entre plusieurs morphologies n'a pas forcément de sens, car il n'y a plus aucune continuité. Pour ces raisons, la procédure de REALISME consiste en un bouclage sur la recherche dans plusieurs bases de cas, dédiées chacune à une morphologie. Ne sont donc codifiées que les dimensions.

En pratique, les bases de cas sont créées pour les morphologies les plus simples ou les moins coûteuses. En ce qui concerne les critères permettant, avant même le choix des dimensions, de privilégier par une estimation des coûts une morphologie par rapport à une autre, il existe dans la littérature des travaux, basés essentiellement sur la définition de mesures de la complexité (voir par exemple *Sardain* [Sar93, p. 130–139], qui distingue entre la complexité de conception et la complexité de fabrication). Pour l'instant, ces critères, d'ailleurs difficiles à pondérer, n'ont pas été implémentés dans REALISME.

9.5.2 Dimensions

Les dimensions des mécanismes peuvent être décrites par différents paramétrages. Comme pour l'optimisation dimensionnelle (voir §8.2.2), REALISME utilise le paramétrage cartésien absolu par les coordonnées des liaisons dans la position initiale du mécanisme. Pour la méthode de synthèse proposée dans le présent chapitre, les avantages de ce paramétrage sont l'unicité de définition (un seul assemblage possible) et l'homogénéité des variables (pas de mélange entre coordonnées et angles).

Afin de réaliser une approximation par un réseau neuronal de l'application entre les dimensions et les formes de trajectoires, le choix des paramètres dimensionnels et des intervalles de valeurs pour chaque paramètre est assujetti aux contraintes suivantes :

- la minimisation du nombre de paramètres (pour minimiser la dimension de la fonction dont on cherche l'approximation) ;
- l'élimination, autant que possible, des phénomènes qui risquent de gêner l'approximation :
  - la présence de plusieurs jeux de dimensions différents générant des formes *identiques* de trajectoires ;
  - l'existence, à l'intérieur des plages de valeurs des dimensions, de jeux de dimensions infaisables (pour lesquels aucune trajectoire fermée ne peut être générée par le mécanisme) ;
- la taille des intervalles de valeurs pour chaque paramètre dimensionnel doit être limitée, car des intervalles trop larges diminuent le taux de remplissage avec des échantillons lors de la création de la base de cas, c'est-à-dire augmentent la distance moyenne entre deux cas (voir

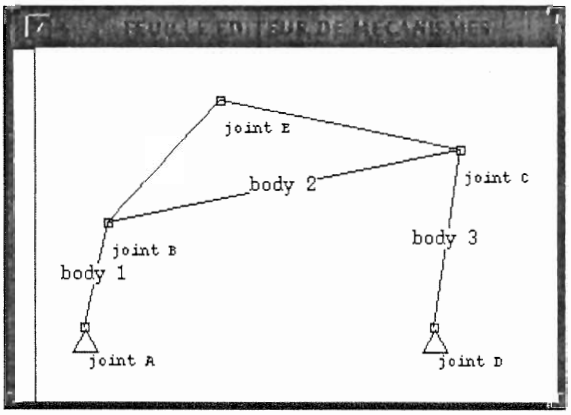


FIG. 9.12 – *Mécanisme* quatre barres. *La liaison motrice est en* A *et le point effecteur est le point* E.

§9.6) ; pourtant, les intervalles ne doivent être trop étroits non plus, afin de préserver une grande variété de formes de trajectoires générées.

Nous expliquerons ci-après les choix effectués dans REALISME sur l'exemple du mécanisme *quatre barres* (voir Figure 9.12).

9.5.2.1 Choix d'un paramétrage pour le mécanisme *quatre barres*

Les liaisons pivot attachées au bâti, A et D, sont fixées à des positions conventionnelles : $x_A = y_A = 0$, $x_D = 100$, $y_D = 0$. Toute position de la manivelle pouvant être la position initiale[14], nous considérons que la manivelle est initialement horizontale et orientée à gauche. Nous avons donc, pour l'instant, $x_B < 0$, $y_B = 0$. Ces choix ne vont pas réduire la diversité de formes de trajectoires réalisables, car il suffit ensuite d'effectuer une transformation géométrique affine pour retrouver les trajectoires correspondant à d'autres positions pour les liaisons A et D. Restent donc à prendre en compte cinq paramètres indépendants : x_B, x_C, y_C, x_E et y_E.

9.5.2.2 Choix des intervalles de valeurs pour chaque paramètre

Il existe plusieurs solutions pour le choix des intervalles, compte tenu des contraintes énoncées précédemment et notamment celle concernant la faisabilité d'un maximum de jeux de dimensions à l'intérieur des intervalles choisis. Une solution consisterait à utiliser, pour le mécanisme *quatre barres* (Figure 9.12), la condition de Grashof (cf. *Barker* [Bar85]) pour que la manivelle puisse exécuter une rotation complète :

$$p + g < m + n \tag{9.27}$$

$$p = \begin{cases} AD & \text{ou} \\ AB & \end{cases} \tag{9.28}$$

où p et la plus petite longueur de barre, g la plus grande, m et n étant les deux autres longueurs ; la plus petite longueur doit être soit AD, soit AB (la manivelle). Malheureusement, le choix d'intervalles afin de respecter cette condition est très difficile, surtout à cause de la disjonction introduite

14. L'angle initial de la manivelle n'influence pas la forme de la trajectoire générée (si la position du point de départ n'est pas un critère fonctionnel).

par 9.28. Par conséquent, il faudrait réaliser, en fin de compte, plusieurs réseaux neuronaux correspondant chacun à un sous-espace des dimensions. De plus, nous ne disposons pas d'une condition comme celle de Grashof pour toutes les morphologies.

Dans REALISME, nous avons donc choisi les intervalles sans tenir compte rigoureusement de la condition de Grashof, en admettant ainsi la présence, dans notre espace dimensionnel, de sous-espaces infaisables. Néanmoins, ces intervalles sont choisis d'une manière qui réduise, autant que possible, la présence de ces sous-espaces. Lorsqu'un jeu de dimensions s'avère être infaisable, le cas ne sera pas enregistré dans la base de cas. Au moins dans le cas du *quatre barres*, ce filtrage *a posteriori* n'affectera que dans une mesure acceptable la qualité de l'approximation et des résultats fournis par le réseau neuronal, surtout dans le voisinage des sous-espaces infaisables. La situation correspond à une fonction définie avec des « trous ».

Ces considérations nous ont conduit au choix, pour le mécanisme *quatre barres*, des intervalles et des positions suivants :

$$\begin{array}{cc} x_A = 0 & y_A = 0 \\ -40 \leqslant x_B \leqslant -5 & y_B = 0 \\ 0 \leqslant x_C \leqslant 55 & 5 \leqslant y_C \leqslant 100 \\ x_D = 100 & y_D = 0 \\ 0 \leqslant x_E \leqslant 100 & 0 \leqslant y_E \leqslant 100 \end{array}$$

Ce choix des intervalles est assez empirique[15] et, comme nous l'avons déjà dit, il diminue la qualité des résultats. Pour s'affranchir de ces limites, il suffirait de pouvoir réaliser l'approximation à la fois pour des formes de trajectoires fermées et ouvertes. Malheureusement, nous ne connaissons pas une telle représentation continue, homogène, à la fois des courbes ouvertes et fermées (voir chap. 11). Si l'analyse de Fourier s'applique aux fonctions périodiques et donc aux trajectoires fermées, une modélisation de type splines ou nurbs est spécifique aux trajectoires ouvertes. Dans ce dernier cas, pour les trajectoires fermées, il faudrait choisir arbitrairement un point de coupure.

9.6 Création de la base de cas

9.6.1 Méthode de génération aléatoire

La complexité de la fonction dont on cherche l'approximation impose l'utilisation d'un grand nombre d'échantillons de valeurs, donc de beaucoup de cas[16]. Dans REALISME, la génération des cas est automatisée (voir Figure 9.13), à partir du choix, par l'utilisateur, du paramétrage et des intervalles de valeurs.

L'échantillonnage aléatoire (par un tirage aléatoire des paramètres dimensionnels) est préférable – vue la non linéarité de la fonction – à l'échantillonnage par un balayage régulier, à intervalle constant.

La procédure de génération aléatoire des cas (voir Figure 9.14) est donnée dans l'Algorithme 9.2.

15. Nous ne disposons pas d'une procédure générale permettant de déterminer les intervalles faisables pour toute morphologie de mécanisme, bien que nous disposions des contraintes de non-blocage locales pour chaque groupe multipolaire (voir §4.2.1.1). En effet, ces contraintes ne portent pas sur la seule position initiale du mécanisme, mais sur toutes ses positions successives.

16. Pour le mécanisme quatre barres, 60000 cas ont été générés.

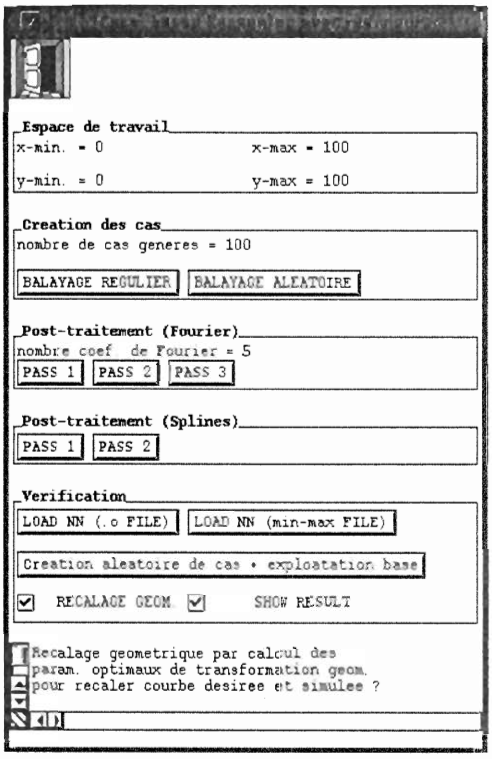


FIG. 9.13 – *Panneau de* REALISME *permettant la génération des cas pour la méthode de synthèse par réseaux neuronaux.*

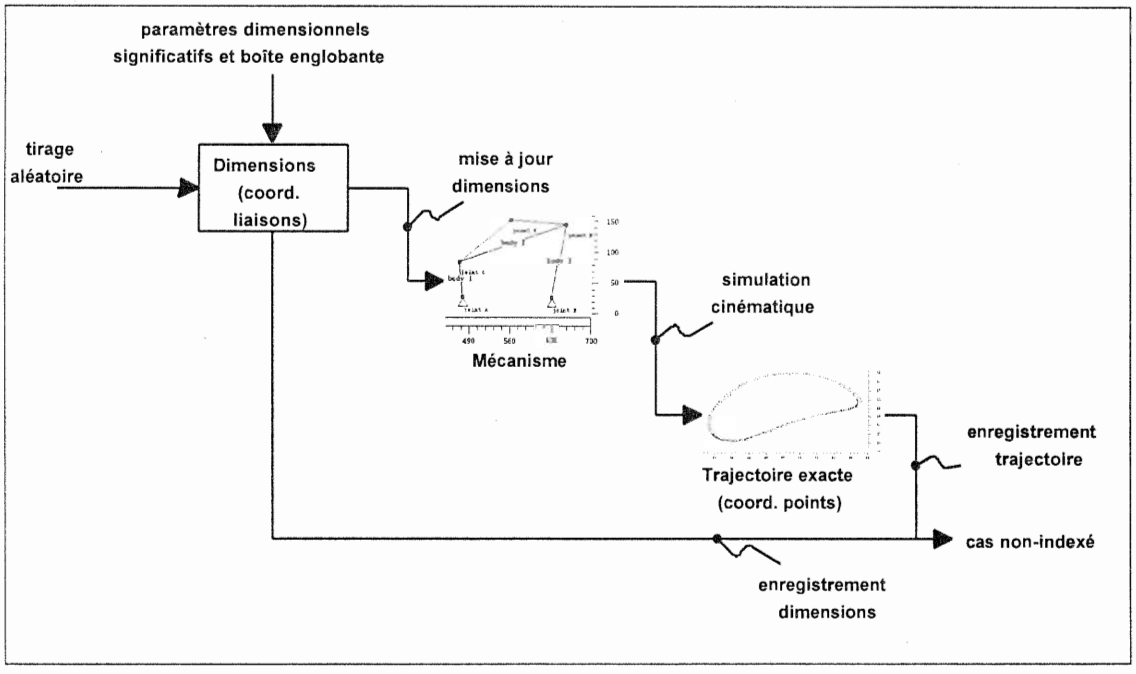


FIG. 9.14 – *Méthodologie de génération de cas par tirage aléatoire des paramètres dimensionnels*

Etape 1 Choix de la morphologie de mécanisme ;

Etape 2 Choix des paramètres dimensionnels (coordonnées de liaisons) et des intervalles de valeurs pour chaque paramètre (voir §9.5.2) ;

Etape 3 Tirage aléatoire des valeurs des coordonnées[a], à l'intérieur des intervalles choisis ;

Etape 4 Mise à jour (dans la structure informatique de données représentant le mécanisme) de la position initiale du mécanisme, à partir des coordonnées tirées ;

Etape 5 Simulation cinématique (pour une rotation complète de la manivelle)[b] et enregistrement de la trajectoire du point effecteur, c'est-à-dire des coordonnées des points atteints ;

Etape 6 Calcul de l'index du cas pour la trajectoire effectuée et pour les paramètres dimensionnels du mécanisme (voir §9.5.2 et §9.4) ;

Etape 7 Enregistrement du cas (indexé).

[a] Il est indispensable d'employer un « bon » générateur de nombres aléatoires, avec une période suffisamment grande (compte tenu du grand nombre de cas nécessaires – 60000 ont été utilisés pour le seul *quatre barres*) et une distribution uniforme des valeurs.

[b] La discrétisation de la position de la manivelle durant la simulation cinématique doit être suffisamment fine (200 positions dans REALISME) pour que l'erreur due à l'approximation par l'analyse en série de Fourier de la trajectoire générée soit réduite.

Algorithme 9.2: *Génération aléatoire de cas pour la constitution d'une base de cas.*

9.6.2 Le réseau neuronal

La méthode mathématique d'approximation de fonction utilisée dans REALISME est un réseau neuronal artificiel (RNA). Une présentation sommaire de cette méthode est donnée dans l'annexe C.

9.6.2.1 Structure du réseau

La structure de réseau neuronal utilisée est une structure classique, de type *feedforward* (cf. l'annexe C.1), avec les caractéristiques suivantes (voir la représentation graphique du RNA dans le logiciel *SNNS* dans la Figure 9.15) :

- 17 neurones d'entrée, correspondant aux valeurs numériques significatives pour les coefficients de Fourier modélisant la forme des trajectoires, après normalisation (voir §9.4.4.6) ;
- 2 couches cachées de 22 neurones chacune[17] ;
- 5 neurones de sorties, correspondant aux paramètres dimensionnels du mécanisme ;
- une connectivité totale entre tous les neurones de la couche i avec tous les neurones de la couche $i+1$ (c'est la topologie classique pour un réseau *feedforward*). Le nombre total de connexions est donc de $968 = 17 \times 22 + 22 \times 22 + 22 \times 5$;

17. Si le nombre de neurones d'entrée et de sortie est imposé par le nombre de paramètres du problème, il n'existe pas de critère clair pour le choix de la composition des couches cachées. Dans la littérature, certains principes sont néanmoins fournis (cf. §2.5.3). Après plusieurs essais, le choix de deux couches cachées à 22 neurones chacune semble satisfaisant.

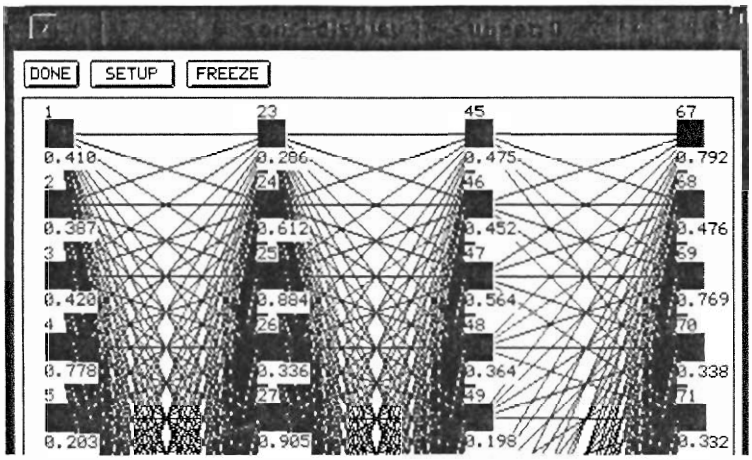


FIG. 9.15 – *Structure du réseau neuronal (copie d'écran du logiciel* SNNS*).*

- une fonction d'activation des neurones (sauf pour la couche d'entrée) de la forme $f_{act}(x) = \frac{1}{1+e^{-x}}$ (fonction souvent utilisée dans la littérature pour ses propriétés d'approximation; cf. l'annexe C.1).

9.6.2.2 Partitionnement des cas

Pour l'utilisation des cas, c'est-à-dire des couples dimensions-formes de trajectoires, comme *jeu de données* lors de l'apprentissage du RNA, l'ensemble des cas obtenus doit être segmenté en trois parties: *jeu de données d'apprentissage* (utilisé pour calculer la fonction erreur durant l'apprentissage), *jeu de données de validation* (pour calculer, pendant l'apprentissage, l'évolution de la capacité de généralisation du réseau) et *jeu de donnée de test* (pour tester la capacité de généralisation sur des données non utilisées durant l'apprentissage. (Voir l'annexe §C.3.) Pour le mécanisme *quatre barres*, les 60000 cas ont été répartis de la manière suivante:

- jeu de données d'apprentissage: 30000 cas;
- jeu de données de validation: 15000 cas;
- jeu de données de test: 15000 cas.

9.6.2.3 Algorithmes d'apprentissage et d'initialisation

Plusieurs algorithmes d'apprentissage du réseau neuronal ont été testés: *rétropropagation* standard, *vanilla*, *BackpropMomentum* (Rétropropagation avec terme de moment et *flat spot elimination*), *BackpropWeightDecay* (Rétropropagation avec *weight decay*), *Quickprop*, *backpercolation*, ainsi que des méthodes globales comme *MonteCarlo* et recuit simulé.

L'algorithme *Rprop* (*Resilient backpropagation*) est celui qui a donné les meilleurs résultats (voir les détails dans §C.5, p. 208).

Utilisant le jeu de données d'apprentissage (voir §9.6.2.2), l'apprentissage a été effectué en 14000 itérations. La courbe de convergence est donnée dans la Figure 9.16. La durée totale de l'apprentissage a été d'environ 30 heures, pour une erreur quadratique d'approximation E de moins de 5%. Cette erreur est définie par la relation:

$$E = \sum_{j \in jeux_données} \sum_{s \in var_sortie} (t_{js} - o_{js})^2 \tag{9.29}$$

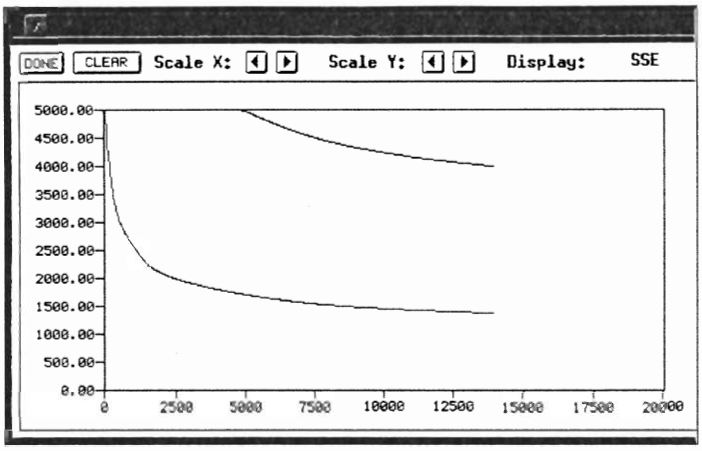


FIG. 9.16 – *Courbe de convergence de l'algorithme d'apprentissage. La courbe en bas représente l'évolution de l'erreur sur le jeu de données d'apprentissage et la courbe en haut celle de l'erreur sur le jeu de données de validation. (Copie d'écran du logiciel* SNNS.*)*

où t_{js} est la valeur obtenue en sortie du réseau et o_{js} est la valeur de sortie contenue dans le jeux de données.

9.6.2.4 Adimensionnalisation des entrées et des sorties

Les valeurs numériques représentant la forme des trajectoires et les dimensions des mécanismes obtenues par la méthode d'indexation des cas (cf. §9.4 et §9.5.2) seront ensuite utilisées comme valeurs d'entrée et de sortie du réseau neuronal artificiel. La nature mathématique de la fonction de transfert (activation) utilisée (voir page 149) impose d'étaler ces valeurs entre 0 et 1. Une procédure d'adimensionalisation est donc nécessaire, afin de ramener la valeur minimum à 0 et la valeur maximum à 1.

Il reste cependant à déterminer s'il s'agit des valeurs minimales et maximales séparément pour chaque entrée et sortie, ou globalement. Dans REALISME, la méthode utilisée n'est pas la même pour les entrées et pour les sorties :

- *entrées* : détermination du minimum et du maximum sur tous les cas séparément pour chaque entrée, afin que la fonction dont on réalise l'approximation soit définie par des échantillons répartis sur l'intégralité de l'intervalle de valeurs de chaque paramètre ;

- *sorties* : détermination d'un minimum et d'un maximum global pour toutes les sorties (car pour les sorties il n'est pas indispensable que les échantillons soient répartis sur l'intégralité de l'intervalle).

Si $e_i^{(k)}$ et $s_i^{(k)}$ sont la $i^{\text{ème}}$ entrée, respectivement sortie dans le $k^{\text{ème}}$ cas, les valeurs adimensionnelles correspondantes $e_i^{(k)\prime}$ et $s_i^{(k)\prime}$ sont donc calculées ainsi :

$$e_i^{(k)\prime} = \frac{e_i^{(k)} - \min\limits_k \left(e_i^{(k)}\right)}{\max\limits_k \left(e_i^{(k)}\right) - \min\limits_k \left(e_i^{(k)}\right)} \quad , \quad i = 1, \ldots, n_e \text{ et } k = 1, \ldots, N$$

$$s_j^{(k)\prime} = \frac{s_j^{(k)} - \min\limits_j \left(\min\limits_k \left(s_j^{(k)}\right)\right)}{\max\limits_j \left(\max\limits_k \left(s_j^{(k)}\right)\right) - \min\limits_j \left(\min\limits_k \left(s_j^{(k)}\right)\right)} \quad , \quad j = 1, \ldots, n_s \text{ et } k = 1, \ldots, N \tag{9.30}$$

où n_e = nombre d'entrées, n_s = nombre de sorties et N = nombre de cas dans la base.

9.6.3 Intégration du réseau neuronal dans la plateforme de préconception

La construction du jeu de données servant à l'apprentissage du RNA est complètement intégrée dans REALISME (voir §9.6.1). Après la réalisation du fichier contenant les cas indexés, la construction du réseau neuronal et son apprentissage sont réalisés dans le logiciel *SNNS* (*Stuttgart Neural Network Simulator*). Ensuite, un utilitaire fourni avec *SNNS* permet de générer un programme en langage *C* à partir du fichier contenant la description du réseau au format *SNNS*. En phase d'utilisation, l'exploitation du réseau neuronal se fait sans jamais quitter REALISME et d'une façon complètement transparente pour l'utilisateur, par le mécanisme d'édition de liens dynamiques. A partir d'un menu, l'utilisateur peut charger différents fichiers objet correspondants à autant de réseaux neuronaux.

9.6.4 Comparaison avec la méthode de McGarva

McGarva [MM93] [McG94] présente une méthode de synthèse de générateurs de trajectoires basée sur la création d'un catalogue de cas, puis la recherche du cas le plus proche et son utilisation comme solution initiale pour une optimisation dimensionnelle « classique ». L'originalité de cette méthode consiste en l'approximation des trajectoires par leurs premières harmoniques (cette procédure, décrite au §9.4.2, a été utilisée dans notre propre méthode de synthèse). Pour un certain nombre de morphologies prédéfinies, les coefficients de Fourier normalisés et les dimensions caractéristiques du mécanisme correspondant à différents jeux de dimensions sont stockés dans une base de données.

En phase d'utilisation de la base de cas, la trajectoire désirée est elle aussi codifiée par une série de coefficients normalisés. Le cas précédent le plus proche est retrouvé dans la base de cas par une recherche de proximité des coefficients basée sur la fonction distance E :

$$E = \sum_{i=-M}^{M} \left(\left(A_i - A_i^{'}\right)^2 + \left(B_i - B_i^{'}\right)^2 \right) \tag{9.31}$$

où M est le nombre d'harmoniques utilisées pour l'indexation, A_i et B_i sont la partie réelle et imaginaire des coefficients normalisés pour la trajectoire désirée et $A_i^{'}$ et $B_i^{'}$ pour les trajectoires de la base de cas.

La méthode de codification par analyse harmonique est limitée aux trajectoires fermées. Il ne sera donc pas possible d'avoir comme objectif de synthèse un trajectoire ouverte ou un morceau de trajectoire fermée. Cette limitation, de taille, est commune à la méthode de McGarva et à celle proposée dans REALISME.

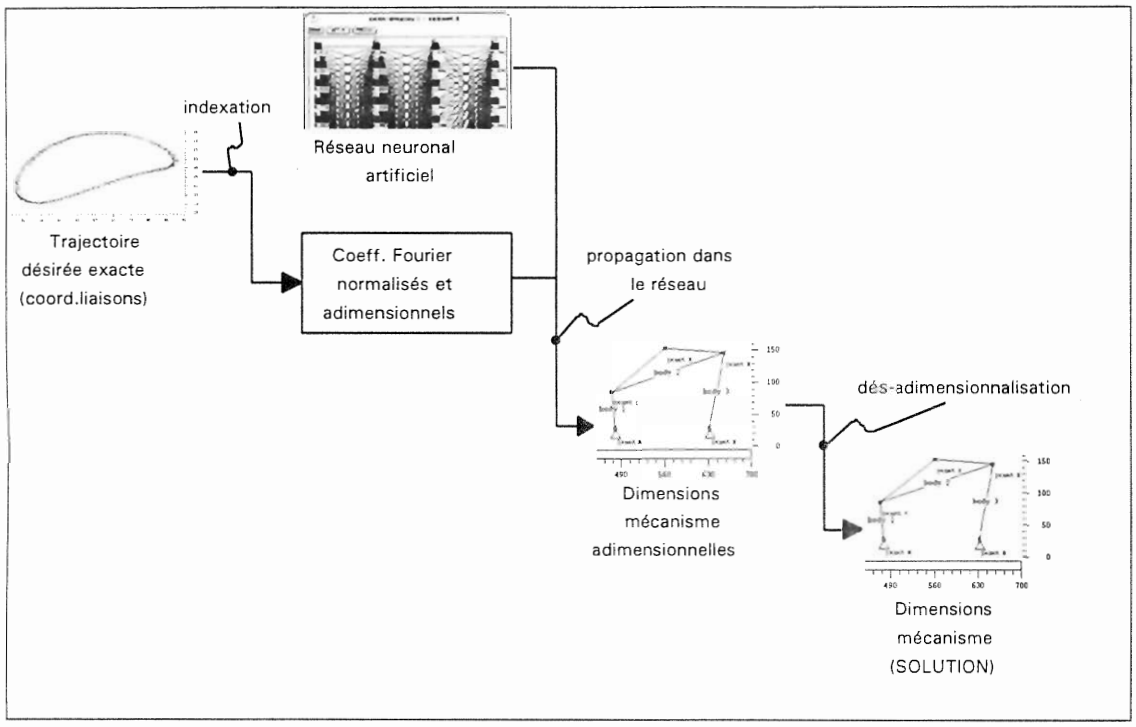


FIG. 9.17 – *Méthodologie d'utilisation du réseau neuronal artificiel.*

La deuxième limitation (ou plutôt désavantage) de l'approche de McGarva provient du choix de la méthode de « capitalisation » des cas. En effet, aucune interpolation des cas n'est réalisée lors de la recherche de solution. Vue la non linéarité et la multi-dimensionalité de la fonction reliant les trajectoires aux dimensions, l'utilisation d'une méthode permettant d'interpoler, telle que les RNA utilisés dans REALISME, s'avère plus efficace, car elle permet d'utiliser un grand nombre de cas, indispensable pour couvrir convénablement l'espace multi-dimensionnel, en temps masqué, autrement dit dans une phase préalable à son utilisation. Une fois l'interpolation réalisée, le logiciel de conception ne doit plus stocker qu'une quantité d'informations très réduite, correspondant au réseau neuronal. Le temps d'obtention de la solution est également très court.

9.7 Utilisation de la base de cas

9.7.1 Méthode

En phase d'utilisation de la base de cas, le concepteur dispose d'un certain nombre de réseaux neuronaux, correspondant à différentes morphologies de mécanismes. Les étapes à suivre, allant de la spécification de la forme de trajectoire désirée au mécanisme synthétisé, sont indiquées dans la Figure 9.17 et seront détaillées dans les sous-chapitres suivants.

9.7.2 Description du cas : spécification de la forme de trajectoire désirée

La saisie de la forme de trajectoire désirée est réalisée dans REALISME par l'intermédiaire d'un éditeur de trajectoires (voir aussi la description au §6.3.1 ; une copie d'écran est donnée dans la Figure 6.4, p. 92). Le concepteur dispose d'une fenêtre graphique, dimensionnée automatiquement en fonction de la boîte englobante utilisée pour la génération aléatoire des cas (voir §9.5.2). A l'aide de la souris, des points de contrôle d'une courbe spline sont définis. Des fonctions d'effacement,

d'insertion de points de contrôle, de retour en arrière sont accessibles par des boutons. Ces points de contrôle sont utilisés pour calculer la courbe spline ; une série de points équirépartis sur cette courbe est affichée, superposée sur les points de contrôle définis par l'utilisateur. La procédure est résumée dans l'algorithme 9.3. Finalement, cette série de points subit une analyse de Fourier. La courbe construite à partir des cinq premières harmoniques est présentée à l'utilisateur, afin qu'il décide si l'approximation ainsi faite est acceptable.

