



**HAL**  
open science

# Contribution à l'étude, à la simulation et à la commande d'un robot de type serpent

Renaud Aubin

► **To cite this version:**

Renaud Aubin. Contribution à l'étude, à la simulation et à la commande d'un robot de type serpent. Automatique / Robotique. Université de Versailles-Saint Quentin en Yvelines, 2006. Français. NNT : . tel-00437990

**HAL Id: tel-00437990**

**<https://theses.hal.science/tel-00437990>**

Submitted on 2 Dec 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.

---

# Contribution à l'étude, à la simulation et à la commande d'un robot de type serpent

## THÈSE

présentée et soutenue publiquement le 26 octobre 2006

pour l'obtention du

**Doctorat de l'université de  
Versailles-Saint-Quentin-en-Yvelines**

**(Spécialité robotique)**

par

Renaud AUBIN

### Composition du jury

<i>Directeur de thèse :</i>	M. Pierre BLAZEVIC	Professeur UVSQ
<i>Rapporteurs :</i>	M. Wisama KHALIL	Professeur École Centrale de Nantes
	M. Ahmet KUZUCU	Professeur ITÜ Istanbul
<i>Examineurs :</i>	M. Alain BOURJAULT	Professeur ENSMM Besançon
	M. Olivier CHOCRON	Maître de Conférence ENIB
	M. Jean-Pierre GASC	Professeur Émérite MNHN Paris
<i>Invités :</i>	M. Aurélien GODIN	DGA - ETAS d'Angers

---



*The hardest thing is to go to sleep at night, when there are so many urgent things needing to be done. A huge gap exists between what we know is possible with today's machines and what we have so far been able to finish.*

Donald E. KNUTH



# Remerciements

Les travaux de recherche présentés dans ce manuscrit ont été réalisés au Laboratoire d'Ingénierie des Systèmes de Versailles dans le cadre d'un financement DGA/CNRS sous la direction du Professeur Pierre BLAZEVIC. Je le remercie de la confiance qu'il a placée en moi.

Je remercie tout particulièrement MM. Wisama KHALIL, Professeur à l'École Centrale de Nantes, et Ahmet KUZUCU, Professeur à l'Université Technique d'Istanbul, pour leur participation au jury en tant que rapporteurs de ce manuscrit. De la même manière, je remercie MM. Alain BOURJAULT, Professeur à l'École Nationale Supérieure de Mécanique et des Microtechniques de Besançon, et Jean-Pierre GASC, Professeur Émérite du Muséum national d'histoire naturelle de Paris d'avoir répondu positivement à notre invitation à participer à ce jury.

Je remercie M. Olivier CHOCRON pour sa participation au jury, pour la qualité de ses remarques lors de nos échanges ainsi que pour le soutien qu'il m'a apporté.

Je remercie également M. Aurélien GODIN, en charge du pilotage de ces travaux de doctorat pour la DGA, de sa participation en tant que membre invité du jury.

Je remercie enfin MM. Thomas COSTIS et Olivier NEGRO pour leur support inconditionnel et quotidien dans les bons moments comme dans les moins bons. Je n'oublie évidemment pas le soutien prodigué par mes autres collègues et le personnel du LISV.

Enfin, je souhaiterais témoigner à mon épouse toute ma gratitude pour toute la patience et l'écoute dont elle a fait preuve durant ces trois années. Ce manuscrit lui est dédié, ainsi qu'à la nouvelle génération qu'elle porte et qui va prochainement faire de moi le plus heureux des pères.

Renaud AUBIN  
Carrières-sous-Poissy, août 2006



# Table des matières

<b>Remerciements</b> .....	<b>v</b>
<b>Table des matières</b> .....	<b>vii</b>
<b>Liste des figures</b> .....	<b>xi</b>
<b>Liste des tableaux</b> .....	<b>xv</b>
<b>Introduction</b> .....	<b>1</b>
<b>1 La simulation numérique appliquée à la robotique</b> .....	<b>3</b>
1.1 Introduction .....	3
1.2 Les environnements et les moteurs de simulation de solides indéformables .....	5
1.2.1 Architecture .....	5
1.2.2 Solutions basées sur ODE .....	7
1.2.3 Autres solutions .....	8
1.3 Conclusion .....	9
<b>2 La problématique de la locomotion des robots de type serpent</b> .....	<b>11</b>
2.1 La problématique de la locomotion robotique .....	12
2.1.1 Contexte : la robotique d'intervention .....	12
2.1.2 Robots à roues .....	12
2.1.3 Robots à chenilles .....	14
2.1.4 Robots à pattes .....	15
2.1.5 Robots de type serpent .....	16
2.2 La locomotion des serpents terrestres .....	17
2.2.1 Les différents modes de locomotion .....	17
2.2.1.1 Progression par ondulation horizontale .....	18
2.2.1.2 Progression par mouvements d'accordéon .....	19
2.2.1.3 Progression rectilinéaire .....	19
2.2.1.4 Progression par déroulement latéral .....	20
2.2.1.5 Autres modes de locomotion .....	20
2.2.2 Le mode de locomotion par progression rectilinéaire .....	20
2.2.2.1 Mécanisme de locomotion : point de vue élémentaire .....	21
2.2.2.2 Changement de direction .....	24
2.2.2.3 Considérations énergétiques .....	24
2.2.3 Conclusion .....	24
2.3 État de l'art des robots-serpents .....	25
2.3.1 Systèmes robotiques bio-inspirés par l'ondulation latérale .....	25
2.3.2 Les systèmes non basés sur le mode d'ondulation latérale .....	29
2.3.3 Les robots de type serpent et l'apprentissage .....	30
2.3.4 Les autres travaux .....	30
2.4 Conclusion .....	31

---

<b>3</b>	<b>Proposition d'un environnement intégré de simulation robotique</b>	<b>33</b>
3.1	Architecture	34
3.2	Description fonctionnelle des objets	35
3.2.1	Simulateur	35
3.2.2	Robot	36
3.2.3	Commande	39
3.2.4	Trajectoire	40
3.2.5	Interface utilisateur	41
3.2.6	Gestion de simulation pour l'utilisation de techniques d'apprentissage	42
3.3	Algorigrammes	43
3.3.1	Programme principal	43
3.3.2	Gestion par techniques d'apprentissage	43
3.3.3	Simulateur	43
3.4	Rendu réaliste	48
3.5	Utilisation de cartes de niveaux	48
3.6	Conclusion	49
<b>4</b>	<b>Modélisation d'une première structure robotique bio-inspirée</b>	<b>51</b>
4.1	Le projet VIPeRe	51
4.1.1	Brevets	52
4.1.2	Prototype	53
4.2	Mécanisme de propulsion	53
4.2.1	Caractérisation et définition du mécanisme	53
4.2.1.1	Schéma cinématique	53
4.2.1.2	Graphe des liaisons	54
4.2.2	Simplification du mécanisme en un mécanisme plan	55
4.2.3	Degrés de liberté du mécanisme	55
4.2.4	Modélisation géométrique directe	56
4.2.5	Modélisation cinématique directe	58
4.3	Simulation d'un module à 8 mécanismes	60
4.3.1	Définition du module	60
4.3.2	Principe de locomotion d'un module à 8 mécanismes	61
4.3.3	Comportement local d'un mécanisme simulé	61
4.3.3.1	Résultats géométriques	61
4.3.3.2	Résultats cinématiques	61
4.3.4	Comportement du module simulé	66
4.3.4.1	Résultats géométriques	66
4.3.4.2	Résultats cinématiques	66
4.4	Conclusion	66
<b>5</b>	<b>Proposition d'une nouvelle structure robotique</b>	<b>67</b>
5.1	Optimisation et performances	67
5.1.1	Critères d'optimisation	68
5.1.2	Variation d'altitude de la plate-forme	68
5.1.3	Amplitude verticale	69
5.1.4	Performance locale de progression dans l'axe longitudinal	71
5.1.5	Exemples de critères croisés	71

---

5.1.6	Application à un module à 8 mécanismes .....	72
5.2	Proposition d'une nouvelle structure robotique .....	72
5.2.1	Motivations .....	72
5.2.2	Module .....	76
5.2.3	Articulations inter-modulaire .....	77
5.2.4	Conclusion .....	77
<b>6</b>	<b>Utilisation de la simulation pour le développement d'algorithmes de commande</b>	<b>81</b>
6.1	Approche géométrique en boucle ouverte .....	81
6.1.1	Définition de la commande pour une tâche de suivi de trajectoire simple sur sol plat .....	82
6.1.2	Résultats de simulation .....	83
6.1.3	Conclusion .....	83
6.2	Approche cinématique .....	86
6.2.1	Description de la méthode .....	86
6.2.2	Trajectoires .....	88
6.3	Application de l'approche cinématique .....	88
6.3.1	Suivi de trajectoire sur sol plat avec 4 modules .....	88
6.3.1.1	Description de la tâche .....	88
6.3.1.2	Résultats .....	89
6.3.2	Suivi de trajectoire sur sol plat avec 6 modules .....	89
6.3.2.1	Description de la tâche .....	89
6.3.2.2	Résultats .....	93
6.3.3	Suivi de ligne droite sur sol plat avec 4 modules et état initial perturbé .....	93
6.3.3.1	Description de la tâche .....	93
6.3.3.2	Résultats .....	93
6.4	Conclusion .....	93
<b>7</b>	<b>Conclusions et perspectives</b> .....	<b>99</b>
7.1	Conclusions .....	99
7.2	Perspectives .....	100
	<b>Abréviations</b> .....	<b>103</b>
	<b>Notations</b> .....	<b>105</b>
	<b>Références bibliographiques</b> .....	<b>107</b>
<b>A</b>	<b>Simulation dynamique de solides indéformables avec <i>Open Dynamics Engine</i> (ODE)</b> .....	<b>113</b>
A.1	Équations de mouvement .....	113
A.2	Approximation numérique .....	117
A.3	Utilisation de contraintes en vitesse pour la résolution .....	118
A.3.1	Contraintes en vitesse entre deux solides .....	118
A.3.2	Exemple de contrainte : liaison rotule .....	119
A.3.3	Formulation globale des équations de contraintes .....	120
A.4	Contacts .....	121



# Liste des figures

1.1	Structure générale d'un moteur de simulation dynamique, d'après [COUTINHO, 2001] .....	6
1.2	Architecture générale modulaire complète, d'après [ERLEBEN, 2004] .....	6
1.3	Modélisation de l'architecture de ODE, d'après [ERLEBEN, 2004] .....	8
2.1	Hylos (LRP6) .....	13
2.2	Shrimp III (EPFL) .....	13
2.3	WorkPartner (HUT) .....	13
2.4	RobuROC 6 (ECA - ROBOSOFT) .....	14
2.5	ROBHAZ-DT3 (KIST) .....	14
2.6	PackBot (iRobot) .....	15
2.7	BigDog (Boston Dynamics) .....	16
2.8	HRP-3P (Kawada) .....	16
2.9	RHex (Boston Dynamics) .....	16
2.10	Illustrations de l'ondulation latérale, d'après [DOWLING, 1997 et GANS, 1980] ..	18
2.11	Illustration de la progression par mouvements d'accordéon, d'après [DOWLING, 1997] .....	19
2.12	Illustration de la progression rectilinéaire, d'après [DOWLING, 1997] .....	19
2.13	Illustration du déroulement latéral, d'après [GANS, 1980] .....	20
2.14	Reconstruction simplifiée de l'anatomie d'un serpent, d'après [GANS, 1980] ....	21
2.15	Décomposition du mode de locomotion rectilinéaire, d'après [GASC, 1974] .....	22
2.16	Vitesses d'une écaille ( $V'$ ) et de la colonne vertébrale ( $V$ ) d'un serpent par rapport au sol (mode de progression rectilinéaire), d'après [GANS, 1980] .....	23
2.17	Vertèbre de serpent .....	24
2.18	ACM III (HIROSE - 1972) .....	25
2.19	Exemples de courbes serpentines, d'après [SAITO <i>et al.</i> , 2002] .....	26
2.20	Robots de MILLER .....	27
2.21	Robots ACM-RX (HIROSE) .....	28
2.22	Robots Souryu (HIROSE) .....	28
2.23	Robot KOHGA .....	28
2.24	Robots OmniTread .....	29
2.25	Robot SSR-I, [AOKI <i>et al.</i> , 2004] .....	30
3.1	Architecture simplifiée de l'application .....	36
3.2	Composition de l'objet « Robot » .....	37
3.3	Composition de l'objet « Commande » .....	39
3.4	Composition de l'objet « Trajectoire » .....	40
3.5	Composants génériques de l'interface utilisateur .....	41
3.6	Composition de l'objet « Gestion de l'apprentissage » .....	42
3.7	Algorithme de l'application .....	44

3.8	Algorigrammes (a) de la tâche de gestion par algorithme évolutionnaire et (b) de l'évaluation des individus	45
3.9	Algorigrammes de (a) l'initialisation du simulateur, (b) du chargement de la configuration et (c) de la création de l'objet « Commande »	46
3.10	Algorigramme de l'initialisation de l'objet « Commande »	47
3.11	Vue en transparence d'un module à 8 mécanismes obtenue avec POV-Ray	48
3.12	Exemple de sol généré par Terraform	49
3.13	Exemple de sol créé avec GIMP	49
4.1	Étude CAO du prototype VIPeRe	52
4.2	Schéma cinématique du mécanisme	54
4.3	Graphe des liaisons du mécanisme	55
4.4	Graphe des liaisons du mécanisme simplifié	55
4.5	Vue de côté d'un mécanisme	56
4.6	Trajectoire et sens de parcours du point $M$ pour différentes valeurs de $\varphi$	59
4.7	Rendu réaliste d'un module à 8 mécanismes	59
4.8	Vue de côté d'un module à 8 mécanismes	60
4.9	Locomotion d'un module à 8 mécanismes	62
4.10	Comparaison de la trajectoire simulée avec la trajectoire obtenue à l'aide du MGD ( $\varphi = \frac{5\pi}{8}$ )	62
4.11	Projection suivant l'axe $\vec{x}$ de la vitesse linéaire de l'organe terminal par rapport au corps principal du module	63
4.12	Projection suivant l'axe $\vec{z}$ de la vitesse linéaire de l'organe terminal par rapport au corps principal du module	63
4.13	Capture de la simulation d'un module à 8 mécanismes	64
4.14	Projection dans le plan $(\vec{x}, \vec{z})$ de la trajectoire du centre géométrique du corps principal du module	64
4.15	Projections suivant l'axe $\vec{x}$ de la vitesse absolue d'un module et de la vitesse absolue de l'organe terminal d'un mécanisme	65
4.16	Projections suivant l'axe $\vec{z}$ de la vitesse absolue d'un module et de la vitesse absolue de l'organe terminal d'un mécanisme	65
5.1	Représentation des critères d'optimisation pour la trajectoire de déphasage $\varphi = \frac{5\pi}{8}$	68
5.2	Caractéristique du critère variation d'altitude en fonction du déphasage $\varphi$	69
5.3	Amplitude selon $\vec{z}$ du mouvement de l'organe terminal d'un mécanisme en fonction du déphasage $\varphi$	70
5.4	Caractéristique du critère performance locale de progression dans l'axe longitudinal en fonction du déphasage $\varphi$	70
5.5	Critère $k_{\delta_x} \delta_x + k_{\delta_z} \delta_z + k_{\Delta_z} \Delta_z$ pour différents triplets $(k_{\delta_x}, k_{\delta_z}, k_{\Delta_z})$	71
5.6	Franchissement de la première marche ( $\varphi = \pi$ )	73
5.7	Franchissement de la deuxième marche ( $\varphi = \pi$ )	74
5.8	Simulation d'une situation de blocage d'un module à 8 mécanismes	75

5.9	Modélisation cinématique de l'interaction d'un robot composé de 2 modules à 8 mécanismes avec un sol plan au repos – (a) modèle de simulation, (b) schéma cinématique (débattements en roulis et tangage négligeables), (c) schéma cinématique (débattements en roulis et tangage non négligeables) .....	75
5.10	Courbure maximale admissible .....	76
5.11	Vue de dessus de la structure proposée .....	78
5.12	Simulation d'un robot constitué de 2 modules à 4 mécanismes .....	78
5.13	Modélisation cinématique de l'interaction d'un robot composé de 2 modules à 4 mécanismes avec un sol plan au repos .....	79
6.1	Trajectoire de référence (approche géométrique) .....	82
6.2	Trajectoires des centres géométriques des corps principaux des modules (approche géométrique) .....	84
6.3	Étapes de simulation d'un robot constitué de 5 modules à 8 mécanismes (approche géométrique) .....	85
6.4	Module $i$ .....	86
6.5	Trajectoires des 4 modules pour une tâche de suivi de courbe interpolée .....	90
6.6	Évolution des consignes de vitesses articulaires au cours du temps .....	90
6.7	Évolution de l'erreur en distance $\delta_{v_i}$ au cours du temps .....	91
6.8	Évolution de l'erreur en orientation $\beta_i$ au cours du temps .....	91
6.9	Déroulement de la simulation d'une tâche de suivi de trajectoire pour un robot à 4 modules .....	92
6.10	Trajectoires des 6 modules pour une tâche de suivi de courbe interpolée .....	94
6.11	Évolution de l'erreur en distance $\delta_{v_i}$ au cours du temps .....	95
6.12	Évolution de l'erreur en orientation $\beta_i$ au cours du temps .....	95
6.13	Trajectoires des 4 modules pour une tâche de suivi de ligne droite (état initial perturbé) .....	96
6.14	Évolution des consignes de vitesses articulaires au cours du temps .....	96
6.15	Évolution de l'erreur en distance $\delta_{v_i}$ au cours du temps .....	97
6.16	Évolution de l'erreur en orientation $\beta_i$ au cours du temps .....	97
A.1	Méthode d'approximation EULER explicite .....	117
A.2	Liaison rotule (avec représentation d'une erreur en position du centre de la liaison) .....	119



# Liste des tableaux

1.1	Les principales objections à l'expérimentation directe .....	4
2.1	Correspondances et fréquences des modes de locomotion des serpents, d'après [GANS, 1980] .....	17
3.1	Convention graphique utilisée pour la description du programme .....	35
3.2	Structures de données ODE pour la définition d'un robot .....	38
4.1	Valeurs des constantes géométriques du mécanisme de locomotion (issues du modèle CAO) .....	57
6.1	Coordonnées des points de passage utilisés pour le calcul de la trajectoire de référence .....	88
6.2	Coordonnées des points de passage utilisés pour le calcul de la trajectoire de référence .....	89



# Introduction

La problématique de la locomotion dans le contexte de la robotique d'intervention est un sujet très vaste principalement à cause de la diversité des milieux étudiés. Des exemples de milieux naturels susceptibles d'accueillir une application de robotique d'intervention sont, entre autres :

- le milieu désertique pour lequel il faut prendre en compte le caractère instable et déformable d'un sol présentant une faible granulométrie,
- le milieu glaciaire pour lequel on rencontrera des problèmes analogues à ceux du milieu désertique dans le cas de sol neigeux auxquels on ajoutera la prise en compte de conditions d'adhérence très spécifiques,
- le milieu rocheux présentant des obstacles de dimensions relativement importantes,
- le milieu urbain structuré ou déstructuré,
- le milieu forestier pour lequel le substrat pourra être composé de terre, d'herbes ou encore de feuilles.

Par conséquent, il n'existe pas de solution idéale. De même, un problème particulier admet généralement plusieurs solutions.

L'observation du vivant fournit une source d'inspiration aux roboticiens pour l'élaboration de systèmes robotiques. En considérant l'exemple des serpents, on constate qu'ils sont représentés au sein de tous les biotopes de notre planète, à l'exception du milieu glaciaire. En plus de leur capacité à adapter leurs modes de locomotion à leurs milieux, les serpents présentent pour la robotique plusieurs intérêts majeurs. En effet, ils ont trouvé des réponses à ce que l'on peut considérer comme des problèmes de mobilité du point de vue de la robotique, en élaborant des stratégies de locomotion leur permettant notamment de se faufiler dans les espaces les plus restreints, de se mouvoir sans efforts apparents grâce au caractère distribué et périodique de leur locomotion ou encore de progresser en toute discrétion au sein de leur environnement.

Les travaux de recherches présentés dans ce manuscrit reposent sur l'utilisation d'une structure robotique originale inspirée par l'observation des modes de locomotion des serpents terrestres.

L'objectif initial était de proposer un algorithme de commande pour un robot de type serpent strictement basé sur la conception VIPeRe de l'entreprise TDA [TDA, 2006]. Cette conception repose sur l'utilisation conjointe de deux modes de locomotion des serpents, la progression rectilinéaire et l'ondulation latérale. Le substrat de référence est un substrat naturel tel que l'on peut le trouver en forêt, constitué principalement de terre et d'herbe. L'étude de l'interaction avec un substrat de cette nature implique la prise en compte du caractère déformable de l'environnement ainsi que l'élaboration d'une commande 3D complexe. Le contexte de cette étude est la robotique d'intervention en milieu naturel *a priori* inconnu et complexe pour des applications militaires de renseignement ou d'action pour lesquelles la furtivité revêt une grande importance. Le choix de l'utilisation d'un robot de type serpent pour de telles applications est orienté par les critères de furtivité et d'adaptation au milieu. En effet, les dimensions que présente cette catégorie de robot sont relativement

bien adaptées à ces critères. Les robots de type serpent présentent généralement des dimensions avantageuses pour la furtivité du fait de leur section réduite et par conséquent de leur hauteur limitée permettant une progression discrète au sein d'un environnement herbeux. De plus, les interactions avec le sol induites par la locomotion sont distribuées le long du robot et impliquent des perturbations moindres sur l'environnement.

Ces objectifs très ambitieux ont été finalement décomposés en plusieurs étapes. Les travaux présentés dans ce manuscrit constituent une première étape. Notre étude se base sur l'utilisation du mode de locomotion rectilinéaire pour la propulsion et sur l'ondulation de la structure robotique pour assurer les changements de directions. Le mode de progression rectilinéaire est reproduit par le biais de l'utilisation de plusieurs mécanismes.

Ces travaux comportent deux phases majeures :

- l'étude de la structure robotique (mécanisme de locomotion, topologie de la structure robotique, optimisation du mécanisme de locomotion et commande de la structure complète),
- la simulation de cette structure à l'aide d'un environnement spécialement élaboré.

Le prototype de TDA n'étant pas disponible pour effectuer des expérimentations, l'élaboration d'un environnement de simulation robotique a été nécessaire. Ainsi, le premier chapitre est consacré aux principes de la simulation numérique de solides indéformables à l'aide de la bibliothèque *Open Dynamics Engine*. En effet, la modélisation numérique de la dynamique proposée par cette bibliothèque repose sur des hypothèses simplificatrices qu'il est nécessaire de connaître afin de mieux appréhender la portée de notre étude.

Le deuxième chapitre positionne nos travaux par rapport à la problématique de locomotion dans le contexte de la robotique d'intervention. Une étude des modes de locomotion des serpents terrestres introduit ensuite l'état de l'art des robots de type serpent.

Le troisième chapitre décrit l'implémentation de l'environnement intégré de simulation que nous avons développé pour servir de base à nos travaux de recherches.

Le quatrième chapitre détaille l'étude géométrique et cinématique du mécanisme sur laquelle nous nous appuyons pour valider la simulation d'un module à 8 mécanismes ainsi que pour préciser le principe de propulsion de la structure robotique.

Le cinquième chapitre présente les optimisations potentielles des mécanismes locomoteurs ainsi que la définition d'une nouvelle structure robotique. Les motivations de notre proposition reposent sur l'analyse critique de la topologie du robot VIPeRe basé sur l'utilisation de 8 mécanismes de locomotion par module ainsi que sur l'utilisation de liaisons inter-modulaires de type pivot.

Le sixième chapitre détaille la proposition de premiers algorithmes de commande. Deux approches sont présentées, l'approche géométrique en boucle ouverte et l'approche cinématique.

Le septième et dernier chapitre synthétise nos conclusions ainsi que les perspectives ouvertes par ces travaux de thèse.

# Chapitre 1

## La simulation numérique appliquée à la robotique

---

### Sommaire

---

1.1	Introduction .....	3
1.2	Les environnements et les moteurs de simulation de solides indéformables .	5
1.2.1	Architecture .....	5
1.2.2	Solutions basées sur ODE .....	7
1.2.3	Autres solutions .....	8
1.3	Conclusion .....	9

---

### 1.1 Introduction

Une grande part des travaux des ingénieurs et des chercheurs est dévolue à la recherche des conséquences d'une action sur un système ou sur une partie élémentaire d'un système plus complexe. Ils cherchent de cette façon à prévoir le comportement du système, d'en optimiser le fonctionnement ou encore de se prémunir d'éventuels défauts de fonctionnement. L'expérience, son observation et l'exploitation de ses résultats constitue à première vue le moyen le plus simple et le plus précis pour quantifier le comportement d'un système. Néanmoins, la mise en œuvre de telles expériences n'est pas toujours possible. Le tableau 1.1 reprend les principales objections à de telles mises en œuvre et propose un exemple pour chacune d'entre elles.

Des solutions alternatives sont depuis longtemps mises en œuvre afin de connaître le comportement d'un système physique sans recourir à une expérimentation en vraie grandeur. Une première solution consiste à élaborer une expérience représentative à échelle réduite. L'utilisation de maquettes ou d'expérimentations à échelle réduite sur des populations de référence en fait partie. Une autre solution a vu le jour en même temps que l'apparition des ordinateurs, celle basée sur l'utilisation de simulations numériques.

Russell SMITH, l'instigateur du développement de la bibliothèque de simulation de solides rigides ODE [ODE, 2006] définit d'une manière simplifiée la mise en œuvre d'une simulation numérique par la séquence suivante :

- observation d'un phénomène physique (systèmes mécanique, climatique, de communication, militaire, etc.),
- élaboration d'un modèle (mise en équations du phénomène),
- élaboration d'un algorithme de simulation (une méthode de résolution des équations pour déterminer l'évolution du système dans le temps),

TABLEAU 1.1. Les principales objections à l'expérimentation directe

<i>Objections</i>	<i>Exemples</i>
Coût financier	Construction d'un nouveau véhicule pour tester de nouveaux amortisseurs en conditions réelles. Construction d'un robot original dans le but de tester sa mobilité et sa commandabilité.
Coût politique	Augmentation de la TVA pour en mesurer les effets sur le budget au bout d'un an.
Contraire à l'éthique	Test d'une bombe sur une population pour en mesurer les effets. Essai d'accident sur un véhicule avec des passagers à bord.
Impossibilité matérielle	Pas d'exemple. Le matériel ou la population de référence n'existe pas.
Non reproductible	Engagement de forces armées pour mesurer l'impact d'une stratégie militaire. L'expérience ne peut pas être réalisée plusieurs fois dans des conditions identiques. Cet exemple rentre également dans le cadre de l'objection « Contraire à l'éthique ».
Délais	La durée totale d'un projet industriel de grande envergure peut s'étaler sur plusieurs années et engager des investissements très importants (projets spatiaux, aéronautiques, ferroviaires, etc.)

- écriture d'un programme informatique (implémentation de l'algorithme précédent),
- simulation (on exécute le programme).

Dans le cas de la simulation numérique de systèmes robotiques ou mécatroniques, de nombreux modèles physiques sont déjà disponibles. Ainsi, dans le cas particulier de la simulation de la dynamique d'un bras manipulateur à base fixe, on peut utiliser des modélisations issues de la mécanique des solides rigides, au moins en première approximation. En effet, la précision désirée ainsi que la nature physique du phénomène observé conditionne le choix, ou le cas échéant l'élaboration, du modèle à partir duquel on construira une simulation.

Modéliser et simuler correctement un mécanisme réel apparemment simple est très complexe. L'exemple du jouet du pic-vert en bois décrit par [PFEIFFER et GLOCKER, 2000] en est d'ailleurs une illustration. D'autres exemples d'élaboration de modèles complexes d'interaction et des algorithmes mis en œuvre pour leurs résolutions sont donnés par [ANDRADE-BARROSO, 2000 et BOYER *et al.*, 2006]. [BOYER *et al.*, 2006 et CHOCRON, 2000] illustrent également l'utilité de la simulation pour la conception d'un robot et de ses algorithmes de commande.

Selon [SMITH, 2006], simuler un robot est généralement plus rapide que de le construire. De plus, la destruction d'un robot simulé n'a pas les mêmes conséquences que celle d'un robot réel. Pour ces raisons, le coût de la simulation devrait être moins élevé que la construction d'un prototype. Néanmoins, la mise en œuvre d'une simulation réaliste est très difficile de par la complexité d'implémenter des modèles fins pour les actionneurs, les capteurs, les interactions ou encore la répartition des masses des solides. Enfin, la simulation peut

aboutir à des résultats corrects lors de l'élaboration de contrôleurs par l'exploitation de failles du monde simulé. De tels contrôleurs ne seront pas efficaces lors du passage à une application réelle. Il est, par conséquent, nécessaire de trouver un compromis au cas par cas, à savoir, prototyper les algorithmes de commande avec une simulation suffisamment représentative pour passer par la suite à une implémentation sur le robot réel. Le problème vient de la qualification du terme suffisamment. Ce terme traduit l'équilibre à atteindre entre les hypothèses simplificatrices sur lesquelles reposent la simulation et leur validité relativement à l'application que l'on cherche à simuler.

Dans de nombreux cas et dans l'optique présentée ci-dessus, la simulation d'une application robotique peut reposer sur une modélisation par des solides indéformables articulés. Nous avons restreint notre étude à ce type de simulation tout en sachant que, par la suite, l'interaction avec un sol meuble pourrait être implémentée sous la forme décrite par [ANDRADE-BARROSO, 2000].

## 1.2 Les environnements et les moteurs de simulation de solides indéformables

On constate l'émergence de nombreuses solutions permettant la simulation de systèmes composés de solides indéformables articulés. Ceci s'explique par plusieurs facteurs :

- l'augmentation de la puissance de calcul des ordinateurs personnels qui permet de simuler des systèmes de plus en plus complexes avec un ratio temps simulé/temps de calcul raisonnable (même si la simulation de systèmes constitués d'un nombre important de solides reste très gourmande en ressources physiques telles que mémoire et temps processeur),
- la généralisation de l'utilisation de telles simulations numériques dans certains domaines de l'industrie (p. ex. l'industrie automobile) comme conséquence du facteur précédent,
- le développement de l'industrie du jeu utilisant la simulation de phénomènes physiques afin d'obtenir un certain réalisme,
- le développement de projets basés sur le modèle économique *Open Source*.

L'utilisation de plus en plus répandue du modèle économique de type *Open Source* a joué et joue actuellement un rôle très important dans cette émergence car il implique une plus grande dissémination des logiciels ou des bibliothèques. De ce fait, on distingue les logiciels libres et les solutions commerciales.

### 1.2.1 Architecture

[COUTINHO, 2001] décrit le fonctionnement général d'un moteur de simulation dynamique sous la forme de l'algorithme représenté par la figure 1.1. Ce fonctionnement peut être décomposé en deux types de problèmes inter-dépendants :

- les problèmes d'interactions géométriques ou de détections de collisions,
- les problèmes d'ordre mécanique.