Etape 1	Spécification d'un ensemble de points de contrôle de spline.
Etape 2	Détermination d'une série de points équirépartis sur la courbe spline.
Etape 3	Analyse de Fourier de cette série de points.

Algorithme 9.3: *Détermination des coefficients de Fourier d'après les points de contrôle de splines spécifiés par l'utilisateur.*

Description déclarative de trajectoires : une ouverture Nous avons pensé que l'application des approches de modélisation déclarative des courbes (cf. *Daniel et al.* [DD94] [DL96], voir l'état de l'art §2.5.1) pourrait mettre à disposition du concepteur une interface pour spécifier les formes en langage naturel (qualitatif). Cette spécification déclarative serait ensutie traduite en un certain nombre de « splines probables » qui pourraient elles-mêmes alimenter notre méthode de calcul des coefficients de Fourier. Les perspectives de la synthèse déclarative de mécanismes générateurs de trajectoire sont importantes.

9.7.3 Indexation du cas

Le calcul de l'index pour le cas présent concerne uniquement la forme de trajectoire désirée, les dimensions du mécanisme étant à déterminer. La procédure d'indexation est la même que celle utilisée dans la construction de la base (voir §9.4), le résultat étant une série de coefficients de Fourier normalisés et adimensionnels. Les valeurs minimum et maximum utilisées pour adimensionaliser les coefficients sont celles enregistrées, pour le même coefficient, lors de la création de la base (voir §9.6.2.4) [18]. Cette procédure est effectuée successivement pour chacun des RNA disponibles, puisque les bornes minimum et maximum sont *a priori* différentes.

9.7.4 Obtention de la solution

9.7.4.1 Recherche du cas le plus proche

L'obtention de la solution (ou, dans les termes du *raisonnement à base de cas*, la recherche du cas le plus proche) est réalisée simplement par la propagation à travers le réseau neuronal des valeurs d'entrée, c'est-à-dire des coefficients de Fourier normalisés et adimensionalisés, correspondant à la forme de trajectoire désirée. Les valeurs obtenues en sortie, après une transformation inverse à l'adimensionalisation, représentent les dimensions du mécanisme synthétisé.

En utilisant ces valeurs pour déterminer la position initiale du mécanisme, une simulation cinématique est automatiquement effectuée. La trajectoire décrite par ce mécanisme est ensuite visualisée. Des exemples typiques de résultats seront donnés dans la Figure 9.22 (p. 158).

18. Si les coefficients de la trajectoire désirée dépassent ces valeurs extrêmes, la synthèse n'est pas possible. L'utilisateur en sera averti.

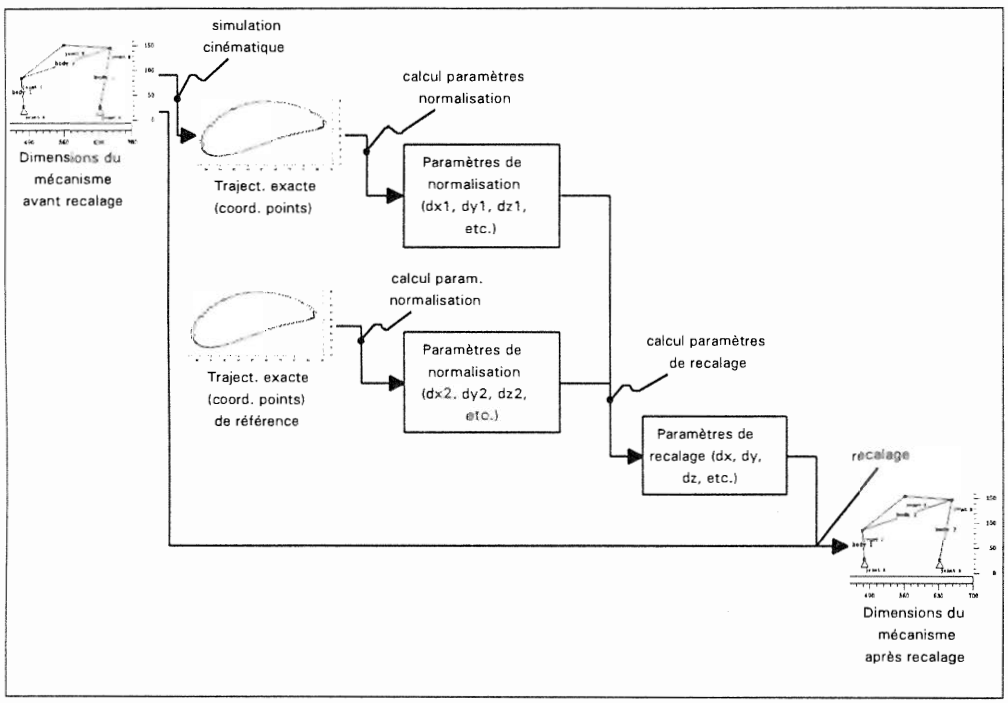


FIG. 9.18 – *Méthodologie de recalage géométrique.*

Translation :	$\Delta x_1 = \mathrm{Re}(a_0)$, $\Delta y_1 = \mathrm{Im}(a_0)$	(cf. éq. 9.12)
Point de départ :	$\Delta\beta_1 = \dfrac{\Phi_1 - \Phi_{-1}}{2}$	(cf. éq. 9.14)
Rotation :	$\Delta\alpha_1 = \dfrac{\Phi_1 + \Phi_{-1}}{2}$	(cf. éq. 9.20)
Direction :	selon signe de $(\lvert a_1\rvert - \lvert a_{-1}\rvert)$	(cf. éq. 9.23)
Echelle :	$\Delta z_1 = \dfrac{1}{\lvert a_1\rvert}$	(cf. éq. 9.24)

TAB. 9.3 – *Paramètres de normalisation pour la trajectoire générée.*

9.7.4.2 Adaptation par recalage géométrique

La dernière étape d'une approche de raisonnement basé sur les cas est l'adaptation du cas retrouvé dans la base de cas au problème actuel à résoudre (cf. l'étape **D**, §B.1). Dans un certain sens, la solution fournie par le réseau neuronal est déjà une adaptation des cas enregistrés dans la base, grâce à l'interpolation. Mais une adaptation supplémentaire peut être obtenue.

Comme nous l'avons montré pour l'optimisation dimensionnelle (cf. §8.2.5), la synthèse de trajectoire présente une particularité : les trajectoires générées sont géométriquement solidaires avec le mécanisme. Autrement dit, une transformation géométrique (translation, rotation, changement d'échelle...) appliquée aux coordonnées des liaisons dans la position initiale du mécanisme a le même effet sur la trajectoire décrite.

La solution donnée par le réseau neuronal peut donc être adaptée en calculant la transformation géométrique optimale qui, appliquée aux coordonnées, rapproche au mieux la forme de la trajectoire générée de celle désirée. La méthode utilisée dans REALISME, et qui lui est spécifique, est présentée dans la Figure 9.18. La méthode diffère de celle utilisée au §8.2.5, car là il s'agissait d'un recalage de trajectoires exactes, exprimées par des liste de points, tandis qu'ici nous devons recaler les trajectoires en termes de forme.

Translation :	$\Delta x = \Delta x_2 - \Delta x_1,\ \Delta y = \Delta y_2 - \Delta y_1$
Point de départ :	$\Delta\beta = \Delta\beta_2 - \Delta\beta_1$
Rotation :	$\Delta\alpha = \Delta\alpha_2 - \Delta\alpha_1$
Echelle :	$\Delta z = \dfrac{\Delta z_2}{\Delta z_1}$

TAB. 9.4 – *Paramètres optimaux de recalage pour les trajectoires modélisées par des coefficients de Fourier.*

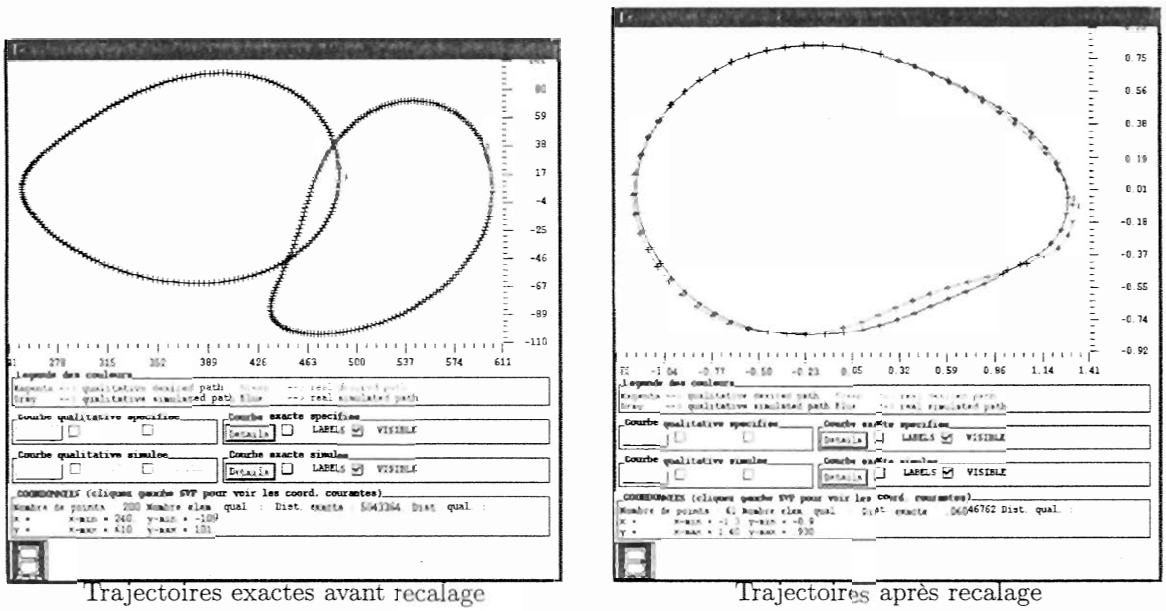


FIG. 9.19 – *Exemple de recalage géométrique de deux trajectoires exactes (à gauche) par la normalisation des deux trajectoires correspondant aux cinq premières harmoniques (à droite).*

Le calcul de la transformation géométrique est basé sur la normalisation des trajectoires générée et désirée. Si les paramètres de normalisation (conformément à la procédure décrite au §9.4.4) pour la courbe générée sont ceux donnés dans le Tableau 9.3 et les paramètres pour la courbe désirée sont Δx_2, Δy_2, $\Delta\beta_2$, $\Delta\alpha_2$, Δz_2, alors les paramètres du recalage se calculent par les relations figurant dans le Tableau 9.4.

Quant à la direction, si pour seulement une des deux trajectoires on change de direction pendant la normalisation, le recalage consistera en un changement de direction pour la trajectoire générée. Un exemple de recalage est donné dans la Figure 9.19.

9.7.4.3 Adaptation par optimisation dimensionnelle

Dans certains cas, la précision d'approximation de la forme de trajectoire désirée correspondant au mécanisme synthétisé par l'utilisation du RNA, éventuellement après recalage géométrique, peut suffire. Si une précision plus grande est nécessaire, le mécanisme synthétisé peut être utilisé comme solution initiale dans un processus d'optimisation dimensionnelle avec indépendance du temps, tel que celui décrit dans le chap. 8.

9.8 Procédures de validation

La méthode de synthèse décrite dans ce chapitre est évidemment sous-optimale. Elle comporte plusieurs approximations lors de la création du jeu de données pour l'apprentissage du réseau

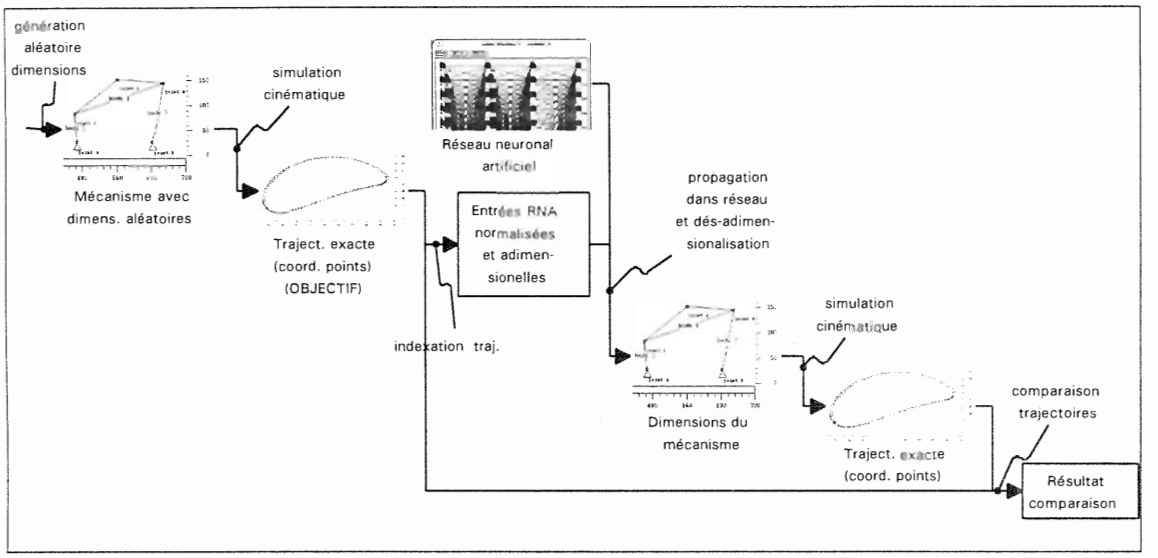


FIG. 9.20 – *Procédure de validation avec garantie de l'existence d'une solution exacte.*

neuronal et la solution fournie par celui-ci est, par définition, approximative. Cependant, quand l'objectif de synthèse est formulé en termes de forme, il est rare qu'une très grande précision soit nécessaire. De toute façon, contraindre d'une manière très précise la forme de la trajectoire désirée ne servirait à rien, puisque, pour une classe finie de morphologies, aucune garantie d'existence d'une solution exacte ne peut exister.

Toutefois, nous avons cherché quelques critères d'estimation de la qualité des solutions fournies par le réseau neuronal. Il ne serait pas raisonnable de comparer une forme de trajectoire désirée quelconque (objectif de la synthèse) à celle de la trajectoire générée par le mécanisme synthétisé, car l'existence d'une solution exacte n'est pas garantie. En revanche, il est possible d'effectuer cette comparaison pour un problème de synthèse pour lequel on connait d'avance soit la solution exacte (en termes de mécanisme), soit une borne minimum de la solution.

9.8.1 Validation pour une solution exacte connue

Dans le premier cas (voir Figure 9.20), pour la morphologie de mécanisme utilisée pour la création de la base de cas, un jeu de dimensions est choisi aléatoirement (à l'intérieur des mêmes intervalles de dimensions que lors de la création de la base). La forme de la trajectoire générée par ce mécanisme est utilisée comme objectif (trajectoire désirée) dans le processus de synthèse. De cette manière, les dimensions du mécanisme choisies aléatoirement au départ constituent en réalité la solution exacte au problème de synthèse, qui comporte les étapes d'indexation, d'utilisation du réseau neuronal artificiel et de désindexation, pour aboutir aux dimensions du mécanisme solution. La forme de la trajectoire obtenue par la simulation cinématique du mécanisme solution est ensuite comparée à celle de la trajectoire désirée de départ pour estimer l'erreur de la méthode.

Cette procédure de validation a l'avantage de permettre une évaluation de la capacité de la base de cas (du RNA) à interpoler les *cas* qui lui ont été présentés durant l'apprentissage (autrement dit, avec le terme usuel dans le domaine des RNA, sa capacité à généraliser à partir des cas). En effet, par un choix aléatoire des dimensions, le mécanisme utilisé au départ a des fortes chances d'être différent de tous ceux utilisés pour l'apprentissage. Le rôle du RNA est d'utiliser les couplages dimensions/forme de trajectoire connus, par exemple $D_1 - T_1$ et $D_3 - T_3$ (voir Figure 9.21) pour

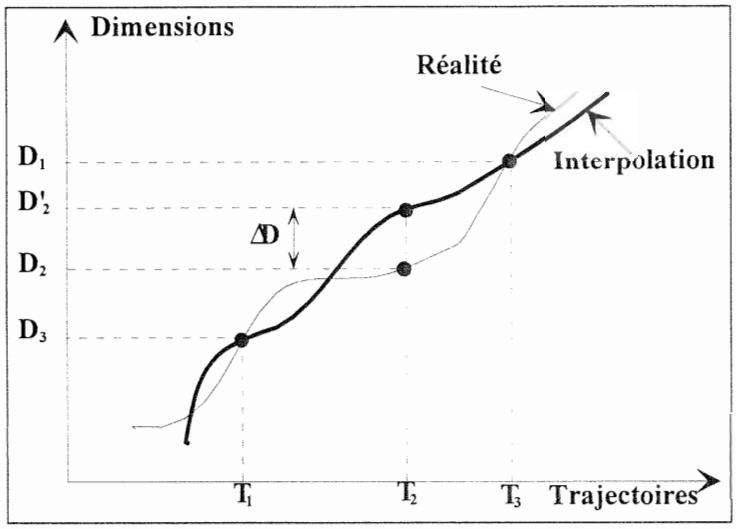


FIG. 9.21 – *Interpolation des couplages entre les dimensions du mécanisme et les formes de trajectoire.*

déterminer une interpolation capable de déduire une valeur approximative D_2' en un point T_2, avec l'erreur intrinsèque d'interpolation ΔD par rapport au point D_2.

Evidemment, ce principe d'interpolation est basé sur la continuité de la relation entre les formes de trajectoires et les dimensions, ce qui, en réalité, n'est vrai que partiellement, à cause des différents problèmes évoqués au §9.5.2.2. Par conséquent, il faut s'attendre à obtenir parfois (selon la région de l'espace dimensionnel) des résultats inacceptables ou très approximatifs. Par exemple, si dans le jeu de données d'apprentissage étaient présents deux couples trajectoire-dimensions (T_1, D_1) et (T_2, D_2), avec $T_1 \approx T_2$ et $D_1 \neq D_2$, alors l'interpolation pour une valeur d'entrée proche de T_1 et de T_2 n'a plus de sens et on obtiendra une solution D_3 différente à la fois de D_1 et de D_2 et telle que $F(D_3) = T_3 \neq T_1$. Lorsque la solution s'avère incorrecte, ce qui est vérifié simplement par une simulation cinématique à partir des dimensions du mécanisme solution, le concepteur peut faire appel à d'autres méthodes de synthèse ou modifier la trajectoire désirée.

Les résultats de cette procédure de validation peuvent être estimés numériquement, sur un certain nombre de tirages aléatoires, par la moyenne de l'erreur ΔD. Cette erreur correspond, en fait, à l'erreur obtenue sur le jeu de test pendant l'apprentissage du réseau (voir §9.6.2.3, équation 9.29), c'est-à-dire environ 5% pour le mécanisme *quatre barres*. Cependant, cette erreur est calculée par rapport aux coefficients de Fourier normalisés, et non pas sur le vrai objectif de synthèse, qui est, en réalité, la forme de la trajectoire. Malheureusement, nous ne disposons pas d'une procédure objective d'évaluation directe de la différence de forme entre deux trajectoires. Nous évaluerons, donc, les résultats principalement d'une façon qualitative et visuelle. La « qualité » des résultats ressort en examinant les représentations superposées de la trajectoire désirée et de la trajectoire décrite par le mécanisme synthétisé (voir quelques exemples représentatifs dans la Figure 9.22). Dans cette figure, apparaît également la distance exprimée par la somme quadratique des différences entre les coefficients de Fourier, pour les deux trajectoires[19].

Ainsi, il y est visible que la forme des trajectoires synthétisées est une approximation de celle de la trajectoire désirée. L'erreur commise prend différentes valeurs pour les différents points de l'espace de solutions.

19. Cette mesure est sujet à discussion, car il est difficile d'estimer l'importance relative des différents coefficients quant à la forme de la trajectoire.

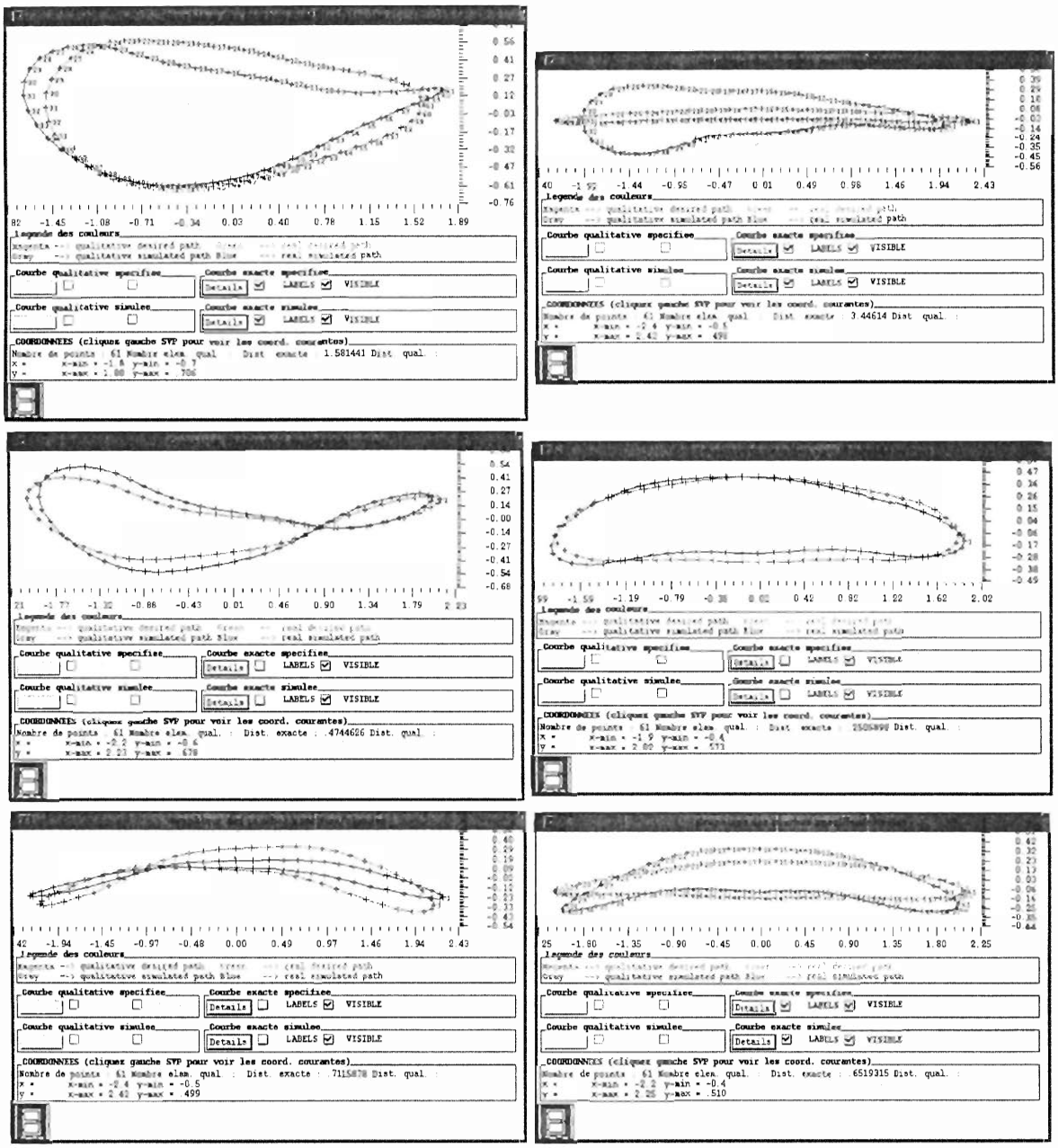


FIG. 9.22 – *Exemples de solutions synthétisées pour sixtrajectoires désirées tirées aléatoirement. La trajectoire désirée et la trajectoire synthétisée sont représentées superposées.*

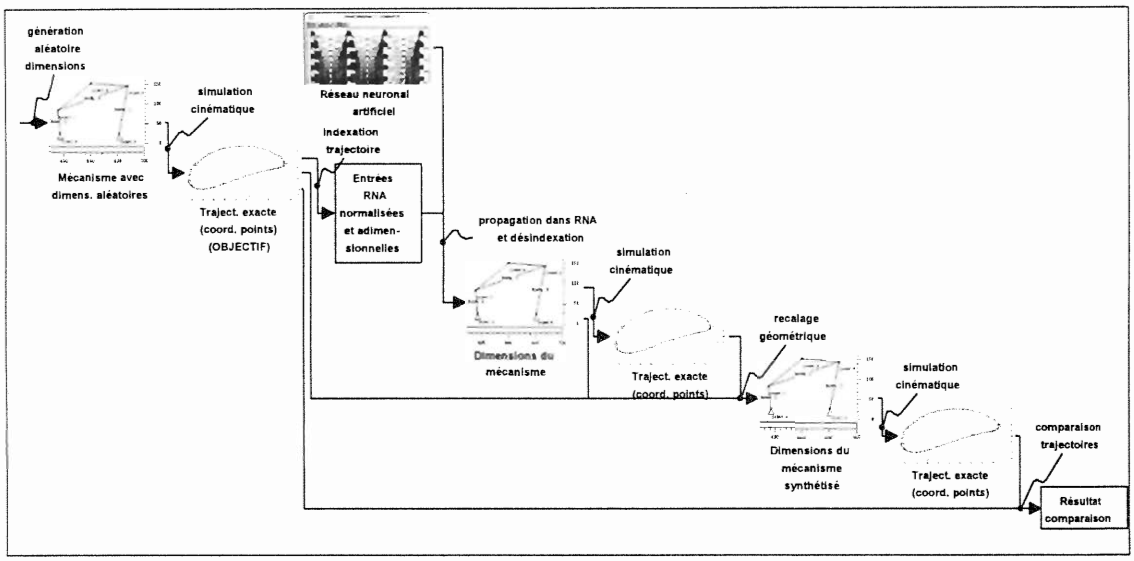


FIG. 9.23 – *Méthodologie de validation avec recalage géométrique.*

Les résultats obtenus peuvent être améliorés par la procédure de recalage géométrique (voir §9.7.4.2) qui permet de superposer au mieux (en termes de forme) les deux trajectoires. Il en résulte une variante de la procédure de validation initiale comportant une phase de *recalage géométrique* (voir Figure 9.23).

9.8.2 Validation pour une borne minimum de l'erreur connue

La procédure (méthodologie) de validation présentée dans le chapitre précédent a l'inconvénient de considérer comme formes de trajectoire désirées uniquement celles pour lesquelles il existe un mécanisme capable de les générer exactement. Mais, en général, les formes spécifiées par le concepteur ne font pas partie de cette catégorie. La comparaison entre la solution synthétisée et la solution exacte connue a priori n'est donc pas représentative des situations rencontrées durant l'application de la méthode et surestime la qualité des résultats.

Cependant, il est impossible d'effectuer une validation à partir d'une forme de trajectoire désirée complètement arbitraire, puisque aucune information sur la solution optimale (globale) n'est disponible. Pour cette raison, une solution de compromis consiste à utiliser comme forme de trajectoire désirée celle d'une trajectoire intermédiaire, ni générée a priori par un mécanisme (une trajectoire générée est appelée dorénavant faisable), ni complètement arbitraire. Pour cela, la forme de la trajectoire désirée peut être obtenue à partir d'une trajectoire faisable, en lui appliquant des « perturbations » (voir Fig. 9.24). En pratique, ces perturbations peuvent être appliquées sur les valeurs des coefficients de Fourier obtenus par l'approximation de la trajectoire faisable.

De cette manière, la distance entre la forme de la trajectoire faisable et la trajectoire générée est une sorte de borne minimum de l'erreur pour le problème de synthèse ayant la forme de la trajectoire perturbée comme objectif. Autrement dit, il existe un jeu de dimensions de mécanisme dont la forme de la trajectoire générée est au moins aussi proche de celle de la trajectoire perturbée que la forme de la trajectoire faisable. Il est donc légitime de comparer la forme de la trajectoire générée par le mécanisme synthétisé avec celle de la trajectoire faisable (voir Figure 9.25).

Le but de cette procédure de validation est de tester la capacité de synthèse sur des formes de trajectoires désirées non-faisables. Cependant, les conclusions restent qualitatives, puisque, entre

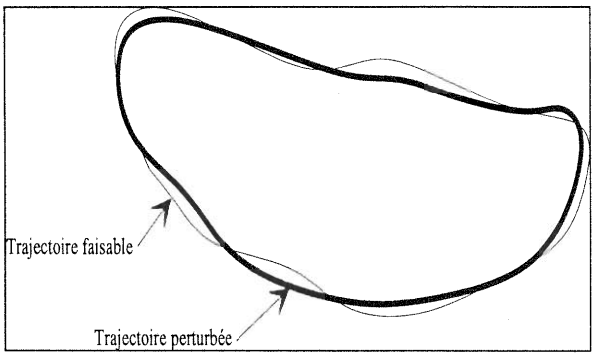


FIG. 9.24 – *Trajectoire désirée obtenue par perturbation d'une trajectoire faisable*

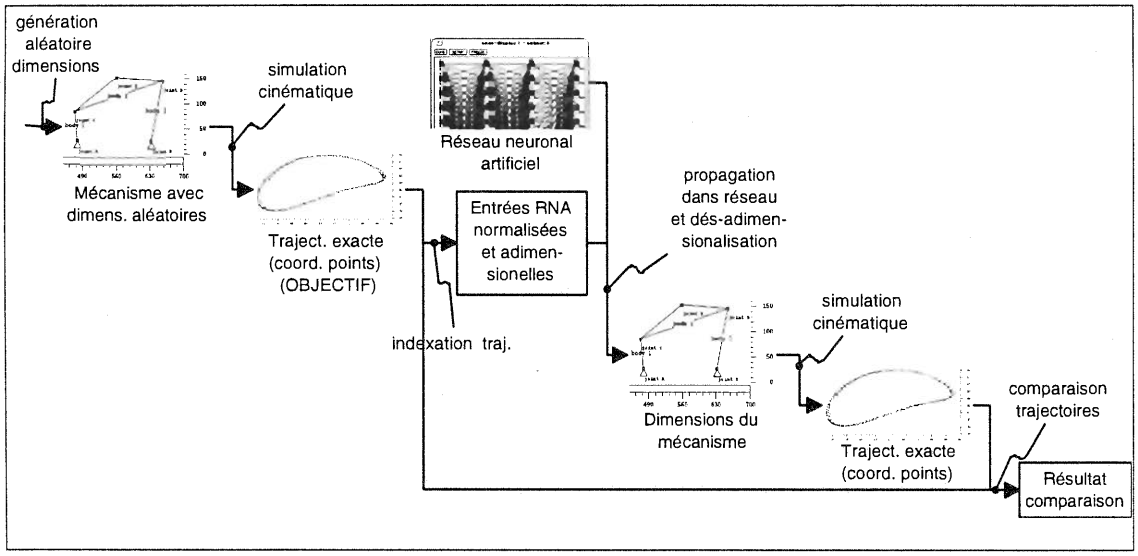


FIG. 9.25 – *Procédure de validation sans garantie d'existence d'une solution.*

autres, la qualité de l'approximation de fonction peut varier fortement selon les régions de l'espace dimensionnel.

9.9 Conclusion

Résumons, pour conclure ce chapitre, les particularités de cette méthode de synthèse, par comparaison avec un catalogue de trajectoires classique (voir Tableau 9.5). Ces particularités sont, essentiellement, la description des trajectoires par les coefficients de Fourier et l'interpolation par un réseau neuronal.

Ainsi, premièrement, le choix de l'analyse harmonique des trajectoires est à même de permettre une synthèse pour des spécifications en termes de forme. Avec un catalogue de trajectoires exactes, ceci n'est possible que par un choix visuel (comme dans *Zhang et al.* [ZNH84]) ou par l'utilisation d'algorithmes très coûteux de comparaison entre les trajectoires sur un critère de forme (comme dans *Kota et Chiou* [KC93]).

Deuxièmement, la « compilation » préalable de la base de cas par une interpolation dans un réseau neuronal est un moyen de concentrer, de réduire énormément la taille de la base. Commençons par répondre à la question : est-ce nécessaire d'utiliser, par exemple, 30000 cas de trajectoires pour

	Base de cas par catalogue de trajectoires exactes	Base de cas par Fourier + RNA
Synthèse sur critère de forme ou avec indépendance du temps	très coûteuse (ou par interpolation visuelle)	oui (directe, naturelle)
Taille	croissance linéaire avec le nombre de trajectoires	taille très réduite et indép. du n° de cas
Limite du nombre de cas	limitée par la taille en en phase d'utilisation	limitée par la taille en phase d'apprentissage
Précision d'enregistrement des trajectoires	bonne	approximative

TAB. 9.5 – *Comparaison de la synthèse par catalogue de trajectoires exactes et par base de cas utilisant l'analyse de Fourier et l'interpolation avec un réseau neuronal.*

le mécanisme quatre barres ? Un raisonnement simple nous prouve que oui. En effet, même en réduisant le nombre de paramètres dimensionnels à cinq (comme nous l'avons fait), on constate que pour chaque paramètre nous pouvons prendre en compte seulement $30000^{1/5} = 7,86$ valeurs distinctes ! Il est donc, en pratique, impossible d'avoir des cas répartis, d'une façon fine, sur l'ensemble de l'espace dimensionnel. Autrement dit, même si la taille d'une base de cas de 30000 trajectoires exactes n'était pas trop grande pour une utilisation en conception, ce nombre serait trop faible pour couvrir vraiment l'ensemble des trajectoires possibles. En revanche, l'interpolation, même si elle est une source potentielle d'erreurs, dûes essentiellement à la nature de la fonction à approximer, permet non seulement de comprimer cette base de cas dans un petit nombre de coefficients, mais aussi d'obtenir des solutions approchées, intermédiaires entre celles enregistrées dans la base de cas initiale. C'est un avantage important, compte tenu de la grande sensibilité des trajectoires à une petite variation d'une dimension. En revanche, l'application de notre méthode à des mécanismes plus complexes, comportant donc un plus grand nombre de paramètres dimensionnels, si elle est possible en principe, comporte des difficultés d'ordre pratique, liées au nombre de cas nécessaires. En effet, avec seulement deux paramètres de plus, le nombre de valeurs distinctes, toujours pour 30000 cas, diminue fortement ($30000^{1/7} = 4,36$). Il n'est néanmoins pas impossible de générer un nombre plus grand de cas.

Comparons maintenant la taille d'une base de cas comprenant 30000 trajectoires exactes avec la taille du réseau neuronal (voir Tableau 9.6). Pour une base de cas enregistrant des trajectoires exactes, nous pouvons nous contenter de 36 points par trajectoire. Pour la base de cas nécessaire pour réaliser l'apprentissage du RNA, nous utilisons 200 points par trajectoire, afin que l'approximation par l'analyse harmonique soit bonne. Cette base aura, initialement, une taille supérieure au catalogue de trajectoires exactes. Mais, une fois les coefficients de Fourier calculés (ce qui peut être fait au fur et à mesure, pour chaque trajectoire), la dimension de la base de cas se réduit fortement. Une trajectoire modélisée par ses 5 premières harmoniques prend 4 fois moins de place que les coordonnées de 36 points. Ensuite, après apprentissage, l'information contenue dans la base de cas est concentrée, moyennant l'interpolation, dans les coefficients de poids des connexions du RNA, en nombre très limité et indépendant du nombre de cas utilisé pour l'apprentissage. Ceci est un avantage important, car de cette manière il est possible d'augmenter presque sans limite le nombre de cas pouvant être pris en compte.

Pour finir, signalons l'intérêt de l'utilisation, dans REALISME, de la génération des cas de trajectoires par une simulation cinématique utilisant l'approche systémique multipolaire (cf. chap. 4). Vu le très grand nombre de simulations nécessaires, l'efficacité de la méthode de simulation devient un facteur critique. Sur un simulateur classique, la réalisation d'un grand nombre de si-

	Base de cas par catalogue de trajectoires exactes	**Base de cas par Fourier + RNA**
Nombre de cas	30000	30000
Nombre de points par traj.	36	200
Taille totale pour stocker les trajectoires	$2,16 \times 10^6$ $= 2 \times 36 \times 30000$	12×10^6 $= 2 \times 200 \times 30000$
Taille totale pour stocker trajectoires et dimensions	$2,31 \times 10^6$ $= 2,16 \times 10^6 + 5 \times 30000$	12.15×10^6 $= 12 \times 10^6 + 5 \times 30000$
Taille après analyse harmonique	–	$0,51 \times 10^6 = 17 \times 30000$
Taille totale à l'utilisation	$2,31 \times 10^6$	968

TAB. 9.6 – *Comparaison des deux approches de synthèse selon la taille de la base de cas.*

mulations pour un mécanisme gardant la même morphologie mais avec des dimensions différentes nécessiterait de refaire, à chaque fois, l'étape de mise en équations. Le caractère *paramétrique* de l'approche systémique permet de remplacer cette étape par la détermination du *schéma multipolaire* (cf. §4.1.2.1), jouant un rôle à peu près équivalent, mais unique pour une même morphologie. De plus, l'utilisation, la plupart des fois, de solutions explicites contribue à rendre les temps de calcul acceptables.

Chapitre 10

Synthèse basée sur les cas : l'approche par trajectoires qualitatives

10.1 Introduction

La *cinématique qualitative* développée par Yannou (*Yannou et Vasiliu* [YV95] [YVM95]), présentée au chap. 5, a déjà été intégrée dans REALISME pour fournir une méthode d'optimisation dimensionnelle qualitative (cf. §8.3). Dans le présent chapitre, la modélisation des trajectoires spécifique à la cinématique qualitative est employée dans le cadre d'une méthode de synthèse par l'utilisation d'un catalogue de trajectoires qualitatives, dans le cadre d'une approche de *raisonnement à base de cas* (cf. l'état de l'art au §2.5.2 et une introduction générale dans l'annexe B).

En effet, comme cela a déjà été montré au chapitre précédent, une base de cas pour la synthèse des générateurs de trajectoire nécessite la prise en compte d'un très grand nombre de cas, afin que cette base soit représentative de la complexité de la fonction reliant les trajectoires aux dimensions. Par conséquent, les cas doivent être codifiés et indexés d'une manière efficace, c'est-à-dire à la fois compacte et représentative. Or, l'enregistrement pur et simple d'une liste de points (trajectoire *exacte*), obtenue par simulation cinématique, conduit à une taille énorme de la base de cas et rend coûteuse la comparaison avec le cas présent. De plus, la représentation par une liste de points est souvent très éloignée d'une représentation fonctionnelle, notamment dans le cas où le concepteur s'intéresse à certaines caractéristiques de forme des trajectoires.

D'une façon alternative – mais aussi complémentaire – à la méthode d'indexation par coefficients de Fourier normalisés présentée au chap. 9, une des solutions possibles est l'utilisation de *clés qualitatives* basées sur la modélisation qualitative des trajectoires, c'est-à-dire l'approximation des trajectoires exactes en arcs de cercle (pour une discussion comparative des avantages et des désavantages de ces deux méthodes, voir §10.7). L'objet du présent chapitre est la présentation d'une méthode de synthèse des générateurs de trajectoires utilisant une base de cas qualitatifs.

Pour bien comprendre la portée de cette solution, il faut la comparer à l'utilisation d'un catalogue – papier ou informatisé – de trajectoires. L'enregistrement de clés qualitatives à la place des trajectoires données par des suites de points permet de réduire la taille de la base de cas, ou, pour une même taille, d'augmenter le nombre de cas. Ce caractère compact est le résultat, d'une part, de la discrétisation des trajectoires par arcs de cercles (cf. §5.2) et, d'autre part, de la maîtrise dont ont dispose, pendant le remplissage de la base, du degré de couverture des formes qualitatives de trajectoires (cf. §10.5). En effet, grâce au caractère discret des clés qualitatives, il

est facile de comparer une clé-candidate à rentrer dans la base avec les clés déjà rentrées afin de ne pas l'avoir en double. Cette comparaison peut se faire par accès direct, sans calcul, alors que pour des trajectoires définies par des suites de points, la mesure de similitude devrait se faire par un calcul de distance entre la trajectoire désirée et toutes les trajectoires de la base de cas, ce qui est inconcevable.

L'avantage déterminant d'une base de clés qualitatives est la possibilité d'exprimer, lors de l'utilisation de la base, des requêtes qualitatives, pour aborder le domaine prometteur de la préconception déclarative. Avec un catalogue informatisé contenant des trajectoires exactes, il ne sera pas possible, par exemple, de rechercher une trajectoire comportant une bosse de grand rayon, suivie par une de rayon moyen etc.

La force de cette approche, dont on commence seulement à explorer les possibilités, est de pouvoir mettre une « algorithmique intelligente » entre la ou les spécifications de préconception et la détermination d'un jeu de clés avec des coefficients de vraisemblance.

L'approche permet la synthèse des trajectoires ouvertes avec indépendance du temps. Elle semble extensible également aux trajectoires avec dépendence du temps.

10.2 Modélisation et indexation des cas

10.2.1 Définition du cas qualitatif

Dans le modèle de cinématique qualitative (présenté au chap. 5), les trajectoires sont approximées par des arcs de cercle et des segments. Afin d'avoir une représentation plus homogène, pour la présente application sont considérés uniquement des arcs de cercles, les parties de trajectoires quasi linéaires étant modélisées par des arcs d'un grand rayon.

La valeur d'un cas est constituée par l'association[1] entre quatre éléments :

- la trajectoire qualitative normalisée (les valeurs des paramètres de chaque arc sont stockés),
- la précision d'approximation (l'erreur maximum commise),
- la morphologie du mécanisme (représentée par un symbole),
- les dimensions normalisées du mécanisme.

La clé qualitative d'indexation d'un cas est une séquence comprenant[2] :

- le classement des arcs par des symboles qualitatifs portant sur la longueur
- le classement des arcs par des symboles qualitatifs portant sur le rayon,
- le sens de parcours de l'arc.

Par rapport aux trajectoires qualitatives normalisées, la clé qualitative est une représentation située à un niveau d'abstraction plus élevé, plus « qualitatif » (voir Figure 10.1). Par conséquent, elle permet une discrimination plus rapide des cas de la base, mais en même temps n'est pas assez précise pour servir, seule, de critère de choix de la similitude entre la spécification et les cas de la base. C'est pour cette raison que, sous la même clé qualitative, seront stockées plusieurs valeurs de cas, correspondant à un certain nombre de trajectoires qualitatives différentes et même à des morphologies différentes (voir les détails au §10.3.3).

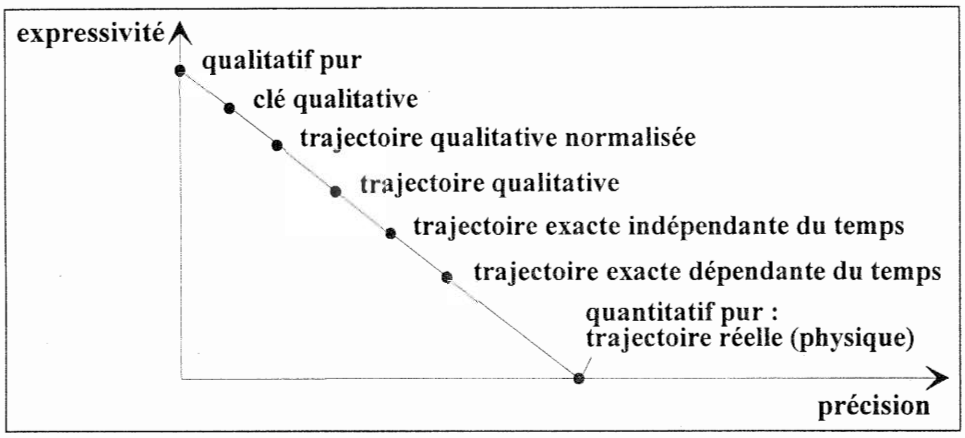


FIG. 10.1 – *Echelle des niveaux de représentation des trajectoires générées par un mécanisme.*

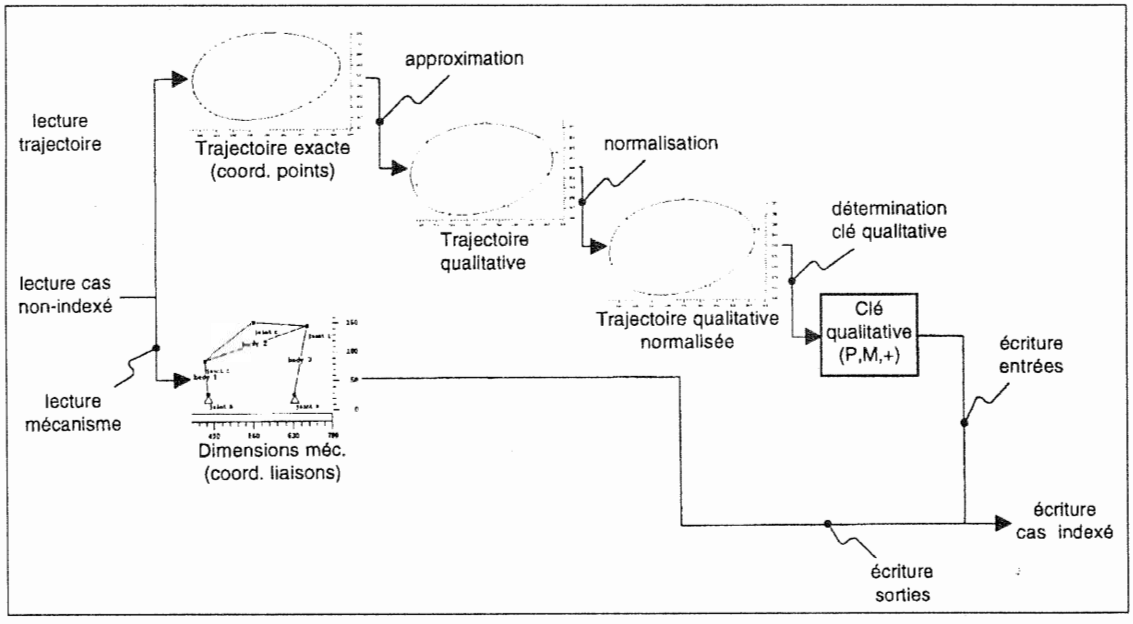


FIG. 10.2 – *Méthodologie d'indexation des cas qualitatifs.*

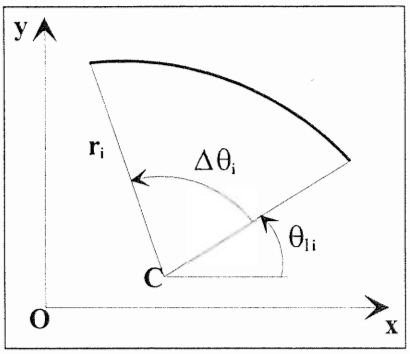


FIG. 10.3 – *Paramétrage d'un arc i de la trajectoire qualitative.*

10.2.2 Méthode d'indexation

La méthode d'indexation utilisée dans REALISME, présentée dans la Figure 10.2, consiste en la détermination, pour chaque morphologie et jeu de dimensions d'un mécanisme, de plusieurs clés d'indexation qualitatives correspondante. Premièrement, une trajectoire générée exacte est déterminée par simulation cinématique. Ensuite, plusieurs approximations de cette trajectoire en arcs de cercle, pour des précisions différentes, donnent des trajectoires qualitatives correspondantes. Après un processus de normalisation visant à s'abstraire des transformations géométriques affines, ces trajectoires vont aller se « ranger » sous différentes clés d'indexation. De cette manière, la base de cas sera multi-résolution et permettra de fournir des solutions dans une gamme plus large de précisions demandées. Ceci correspond au principe de la conception qualitative de permettre au concepteur d'exprimer des requêtes approximatives, selon les nécessités.

Les détails sur les différentes étapes de la méthode seront présentés dans les sous-chapitres suivants.

10.3 Indexation des trajectoires

10.3.1 Paramétrage des arcs

Le paramétrage des arcs de la trajectoire qualitative est indiqué dans la Figure 10.3. Les paramètres dimensionnels d'un arc i ($i = 1, \ldots, n$, où n est le nombre d'arcs de la trajectoire qualitative) sont r_i = le rayon, θ_{1_i} = l'angle initial (mesuré en sens trigonométrique, par rapport au centre C), $\Delta\theta_i$ = l'amplitude angulaire. (L'angle final θ_{2_i} peut être calculé par la relation $\theta_{2_i} = \theta_{1_i} + \Delta\theta_i$.) Si nous notons L_i la longueur curviligne de l'arc i, L_{tot} la longueur totale de la trajectoire qualitative et L_{moy} la longueur moyenne des arcs, nous avons les relations suivantes :

$$\begin{aligned} L_i &= r_i\,|\Delta\theta_i| \\ L_{\text{tot}} &= \sum_{i=1}^{n} L_i \\ L_{\text{moy}} &= \frac{L_{\text{tot}}}{n} \end{aligned} \tag{10.1}$$

1. Implémentée informatiquement comme un objet avec des champs.
2. Cette méthode d'indexation n'est pas la seule possible. Voir à ce sujet §10.3.3.

10.3.2 Normalisation de la trajectoire qualitative

L'espace des paramètres dimensionnels et des trajectoires étant extrêmement vaste, il est nécessaire de réduire cette complexité, pour avoir des cas représentatifs d'un maximum de structures. La modélisation qualitative des trajectoires est en soi une réponse à ce problème, car une trajectoire qualitative se représente de manière succincte. En plus, elle permet de s'abstraire de la répartition particulière des points sur la trajectoire exacte, en discriminant les cas par une logique de forme qualitative des trajectoires. Cette logique est très pertinente dans les premières phases de la conception ou lorsque les spécifications n'ont pas lieu d'être plus précises.

Une réduction supplémentaire de la complexité est ensuite obtenue par la *normalisation*, en s'abstrayant ainsi des transformations géométriques affines. La normalisation modifie un cas de manière qu'il soit représentatif d'une classe entière de trajectoires, ayant les mêmes caractéristiques de forme qualitative. Les objectifs de la normalisation sont les mêmes que pour la normalisation des trajectoires représentées par des coefficients de Fourier (cf. §9.4.4).

La procédure spécifique de normalisation est donnée dans l'Algorithme 10.1 et sera détaillée dans les sous-chapitres suivants.

Etape 1 Détermination du centre de gravité (au sens des longueurs d'arcs).
Etape 2 Détermination des directions principales d'inertie, des valeurs et des vecteurs propres.
Etape 3 Normalisation en translation (on redéfinit l'origine du repère au centre de gravité).
Etape 4 Normalisation en rotation (on oriente la trajectoire en fonction du repère principal d'inertie).
Etape 5 Normalisation en symétrie.
Etape 6 Normalisation en échelle.
Etape 7 Normalisation du point de départ et de la direction.

Algorithme 10.1: *Algorithme pour la normalisation d'une trajectoire qualitative.*

10.3.2.1 Centre de gravité d'un arc

Avec des relations classiques, nous calculons tout d'abord les coordonnées (x_{g_i}, y_{g_i}) du centre de gravité d'un arc i. En considérant la masse proportionnelle à la longueur curviligne de l'arc L_i et en notant avec θ_{m_i} l'angle moyen qui référence l'axe de symétrie de l'arc, nous avons les formules :

$$x_{g_i} = x_{c_i} + r_i \frac{\sin\frac{\Delta\theta_i}{2}}{\frac{\Delta\theta_i}{2}} \cos\theta_{m_i} \tag{10.2}$$

$$y_{g_i} = y_{c_i} + r_i \frac{\sin\frac{\Delta\theta_i}{2}}{\frac{\Delta\theta_i}{2}} \sin\theta_{m_i} \tag{10.3}$$

$$\text{avec } \theta_{m_i} = \theta_{1_i} + \frac{\Delta\theta_i}{2} \tag{10.4}$$

10.3.2.2 Centre de gravité d'une trajectoire qualitative

Connaissant maintenant le centre de gravité et la longueur de chaque arc i, il est possible de calculer les coordonnées (x_G, y_G) du centre de gravité et la longueur L_{tot} de la trajectoire qualitative dans son ensemble :

$$L_{\text{tot}} = \sum_{i=1}^{n} L_i \tag{10.5}$$

$$x_G = \frac{\sum_{i=1}^{n} L_i x_{g_i}}{L_{\text{tot}}} \tag{10.6}$$

$$y_G = \frac{\sum_{i=1}^{n} L_i y_{g_i}}{L_{\text{tot}}} \tag{10.7}$$

10.3.2.3 Vecteurs propres au repère O

Si nous notons P un point de l'arc A, avec dl l'élément infinitésimal de longueur de l'arc, nous pouvons exprimer la matrice d'inertie de cet arc dans le repère (O, x, y) :

$$\overline{\overline{I}}_{(O,S)} = \begin{pmatrix} \int_{P\in A} y^2 \, dl(P) & -\int_{P\in A} xy \, dl(P) \\ -\int_{P\in A} xy \, dl(P) & \int_{P\in A} x^2 \, dl(P) \end{pmatrix} \tag{10.8}$$

Pour un arc de cercle dont les coordonnées du centre sont (x_C, y_C), nous avons (en fonction cette fois-ci d'un paramètre angulaire θ) :

$$\begin{aligned}
\int_{P\in A} y^2 \, dl(P) &= \int_{\theta_1}^{\theta_2} (y_c + r\sin\theta)^2 r \, d\theta \\
&= r\int_{\theta_1}^{\theta_2} \left((y_C^2 + 2y_c r\sin\theta + \frac{r^2}{2}(1-\cos 2\theta) \right) d\theta \\
&= r\left(y_C^2(\theta_2-\theta_1) + 2y_c r(-\cos\theta)_{\theta_1}^{\theta_2} + \frac{r^2}{2}(\theta - \frac{1}{2}\sin 2\theta)_{\theta_1}^{\theta_2} \right) \\
&= r\left((\theta_2-\theta_1)(y_C^2 + \frac{r^2}{2}) + 2y_c r(\cos\theta_1 - \cos\theta_2) \right. \\
&\quad \left. -\frac{r^2}{4}(\sin 2\theta_2 - \sin 2\theta_1) \right) \qquad (10.9) \\
\int_{P\in A} x^2 \, dl(P) &= r\left((\theta_2-\theta_1)(x_C^2 + \frac{r^2}{2}) + 2x_c r(\sin\theta_2 - \sin\theta_1) \right. \\
&\quad \left. +\frac{r^2}{4}(\sin 2\theta_2 - \sin 2\theta_1) \right) \qquad (10.10) \\
\int_{P\in A} xy \, dl(P) &= \int_{\theta_1}^{\theta_2} (x_c + r\cos\theta)(y_c + r\sin\theta) r \, d\theta \\
&= r\left(x_c y_c(\theta_2-\theta_1) + r y_c(\sin\theta_2 - \sin\theta_1) \right. \\
&\quad \left. -r x_c(\cos\theta_2 - \cos\theta_1) - \frac{r^2}{4}(\cos 2\theta_2 - \cos 2\theta_1) \right) \qquad (10.11)
\end{aligned}$$

Pour la trajectoire qualitative dans son ensemble, il faut sommer les intégrales 10.9, 10.10 et 10.11 pour les n arcs :

$$\overline{\overline{I}}_{(O,\Sigma)} = \sum_{i=1}^{n} \overline{\overline{I}}_{(O,S_i)}$$

10.3.2.4 Vecteurs et valeurs propres au centre de gravité G

Si on change maintenant le repère O avec le repère G, nous avons la relation suivante, provenant du théorème d'Huyghens généralisé :

$$\overline{\overline{I}}_{(G,\Sigma)} = \overline{\overline{I}}_{(O,\Sigma)} - \begin{pmatrix} My_G^2 & -Mx_Gy_G \\ -Mx_Gy_G & Mx_G^2 \end{pmatrix} \tag{10.12}$$

Diagonalisons cette matrice d'inertie de manière à obtenir les valeurs propres λ_1 et λ_2 et les vecteurs propres V_1 et V_2 qui nous donneront les directions principales d'inertie utiles pour la normalisation en rotation :

$$\begin{cases} \overline{\overline{I}}_{(G,\Sigma)} V_1 = \lambda_1 V_1 \\ \overline{\overline{I}}_{(G,\Sigma)} V_2 = \lambda_2 V_2 \end{cases} \tag{10.13}$$

Posons une forme simplifiée de $\overline{\overline{I}}_{(G,\Sigma)} = \begin{pmatrix} A & -C \\ -C & B \end{pmatrix}$. Le calcul des vecteurs propres revient à :

$$\begin{aligned} \begin{bmatrix} A-\lambda & -C \\ -C & B-\lambda \end{bmatrix} &= 0 \Rightarrow \\ \lambda^2 - (A+B)\lambda + AB - C^2 &= 0 \end{aligned} \tag{10.14}$$

Nous avons donc les solutions :

$$\lambda_{1,2} = \frac{(A+B) \pm \sqrt{(A+B)^2 - 4(AB - C^2)}}{2} \tag{10.15}$$

Calculons maintenant les vecteurs propres :

$$\begin{pmatrix} A & -C \\ -C & B \end{pmatrix} \begin{pmatrix} v_{11} \\ v_{12} \end{pmatrix} = \begin{pmatrix} \lambda_1 v_{11} \\ \lambda_1 v_{12} \end{pmatrix} \Rightarrow \tag{10.16}$$

$$v_{12} = \frac{(A-\lambda_1)v_{11}}{C} \tag{10.17}$$

En posant la condition que les vecteurs soient normés, nous obtenons les vecteurs propres :

$$\begin{aligned} & v_{i1}^2 + v_{i2}^2 = 1 \\ & v_{i1} = \frac{1}{\sqrt{1 + \frac{(A-\lambda_i)^2}{C^2}}} \\ & v_{i2} = \frac{(A-\lambda_i)}{C\sqrt{1 + \frac{(A-\lambda_i)^2}{C^2}}} \\ & \text{pour } i = 1, 2 \end{aligned} \tag{10.18}$$

Si la relation $\lambda_2 \geq \lambda_1$ n'est pas respectée, alors il faut inverser les valeurs de λ_1 et λ_2, de manière à avoir une trajectoire plutôt « allongée » le long du vecteur V_1.

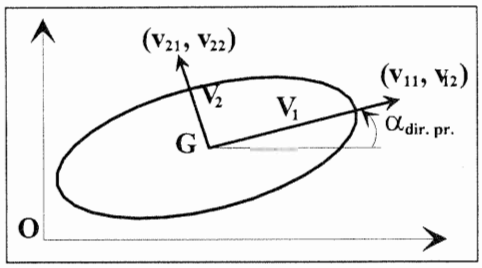


FIG. 10.4 – *Vecteurs propres pour la trajectoire.*

10.3.2.5 Abstraction de la translation

Le centre de gravité de la trajectoire qualitative (le point G, dont les coordonnées x_G, y_G ont été déterminées par les équations 10.6 et 10.7) est déplacé à l'origine (le point O, de coordonnées 0,0). Pour cela, chaque arc subit une translation caractérisée par les paramètres :

$$\begin{aligned} \Delta x &= -x_G \\ \Delta y &= -y_G \end{aligned} \tag{10.19}$$

Cette transformation se traduit pour chaque arc par la translation du centre C (les autres paramètres restent inchangés) :

$$\begin{aligned} x_c &\leftarrow x_c + \Delta x \\ y_c &\leftarrow y_c + \Delta y \end{aligned} \tag{10.20}$$

10.3.2.6 Abstraction de la rotation

Pour absorber les équivalences en rotation, la trajectoire qualitative doit subir une rotation, qui la ramène dans une position normalisée. Cette position est celle pour laquelle le grand axe d'inertie est parallèle à l'axe des abscisses. A partir des vecteurs propres calculés par les équations 10.18, l'angle du grand axe d'inertie par rapport à l'axe des abscisses (voir Figure 10.4) est calculé par la formule suivante :

$$\alpha_{\text{dir. pr.}_1} = \tan^{-1} \frac{v_{12}}{v_{11}} \tag{10.21}$$

La normalisation en rotation consistera en une rotation d'angle $\Delta\alpha$ de l'ensemble de la trajectoire qualitative, avec :

$$\Delta\alpha = -\alpha_{\text{dir. pr.}} = -\tan^{-1} \frac{v_{12}}{v_{11}} \tag{10.22}$$

On constate que cette rotation se fait à π près, il faudra donc faire un choix ; celui-ci sera déterminé par une absorption de symétrie (cf. §10.3.2.7).