[ERLEBEN, 2004] propose une approche modulaire pour la conception et l'analyse de l'architecture des simulateurs de solides indéformables. Cette approche met en évidence

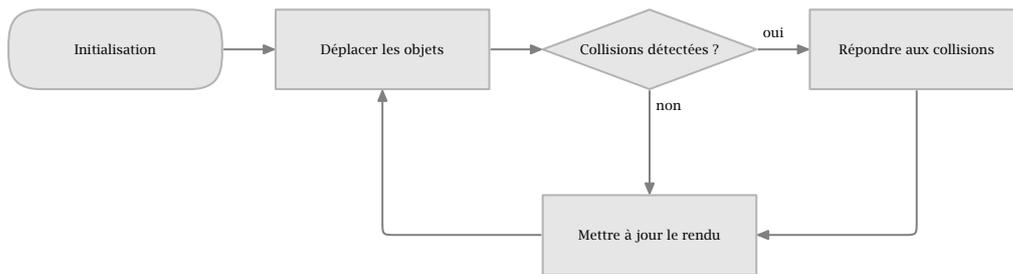


FIGURE 1.1. Structure générale d'un moteur de simulation dynamique, d'après [COUTINHO, 2001]

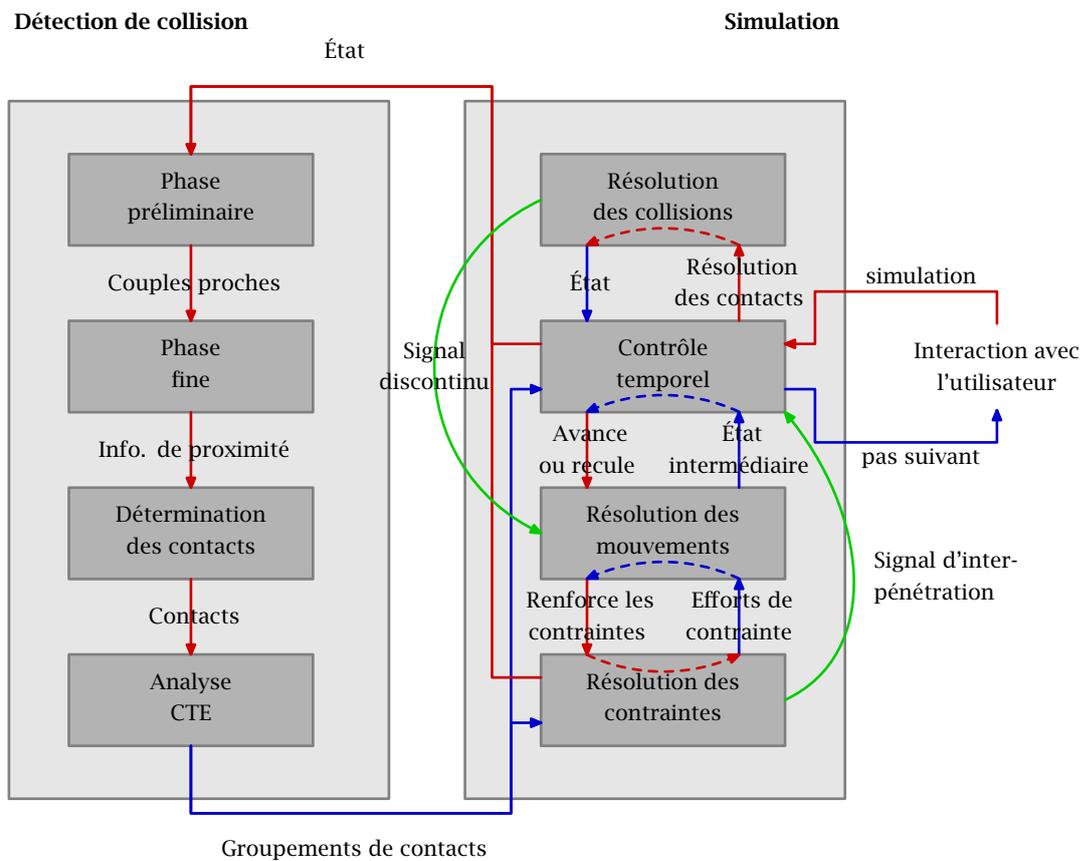


FIGURE 1.2. Architecture générale modulaire complète, d'après [ERLEBEN, 2004]

clairement les deux types de problèmes évoqués ci-dessus. La figure 1.2 représente l'architecture générale modulaire des simulateurs proposée par ces travaux. Cette architecture désigne les problèmes d'ordre mécanique par le terme simulation.

L'intérêt de cette approche réside dans le fait qu'elle fournit un outil complet pour l'analyse et la description du fonctionnement des environnements de simulation physique. Son caractère modulaire et général nous permet de distinguer clairement les différents éléments. On dispose alors d'un outil efficace permettant la comparaison objective de plusieurs simulateurs basés sur des choix d'implémentation ou d'architecture différents.

[ERLEBEN, 2004] fournit une description détaillée de la fonction de chacun des modules

représentés à la figure 1.2 ainsi qu'une présentation des différents types de simulateurs. Nous résumons ici la fonction principale de ces modules dans le but de positionner la bibliothèque de simulation ODE dans le contexte global de la problématique de simulation de solides indéformables.

Le module de détection de collision s'intéresse comme nous l'avons évoqué plus haut au problème purement géométrique d'interaction entre différents volumes. Il comprend les éléments suivants sous forme de modules :

- la phase préliminaire de détection des collisions dont le but est de retourner les couples d'objets proches, c'est-à-dire dont l'intersection est potentiellement non nulle, en comparant toutes les intersections possibles de tout objet avec tout autre objet; une approximation des volumes des objets par des volumes englobants est souvent utilisée lors de cette phase;
- la phase de détection fine des collisions dont le but est de déterminer si les couples d'objets proches s'interpénètrent ou non; cette phase peut également fournir des informations complémentaires sur les interactions inter-objets;
- la détermination des contacts dont le but est de retourner des points de contact en exploitant les informations de proximité fournies par la phase de détection fine des collisions;
- l'analyse de la cohérence temps-espace (CTE) détecte les groupes de contacts indépendant les uns des autres dans le but d'optimiser la simulation.

Le module de simulation a pour objectif de calculer les déplacements des objets simulés. Il est composé des éléments suivants :

- le contrôle temporel qui constitue la partie centrale d'un simulateur; son rôle est de déterminer la séquence suivant laquelle les autres modules interviennent dans la boucle de simulation; le fonctionnement de ce module sera déterminé par l'utilisation d'algorithmes de simulation à incrément temporel soit fixe soit adaptatif;
- la résolution de mouvement, responsable de la gestion des mouvements continus de tous les solides simulés relativement aux équations de mouvement;
- la résolution des contraintes, appelée par le module de résolution de mouvement pour la récupération des efforts de contraintes utilisés pour calculer les déplacements des objets; les efforts de contraintes sont calculés de manière à s'opposer aux interpénétrations entre solides et à satisfaire des conditions de liaison entre les objets;
- la résolution des collisions, appelée par le module de résolution des contraintes, dont le but est de calculer et d'appliquer des impulsions aux objets s'interpénétrant afin de modifier leurs positions relatives; ceci dans le but de minimiser les interpénétrations.

### **1.2.2 Solutions basées sur ODE**

La bibliothèque ODE est une bibliothèque de simulation dynamique de solides indéformables très répandue, notamment pour des applications robotiques. Elle a été initiée en 2000 par Russell SMITH, suite à ses travaux de doctorat [SMITH, 1998]. Au début des recherches présentées dans ce manuscrit, ODE était une des rares bibliothèques de simulation dont le code était disponible et fournissant une interface de programmation relativement simple et complète. Le site web du projet [ODE, 2006] fournit une liste des projets utilisant ODE. Cette bibliothèque est distribuée avec une double licence permettant notamment son utilisation pour des produits commerciaux.

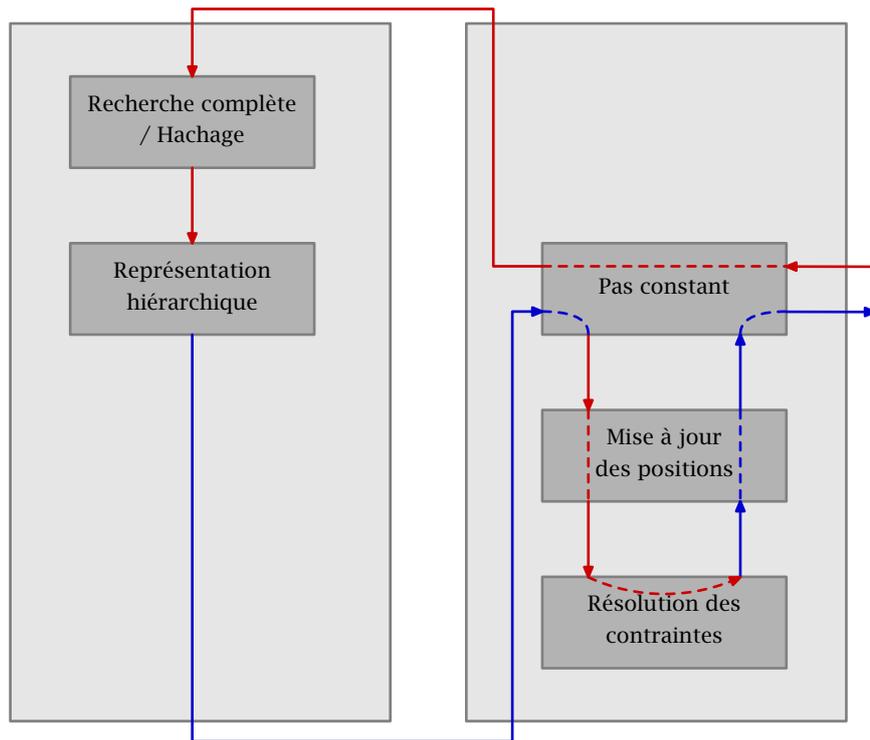


FIGURE 1.3. Modélisation de l'architecture de ODE, d'après [ERLEBEN, 2004]

Pour le domaine de la robotique, on pourra citer les projets libres [SIMULATORBOB, 2006, EVOLVINGROBOTS, 2006, OPENSIM, 2006 et GAZEBO, 2006]. Le simulateur commercial Webots initié par l'EPFL [MICHEL, 2004 et WEBOTS, 2006] est également basé sur cette bibliothèque. Enfin, OpenMASK [OPENMASK, 2006], produit logiciel développé à l'occasion du projet SIAMES, utilise ODE pour une partie de son moteur de simulation dynamique.

La figure 1.3 représente l'architecture de ODE et illustre son fonctionnement. La boucle de simulation utilise un pas de temps fixe. L'annexe A présente le fonctionnement interne de cette bibliothèque sans toutefois détailler le fonctionnement du moteur de détection de collisions.

Nous avons choisi d'utiliser ODE pour le développement de notre environnement de simulation robotique pour les raisons suivantes :

- gratuité et disponibilité du code source,
- multi-plateforme,
- moteur de simulation sous forme de bibliothèque C (facilitant l'intégration au sein d'un programme plus complexe) relativement légère à installer disposant d'une interface de programmation simple et adaptée à nos besoins,
- projet basé sur le modèle *Open Source* bénéficiant d'une communauté relativement active.

### 1.2.3 Autres solutions

De nombreux autres bibliothèques ou environnements de simulation sont disponibles. Là aussi, nous distinguons les produits commerciaux des solutions issues du logiciel libre.

Les produits commerciaux ou gratuits mais à code source fermé (liste non exhaustive) :

- le kit de développement logiciel PhysX d'Ageia [PHYSX, 2006],
- SimMechanics, la boîte à outils de simulation dynamique de solides indéformables pour Matlab Simulink [SIMMECHANICS, 2006],
- le logiciel ADAMS [ADAMS, 2006],
- le moteur de simulation CMLabs Vortex [VORTEX, 2006],
- la bibliothèque Newton Game Dynamics [NEWTON, 2006],
- le kit de développement logiciel Tokamak [TOKAMAK, 2006],
- le kit de développement logiciel Microsoft Robotics Studio [MRS, 2006].

Le problème principal des solutions commerciales est que l'on ne peut que supposer de leur fonctionnement interne puisque l'on n'a pas accès au code source. De plus, pour certaines d'entre elles, il est difficile, si ce n'est impossible, de les intégrer ou les interfacer avec un autre programme. Ajoutons à ces problèmes les arguments de gratuité et de disponibilité du code source, en faveur des solutions libres, nous n'avons retenu aucune des solutions commerciales que nous venons d'évoquer. Enfin, certaines d'entre elles ne sont apparues que très récemment.

En ce qui concerne les bibliothèques libres, une solution récente, quoique non encore totalement aboutie, consiste à utiliser la bibliothèque Bullet Physics [BULLET, 2006] seule ou couplée avec la suite graphique libre Blender [BLENDER, 2006]. On citera également la bibliothèque OpenTissue [OPENTISSUE, 2006] auquel Kenny ERLEBEN participe activement ([ERLEBEN, 2004]). L'installation de cette dernière n'est pas aisée et requiert l'installation préalable de nombreuses bibliothèques tierces. Ces solutions n'ont pas été retenues du fait de leurs apparitions récentes.

### 1.3 Conclusion

Nombre des solutions que nous citons ici n'existaient pas au début de nos recherches. Néanmoins, il convient de les évoquer dans le but de présenter les possibilités alternatives à l'utilisation de ODE à l'instant de l'écriture de ce manuscrit. Les pistes ouvertes récemment par PhysX et Bullet Physics sont les plus prometteuses avec ODE, dont le développement se poursuit. En effet, l'arrivée des PPU PhysX d'Ageia pourrait permettre une augmentation significative des ressources physiques dédiées aux calculs dynamiques. Malheureusement, cela ne sera possible qu'avec le kit de développement logiciel PhysX dont le code source n'est pas disponible. D'autre part, la suite de conception d'image de synthèse Blender fournit déjà un environnement de simulation agréable avec la possibilité d'édition graphique de la géométrie et des propriétés physiques ainsi qu'un interfaçage Python avec Bullet Physics. On peut déjà imaginer la possibilité de simuler des algorithmes de commande écrits avec Python<sup>1</sup>.

---

<sup>1</sup> Python est un langage de programmation interprété portable et dont la syntaxe se caractérise par sa simplicité et sa compacité.



# Chapitre 2

## La problématique de la locomotion des robots de type serpent

---

### Sommaire

---

2.1	La problématique de la locomotion robotique .....	12
2.1.1	Contexte : la robotique d'intervention .....	12
2.1.2	Robots à roues .....	12
2.1.3	Robots à chenilles .....	14
2.1.4	Robots à pattes .....	15
2.1.5	Robots de type serpent .....	16
2.2	La locomotion des serpents terrestres .....	17
2.2.1	Les différents modes de locomotion .....	17
2.2.1.1	Progression par ondulation horizontale .....	18
2.2.1.2	Progression par mouvements d'accordéon .....	19
2.2.1.3	Progression rectilinéaire .....	19
2.2.1.4	Progression par déroulement latéral .....	20
2.2.1.5	Autres modes de locomotion .....	20
2.2.2	Le mode de locomotion par progression rectilinéaire .....	20
2.2.2.1	Mécanisme de locomotion : point de vue élémentaire .....	21
2.2.2.2	Changement de direction .....	24
2.2.2.3	Considérations énergétiques .....	24
2.2.3	Conclusion .....	24
2.3	État de l'art des robots-serpents .....	25
2.3.1	Systèmes robotiques bio-inspirés par l'ondulation latérale .....	25
2.3.2	Les systèmes non basés sur le mode d'ondulation latérale .....	29
2.3.3	Les robots de type serpent et l'apprentissage .....	30
2.3.4	Les autres travaux .....	30
2.4	Conclusion .....	31

---

Ce chapitre aborde la problématique de la locomotion robotique par l'utilisation de robots inspirés par l'observation biologique des modes de locomotion des serpents terrestres. Dans une première partie, nous présenterons le contexte général de la locomotion robotique. Une deuxième partie sera dédiée à l'état de l'art des modes de locomotion des serpents terrestres du point de vue de la biologie. Enfin, la dernière partie présentera les dernières avancées scientifiques de la conception et de l'élaboration d'algorithmes de commande des robots serpents.

---

## 2.1 La problématique de la locomotion robotique

### 2.1.1 Contexte : la robotique d'intervention

Dans ce contexte, le robot a pour but d'effectuer des tâches parfois difficiles à définir *a priori*. L'intervention peut survenir au sein d'une gamme assez étendue d'environnements. L'application elle-même détermine les caractéristiques de l'environnement dans lequel est destiné à évoluer le robot. Ainsi, dans le cas d'une application industrielle sur un site déterminé, l'environnement est relativement bien connu tandis que dans le cas d'une application militaire ou de sécurité civile, l'environnement peut être complètement inconnu. À chaque environnement, qu'il soit contraint ou non, correspond un problème plus ou moins large de mobilité robotique, l'objectif généralement premier du robot étant de pouvoir se rendre à une position donnée afin d'y effectuer la ou les tâches pour lesquelles il a été conçu.

Actuellement, les champs applicatifs les plus actifs de la robotique d'intervention sont :

- la maintenance et l'exploration non critique en milieu industriel,
- la sécurité civile (avec l'exemple du développement de la robotique de sauvetage),
- les applications militaires et policières (renseignement, déminage, etc.)

De plus, l'existence d'un marché commercial en plein essor gravitant autour de ces champs applicatifs témoigne :

- du développement des techniques (ce qui est possible aujourd'hui ne l'était pas hier),
- de l'intérêt des états et des collectivités publiques ou industrielles,
- de l'intérêt scientifique du monde de la recherche envers les problématiques soulevées.

Il faut enfin préciser qu'il existe une certaine convergence entre ces applications. En effet, en considérant les offres commerciales actuelles, on constate que, pour une base robotique donnée, il existe parfois plusieurs déclinaisons, adaptées à des applications différentes, comme nous le verrons par la suite.

Les techniques liées à la robotique d'intervention sont les suivantes :

- la mobilité,
- la téléopération (interfaces homme-machine, transmissions et lois de commande),
- l'autonomie (lois de commande, prises de décision),
- la perception (capteurs et analyse/traitement/restitution de données).

L'évaluation de ces techniques est complexe dès lors que l'on souhaite les utiliser pour des applications en milieux naturels, urbains structurés ou urbains déstructurés. Ce besoin d'évaluer les performances des robots a motivé la création d'aires d'évolution spécifiques (DGA) ou de concours robotiques tel que ELROB [ELROB, 2006] ou encore la RoboCup catégorie Rescue [ROBOCUPRESCUE, 2006].

### 2.1.2 Robots à roues

La première réponse au problème de locomotion robotique est l'utilisation de roues. La gamme de robots basés sur cette solution technologique est très étendue en allant du plus simple robot à commande différentielle et roue folle au plus complexe, disposant d'un châssis poly-articulé actifs. Il est évident que les performances de ces robots, en terme de mobilité, ne sont pas identiques.

Les plus simples des robots à roues sont les robots à châssis non articulé à deux roues motrices et roues folles. Ces robots de conception très simple et peu coûteux disposent



FIGURE 2.1. Hyllos (LRP6)

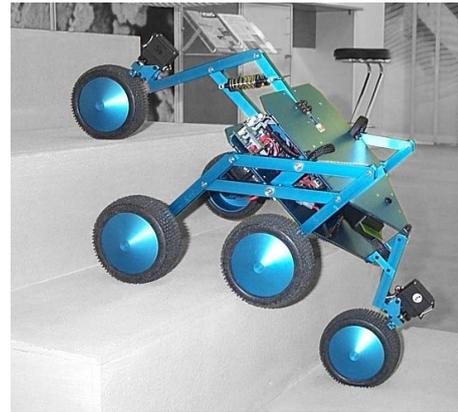


FIGURE 2.2. Shrimp III (EPFL)



Base roulante



Démonstration

FIGURE 2.3. WorkPartner (HUT)

généralement d'une mobilité restreinte à un monde plan. De plus, la non-holonomie de ce type de plate-formes roulantes induit des problèmes de génération de trajectoire. Dans le même cadre applicatif de mobilité sur sols plans, on pourra citer les robots à roues omnidirectionnelles permettant la réalisation de mouvements complexes.

En terme de mobilité, les solutions les plus intéressantes basées sur l'utilisation de roues sont des robots à châssis articulés ou disposant de suspensions à grand débattement.

Les travaux originaux menés à l'EPFL [ESTIER *et al.*, 2000] dont le Shrimp III (figure 2.2) est le résultat sont une bonne illustration des capacités de franchissement dont peuvent disposer les structures à châssis poly-articulé. BlueBotics propose une version commerciale de ce robot [SHRIMP, 2006] à structure passive pour des applications de navigation en extérieur sur terrains irréguliers. La stabilité intrinsèque de cette structure ainsi qu'une relative simplicité de commande sont obtenues par le biais de l'utilisation d'articulations passives.

Des recherches ont été menées dans la direction de structures robotiques hybrides roues-pattes [GRAND *et al.*, 2005 et YLÖNEN et HALME, 2002]. Le robot Hyllos (figure 2.1) du LRP6 est destiné à l'exploration planétaire tandis que la vocation du WorkPartner (figure 2.3) de l'université d'Helsinki est plutôt liée à la robotique industrielle de service en combinant une base mobile et des capacités de manipulation.

Le problème principal des robots à 4 roues est la présence d'une zone morte du point de vue de la locomotion. En effet, le robot peut rester bloqué sur un obstacle en un point milieu de son châssis si la garde au sol n'est pas suffisante. Les châssis articulés et l'augmentation



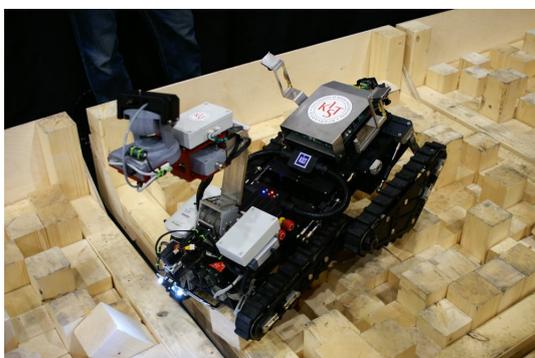
FIGURE 2.4. RobuROC 6  
(ECA - ROBOSOFT)

du nombre de roues sont une réponse partielle à ce problème. L'augmentation du nombre de roues est à relativiser car elle implique généralement une diminution du rendement énergétique due au glissement. La figure 2.4 montre un exemple de robot à 6 roues et châssis articulé : le robot RobuROC 6 [ROBUROC6, 2006], fruit du partenariat entre les entreprises françaises ECA et ROBOSOFT et développé dans le cadre du programme d'études amont DGA MINIROC (Mini Robots de Combat).

### 2.1.3 Robots à chenilles

Les robots à chenilles présentent une alternative aux robots à roues. Ils ont longtemps été délaissés pour des raisons de rendement énergétique. En effet, du fait du glissement obligatoirement induit par l'utilisation de chenilles, la dépense énergétique d'une rotation sur place de la structure est élevée. De plus, ces phénomènes de glissement impliquent de mauvais résultats du point de vue de l'odométrie. Néanmoins, les problèmes de zones mortes observables sur les robots à roues sont minimisés par l'utilisation de chenilles. Enfin, l'utilisation de chenilles permet de répartir le poids du robot sur toute la surface de contact. On obtient ainsi de bonnes performances de progression sur des sols meubles.

Depuis quelques années, de nombreuses recherches ont abouti à la conception de robots basée sur l'utilisation de chenilles en conjonction avec une redondance de la structure comme, par exemple, l'utilisation de flippers mobiles. Ainsi, le robot ROBHAZ, issu de recherches coréennes [ROBHAZ, 2006 et LEE *et al.*, 2005], a été conçu pour répondre à des problèmes de mobilité liés à des applications de sauvetage ou militaires en milieu



Version Rescue 2006



Version militaire (Irak)

FIGURE 2.5. ROBHAZ-DT3 (KIST)

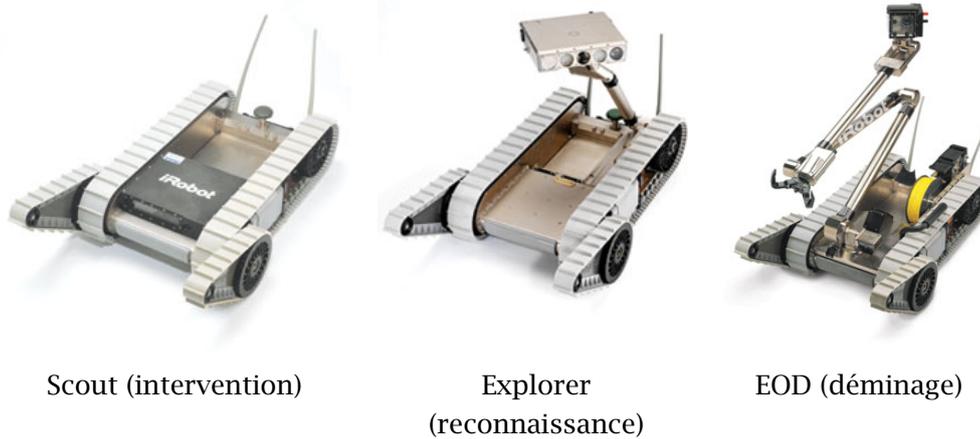


FIGURE 2.6. PackBot (iRobot)

déstructuré. La figure 2.5 montre deux des nombreuses déclinaisons du ROBHAZ-DT3. La base ROBHAZ est également utilisée par l'UNSW pour le développement du robot CASTER [KADOUS *et al.* , 2005] pour la RoboCup Rescue 2005. Les objectifs de CASTER sont, entre autres, la détection/localisation de victimes visibles ou non et la cartographie.

On citera, pour le même type d'application, la famille de robot PackBot [PACKBOT, 2006] représentée à la figure 2.6.

Quelques robots basés sur l'utilisation de chenilles sont inspirés de la biologie des serpents. Nous avons fait le choix de classer ces robots dans la famille des robots serpents présentée un peu plus loin dans ce chapitre.

#### 2.1.4 Robots à pattes

Les robots à pattes regroupent un grand nombre de robots assez différents : bipèdes, humanoïdes, hexapodes, quadrupèdes. L'avantage de la plupart de ces robots est qu'ils sont potentiellement capables de progresser sur les mêmes terrains que leurs modèles. Toutefois, cette caractéristique reste un objectif à atteindre à moyen et long terme.

Les problèmes liés à l'utilisation de pattes pour la robotique sont la stabilisation de la locomotion et la gestion d'un nombre élevé de degrés de liberté.

Concernant les robots quadrupèdes, Sony a initié de nombreuses recherches [HUGEL, 1999, FUJITA, 2001, RÖFER, 2004 et KOHL et STONE, 2004] avec l'industrialisation du robot Aibo, notamment avec la création de la RoboCup en 1998. Le robot Aibo étant de dimensions restreintes, ses capacités de locomotion sur un substrat irrégulier ou encombré ne sont pas très développées. En revanche, cette plate-forme robotique a permis d'élaborer des algorithmes de locomotion très efficaces sur sol plat. Plus récemment, l'entreprise BostonDynamics a développé le robot quadrupède BigDog [BDI, 2006] pour des applications de défense. Les dimensions de ce robot (1 m de long et 70 cm de haut pour une masse de 75 kg) sont plus importantes, ce qui lui permet de disposer de capacités de locomotion supérieures. Les algorithmes de locomotion et de stabilisation de ce robot quadrupède lui permettent de progresser sur toutes sortes de substrat tels que sols boueux, ballast d'une voie de chemin de fer ou encore terrain rocailleux. Enfin, sa capacité à embarquer une charge utile importante, de l'ordre de 60 kg, lui confère des qualités certaines pour des applications de support militaire de forces terrestres opérationnelles. Ce robot est représenté à la figure 2.7.



FIGURE 2.7. BigDog  
(Boston Dynamics)

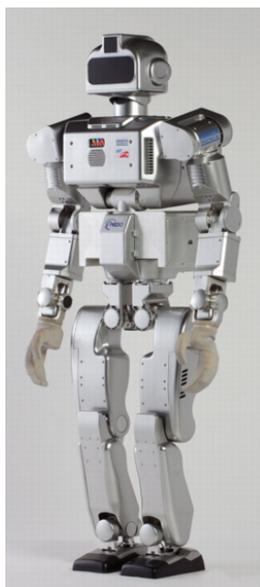


FIGURE 2.8.  
HRP-3P (Kawada)

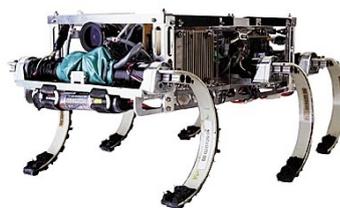


FIGURE 2.9. RHex  
(Boston Dynamics)

Actuellement, la recherche en robotique humanoïde est très active. De nombreux prototypes conçus principalement par des groupes industriels (Honda, Sony ou encore Kawada, pour ne citer qu'eux) en sont les objets. Les robots HRP témoignent de l'intérêt des industriels pour les applications de la robotique humanoïde. La figure 2.8 représente le dernier né des robots HRP, le HRP-3P.

Un exemple de recherche sur la conception de robots hexapodes est le robot RHex (figure 2.9), développé par Boston Dynamics capable de progresser sur tous les types de substrat.

### 2.1.5 Robots de type serpent

Les avantages de la conception généralement modulaire de ce type de robot sont dimensionnels et cinématiques. De manière générale, un robot de type serpent présente deux ordres de grandeurs dimensionnelles différents pour sa longueur et pour sa section. Ainsi, sa longueur est supérieure à l'ordre de grandeur des obstacles tandis que sa section est inférieure à celle des conduits susceptibles d'être traversés. Cet équilibre permet à ce type de robots de s'adapter à une gamme très étendue d'application de locomotion. Enfin, la cinématique redondante de telles structures autorise la réalisation de tâches spécifiques, par exemple en utilisant la structure dédiée à la locomotion pour une tâche apparentée à de la manipulation robotique.

Les problèmes que posent les robots de type serpent sont nombreux :

- conception mécanique,
- interactions avec le sol,
- planification de trajectoire,
- commande.

Les sections suivantes présentent les différents aspects de la locomotion des serpents terrestres du point de vue de la biologie ainsi qu'un état de l'art de la robotique bio-inspirée par les serpents.

## 2.2 La locomotion des serpents terrestres

Les serpents terrestres utilisent la locomotion apode, c'est-à-dire une propulsion essentiellement basée sur le travail des muscles du tronc. Ce type de locomotion est employée d'une manière générale par de nombreuses espèces dont les membres inférieurs, lorsqu'ils existent, assurent des points d'appui plutôt qu'un rôle d'organes moteurs. Certains poissons, de nombreux lézards ainsi que les amphibiens font appel à la locomotion apode. Cependant, les déclinaisons les plus nombreuses de la locomotion apode se retrouvent chez les serpents, faisant de ce groupe le plus représentatif pour l'étude de la locomotion apode. Cet état de fait s'explique par la diversité des milieux et des biotopes<sup>2</sup> dans lesquels évoluent les serpents.

Jusqu'à une période assez récente, les biologistes étaient partagés entre deux hypothèses. La première soutenait que les serpents avaient perdu leurs pattes dans un milieu aquatique. La seconde soutenait que les serpents étaient apparentés aux lézards terrestres et ont perdus leurs membres sur la terre ferme. Des études récentes [VIDAL et HEDGES, 2004 et VIDAL et HEDGES, 2005] apportent la preuve de la validité de la seconde hypothèse, ainsi les serpents terrestres ont perdu leurs pattes sur terre pour améliorer leur capacité de progression au sein de leur environnement.