Pour effectuer la rotation d'angle $\Delta\alpha$ de la trajectoire, chaque arc doit subir une rotation du même angle. Le centre de l'arc étant situé à l'angle α (voir Figure 10.5), la transformation sera :

$$\alpha = \tan^{-1} \frac{y_C}{x_C} \tag{10.23}$$

$$\text{puis } \alpha \leftarrow \alpha + \Delta\alpha \tag{10.24}$$

Puisque la distance $\left\|\overrightarrow{OC}\right\| = \sqrt{x_C^2 + y_C^2}$ reste inchangée, la rotation se traduit pour chaque arc en un déplacement du centre C et une modification de l'angle initial θ_1 [3] :

3. Afin de pouvoir par la suite comparer des valeurs d'angle, il est nécessaire de ramener entre 0 et 2π les valeurs qui sortent de cet intervalle après transformation.

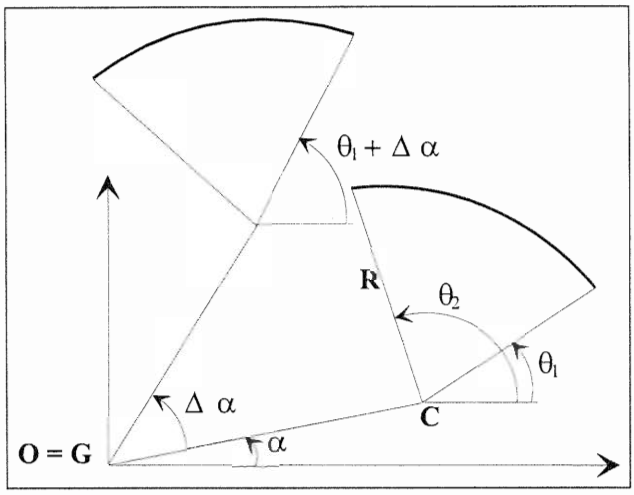


FIG. 10.5 – *Rotation d'un arc de la trajectoire qualitative.*

$$x_c = \left\|\overrightarrow{OC}\right\| \cdot \cos\alpha \tag{10.25}$$

$$y_c = \left\|\overrightarrow{OC}\right\| \cdot \sin\alpha \tag{10.26}$$

$$\theta_1 \leftarrow \theta_1 + \Delta\alpha \tag{10.27}$$

10.3.2.7 Abstraction des symétries

Afin d'absorber les équivalences des trajectoires moyennant les symétries en x et en y[4], le critère de normalisation choisi est le positionnement d'un point particulier de la trajectoire dans le quadrangle en haut à droite (par rapport au repère ayant son origine au centre de gravité). Ce point particulier a été choisi arbitrairement comme étant le point de la trajectoire le plus éloigné par rapport au centre de gravité de la trajectoire qualitative (de coordonnées x_G, y_G).

La détermination du point le plus éloigné du centre de gravité est réalisée d'abord pour chaque arc séparément ; finalement, le plus éloigné parmi ces points est identifié. Pour chaque arc, l'angle α entre l'horizontale et le vecteur $\overrightarrow{GC}$ est déterminé :

$$\alpha = \tan^{-1}\frac{y_C - y_G}{x_C - x_G} \tag{10.28}$$

La seule situation pour laquelle le point le plus éloigné n'est pas une des extrémités de l'arc est celle quand $\theta_1 < \alpha < (\theta_1 + \Delta\theta)$, c'est-à-dire quand le vecteur $\overrightarrow{GC}$ intersecte l'arc en un point situé, par rapport à C, du côté inverse de G (voir Figure 10.6.a). Dans ce cas, le point le plus éloigné de G a les coordonnées :

$$\begin{aligned} x_{\text{dist. max}} &= (R + \sqrt{(x_C - x_G)^2 + (y_C - y_G)^2})\cos\alpha \\ y_{\text{dist. max}} &= (R + \sqrt{(x_C - x_G)^2 + (y_C - y_G)^2})\sin\alpha \end{aligned} \tag{10.29}$$

Dans tous les autres cas (deux exemples sont donnés dans les Figures 10.6.b et 10.6.c), le point le plus éloigné est soit la première extrémité de l'arc (dans l'ordre de parcours), de coordonnées (x_1, y_1), soit la deuxième, de coordonnées (x_2, y_2). Pour savoir laquelle des deux est la plus éloignée, on calcule les distances :

4. On constate qu'une rotation de π est équivalente à deux symétries en x et en y.

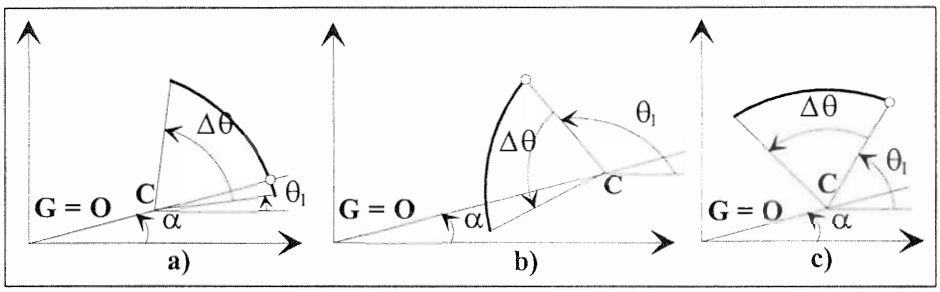


FIG. 10.6 – *Exemples de positions d'un arc, correspondant à un calcul différent de la distance maximale par rapport à l'origine. a) : $\theta_1 < \alpha < (\theta_1 + \Delta\theta)$; b) et c) : deux autres cas. (Le point le plus éloigné de l'origne est marqué par un petit cercle.)*

Cas de symétrie	Transformations
Symétrie en x ($x_{\text{dist. max}} < 0$, $y_{\text{dist. max}} > 0$)	$x_C \leftarrow -x_C$ $\theta_1 \leftarrow \pi - \theta_2$ $\theta_2 \leftarrow \theta_1 + \Delta\theta$
Symétrie en y ($x_{\text{dist. max}} > 0$, $y_{\text{dist. max}} < 0$)	$y_C \leftarrow -y_C$ $\theta_1 \leftarrow 2\pi - \theta_2$ $\theta_2 \leftarrow \theta_1 + \Delta\theta$
Symétrie en x et y ($x_{\text{dist. max}} < 0$, $y_{\text{dist. max}} < 0$)	$x_C \leftarrow -x_C$ $y_C \leftarrow -y_C$ $\theta_1 \leftarrow \pi + \theta_1$ $\theta_2 \leftarrow \theta_1 + \Delta\theta$

TAB. 10.1 – *Transformations par symétrie pour chaque arc.*

$$\begin{aligned} d_1 &= \sqrt{(x_1 - x_G)^2 + (y_1 - y_G)^2} \\ d_2 &= \sqrt{(x_2 - x_G)^2 + (y_2 - y_G)^2} \end{aligned} \tag{10.30}$$

Si $d_1 > d_2$, le point le plus éloigné est la première extrémité de l'arc :

$$\begin{aligned} x_{\text{dist. max}} &= x_1 \\ y_{\text{dist. max}} &= y_1 \end{aligned} \tag{10.31}$$

sinon il s'agit de l'autre extrémité :

$$\begin{aligned} x_{\text{dist. max}} &= x_2 \\ y_{\text{dist. max}} &= y_2 \end{aligned} \tag{10.32}$$

Dans le cas singulier $d_1 = d_2$, la première extrémité est prise en considération. Une fois le point le plus éloigné connu, la trajectoire va subir une transformation par symétrie afin de ramener ce point dans le premier quadrangle (par rapport au repère O). Pour chaque arc, les transformations par symétrie sont indiquées dans le Tableau 10.1.

10.3.2.8 Abstraction de l'échelle

Pour absorber les différences d'échelle, une taille normalisée de la trajectoire qualitative doit être choisie. Pour cela, on considère que la distance moyenne entre la trajectoire et le centre de gravité, d_{moy}, est représentative de l'échelle. Cette distance moyenne peut être calculée au sens quadratique par la formule :

$$d_{\text{moy}}^2 L_{\text{tot}} = \int_{P \in A} (x^2 + y^2)\ dl(P), \text{ soit} \tag{10.33}$$

$$d_{\text{moy}} = \sqrt{\frac{\lambda_1 + \lambda_2}{L_{\text{tot}}}} \tag{10.34}$$

En choisissant une valeur normalisée conventionnelle pour cette distance :

$$d_{\text{moy}_{\text{norm}}} = 100 \tag{10.35}$$

le facteur de transformation d'échelle z pour la trajectoire caractérisée par une distance moyenne d_{moy} est déterminé par la relation :

$$z = \frac{d_{\text{moy}_{\text{norm}}}}{d_{\text{moy}}} \tag{10.36}$$

Le facteur z est utilisé ensuite pour obtenir la transformation d'échelle qui permet d'avoir pour la trajectoire transformée la distance moyenne donnée par 10.35. Cette transformation se traduit pour chaque arc par les transformations suivantes :

$$\begin{array}{lcl} x_C & \leftarrow & z \cdot x_C \\ y_C & \leftarrow & z \cdot y_C \\ R & \leftarrow & z \cdot R \end{array} \tag{10.37}$$

10.3.2.9 Abstraction du point de départ

La liste d'arcs d'une trajectoire fermée est, en réalité, circulaire. Afin de pouvoir l'utiliser dans la base de cas sous la forme d'une clé, cette liste doit être « coupée » en un point, en choisissant un premier arc. Pour cela, est désigné comme premier arc celui qui contient le point le plus éloigné du centre de gravité (déjà déterminé pour l'abstraction de symétrie, §10.3.2.7).

La transformation pour la normalisation du point de départ consiste donc en une permutation circulaire de la liste d'arcs.

10.3.2.10 Abstraction de la direction

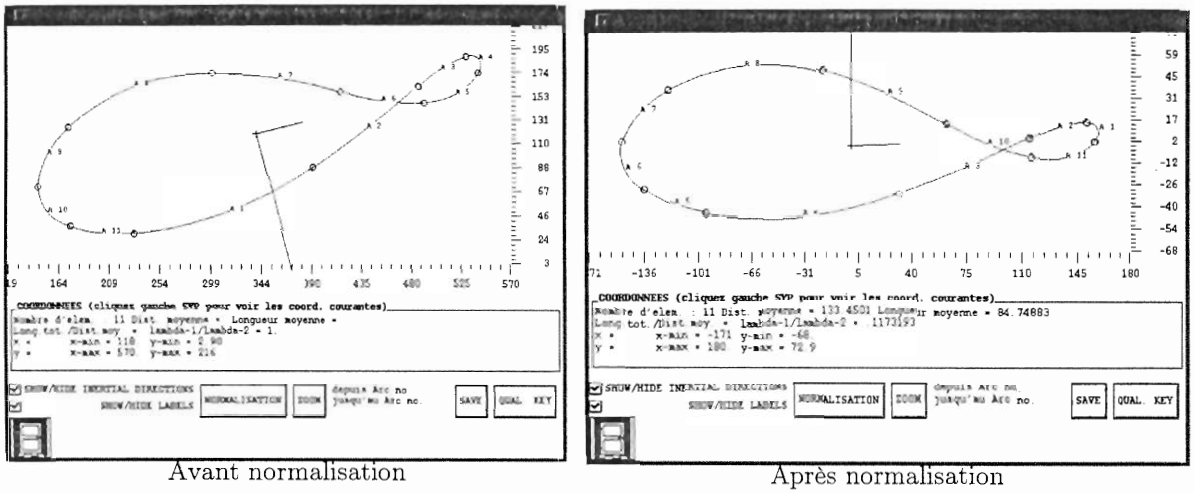
La direction de parcours d'une trajectoire est un invariant par rapport à sa forme. Afin de ne garder dans la base de cas qu'une seule des deux trajectoires équivalentes à une différence de direction près, il faut choisir un sens de parcours normalisé. Puisque la trajectoire qualitative peut éventuellement former des boucles (en huit par exemple), nous nous intéressons seulement au sens de parcours de l'arc auquel appartient le point de la plus grande abscisse.

Après la normalisation du point de départ, déjà effectuée, ce point appartient au premier arc. Le critère conventionnel de normalisation de la direction est le suivant : la trajectoire est parcourue dans le sens normalisé lorsque $\Delta\theta > 0$ pour le premier arc. Si cette condition n'est pas respectée, la direction doit être inversée, ce qui est obtenu en inversant l'ordre des arcs dans la liste et en modifiant pour chaque arc les angles :

$$\begin{array}{l} \theta_1 \leftarrow \theta_1 + \Delta\theta \\ \Delta\theta \leftarrow -\Delta\theta \end{array} \tag{10.38}$$

10.3.2.11 Exemple

Un exemple de trajectoire qualitative avant et après normalisation est donné dans la Figure 10.7.



Avant normalisation Après normalisation

FIG. 10.7 – *Exemple de normalisation d'une trajectoire qualitative (les vecteurs propres sont également représentés).*

10.3.3 Détermination de la clé qualitative

Le principe guidant la définition d'une clé qualitative est de décrire une trajectoire qualitative par une série de qualificatifs discrets, de haut niveau. De cette manière, on fait abstraction d'une partie de l'information quantitative, numérique, encore présente dans les trajectoires qualitatives (coordonnées du centre, angle initial et final, etc). Une telle clé a deux avantages :

- elle permet au concepteur de lancer des requêtes formulées approximativement, qualitativement ;
- étant beaucoup plus compacte, elle facilite la comparaison entre les cas enregistrés et la spécification.

En revanche, par rapport à une trajectoire qualitative, une telle clé est une description plus ambiguë d'une trajectoire exacte, dans le sens qu'il est possible que la même clé corresponde à des trajectoires de formes différentes. C'est pour cela que, dans la base de cas qualitatifs, nous enregistrerons sous chaque clé plusieurs trajectoires qualitatives (voir §10.5).

La clé qualitative d'indexation d'un cas est une séquence comprenant, pour chaque arc, trois valeurs qualitatives, portant sur la longueur, le rayon et le sens de parcours de l'arc. A cela se rajoute un symbole indiquant si la trajectoire est ouverte ou fermée. Le stockage des trajectoires ouvertes sera utilisé lors d'une spécification ne portant que sur une partie de trajectoire.

Les descripteurs qualitatifs relatifs à la longueur et au rayon devront se référencer par rapport à des valeurs de référence, L_{moy} et R_{moy}, pour échelonner les valeurs qualitatives {PETIT, MOYEN, GRAND} (ou autres échelles). Les variables exprimant les valeurs relatives L_{rel} et R_{rel} de la longueur et du rayon d'un arc par rapport à ces valeurs de référence et qui permettent donc de déterminer directement les valeurs qualitatives des descripteurs de clés, sont données par :

$$L_{\text{rel}} = \frac{L_i}{L_{\text{moy}}} \tag{10.39}$$

$$R_{\text{rel}} = \frac{r_i}{d_{\text{moy}}} \tag{10.40}$$

Ensuite, les valeurs qualitatives sont échellonnées sur des intervalles de L_{rel} et R_{rel} :

Cas	Symbole qualitatif
$L_{\text{rel}} < 0.5$ $R_{\text{rel}} < 0.5$	PETIT
$0.5 < L_{\text{rel}} < 1.5$ $0.5 < R_{\text{rel}} < 1.5$	MOYEN
$L_{\text{rel}} > 1.5$ $R_{\text{rel}} > 1.5$	GRAND

(10.41)

La valeur de référence L_{moy} est naturellement la moyenne des longueurs des arcs :

$$L_{\text{moy}} = \frac{\sum_{i=1}^{n} L_i}{n} = \frac{L_{\text{tot}}}{n} \tag{10.42}$$

Par contre, il n'est pas raisonnable d'adopter comme valeur de référence du rayon le rayon moyen des arcs. En effet, sous une même clé qualitative on doit avoir des courbes qui ont qualitativement la même forme. Or, une petite différence de forme entre deux courbes peut coexister avec une grande différence de valeur pour certains rayons et, par conséquent, pour le rayon moyen. Ainsi, le rayon moyen n'est pas caractéristique de la forme d'une trajectoire comme peut l'être la longueur moyenne. Nous préférons donc comme valeur de référence du rayon la distance quadratique moyenne des points de la trajectoire par rapport à l'origine du repère, d_{moy}, soit :

$$d_{moy}^2 \cdot L_{\text{tot}} = \int_{P \in \Sigma} \left(x^2 + y^2\right) dl(P) = \left(x_{moy}^2 + y_{moy}^2\right) L_{\text{tot}} \tag{10.43}$$

La troisième valeur qualitative, correspondant à la direction de parcours de chaque arc, est donnée simplement par le signe de $\Delta\theta$.

Des exemples de trajectoire qualitative, trajectoire qualitative normalisée et clé qualitative correspondante sont donnés dans la Figure 10.8.

Cette façon d'obtenir une clé qualitative n'est pas la seule possible. On peut envisager différentes manières pour enrichir la clé :

- en utilisant un nombre plus grand de valeurs qualitatives (par exemple $\frac{\Delta\theta}{\Delta\theta_{\text{moy}}}$),
- par la prise en compte des types de tangences entre les arcs (continuité $C1$ ou rebroussement, angles...),
- par la considération de l'aplatissement de la courbe, par exemple par le ratio :

$$\frac{x_{\text{moy}}}{y_{\text{moy}}} = \sqrt{\frac{\int_{P \in \Sigma} x^2 \, dl(P)}{\int_{P \in \Sigma} y^2 \, dl(P)}} = \sqrt{\frac{\lambda_2}{\lambda_1}} \tag{10.44}$$

D'une manière plus générale, on peut adopter une forme de clé qualitative correspondant à un langage symbolique de description qualitative des courbes (voir en ce sens les travaux sur la modélisation déclarative des trajectoires de *Daniel et al.* [DD94] [DL96] ; cf. §2.5.1). Néanmoins, dans le contexte de la synthèse des générateurs de trajectoire, il est important de construire la description qualitative seulement une fois que la trajectoire a été normalisée.

10.4 Stockage de la morphologie et des dimensions

10.4.1 Morphologie

Lors de l'enregistrement du cas qualitatif, la morphologie du mécanisme correspondant est stockée sous une forme symbolique, en pratique le nom du fichier contenant le graphe du mécanisme.

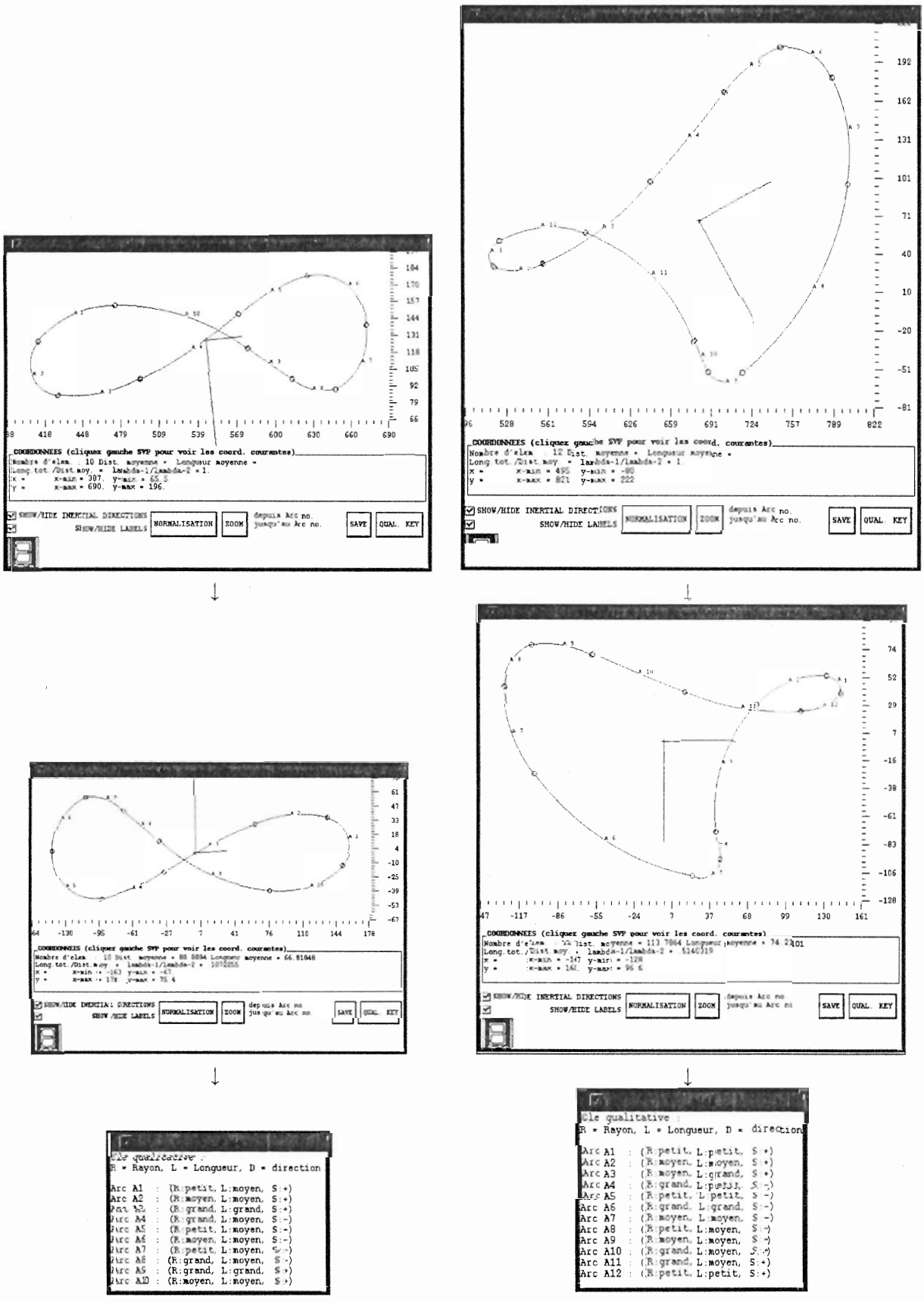


FIG. 10.8 – *Exemples de trajectoire qualitative (en haut), trajectoire qualitative normalisée (au milieu) et clé qualitative (en bas).*

10.4.2 Dimensions

10.4.2.1 Choix des paramètres indépendants

Les principes de choix des paramètres dimensionnels indépendants pour la méthode de synthèse par réseaux neuronaux (cf. §9.5.2.1) sont également valables ici. Il s'agit, principalement, de choisir le nombre minimal de paramètres, afin de réduire le plus possible la dimension de l'espace, sans pour autant réduire la diversité de formes de trajectoires. Ainsi, pour l'exemple du mécanisme *quatre barres*, sera donc enregistré dans le cas qualitatif le vecteur $(x_B,\ x_C,\ y_C, x_E, y_E)$ (mêmes notations que dans la Figure 9.12, p. 145).

Il n'est pas indispensable d'appliquer aux dimensions du mécanisme, avant l'enregistrement dans la base de cas, la même transformation que celle subie par les trajectoires lors de la normalisation. En effet, lors de l'utilisation de la base, la transformation des dimensions appropriée pourra être déterminée par la méthode de recalage géométrique (cf. §10.6.4.2).

10.4.2.2 Choix des intervalles de valeurs pour chaque paramètre

Le choix des intervalles de valeurs pour la synthèse par réseau neuronal (cf. §9.5.2.2) devait assurer un pourcentage élevé de jeux de dimensions faisables (c'est-à-dire correspondant à une trajectoire fermée). Ceci était dicté par l'impossibilité d'une analyse de Fourier d'une trajectoire ouverte. Dans le cas présent, on peut stocker dans la base de cas également les trajectoires ouvertes. Ces cas de trajectoires ouvertes ne seront pris en compte dans les requêtes que lorsque la spécification portera sur une portion de trajectoire.

Pour le mécanisme *quatre barres*, nous avons utilisé les mêmes intervalles de valeurs qu'au §9.5.2.2.

10.5 Construction de la base de cas

La première étape de la génération des cas est en tous points analogue à celle utilisée dans le chapitre précédent (voir §9.6.1). Il s'agit donc d'un tirage aléatoire de jeux de dimensions à l'intérieur des intervalles préalablement choisis. En pratique, le même fichier de données (dimensions et trajectoires exactes) peut être utilisé.

La différence entre les deux méthodes consiste en la transformation des cas. Ici, les trajectoires exactes sont rendues qualitatives par une approximation en arcs de cercle. Ces trajectoires sont ensuite normalisées (cf. §10.3.2) et, finalement, la clé qualitative est déterminée (cf. §10.3.3).

La différence majeure est que, pour tout cas à stocker, on procède à plusieurs approximations qualitatives suivant plusieurs degrés de précision, en pratique une dizaine. Plus la précision d'approximation est importante, plus la trajectoire qualitative comportera d'arcs et vice-versa. Néanmoins, ces résultats d'approximation qualitative sont tous représentatifs de la trajectoire exacte initiale, à différents niveaux de résolution (précision). Pour chaque trajectoire qualitative ainsi obtenue, la clé qualitative est déterminée. Pour chaque apparition d'une nouvelle clé, est stocké le cas correspondant à la précision la plus élevée.

La génération de cas est arrêtée lorsque le taux d'apparition de nouvelles clés descend en dessous d'une certaine limite. (Voir Algorithme 10.2.)

Dans la Figure 10.10, pour la même trajectoire exacte (représentée dans la Figure 10.9) sont présentées trois trajectoires qualitatives obtenues pour trois valeurs de la précision d'approximation, ainsi que les clés qualitatives correspondantes.

Etape 1 Tirage aléatoire d'un jeu de dimensions D pour le mécanisme Mec.

Etape 2 Détermination de la trajectoire exacte T_{ex} (par simulation cinématique).

Etape 3 Approximation de la trajectoire T_{ex} par les trajectoires qualitatives T_{qual_i} pour $i = 1, \ldots, n$ degrés de précision.

Etape 4 Obtention, par normalisation, des trajectoires qualitatives normalisées $T_{qual_norm_i}$.

Etape 5 Pour chaque $T_{qual_norm_i}$, détermination de la clé qualitative K_i.

Etape 6 Pour toutes les valeurs de K_i distinctes, enregistrement du cas pour la plus grande précision, soit : $(K_i, Mec, D, T_{qual_norm_i}, \max_j(p_{ij}))$ (cf. Tableau 10.2).

Si la clé existait déjà dans la base, plusieurs valeurs de cas seront stockées sous la même clé, dans la limite d'un nombre acceptable.

Etape 7 Arrêt lorsque la fréquence d'apparition de nouvelles clés devient trop faible, sinon retour à l'étape 1.

Algorithme 10.2: *Procédure de création des cas qualitatifs.*

clés qualitatives	K_1	$\boxed{K_1}$	$\boxed{K_2}$	K_3	K_3	$\boxed{K_3}$	...
degrés de précision	p_1	$\boxed{p_2}$	$\boxed{p_3}$	p_4	p_5	$\boxed{p_6}$	...

TAB. 10.2 – *Génération de cas qualitatifs à partir d'une trajectoire exacte. La trajectorie exacte est approximée avec des précisions p_i croissantes et donne des clés qualitatives K_i. Les précisions et les clés encadrées sont celles enregistrées dans la base de cas.*

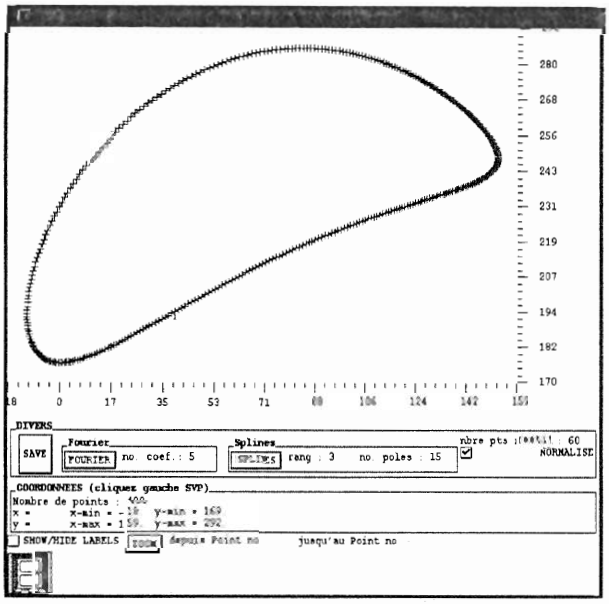
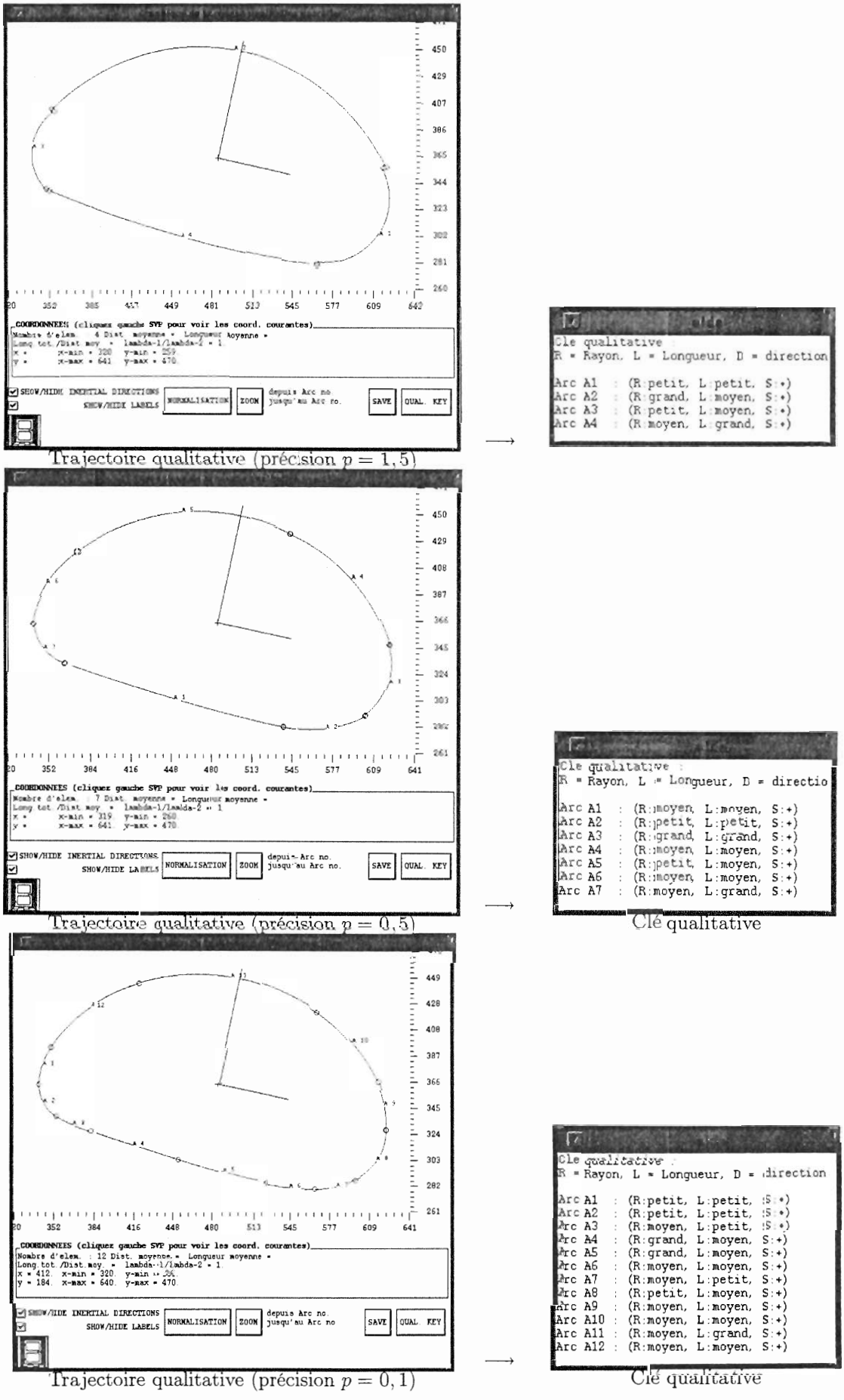


FIG. 10.9 – *Trajectoire exacte dont l'approximation qualitative est donnée dans la Figure 10.10.*



Trajectoire qualitative (précision $p = 1,5$)

Trajectoire qualitative (précision $p = 0,5$)

Clé qualitative

Trajectoire qualitative (précision $p = 0,1$)

Clé qualitative

FIG. 10.10 – *Exemple d'approximation de la même trajectoire exacte (en haut) pour trois précisions différentes.*

Spécification	Base de cas
K_i, p_i	$K_j, (p_{j1}, Mec_1), \ldots, (p_{jk}, Mec_k), \ldots, (p_{jn}, Mec_n)$

TAB. 10.3 – *Recherche des cas les plus proches dans la base de cas qualitatifs. p_i, respectivement p_{jk} sont les précisions de l'approximation de la requête et du cas, Mec_k est le mécanisme du cas qualitatif.*

10.6 Utilisation de la base de cas

10.6.1 Méthode

Une fois que la base de cas qualitatifs est réalisée, la méthode de synthèse consiste en la spécification d'une forme de trajectoire, suivie par la recherche des cas les plus proches dans la base et leur classement selon l'adéquation probable avec la spécification, puis, éventuellement, leur adaptation. Ces étapes seront détaillées dans les sous-chapitres suivants.

10.6.2 Description du cas : spécification de la forme de trajectoire désirée

Pour les méthodes analytiques de synthèse, la spécification doit impérativement comporter une liste de points de passage sur la trajectoire désirée. Dans notre méthode de synthèse de forme basée sur les cas qualitatifs, le concepteur à une plus grande liberté pour exprimer le besoin en termes de forme. Ainsi, il est possible de spécifier la forme désirée de plusieurs manières, à différents niveaux d'abstraction (cf. Figure 10.1) :

- directement dans le langage qualitatif utilisé pour les clés qualitatives (cf. §10.3.3) ;
- par une trajectoire spécifiée qualitative (arcs de cercles) ;
- par une trajectoire exacte, qui sera ensuite rendue qualitative par une approximation[5].

10.6.3 Calcul de l'index

Si la forme de trajectoire désirée est spécifiée par une trajectoire exacte, les clés qualitatives de cette trajectoire sont calculées de la même manière que lors de la création de la base de cas (cf. §10.3), par une approximation selon plusieurs degrés de précision. Si la spécification est donnée sous la forme d'une trajectoire qualitative, la seule opération à faire est le calcul de la clé qualitative. Finalement, si la spécification est donnée directement par une clé qualitative, nous disposons directement de l'index et il n'y a plus aucune opération à faire.

10.6.4 Obtention de la solution

10.6.4.1 Recherche du cas le plus proche

Après l'obtention des clés qualitatives correspondant à la trajectoire désirée (cf. §10.6.3), pour chaque clé différente K_i et pour chaque précision maximale correspondante p_i, on effectue une requête dans la base de cas (voir Tableau 10.3).

Le résultat des requêtes est stocké sous la forme de triplets (p_1, p_2, *Liste_Mécanismes*), où p_2 est la plus grande précision d'un cas de la base ayant la même clé qualitative. Un classement qualitatif de ressemblance peut être donné par le critère $\max(\min(p_1, p_2))$. Un classement numérique

5. Dans ce dernier cas, l'éditeur de trajectoires de REALISME (voir §6.3.1, Figure 6.4, p. 92), utilisant une modélisation par splines, peut être utilisé. Evidemment, la répartition particulière des points de contrôle ne sert qu'à la description de la forme de la trajectoire.

Translation :	$\Delta x_1 = -x_G,\ \Delta y_1 = -y_G$	(cf. éq. 10.20)
Rotation :	$\Delta\alpha_1 = -\tan^{-1}\frac{v_{12}}{v_{11}}$	(cf. éq. 10.22)
Symétrie :	selon les signes de $x_{\text{dist. max}}$ et $y_{\text{dist. max}}$	(cf. Tab. 10.1)
Echelle :	$\Delta z_1 = \frac{d_{\text{moy}_{\text{norm}}}}{d_{\text{moy}}}$	(cf. éq. 10.37)
Point de départ :	n° arc avec point le plus éloigné	(cf. §10.3.2.9)
Direction :	selon le signe de $\Delta\theta$ pour le premier arc	(cf. §10.3.2.10)

TAB. 10.4 – *Paramètres de normalisation d'une trajectoire qualitative.*

Translation :	$\Delta x = \Delta x_2 - \Delta x_1 \quad \Delta y = \Delta y_2 - \Delta y_1$
Rotation :	$\Delta\alpha = \Delta\alpha_2 - \Delta\alpha_1$
Echelle :	$\Delta z = \frac{\Delta z_2}{\Delta z_1}$

TAB. 10.5 – *Paramètres optimaux de recalage pour les trajectoires qualitatives.*

peut aussi être effectué, avec des résultats éventuellement différents, selon le critère donné par la distance entre la trajectoire qualitative désirée et chacune des trajectoires enregistrées dans la base de cas sous la même clé (voir équation 8.23, p. 124).

10.6.4.2 Adaptation par recalage géométrique

Une méthode de recalage géométrique des trajectoires générée et désirée, utilisée pour l'optimisation dimensionnelle exacte, a déjà été présentée en §8.2.5. Cette méthode, spécifique aux trajectoires exactes (liste de points), consistait en une détermination directe des paramètres optimaux de transformation de la trajectoire générée afin qu'elle s'approche au mieux de la trajectoire désirée. Lorsque les trajectoires sont modélisées qualitativement, par une approximation en arcs de cercle, cette méthode de détermination directe n'est plus applicable ; une méthode de recalage différente, basée sur la normalisation des deux trajectoires (comme dans §9.4), sera donc utilisée.

Lors de la normalisation de la trajectoire synthétisée, les paramètres $\Delta x_1, \Delta y_1, \Delta\alpha_1, \Delta z_1$, ainsi que le point de départ et la direction, sont enregistrés (voir Tableau 10.4). Les paramètres optimaux de recalage résulteront de la différence par rapport aux paramètres de normalisation $\Delta x_2, \Delta y_2, \Delta\alpha_2, \Delta z_2$ et par rapport au point de départ et à la direction de la trajectoire désirée (voir Tableau 10.5).

Un exemple de recalage géométrique de deux trajectoires qualitatives est donné dans la Figure 10.11.

10.6.4.3 Adaptation par optimisation dimensionnelle

Comme pour la méthode de synthèse utilisant un réseau neuronal (cf. §9.7.4.3), le mécanisme synthétisé peut être utilisé comme solution initiale dans un processus d'optimisation dimensionnelle avec indépendance du temps (cf. chap. 8).

10.7 Conclusion

Pour conclure ce chapitre, nous allons comparer l'approche « catalogue de trajectoires qualitatives » et l'approche par coefficients de Fourier et réseaux neuronaux.

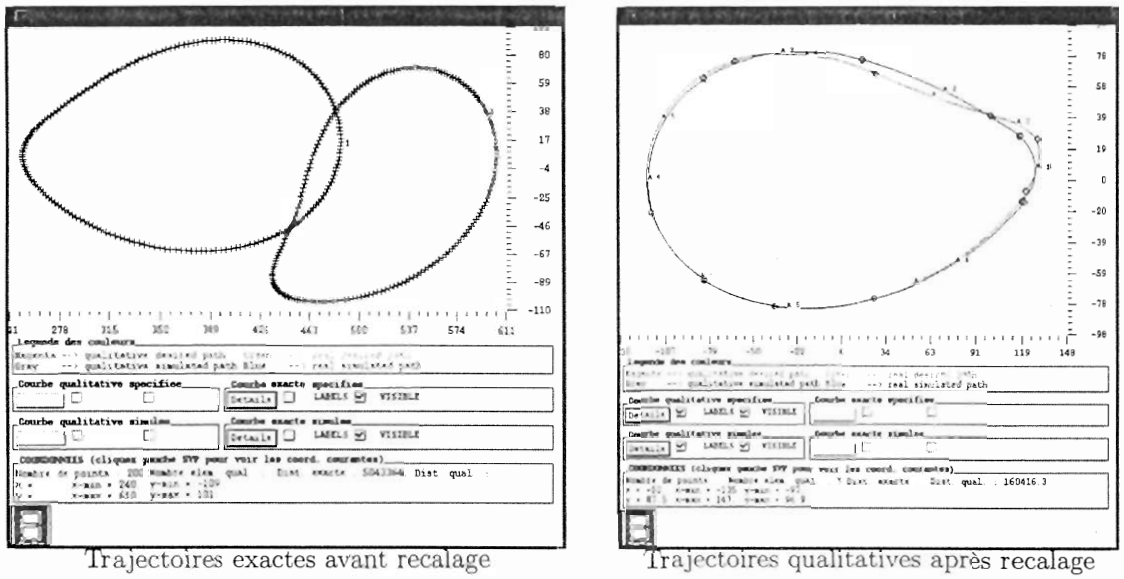


FIG. 10.11 – *Représentation superposée de deux trajectoires exactes (à gauche) et des trajectoires qualitatives correspondantes après recalage géométrique (à droite).*

L'utilisation d'un réseau neuronal artificiel pour approximer la fonction reliant les dimensions aux trajectoires a comme conséquence la restriction à une seule morphologie pour un même réseau. Afin de prendre en compte plusieurs morphologies, il est donc nécessaire de comparer les résultats obtenus séparément avec plusieurs RNA. En revanche, pour une base de cas qualitatifs cette restriction à une seule morphologie disparaît. Ainsi, la morphologie du mécanisme figurera, de façon symbolique, parmi les données enregistrées pour chaque cas.

Une deuxième restriction levée concerne l'inclusion dans la base de cas non seulement des trajectoires fermées, mais également des trajectoires ouvertes (correspondant à un jeu de dimensions pour lequel la rotation complète de la manivelle n'est pas possible). Par conséquent, lors de la création de la base de cas qualitatifs, ne sont plus nécessaires les précautions indispensables dans le cas des RNA afin de limiter les domaines de valeurs des dimensions aux jeux de dimensions correspondants à des trajectoires fermées.

Pour les deux méthodes de synthèse, il est possible de spécifier d'une façon déclarative la forme de trajectoire désirée. Néanmoins, la méthode basée sur les cas qualitatifs est mieux adaptée à une telle spécification, car elle emploie une description déclarative au sein même de la représentation des cas. La comparaison avec la spécification est donc directe.

Lors de l'approximation des trajectoires par des coefficients de Fourier normalisés, le concepteur n'a pas le contrôle direct de l'erreur commise. De plus, les solutions fournies par le réseau neuronal sont issues d'une interpolation, qui introduit un deuxième facteur d'erreur difficilement contrôlable. En revanche, l'erreur d'approximation maximum commise lors de l'approximation qualitative est parfaitement maîtrisée, puisque c'est un paramètre de l'algorithme d'approximation.

Le réseau neuronal fournit une solution unique. Afin de comparer différentes solutions alternatives, il faudra avoir recours à plusieurs RNA, correspondant à des morphologies différentes. En revanche, cette solution est interpolée et sera le plus souvent meilleure qu'une liste de cas proches. Par contre, une requête effectuée dans la base de cas qualitatifs peut fournir un nombre désiré de solutions alternatives, classées en fonction d'un critère numérique.

Ces éléments de comparaison entre les deux méthodes sont résumés dans le Tableau 10.6.

	CBR Fourier + RNA	**CBR qualitatif**
Synthèse conjointe morphologie+dimensions	par bouclage sur $\neq$ RNA	oui
Trajectoires ouvertes	non	oui
Trajectoires fermées	oui	oui
Spécification pour un morceau de traj.	non	envisageable
Traj. avec dépendance du temps	envisageable	envisageable
Traj. avec indépendance du temps	oui	oui
Maîtrise de l'erreur (précision)	non	oui
Obtention de plusieurs solutions	non	oui
Solution interpolée	oui	non

TAB. 10.6 – *Comparaison de la synthèse par CBR utilisant l'analyse harmonique et les réseaux neuronaux avec la synthèse par base de cas qualitatifs.*

Une version « minimale » de la méthode présentée dans ce chapitre a été implémentée dans REALISME. Toutefois, elle semble receler encore plus de potentialités, très prometteuses pour la préconception, que nous envisageons d'étudier plus finement par la suite.

Quatrième partie

Conclusion et annexes

Chapitre 11

Conclusions et perspectives

11.1 Contributions de cette thèse

Cette thèse a comme objectif le développement de méthodes de préconception pour les générateurs de trajectoire. Dans ce domaine, il existe plusieurs classes de problèmes ou de spécifications, que l'on peut classifier selon différents critères : spécification exacte ou approximative, portant sur tout ou partie de la trajectoire, avec ou sans prise en compte du temps... A cette diversité de situations de conception correspondent, dans REALISME, des approches d'aide à la conception qui se déclinent en plusieurs modèles et méthodes, ayant fait l'objet de différents chapitres de la thèse. Nous précisons, dans ce chapitre, les contributions originales et les apports par rapport à l'existant.

11.1.1 Modèles de cinématique

Dans REALISME, ont été utilisés deux modèles de cinématique déjà existants : le *modèle systémique multipolaire* (*Pelecudi et Simionescu* [PDS82] [PS82] [PS86a] [PS86b]) et le *modèle qualitatif de cinématique* (*Yannou et al.* [YV95] [YVM95]).

11.1.1.1 Le modèle systémique multipolaire

Le *modèle systémique multipolaire* (cf. chap. 4) est une évolution de l'approche de décomposition des mécanismes plans en groupes d'Assur. De nombreux logiciels d'analyse utilisant les groupes d'Assur ont été présentés dans la littérature. Cependant, l'implémentation dans REALISME de l'approche systémique multipolaire (cf. *Vasiliu* [Vas94]), faisant appel à une symbolique graphique expressive, a des qualités intéressantes en préconception. A notre connaissance, il s'agit de **la première utilisation de cette approche dans un logiciel graphique interactif**.

Les particularités de cette approche nous ont permis de réaliser **un simulateur cinématique particulièrement adapté à des cycles prolongés de modification/analyse** (cf. chap. 6). En effet, après une construction à main levée du schéma cinématique, la simulation est réalisée en un temps très court, l'animation graphique étant intercalée avec le calcul. Une fois la morphologie définie, un cycle de modification dimensionnelle/analyse peut être accompli simplement en déplaçant à la souris une liaison et en relançant la simulation. La mise en équations étant réalisée une fois pour toutes pour la même morphologie, grâce à un algorithme original de décomposition automatique, la simulation prend en compte directement les nouvelles dimensions, ayant ainsi un caractère « paramétrique». Lors des blocages ou des positions singulières, des messages appropriés sont affichés, indiquant leur cause et suggérant des remèdes éventuels. Nous pensons que **cette combinaison de caractéristiques et cette convivialité sont très rares parmi les logiciels existants**.

11.1.1.2 Le modèle de cinématique qualitative

Nous avons intégré dans REALISME (cf. chap. 5 et *Yannou et Vasiliu* [YV95] [YVM95]) le noyau de *cinématique qualitative*, conçu et programmé par B. Yannou, afin de l'exploiter pour la synthèse des générateurs de trajectoire. Ce noyau utilise l'interface de REALISME pour la définition du mécanisme, pour la simulation cinématique exacte préalable et pour l'interface de commande et d'affichage des résultats.

11.1.2 Méthodes de synthèse

La synthèse des mécanismes se divise traditionnellement en synthèse morphologique et dimensionnelle. Cette thèse est consacrée principalement à la synthèse dimensionnelle, mais elle a donné lieu également à des contributions à la synthèse morphologique. Ainsi, d'une part, la *construction ou la modification des morphologies* en utilisant la représentation systémique, par le schéma multipolaire, est en soi un moyen d'aider à la rationalisation des choix morphologiques du concepteur.

D'autre part, un module de *représentation graphique dépliée* des graphes correspondants aux mécanismes, utilisant **une approche de dépliage originale et efficace basée sur l'optimisation du positionnement topologique relatif des cycles d'une base de cycles fondamentaux** (cf. chap. 7 et *Vasiliu* [Vas92] [Vas93]), a été inclus dans REALISME. Ce module, utilisable pour toute application manipulant des graphes, présente un intérêt pour un grand nombre de recherches de la littérature, portant sur la génération automatique de morphologies, afin d'améliorer l'interface avec le concepteur. Le modèle par graphes n'est pas, pour l'instant, employé directement au sein de méthodes de synthèse morphologique de REALISME et il sert simplement de structuration informatique des données morphologiques et dimensionnelles.

11.1.2.1 L'optimisation dimensionnelle

Dans la littérature, de très nombreux travaux ont abordé les méthodes d'optimisation dimensionnelle des mécanismes. Sans prétendre « révolutionner » ce domaine, le module d'*optimisation traditionnelle* (ou *exacte*) de REALISME (cf. chap. 8 et *Yannou et Vasiliu* [YV95] [YVM95]) utilise une approche générale, basée sur un bouclage sur des simulations cinématiques. Le module d'*optimisation qualitative* est **une application originale de la méthode de simulation cinématique qualitative au sein d'une procédure d'optimisation**. La rapidité qui caractérise également les deux méthodes devient un facteur très intéressant pour ces méthodes d'optimisation, qui nécessitent un grand nombre de simulations.

Nous avons reconsidéré le choix de la formulation pour la fonction objectif, notamment pour les cas où le mécanisme atteint en cours de simulation des positions singulières ou des blocages. Pour cela, notre solution consiste en **l'utilisation d'un terme supplémentaire dans la fonction objectif, dépendant de la différence de longueur entre les trajectoires désirée et générée**.

D'une façon complémentaire aux méthodes de continuation de la littérature, nous proposons **une méthode d'accélération de la convergence par *recalage géométrique***. Cette méthode part du constat que les trajectoires sont solidaires géométriquement des mécanismes ; les transformations géométriques appliquées à l'un se répercutent d'une façon identique à l'autre. Il est ainsi possible de déterminer une transformation optimale qui « recale » au mieux la trajectoire générée sur la trajectoire désirée. Pour l'optimisation, l'idée du recalage se décline en deux procédures de calcul de la transformation optimale, selon le mode de représentation des trajectoires : trajectoires exactes ou trajectoires qualitatives. La première procédure conduit à la résolution d'un système d'équations linéaires, tandis que la deuxième est basée sur la définition d'une procédure de normalisation originale pour les trajectoires qualitatives.

11.1.2.2 La synthèse par capitalisation de cas

Le principal apport à la synthèse dimensionnelle de REALISME est représenté par les **deux approches originales de synthèse par capitalisation de cas**. Ces approches partent du principe général d'utilisation d'un catalogue de trajectoires pour le choix des dimensions des générateurs de trajectoires, ce qui est une pratique existant dans les bureaux d'études. A la différence des approches analytiques de synthèse, qui déterminent directement et d'une façon explicite les solutions dimensionnelles correspondant à la spécification de synthèse (malheureusement fortement limitée), les approches par capitalisation tentent d'**élargir la gamme de spécifications possibles** par la capitalisation d'un ensemble de cas. En particulier, on vise à permettre au concepteur de concevoir des mécanismes décrivant une trajectoire spécifiée en termes de *forme* ou approximativement (*qualitativement*) pour tout ou partie de la trajectoire.

Pour ces types de spécification, les méthodes de synthèse existantes ne sont pas du tout adaptées. La principale difficulté est l'impossibilité d'établir un lien explicite entre la structure du mécanisme et les caractéristiques attendues de la trajectoire. C'est pour cette raison que l'on fait appel à la capitalisation des cas, c'est-à-dire au stockage intelligent des couples trajectoire/structure du mécanisme. Une telle base de cas contient d'une façon implicite des solutions pour un grand nombre de spécifications virtuelles. En phase d'utilisation, il suffit de mettre en regard la spécification avec la base de cas pour déterminer une solution lui répondant au mieux.

Du point de vue du concepteur, les méthodes par capitalisation de cas, par opposition aux méthodes analytiques ou d'optimisation, réduisent la tâche du concepteur au seul recours à une « mémoire » prédéfinie. L'utilisation de cette « mémoire » est relativement simple et rapide, parce que l'essentiel du travail a été effectué « en temps masqué », lors de sa mise au point. Ce type d'approche est justifié et rentable lorsqu'une même base peut être utilisée un grand nombre de fois par un grand nombre de concepteurs.

Les deux approches de réalisation d'une base de cas diffèrent quant aux modes de représentation du cas, ainsi que par le mode d'enregistrement des cas.

Ainsi, l'approche de *synthèse par capitalisation de cas dans un réseau neuronal* (chap. 9) utilise pour la codification des trajectoires les coefficients de Fourier correspondant à une analyse de Fourier des trajectoires exactes. Cette représentation des trajectoires s'inspire des travaux de *McGarva* [MM93] [McG94], qui fournissent également une procédure de calcul des coefficients correspondants aux trajectoires normalisées (afin de s'abstraire des transformations géométriques) et proposent d'enregistrer les couples trajectoires codifiées/dimensions dans une base de données, à des fins de synthèse. En plus de certaines **modifications et corrections apportées à l'algorithme de calcul des coefficients par McGarva**, nous utilisons **une méthode d'une nature complètement différente de la sienne pour stocker les cas**. McGarva obtient, par des simulations cinématiques, les trajectoires correspondant à des dimensions qu'il fait varier de manière uniforme, et enregistre ces dimensions et les coefficients dans des fichiers. Lors de la synthèse, les coefficients sont calculés pour la trajectoire spécifiée et une recherche de proximité dans le fichier permet de retrouver le cas enregistré le plus proche.

Dans notre approche, au lieu de les enregistrer tels quels, les cas – générés également par des simulations cinématiques, mais pour des jeux de dimensions répartis aléatoirement à l'intérieur d'intervalles de dimensions – sont utilisés par un **algorithme d'apprentissage d'un réseau neuronal**. Le but est de réaliser l'approximation de la fonction reliant la codification des formes de trajectoires aux dimensions du mécanisme, ce qui permet de **synthétiser le mécanisme par simple propagation de la spécification dans le réseau** (lorsqu'elle est faite en termes de forme).

La codification des trajectoires par les coefficients de Fourier (ou autre méthode apparentée) est un prérequis indispensable pour l'emploi d'un réseau neuronal, car elle permet de traduire la

forme d'une trajectoire par un certain nombre, relativement réduit mais fixe, de valeurs numériques (coefficients).

Le grand avantage de l'interpolation des cas par le réseau neuronal est l'**agrandissement de la capacité de synthèse à nombre initial de cas égal**. En effet, même pour des mécanismes simples, pour un nombre de cinq paramètres dimensionnels, il faut un nombre énorme de cas pour remplir d'une façon fine l'espace des jeux de dimensions. La distance moyenne entre deux cas voisins restera donc, fatalement, assez grande. Il est clair, dans ces conditions, qu'une interpolation permet d'avoir une meilleure solution, pour un cas à synthétiser situé entre les deux, que le simple choix du plus proche des deux cas.

Cependant, réaliser une telle interpolation est une tâche difficile à plusieurs titres : nature et dimension de la fonction, nombre de cas nécessaires, gestion des cas de mécanismes pour lesquels il n'y a pas de trajectoire générée fermée (blocages et positions singulières). Dans REALISME, nous avons pu réaliser l'apprentissage d'un réseau pour les trajectoires générées par le mécanisme *quatre barres*. La validation de la faisabilité de l'approche pour des mécanismes plus complexes fait partie des suites possibles de ce travail.

L'exploration rigoureuse des limites d'application et d'efficacité de l'interpolation par réseaux neuronaux exigerait de mener des recherches plus approfondies, notamment par des spécialistes des techniques neuronales.

Une partie des limites d'applicabilité de cette première approche de capitalisation de cas ont néanmoins été clairement identifiées. Il est notamment impossible de stocker des trajectoires ouvertes ou de s'intéresser, dans la spécification, seulement à des *parties* de trajectoires fermées. Ceci s'explique par la nature forcément *périodique* des fonctions pour l'analyse de Fourier. De plus, nous ne savons pas comment réaliser des réseaux neuronaux correspondant à plusieurs morphologies différentes. En tout cas, même si elle présente des fonctionnalités sans équivalent dans la littérature (**obtenir en moins d'une seconde une solution approximative en termes de forme**), il est clair que cette méthode de synthèse n'est pas, ni en son ensemble, ni dans chaque détail, une solution absolue et définitive, qui interdirait de chercher d'autres méthodes alternatives.

C'est pour cela qu'a été développée la deuxième approche de *synthèse par capitalisation de cas utilisant la modélisation qualitative des trajectoires* (cf. chap. 10). Issues de la cinématique qualitative (déjà évoquée), les trajectoires qualitatives sont simplement une approximation en arcs de cercle et segments des trajectoires exactes générées. (Ici seuls les arcs de cercle sont utilisés.) Comme la modélisation par coefficients de Fourier, cette approximation des trajectoires est à la fois compacte et expressive : elle prend moins de place, mais permet de s'abstraire d'une répartition particulière, souvent non fonctionnelle, des points sur la trajectoire.

La base de cas est réalisée par la même procédure que pour l'approche précédente, utilisant un grand nombre de simulations cinématiques pour des jeux de dimensions générés aléatoirement. L'intérêt de la modélisation qualitative par arcs de cercle est qu'elle permet de bâtir une **représentation des trajectoires encore plus qualitative**, que l'on appelle *clé qualitative*. Cette clé exprime de manière symbolique les grandeurs relatives des rayons et des longueurs de chaque arc, ainsi que le sens de parcours. Etant symbolique et donc discrète, la clé constitue l'élément essentiel pour une constitution et une utilisation efficaces de la base de cas.

En effet, la clé qualitative permet, en phase de constitution de la base de cas, de comparer un cas nouveau avec les cas déjà enregistrés, d'une manière beaucoup plus rapide que par une fonction distance habituelle. A son tour, cette comparaison donne la possibilité de **sélectionner les cas à enregistrer, de façon à ne pas augmenter le nombre de cas inutilement**, lorsque des trajectoires de forme proche ont déjà été enregistrées.

	optimisation dim. exacte	**optimisation dim. qual.**	**capitalisation Fourier+RNA**	**capitalis. traj.qual.**
dépendance du temps	oui	oui	envisageable, àexpérimenter	
indépendance du temps	difficile	difficile	**oui**	**oui**
trajectoire désirée ouverte	difficile pour la fonction coût		non	oui
trajectoire désirée fermée	oui	oui	oui	**oui**
déclarativité pour traj.dés.	non	oui	non	**oui**
n° et qualité des solutions	1 (dép.des C.I.)	1 (dép.des C.I.)	1 (interpolation)	n (classement)

TAB. 11.1 – *Tableau récapitulatif des domaines d'application des différentes méthodes utilisées en* REALISME.

En phase d'utilisation, l'indexation de la base selon les clés qualitatives est également un moyen d'**accéder très rapidement aux cas de la base** présentant la même clé que la spécification désirée par le concepteur.