Cette section traite de la locomotion apode des serpents terrestres. Ne sont pas présentés ici, les modes de locomotion spécifiques non terrestres, c'est-à-dire aquatiques ou arboricoles.

Plusieurs arguments tendent à favoriser l'hypothèse selon laquelle la locomotion apode présenterait un faible coût du point de vue énergétique [WALTON *et al.*, 1990 et GANS, 1980] :

- il n'y a pas de coût associé au déplacement vertical du centre de gravité,
- il n'y a pas de coût associé à l'accélération ou à la décélération des membres.

Après un inventaire des différentes possibilités en matière de reptation, nous nous intéresserons plus particulièrement au mode de progression rectilinéaire. Ce mode présente la particularité d'assurer une progression en ligne droite.

### 2.2.1 Les différents modes de locomotion

On rencontre chez les serpents quatre modes principaux de locomotion terrestre, plus ou moins répandus comme le résume le tableau 2.1.

TABLEAU 2.1. Correspondances et fréquences des modes de locomotion des serpents, d'après [GANS, 1980]

<i>ondulation latérale</i>	<i>mouvements en accordéon</i>	<i>rectilinéaire</i>	<i>déroulement latéral</i>
commun	commun	commun (boïdés, vipéridés et quelques colubridés)	commun pour quelques espèces : influence environnementale importante

<sup>2</sup> En écologie, un biotope désigne un ensemble d'éléments caractérisant un milieu physico-chimique déterminé et uniforme qui héberge une flore et une faune spécifique.

On rappelle dans un premier temps les caractéristiques de chacun de ces modes pour permettre une meilleure compréhension des termes biologiques. On précisera les caractéristiques énergétiques de chaque mode de propulsion.

### 2.2.1.1 Progression par ondulation horizontale

Ce mode, également désigné sous la dénomination de progression par ondulation latérale, est le plus répandu parmi les espèces de serpents terrestres. Il est habituellement pratiqué par des espèces de petite à moyenne taille. Le principe physique régissant la progression par ondulation horizontale repose sur l'utilisation de points d'appui latéraux répartis discrètement. La résultante des efforts ainsi exercés sur le substrat comporte une composante de propulsion suivant l'axe longitudinal du corps du serpent. Le corps adopte alors une forme composée de plusieurs courbes comme le montre la figure 2.10. Lors de la progression, la sélection des points d'appui est établie par la tête du serpent. Ainsi, toutes les parties du corps se déplacent simultanément suivant la trajectoire sélectionnée. L'onde de contraction se propage de l'avant vers l'arrière du corps. Selon [GRAY, 1968], trois points d'appui au minimum sont nécessaires : deux pour générer les efforts et un pour les équilibrer et orienter la résultante.

Outre les pertes énergétiques engendrées par les frottements de la partie ventrale, la dépense énergétique induite par l'équilibrage des efforts latéraux n'est, malgré tout, pas négligeable. Il est établi dans [WALTON *et al.*, 1990] que le coût énergétique de l'ondulation latérale pour un serpent *Coluber Constrictor* est équivalent au coût énergétique de la course pour un animal à pattes (mammifère, lézard, etc.) de masse identique.

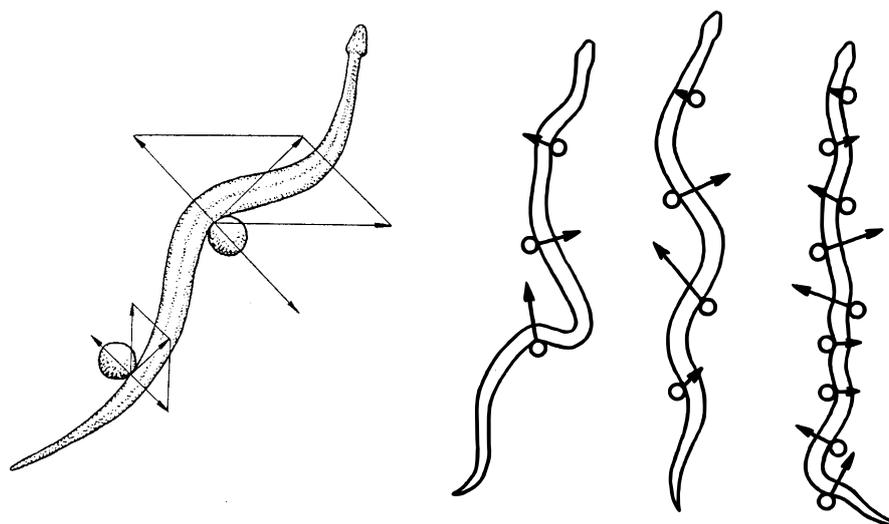


FIGURE 2.10. Illustrations de l'ondulation latérale, d'après [DOWLING, 1997 et GANS, 1980]

### 2.2.1.2 Progression par mouvements d'accordéon

D'après [GASC, 1974], le recours à ce mode de locomotion correspond à une absence de points d'appui suffisamment nombreux. Il est également rencontré lorsque l'utilisation des autres modes se révèle impossible ou inadaptée. Lorsque l'environnement est étroit comme dans le cas d'une branche d'arbre ou encore lorsque le serpent est appelé à progresser dans un milieu confiné (un tunnel p. ex.), la progression par mouvements d'accordéon est quasiment la seule alternative. Dans un premier temps, le serpent replie son corps et prend un certain nombre d'appuis statiques. La partie antérieure de son corps se déplie et progresse vers l'avant pour établir de nouveaux points de contact en amont et pouvoir ramener la partie postérieure de son corps. La répétition de ce schéma lui permet alors de progresser vers l'avant (figure 2.11).

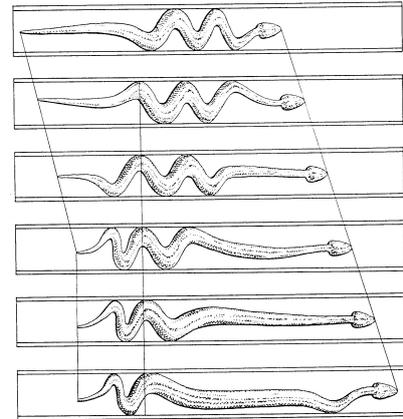


FIGURE 2.11. Illustration de la progression par mouvements d'accordéon, d'après [DOWLING, 1997]

Les observations présentées par [WALTON *et al.*, 1990] établissent que ce mode de locomotion est relativement inefficace en terme de consommation d'énergie. Il est en effet sept fois supérieur au coût énergétique de l'ondulation horizontale. Ce fait s'explique par la perte d'une partie de l'énergie lors de la décélération de la partie mobile lorsqu'elle passe en phase motrice.

### 2.2.1.3 Progression rectilinéaire

Presque tous les serpents sont capables de mettre en application ce mode de locomotion. C'est le mode de prédilection des serpents de masse importante lorsqu'il n'y a pas de danger immédiat.

On peut établir un lien entre la progression rectilinéaire et la progression par mouvements d'accordéon dans le sens où elle constitue un perfectionnement au niveau unisegmentaire. On obtient finalement « la transformation du mouvement alternatif local en un mouvement continu général » [GASC, 1974]. Ce mode de locomotion est traité en détail à la section 2.2.2.

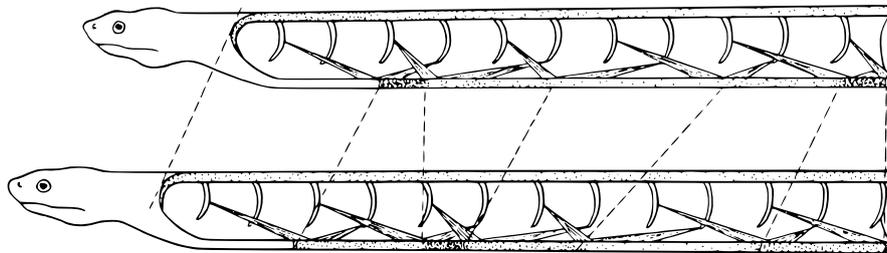


FIGURE 2.12. Illustration de la progression rectilinéaire, d'après [DOWLING, 1997]

#### 2.2.1.4 Progression par déroulement latéral

Ce mode de locomotion constitue également un perfectionnement du mode de progression par mouvements d'accordéon. Il est utilisé dans le cas d'interaction du serpent avec un substrat ne pouvant convenir aux autres modes de locomotion (sables des déserts, typiquement). Les caractéristiques du substrat ne permettent pas d'utiliser des points d'appui discrets statiques. Les différences avec le mode de progression par mouvements d'accordéon portent sur la diminution des points moteurs et l'augmentation de la charge au niveau de ces mêmes points moteurs. L'application physique de ce mode de locomotion repose sur l'utilisation d'au moins deux zones d'appui statiques tandis que le reste du corps est soulevé du sol. La figure 2.13 illustre la progression d'un serpent utilisant le déroulement latéral.

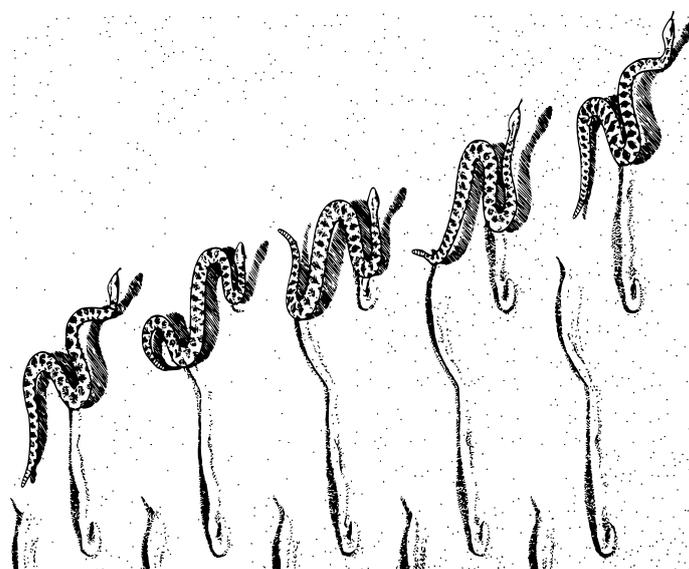


FIGURE 2.13. Illustration du déroulement latéral, d'après [GANS, 1980]

D'un point de vue énergétique, [SECOR *et al.*, 1992] ont établi que ce mode de locomotion est le plus efficace sur ce type de substrat.

#### 2.2.1.5 Autres modes de locomotion

D'autres modes de locomotion peuvent être observés dans la nature, ce sont pour la plupart des modes hybrides ou très spécialisés.

### 2.2.2 Le mode de locomotion par progression rectilinéaire

Cette section traite plus particulièrement du mode de locomotion par progression rectilinéaire. L'intérêt de ce mode de locomotion réside dans la décomposition des fonctions locomotrices et de changement de direction. En effet, l'utilisation de points d'appuis statiques au niveau de la partie ventrale du serpent permet la locomotion tandis que la mobilité de la colonne vertébrale permet les changements de direction. De manière générale, ce

mode de locomotion est observé chez les serpents de grande taille. Néanmoins, la plupart des serpents sont tout à fait capables d'adopter ce mode de locomotion. Dans un premier temps, nous présenterons donc le principe physique à la base de ce mécanisme de locomotion. Ensuite, nous présenterons le mécanisme de changement de direction employé par les serpents adoptant ce mode de locomotion. Cette section se terminera par une étude énergétique.

### 2.2.2.1 Mécanisme de locomotion : point de vue élémentaire

Une idée erronée, issue d'une simple observation externe, consisterait à considérer ce mode de locomotion comme étant un mode de locomotion par ondulation verticale. Cette idée s'appuierait sur le postulat suivant : le serpent marche sur ses côtes, ces dernières étant mobiles par rapport à la colonne vertébrale. Quel que soit le mécanisme interne, il est clair que le serpent utilise une série d'appuis statiques pour progresser. [LISSMAN, 1950 et BOGERT, 1947] ont démontré le caractère erroné de cette explication en proposant celle que nous présentons ci-après.

Considérons l'appareil musculo-squelettique de l'axe vertébral d'un serpent tel que représenté par la figure 2.14. On constate, du point de vue fonctionnel, la présence d'un motif constitué principalement des muscles costo-cutanés inférieur et supérieur. Ce sont ces

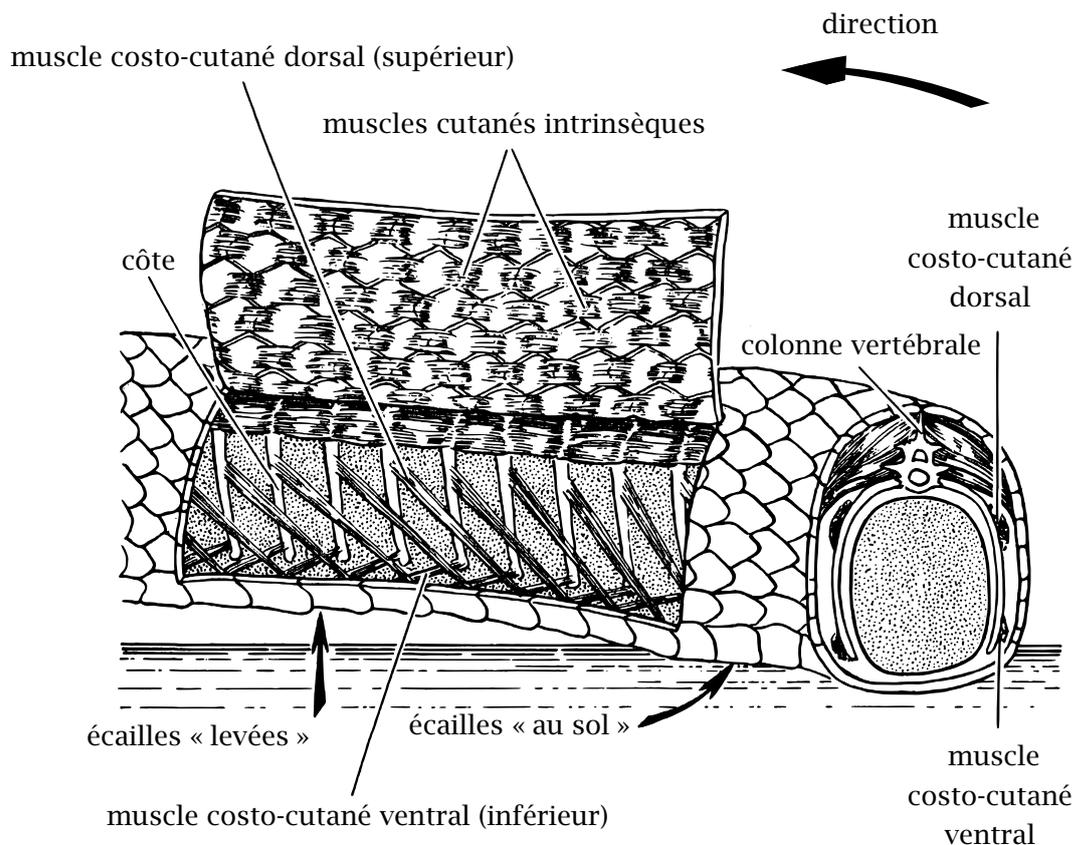


FIGURE 2.14. Reconstruction simplifiée de l'anatomie d'un serpent, d'après [GANS, 1980]

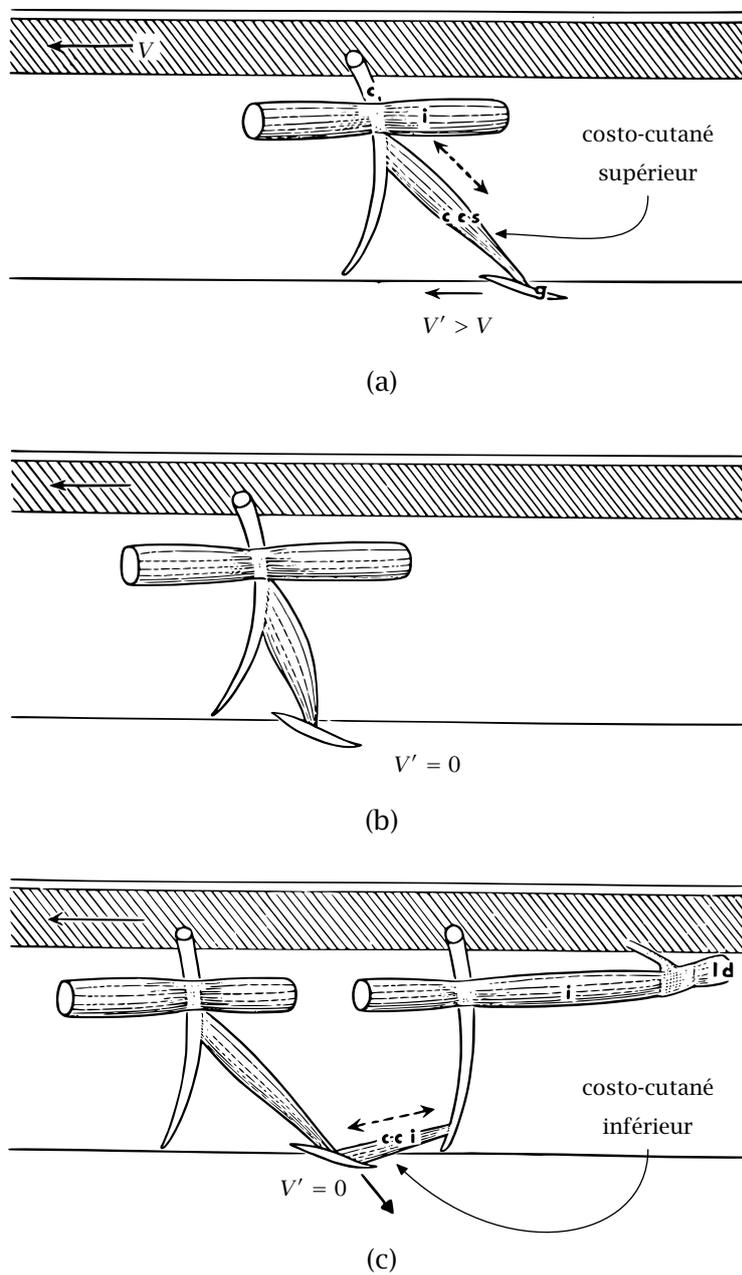


FIGURE 2.15. Décomposition du mode de locomotion rectilinéaire, d'après [GASC, 1974]

muscles antagonistes qui permettent l'exercice de la locomotion rectilinéaire. Les côtes ne bougent pas relativement à la colonne vertébrale. Chaque muscle costo-cutané relie une côte à une section ventrale de la peau du serpent. Ainsi, seule la peau possède un mouvement relatif par rapport à la colonne vertébrale.

La figure 2.15 illustre la décomposition des mouvements à l'origine de la progression rectilinéaire.  $V$  représente la vitesse de déplacement du serpent par rapport au sol.  $V'$  représente la vitesse de la partie ventrale du serpent par rapport au sol. Le mode de locomotion peut être décomposé en trois phases :

figure 2.15(a) Première phase : le muscle costo-cutané supérieur soulève du sol une section ventrale du serpent ;

figure 2.15(b) Deuxième phase : la section soulevée est placée en avant de sa position initiale pour établir un point d'appui statique ;

figure 2.15(c) Troisième phase : le muscle costo-cutané inférieur se contracte. Les écailles ventrales sont en appui sur le sol, propulsant ainsi le serpent vers l'avant.

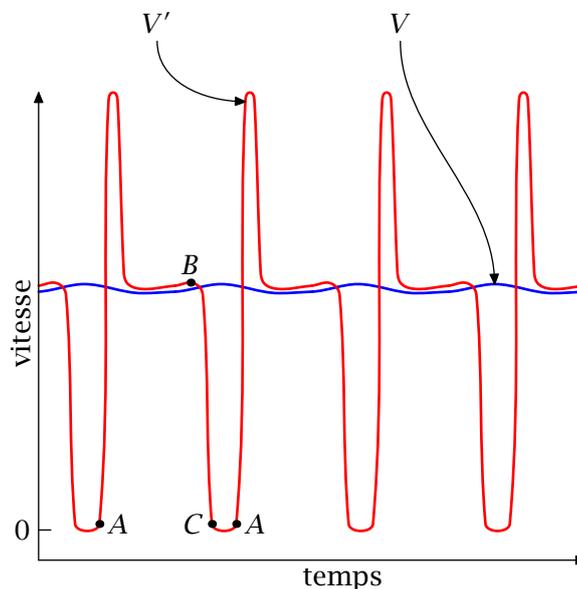


FIGURE 2.16. Vitesses d'une écaille ( $V'$ ) et de la colonne vertébrale ( $V$ ) d'un serpent par rapport au sol (mode de progression rectilinéaire), d'après [GANS, 1980]

La figure 2.16 représente la vitesse d'une écaille et la vitesse du serpent, on identifie le trajet  $AB$  à la première phase, le trajet  $BC$  à la deuxième phase et le trajet  $CA$  à la troisième phase du cycle élémentaire de locomotion. On remarque que la vitesse  $V$  du serpent est presque constante.

D'après [GASC, 1974], deux trains de deux ondes parcourent simultanément le corps du serpent. Ces ondes se propagent le long du corps d'avant en arrière. Les deux ondes considérées sont une onde de contraction des muscles costo-cutanés supérieurs (temps mobile) et une onde de contraction des muscles costo-cutanés inférieurs (temps moteur par réaction du substrat). L'effort résultant de tous les efforts élémentaires implique une progression, en translation, lente et continue suivant l'axe longitudinal (ou antéro-postérieur) du serpent.

### 2.2.2.2 Changement de direction

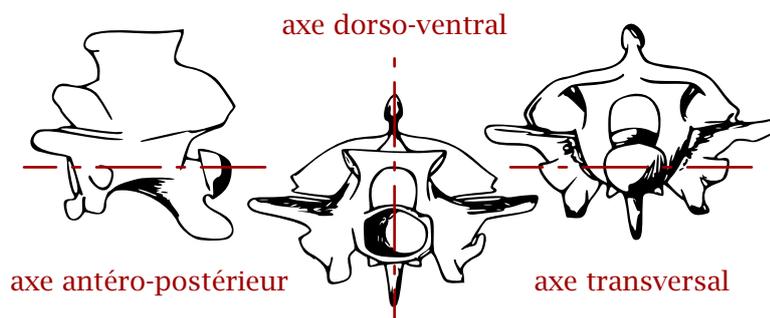


FIGURE 2.17. Vertèbre de serpent

La colonne vertébrale d'un serpent compte de 120 à 400 vertèbres. La figure 2.17 représente une vertèbre de serpent. Les mouvements possibles entre deux vertèbres consécutives sont [GASC, 1974] :

- une rotation suivant l'axe dorso-ventral (flexion latérale) d'amplitude d'environ  $10^\circ$  (os sec),
- une rotation suivant l'axe transversal (flexion dorso-ventrale) d'amplitudes  $5^\circ$  en direction ventrale et  $7^\circ$  en direction dorsale (os sec),
- une rotation suivant l'axe antéro-postérieur (torsion) d'amplitude limitée par des dispositifs extra-osseux,
- une translation suivant le même axe (étirement-compression) d'amplitude limitée par des dispositifs extra-osseux.

Le changement de direction est possible grâce à la mobilité en rotation des vertèbres suivant l'axe dorso-ventral.

### 2.2.2.3 Considérations énergétiques

Le mode de locomotion par progression rectilinéaire n'est pas le plus efficace. Néanmoins, il convient parfaitement aux serpents de masse importante avec un nombre de vertèbres limité (impliquant l'impossibilité d'utiliser le mode par ondulation horizontale). Le principal argument avancé pour juger de l'efficacité de ce mode de locomotion est le principe de conservation de l'énergie cinétique. En effet, le serpent pratiquant ce mode de locomotion possède une vitesse quasiment constante. Comparativement aux autres modes de locomotion terrestre des serpents, sa supériorité par rapport au mode par mouvements d'accordéon est indéniable, ce dernier mode étant très coûteux du fait de phases d'arrêt et de reprise macroscopiques. En revanche, peu de données existent sur un comparatif avec le mode par ondulation horizontale ou le déroulement latéral.

## 2.2.3 Conclusion

Bien que le mode de progression rectilinéaire ne soit pas le mode le plus représentatif de locomotion des serpents terrestres, il reste le mode de prédilection pour les serpents de masse importante. Nous présentons dans la suite de cette étude un mécanisme de locomotion inspiré par ce mode afin de pouvoir le transposer à une application de robotique mobile.

## 2.3 État de l'art des robots-serpents

Nous nous intéresserons plus particulièrement aux robots de type serpent pour la locomotion. Ces robots, bien que redondants du point de vue de la cinématique, ne le sont pas vis-à-vis d'une tâche de locomotion. Si de nombreuses publications traitent de l'application du mode par ondulation latérale, la tendance actuelle s'oriente vers l'étude de nouvelles structures robotiques non principalement basées sur ce mode de locomotion. En effet, on constate dans la littérature une volonté de trouver de nouvelles solutions technologiques pour régler le problème d'interaction avec le substrat ainsi que la génération des efforts propulsifs. On notera notamment que l'utilisation de chenilles est de plus en plus courante sur les prototypes les plus récents.

### 2.3.1 Systèmes robotiques bio-inspirés par l'ondulation latérale

Les premiers travaux concernant l'analyse et la reproduction de la locomotion des serpents appliquées à la robotique mobile sont apparus dans les années 1970 à l'initiative du professeur Shigeo HIROSE au *Tokyo Institute of Technology*. De nombreux prototypes ont été développés au cours de ces recherches et sont présentés dans l'ouvrage de référence [HIROSE, 1993]. Le plus représentatif de ces robots est l'ACM III, présenté par la figure 2.18 et basé sur le mode de locomotion par ondulation latérale. Ce prototype est constitué de plusieurs modules identiques articulés en contact au sol par l'intermédiaire de roues non motorisées. Le principe de locomotion développé par HIROSE repose sur une modélisation des trajectoires des serpents utilisant le mode de locomotion par ondulation latérale, les courbes serpentine<sup>3</sup>. De telles courbes sont définies par



FIGURE 2.18. ACM III (HIROSE - 1972)

$$x(s) = \int_0^s \cos(\xi_\sigma) d\sigma$$

$$y(s) = \int_0^s \sin(\xi_\sigma) d\sigma$$

où

$$\xi_\sigma = a \cos(b\sigma) + c\sigma$$

avec

$s$  l'abscisse curviligne,

$a$  un paramètre correspondant au degré d'ondulation,

<sup>3</sup> [HIROSE, 1993] utilise en anglais le terme *Serpentoid Curves* pour la qualification de ce type de courbes.

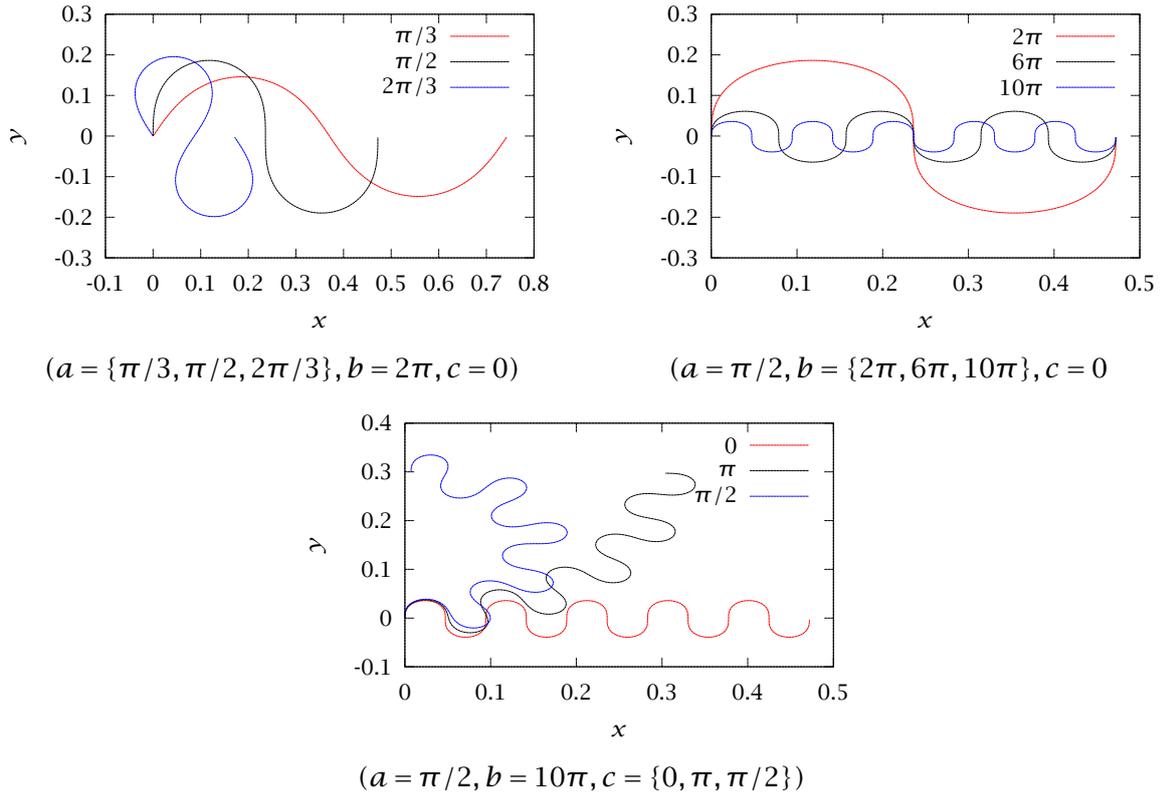


FIGURE 2.19. Exemples de courbes serpentine, d'après [SAITO *et al.*, 2002]

- $b$  un paramètre correspondant au nombre de périodes par unité de longueur,
- $c$  un paramètre correspondant à la courbure macroscopique de la trajectoire.

La courbure d'une courbe serpentine en fonction de l'abscisse curviligne est alors

$$K(s) = \sqrt{\left(\frac{d^2x}{ds^2}\right)^2 + \left(\frac{d^2y}{ds^2}\right)^2} = |ab \sin(bs) - c|.$$

Cette grandeur varie sinusoidalement par rapport à l'abscisse curviligne  $s$ . Elle est relative à l'effort musculaire qu'un serpent doit fournir pour suivre une trajectoire donnée. Par conséquent, les courbes serpentine sont considérées comme des trajectoires naturelles pour un serpent [SAITO *et al.*, 2002 et HIROSE, 1993]. [SAITO *et al.*, 2002] propose une approximation des courbes serpentine par une série de  $n$  segments de longueur  $1/n$  :

$$x(s_i) = \sum_{k=1}^i \frac{1}{n} \cos\left(a \cos\left(\frac{kb}{n}\right) + \frac{kc}{n}\right)$$

$$y(s_i) = \sum_{k=1}^i \frac{1}{n} \sin\left(a \cos\left(\frac{kb}{n}\right) + \frac{kc}{n}\right)$$

avec  $s_i = i/n$ . Des exemples de courbes serpentine sont représentés à la figure 2.19.