Une base de cas de ce type présente l'avantage considérable de pouvoir **contenir des cas correspondant à des mécanismes différents**, avec des **trajectoires générées ouvertes ou fermées indifféremment**. Par contre, nous n'avons pas encore assez étudié l'extension de l'approche pour les trajectoires avec dépendance du temps.

On peut caractériser de manière synthétique la différence entre les deux approches de capitalisation des cas, par analogie avec l'informatique et de façon plutôt métaphorique, en disant que l'interpolation par réseaux neuronaux est comparable au stockage de données dans un fichier *binaire* (très compact et d'accès rapide, mais illisible directement à cause du codage), tandis que le catalogue de trajectoires qualitatives ressemble à un fichier *texte* (ASCII) (plus grand et d'accès plus lent, mais dans un format lisible). Quant au type d'accès, dans les deux cas il s'agit d'un *accès direct* et non pas séquentiel. Effectivement, le réseau neuronal permet de concentrer dans très peu de données des informations portant sur un grand nombre de cas, mais il impose une codification bien définie et unique des cas, et fournit une solution unique. En revanche, la base de trajectoires qualitatives, d'une taille beaucoup plus importante, permet d'imaginer diverses codifications qualitatives pour les clés et divers systèmes de requêtes, fournissant éventuellement plusieurs solutions candidates, classées selon différents critères. C'est donc une approche plus souple, dans le sens qu'il est possible d'imaginer de nombreuses fonctionnalités d'utilisation de cette base.

11.2 Domaines d'application

Le Tableau 11.1 est une récapitulation des domaines d'applications des méthodes de synthèse pour les générateurs de trajectoire, développées dans REALISME.

Des couplages entre ces méthodes sont possibles. Ainsi, les méthodes de synthèse par capitalisation de cas peuvent fournir une solution initiale approximative pour les méthodes d'optimisation dimensionnelle.

11.3 Recherches futures

Cette thèse traite d'un domaine relativement vaste de la conception mécanique. Pour ce domaine, incluant une diversité de problématiques, des solutions diverses ont été proposées. Il est naturel, donc, qu'il existe une multitude d'aspects ou de points qui pourront faire l'objet de recherches futures. La liste ci-après est loin d'être exhaustive :

- Le modèle systémique multipolaire :
  - Aide « intelligente » à la modification des dimensions du mécanisme afin de rendre possible la rotation complète de la manivelle, par propagation aux groupes adjacents des contraintes géométriques spécifiques à chaque groupe multipolaire (cf. §6.4.4).
- Méthodes de synthèse en général :
  - Développer la génération d'*explications qualitatives* pour la synthèse combinée morphologique et dimensionnelle.
  - Meilleure intégration des contraintes de coût et d'environnement : contraintes spécifiques ou pénalisation pour l'optimisation, critères de classement des solutions pour les catalogues de trajectoires qualitatives.
- Les algorithmes d'optimisation pour la synthèse dimensionnelle :
  - Quel sens peut avoir l'optimisation conjointe morphologique et dimensionnelle, qui impliquerait de mélanger des variables continues (dimensions) et discrètes (morphologie) ?
  - Comment utiliser l'identification automatique des causes de blocage et de position singulière pour privilégier certaines directions de recherche dans l'espace de solutions, dans le cadre de stratégies d'optimisation intelligentes et contextuelles ?
  - Définir un langage complet de spécification d'un générateur de trajectoire qui prenne notamment en compte le fait que les spécifications portent sur tout ou partie de la trajectoire et qui prenne en compte des contraintes diverses dans la fonction coût. Une spécification générée dans ce langage devrait ensuite pouvoir être formulée automatiquement en un algorithme d'optimisation, voire en une requête intelligente dans une des bases de cas.
  - Etude de l'optimisation dimensionnelle pour les mécanismes à plusieurs degrés de liberté (assez rares comme générateurs de trajectoire).
- La synthèse par capitalisation de cas dans un réseau neuronal :
  - Nous avons évoqué dans l'état de l'art (§2.5.3, p. 35) les modèles de réseaux neuronaux hiérarchiques et multi-résolution, appliqués au problème d'inversion cinématique du robot *OSCAR-6* (cf. *Groenewoud* [vhG95] et *van Smagt et Groen* [vdSG95]). La mise au point de ce type de réseaux a été motivée par l'impossibilité d'atteindre, avec les RNA de type *feedforward,* le niveau de précision requis pour les problèmes à grande dimension. Il est à envisager l'utilisation de cette classe de réseaux pour la synthèse des générateurs de trajectoire.
  - Existe-t-il des techniques neuronales avancées permettant de réaliser une approximation par le même réseau neuronal pour plusieurs morphologies, c'est-à-dire en mixant des variables continues et discrètes ?

  - Réalisation d'un réseau neuronal dédié au mécanisme *quatre barres* générant des trajectoires *symétriques*. En effet, ceci implique la condition dimensionnelle d'égalité entre deux longueurs de barres (la condition de symétrie, cf. *Kota et Chiou* [KC93]), ce qui diminue le nombre minimal de paramètres dimensionnels nécessaires. Il est probable que l'approximation par le réseau est, dans ce cas, plus facile à faire et de meilleure qualité.
  - Est-il possible de réaliser l'apprentissage du réseau lorsque les trajectoires sont modélisées par l'analyse de Fourier avec dépendance du temps ?

- Synthèse par capitalisation des cas sous forme de trajectoire qualitative :
  - Mettre au point une interface *déclarative* pour les trajectoires désirées. On pourrait s'inspirer des travaux portant sur la description déclarative de courbes entrepris à l'Ecole Centrale de Nantes (*Daniel et al.* [DD94] [DL96]) pour aborder la conception déclarative de générateurs de trajectoire.
  - Réaliser une interface avec un système de gestion de base de données, qui se chargerait du stockage des cas qualitatifs et des requêtes.
  - Mise au point d'une clé qualitative enrichie.

- Synergie des méthodes :
  - Il serait intéressant, à plus long terme, de lancer des résolutions en parallèle par toutes les méthodes disponibles et de traiter les résultats par un système expert, afin d'arriver à en extraire le meilleur résultat ou à fusionner ces résultats.

La synthèse des générateurs de trajectoire est un cas de problème scientifique assez paradoxal : la compréhension et la modélisation sont relativement simples, mais la résolution est très difficile. En effet, pour une partie des besoins de conception, les solutions de la littérature ne sont pas entièrement satisfaisantes, malgré de longues décennies de recherches. C'est probablement ce caractère de « défi » qui explique l'attrait représenté pour les chercheurs par la synthèse des générateurs de trajectoire.

Ayant nous-même subi cet attrait, nous pensons avoir apporté une contribution à l'étude de cette problématique de conception et à la définition de nouvelles méthodes. Il est vraissemblable que cette contribution soit, à l'instar des précédents travaux, une étape intermédiaire dans l'avancement de la recherche...

Annexe A

Glossaire de termes et de concepts

Mécanisme articulé plan (*linkage*) : mécanisme en chaînes cinématiques fermées, à topologie fixe, composé de pièces rigides connectées par des liaisons élémentaires parfaites, de types pivot et glissière.

Schéma cinématique : représentation abstraite d'un *mécanisme*, comportant des symboles graphiques pour les liaisons et des segments de droite pour les pièces (polygones si plus de deux liaisons sur une pièce).

Morphologie d'un mécanisme : représentation abstraite de la topologie des liaisons composant le mécanisme. Souvent représentée par un *graphe*.

Dimensions d'un mécanisme : valeurs de paramétrage du schéma cinématique.

Préconception des mécanismes articulés plans : étape préliminaire du processus de conception du mécanisme, consistant en le choix de la *morphologie* et des *dimensions* du schéma cinématique, en fonction d'un cahier des charges.

Liaison cinématique : contact entre les surfaces de deux pièces appartenant à un mécanisme. Une liaison est élémentaire si le mouvement relatif peut être décrit par un seul paramètre (pivots et glissières).

Liaison motrice (active) : liaison entre deux pièces entre lesquelles se manifeste une action motrice ; une des pièces est menante par rapport à l'autre et reçoit le mouvement ou l'énergie de l'extérieur du mécanisme.

Analyse : détermination du comportement d'un mécanisme dont la *morphologie* et les *dimensions* sont connues. Dans le contexte des méthodes de conception, on entend par *analyse* une simulation suivie de la comparaison du comportement du mécanisme avec le comportement spécifié par le cahier des charges fonctionnel.

Analyse cinématique : détermination des positions (éventuellement aussi vitesses et accelérations) en fonction du mouvement connu du (des) moteur(s). Synonyme souvent de *simulation cinématique*. Ne peut être faite que si le système est cinématiquement déterminé : nombre de moteurs égal au nombre de degrés de liberté.

Analyse dynamique inverse : s'applique à un système cinématiquement déterminé et utilise les résultats de l'analyse cinématique pour déterminer les efforts et les moments générateurs de cette cinématique. Cela suppose la connaissance de la loi horaire des moteurs, ce qui ne peut être qu'une approximation du comportement réel (dynamique).

Analyse dynamique : calcul des mêmes variables que dans l'analyse dynamique inverse, mais sur un système cinématiquement indéterminé, avec un nombre de moteurs plus petit que le nombre de degrés de liberté. Le système est ainsi gouverné non plus par des équations algébriques, mais par une combinaison d'équation différentielles et algébriques.

Analyse statique : détermination des positions d'équilibre et analyse d'efforts au repos. (Cela correspond à une minimisation de l'énergie potentielle du système).

Synthèse : méthode de conception qui consiste à déterminer la morphologie et les dimensions d'un ou de mécanismes à partir de son comportement désiré.

Organe effecteur : pièce ou solide du mécanisme qui remplit la fonction attendue du mécanisme.

Point effecteur : Point appartenant à l'organe effecteur qui effectue la tâche attendue, en particulier qui génère la trajectoire utile.

Trajectoire générée : trajectoire réellement décrite par le point effecteur pour un mécanisme de dimensions données.

Trajectoire désirée : trajectoire du point effecteur spécifiée par le concepteur.

Synthèse morphologique (*structural synthesis*) : détermination, en fonction des tâches à effectuer, de la *morphologie* du mécanisme, c'est-à-dire du nombre d'éléments (*number synthesis*), du nombre de liaisons et de leurs types (pivot, glissière), de la connectivité des éléments et des liaisons, choix des éléments qui seront le bâti et élément(s) moteur(s).

Synthèse dimensionnelle : détermination des *dimensions* d'un mécanisme, en termes de longueurs de barres ou coordonnées de liaisons, ainsi que de la position initiale, pour une morphologie connue, afin d'obtenir le comportement désiré.

Génération de trajectoire (*path generation*) : mouvement imposé d'un *point effecteur* le long d'une trajectoire.

Génération de mouvement (*motion generation* ou *rigid-body guidance*) : mouvement imposé pour l'organe effecteur (trajectoire d'un point effecteur et orientations successives de l'organe effecteur).

Génération de fonction (*function generation*) : corrélation imposée entre un mouvement d'entrée (du moteur) et un paramètre cinématique (rotation ou translation) de l'organe effecteur en sortie.

Dépendance du temps : dans le cas d'une génération de trajectoire, signifie que chaque position sur la *trajectoire désirée* doit correspondre à une position précise du moteur (le concepteur désire que le passage du point effecteur par les différents points de la trajectoire se fasse à des moments de temps précis).

Indépendance du temps : dans le cas d'une génération de trajectoire, signifie qu'aucune correlation n'est à respecter avec les positions correspondantes du moteur, c'est-à-dire que le point effecteur parcoure la trajectoire à n'importe quelle vitesse.

Groupe systémique multipolaire : sous-ensemble ou partie d'un mécanisme, identifiée selon des critères d'autonomie en termes de degrés de liberté.

Moto-élément : groupe multipolaire composé d'une manivelle.

Zéropôle : groupe multipolaire correspondant au bâti.

Diade : groupe multipolaire composée de deux barres reliées par une liaison.

Schéma multipolaire : représentation graphique du mode de connexion des groupes multipolaires dans le mécanisme.

Relation multipolaire : ordre de parcours des groupes permettant de résoudre à chaque étape les équations de cinématique pour chaque groupe en fonction de paramètres déjà calculés.

Trajectoire exacte : (ici) trajectoire générée par un point d'un mécanisme (notamment par le point effecteur), décrite sous la forme d'une liste de points (en coordonnées cartésiennes).

Trajectoire qualitative : (ici) représentation d'une trajectoire en arcs de cercle et segments de droite pas nécessairement de continuité C^0.

Graphe d'un mécanisme : représentation abstraite de la morphologie d'un mécanisme, dans laquelle les pièces sont représentées par des sommets et les liaisons par des arêtes du graphe.

Graphe planaire : cas particulier de graphe qui admet une représentation graphique dans le plan sans entrecroisement d'arêtes.

Dépliage de graphe : détermination d'une représentation graphique d'un graphe abstrait avec le moins possible d'entrecroisement d'arêtes (c'est-à-dire le plus lisible). Cela consiste à rechercher un positionnement approprié des sommets.

Optimisation dimensionnelle : utilisation d'un algorithme d'optimisation (locale ou globale), afin de minimiser une fonction objectif qui dépend du comportement du mécanisme, en modifiant un vecteur de conception composé de paramètres dimensionnels.

Optimisation exacte : (ici) optimisation dimensionnelle par bouclage sur des simulations cinématiques, lorsque les trajectoires générées, ainsi que la trajectoire désirée, sont des *trajectoires exactes*.

Optimisation qualitative : (ici) optimisation dimensionnelle par bouclage sur des simulations cinématiques, lorsque les trajectoires générées, ainsi que la trajectoire désirée, sont des *trajectoires qualitatives*.

Recalage géométrique : (ici) procédure consistant à appliquer à la géométrie du mécanisme une transformation géométrique de sorte que la trajectoire générée approche au mieux la trajectoire désirée. Pour cela, les paramètres de transformation optimaux sont déterminés par trois algorithmes différents, selon le mode de représentation des trajectoires : exactes, qualitatives ou par coefficients de Fourier.

Raisonnement à base de cas (CBR) : domaine de l'Intelligence Artificielle étudiant le mode de raisonnement consistant à utiliser l'expérience du passé pour la résolution de nouveaux problèmes.

Réseau neuronal artificiel (RNA) : organisation dans un graphe de cellules modélisées par une fonction et reliées par des connexions caractérisées par un *poids*. Utilisé en général pour obtenir une transformation souhaitée des signaux d'entrée en signaux de sortie. Les sorties dépendent de la morphologie du réseau, de la forme des fonctions et des poids.

Apprentissage du RNA : phase nécessaire avant l'utilisation d'un RNA, consistant en l'acquisition d'une expérience par la capitalisation d'un ensemble de jeux de données (valeurs d'entrée et de sortie). L'apprentissage est fait par l'optimisation des poids du RNA afin d'obtenir des signaux de sortie les plus proches possibles des valeurs dans le jeu de données.

Coefficients de Fourier : paramètres issus d'une analyse harmonique d'une fonction périodique et correspondants aux axes des ellipses. Utilisés pour décrire les trajectoires générées.

Cas : (ici) association d'une codification des dimensions (et, éventuellement, de la morphologie) d'un mécanisme et de la trajectoire générée par ce mécanisme.

Cas qualitatif : (ici) cas dans lequel les trajectoires sont codifier de manière qualitative (en particulier, par des *trajectoires qualitatives*).

Base de cas : ensemble de cas, stockés en vue de leur utilisation dans un système de raisonnement à base de cas.

Clé qualitative : (ici) représentation de très haut niveau, utilisant des symboles, de la forme d'une trajectoire générée. La clé qualitative est utilisée pour indexer les cas qualitatifs dans une *base de cas*.

Annexe B

Le raisonnement à base de cas

Le *Raisonnement à base de cas* (en anglais *Case-based Reasoning*, abrégé CBR) est une branche de l'Intelligence Artificielle datant d'environ 12 ans. Les débuts du CBR sont liés à des travaux sur des modèles psychologiques de la structure de la mémoire humaine. La discipline est en plein essor actuellement et ses applications couvrent des domaines très divers.

Le CBR part de la question très générale : comment mettre à contribution l'expérience du passé pour gérer le présent ? En fait, ce type d'approche a toujours été utilisé par l'homme pour trouver rapidement des solutions à des problèmes nouveaux. Ce qui est nouveau, c'est la formalisation et l'implémentation informatique de systèmes basés sur cette approche. En pratique, cela correspond à une consultation de tables ou de bases de données, afin de retrouver des cas ou des situations précédents plus ou moins proches du problème présent à résoudre. On peut donc définir le CBR comme *l'approche consistant à informatiser la comparaison d'un problème avec des cas précédents afin d'utiliser l'information contenue dans les cas retrouvés pour l'analyse et la résolution des nouveaux problèmes.*

Un *cas* rassemble généralement des informations de plusieurs types : *nom* (particularités), *faits* (ensemble de circonstances empiriques), *solution* (résultats du problème, décision, solution, classification), *contexte de validité* de la solution (alternatives, justifications, raisons des échecs), *liens avec d'autres cas*. La base de cas représente le stockage indexé des cas, sous forme de base de données le plus souvent. Les index représentent les circonstances dans lesquelles un cas peut avoir une utilité pour l'analyse ou la résolution d'un problème. Ils facilitent la recherche des cas pertinents et leur comparaison avec les problèmes à traiter.

Pour résumer, un *cas* (ou *problème*) représente l'association entre, d'une part, la description du problème (qu'on peut appeler, selon les domaines et le contexte d'utilisation, *quoi*, *fonction* ou *index*) et, d'autre part, la solution adoptée (*comment*, *structure*, *contenu* du cas).

B.1 Structure générique d'un système de Raisonnement à base de cas

La Figure B.1 présente l'organigramme général d'un système utilisant le raisonnement à base de cas. Les étapes principales sont les suivantes :

A Description du problème afin de le comparer aux entrées d'index.

B Recherche des cas-candidats dont les entrées d'index correspondent.

C Sélection des cas les plus similaires (si pas de cas similaire acceptable, tentative d'utiliser d'autres méthodes, après quoi on va à **F**).

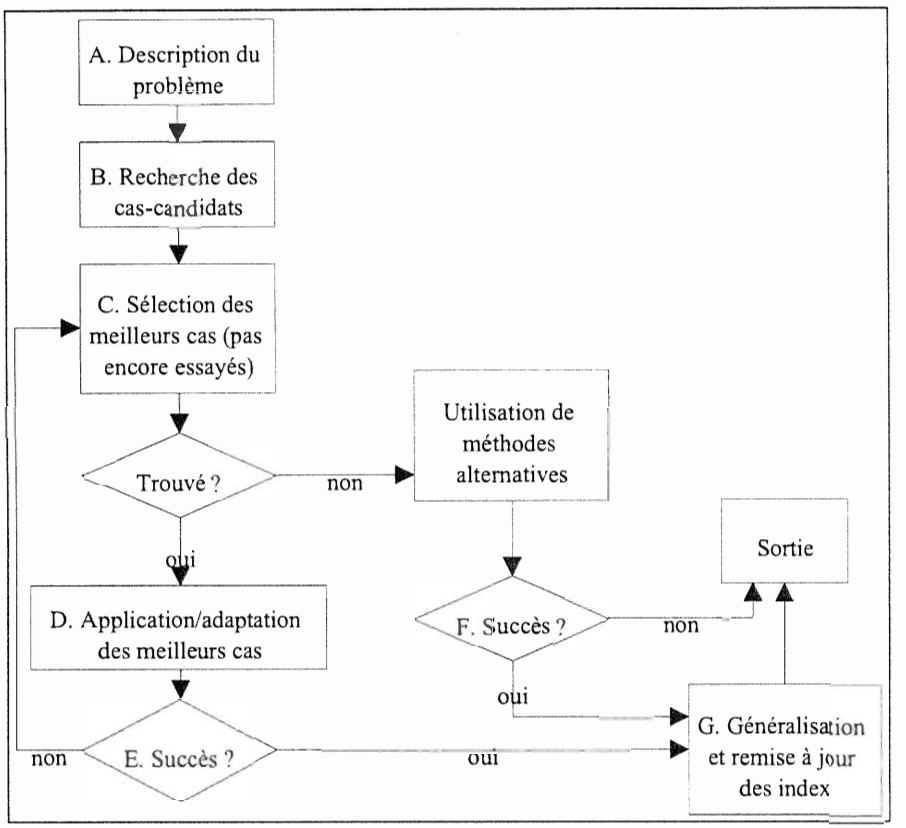


FIG. B.1 – *Organigramme d'un système de CBR typique.*

D Utilisation des meilleurs cas afin d'analyser/résoudre le problème. Si nécessaire, adaptation du cas au problème.

E Evaluation de la pertinence de la solution trouvée par rapport au problème (sinon, retour à **C** pour essayer d'autres cas candidats).

F Validation finale de la solution.

G Généralisation de la solution, remise à jour des index et FIN.

B.2 Domaines d'applications

Le CBR a été utilisé dans différents domaines. Une liste non exhaustive, comprenant un exemple suggestif de problématique, est la suivante :

- *Classification* : « Les troubles auditifs du patient ressemblent à ce cas prototype d'otite ».
- *Recherche de solutions* : « Les symptômes cardiaques du patient A sont à expliquer de la même manière que ceux de l'ancien patient B ».
- *Apprentissage par l'erreur* : « L'application de cette solution dans ce cas a conduit à l'échec, elle est donc à écarter ».
- *Extrapolation de valeurs* : « Ma maison ressemble à celle-là plus bas sur la rue, qui a été vendue 500 000 F, mais elle a une meilleure vue ».
- *Justification* : « Ce cas judiciaire doit être résolu de cette manière parce que dans un cas similaire le juge a déjà donné un tel verdict ».

- *Evaluation d'alternatives* : « Si on attaque les missiles russes à Cuba, ça va être comme à Pearl Harbor ».
- *Conception* : « On peut adapter notre ancien produit pour répondre à cet appel d'offre ».

B.3 CBR en conception

B.3.1 Rôle du CBR en Conception

Le raisonnement à base de cas a un intérêt tout à fait particulier dans le domaine de la conception, surtout en phase de préconception, pour trois raisons principales :

- les concepteurs créent le plus souvent des nouveaux produits par réutilisation de solutions ;
- les modèles analytiques existants sont généralement inadéquats, pas assez riches et mal maîtrisés pour servir à la conception ;
- les concepteurs ont l'habitude de garder une trace de leurs réussites et de leurs échecs et de les utiliser (dans une certaine mesure) dans le processus de conception.

On peut remarquer que ces trois aspects se retrouvent, en grande partie, dans la préconception de mécanismes : souvent le concepteur procède par modification d'un mécanisme existant.

Le contenu d'un cas en conception peut être de nature diverse, par exemple :

- tous les types d'informations créées durant la conception (composants, résultats de simulation, coûts des composants, etc) ;
- un seul type d'information (par exemple les composants) ;
- seulement les informations contenues dans le produit final pour chaque décision de conception, une alternative considérée et son résultat.

B.3.2 Fonctionnalités des systèmes de CBR en Conception

Une liste non-exhaustive des différentes fonctionnalités possibles d'un système de CBR dédié à la conception :

- Retrouver les produits existants (ou parties de produit) les plus proches des spécifications.
- Adapter un cas retrouvé pour le rendre adéquat avec le nouveau cahier des charges.
- Avertir l'utilisateur sur le risque d'échec sur la base des échecs précédemment enregistrés.
- Fournir à l'utilisateur les moyens d'essayer rapidement différentes solutions.
- Fournir les justifications rationnelles des décisions de conception.

B.4 Sources d'information

Un tutorial très complet, utilisé pour réaliser cette présentation du CBR, est *Ashley et al.* [ASS93]. Le centre de recherche sur le CBR le plus important est celui de *Carneggie Melon University* aux Etats-Unis (`http://www.cmu.edu`).

Annexe C

Présentation sommaire des réseaux neuronaux artificiels

Il n'est pas possible de présenter ici d'une manière complète les *réseaux de neurones artificiels* (RNA). Nous nous limiterons donc aux éléments nécessaires pour la compréhension de l'application qui en est faite dans REALISME.

Le développement des RNA dans les deux dernières décennies est issu de travaux sur la modélisation du fonctionnement du cerveau, réalisés dans les années quarante. En réalité, l'analogie entre les RNA et la biologie concerne plutôt la motivation originaire des chercheurs, car les RNA sont un modèle trop simple, très éloigné de la réalité biologique.

Le principe général reste néanmoins proche du fonctionnement du cerveau, car l'emploi des RNA comporte une phase d'acquisition d'informations (ou *cas*), suivie par une sorte de recherche dans cette base de cas. On remarque que ce contexte est semblable à celui typique pour une approche de *raisonnement à base de cas* (cf. l'annexe B). En cela, les RNA peuvent être vus comme une méthode de raisonnement à base de cas.

Actuellement, les RNA sont présentés le plus souvent comme un « modèle de calcul» (« *computational model* ») informatique, algorithmique ou hardware possédant des propriétés d'*adaptation*, d'*apprentissage*, de *généralisation*, de *classification* et d'*organisation* de l'information (cf. *Kröse et van der Smagt* [KvdS93]). Les RNA se caractérisent également par la tolérance aux erreurs ou au « bruit » des données. Cependant, selon certains points de vue, ces propriétés des RNA sont partagées avec d'autres méthodes mathématiques ; les RNA ne seraient donc qu'une façon particulière de mettre en œuvre des méthodes mathématiques déjà connues. Leur avantage est d'employer des méthodes générales, applicables en absence d'une formulation explicite d'une connaissance de cause à effet entre phénomènes, simplement par l'utilisation des données disponibles (par exemple les paramètres d'entrée et de sortie d'un système physique)[1].

Les applications pratiques des RNA sont très diverses. Dans la plupart des cas, elles concernent les problèmes qui peuvent être modélisés comme une mise en correspondance entre deux espaces vectoriels. Les applications les plus populaires sont la reconnaissance de caractères, le traitement de signal, le contrôle adaptatif, la robotique, les finances, etc. Quelques applications des RNA à la conception ont été présentées dans l'état de l'art, §2.5.3.

1. Cf. par exemple *Smith* [Smi96] : « *Neural networks cannot do anything that cannot be done using traditional computing techniques, but they can do some things wich would otherwise be very difficult. In particular, they can form a model from their training data alone.* »

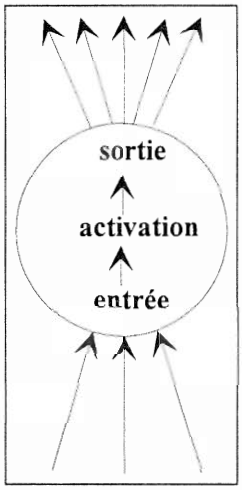


FIG. C.1 – *Circulation des signaux à travers un neurone. (D'après* Zell et al. *[ZMV+ 95].)*

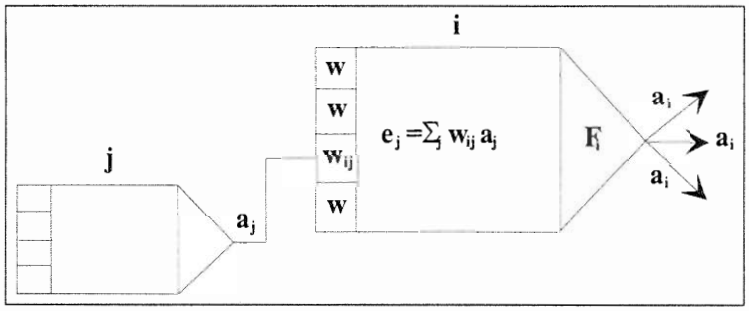


FIG. C.2 – *Structure générique d'un neurone artificiel. (D'après* Kröse *[KvdS93]).*

C.1 Structure d'un RNA

Un RNA consiste en un certain nombre d'unités élémentaires de calcul, appelées *neurones* ou *cellules*, entre lesquelles circulent des signaux – généralement numériques – à travers un grand nombre de *connexions* unidirectionnelles, caractérisées par un *coefficient synaptique* ou *poids*. En fait, chaque neurone est équivalent à une fonction calculant un résultat (unique) à partir d'une valeur (dite d'*activation*), calculée comme une somme pondérée (selon les *coefficients synaptiques*) des valeurs transmises par les neurones situés en amont (cf. Figure C.1). Selon la position des neurones dans le réseau, le résultat est ensuite transmis aux neurones situés en aval ou constitue le résultat global, qui est transmis à l'extérieur du réseau. Réciproquement, pour certains neurones le calcul de la fonction est fait à partir des données externes.

Ainsi, chaque neurone apporte à l'information qu'il reçoit une transformation qui dépend uniquement des données locales et des celles reçues par l'intermédiaire des connexions. Dans ce sens, les RNA sont un modèle *distribué*, adapté à la parallélisation du calcul, car beaucoup de calculs peuvent être faits en même temps, grâce à leur caractère local.

Plus en détail, voici la description du neurone en tant qu'automate :

- un certain nombre de *neurones* (unités de calcul) (voir Figure C.2) ;
- des *états d'activation* d'entrée a_j pour les neurones connectés en entrée ;
- les *coefficients synaptiques* ou *poids* w_{ij} affectés à ces états d'activation d'entrée a_j ;

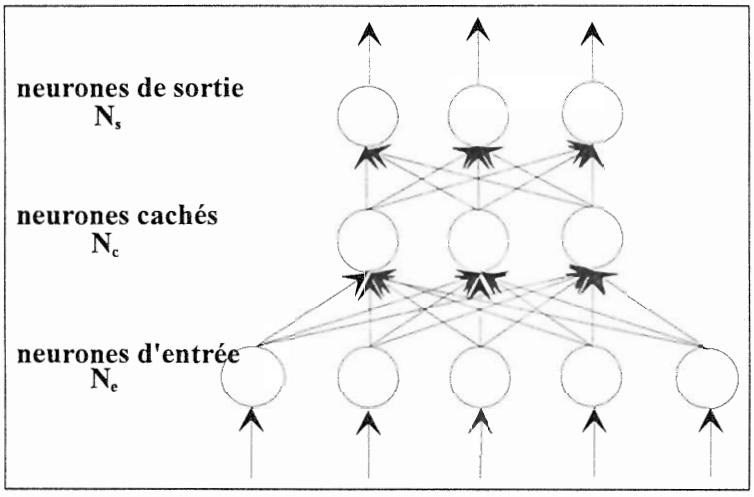


FIG. C.3 – *Structure générique d'un RNA de type* feedforward.