Le suivi d'une trajectoire définie par une courbe serpentine s'effectue en boucle ouverte en commandant les angles relatifs entre deux modules adjacents de la manière suivante [SAITO *et al.*, 2002] :

$$\phi_i(t) = \alpha \sin(\omega t + (i - 1)\beta) + \gamma$$

avec

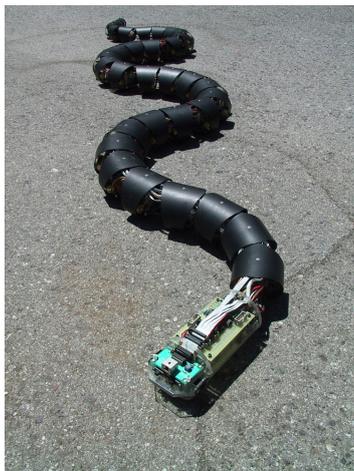
$$\alpha = a \left| \sin \left( \frac{\beta}{2} \right) \right|, \quad \beta = \frac{b}{n}, \quad \gamma = -\frac{c}{n} \quad \text{et } \omega, \text{ la vitesse de propagation de l'ondulation.}$$

Ce principe de locomotion repose sur une condition supplémentaire. Au niveau des points de contact entre le robot et le sol, le rapport des coefficients de frottement  $\mu_{\text{latéral}} / \mu_{\text{longitudinal}}$  doit être élevé afin d'éviter le glissement suivant l'axe normal à la courbe et ainsi permettre la génération d'efforts propulsifs. Le coefficient de frottement suivant l'axe longitudinal doit être assez faible pour permettre un glissement de la structure suivant cet axe. Cette condition s'inspire des caractéristiques de la peau des serpents et, du point de vue de la conception de robot de type serpent, elle est très souvent réalisée par l'intermédiaire de l'utilisation de roues passives. Néanmoins, ce choix de conception restreint les applications aux sols relativement plats et homogènes du point de vue des caractéristiques de frottement. Par conséquent, les capacités d'adaptation au milieu dont font preuve les serpents réels ne peuvent pas être reproduites. Enfin, la génération de mouvements étant réalisée en boucle ouverte afin de suivre une courbe serpentine pré-calculée, on ne peut pas réaliser l'évitement d'un éventuel obstacle situé sur la trajectoire.

Les travaux de HIROSE sont à la base de nombreuses réalisations de robots de type serpent pour la locomotion. Ainsi, les travaux (à titre personnel) de l'américain Gavin MILLER, motivés par ses recherches dans le domaine de la modélisation graphique réaliste [MILLER, 1988], ont abouti à la conception et à la réalisation de nombreux prototypes à roues depuis 1992. La figure 2.20 représente deux de ces prototypes.

De nombreuses contributions s'ajoutent aux principes de base posés par HIROSE pour la conception, la réalisation et la commande de robots de type serpent [MA, 1999, KLAASSEN et PAAP, 1999, PAAP *et al.*, 2000, MA *et al.*, 2001b, DATE *et al.*, 2001b et SAITO *et al.*, 2002], notamment pour la résolution des problèmes de configurations singulières liés à l'utilisation de roues [MATSUNO et SUENAGA, 2002].

Plus récemment, les recherches sur les applications de l'ondulation latérale se sont orientées vers la conception et la commande de robots pour des applications tridimensionnelles. La figure 2.21 montre les derniers prototypes développés au *Tokyo Institute of Technology*. Ces prototypes sont issus des derniers travaux sur l'ACM [HASHIMOTO *et al.*, 2002, YAMADA



S5

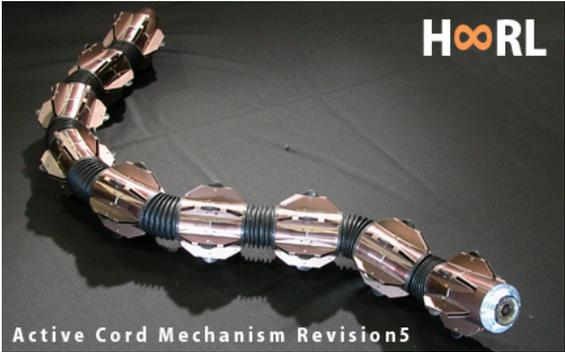


S7

FIGURE 2.20. Robots de MILLER



ACM-R3



ACM-R5

FIGURE 2.21. Robots ACM-RX (HIROSE)



Souryu II



Souryu III

FIGURE 2.22. Robots Souryu (HIROSE)

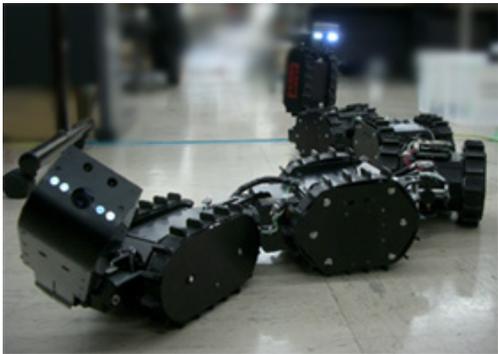


FIGURE 2.23. Robot KOHGA

*et al.*, 2003, YAMAKITA *et al.*, 2003, MATSUNO et SUENAGA, 2003 et MATSUNO et SATO, 2005]. Le robot ACM-R5 n'a pas encore fait l'objet de publication mais la diffusion de l'information par Internet nous a permis de constater son caractère amphibie.

### 2.3.2 Les systèmes non basés sur le mode d'ondulation latérale

L'utilisation de roues pose un certain nombre de problèmes de mobilité lors d'une utilisation sur des sols irréguliers. Ainsi, depuis quelques années, de nouvelles solutions s'inspirant à l'échelle macroscopique des modes de locomotion des serpents ont été proposées. Elles s'appuient généralement sur des applications assez précises. L'équipe du professeur HIROSE préconise dans [TAKAYAMA et HIROSE, 2000 et ARAI *et al.*, 2004] l'utilisation de chenilles pour des applications de sauvetage en milieu encombré avec la famille de robots Souryu (figure 2.22). Le robot KOHGA, représenté à la figure 2.23, est le fruit de recherches [KAMEGAWA *et al.*, 2004 et SATO *et al.*, 2004] répondant au même type de besoin. Le KOHGA participe à la *RoboCup* en catégorie *Rescue*. Ces robots sont autonomes et télé-opérés.



OT-8

OT-4

FIGURE 2.24. Robots OmniTread

Des travaux majeurs dans le domaine de l'inspection et de la surveillance industrielle ont été réalisés au *Mobile Robotics Lab.* de l'université du Michigan. Deux robots sont le fruit de ces recherches : les *OmniTread* OT-8 et OT-4, représentés à la figure 2.24. Ces robots, quoique non autonomes, disposent de capacités de franchissement impressionnantes. Dans [GRANOSIK *et al.*, 2005], en plus de la description des actionneurs pneumatiques, figure une étude assez complète des matériaux pour la réalisation de chenilles. Enfin, ce robot a été évalué en conditions réalistes sur le *Small Robotic Vehicle Test Bed* du *Southwest Research Institute* de San Antonio au Texas. Les résultats obtenus sont très positifs : franchissements d'une voie de chemin de fer, d'une marche de 46 cm de haut ou encore d'une tranchée de 66 cm de large et progression dans un tube de 30 cm de diamètre incliné à 22 ° relativement à l'horizontale.

### 2.3.3 Les robots de type serpent et l'apprentissage

De plus en plus de travaux utilisent des méthodes d'apprentissage pour la conception ou la commande de robots de type serpent. Les travaux de Kevin DOWLING [DOWLING, 1997 et DOWLING, 1999] du *Robotics Institute* de CMU comptent parmi les premiers à appliquer l'apprentissage aux problèmes de la locomotion des robots de type serpent. Outre une étude générale de la locomotion apode, DOWLING présente une étude technologique sur le dimensionnement des actionneurs ainsi que sur les peaux synthétiques. L'apprentissage porte sur la conception de méthodes de locomotion suivant les différents modes observés dans la nature.

Les recherches [ITO et MATSUNO, 2002a et ITO *et al.* , 2003] de MATSUNO du *Tokyo Institute of Technology* portent également sur l'application du *Q-Learning* à la commande de robots redondants (dont les robots de type serpent). De la même manière, [KAMEGAWA *et al.* , 2002] présente des recherches basées sur l'application des algorithmes génétiques à la planification de mouvements de transition pour le changement de mode de locomotion d'un robot-serpent. Des recherches similaires sont proposées par [ERKMEN *et al.* , 2002] pour une application dans le domaine de la robotique de sauvetage.

### 2.3.4 Les autres travaux

L'université de Versailles s'est impliquée dans ces recherches avec une thèse de doctorat [BAYRAKTAROĞLU, 2002] en collaboration avec l'ITÜ. L'évolution récente de ces travaux [BAYRAKTAROĞLU et BLAZEVIC, 2004] a abouti à la réalisation de deux prototypes basés sur l'utilisation de modules identiques. Du point de vue pratique, l'ondulation latérale ne peut être réalisée que par l'intermédiaire de l'usage de points d'appui verticaux. Néanmoins, la conception modulaire proposée permet l'implémentation de plusieurs modes de locomotion. Des travaux similaires [CRESPI *et al.* , 2005a et CRESPI *et al.* , 2005b] sont conduits à l'EPFL avec, toutefois, une valeur ajoutée : le caractère amphibie du robot proposé.

Une équipe du *Tokyo Institute of Technology* travaille également sur une famille de prototypes basés sur des actionneurs pneumatiques originaux dont un représentant, le SSR-I, est présenté par la figure 2.25.



FIGURE 2.25. Robot SSR-I, [AOKI *et al.* , 2004]

## 2.4 Conclusion

Les robots de type serpent sont principalement caractérisés par l'utilisation d'une structure modulaire généralement redondante du point de vue de la cinématique. Si cette caractéristique présente un intérêt certain du point de vue de la locomotion robotique, elle pose néanmoins un certain nombre de problèmes de conception technologique et de locomotion.

Du point de vue de la conception technologique, la répartition d'un nombre important de liaisons et des actionneurs qui les accompagnent impose obligatoirement l'élaboration de solutions innovantes. De plus, ces solutions doivent généralement satisfaire des contraintes importantes d'encombrement. Ces contraintes ne sont pas strictement liées à la conception de robots de type serpent mais plutôt à la conception de robots modulaires inspirés ou non du vivant. Des exemples de robot posant des problèmes de conception similaires aux robots de type serpent sont le robot anguilliforme [BOYER *et al.*, 2006] et, dans un contexte de robotique reconfigurable, PolyBot [YIM *et al.*, 2002]. Les recherches présentées dans ce manuscrit ne traitent pas l'aspect technologique de la conception de robot de type serpent.

Les principes de locomotion des robots de type serpent reposent sur des interactions complexes et distribuées avec l'environnement. Par conséquent, l'étude et la modélisation des organes de propulsion sont nécessaires et sont autant de pré-requis à l'élaboration de lois de commande et d'interface de contrôle. Les travaux que nous présentons dans la suite de ce manuscrit apportent des éléments de réponse à cette catégorie de problèmes pour une structure originale de robot de type serpent.



# Chapitre 3

## Proposition d'un environnement intégré de simulation robotique

---

### Sommaire

---

3.1	Architecture .....	34
3.2	Description fonctionnelle des objets .....	35
3.2.1	Simulateur .....	35
3.2.2	Robot .....	36
3.2.3	Commande .....	39
3.2.4	Trajectoire .....	40
3.2.5	Interface utilisateur .....	41
3.2.6	Gestion de simulation pour l'utilisation de techniques d'apprentissage ..	42
3.3	Algorigrammes .....	43
3.3.1	Programme principal .....	43
3.3.2	Gestion par techniques d'apprentissage .....	43
3.3.3	Simulateur .....	43
3.4	Rendu réaliste .....	48
3.5	Utilisation de cartes de niveaux .....	48
3.6	Conclusion .....	49

---

Ce chapitre présente les travaux effectués sur la conception d'un environnement de simulation intégré pour la robotique.

Il existe de nombreuses méthodes de développement logiciel. À notre niveau, on peut restreindre le choix à deux méthodes principales :

- une méthode inspirée par le cycle en V où une importante phase préalable d'analyse du besoin est obligatoirement nécessaire,
- une méthode inspirée par les méthodes agiles basée sur de nombreux cycles de développement courts et une interaction forte entre le concepteur et le demandeur (un exemple de méthode agile est l'*Extreme Programming*, décrite dans [BECK et ANDRES, 2004]).

Dans notre cas, le concepteur et le demandeur sont la même personne. De plus, il est difficile de définir à l'avance tous les besoins. Par conséquent, nous avons choisi de baser le développement de notre environnement de simulation sur des cycles très court, inspirés des méthodes agiles de développement logiciel, afin de répondre rapidement aux besoins au fur et à mesure de leurs apparitions.

Le besoin initial se définit par :

- l'utilisation de bibliothèques à code source ouvert,
- la simulation de solides indéformables,

- une interface graphique extensible pour la restitution graphique de la simulation et pour l'interaction avec l'utilisateur,
- une grande flexibilité, soit la capacité d'ajouter rapidement et simplement des fonctionnalités supplémentaires (sous la forme d'intégration de bibliothèques logicielles par exemple),
- la portabilité de l'application.

Afin de satisfaire le besoin initial, nous avons choisi de baser notre développement sur les bibliothèques ODE, OpenGL et GTK+ ([GTK+, 2006 et GTKMM, 2006]) en utilisant le langage C++, orienté objet.

### **3.1 Architecture**

L'objectif à atteindre est l'obtention d'un simulateur intégrant et pouvant intégrer rapidement des outils externes. De par la diversité et la qualité des librairies dont il dispose, nous avons choisi le langage de programmation C++ pour développer notre application. L'utilisation de ce langage de programmation reflète le meilleur compromis entre complexité de développement et flexibilité. De plus, le C++ dispose de capacités d'encapsulation permettant d'avoir une architecture orientée objet. Enfin, la bibliothèque de simulation de solides rigides retenue est initialement proposée sous la forme d'une bibliothèque C, aisément interfaçable avec du C++.

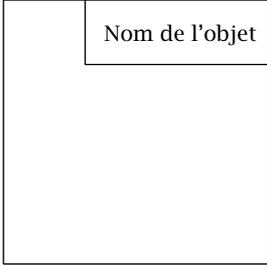
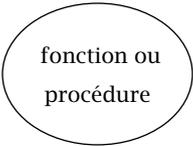
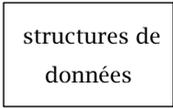
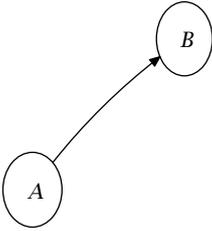
La portabilité de l'applicatif présenté dans ce chapitre est limitée par les spécificités du langage C++. L'application a été développée initialement pour des plate-formes Linux. En revanche, il est tout à fait possible de compiler et d'exécuter ce programme sur une plate-forme Windows sous certaines conditions. S'il est vrai que nous aurions pu employer d'autres langages afin d'accroître les possibilités de portabilité, la perte en terme de performances représentait un coût trop important.

Afin d'expliquer le fonctionnement de notre application, nous aurions pu utiliser une représentation normalisée telle que UML par exemple. Nous avons fait le choix, afin d'alléger les figures et dans un souci de clarté, de mettre au point une convention graphique minimale nécessaire et suffisante pour décrire notre application sans avoir recours à un certain nombre de connaissances informatiques spécifiques. Tout au long de ce chapitre, nous utiliserons donc les conventions graphiques présentées par le tableau 3.1.

Ainsi, en utilisant ces conventions graphiques, la composition de notre application est présentée par la figure 3.1. Les détails internes et les interactions avec les bibliothèques externes ne sont pas représentés sur cette figure. Les objets composant cette application sont :

- le moteur de simulation à partir duquel on gère la simulation dynamique ainsi que les entrées/sorties des fichiers de configuration ou de stockage des données issues de la simulation,
- un objet de définition graphique et dynamique du robot ou de la structure robotique à étudier,
- un objet au sein duquel sera élaborée la commande de la structure robotique étudiée,
- un objet de génération de trajectoire qui permet la création d'une trajectoire en fonction de points de passage ainsi que les calculs d'erreurs de positionnement du robot par rapport à cette trajectoire,

TABLEAU 3.1. Convention graphique utilisée pour la description du programme

Élément graphique	Signification
	Une classe ou une instance de classe (objet). Suivant les cas, cette représentation correspondra à la définition d'une classe ou à l'utilisation d'un objet.
	Une procédure ou une fonction membre de classe. Les variables d'entrées/sorties de la fonction ou procédure ne sont pas détaillées.
	Un ensemble de variables ou de structures de données simple, à l'exclusion des classes définies par l'utilisateur. Cette représentation graphique ne donne pas le détail de chacune des composantes de l'ensemble.
	Cette représentation schématise les interactions entre les différentes parties fonctionnelles du programme. Elle peut représenter une lecture, une écriture ou un appel de fonction en fonction du contexte d'utilisation. Par exemple, si A est une fonction et B une structure de données, la liaison orientée représentera alors l'écriture d'une donnée de B par A.

- une interface graphique que l'on peut utiliser ou non afin de favoriser soit la restitution graphique d'une simulation en cours ou calculée auparavant, soit l'exécution d'une ou de plusieurs simulations sur une machine distante,
- un objet optionnel de gestion des simulations, pour l'utilisation d'algorithmes d'évolution.

Les sections suivantes de ce chapitre donnent une description plus détaillée de chacun de ces objets.

## 3.2 Description fonctionnelle des objets

Dans un premier temps, nous présentons chacun des objets et les fonctionnalités proposées.

### 3.2.1 Simulateur

Le simulateur proposé s'appuie fortement sur la librairie de simulation de solides rigides ODE. Il encapsule les propriétés physiques du monde simulé, les instances de la plupart des autres objets, la gestion des fichiers d'entrée/sortie.

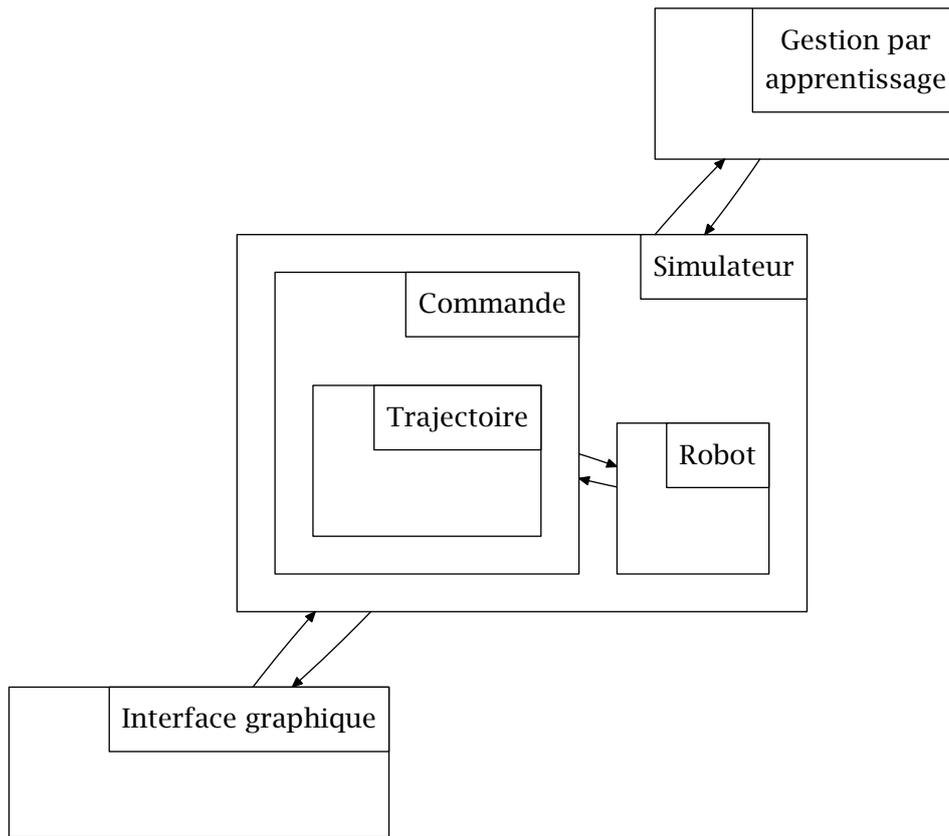


FIGURE 3.1. Architecture simplifiée de l'application

Du point de vue de la simulation de solides rigides, le simulateur contient les éléments essentiels suivants :

- le monde physique (pour l'aspect dynamique),
- l'espace de collision (pour l'aspect géométrique),
- la procédure globale de mise à jour de la simulation,
- les contacts générés par la détection de collision,
- les informations concernant l'environnement (le sol ainsi que d'éventuels obstacles),
- l'information temporelle.

De plus, cet objet est l'objet de plus haut niveau hiérarchique du point de vue de la simulation. Il est donc destiné à contenir les instances des objets « Robot » et « Commande » décrits ci-après.

Toute la gestion de fichiers de configuration et du stockage des résultats de simulation est effectuée par cet objet.

Enfin, il contient également la procédure globale de mise à jour des éléments graphiques.

### 3.2.2 Robot

Tout l'intérêt d'un environnement tel que celui que l'on se propose de décrire dans ce chapitre réside dans la possibilité de simuler le comportement dynamique d'un ou de plusieurs robots au sein d'un environnement plus ou moins complexe. Le robot à simuler pourra être un bras manipulateur, un robot mobile à roues ou encore un robot hexapode. Ainsi, nous n'avons pas défini une structure générique figée qui serait susceptible de nous restreindre dans le choix du type de robot pouvant être simulé. L'utilisateur peut définir

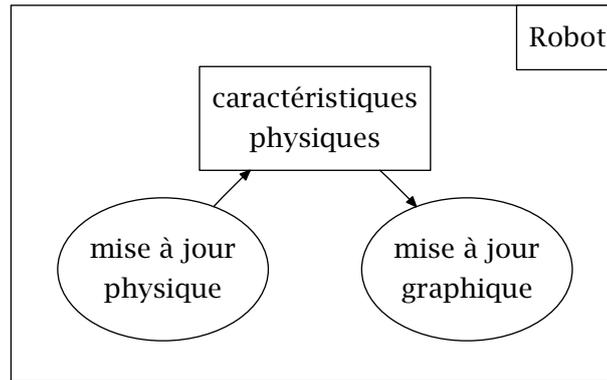


FIGURE 3.2. Composition de l'objet « Robot »

son robot en le définissant sous la forme d'une classe C++. Du point de vue de l'application informatique et des bibliothèques que nous utilisons, un robot sera complètement défini par :

- ses caractéristiques physiques (géométries, sa cinématique interne et ses paramètres dynamiques),
- une méthode de mise à jour purement graphique,
- une méthode de mise à jour de ses paramètres physiques.

La figure 3.2 illustre le découpage générique d'un robot tout en faisant apparaître les liens de dépendances entre les différents sous-ensembles. Dans le cas d'un robot, on obtient un objet de structure très simple car nous avons fait le choix de regrouper de nombreuses informations au sein de l'ensemble de structures de données « caractéristiques physiques ».

Dans la suite de cette section, nous allons décrire l'utilité et la constitution de chacun des sous-ensembles, au sens de la modélisation informatique, d'un robot. Il faut néanmoins préciser que la mise à jour de la dynamique du robot se décompose en deux parties :

- une partie explicite ou locale gérant des consignes prédéterminées comme les consignes des actionneurs internes au robot par exemple,
- une partie implicite ou globale qui sera gérée par le moteur de simulation et qui traite de tous les autres aspects de l'évolution de la dynamique du robot comme l'effet de la gravité sur les solides constituant le robot par exemple.

La gestion globale de la dynamique sera abordée dans la section 3.2.1.

Enfin, du point de vue de l'implémentation informatique, tout robot ou structure robotique sera décrit sous la forme d'une classe C++. Ce choix se justifie principalement par l'analogie entre l'approche objet inhérente à l'utilisation du C++ et le fait de considérer un robot réel comme un objet. Cette approche nous permet alors :

- de respecter une cohérence relative avec la réalité physique,
- de simuler plusieurs instances d'un même robot,
- de disposer de plusieurs robots et de les faire évoluer au sein de la même simulation,
- de composer une structure robotique complexe à partir de robots définis au préalable.

Tout au long de ce chapitre, les notions de création et de destruction des objets seront implicites. Tous les langages orientés objet utilisent ces notions et nous n'avons pas jugé utile de surcharger la représentation des objets en les faisant apparaître.

## Caractéristiques physiques

Le terme « caractéristiques physiques » est utilisé ici d'une manière générique pour dé-

TABLEAU 3.2. Structures de données ODE pour la définition d'un robot

	Solides rigides	Volumes géométriques	Articulations
Structure de données ODE	dBodyID	dGeomID	dJointID
Utilisation	stockage de la dynamique d'un solide rigide	détection de collision du solide avec l'environnement (ou un autre solide)	définition des liaisons entre solides

signer les variables et méthodes utiles à la simulation du robot et n'appartenant pas aux autres sous-ensembles.

Les premières variables nécessaires à la simulation seront définies en utilisant les structures de données spécifiques au moteur de simulation ODE. Le tableau 3.2 présente ces structures de données.

Ainsi, à chacun des solides composant le robot on associera une variable de type dBodyID et une variable de type dGeomID. De la même manière, à chacune des liaisons internes du robot sera associée une variable de type dJointID.

L'objet robot étant une instance de classe, les méthodes de création et de destruction agissent sur les structures de données des caractéristiques physiques. La méthode de construction devra définir la géométrie complète du robot lors de son état initial. Ceci inclut la définition des solides, de leurs géométries et des liaisons internes au robot.

Durant la simulation, l'état courant du robot sera stocké implicitement dans les structures de données ODE.

## Mise à jour graphique

La mise à jour graphique est une simple méthode de classe qui utilise la géométrie des solides composant le robot pour en construire une représentation graphique.

Cet objet repose sur l'utilisation de deux bibliothèques :

- ODE pour la récupération de la géométrie, de la position et de la rotation de chacun des solides constituant le robot,
- OpenGL pour la restitution du robot via l'utilisation de primitives graphiques.

Les appels à la librairie OpenGL seront la plupart du temps indirects dans le but de simplifier la programmation de cette méthode. L'espace de nommage Dtk regroupe des procédures C implémentant le dessin de primitives géométriques directement à partir d'objet dGeomID. Le recours à ces procédures augmente la lisibilité, la simplicité ainsi que la maintenance de l'objet robot. On peut toutefois toujours utiliser directement des appels OpenGL au sein de cette méthode.

Le comportement des liaisons se traduisant par des mouvements relatifs entre solides, nous n'avons pas prévu de les représenter explicitement.

## Mise à jour physique

Les changements physiques implicites tels que l'influence de la gravité ou les interactions

avec l'environnement sont gérés par le simulateur. Il en va de même pour les interactions issues de la détection de collision entre plusieurs solides d'un même robot.

La mise à jour physique implémente uniquement les interventions du point de vue de l'utilisateur, c'est-à-dire de l'actionnement du robot. On peut, par exemple, implémenter cette méthode pour appliquer une vitesse articulaire à une articulation précise du robot à partir de l'interface utilisateur ou encore de l'objet de commande.

### 3.2.3 Commande

Dans le cadre d'une application destinée à la simulation robotique, la commande tient un rôle prépondérant. Comme cela est le cas pour l'objet « Robot », si l'on souhaite une application flexible utilisable par un plus grand nombre, au sens de la diversité des applications, il est raisonnable de concevoir la commande en tant qu'objet. De la même manière que pour l'objet « Robot », plusieurs objets de commande (différents et identiques) pourront coexister au sein de l'application. Néanmoins, les commandes développées au sein de cette application devront respecter une trame du point de vue de l'architecture logicielle afin d'obtenir une inter-opérabilité minimale avec l'environnement de simulation. Afin d'illustrer ce propos, la figure 3.3 représente schématiquement la composition de l'objet « Commande ».

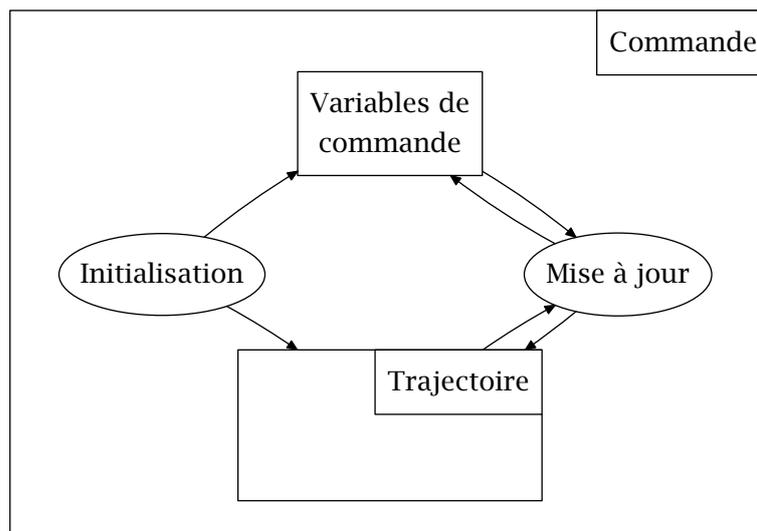


FIGURE 3.3. Composition de l'objet « Commande »

#### Paramètres de commande

Les paramètres de commande correspondent aux consignes à envoyer au robot simulé. Ces paramètres sont fortement dépendant du type de robot et de la nature de la commande employée. Par exemple, dans le cas d'un robot manipulateur, la commande pourra être dynamique ou cinématique. Ainsi, on n'utilisera pas la même nature de consigne et le même modèle pour la commande. Ces paramètres sont à définir en fonction de l'application et en adéquation avec la définition des paramètres de commande d'un robot tels que présentés à la section 3.2.2.

## Initialisation

Cette méthode est utilisée pour initialiser les variables de commande nécessaires à la simulation d'algorithmes de commande. Ces variables incluent évidemment, outre les variables de type simple, les structures de données plus complexes telles que, par exemple, la structure représentant la courbe à suivre si la tâche correspond à un suivi de trajectoire.

## Mise à jour

La commande dépendant de l'état courant de la simulation, elle nécessite un calcul à chaque nouveau pas temporel. Cette méthode fournit l'interface adéquate de mise à jour de la commande au simulateur. Les calculs inhérents à la commande seront réalisés à ce niveau.

### 3.2.4 Trajectoire

Dans le cas général, de nombreuses applications de commande robotique nécessitent l'utilisation d'une trajectoire de référence. Cette trajectoire de référence fait partie de la définition de la tâche à exécuter par l'intermédiaire de la commande. Ainsi, l'objet « Trajectoire » est ici considéré comme un sous-ensemble de l'objet « Commande ».

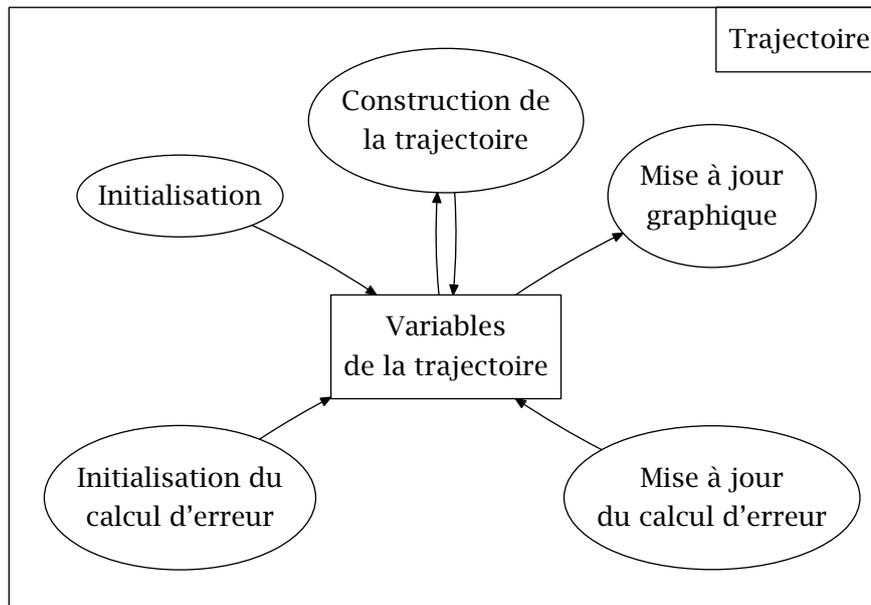


FIGURE 3.4. Composition de l'objet « Trajectoire »

La figure 3.4 représente schématiquement la composition de cet objet.

Cet objet utilise les interpolations fournies par la bibliothèque GSL.

## Variables de la trajectoire

Afin de construire une trajectoire exploitable du point de vue de la commande, nous utiliserons des structures de données. Ces structures de données peuvent être entrantes,

sortantes ou internes. Par exemple, la liste de points de passage en entrée ou sous la forme d'un tableau de valeurs interpolées en sortie. Enfin, les données intermédiaires de commande spécifique à la trajectoire telles que l'erreur en position seront stockées au sein de cet ensemble.

## Initialisations

L'initialisation de l'objet « Trajectoire » correspond à l'utilisation des points de passage stockés dans les variables de trajectoire pour la création d'une courbe interpolée sous la forme d'une structure de données. Cette structure peut ne pas être directement exploitable par la simulation, d'où la nécessité de la procédure membre « Construction de la trajectoire » afin de rendre exploitable cette interpolation du point de vue de la simulation.

L'initialisation du calcul d'erreur prépare simplement les variables attachées au calcul de l'erreur. Le choix de ces variables d'erreur dépend fortement de la commande utilisée.

## Construction de la trajectoire

Cette méthode utilise la courbe d'interpolation sous sa forme initiale pour générer les structures de données exploitables par la simulation.

## Mises à jour

La trajectoire est directement assimilable en tant qu'objet graphique. Ainsi, de la même manière que pour les robots ou l'environnement, il nous faut implémenter une fonction de mise à jour graphique afin de correctement gérer son affichage.

L'erreur se doit d'être mise à jour régulièrement pour que la commande soit cohérente dans le cas où la tâche à effectuer est un suivi de trajectoire. Ainsi, il est logique de prévoir une procédure au sein de l'objet « Trajectoire ».

### 3.2.5 Interface utilisateur

Un des principaux objectifs du développement d'un environnement intégré de simulation est de pouvoir interagir avec l'utilisateur. Ces interactions peuvent être :

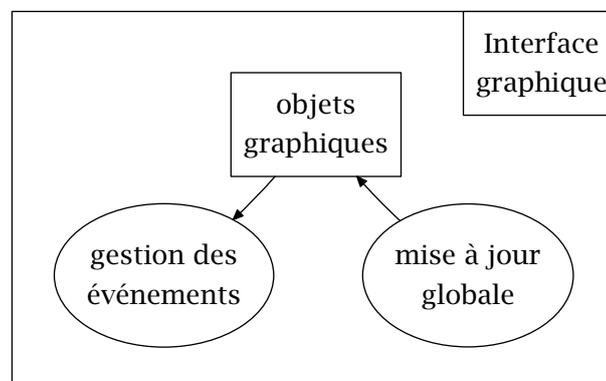


FIGURE 3.5. Composants génériques de l'interface utilisateur

- veille et observation de la simulation en cours en changeant de point de vue,
- modification de paramètres de simulation ou des consignes de commande du robot,
- réinitialisation d'une simulation,
- visualisation d'une simulation calculée auparavant.

La bibliothèque utilisée pour le développement de l'interface graphique est gtkmm. On dispose à l'aide de cette bibliothèque d'une palette complète d'outils graphiques, de gestion de tâches ainsi que de fichiers. L'utilisation de cette bibliothèque nous permet de disposer d'un environnement de simulation multi-plateforme.

De plus, la restitution tridimensionnelle de la simulation est implémentée par le biais d'OpenGL. La bibliothèque de simulation ODE dispose à l'origine d'une interface OpenGL. Néanmoins, ses possibilités d'interaction se limitant au changement de point de vue et à la programmation de fonctions ou procédures liées aux touches du clavier, nous avons choisi de réimplémenter cette partie de l'environnement pour pouvoir disposer d'une meilleure flexibilité. Le module développé se présente sous la forme d'un espace de nommage, Dtk (pour Drawing toolkit).

Enfin, le paramétrage des simulations est basé sur l'utilisation de fichiers de configuration. Il est toutefois tout à fait possible d'imaginer et d'implémenter une interface de modification de ces paramètres conçue à l'aide de gtkmm. Dans tous les cas, les fichiers de configuration sont modifiables par l'intermédiaire d'un simple éditeur de texte et réutilisables avec une interface de configuration.

La mise à jour globale est effectuée par l'intermédiaire de cet objet dans le cas où l'on utilise l'interface graphique. Dans le cas contraire, ce sera le simulateur qui se chargera de cette tâche.

### 3.2.6 Gestion de simulation pour l'utilisation de techniques d'apprentissage

L'intégration de techniques d'apprentissage est rendue possible par l'utilisation de la librairie EO. L'utilisation de stratégies évolutionnaires pour la commande est une option possible. Elle sous-entend en terme d'implémentation la gestion d'un ou de plusieurs jeux de simulations ainsi que la sauvegarde des résultats. Cette gestion des simulations dédiée à l'application de techniques d'apprentissage est implémentée sous la forme d'une tâche activable prenant la main (hiérarchiquement parlant) sur le simulateur. On peut, au sein de cette

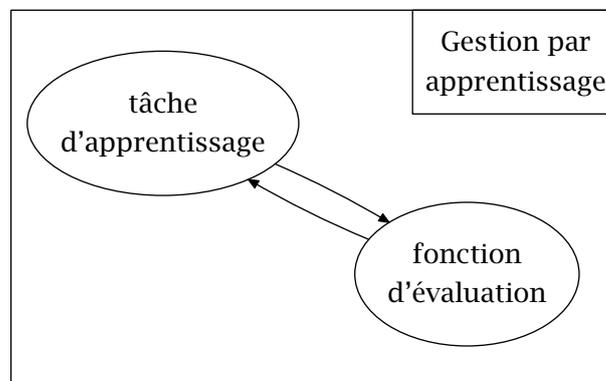


FIGURE 3.6. Composition de l'objet « Gestion de l'apprentissage »

---

tâche, lancer une simulation, modifier des paramètres, les sauvegarder, relancer et gérer de nouveaux individus sous la forme de simulations alimentant l'algorithme d'évolution.

## 3.3 Algorigrammes

Nous avons vu précédemment l'aspect fonctionnel des sous-ensembles du programme proposé. Nous nous intéressons dans la suite au fonctionnement temporel du programme.

### 3.3.1 Programme principal

La figure 3.7 décrit le fonctionnement du programme principal. On y distingue clairement les caractères optionnels de l'interface graphique et de la gestion des simulations par algorithmes évolutionnaires. Ces options sont déterminées à l'exécution du programme par un simple passage d'options de ligne de commande. Un cas simple d'utilisation est l'exécution sur une machine distante de simulations gérées par un algorithme évolutionnaire. Ceci n'étant possible que sur des plate-formes de type Linux, cela justifie en partie le choix de cette plate-forme pour le développement de cette application.

### 3.3.2 Gestion par techniques d'apprentissage

La figure 3.8 décrit le fonctionnement de la bibliothèque EO. Du point de vue de l'implémentation, cette tâche est automatisée à partir du moment où l'on donne la fonction d'évaluation et les paramètres de l'algorithme évolutionnaire. La fonction d'évaluation consiste dans notre cas à lancer une simulation respectant les paramètres de l'individu à tester. Le simulateur prend alors la main et ne la rend que lorsque la condition de terminaison de la simulation est satisfaite. La tâche de gestion préempte en traitant l'évaluation retournée par la simulation et passe à l'évaluation d'un autre individu, soit, une autre simulation. Les individus sont initialement issus de l'initialisation d'une population. L'évaluation de tous les individus de la population initiale est obligatoire. Une fois la population évaluée, EO vérifie si un comportement optimal local a été trouvé. Si cela n'est pas le cas elle applique un classement des meilleurs individus de la population ainsi que des opérations telles que croisement ou mutation afin de créer une nouvelle population. Cette nouvelle population subit alors le même traitement que la population initiale. Si l'algorithme évolutionnaire trouve un optimum local, la population gagnante est retournée. Dans le cas contraire, la meilleure population est retournée en précisant qu'aucun optimum local n'a été trouvé.

### 3.3.3 Simulateur

La figure 3.9(a) représente l'initialisation de l'objet « Simulateur ». Tous les sous-objets du simulateur sont créés à partir de cette initialisation. L'algorigramme de la figure 3.9(b) illustre la lecture des paramètres de la simulation tels que :

- les caractéristiques de l'environnement (type de sol, obstacles, paramétrage des contacts avec le sol),
- le ou les types de robots à simuler et leurs éventuelles caractéristiques (le nombre de modules d'un robot de type serpent p. ex.),
- le pas et la durée de la simulation.

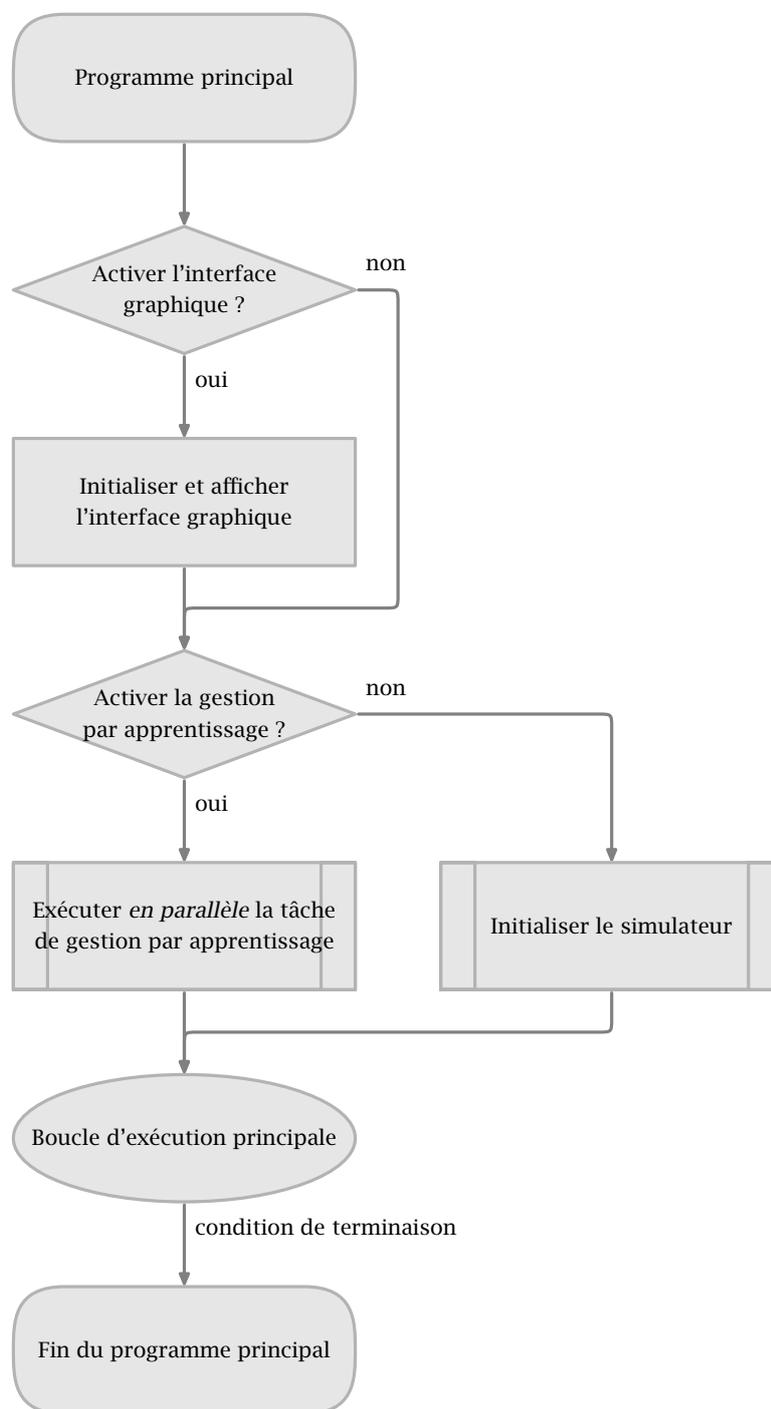


FIGURE 3.7. Algorithme de l'application

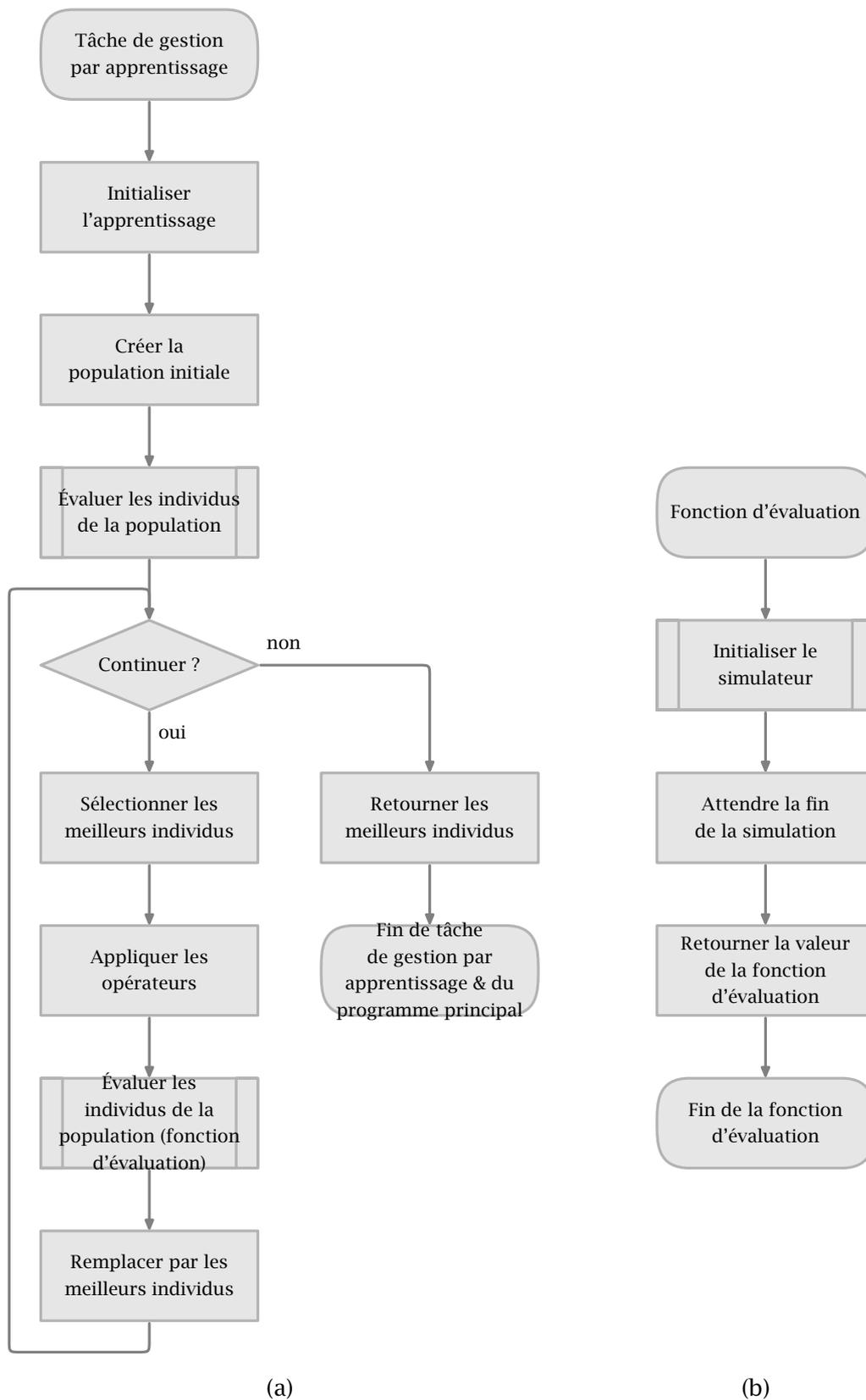
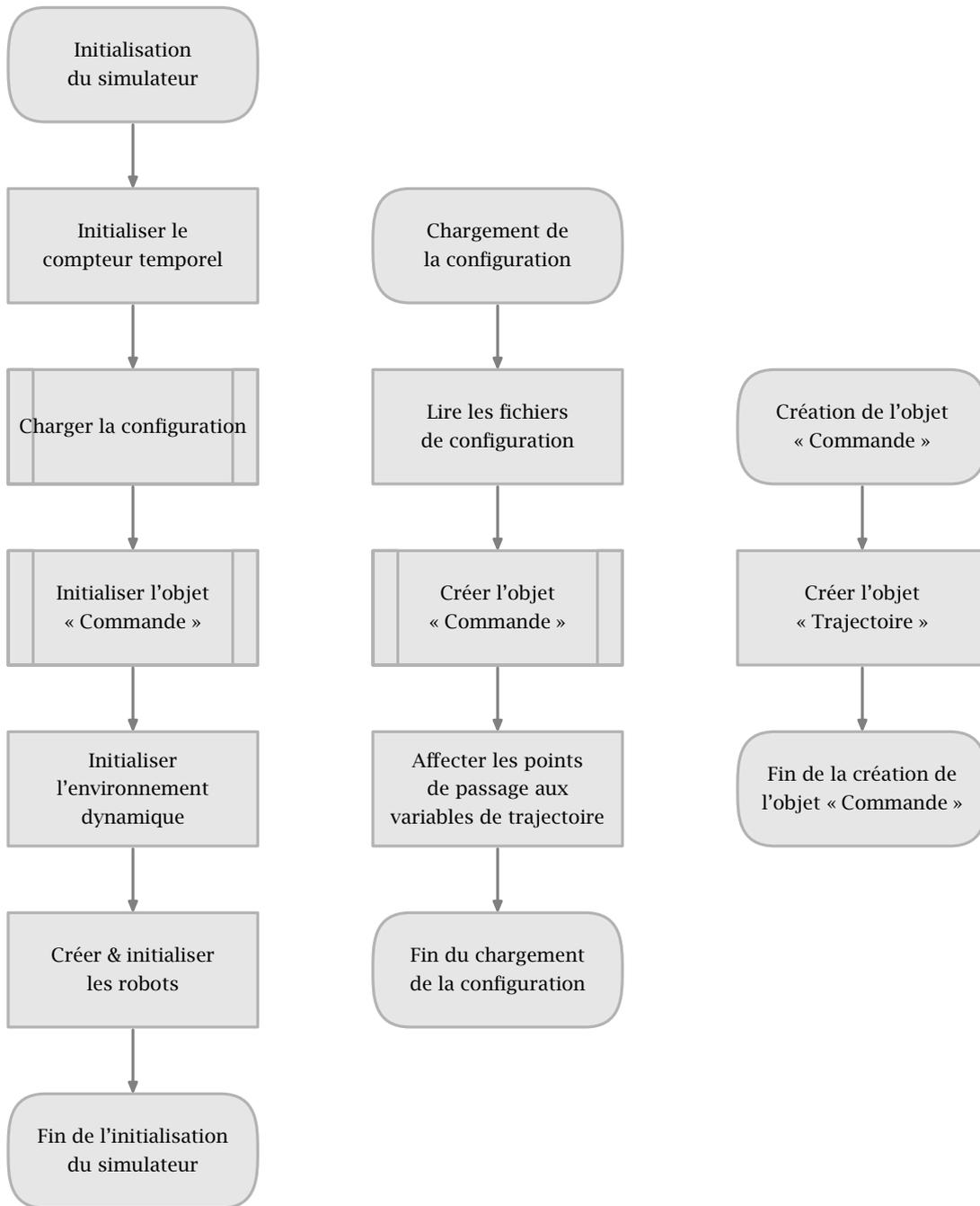


FIGURE 3.8. Algorigrammes (a) de la tâche de gestion par algorithme évolutionnaire et (b) de l'évaluation des individus



(a)

(b)

(c)

FIGURE 3.9. Algorithmes de (a) l'initialisation du simulateur, (b) du chargement de la configuration et (c) de la création de l'objet « Commande »

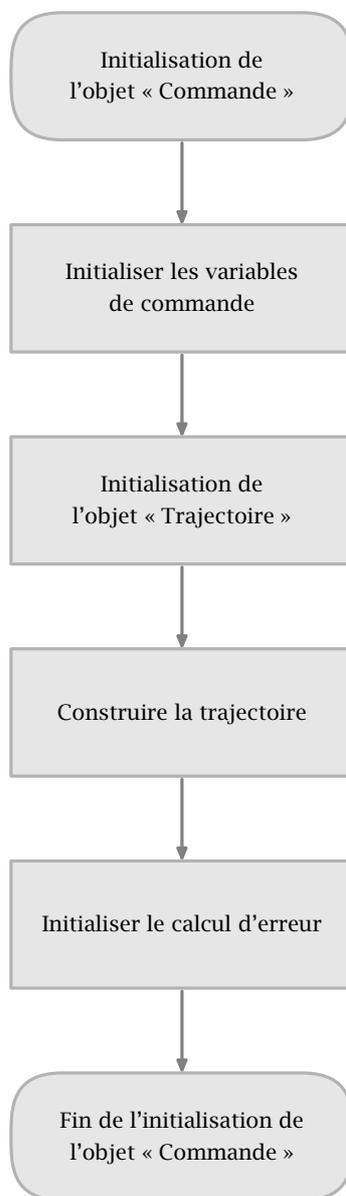


FIGURE 3.10. Algorigramme de l'initialisation de l'objet « Commande »

La figure 3.10 détaille l'initialisation de l'objet « Commande ». La création et l'initialisation de l'objet « Robot » ne sont pas détaillées ici.

Le détails de l'algorithme correspondant à la boucle principale d'exécution, telle que représentée par la figure 3.7, ne font pas l'objet d'une figure du fait de la simplicité de la séquence. Le fonctionnement de cette boucle d'exécution principale est très simple car elle se contente d'appeler séquentiellement les fonctions de mise à jour de tous les sous-objets du programme.

### 3.4 Rendu réaliste

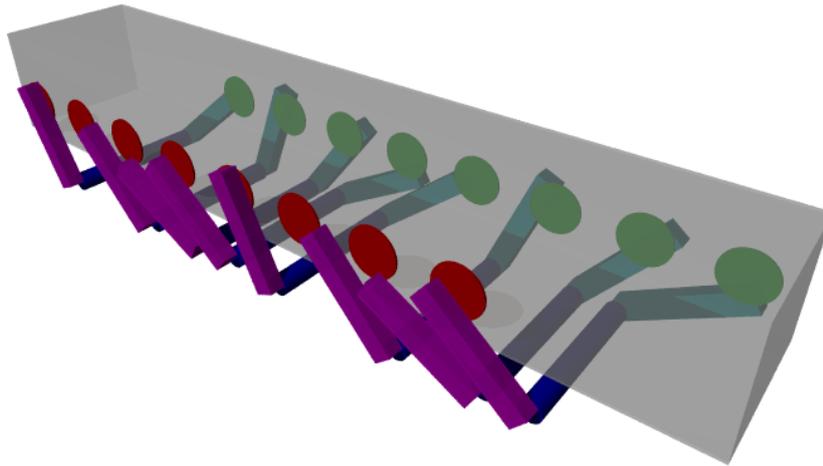
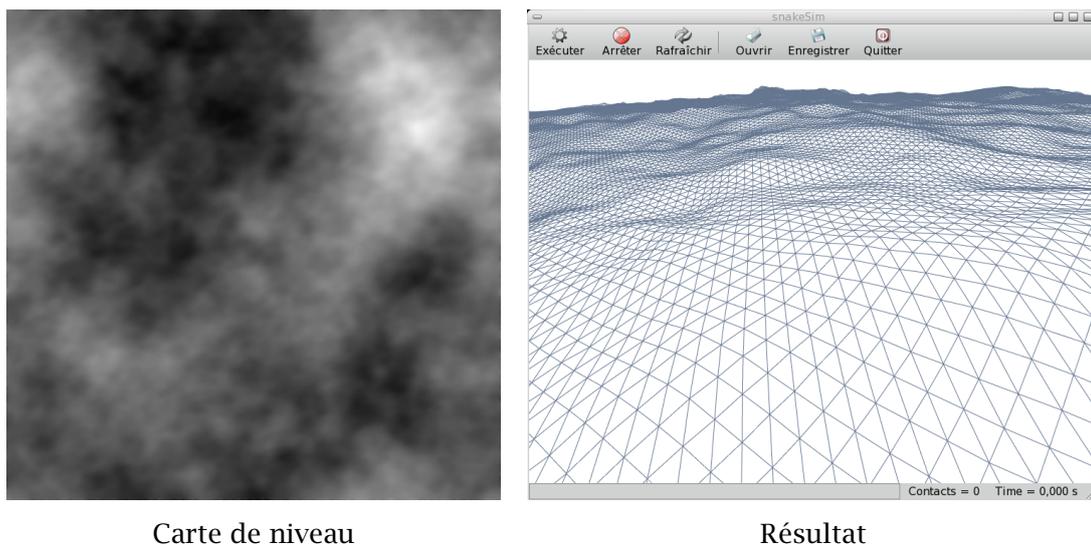


FIGURE 3.11. Vue en transparence d'un module à 8 mécanismes obtenue avec POV-Ray

Lors de l'implémentation de l'applicatif proposé dans ce chapitre, il nous a semblé intéressant de pouvoir restituer graphiquement une simulation avec une qualité supérieure à celle proposée par la bibliothèque OpenGL. Ainsi, le fichier binaire de sauvegarde de simulation peut être utilisé, par le biais d'une application autonome, pour la génération d'images de synthèse. Les images de synthèse sont obtenues par lancé de rayon lumineux avec le logiciel POV-Ray [POV-RAY, 2006]. Le résultat permet d'illustrer la géométrie d'un module de notre robot par une image statique comme en témoigne la vue en transparence d'un module représentée à la figure 3.11. En outre, un tel outil facilite la démonstration du principe de locomotion par l'utilisation d'animations.

### 3.5 Utilisation de cartes de niveaux

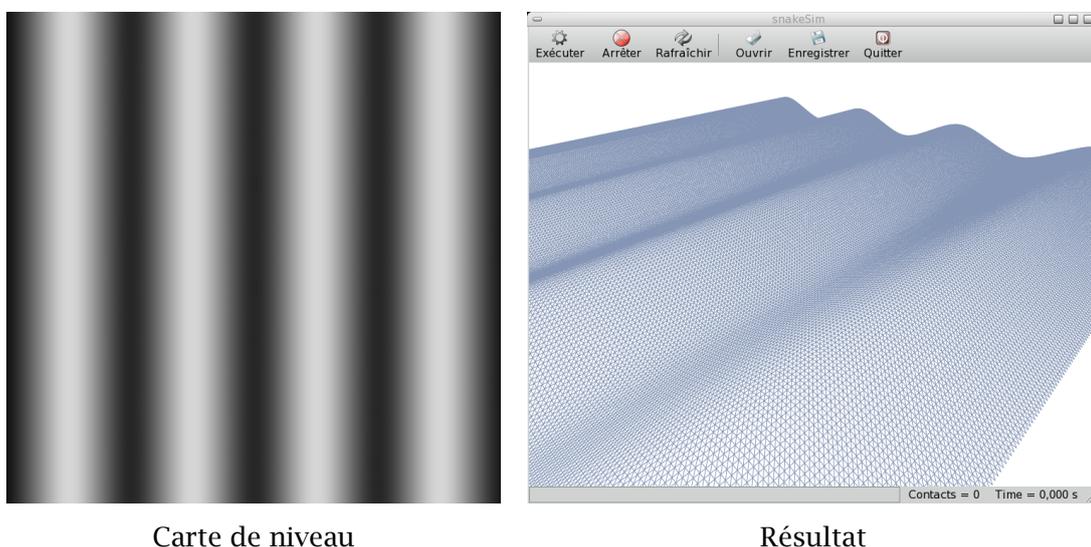
Notre environnement de simulation dispose de la capacité de modéliser le sol par des cartes de niveaux sous la forme d'images en niveaux de gris de résolution 256\*256. La hauteur de référence et l'amplitude correspondant à la différence de niveau entre une hauteur définie par un niveau de gris à 0 et celle définie par un niveau de gris à 255 sont paramétrables par l'intermédiaire de fichiers de configuration. Par conséquent, une carte de niveau peut être utilisée sans avoir recours à la recompilation du programme. L'intérêt d'utiliser une image pour la génération de sol 3D est que l'on peut la modifier très simplement avec n'importe quel utilitaire de retouche d'images tel que [GIMP, 2006] par exemple. De plus, un logiciel tel que [TERRAFORM, 2006] peut être utilisé pour la génération de terrain. Les figures 3.12 et 3.13 présentent des exemples de sols modélisés par cartes de niveaux.



Carte de niveau

Résultat

FIGURE 3.12. Exemple de sol généré par Terraform



Carte de niveau

Résultat

FIGURE 3.13. Exemple de sol créé avec GIMP

### 3.6 Conclusion

Le fait d'avoir choisi d'utiliser des cycles de développement courts a permis de créer un outil qui a su évoluer à partir des besoins. À chaque étape, des fonctionnalités ont été ajoutées en réponse à l'avancement des travaux de recherche. Cette approche incrémentale nous a permis d'obtenir un environnement complet de simulation robotique, validé pas à pas, dont les principales évolutions ont été :

- le développement initial d'un simulateur intégrant ODE et une interface graphique utilisant gtkmm et OpenGL,
- le développement du modèle de simulation d'un module à 8 mécanismes de notre robot ainsi que la gestion des paramètres de simulation par l'intermédiaire de fichiers de configuration,
- le développement du modèle de simulation d'un robot à  $N$  modules à 8 mécanismes,
- le développement de la modélisation du sol par cartes de niveau,

- le développement de la sauvegarde et de la restitution de simulation basées sur l'utilisation de fichiers binaires,
- l'intégration de GSL pour la génération de trajectoire et développement de l'objet correspondant,
- le développement d'un objet pour la commande,
- le développement de la fonctionnalité d'exécution distante de l'application,
- la modification du modèle de simulation du robot afin de pouvoir choisir par l'intermédiaire d'un paramètre de configuration entre l'utilisation de modules à 8 mécanismes ou l'utilisation de modules à 4 mécanismes,
- l'intégration de EO et le développement de la gestion des simulations par l'apprentissage,
- le développement d'autres modèles de simulation (un bras manipulateur et un autre robot serpent).