- une *fonction d'initialisation*, qui permet le calcul de l'entrée effective e_i du neurone par combinaison linéaire des entrées a_j et des poids w_{ij} (voir équation C.1) ;

$$e_i = \sum_j w_{ij}\, a_j \tag{C.1}$$

- une *fonction d'activation* F_i qui permet de calculer l'état d'activation propre a_i du neurone en fonction de l'entrée effective e_i :

$$a_i = F_i(e_i) \tag{C.2}$$

Souvent, est utilisée comme fonction d'activation la fonction sigmoïde :

$$a_i = F(e_i) = \frac{1}{1 + e^{-e_i}} \tag{C.3}$$

De plus, on doit définir pour un réseau de neurones :

- une méthode d'acquisition des cas (à partir d'expériences, de simulations, d'archives, etc.) ;
- un algorithme d'apprentissage, c'est-à-dire d'optimisation des coefficients synaptiques à partir de la base de cas ;
- une structure de réseau (graphe de connexion des neurones).

Le type de structure du réseau le plus répandu est le réseau *feedforward*, dans lequel le flux de données parcourt le réseau dans une direction bien déterminée, à partir des *neurones d'entrée* N_e vers les *neurones de sortie* N_s (voir Figure C.3). Les neurones d'entrée et de sortie sont souvent séparés par une ou plusieurs couches de neurones intermédiaires N_c, appelées *couches cachées*. Aucune connexion n'existe entre des neurones de la même couche. Les neurones d'entrée reçoivent les signaux d'entrée N_e et les transmettent aux neurones de la première couche cachée $N_{c,1}$, généralement sans aucune transformation (leur fonction d'activation est la fonction identité). La sortie des neurones cachés est transmise aux neurones $N_{c,2}$ de la couche suivante et ainsi de suite jusqu'à la couche de sortie N_s.

C.2 Apprentissage

Avant son utilisation, un RNA subit une phase d'*apprentissage* à partir d'exemples ou de cas représentant la connaissance qu'on a du domaine. Pour comprendre la notion d'apprentissage, prenons l'exemple d'un système physique comportant des paramètres d'entrée et des paramètres de sortie. Lorsque la loi qui relie les sorties aux entrées est inconnue mais un certain nombre d'exemples d'entrées et de sorties sont disponibles (soit de manière expérimentale, soit par calcul ou simulation), un RNA peut être utilisé pour apprendre à calculer les valeurs de sortie en fonction des valeurs d'entrée. Durant l'apprentissage, les coefficients synaptiques des connexions sont modifiés, de façon itérative, afin d'obtenir le comportement souhaité du réseau. Cette modification a comme objectif de minimiser l'erreur du réseau, c'est-à-dire la différence entre, d'une part, les valeurs obtenues en sortie par propagation dans le réseau des valeurs d'entrée et, d'autre part, les valeurs de sortie correspondant aux valeurs d'entrée.

Deux classes de méthodes ou algorithmes d'apprentissage existent : l'apprentissage *supervisé* (par la prise en compte des jeux d'entrées et de sorties) et *non-supervisé* (le réseau « découvre » des propriétés statistiquement pertinentes des entrées et produit des réponses classifiant les entrées sans que les catégories de classification soient établies à l'avance). Nous nous intéressons ci-après uniquement à l'apprentissage supervisé des RNA de structure *feedforward* avec l'algorithme d'apprentissage par rétropropagation (*backpropagation*).

La rétropropagation comporte deux phases :

1. propagation vers l'avant (*forward pass*) : par propagation dans le RNA d'un jeu de valeurs d'entrée, les valeurs de sortie s_j sont calculées ; pour chaque jeu d'entrées, est calculée l'erreur $\delta_j = s_j - t_j$, où t_j est la valeur de sortie souhaitée (correspondant aux données disponibles).

2. propagation vers l'arrière (*backward pass*) : durant cette phase, les erreurs δ_j et la sortie s_i d'un neurone i situé en entrée du neurone de sortie j sont utilisées pour calculer la modification appropriée (voir ci-après) du coefficient synaptique w_{ij}. Ensuite, l'erreur pour les neurones cachés est calculée par rétropropagation des erreurs δ_j à travers le réseau, de façon à pouvoir calculer la modification des autres coefficients synaptiques.

La rétropropagation utilise un ajustement des coefficients synaptiques basé sur l'erreur du gradient de la fonction d'activation F (cf. équation C.2), donné par la formule (cf. *Zell et al.* [ZMV+95]) :

$$\begin{aligned} \Delta w_{ij} &= \eta\, \delta_j\, s_i \\ \delta_j &= \begin{cases} F_j'(e_j)(t_j - o_j) & , \text{ si le neurone } j \text{ est en sortie} \\ F_j'(e_j) \sum_k \delta_k\, w_{jk} & , \text{ si le neurone } j \text{ est cachée} \end{cases} \end{aligned} \tag{C.4}$$

où

η	=	coefficient d'apprentissage (paramètre constant)
δ_j	=	erreur pour le neurone j
t_j	=	valeurs de sortie désirée du neurone j
e_j/s_j	=	entree/sortie du neurone i
i	=	indice de le neurone précédant le neurone courant j avec la connexion w_{ij} de i à j
j	=	indice du neurone courant
k	=	indice du successeur du neurone courant j avec la connexion w_{jk} de j à k

Lorsque la fonction d'activation est la fonction sigmoïde (cf. équation C.3), la dérivée F' est :

$$F'(e_i) = \frac{\partial}{\partial e_i} F(e_i) = F(e_i)(1 - F(e_i)) \tag{C.5}$$

C.3 Approximation de fonction et généralisation

Dans la littérature, il a été démontré qu'un RNA composé d'une couche cachée est capable d'approximer toute fonction $\boldsymbol{G}(\bar{\boldsymbol{x}})$ ($\boldsymbol{G} : \mathbf{R}^n \rightarrow \mathbf{R}^m$) avec une précision arbitraire[2]. Cela signifie que grâce à un apprentissage utilisant un certain nombre de couples de valeurs ($\bar{\boldsymbol{x}}$, $\boldsymbol{G}(\bar{\boldsymbol{x}})$) comme jeux de valeurs d'entrée et de sortie, le RNA est capable d'interpoler ces couples de valeurs. Cette capacité d'approximation de fonctions est appelée souvent capacité de *généralisation*, suggérant ainsi le fait qu'un RNA est capable de fournir un jeu de valeurs de sortie correct (à une précision près) pour des jeux de valeurs d'entrée qui n'ont pas été utilisés durant l'apprentissage.

L'algorithme d'apprentissage est un processus de minimisation de l'erreur sur un certain jeu de données. Cette erreur, appelée *erreur de représentation,* décrit la capacité du RNA à « mémoriser » les jeux d'entrées/sorties fournis lors de l'apprentissage. Elle est liée à la taille finie du RNA (nombre fini de neurones et de connexions).

Par rapport à l'erreur de représentation, il faut distinguer l'*erreur de généralisation,* c'est-à-dire l'erreur obtenue pour des jeux de données différents de ceux utilisés durant l'apprentissage. Cette erreur provient de la taille finie du jeu de données utilisé pour l'apprentissage.

Malheureusement, l'état du RNA (caractérisé par les coefficients synaptiques) minimisant l'erreur de représentation n'est par le même que celui minimisant l'erreur de généralisation. En effet, lorsque l'apprentissage est poursuivi trop longtemps, le RNA améliore le premier type d'erreur en dégradant le second, car il mémorise de mieux en mieux les particularités des cas individuellement, plutôt que de mémoriser ce qui les caractérise dans leur ensemble. Ce phénomène est appelé *sur-apprentissage* (*overfitting*).

Par conséquent, afin d'obtenir une capacité de généralisation maximale, les jeux de données disponibles sont souvent divisés en trois parties : :

- jeu de données d'apprentissage (*training set*), utilisé pendant l'apprentissage pour calculer l'erreur de représentation ;
- jeu de données de validation (*validation set*), utilisé pour déterminer l'erreur de généralisation ;
- jeu de données de test (*test set*), utilisé pour la vérification finale des performances du réseau.

Le rôle du jeu de données de validation est de permettre d'éviter le sur-apprentissage. Ainsi, après chaque itération de l'apprentissage, l'erreur de généralisation est calculée. Dans la première partie de l'apprentissage, cette erreur diminue, de même que l'erreur de représentation (mais plus lentement). Après un certain temps, une augmentation de l'erreur de généralisation apparaît. C'est à ce moment que l'apprentissage doit être arrêté. Autrement dit, le critère d'arrêt de l'apprentissage est basé sur l'erreur commise sur le jeu de données de validation.

C.4 Déficiences de la rétropropagation standard

L'algorithme standard de rétropropagation (cf. équation C.4) présente certains inconvénients. Le principal inconvénient est la durée très longue de l'apprentissage. Cette durée peut être due, en partie, à un mauvais choix du coefficient d'apprentissage η (cf. équation C.4), mais aussi à d'autres types de « dangers » :

- La *paralysie* du réseau : durant l'apprentissage, les poids des connexions peuvent devenir très grands, conduisant à des valeurs d'activation proches de 0 ou de 1 (cf. équation C.3) ; par conséquent, l'ajustement des coefficients tend à devenir nul (cf. équation C.5).

2. A condition toutefois que la fonction d'activation des neurones cachés soit non linéaire.

- Les *minima locaux* : la surface de la fonction erreur comporte un grand nombre de bosses et de vallées qui peuvent bloquer les algorithmes basés sur le gradient.
- Le *sur-apprentissage* (*overfitting*) (expliqué ci-avant, §C.3).

Par conséquent, un grand nombre d'algorithmes d'apprentissage, utilisant différentes lois de modification des coefficients, ont été mis au point. Ces algorithmes ont des caractéristiques très variées quant à la vitesse de convergence, à la capacité de sortir des minima locaux et à éviter la *paralysie* et le *sur-apprentissage*.

C.5 L'algorithme d'apprentissage *Rprop*

L'algorithme d'apprentissage pour les RNA utilisés dans REALISME est l'algorithme *Rprop (Resilient propagation)*. Il s'agit d'un algorithme d'apprentissage supervisé selon un schéma adaptif local (cf. [ZMV+95]). La fonction objectif comporte deux termes, le premier visant à réduire l'erreur sur les valeurs de sortie et le deuxième (appelé *weight-decay*) visant à réduire les poids des neurones afin d'améliorer les capacités de généralisation (voir §C.3). Ces deux termes sont pondérés par un coefficient, l'exposant α. Ainsi, la fonction d'erreur s'écrit :

$$E = \sum (t_i - s_i)^2 + 10^{-\alpha} \sum w_{ij}^2 \tag{C.6}$$

L'algorithme élimine l'influence de la dérivée partielle sur la modification des poids des neurones. Pour cela, seulement le signe de la dérivée est pris en compte comme indication de la direction de modification des poids. La taille de la modification dépend uniquement d'une « valeur de mise à jour » $\Delta_{ij}^{(t)}$:

$$\Delta w_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)} & , \text{ si } \frac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ +\Delta_{ij}^{(t)} & , \text{ si } \frac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ 0 & , \text{ sinon} \end{cases} \tag{C.7}$$

où $\frac{\partial E}{\partial w_{ij}}^{(t)}$ est l'information de gradient sommée sur tous les jeux de données.

La deuxième étape de *Rprop* est le calcul de la nouvelle valeur de mise à jour $\Delta_{ij}(t)$ selon un processus adaptatif dépendant du signe :

$$\Delta_{ij}^{(t)} = \begin{cases} \eta^+ \Delta_{ij}^{(t-1)} & , \text{ si } \frac{\partial E}{\partial w_{ij}}^{(t-1)} \cdot \frac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ \eta^- \Delta_{ij}^{(t-1)} & , \text{ si } \frac{\partial E}{\partial w_{ij}}^{(t-1)} \cdot \frac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ \Delta_{ij}^{(t-1)} & , \text{ sinon} \end{cases} \tag{C.8}$$

$$\text{où } 0 < \eta^- < 1 < \eta^+$$

Le principe de cette méthode est d'adapter l'apprentissage à la topologie de la fonction d'erreur. A chaque changement de signe de la dérivée partielle du coefficient w_{ij} (ce qui indique que la dernière valeur de mise à jour était trop grande et que l'algorithme vient de sortir d'un minimum local), la valeur de mise a jour $\Delta_{ij}^{(t)}$ est diminuée par le facteur η^-. Si la dérivée garde le même signe, la valeur de mise à jour est augmentée afin d'accélérer la convergence dans les régions « plates ». En cas de changement de signe, il n'y a pas d'adaptation du pas d'apprentissage.

Dans le logiciel *SNNS*, utilisé pour l'apprentissage du RNA pour la méthode de synthèse présentée au chap. 9, l'utilisateur doit choisir trois paramètres de l'algorithme *Rprop* : la valeur initiale pour les Δ_{ij}, la borne supérieure $\Delta_{\max}$ pour Δ_{ij} et le paramètre de pondération σ. *Rprop* ne

semble pas très sensible par rapport aux valeurs de ces paramètres ; en tous cas il admet des variations relativement importantes autour des valeurs conseillées, respectivement $\mathbf{\Delta}_{ij} = \mathbf{0,1}$, $\mathbf{\Delta}_{\max} = \mathbf{50}$ et $\boldsymbol{\sigma} = \mathbf{4}$. D'autres paramètres ont des valeurs fixées a priori : $\boldsymbol{\eta}^{+} = \mathbf{1,2}$, $\boldsymbol{\eta}^{-} = \mathbf{0,5}$, $\mathbf{\Delta}_{\min} = \mathbf{10^{-6}}$ (cf. *Zell et al.* [ZMV+95]).

Bibliographie

[AA90] R. AKHRAS et J. ANGELES. « Unconstrained nonlinear least-square optimization of planar linkages for rigid-body guidance ». *Mechanism and Machine Theory*, 25(1):97–118, 1990.

[AAA88] J. ANGELES, A. ALIVIZATOS et R. AKHRAS. « An unconstrained nonlinear least-square method of optimization of RRRR planar path generators ». *Mechanism and Machine Theory*, 23(5):343–353, 1988.

[AAGdJ85] R. AVILÉS, M. B. AJURIA et J. Garcia de JALON. « A fairly general method for optimum synthesis of planar mechanisms ». *Mechanism and Machine Theory*, 20(4):321–328, 1985.

[ANAH94] R. AVILÉS, S. NAVALPOTRO, E. AMEZUA et A. HERNANDEZ. « An energy-based general method for the optimum synthesis of mechanisms ». *Journal of Mechanical Design*, 116:127–136, 1994.

[AR87] V. AGRAWAL et J. RAO. « The mobility properties of kinematic chains ». *Mechanism and Machine Theory*, 22(5):497–504, 1987.

[Art75a] I. ARTOBOLEVSKI. *Les mécanismes dans la technique moderne.* Editions de Moscou, 5 vol., 1975.

[Art75b] I. ARTOBOLEVSKI. *Théorie des mécanismes et des machines.* Editions Mir, Moscou, 1975.

[ASR75] R. I. ALIZADE, G. SANDOR et A. V. RAO. « Optimum synthesis of four-bar and offset slider-crank planar and spatial mechanisms using the penalty function approach with inequality and equality constraints ». *Trans. ASME Journal of Engineering in Industry*, 97(3):785–790, 1975.

[ASS93] K. ASHLEY, E. SIMOUDIS et K. SYCARA. « Case-based Reasoning : theory and practice ». In *IJCAI'93*, Chambéry, France, 1993.

[Bak81] J. E. BAKER. « On mobility and relative freedoms in multiloop linkages and structures ». *Mechanism and Machine Theory*, 16(6):583–597, 1981.

[Bar85] C. R. BARKER. « A complete classification of planar four-bar linkages ». *Mechanism and Machine Theory*, 20(6):565–554, 1985.

[BB93] C. BAGCI et D. BURKE. « Optimum synthesis of coupler curve and uniform rotary motion driven multiloop mechanisms generating complex output motions ». *Journal of Mechanical Design*, 115(12):967–977, 1993.

[BD91] J. BLAND et G. DAWSON. « Tabu search and design optimization ». *Computer-Aided Design*, 23(3):195–201, 1991.

[BDBL95] P. BERTOLAZZI, G. DI BATTISTA et G. LIOTTA. « Parametric graph drawing ». *IEEE Transactions on software engineering*, 21(8):662–673, 1995.

[Ber58] C. BERGE. *Théorie des graphes et ses applications.* Dunod, Paris, 1958.

[Ber73] C. BERGE. *Graphes et hyper-graphes.* Dunod, Paris, 1973.

[BGT92] C. BOHATIER, J. GUILLOT et B. TROUSSE. « Approche fonctionnelle de la conception et définition d'un atelier de conception intelligent ». In *La conception en l'An 2000 et au-delà : outils et technologies*, Strasbourg, 1992.

[BH67] R. BOGDAN et T. HUNCHER. « General systematization and unified calculation of five- and four-bar plane basic mechanisms ». *Journal of Engineering in Industry, Trans. ASME*, 89:189–198, 1967.

[BK85] A. BAJBAI et S. KRAMER. « Detection and elimination of mechanism defects in the selective precision synthesis of planar mechanisms ». *Mechanism and Machine Theory*, 20(6):521–534, 1985.

[BK95] S. BAWAB et G. L. KINZEL. « A hybrid approach to solving the position equations for planar mechanisms ». *Journal of Mechanical Design*, 117:627–632, 1995.

[BL93] R. BRAUNE et J. LÜCKE. « Effektive Auslegung von Koppelgetrieben durch Integration von genaulagen-Synthese und allgemeiner Optimierung ». *Mechanism and Machine Theory*, 28(4):507–514, 1993.

[BO92] B. BAHGAT et M. OSMAN. « The parametric coupler curve equations of an eight link planar mechanism containing revolute and prismatic joints ». *Mechanism and Machine Theory*, 27(2):201–211, 1992.

[BP94] N. P. BELFIORE et E. PENNESTRI. « Automatic sketching of planar kinematic chains ». *Mechanism and Machine Theory*, 29(1):177–193, 1994.

[Bru94] M. W. BRUNKHART. « Interactive geometric constraint systems ». Master's project, University of California, Berkeley, Mai 1994.

[BS] R. G. BUSACKER et T. L. SAATY. *Finite graphs and networks : an introduction with applications.* McGraw-Hill Book Company.

[BU86] J. L. BLECHSCHMIDT et J. J. UICKER. « Linkage synthesis using algebraic curves ». *Journal of Mechanisms, Transmissions, and Automation in Design*, 108:543–548, Déc. 1986.

[Can96] O. CANTZLER. « *Une architecture conceptuelle pour la pérénisation d'historiques globaux de conception de produits industriels complexes* ». Thèse de doctorat, Ecole Centrale Paris, 1996.

[CC] T. CHEN et R. CHEN. « Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems ». *IEEE Transactions on Neural Networks.* manuscrit accepté.

[CC91] T. CHEN et H. CHEN. « Approximation capability in $C(\bar{R}^n)$ by multilayer feedforward networks and related problems ». *IEEE Transactions on Neural Networks*, 1991. manuscrit accepté.

[CC95] D. F. COOK et C.-C. CHIU. « Combining a radial basis neural network with time series analysis techniques to predict manufacturing process parameters ». *Applied Artificial Intelligence*, 9:623–631, 1995.

[CCM92] C. CONTI, P. CORRON et P. MICHOTE. « A computer-aided kinematic analysis system for mechanism design and computer simulation ». *Mechanism and Machine Theory*, 27(5):563–574, 1992.

[CDPS92] V. COSSALTER, A. DORIA, M. PASINI et C. SCATTOLO. « Simple numerical approach for optimum synthesis of a class of planar mechanisms ». *Mechanism and Machine Theory*, 27(3):357–366, 1992.

[CF84] W. L. CLEGHORN et R. G. FENTON. « Optimum synthesis of an angular function generating mechanism with prescribed time ratio and rocker angular swing amplitude». *Mechanism and Machine Theory*, 19(3):319–324, 1984.

[CGMS75] F. L. CONTE, G. R. GEORGE, R. W. MAYNE et J. P. SADLER. « Optimum mechanism design combining kinematic and dynamic-force considerations ». *Journal of Engineering for Industry*, 5:662–670, 1975.

[CH88] W.-H. CHIENG et D. A. HOELTZEL. « Computer-aided kinematic analysis of planar mechanisms based on symbolic pattern matching of independent kinematic loops ». 15(1):197–207, 1988.

[CH90] W.-H. CHIENG et D. HOELTZEL. « A combinatorial approach for the automatic sketching of planar kinematic chains and epicyclic gear trains». *Transaction of the ASME*, 112(3):6–14, 1990.

[Chr75] N. CHRISTOFIDES. *Graph theory. An algorithmic approach.* Londres, Academic Press, 1975.

[CP95] M. CHROBAK et T. PAYNE. « A linear-time algorithm for drawing a planar graph on a grid ». *Information Processing Letters*, 54:241–246, 1995.

[CR82] M. CHOUBEY et A. C. RAO. « Synthesizing linkages with minimal structural and mechanical error based upon tolerance allocation ». *Mechanism and Machine Theory*, 17(2):91–97, 1982.

[CS71] F. CROSSLEY et N. SESHACHAR. « Analysis of the displacement of planar Assur groups by computer ». In *Proceedings of the 3rd World Congress TMM*, p. 71–82, 1971.

[CSI95] M. CHEW, S. N. T. SHEN et G. F. ISSA. « Kinematic structural synthesis of mechanisms using knowledge-based systems». *Journal of Mechanical Design*, 1995(3):96–103, 1995.

[Dav81] T. H. DAVIES. « Kirkhhoff's circulation law applied to multi-loop kinematic chains ». *Mechanism and Machine Theory*, 16:171–183, 1981.

[DD94] M. DANIEL et C. DEMOISSON. « Vers la modélisation déclarative de courbes ». In *Colloque Priméca*, p. 11–16, Nantes, 7 juillet 1994.

[Der71] J.-C. DERNIAME. *Problèmes de cheminement dans les graphes.* Dunod, Paris, 1971.

[dKB84] J. de KLEER et J. S. BROWN. « A qualitative physics based on confluences». *Artificial Intelligence*, 24:7–83, 1984.

[DL96] M. DANIEL et M. LUCAS. « Towards declarative geometric modeling in mechanics ». In *Proceedings of IDMME'96*, p. 455–464, Ecole Centrale de Nantes, 15-17 Avril 1996.

[DM93a] A. K. DHINGRA et N. K. MANI. « Computer-aided mechanism design : a symbolic-computing approach ». *Computer-Aided Design*, 25(5):300–310, 1993.

[DM93b] A. K. DHINGRA et N. K. MANI. « Finitely and multiply separated synthesis of link and geared mechanisms using symbolic computing ». *Journal of Mechanical Design*, 115(9):560–567, 1993.

[DR91] A. K. DHINGRA et S. S. RAO. « An integrated kinematic-kinetostatic approach to optimal design of planar mechanisms using fuzzy theories ». *Journal of Mechanical Design*, 113(9):306–311, 1991.

[DS93] B. DASGUPTA et G. SCHNITGER. The power of approximating : a comparaison of activation functions. In C. GILES, S. HANSON et J. COWAN, éditeurs, *Advances in Neural Information Processing Systems*, v. 5. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[DSR81] S. DE SA et B. ROTH. « Kinematic mappings – part 1 : Classification of algebraic motions in the plane; – part 2 : Rational algebraic motions in the plane ». *Journal of Mechanical Design*, 103:585–591; 712–717, 1981.

[DT94] E. A. DIJKSMAN et E. A. TIMMERMANS. «Look-out for prime-chains with a prescribed number of mobility-degrees of freedom. Type synthesis of prime-chains ». *Mechanism and Machine Theory*, 29(5):653–672, 1994.

[DY95] P. DAGUE et B. e. a. YANNOU. « Qualitative reasoning ». *Artificial Intelligence Communications*, 8(3–4):119–192, 1995.

[End90] E. ENDERTON. « Interactive type synthesis of mechanisms ». Master's thesis UCB/CSD90/570, University of California, Berkeley, Avril 1990.

[Era79] N. F. ERASLAN. « On the synthesis of certain straight-line mechanisms ». *Mechanism and Machine Theory*, 14:299–307, 1979.

[Erd81] A. G. ERDMAN. « Three and four precision point kinematic synthesis of planar linkages ». *Mechanism and Machine Theory*, 16:227–245, 1981.

[Erd85] A. G. ERDMAN. « Computer-aided design of mechanisms: 1984 and beyond ». *Mechanism and Machine Theory*, 20(4):245–249, 1985.

[Erd95] A. G. ERDMAN. « Computer-aided mechanism design : now and the future ». *Transactions of the ASME, Journal of Mechanical Design, Special 50th Anniversary Design Issue*, 117:93–100, 1995.

[ES84] A. G. ERDMAN et G. N. SANDOR. *Mechanism design : analysis and synthesis*, v. 1. Prentice-Hall, 1984.

[Fal87] B. FALTINGS. « Qualitative kinematics in mechanisms ». In *IJCAI-87*, p. 436–442, 1987.

[Fal92] B. FALTINGS. « A symbolic approach to qualitative kinematics ». *Artificial Intelligence*, 56:139–170, 1992.

[FF90] W. E. FANG et F. FREUDENSTEIN. « A hierarchical approach to the automatic sketching of mechanisms ». In A. PISANO, éditeur, *The 1990 ASME Design Technical Conferences - 21st biennal mechanisms conference*, v. 26, p. 341–347, Chicago, Illinois, 16-19 Sep., 1990.

[FM79] F. FREUDENSTEIN et E. R. MAKI. « The creation of mechanisms according to kinematic structure and function ». *Environment and Planning B*, 6:375–391, 1979.

[FM84] F. FREUDENSTEIN et E. R. MAKI. « Kinematic structure of mechanisms for fixed and variable-stroke axial-piston reciprocating machines ». *Journal of Mechanisms, Transmissions, and Automation in Design*, 106(9):355–364, 1984.

[FNF87] K. D. FORBUS, P. NIELSEN et B. FALTINGS. « Qualitative kinematics : a framework ». In *IJCAI-87*, p. 430–436, 1987.

[FNF91] K. D. FORBUS, P. NIELSEN et B. FALTINGS. « Qualitative spatial reasoning : the Clock Project ». *Artificial Intelligence*, 51:417–471, 1991.

[FS93a] B. FALTINGS et K. SUN. « Computer-aided creative mechanism design ». In *IJCAI-93*, p. 1451–1457, Chambéry, France, 1993.

[FS93b] B. FALTINGS et K. SUN. « Causal inversion : applying kinematic principles to mechanism design ». In *13th International Joint Conference on Artificial Inteligence, IJCAI'93*, p. 81–86, Chambéry, France, 1993.

[FS96a] B. FALTINGS et K. SUN. « FAMING: supporting innovative mechanism shape design ». *Computer-Aided Design*, 28(3):207–216, 1996.

[FS96b] J.-C. FAROUX et M. SARTOR. « Geometrical definition of a speed reducer skeleton ». In *Proceedings of IDMME'96*, p. 787–796, Ecole Centrale de Nantes, 1996.

[GA87] C. GOSSELIN et J. ANGELES. « Représentation graphique de la région de mobilité des mécanismes plans et sphériques à barres articulées ». *Mechanism and Machine Theory*, 22(6):557–562, 1987.

[GA88] C. GOSSELIN et J. ANGELES. « The optimum kinematic design of a planar three-degree-of-freedom parallel manipulator ». *Journal of Mechanisms, Transmissions, and Automation in Design*, 110(3):35–41, 1988.

[Gal86] C. U. GALLETTI. « A note on modular approaches to planar linkage kinematic analysis ». *Mechanism and Machine Theory*, 21(5):385–391, 1986.

[GG81] C. U. GALLETTI et E. I. GIANNOTTI. « Interactive computer system for the functional design of mechanisms ». *Computer-Aided Design*, 13(3):159–163, 1981.

[GH68] R. E. GARRET et A. S. HALL. « Optimal synthesis of randomly generated linkages ». *Journal of Engineering for Industry, Trans. ASME*, 90B:475–480, 1968.

[GM85] M. GONDRAN et M. MINOUX. *Graphes et algorithmes*. Eyrolles, Paris, 1985.

[GMX96] C. G. GIBSON, D. MARSH et Y. XIANG. « An algorithm to generate the transition curve of the planar four-bar mechanism ». *Mechanism and Machine Theory*, 31(4):381–395, 1996.

[Goh95] A. T. C. GOH. « Back-propagation neural networks for modeling complex systems ». *Artificial Intelligence in Engineering*, 9:143–151, 1995.

[GR91] C. GARNIER et P. RIDEAU. « Utilisation du calcul formel. Une application en Maple : le logiciel JAMES». *Revue internationale de CFAO et d'Infographie*, 6(1):55–69, 1991.

[GSF87] M.-C. GAUDEL, M. SORIA et C. FROIDEVAUX. *Types de données et algorithmes.* INRIA, 1987.

[Han92] M. R. HANSEN. « *Development and implementation of methods for analysis and synthesis of mechanical mechanisms* ». Thèse de doctorat, Institute of Mechanical Engineering, Aalborg University, 1992.

[Han93] J. M. HANSEN. « Synthesis of spatial mechanisms using optimization and continuation methods ». In *NATO–Advanced Study Institute on Computer Aided Analysis of Rigid and Flexible Mechanical Systems*, p. 441–453, Troia, Portugal, 1993.