# Chapitre 4

## Modélisation d'une première structure robotique bio-inspirée

---

### Sommaire

---

4.1	Le projet VIPeRe .....	51
4.1.1	Brevets .....	52
4.1.2	Prototype .....	53
4.2	Mécanisme de propulsion .....	53
4.2.1	Caractérisation et définition du mécanisme .....	53
4.2.1.1	Schéma cinématique .....	53
4.2.1.2	Graphe des liaisons .....	54
4.2.2	Simplification du mécanisme en un mécanisme plan .....	55
4.2.3	Degrés de liberté du mécanisme .....	55
4.2.4	Modélisation géométrique directe .....	56
4.2.5	Modélisation cinématique directe .....	58
4.3	Simulation d'un module à 8 mécanismes .....	60
4.3.1	Définition du module .....	60
4.3.2	Principe de locomotion d'un module à 8 mécanismes .....	61
4.3.3	Comportement local d'un mécanisme simulé .....	61
4.3.3.1	Résultats géométriques .....	61
4.3.3.2	Résultats cinématiques .....	61
4.3.4	Comportement du module simulé .....	66
4.3.4.1	Résultats géométriques .....	66
4.3.4.2	Résultats cinématiques .....	66
4.4	Conclusion .....	66

---

Ce chapitre présente dans un premier temps le robot VIPeRe conçu par TDA. Cette structure robotique est constituée de plusieurs modules articulés. Ce chapitre constitue l'étude d'un module isolé et de son mécanisme de locomotion. Ainsi, la définition et la modélisation analytique du mécanisme à barres utilisé pour la locomotion seront traitées afin de fournir des éléments de comparaison à confronter aux résultats obtenus lors de la simulation d'un module à 8 mécanismes tel que défini par la conception de TDA.

### 4.1 Le projet VIPeRe

Le projet VIPeRe est à la base des travaux de thèse présentés au sein de ce manuscrit. Nous introduirons donc ce chapitre par une synthèse des travaux menés par TDA.

### 4.1.1 Brevets

La volonté de TDA de concevoir et de réaliser un prototype de robot-serpent se traduit concrètement par le dépôt de deux demandes de brevet d'invention ainsi que la réalisation de deux prototypes de module.

La première de ces deux demandes s'intéresse plus particulièrement au principe d'ondulation latérale et s'intitule « Procédé de propulsion d'un véhicule à déplacement par reptation et véhicule pour sa mise en œuvre » [GUYVARCH, 1995]. Ce document décrit un « véhicule constitué par une série d'éléments articulés entre eux ». L'articulation entre deux éléments successifs permet de régler l'orientation de l'un par rapport à l'autre suivant l'axe vertical. Le principe de locomotion se résume à la reproduction de proche en proche des consignes de ces articulations. « C'est ainsi qu'à la manière d'un serpent long, sur une surface offrant une résistance au glissement, les éléments successifs du véhicule se déplacent tous en même temps, par translation l'un vers l'autre suivant un trajet sinueux, stable dans l'espace, qui forme des ondulations se départant latéralement de part et d'autre de la direction globale de déplacement du véhicule, conformément à la configuration sinueuse imposée depuis le segment de tête. Chaque élément prend la place de celui qui l'a précédé dans le temps, en restant dans le même sillage de la configuration sinueuse dans l'espace. » Cette solution propose l'utilisation de roue non motorisée.

La seconde demande est intitulée « Véhicule à déplacement par reptation propulsé conjointement par ondulations longitudinales et latérales » [GUYVARCH, 2001]. Elle développe le principe du renforcement de la propulsion par ondulation latérale par l'utilisation d'une propulsion longitudinale s'inspirant du mode de progression rectilinéaire des serpents. Un des objectifs de la collaboration de ces deux types de propulsion est la suppression de l'utilisation des roues, dans la forme préconisée par le brevet précédent. La finalité est l'extension des capacités du véhicule aux déplacements sur des sols irréguliers ou sur substrats spécifiques tels que l'herbe ou la boue.

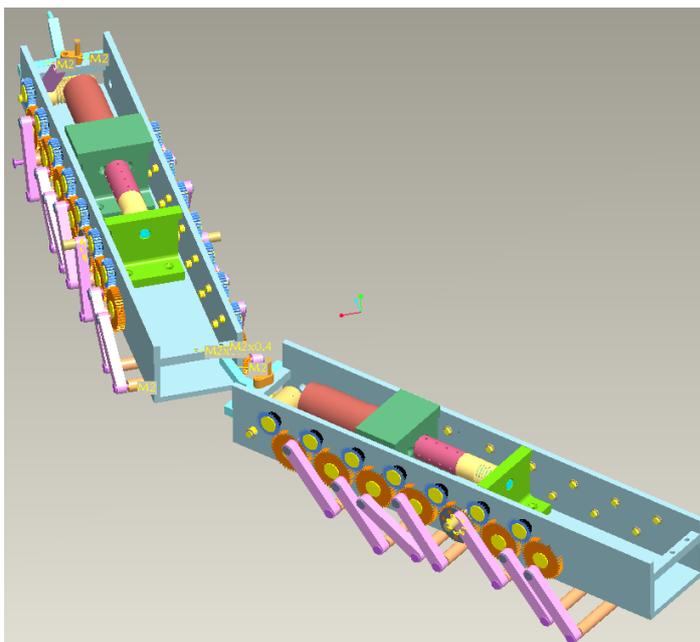


FIGURE 4.1. Étude CAO du prototype VIPeRe

### 4.1.2 Prototype

Les recherches de TDA dans le domaine de la robotique mobile ont finalement abouti à la conception de la structure robotique telle que représentée à la figure 4.1.

Cette structure robotique est basée sur l'utilisation de plusieurs modules à 8 mécanismes articulés entre eux par ce que l'on peut modéliser par une liaison pivot d'axe vertical. La propulsion de la structure est alors obtenue par les mécanismes présents sur chacun des modules tandis que le changement de direction est assuré par le pilotage des liaisons inter-modulaires. Le fruit de cette conception est la réalisation de deux modules sur lesquels nous ne disposons que de peu d'informations. La liaison entre ces deux modules n'a pas été étudiée en détail. Le modèle CAO de cette structure robotique constitue l'unique point de départ des travaux de thèse présentés tout au long de ce manuscrit.

## 4.2 Mécanisme de propulsion

La mobilité du robot repose pour une grande partie sur les capacités locomotrices de ce mécanisme. Le mécanisme en lui-même n'est pas original car les mécanismes à barres présentant une boucle cinématique fermée sont assez bien connus et bien traités dans la littérature [ARTOBOLEVSKI, 1977, KOLOVSKY *et al.*, 2000 et KHALIL et DOMBRE, 1999]. Néanmoins, l'originalité de l'étude réside dans l'utilisation de l'association de plusieurs mécanismes à cinq barres en tant qu'organes de locomotion d'un robot.

Notre étude reprend le mécanisme original du prototype VIPeRe. Ainsi, nous n'étudierons pas les optimisations potentielles résultant des paramétrages des longueurs des éléments constitutifs du mécanisme. Par contre, nous nous intéresserons au paramètre de déphasage angulaire du mécanisme bien qu'il ne soit pas réglable sur le prototype VIPeRe. En effet, le profil de la trajectoire de l'organe terminal d'un mécanisme dépend de la valeur de ce paramètre angulaire comme nous le démontrerons lors de l'élaboration du modèle géométrique.

### 4.2.1 Caractérisation et définition du mécanisme

L'étude du mécanisme présentée au sein de ce chapitre est basée sur l'analyse du modèle CAO VIPeRe ainsi que sur les travaux présentés dans [AUBIN *et al.*, 2006].

Le mécanisme étudié est un mécanisme 3D présentant une boucle cinématique fermée. Cette boucle cinématique fermée et un choix pertinent des coordonnées généralisées permettent d'obtenir une trajectoire en bout de mécanisme exploitable du point de vue de la locomotion comme nous nous proposons de le démontrer ici.

#### 4.2.1.1 Schéma cinématique

La figure 4.2 présente le schéma cinématique 3D d'un mécanisme. Ce dernier est composé de 5 solides et relié au corps principal d'un module que l'on assimilera au bâti pour l'étude locale du mécanisme. Ce schéma ne représente pas l'interaction éventuelle de l'organe terminal du mécanisme avec l'environnement (principalement le sol, du point de vue de la locomotion).

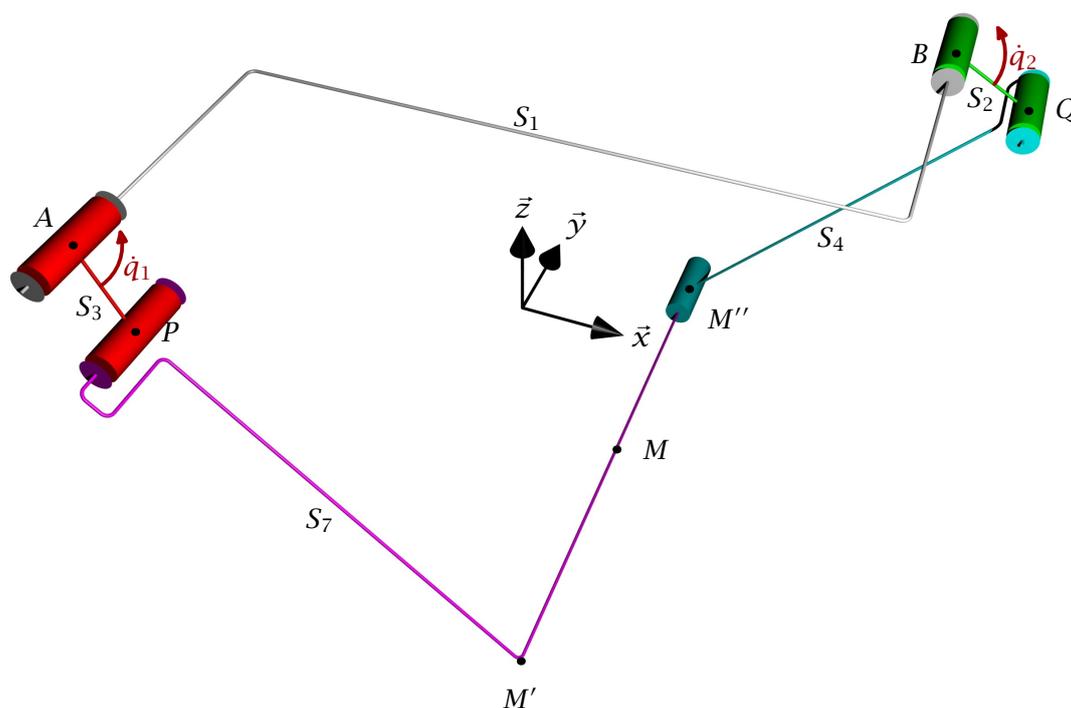


FIGURE 4.2. Schéma cinématique du mécanisme

Les liaisons reliant les différents éléments du mécanisme sont :

- 4 liaisons pivot (1 degré de liberté en rotation), soit 4 couples cinématiques de classe V, localisées aux points A, B, P et Q ;
- 1 liaison pivot glissant (2 degrés de liberté dont 1 en rotation et 1 en translation), soit 1 couple cinématique de classe IV, localisée au point M'' ;
- 1 liaison encastrement (pas de degré de liberté) localisée au point M'.

Il est important de préciser que la validité de cette modélisation cinématique repose sur l'hypothèse selon laquelle les axes des liaisons sont parfaitement parallèles.

Du point de vue de la cinématique et en considérant un sol plan, l'interaction normale entre le mécanisme et le sol est une liaison linéaire rectiligne (4 degrés de liberté dont 2 en rotation et 2 en translation) soit un couple cinématique de classe II.

#### 4.2.1.2 Graphe des liaisons

Le graphe des liaisons (figure 4.3) reprend cette structure cinématique en respectant les notations suivantes pour les solides ainsi que les liaisons constituant un mécanisme :

$S_0$	sol	$L_{12}$	liaison pivot
$S_1$	corps principal du module (bâti)	$L_{13}$	liaison pivot
$S_2$	roue gauche	$L_{24}$	liaison pivot
$S_3$	roue droite	$L_{35}$	liaison pivot
$S_4$	bielle gauche	$L_{46}$	liaison pivot glissant
$S_5$	bielle droite	$L_{56}$	liaison encastrement
$S_6$	traverse	$L_{60}$	liaison linéaire rectiligne (dépend de la phase de locomotion)

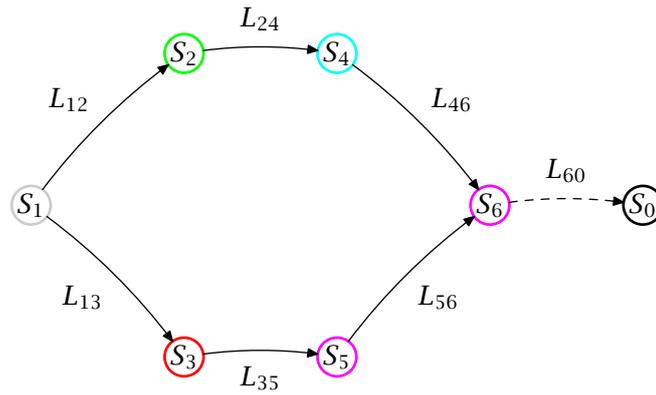


FIGURE 4.3. Graphe des liaisons du mécanisme

La liaison  $L_{60}$  représente l'interaction du mécanisme avec l'environnement (sol ou obstacle) lors de la mise en situation du mécanisme au sein de la structure robotique complète. Cette liaison étant intermittente, elle n'est pas prise en compte dans le cadre de l'étude locale d'un mécanisme.

La liaison  $L_{56}$  étant une liaison encastrement, on peut regrouper  $S_5$  et  $S_6$  au sein d'une classe cinématique et les assimiler à un unique solide  $S_7$ . Le résultat de cette simplification est représenté par la figure 4.4.

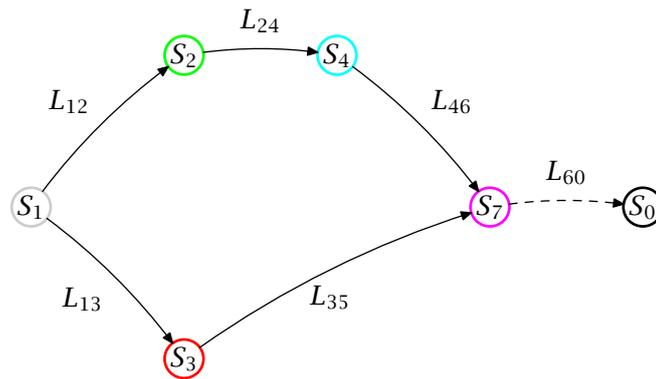


FIGURE 4.4. Graphe des liaisons du mécanisme simplifié

## 4.2.2 Simplification du mécanisme en un mécanisme plan

Les axes des liaisons du mécanisme sont parallèles et les points  $M$ ,  $M'$  et  $M''$  sont alignés suivant l'axe  $\vec{y}$  du repère attaché au corps principal. Par conséquent, le mécanisme de locomotion peut être ramené et étudié dans le plan  $(M, \vec{x}, \vec{z})$ . La figure 4.5 correspond à la vue de côté du mécanisme tridimensionnel. On ramène alors le problème à l'étude dans le plan d'un mécanisme à 5 barres ou pentagone articulé.

## 4.2.3 Degrés de liberté du mécanisme

D'une manière générale, le calcul des degrés de liberté d'un mécanisme tridimensionnel n'est pas trivial. Une solution générale consiste à utiliser le rang de la jacobienne pour obtenir le nombre de contraintes géométriques indépendantes.

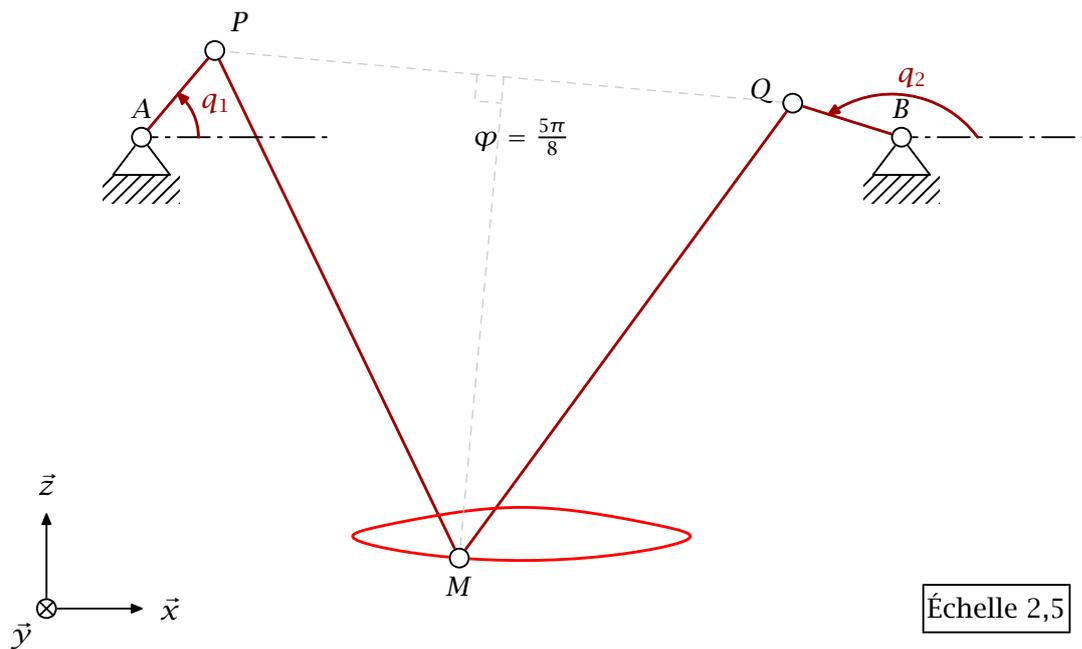


FIGURE 4.5. Vue de côté d'un mécanisme

Dans le cas présent, nous utiliserons la simplification plane du mécanisme ainsi que la formule de TCHEBYCHEV [ARTOBOLEVSKI, 1977] pour connaître le nombre de degrés de liberté du mécanisme. Le mécanisme étant composé de 4 solides mobiles et de 5 liaisons de classe V, on a

$$w = 3 * 4 - 2 * 5 = 2.$$

Le mécanisme possède 2 degrés de liberté. On dispose donc de 2 coordonnées généralisées pour la commande de ce mécanisme. Le choix des coordonnées généralisées correspond aux coordonnées  $q_1$  et  $q_2$ . Si ces coordonnées généralisées sont liées par une relation, le degré de liberté est de 1 et le point  $M$  décrit une courbe, telle que représentée par la figure 4.5.

#### 4.2.4 Modélisation géométrique directe

Nous avons besoin de connaître la trajectoire de l'organe terminal du mécanisme pour relativiser sa pertinence du point de vue de la locomotion. Pour ce faire, le modèle géométrique direct du mécanisme peut être obtenu simplement à partir de l'équation suivante :

$$\overrightarrow{AM} = \overrightarrow{AP} + \overrightarrow{PM}. \quad (4.1)$$

Nous utilisons les notations suivantes pour les constantes géométriques caractéristiques de ce mécanisme :

$$\|\overrightarrow{AP}\| = \|\overrightarrow{BQ}\| = R, \quad (4.2)$$

$$\|\overrightarrow{PM'}\| = \|\overrightarrow{QM''}\| = \rho, \quad (4.3)$$

$$\overrightarrow{AB} \cdot \vec{x} = d. \quad (4.4)$$

Ces égalités reprennent les notations de la figure 4.2. Les valeurs utilisées tout au long de notre étude sont celles du mécanisme original et sont récapitulées dans le tableau 4.1.

TABLEAU 4.1. Valeurs des constantes géométriques du mécanisme de locomotion (issues du modèle CAO)

notation	valeur
$\rho$	6 mm
$R$	29,8 mm
$d$	40 mm

On suppose que les conditions  $2(R - \rho) > d$  et  $\rho < \frac{d}{2}$  sont vérifiées.

En posant,  $q_1$  et  $q_2$  les paramètres d'entrée du mécanisme. L'égalité (4.2) implique

$$\overrightarrow{AP} = R \cos(q_1) \cdot \vec{x} + R \sin(q_1) \cdot \vec{z} \quad (4.5)$$

et

$$\overrightarrow{BQ} = R \cos(q_2) \cdot \vec{x} + R \sin(q_2) \cdot \vec{z}. \quad (4.6)$$

La conception initiale réduit le nombre de coordonnées généralisées en imposant un déphasage angulaire de valeur constante de  $\frac{5\pi}{8}$  entre  $q_1$  et  $q_2$ . Nous faisons le choix de traiter ce déphasage non plus comme une constante fixée et non réglable mais comme un paramètre réglable. Soit  $\varphi$ , le déphasage angulaire, constant pour une phase de locomotion, tel que  $q_2 = q + \varphi$  et  $q_1 = q$ , les équations liant les coordonnées généralisées du problème. L'obtention du modèle géométrique direct revient à déterminer analytiquement  $\overrightarrow{AM}(q, \varphi)$ . En utilisant (4.5), on peut écrire

$$\overrightarrow{AM}(q, \varphi) = \overrightarrow{AP}(q) + \overrightarrow{PM}(q, \varphi) \quad (4.7)$$

où  $\overrightarrow{PM}(q, \varphi)$  reste à déterminer.

L'équation (4.3) implique que le triangle  $PMQ$  est un triangle isocèle. La bissectrice de l'angle  $(\overrightarrow{PM}; \overrightarrow{MQ})$  est alors confondue avec la hauteur du triangle  $PMQ$  issue du sommet  $M$ . On note  $N$ , le pied de la hauteur issue de  $M$  et on écrit alors

$$\overrightarrow{PM} = \overrightarrow{PN} + \overrightarrow{NM}.$$

$\overrightarrow{PN}$  s'exprime facilement en fonction de  $\overrightarrow{PQ}$  en écrivant

$$\overrightarrow{PN} = \frac{\overrightarrow{PQ}}{2} = \frac{\overrightarrow{PA} + \overrightarrow{AB} + \overrightarrow{BQ}}{2}.$$

De plus, la norme et la direction de  $\overrightarrow{NM}$  sont obtenues respectivement en appliquant le théorème de Pythagore au triangle rectangle  $PNM$  et en utilisant le vecteur unitaire perpendiculaire à  $\overrightarrow{PQ}$  (orienté selon  $-\vec{z}$ ) :

$$\overrightarrow{NM} = \frac{\sqrt{PM^2 - PN^2}}{\|\overrightarrow{PN}\|} \cdot \begin{pmatrix} \overrightarrow{PN} \cdot \vec{z} \\ -\overrightarrow{PN} \cdot \vec{x} \end{pmatrix}. \quad (4.8)$$

L'équation (4.7) devient alors

$$\overrightarrow{AM}(q, \varphi) = \overrightarrow{AP}(q) + \overrightarrow{PN}(q, \varphi) + \overrightarrow{NM}(q, \varphi) \quad (4.9)$$

En projetant (4.9) sur  $\vec{x}$  et  $\vec{z}$  et en posant

$$\begin{aligned} S_- &= \sin(q + \varphi) - \sin(q), \\ C_- &= \cos(q + \varphi) - \cos(q), \\ S_+ &= \sin(q + \varphi) + \sin(q), \\ C_+ &= \cos(q + \varphi) + \cos(q). \end{aligned}$$

On obtient finalement l'équation paramétrique de la trajectoire

$$\begin{aligned} \overrightarrow{AM} \cdot \vec{x} &= \frac{RC_+ + d}{2} + \frac{RS_-}{2} \sqrt{\frac{4\rho^2}{(RC_- + d)^2 + R^2S_-^2} - 1}, \\ \overrightarrow{AM} \cdot \vec{z} &= \frac{RS_+}{2} - \frac{RC_- + d}{2} \sqrt{\frac{4\rho^2}{(RC_- + d)^2 + R^2S_-^2} - 1}. \end{aligned}$$

La connaissance du modèle géométrique direct permet d'obtenir analytiquement les trajectoires de l'organe terminal d'un mécanisme en fonction du déphasage angulaire  $\varphi$ . La figure 4.6 représente ces trajectoires pour plusieurs valeurs de déphasage.

Nous remarquons que pour différentes valeurs de  $\varphi$  on trouve des profils de trajectoire très différents. Par ailleurs, pour un sens de rotation fixé de l'entrée  $q$  le sens de parcours de la trajectoire n'est pas toujours le même.

#### 4.2.5 Modélisation cinématique directe

L'obtention du modèle cinématique direct présenté ici se fait par dérivation du modèle géométrique direct. On obtient alors les projections suivant les axes  $\vec{x}$  et  $\vec{z}$  de la vitesse de la traverse par rapport au repère attaché au corps principal du module. La dérivation de l'équation paramétrique est dépendante de l'expression temporelle applicable à  $q$ . Les résultats présentés ici sont basés sur les hypothèses suivantes : on se trouve en régime permanent et  $q = \omega t$ , avec  $\omega$ , une vitesse de rotation constante.

On pose

$$V = R\omega, \quad L_1 = \sqrt{4\rho^2 - (RC_- + d)^2 - R^2S_-^2} \quad \text{et} \quad L_2 = \sqrt{(RC_- + d)^2 + R^2S_-^2}.$$

Après dérivation de l'équation paramétrique, on obtient les expressions suivantes, projections de la vitesse :

$$\frac{d}{dt} (\overrightarrow{AM} \cdot \vec{x}) = V_{x_1} + V_{x_2} + V_{x_3} + V_{x_4} \quad \text{et} \quad \frac{d}{dt} (\overrightarrow{AM} \cdot \vec{z}) = V_{z_1} + V_{z_2} + V_{z_3} + V_{z_4}$$

avec

$$V_{x_1} = V \frac{RS_-^2 d}{2L_1 L_2}, \quad V_{x_2} = V \frac{C_- L_1}{2L_2}, \quad V_{x_3} = V \frac{RS_-^2 d L_1}{2L_2^3}, \quad V_{x_4} = -V \frac{S_+}{2}$$

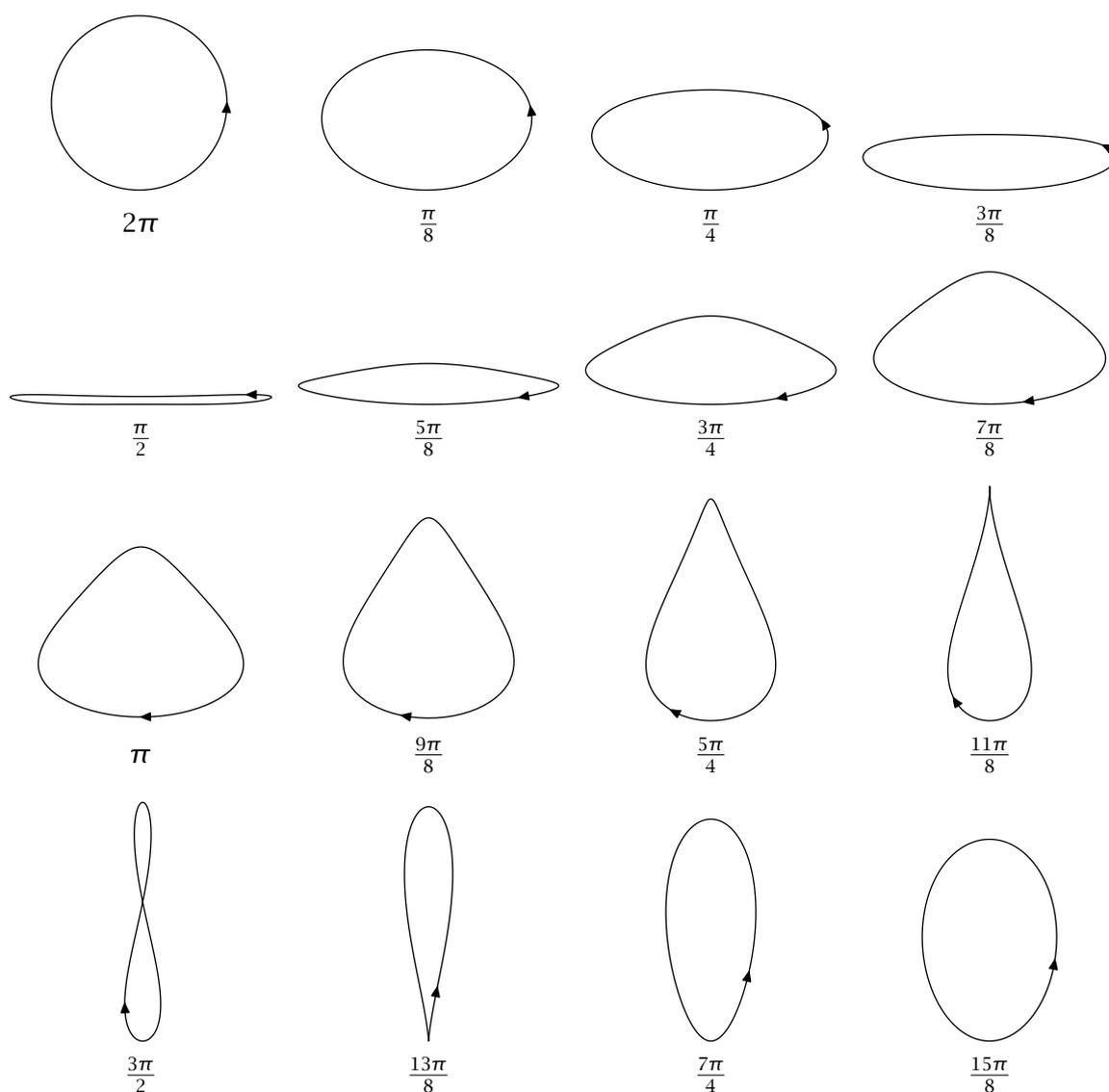


FIGURE 4.6. Trajectoire et sens de parcours du point  $M$  pour différentes valeurs de  $\varphi$

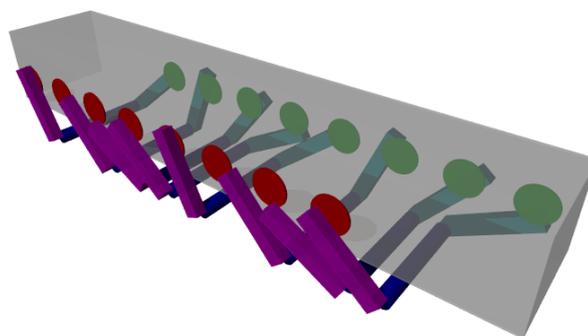


FIGURE 4.7. Rendu réaliste d'un module à 8 mécanismes

et

$$V_{z_1} = V \frac{S-L_1}{2L_2}, \quad V_{z_2} = -V \frac{S-d(RC_- + d)}{2L_1L_2}, \quad V_{z_3} = -V \frac{S-d(RC_- + d)L_1}{2L_2^3}, \quad V_{z_4} = V \frac{C_+}{2}.$$

Les profils de vitesse obtenus seront présentés dans la section 4.3.3 avec la vitesse globale du robot.

### 4.3 Simulation d'un module à 8 mécanismes

Le mécanisme à 5 barres que nous venons de présenter peut être utilisé pour la locomotion robotique. Ceci n'est toutefois possible qu'en considérant une association de plusieurs de ces mécanismes. Afin d'expliquer le fonctionnement de la propulsion d'un module, nous avons choisi de présenter une étude de cas basée sur la simulation d'un module à 8 mécanismes. À cette étape de l'étude, un robot composé d'un unique module n'est pas capable d'effectuer des changements de direction. Néanmoins, cette étape est nécessaire pour évaluer notre environnement de simulation en le confrontant avec l'étude analytique du mécanisme.

#### 4.3.1 Définition du module

Le mécanisme présenté plus haut est une interprétation, au niveau élémentaire, du mode de locomotion rectilinéaire présenté à la section 2.2.2. Ainsi, comme cela est le cas chez les espèces de serpents terrestres pratiquant ce type de locomotion, la locomotion résulte de l'association de plusieurs de ces mécanismes locomoteurs élémentaires. Nous présentons ici une association possible de plusieurs de ces mécanismes locomoteurs.

Soient 8 mécanismes locomoteurs répartis parallèlement le long du corps principal du module. Les mécanismes sont distants, de proche en proche suivant l'axe longitudinal du corps principal, d'une longueur égale à  $\frac{d}{2}$ ,  $d$  étant la distance entre les deux points d'ancrage d'un même mécanisme sur le module. De plus, en posant  $q_i$ , la coordonnée généralisée du  $i$ ème mécanisme, on définit l'état angulaire du  $i$ ème mécanisme par  $q_i = \omega t + \frac{i\pi}{2}$ . Le résultat d'une telle association de 8 mécanismes est illustré par la vue en perspective de la figure 4.7 et par la vue de côté de la figure 4.8.

Le corps principal est dimensionné de façon à ce que les organes terminaux des mécanismes n'entrent jamais en contact avec celui-ci. De plus, sa longueur est fonction du nombre de mécanismes utilisés.

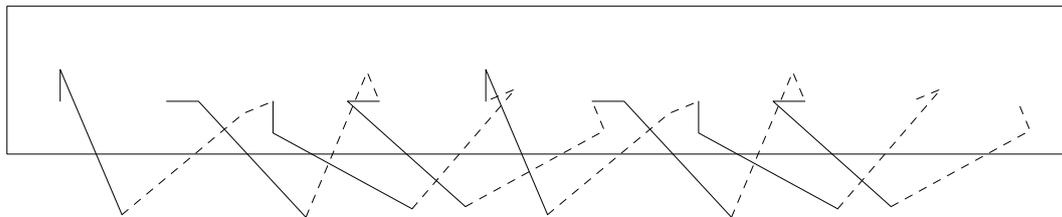


FIGURE 4.8. Vue de côté d'un module à 8 mécanismes

En considérant un sol plan et relativement à la condition  $q_i = \omega t + \frac{i\pi}{2}$ , le module résultant sera toujours en contact avec le sol par l'intermédiaire d'au moins deux mécanismes locomoteurs. Par conséquent, un module constitué de huit mécanismes peut être considéré comme stable puisqu'en appui plan sur le sol.

Chaque mécanisme est paramétré avec un déphasage constant  $\varphi = \frac{5\pi}{8}$ .

### 4.3.2 Principe de locomotion d'un module à 8 mécanismes

La figure 4.9 représente un quart de la période complète de locomotion d'un module à 8 mécanismes, c'est-à-dire la période durant laquelle un couple de mécanismes sera en contact avec le sol. Le couple de mécanismes concerné est constitué des 3<sup>e</sup> et 7<sup>e</sup> mécanismes en partant de la gauche. La figure 4.9(a) correspond à l'instant où le couple de mécanismes arrive au contact du sol. La figure 4.9(b) correspond, elle, à la phase où le couple de mécanismes est en appui au sol, et par conséquent, est locomoteur. Enfin, la figure 4.9(c) correspond à l'instant où le couple de mécanismes quitte le contact. Durant ces phases de locomotion, la structure se déplace de la gauche vers la droite pour un sens de rotation trigonométrique des bras d'entrée de mécanismes.

### 4.3.3 Comportement local d'un mécanisme simulé

On souhaite vérifier que le comportement d'un mécanisme simulé est correct par rapport aux résultats issus des modèles géométrique et cinématique d'un mécanisme.

#### 4.3.3.1 Résultats géométriques

La première vérification consiste à évaluer le respect de la géométrie du mécanisme simulé. La trajectoire simulée peut être obtenue très simplement relativement à la position de l'organe terminal du mécanisme étudié à la position du corps principal du module. En effet, au sein de l'environnement de simulation, les positions et orientations de chacun des solides étant connues, on obtient finalement les résultats présentés par la figure 4.10.

La trajectoire présentée est obtenue avec un pas de simulation de 0,5 ms sur approximativement un cycle de régime permanent d'une durée de 2 s. On constate que la géométrie est bien respectée à une perturbation près. Cette perturbation est induite par la gestion des articulations au sein du moteur de simulation.

#### 4.3.3.2 Résultats cinématiques

De la même manière que les positions et orientations sont connues, les vitesses le sont également. Une composition des vitesses nous permet alors de déterminer les projections de la vitesse linéaire relative de l'organe terminal du mécanisme par rapport au corps principal du module. Le modèle cinématique du mécanisme permet d'obtenir analytiquement ces projections. Les figures 4.11 et 4.12 représentent les résultats en terme de vitesse. Les résultats présentés sont obtenus pour une vitesse de rotation  $\omega = \pi$  rad/s.

Les perturbations observées sont dues aux mêmes phénomènes que ceux évoqués dans la section précédente.

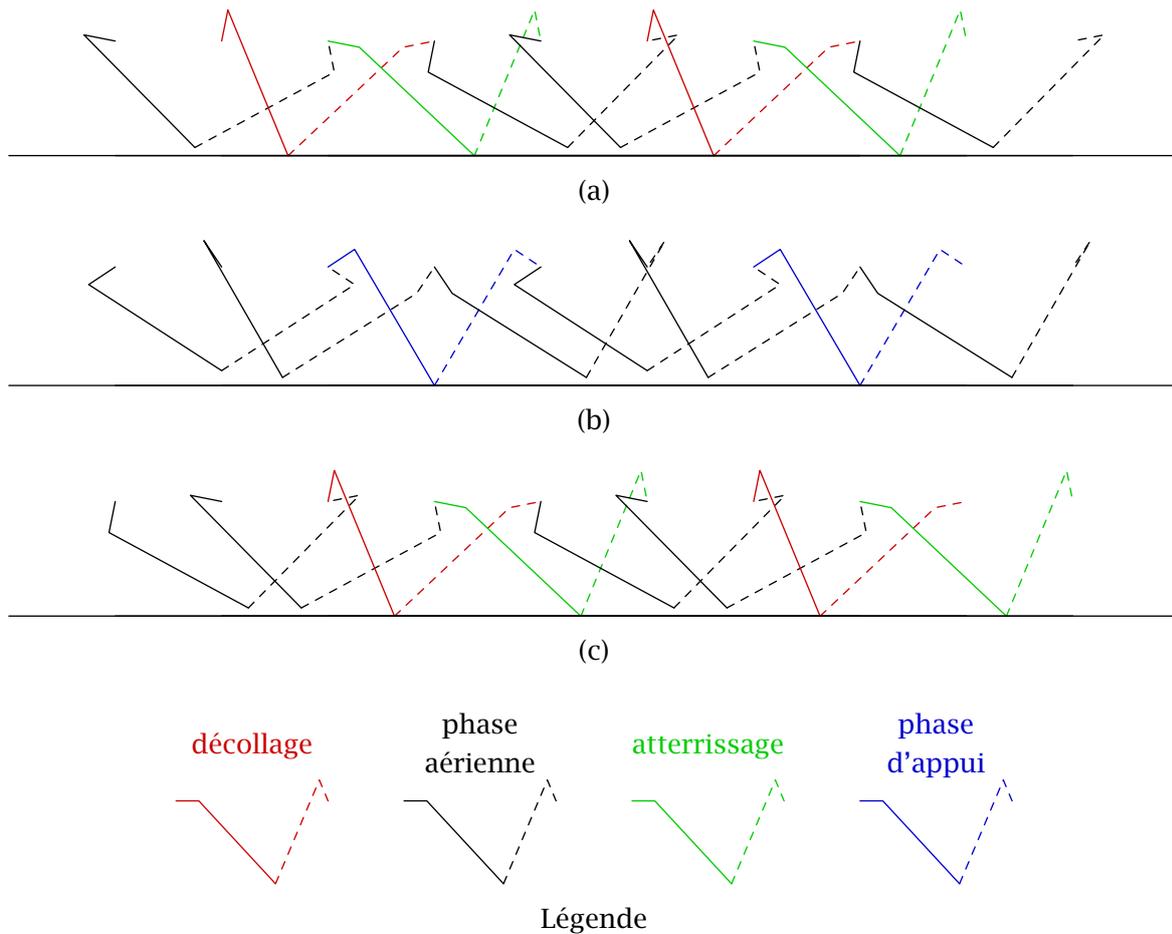


FIGURE 4.9. Locomotion d'un module à 8 mécanismes

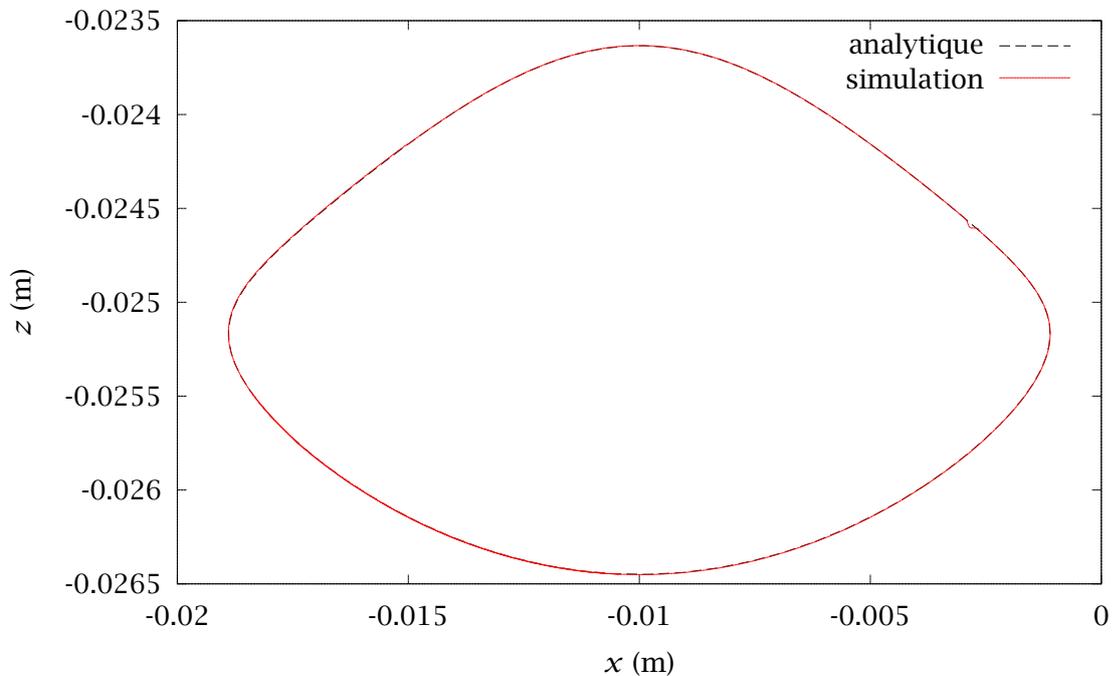


FIGURE 4.10. Comparaison de la trajectoire simulée avec la trajectoire obtenue à l'aide du MGD ( $\varphi = \frac{5\pi}{8}$ )

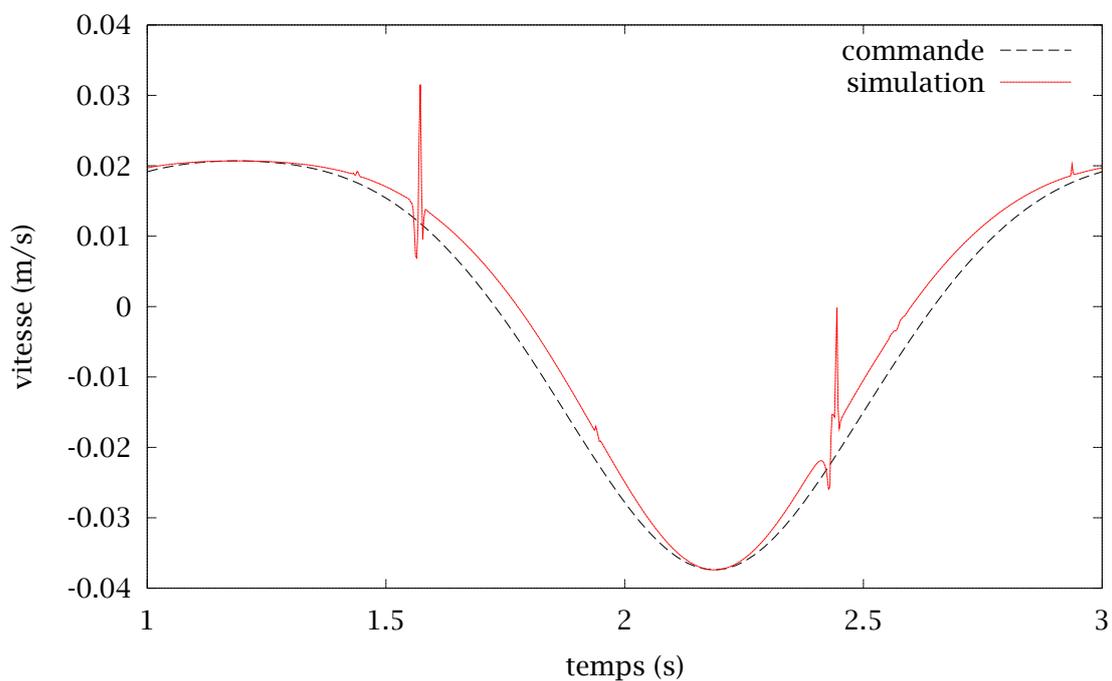


FIGURE 4.11. Projection suivant l'axe  $\vec{x}$  de la vitesse linéaire de l'organe terminal par rapport au corps principal du module

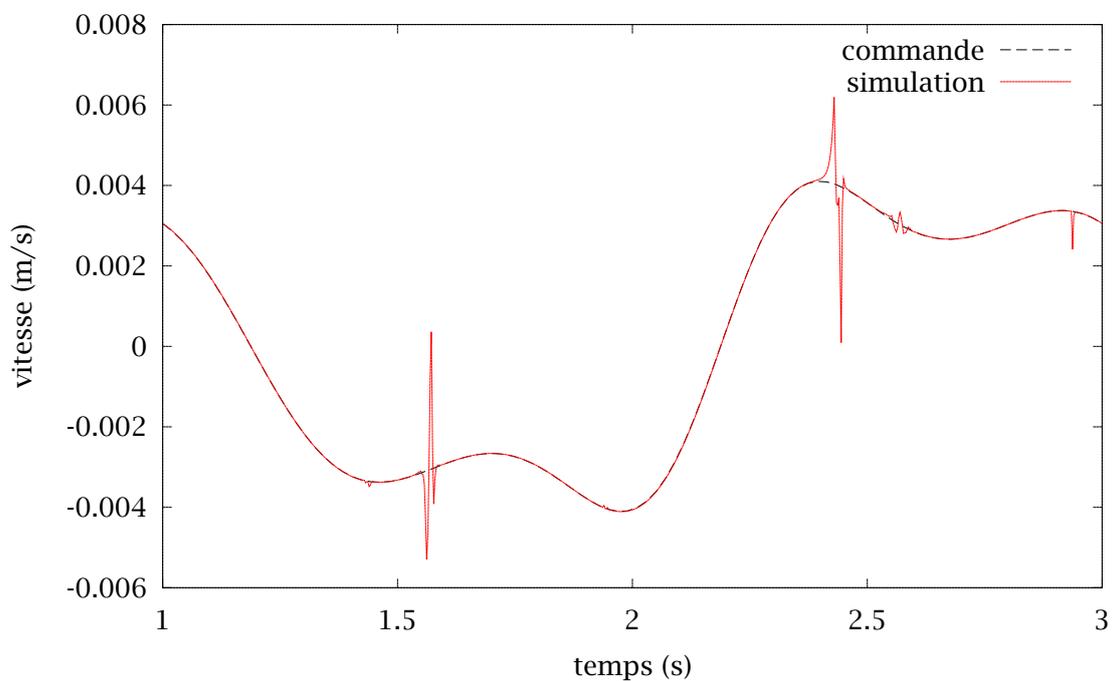


FIGURE 4.12. Projection suivant l'axe  $\vec{z}$  de la vitesse linéaire de l'organe terminal par rapport au corps principal du module

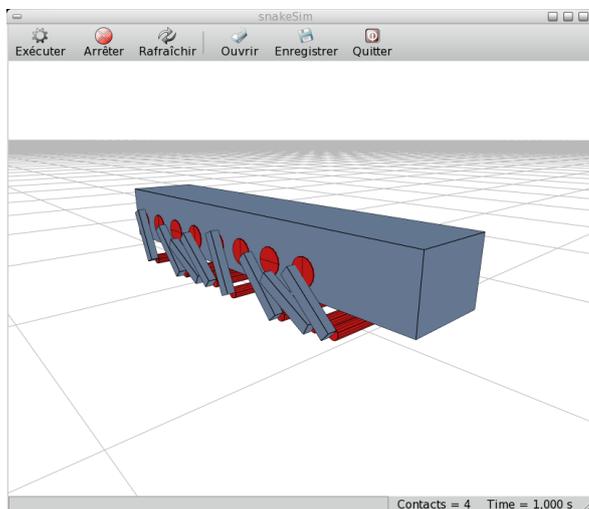


FIGURE 4.13. Capture de la simulation d'un module à 8 mécanismes

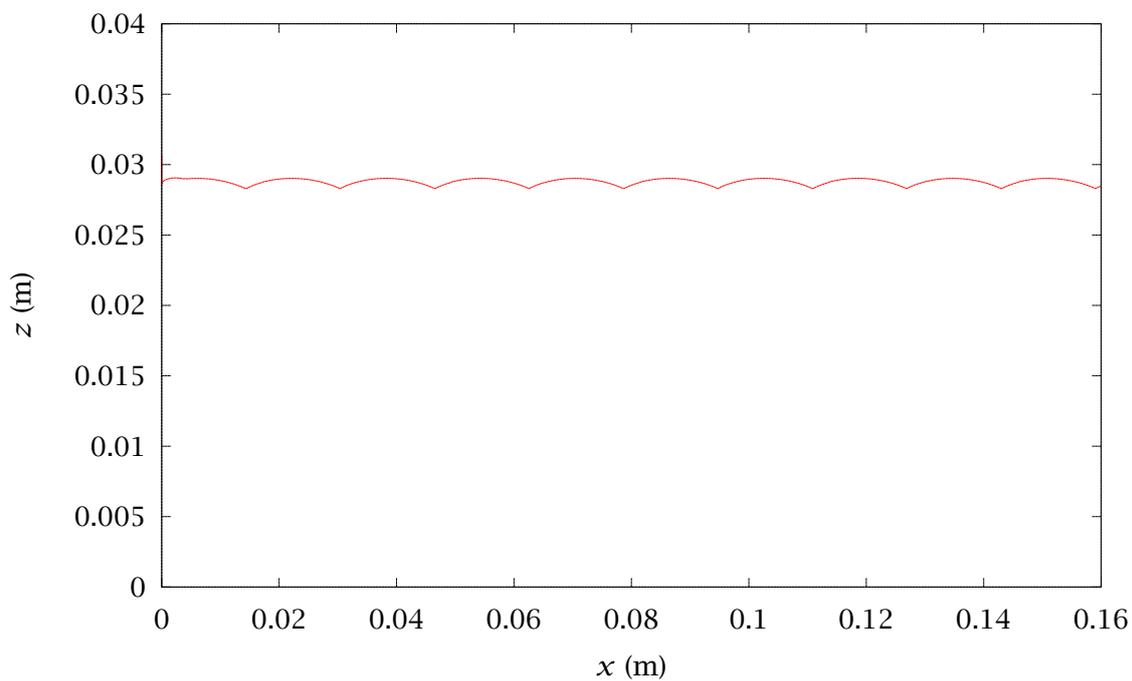


FIGURE 4.14. Projection dans le plan  $(\vec{x}, \vec{z})$  de la trajectoire du centre géométrique du corps principal du module

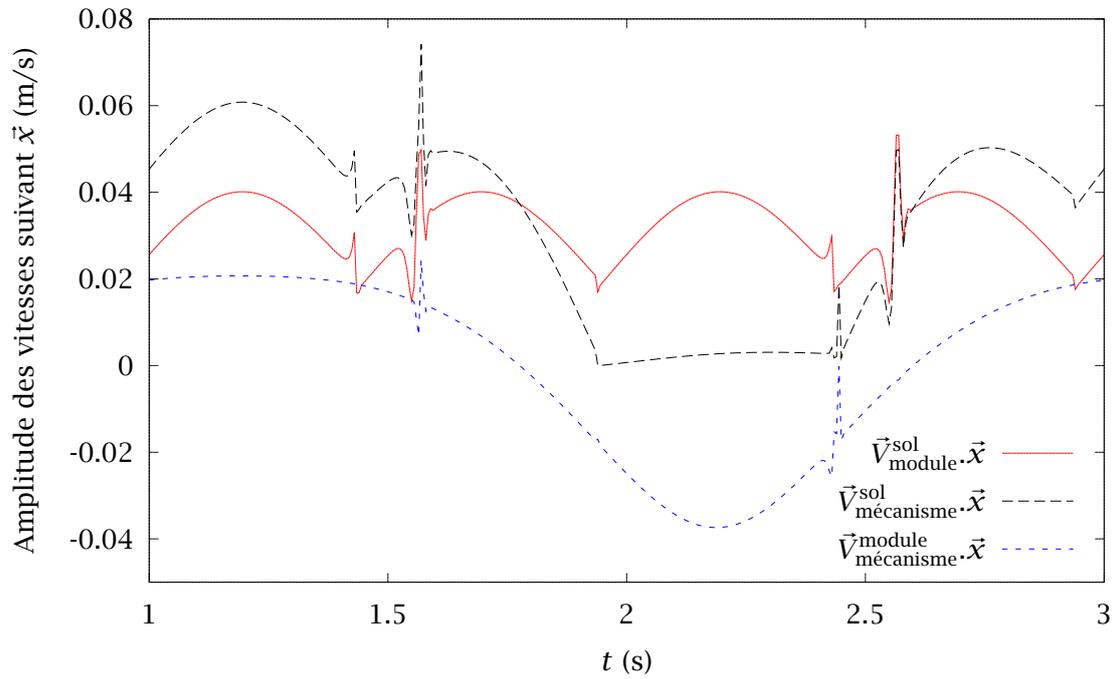


FIGURE 4.15. Projections suivant l'axe  $\vec{x}$  de la vitesse absolue d'un module et de la vitesse absolue de l'organe terminal d'un mécanisme

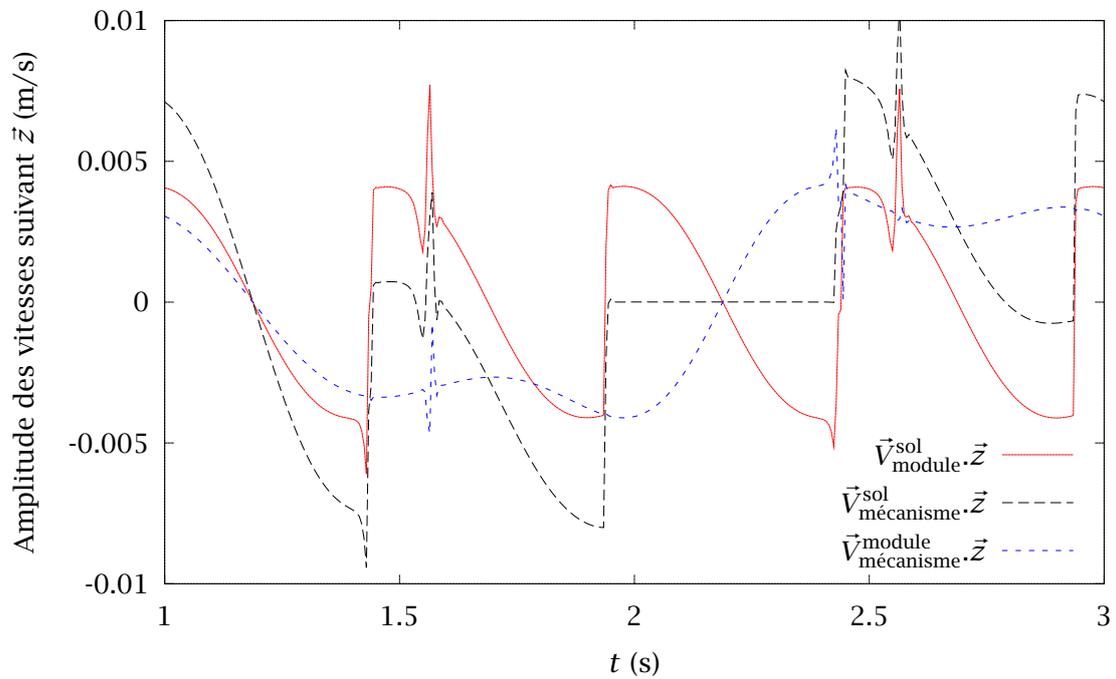


FIGURE 4.16. Projections suivant l'axe  $\vec{z}$  de la vitesse absolue d'un module et de la vitesse absolue de l'organe terminal d'un mécanisme

#### 4.3.4 Comportement du module simulé

Les résultats de simulation étant validés à l'échelle du mécanisme, nous allons nous intéresser aux résultats de simulation du mouvement d'un module à 8 mécanismes. La figure 4.13 représente une capture de la simulation d'un tel module.

##### 4.3.4.1 Résultats géométriques

L'étude portant sur un unique module, les mouvements latéraux de ce dernier seront négligeables. L'étude du mouvement absolu du module se résume donc à :

- la vérification du caractère négligeable des mouvements latéraux,
- la cohérence des mouvements verticaux et longitudinaux par rapport aux mouvements des mécanismes de propulsion.

Nous ne représenterons pas ici les mouvements latéraux bien que nous ayons pris soin de vérifier qu'ils étaient effectivement négligeables.

La figure 4.14 représente la projection dans le plan  $(\vec{x}, \vec{z})$  de la trajectoire du centre géométrique du corps principal du module étudié. Les axes de cette figure ne sont pas normés et on remarque que la variation d'altitude pour la valeur de déphasage utilisée est de l'ordre de 2 mm. La forme d'onde de cette projection est directement liée à la trajectoire des organes terminaux des mécanismes lorsque ceux-ci sont en contact avec le sol. Le lien de cause à effet entre la trajectoire d'un mécanisme et la trajectoire d'un module est traité dans la suite de ce manuscrit.

##### 4.3.4.2 Résultats cinématiques

Les figures 4.15 et 4.16 représentent les projections suivants les axes  $\vec{x}$  et  $\vec{z}$  des vitesses :

- du module par rapport au sol,
- de l'un des mécanismes par rapport au sol,
- de ce même mécanisme par rapport au module.

Le mécanisme contribue à la locomotion du module lors de sa phase de contact. Cette phase de contact démarre autour de  $t = 1,93$  s pour se terminer autour de  $t = 2,43$  s. Durant cet intervalle de temps, la composition des vitesses est :

$$\vec{V}_{\text{module}}^{\text{sol}} = \vec{V}_{\text{mécanisme}}^{\text{sol}} - \vec{V}_{\text{mécanisme}}^{\text{module}}$$

## 4.4 Conclusion

Dans ce chapitre, après avoir défini la géométrie des mécanismes locomoteurs ainsi que celle d'un module, nous avons vérifié la faisabilité de la simulation d'un module à 8 mécanismes pour une valeur de déphasage de chaque mécanisme fixée à  $\frac{5\pi}{8}$ . Cette vérification repose sur la comparaison entre les résultats obtenus lors de la simulation avec ceux issus des modèles géométriques et cinématiques d'un mécanisme.

Cette simulation d'un module TDA fournit un outil de référence pour l'étude des possibilités d'optimisation du mécanisme ainsi que pour l'étude de la locomotion d'un robot constitué de plusieurs modules articulés.

# Chapitre 5

## Proposition d'une nouvelle structure robotique

---

### Sommaire

---

5.1	Optimisation et performances .....	67
5.1.1	Critères d'optimisation .....	68
5.1.2	Variation d'altitude de la plate-forme .....	68
5.1.3	Amplitude verticale .....	69
5.1.4	Performance locale de progression dans l'axe longitudinal .....	71
5.1.5	Exemples de critères croisés .....	71
5.1.6	Application à un module à 8 mécanismes .....	72
5.2	Proposition d'une nouvelle structure robotique .....	72
5.2.1	Motivations .....	72
5.2.2	Module .....	76
5.2.3	Articulations inter-modulaire .....	77
5.2.4	Conclusion .....	77

---

Ce chapitre présente dans un premier temps les travaux effectués concernant les optimisations possibles du mécanisme locomoteur ainsi qu'un exemple d'optimisation d'une tâche de locomotion appliqué à la simulation d'un module à 8 mécanismes. Dans un second temps, nous détaillerons la proposition d'une nouvelle structure robotique dérivée du prototype initial.

### 5.1 Optimisation et performances

Comme on peut le constater en étudiant la figure 4.6, les profils de trajectoire de l'organe terminal d'un mécanisme sont très différents en fonction de la valeur du déphasage angulaire  $\varphi$ . Comme nous l'avons établi dans le chapitre précédent pour une valeur donnée du déphasage, le mouvement global d'un module est conditionné par la géométrie des mécanismes locomoteurs. Par conséquent, le déplacement du centre géométrique du module sera le reflet du déplacement de l'organe terminal du mécanisme en contact avec le sol, s'il n'y a pas glissement. On établit ainsi le lien entre la phase de contact représentée à la figure 5.1 et le résultat obtenu à la figure 4.14.

Il est donc nécessaire de disposer d'informations permettant de régler la valeur du déphasage  $\varphi$  relativement à une tâche de locomotion.

### 5.1.1 Critères d'optimisation

Les optimisations potentielles étudiées ici portent sur la capacité du robot à franchir, du point de vue élémentaire, de petits obstacles locaux tout en conservant une certaine vitesse de progression. Un exemple de mauvais réglage évident consisterait à régler  $\varphi$  à la valeur de  $\frac{13\pi}{8}$ . En effet, pour cette valeur, la capacité d'avance d'un module serait fortement réduite, si ce n'est quasiment nulle. Par conséquent, un des critères à prendre en compte dans le cadre de l'optimisation d'une tâche où les performances d'avance du robot seraient critiques sera relatif à la capacité d'avance d'un module induite par un mécanisme. Ce critère devra évidemment tenir compte du sens de progression du module, le fait étant établi que ce dernier peut changer en fonction de la valeur attribuée au déphasage. L'amplitude du mouvement vertical du corps principal traduisant un coût énergétique lié à la gravité, il est également nécessaire de dégager un critère représentatif de ce phénomène. Néanmoins, ce coût énergétique sera à relativiser si l'on souhaite pouvoir disposer de capacités locales de franchissement, dernier critère évident.