[Han96] M. R. HANSEN. « A general method for analysis of planar mechanisms using a modular approach ». *Mechanism and Machine Theory*, 31(8):1155–1166, 1996.

[HC90a] D. A. HOELTZEL et W.-H. CHIENG. « Pattern matching synthesis as an automated approach to mechanism design ». *Journal of Mechanical Design*, 112:190–199, 1990.

[HC90b] D. HOELTZEL et W.-H. CHIENG. « Knowledge-based approaches for the creative synthesis of mechanisms ». *Computer-Aided Design*, 22(1):57–67, 1990.

[HD64] R. HARTENBERG et J. DENAVIT. *Kinematic synthesis of linkages.* McGraw-Hill, New York, 1964.

[Hei95] A. P. HEINZ. « A tree-structured neural network for real-time adaptive control ». Rapport de recherche, Institut d'informatique, Université de Freiburg, 1995.

[HFS96] K. HUA, B. FALTINGS et I. SMITH. « CADRE : case-based geometric design». *Artificial Intelligence in Engineering*, 10:171–183, 1996.

[HH92] W.-M. HWANG et Y.-W. HWANG. « Computer-aided structural synthesis of planar kinematik chains with simple joints ». *Mechanism and Machine Theory*, 27(2):189–199, 1992.

[HN51] J. A. HRONES et G. L. NELSON. *Analysis of the four-bar linkage.* The Technology Press of the MIT and John Wiley and Sons, Inc., New York, 1951.

[HR93] C. HUANG et B. ROTH. « Dimensional synthesis of closed-loop linkages to match force and position specifications ». *Journal of Mechanical Design*, 115(7):194–198, 1993.

[Hsu92] C.-H. HSU. « Generalization of mechanical devices with any number of general constraints ». *Mechanism and Machine Theory*, 27(5):613–621, 1992.

[HT74] J. HOPCROFT et R. TARJAN. « Efficient planarity testing». *Journal of the Association for Computing Machinery*, 21(4):549–568, 1974.

[HY90] F. HARARY et H.-S. YAN. « Logical foundations of kinematic chains : graphs, line graphs and hypergraphs ». *Journal of Mechanical Design*, 112(3):79–83, 1990.

[Inn94] C. INNOCENTI. « Analytical-form position analysis of the 7-link Assur kinematic chain with four serially-connected ternary links». *Journal of Mechanical Design*, 116(7):622–628, 1994.

[JA88] L. JOSKOWICZ et S. ADDANKI. « From kinematics to shape : an approach to innovative design ». In *Seventh National Conference on Artificial Intelligence*, p. 347–352, St. Paul, Minnesota, USA, 1988.

[Jen84] P. W. JENSEN. « Synthesis of four-bar linkages with a coupler point passing through 12 points ». *Mechanism and Machine Theory*, 19(1):149–156, 1984.

[Jos89] L. JOSKOWICZ. « Reasoning about the kinematics of mechanical devices ». *Artificial Intelligence in Engineering*, 4(1):22–31, 1989.

[JR93] P. A. JAMES et B. ROTH. « A unified theory for finite, infinitesimal, and hybrid kinematic synthesis ». *Journal of Mechanical Design*, 115(9):552–559, 1993.

[JR94] P. A. JAMES et B. ROTH. « A unified theory for kinematic synthesis ». *Journal of Mechanical Design*, 116(3):145–153, 1994.

[JS91] L. JOSKOWICZ et E. P. SACKS. « Computational kinematics ». *Artificial Intelligence*, 51:381–416, 1991.

[JS93] L. JOSKOVICZ et E. SACKS. « Automated modeling and kinematic simulation of mechanisms ». *Computer-Aided Design*, 25(2):106–118, 1993.

[JvdSG95] A. JANSEN, P. van der SMAGT et F. GROEN. Nested networks for robot control. In A. F. MURRAY, éditeur, *Application of Neural Networks*, p. 221–239. Kluwer Academic Publishers, 1995.

[KC84] G. L. KINZEL et C. CHANG. « The analysis of planar linkages using a modular approach ». *Mechanism and Machine Theory*, 19(1):165–172, 1984.

[KC93] S. KOTA et S.-J. CHIOU. « Use of orthogonal arrays in mechanism synthesis ». *Mechanism and Machine Theory*, 28(6):777–794, 1993.

[KER87] S. KOTA, A. G. ERDMAN et D. R. RILEY. « Development of knowledge base for designing linkage-type dwell mechanisms: Part 1 – Theory; Part 2 – Application ». *Journal of Mechanisms, Transmissions, and Automation in Design*, 109(9):308–328, 1987.

[KER88] S. KOTA, A. G. ERDMAN et D. R. RILEY. « Minn-Dwell–Computer aided design and analysis of linkage-type dwell mechanisms ». *Mechanism and Machine Theory*, 23(6):423–433, 1988.

[KH90] S. KUMARA et I. HAM. « Use of associative memory and self-organization in conceptual design ». *Annals of the CIRP*, 39(1):117–120, 1990.

[KK89] T. KAMADA et S. KAWAI. « An algorithm for drawing general undirected graphs ». *Information Processing Letters*, 31:7–15, 1989.

[Koc92] I. S. KOCHEV. « Qualitative theory of the reactions and stresses in high speed planar linkages ». *Mechanism and Machine Theory*, 27(1):59–68, 1992.

[Kot92] S. KOTA. « Automatic selection of mechanism designs from a three-dimensional design map ». *Journal of Mechanical Design*, 114:359–367, Sep. 1992.

[Kra92] G. KRAMER. *Solving geometric constraint systems.* MIT Press, Cambridge, MA, 1992.

[KS75] S. KRAMER et G. SANDOR. « Selective precision synthesis - A general method of optimization for planar mechanisms ». *Trans. ASME Journal of Engineering in Industry*, 97(2):689–701, 1975.

[KS92] A. KUSIAK et E. SZCZERBICKI. « A formal approach to specifications in conceptual design ». *Journal of Mechanical Design*, 114:659–666, Déc. 1992.

[KT92] S. KRISHNAMURTY et D. A. TURCIC. « Optimal synthesis of mechanism using nonlinear goal programming techniques». *Mechanism and Machine Theory*, 27(5):599–612, 1992.

[Kun72] J. KUNTZMANN. *Théorie des réseaux (Graphes)*. Dunod, Paris, 1972.

[KvdS93] B. J. A. KRÖSE et P. P. van der SMAGT. « An introduction to neural networks ». Rapport de recherche, University of Amsterdam, Faculty of Mathematics and Computer Science, 1993.

[LCK87] B. T. LAUW, G. CHANDRASEKHAR et G. L. KINZEL. « Automatic component selection algorithm for kinematic analyses ». In *10th Applied Mechanisms Conference*, v. III, New Orleans, LA, 1987.

[Lee76] T. W. LEE. « Heuristic combinatorial optimisation in the kinematic design of mechanisms – part 1 : Theory, – part 2 : Applications ». *Journal of Engineering for Industry*, 98(11):1277–1284, 1976.

[LG86] L. LIVOVSCHI et H. GEORGESCU. *Sinteza şi analiza algoritmilor*. Editura Ştiinţifică şi Enciclopedică, Bucarest, 1986.

[LIA95] J. LIU, M. IGOSHI et E. ARAI. « Kinematic simulation using qualitative mechanism models». In *1995 Design Engineering Technical Conferences*, v. 2, p. 679–686, Boston, Massachusetts, 1995.

[Lit80] F. L. LITVIN. « Application of theorem of implicit function system existence for analysis and synthesis of linkages ». *Mechanism and Machine Theory*, 15(2):115–120, 1980.

[LM94] K. LUCK et K.-H. MODLER. « Synthesis of guidance mechanisms ». *Mechanism and Machine Theory*, 29(4):525–533, 1994.

[LM95] K. LUCK et K.-H. MODLER. « Burmester theory for four-bar-band mechanisms ». *Journal of Mechanical Design*, 117(3):129–133, 1995.

[LMR85] K. LUCK, K.-H. MODLER et J. REBER. « Computer-aided design in mechanisms ». *Mechanism and Machine Theory*, 20(4):297–302, 1985.

[LP90] T. LEINONEN et A. PAARNI. « AUCASYN – an automatic computer-aided synthesis program for mechanisms and machines ». In A. PISANO, éditeur, *The 1990 ASME Design Technical Conferences - 21st biennial mechanisms conference*, v. 26, p. 367–371, Chicago, Illinois, 16-19 September, 1990. ASME.

[LTC96] C. L. LI, S. T. TAN et K. W. CHAN. « A qualitative and heuristic approach to the conceptual design of mechanisms». *Engineering Applications of Artificial Intelligence*, 9(1):17–31, 1996.

[LY95] T. LIU et C.-H. YU. « Identification and classification of multi-degree-of-freedom and multi-loop mechanisms ». *Mechanism and Machine Theory*, 117(3):104–111, 1995.

[Man68] N. MANOLESCU. « For a united point of view in the study of the structural analysis of kinematic chains and mechanisms ». *Journal of Mechanical Design*, 3:149–169, 1968.

[May72] W. MAYEDA. *Graph theory.* Wiley-Interscience, New York, 1972.

[McG94] J. R. MCGARVA. « Rapid search and selection of path generating mechanisms from a library ». *Mechanism and Machine Theory*, 29(2):223–235, 1994.

[Mer95] J.-P. MERLET. « *4barres user's manual* ». INRIA Sophia-Antipolis, 1995.

[MF84] M. MAYOURIAN et F. FREUDENSTEIN. « The development of an atlas of the kinematic structures of mechanisms ». *Journal of Mechanisms, Transmissions, and Automation in Design*, 106(12):458–461, 1984.

[Min89] M. MINOUX. *Programmation mathématique. Théorie et algorithmes*, v. 1. Dunod, Paris, 1989.

[MK96a] J. MARIAPPAN et S. KRISHNAMURTY. « A generalized exact gradient method for mechanism synthesis ». *Mechanism and Machine Theory*, 31(4):413–421, 1996.

[MK96b] S. MAUSKAR et S. KRISHNAMURTY. « A loop configuration approach to automatic sketching of mechanisms ». *Mechanism and Machine Theory*, 31(4):423–437, 1996.

[Mly96] T. MLYNARSKI. « Position analysis of planar linkages using the method of modification of kinematic units ». *Mechanism and Machine Theory*, 31(6):831–834, 1996.

[MM93] J. MCGARVA et G. MULLINEUX. « Harmonic representation of closed curves ». *Applied mathematical modelling*, 17(4):213–218, 1993.

[MMSM90] F. MISTREE, D. MUSTER, S. SRINIVASAN et S. MUDALI. « Design of linkages : a conceptual exercise in designing for concept ». *Mechanism and Machine Theory*, 25(3):273–286, 1990.

[MR84] T. MRUTHYUNJAYA et M. RAGHAVAN. « Computer-aided analysis of the structure of kinematic chains ». *Mechanism and Machine Theory*, 19(3):357–368, 1984.

[MRE85] W. A. MITTELSTADT, D. R. RILEY et A. G. ERDMAN. « Integrated CAD of mechanisms ». *Mechanism and Machine Theory*, 20(4):303–311, 1985.

[Mru79] T. MRUTHYUNJAYA. « Structural synthesis by transformation of binary chains ». *Mechanism and Machine Theory*, 14:221–231, 1979.

[Mru84] T. S. MRUTHYUNJAYA. « A computerized methodology for structural synthesis of kinematic chains : Part 1 – Formulation ». *Mechanism and Machine Theory*, 19(6):487–495, 1984.

[MS90] S. MANOOCHEHRI et A. A. SEIREG. « A computer-based methodology for the form synthesis and optimal design of robot manipulators ». *Journal of Mechanical Design*, 112(12):501–508, 1990.

[MS94] E. MÄKINEN et M. SIERANTA. « Genetic algorithms for drawing bipartite graphs ». *International Journal of Computer Mathematics*, 53:157–166, 1994.

[Mur94] N. MURATA. « Function approximation by three-layered networks and its error bounds – an integral representation theorem ». Rapport de recherche METR 94-19, University of Tokyo, 1994.

[ND95] J. NIEVERGELT et N. DEO. « Metric graphs elastically embeddable in the plane ». *Information Processing Letters*, 55:309–315, 1995.

[Nis96] M. NISHIOKA. « Modular Structure of spatial cam-linkage mechanism ». *Mechanism and Machine Theory*, 31(6):813–819, 1996.

[NJ92] D. NEVILLE et L. JOSKOWICZ. « A representation language for conceptual mechanism design». In *International Workshop on Qualitative Reasoning about Physical Systems : QR'92*, p. 96–108, 1992.

[NSS91] D. NAVINCHANDRA, K. P. SYCARA et N. S.. « A transformational approach to case-based synthesis ». *AI EDAM*, 5(1):31–45, 1991.

[NZ94] T. M. NABHAN et A. Y. ZOMAYA. « Toward generating neural network structures for function approximation ». *Neural Networks*, 7(1):89–99, 1994.

[OER85] D. G. OLSON, A. G. ERDMAN et D. R. RILEY. « A systematic procedure for type synthesis of mechanisms with literature review ». *Mechanism and Machine Theory*, 20(4):285–295, 1985.

[OER87] D. G. OLSON, A. G. ERDMAN et D. R. RILEY. « Formulation of dimensional synthesis procedures for complex planar mechanisms ». *Journal of Mechanisms, Transmissions, and Automation in Design*, 109(9):322–328, 1987.

[PDS82] C. PELECUDI, G. DRĂGANOIU et I. SIMIONESCU. *Algoritmi şi programe pentru analiza mecanismelor*. Editura Tehnică, Bucarest, 1982.

[PPW79] H. W. PAKES, R. M. PARRY et G. R. WRAY. « The use of optimization techniques with precision synthesis for producing a planar linkage giving parallel motion ». *Mechanism and Machine Theory*, 14:171–178, 1979.

[PS82] C. PELECUDI et I. SIMIONESCU. *Probleme de mecanisme*. E.D.P., Bucarest, 1982.

[PS86a] C. PELECUDI et I. SIMIONESCU. *Mecanisme*. Editura Tehnică, Bucarest, 1986.

[PS86b] C. PELECUDI et I. SIMIONESCU. *MERIG–Mecanica asistată de calculator*. Editura Tehnică, Bucarest, 1986.

[Pug84] J. T. PUGH. « Synthesis of Pareto optimal four-bar function generators with optimum structural error and optimum transmission angles ». *Journal of Mechanisms, Transmissions, and Automation in Design*, 106(4):437–443, 1984.

[Rao79] A. C. RAO. « Synthesis of 4-bar function-generators using geometric programming ». *Mechanism and Machine Theory*, 14:141–149, 1979.

[Rao87] S. S. RAO. « Description and optimum design of fuzzy mechanical systems ». *ASME Journal of mechanisms, transmissions, and automation in design*, 109(1):126–132, 1987.

[Rao95] A. C. RAO. « Topological characteristics of linkage mechanisms with particular reference to platform-type robots ». *Mechanism and Machine Theory*, 30(1):33–42, 1995.

[Ric93] R. J. RICCI. « Spacebar : kinematic design by computer graphics ». *Computer-Aided Design*, 25(11):727–735, 1993.

[RK88] G. A. RIGELMAN et N. S. KRAMER. « A computer-aided design technique for the synthesis of planar four bar mechanisms satisfying specified kinematic and dynamic conditions ». *Journal of Mechanisms, Transmissions, and Automation in Design*, 110(9):263–268, 1988.

[Rom95] L. ROMDHANE. « A dyad search algorithm for solving planar linkages using the dyad method ». *Les Annales Maghrébines de l'Ingénieur*, 9(1):19–35, 1995.

[RR76] R. R. ROOT et K. M. RAGSDELL. « A survey of optimization methods applied to the design of mechanisms ». *Journal of Engineering for Industry*, (8):1036–1041, 1976.

[RR93] A. C. RAO et C. N. RAO. « Loop based pseudo Hamming values – I. Testing isomorphism and rating kinematic chains ». *Mechanism and Machine Theory*, 28(1):113–127, 1993.

[RR95] M. RAGHAVAN et B. ROTH. « Solving polynomial systems for the kinematic analysis and synthesis of mechanisms and robot manipulators ». *Journal of Mechanical Design, Special 50th Anniversary Design Issue*, 117:71–79, 1995.

[Sac92] E. SACKS. « Model-based kinematic simulation ». In *International Workshop on Qualitative Reasoning about Physical Systems : QR'92*, p. 109–120, 1992.

[San90] E. SANDGREN. « A multi-objective design tree approach for the optimization of mechanisms ». *Mechanism and Machine Theory*, 25(3):257–272, 1990.

[Sar93] P. SARDAIN. « *Contribution à la synthèse des mécanismes. Un environnement CAO pour la synthèse topologique et dimensionnelle de générateurs de mouvement* ». Thèse de doctorat, Université de Poitiers, 1993.

[Sar94] P. SARDAIN. « Un environnement CAO pour la synthèse de mécanismes articulés ». *Revue internationale de CFAO*, 9(1):135–154, 1994.

[SC79] I. SIMIONESCU et D. CEZAR. « On a general method for the synthesis of the path approximation mechanisms ». *Mechanism and Machine Theory*, 14:289–298, 1979.

[Sch90] W. SCHIEHLEN, éditeur. *Multibody systems handbook*. Springer-Verlag, 1990.

[SDW88] A. H. SONI, M. H. F. DADO et Y. WENG. « An automated procedure for intelligent mechanism selection and dimensional synthesis ». *Journal of Mechanisms, Transmissions, and Automation in Design*, 110:131–137, 1988.

[SE84] G. SANDOR et A. ERDMAN. *Advanced Mechanism Design : Analysis and Synthesis*, v. 2. Prentice-Hall, 1984.

[SF86] J. SOHN et F. FREUDENSTEIN. « An application of dual graphs to the automatic generation of the kinematic structures of mechanisms ». *Transaction of the ASME*, 108(9):392–398, 1986.

[SF93] G. STARNS et D. R. J. FLUGRAD. « Five-bar path generation synthesis by continuation methods ». *Journal of Mechanical Design*, 115(12):988–994, 1993.

[SF94] T. SUBBIAN et D. R. J. FLUGRAD. « Six and seven position triad synthesis using continuation methods ». *Journal of Mechanical Design*, 116(7):660–665, 1994.

[SH82] V. SOHONI et E. J. HAUG. « A state space method for kinematic optimization of mechanisms and machines ». *Journal of Mechanical Design*, 104:101–107, 1982.

[Shr93] H. E. SHROBE. « Understanding linkages ». In *International Workshop on Qualitative Reasoning about Physical Systems : QR'93*, p. 212–218, Orcas Island, WA/USA, 1993.

[Sie73] R. SIESTRUNCK. « Quelques aspects nouveaux de la théorie des mécanismes et applications. Applications récentes de la théorie des graphes à l'analyse et à la conception de mécanismes ». *Revue française de Mécanique*, 45:5–11, 1973.

[SJ93] E. SACKS et L. JOSKOWICZ. « Automated modeling and kinematic simulation of mechanisms ». *Computer-Aided Design*, 25(2):106–118, 1993.

[SM90] N. H. SALEM et S. MANOOCHEHRI. « Autocad based computer-aided mechanism design system ». In A. PISANO, éditeur, *The 1990 ASME Design Technical Conferences – 21st biennial mechanisms conference*, v. 26, p. 331–339, Chicago, Illinois, 16-19 Sep., 1990.

[Smi96] L. SMITH. « An introduction to neural networks ». In *NSYN meeting*, Edinburgh, Scotland, 28 féb. 1996.

[SNG+] K. SYCARA, D. NAVINCHANDRA, R. GUTTAL, J. KONING et S. NARASIMHAN. « CADET : a case-based synthesis tool for engineering design ». *International Journal of Expert Systems.*

[SOF94] Y. SAKAMOTO, K. OGAZA et H. FUNABASHI. « A kinematic analysis of planar six-bar mechanisms with consideration of the interference of moving links ». *Mechanism and Machine Theory*, 29(3):345–356, 1994.

[SR61] S. SESHU et M. B. REED. *Linear graphs and electrical networks.* Addison-Wesley, Reading, Londres, 1961.

[SR78] C. H. SUH et C. W. RADCLIFFE. *Kinematics and mechanism design.* Robert E. Krieger Publishing Company, Malabar, Florida, 1978.

[SR87] R. P. SUKHIJA et A. C. RAO. « Synthesis of path-generating mechanisms and tolerance allocation using information theory ». *Journal of Mechanical Design*, 109(9):366–371, 1987.

[SS88] D. P. SATHYADEV et A. H. SONI. « Synthesis of planar mechanisms for coupler tangent-line enveloppe generation : an alternative to couple point-path generation ». *Journal of Mechanisms, Transmissions, and Automation in Design*, 110:122–129, 1988.

[SS94] S. SUNDAR et Z. SHILLER. « Constrained optimization of multi-degree-of-freedom mechanisms for near-time-optimal motions ». *Journal of Mechanical Design*, 116(7):412–418, 1994.

[SW93] D. SUBRAMANIAN et C.-S. WANG. « Kinematic synthesis with configuration spaces ». In *International Workshop on Qualitative Reasoning about Physical Systems : QR'93*, p. 228–239, Orcas Island, WA/USA, 1993.

[SXW90] G. N. SANDOR, Y. XU et T.-C. WENG. « A graphical method for solving the Euler-Savary equation ». *Mechanism and Machine Theory*, 25(2):141–147, 1990.

[TC92] S.-H. TONG et C. H. CHIANG. « Synthesis of planar and spherical four-bar path generators by the pole method ». *Mechanism and Machine Theory*, 27(3):261–273, 1992.

[Tin89] K.-L. TING. « Mobility criteria of single-loop n-bar linkages ». *Transaction of the ASME*, 111(12):504 sq., 1989.

[TL93] C. S. TANG et T. LIU. « The degree code – a new mechanism identifier ». *Journal of Mechanical Design*, 115(9):627–630, 1993.

[TMGY96] L. TRAVÉ-MASSUYÈS, F. GUERRIN et B. YANNOU. *Raisonnement qualitatif pour les sciences de l'ingénieur*. Masson, Paris, 1996.

[TPT89a] E. TUTTLE, S. PETERSON et J. TITUS. « Enumeration of basic kinematic chains using the theory of finite groups ». *Transaction of the ASME*, 111(12):498–503, 1989.

[TPT89b] E. TUTTLE, S. PETERSON et J. TITUS. « Further applications of group theory to the enumeration and structural analysis of basic kinematic chains ». *Transaction of the ASME*, 111(12):494–497, 1989.

[TS95] M.-Y. TEO et S.-K. SIM. « Training the neocognitron network using design of experiments ». *Artificial Intelligence in Engineering*, 9:85–94, 1995.

[Tut63] W. T. TUTTE. « How to draw a graph ». *Proceedings of the London Mathematical Society*, 13(3):743–767, 1963.

[Tut96] E. R. TUTTLE. « Generation of planar kinematic chains ». *Mechanism and Machine Theory*, 31(6):729–748, 1996.

[UK95] V. UNRUH et P. KRISHNASWAMI. « A computer-aided design technique for semi-automated infinite point coupler curve synthesis of four-bar linkages ». *Journal of Mechanical Design*, 117:143–149, 1995.

[VAHA95] J. VALLEJO, R. AVILÉS, A. HERNANDEZ et E. AMEZUA. « Nonlinear optimization of planar linkages for kinematic synthesis ». *Mechanism and Machine Theory*, 30(4):501–518, 1995.

[Vas92] A. VASILIU. « Vers une approche non-numérique de la synthèse de mécanismes. Le problème du dépliage des graphes ». Mémoire de DEA, Ecole Centrale Paris, 1992.

[Vas93] A. VASILIU. « A topological graph unfolding algorithm ». In *COMES'93 : First International Workshop on Mechanism Design and Analysis*, Clermont Ferrand, France, 17-18 mai 1993.

[Vas94] A. VASILIU. « REALISME : approche et logiciel d'aide à la réalisation de mécanismes en phase de préconception ». In *Journée PRIMECA sur la conception de mécanismes et des robots*, Douai, France, 1994.

[vdSG95] P. van der SMAGT et F. GROEN. « Approximation with neural networks : between local and global approximation ». In *International Conference on Neural Networks*, Perth, Australia, 1995.

[Vel63] G. R. VELDKAMP. « *Curvature theory in plane kinematics* ». Thèse de doctorat, Technical University, Delft, 1963.

[Ver73] A. VERHO. « Extension of the concept of the group ». *Mechanism and Machine Theory*, 8(2):249–256, 1973.

[vhG95] F. van het GROENEWOUD. « Neural approach in the approximation of eye-hand mapping and the inverse kinematic function : a comparative study ». Rapport de recherche, University of Amsterdam, Faculty of Computer Science, 1995.

[Vys93] V. VYSNIAUSKAS. « The optimal number of learning samples and hidden units in function approximation with a feedforward network ». Rapport de recherche CS-93-15, University of Amsterdam, Faculty of Computer Science and Mathematics, 1993.

[WA96] W. H. WOOD et A. M. AGOGINO. « Case-based conceptual design information server for concurrent engineering ». *Computer-Aided Design*, 28(5):361–369, 1996.

[Wal94] K. J. WALDRON. « Solution rectification in three position motion generation synthesis ». *Journal of Mechanical Design*, 116:88–91, 1994.

[Wat89] H. WATANABE. « Heuristic graph displayer for G-BASE ». *International Journal on Man-Machine Studies*, 30:287–302, 1989.

[Wat92] K. WATANABE. « Application of natural equations to the synthesis of curve generating mechanisms ». *Mechanism and Machine Theory*, 27(3):261–273, 1992.

[WMS90] C. W. WAMPLER, A. P. MORGAN et A. J. SOMMESE. « Numerical continuation methods for solving polynomial systems arising in kinematics ». *Journal of Mechanical Design*, 112(3):59–68, 1990.

[Yan92] H.-S. YAN. « A methodololgy for creative mechanism design ». *Mechanism and Machine Theory*, 27(3):235–242, 1992.

[Yan94] B. YANNOU. « *Aide à la conception qualitative de systèmes dynamiques* ». Thèse de doctorat, Ecole Normale Supérieure de Cachan, 1994.

[YDDK91] B. YANG, P. DATSERIS, U. DATTA et J. KOWALSKI. « An integrated system for design of mechanisms by an expert system-Domes : Applications ». *Journal of Mechanical Design*, 113:25–31, 1991.

[YH81] H. YAN et A. HALL. « Linkage characteristic polynomials : definitions, coefficients by inspection ». *Journal of Mechanical Design*, 103(7):578–584, 1981.

[YV95] B. YANNOU et A. VASILIU. « Design platform for planar mechanisms based on a qualitative kinematics ». In *QR'95 : Ninth International Workshop on Qualitative Reasoning about Physical Systems*, Amsterdam, Pays-Bas, 18–19 mai 1995.

[YVM95] B. YANNOU, A. VASILIU et M. MEKHILEF. « Plate-forme de conception de mécanismes plans basée sur une approche de cinématique qualitative ». In *Quatrième colloque PRIMECA sur la conception integrée*, 3–5 avril, La Plagne, France, 1995.

[ZMV+95] A. ZELL, G. MAMIER, M. VOGT, N. MACHE et R. HÜBNER. « SNNS (Stuttgart Neural Network Simulator) User Manual ». Rapport de recherche 6/95, University of Stuttgart, Institute for parallel and distributed high performance systems, 1995.

[ZNH84] C. ZHANG, R. L. NORTON et T. HAMMOND. « Optimization of parameters for specified path generation using an atlas of coupler curves of geared five bar linkages ». *Mechanism and Machine Theory*, 19:459–466, 1984.

Index

analyse, 38, 40, 42–44, 52, 195
  analyse cinématique, 50, 51
  cinématique, 43, 195
  dynamique, 196
  dynamique inverse (cinétostatique), 195
approximation de fonction, 34, 50, 132, 148, 160, 207

catalogue de trajectoires, 25, 29, 30, 33, 47, 48, 161

erreur structurale, 19, 24, 115–117, 121
erreurs structurale, 26

groupe d'Assur, 9
groupe systémique multipolaire, 55
  groupe actif, 57
  groupe passif, 57
  schéma multipolaire, 59
groupes d'Assur, 9, 12, 13, 16, 17, 19, 25, 32, 55, 70, 74, 116, 117, 187

liaison cinématique, 195
  liaison motrice, 195
liaison motrice, 12, 13, 56–59, 61

méthodes de continuation, 16, 25, 27, 51, 132, 188

optimisation dimensionnelle, 23, 30, 34, 43, 50–52, 115, 116, 121, 122, 126, 129, 132, 134, 144, 181
  exacte, 51, 115, 124, 126
  qualitative, 115, 126
  variables de conception, 118, 122

point effecteur, 23, 29, 32, 37, 43, 45, 51, 75, 76, 78–82, 84, 91, 92, 96, 118, 122, 131, 133, 134, 145, 148
préconception, 9, 10, 32, 37, 39, 43, 44, 52, 76, 87, 94, 164, 183
préconception (conception préliminaire), 3, 4

réseaux neuronaux artificiels
  apprentissage, 35
  fonction d'activation, 34, 205
Raisonnement à base de cas (Case-based Reasoning), 130
recalage géométrique, 51
  trajectoires exactes, 121
  trajectoires qualitatives, 126
robots, robotique, 1, 3, 10, 14, 35, 38, 192, 203

schéma multipolaire, 88
simulation cinématique, 44, 48, 50, 51, 99, 118–121, 124, 126
  qualitative, 51, 52, 81, 115, 124
synthèse
  avec dépendance du temps, 196
  avec indépendance du temps, 47, 49, 51, 196
synthèse dimensionnelle
  avec indépendance du temps, 164, 181

théorie des mécanismes, 3, 4, 37, 73
trajectoire (courbe)
  désirée, 115, 132, 134, 135, 152, 153, 155–157, 159
  désirée (spécifiée), 76
  exacte, 51, 75, 115
  faisable, 159
  perturbée, 159
  qualitative, 48, 75, 76, 115
  trajectoire désirée, 4