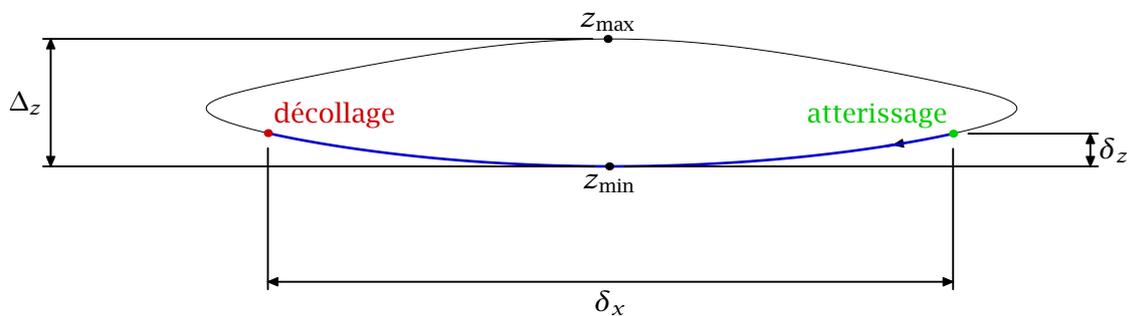


FIGURE 5.1. Représentation des critères d'optimisation pour la trajectoire de déphasage  $\varphi = \frac{5\pi}{8}$

Ainsi, le réglage du déphasage  $\varphi$  devra prendre en compte au moins les critères suivants, tels que représentés relativement à la trajectoire d'un mécanisme par la figure 5.1 :

- $\delta_x$  la capacité de déplacement du module pour la phase de contact d'un mécanisme,
- $\delta_z$  la variation suivant l'axe vertical de la position de l'organe terminal du mécanisme par rapport au corps principal, pour une phase de contact de ce mécanisme,
- $\Delta_z$  la capacité locale de franchissement. .

Les hypothèses permettant de calculer ces critères d'optimisation sont :

- le substrat est un sol plat et rigide,
- l'étude porte sur un module à 8 mécanismes ou deux modules à 4 mécanismes avec une rotation relative bloquée en tangage.

Les courbes caractéristiques de ces critères relatifs au déphasage  $\varphi$  ont été obtenues numériquement.

### 5.1.2 Variation d'altitude de la plate-forme

La variation d'altitude de la plate-forme peut être obtenue localement par l'analyse de la phase de contact d'un mécanisme. Dans un premier temps, considérons la trajectoire d'un mécanisme par rapport au module auquel il appartient. Dans un deuxième temps, on dégage les phases de contact et aérienne du mécanisme dans le cadre de son interaction avec les autres mécanismes du module étudié. Une fois la phase de contact au sol obtenue, on

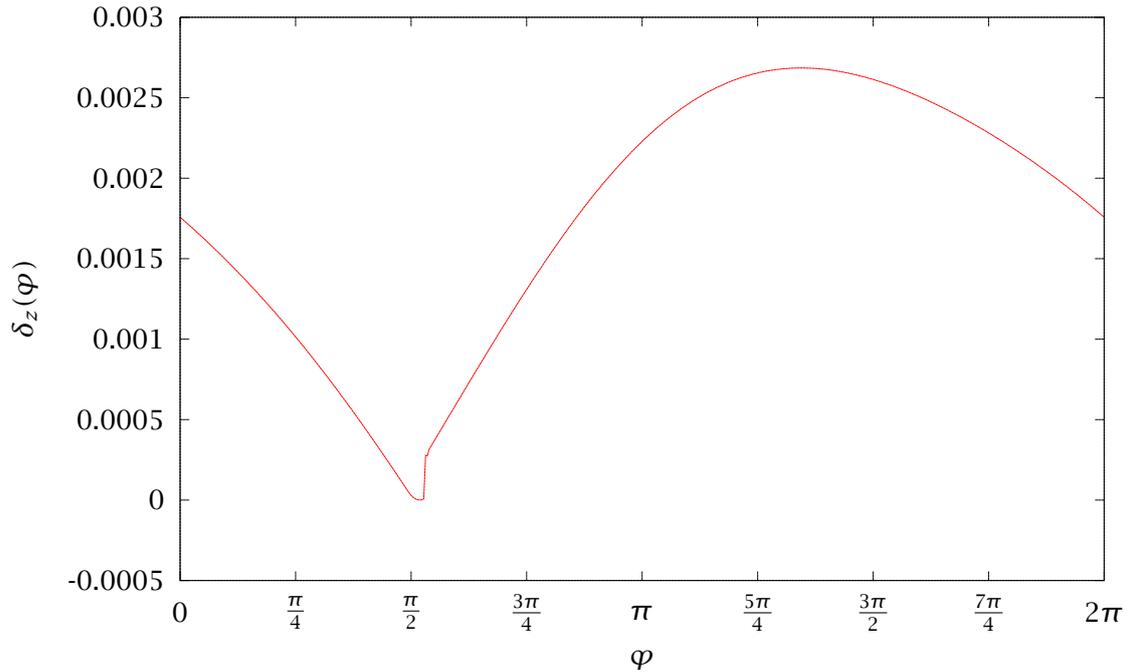


FIGURE 5.2. Caractéristique du critère variation d'altitude en fonction du déphasage  $\varphi$

connaît la localisation par rapport à la plate-forme des points d'atterrissage et de décollage. De la même manière, une analyse simple nous permet d'obtenir le point le plus bas de la trajectoire.

La variation selon l'axe  $\vec{z}$  de la trajectoire du mécanisme lors de la phase de contact au sol correspond au déplacement vertical du module par rapport au sol. Les altitudes des points d'atterrissage et de décollage sont identiques pour les raisons suivantes :

- la trajectoire d'un mécanisme est symétrique (suivant un plan vertical et perpendiculaire à l'axe  $\vec{x}$  local),
- le décollage d'un mécanisme correspond à l'atterrissage d'un autre mécanisme.

Le calcul de ce critère peut alors être effectué à partir des coordonnées verticales du point le plus bas de la trajectoire et, indifféremment, du point de décollage ou d'atterrissage.

Ce critère est intéressant du point de vue énergétique car il correspond à une image partielle du coût énergétique, induite par la pesanteur.

On utilisera la notation  $\delta_z$  pour la désignation de ce critère. La figure 5.2 représente la caractéristique de  $\delta_z$  en fonction du déphasage  $\varphi$  présenté précédemment. Cette caractéristique est obtenue à partir du modèle géométrique direct.

### 5.1.3 Amplitude verticale

L'amplitude verticale d'un mécanisme est obtenue à partir du module géométrique direct. On détermine simplement le point le plus élevé ainsi que le point le plus bas et l'on soustrait leurs coordonnées suivant l'axe vertical.

Ce critère caractérise la capacité locale de franchissement du mécanisme.

On utilisera la notation  $\Delta_z$  pour la désignation de ce critère. La figure 5.3 représente la caractéristique de  $\Delta_z$  en fonction du déphasage  $\varphi$ .

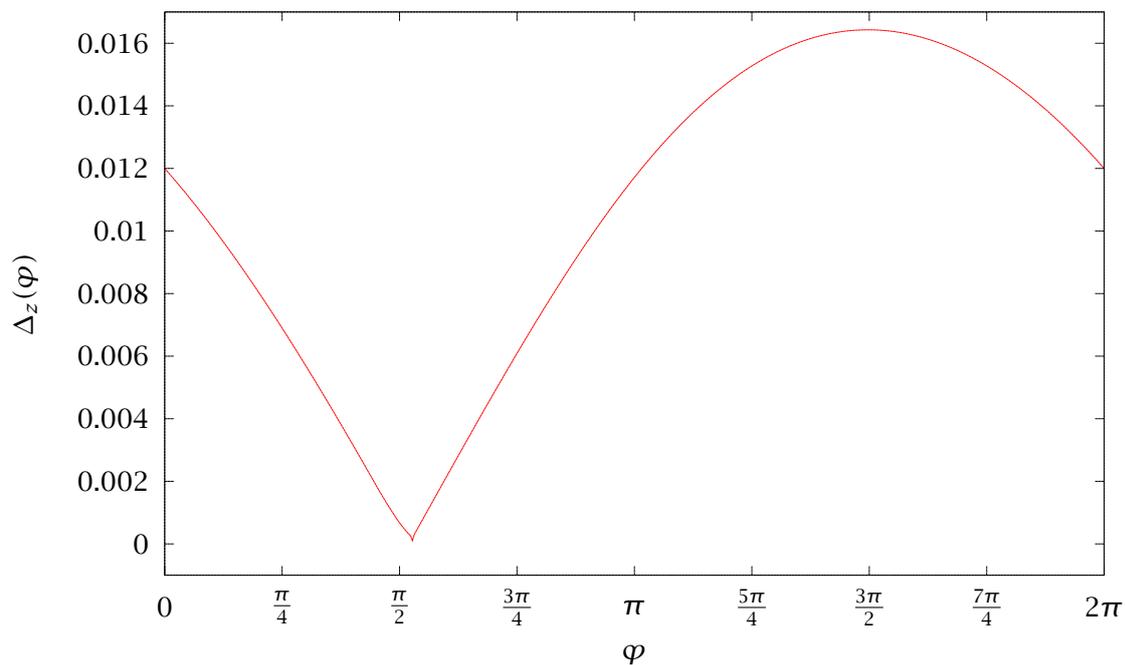


FIGURE 5.3. Amplitude selon  $\vec{z}$  du mouvement de l'organe terminal d'un mécanisme en fonction du déphasage  $\varphi$

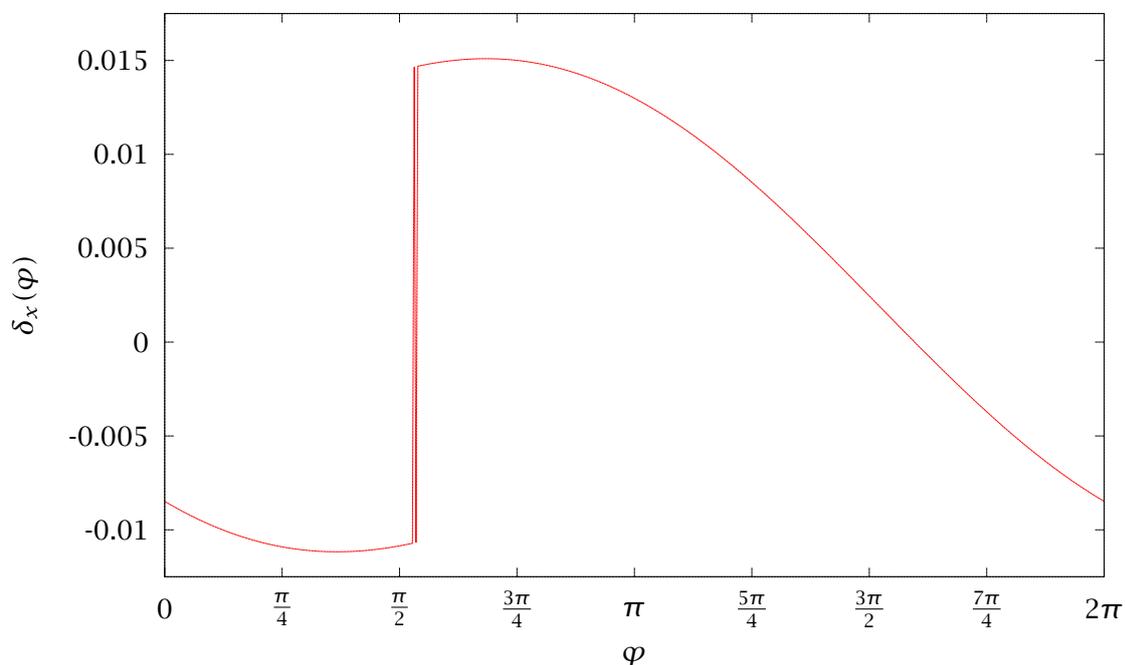


FIGURE 5.4. Caractéristique du critère performance locale de progression dans l'axe longitudinal en fonction du déphasage  $\varphi$

### 5.1.4 Performance locale de progression dans l'axe longitudinal

La performance locale de progression dans l'axe longitudinal correspond à l'amplitude du mouvement de la plate-forme induit par un mécanisme lors de sa phase de contact au sol, relativement à l'axe longitudinal. Pour une entrée identique, le déphasage permet d'obtenir des performances très différentes. Plus cette amplitude sera grande, plus le mouvement du robot dans l'axe longitudinal sera important.

La caractéristique présentée par la figure 5.4 est obtenue directement à partir des points d'atterrissage et de décollage.

On utilisera la notation  $\delta_x$  pour la désignation de ce critère.

### 5.1.5 Exemples de critères croisés

Les critères présentés plus haut sont les plus évidents. D'autres critères peuvent être dégagés en fonction des besoins liés à l'application. De la même manière, à un niveau d'abstraction plus élevé, les influences de ces critères peuvent être combinées dans le but d'aboutir à un critère spécifique, croisé, représentatif de l'optimisation d'une application ou tâche déterminée. En effet, dans la pratique, l'optimisation est complètement dépendante de la tâche. Ainsi, il conviendra de considérer au cas par cas l'introduction d'un critère croisé.

En ne considérant que les critères « unitaires » énoncés précédemment, la combinaison la plus simple est la combinaison linéaire :

$$k_{\delta_x} \delta_x + k_{\delta_z} \delta_z + k_{\Delta_z} \Delta_z \quad (5.1)$$

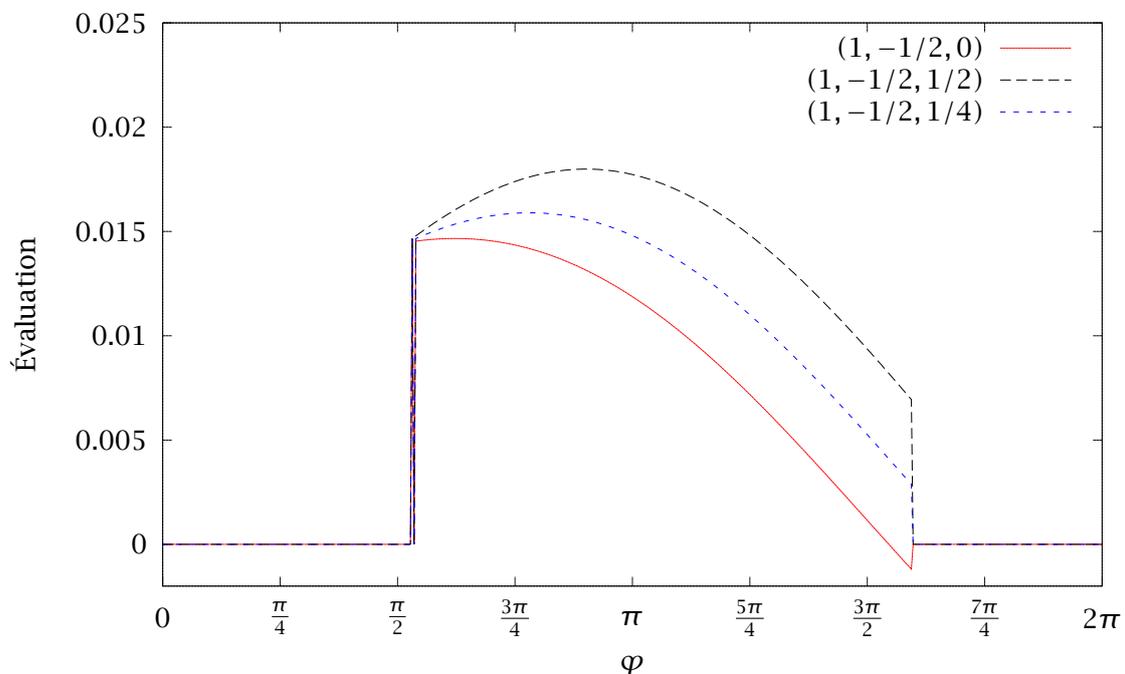


FIGURE 5.5. Critère  $k_{\delta_x} \delta_x + k_{\delta_z} \delta_z + k_{\Delta_z} \Delta_z$  pour différents triplets  $(k_{\delta_x}, k_{\delta_z}, k_{\Delta_z})$

La figure 5.5 représente différents critères possibles pour le choix du déphasage  $\varphi$ . On cherche à maximiser les critères  $\delta_x$  et  $\Delta_z$  et à minimiser le critère  $\delta_z$  (d'où la présence du

signe  $-$ ). Cette figure prend en compte une contrainte supplémentaire : le sens de progression du robot. Cela se traduit par la nullité du critère croisé lorsque  $\delta_x$  est inférieur ou égal à 0.

D'autres critères plus élaborés peuvent être considérés. Par exemple, le coût énergétique de la locomotion.

### **5.1.6 Application à un module à 8 mécanismes**

Considérons l'étude d'un cas d'application simple de l'optimisation. On utilise un module à 8 mécanismes en interaction avec un sol plat. On souhaite franchir une série de marches de hauteur égale à 1 cm. On souhaite régler le déphasage  $\varphi$  de telle sorte que l'on conserve de bonnes performances en terme de vitesse d'avance tout en disposant de la capacité de franchir de petits obstacles. En ne prenant en compte que la figure 5.3, une solution consisterait à choisir le déphasage correspondant au maximum d'amplitude verticale. Or, ce choix n'est pas valable dans le sens où la progression longitudinale due à un mécanisme sera proche de zéro. À cela s'ajoute une importante variation d'altitude de la plate-forme, critère que l'on cherchera à minimiser plutôt que maximiser.

En considérant que le critère le plus important est la performance de progression suivant l'axe longitudinal et que les deux autres critères seront d'importance équivalente, on obtient le critère croisé correspondant à la combinaison linéaire définie par le triplet  $(k_{\delta_x}, k_{\delta_z}, k_{\Delta_z}) = (1, -1/2, 1/2)$ . Le maximum est atteint pour une valeur du déphasage  $\varphi$  légèrement inférieure à  $\pi$ . Pour simplifier le réglage, on choisit  $\varphi = \pi$  et on teste une application en simulation.

La figure 5.6 représente les différentes étapes de franchissement de la première marche par un module à 8 mécanismes. La figure 5.7 correspond à une deuxième marche.

## **5.2 Proposition d'une nouvelle structure robotique**

### **5.2.1 Motivations**

La structure initialement proposée est constituée de plusieurs modules, articulés les uns par rapport aux autres, à 8 mécanismes tels que définis aux sections 4.3.1 et 4.1. La liaison entre deux modules peut être modélisée par une liaison pivot d'axe vertical, bien qu'elle dispose technologiquement de petits débattements angulaires suivant les autres axes principaux de rotation.

Deux problèmes principaux sont apparus lors des premières simulations de structures à  $n$  modules respectant la configuration initialement étudiée.

Le premier de ces problèmes réside dans l'utilisation de modules à 8 mécanismes. En effet, la longueur d'un module, dépendant directement du nombre de mécanismes, est disproportionnée relativement aux autres dimensions du module. Cette disproportion est à l'origine d'une limitation non négligeable de la capacité du robot à suivre une trajectoire de référence présentant une courbure importante. Cette limitation sera d'autant plus importante dans le cas d'une utilisation dans un milieu fermé et sinueux tel que le coude d'un tuyau par exemple. Enfin, dans le cas d'une utilisation sur un sol irrégulier la longueur

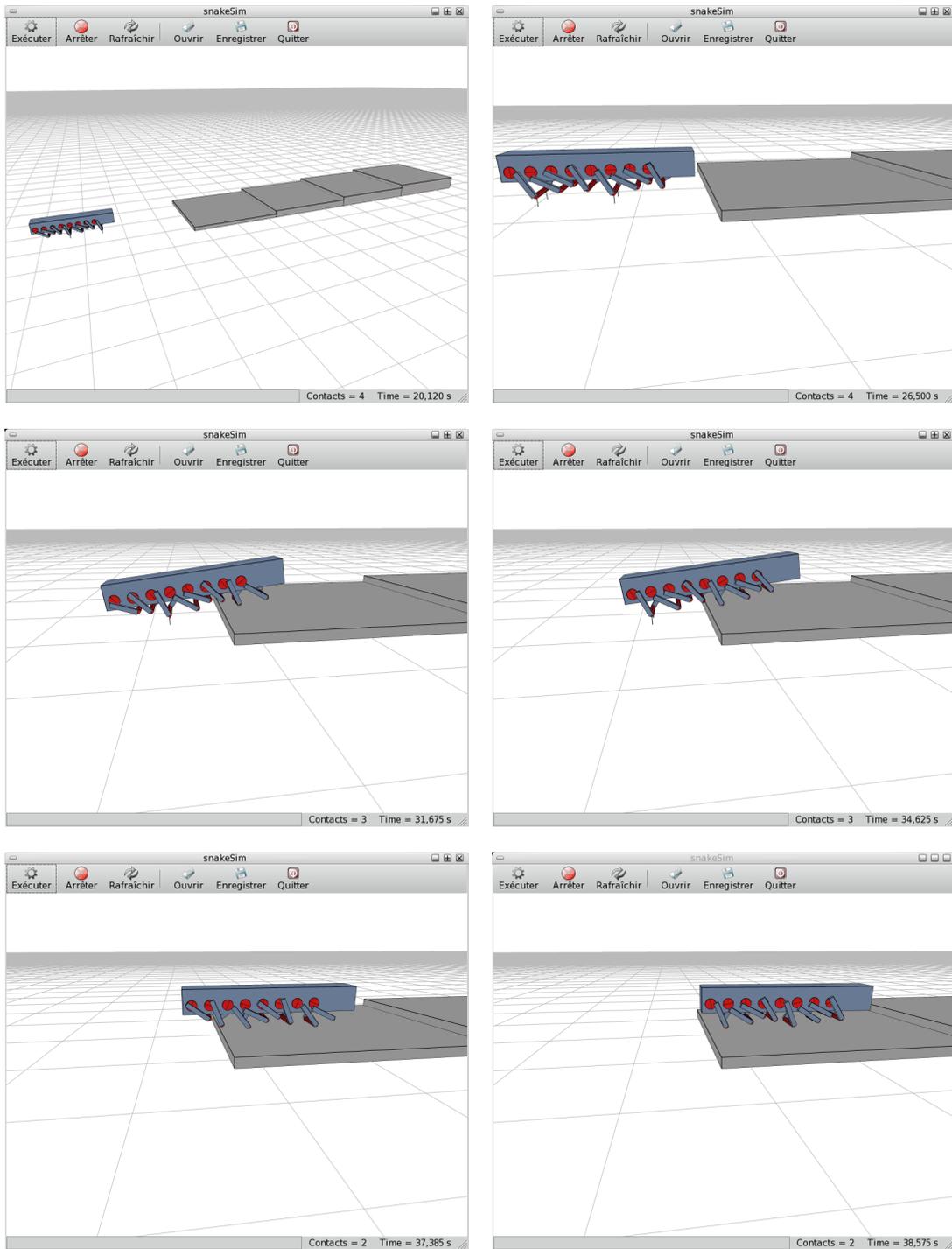


FIGURE 5.6. Franchissement de la première marche ( $\varphi = \pi$ )

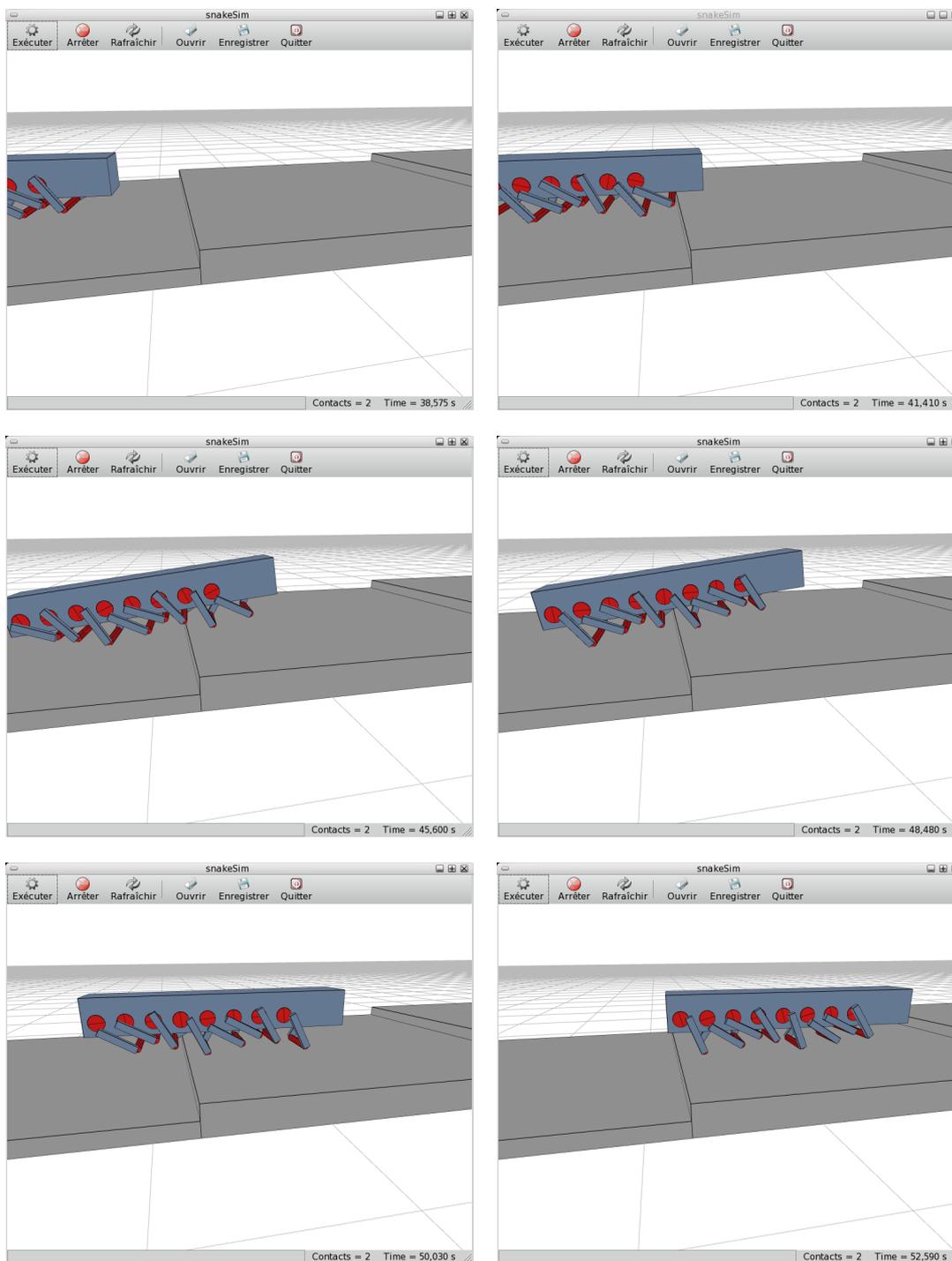


FIGURE 5.7. Franchissement de la deuxième marche ( $\varphi = \pi$ )

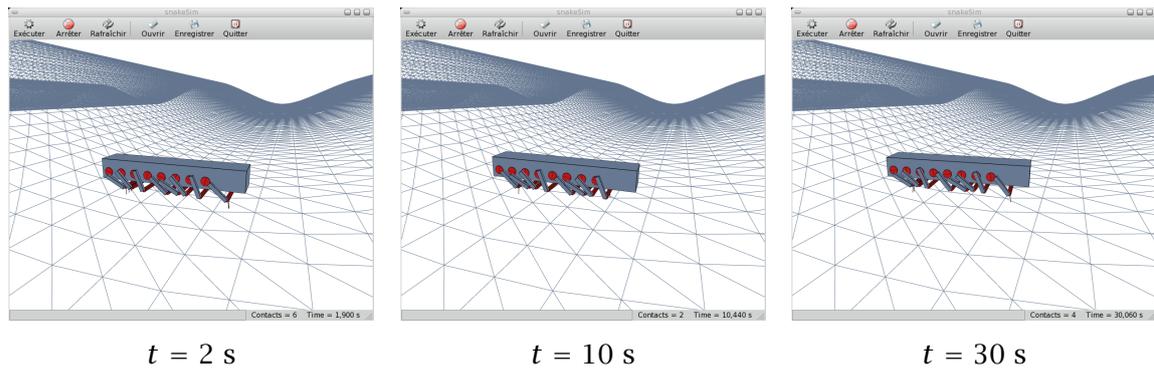


FIGURE 5.8. Simulation d'une situation de blocage d'un module à 8 mécanismes

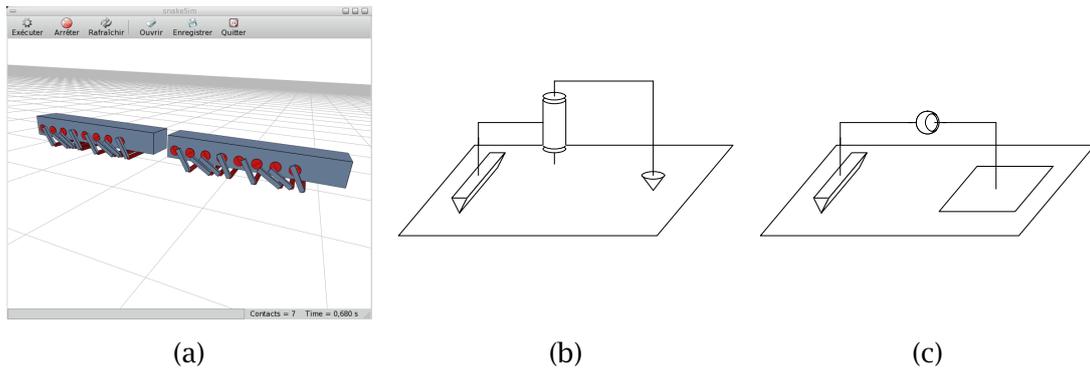


FIGURE 5.9. Modélisation cinématique de l'interaction d'un robot composé de 2 modules à 8 mécanismes avec un sol plan au repos - (a) modèle de simulation, (b) schéma cinématique (débattements en roulis et tangage négligeables), (c) schéma cinématique (débattements en roulis et tangage non négligeables)

disproportionnée d'un module est susceptible de nuire à l'adaptation de la structure robotique au profil des irrégularités, favorisant de fait des situations de blocage ou l'incapacité d'utiliser tous les éléments normalement moteurs. Pour l'exemple, une situation de blocage est représentée à la figure 5.8. Cette situation de blocage est due à la forme concave du sol.

Le second problème est plus délicat à décrire, sa description nécessite un exemple.

Considérons tout d'abord un seul module à 8 mécanismes sur un sol plat. Au repos, les mécanismes fonctionnent par paires comme nous l'avons vu dans le chapitre précédent. Les géométries du sol et du module ne sont pas parfaites. Ainsi, la liaison entre le module et le sol est modélisable par un appui plan ou trois appuis ponctuels. Par conséquent, un mécanisme locomoteur sera lié au sol par l'intermédiaire d'une liaison linéaire rectiligne tandis que l'autre le sera par une liaison ponctuelle.

Maintenant, on associe un second module au premier. La figure 5.9(a) représente cette association de 2 modules. La conception initiale propose de relier les modules par une liaison de type pivot disposant de faibles débattements en roulis et en tangage. Par conséquent, soit on néglige ces débattements et la liaison inter-modulaire est une liaison pivot, soit on ne les néglige pas et la liaison est alors une liaison rotule. Nous nous proposons d'étudier les deux cas de figure.

1. La liaison inter-modulaire est modélisée par une liaison pivot d'axe vertical. On ajoute au système de 2 modules un degré de liberté en rotation dans le plan horizontal. La liaison entre le système et le sol est toujours modélisable par trois contacts ponctuels. Par exemple,

un des mécanismes du premier module est lié au sol par une liaison linéaire rectiligne et un des mécanismes du second module est lié au sol par une liaison ponctuelle. Cet exemple est illustré par la figure 5.9(b). On souhaiterait que 4 mécanismes soient locomoteurs mais avec ce type de liaison, on aura au mieux 3 mécanismes en contact avec le sol. Au pire, seuls deux mécanismes seront locomoteurs.

2. La liaison inter-modulaire est modélisée par une liaison rotule. Trois degrés de liberté en rotation sont ajoutés au système. La liaison entre le système et le sol est modélisable par cinq appuis. Un exemple de ce cas de figure est représenté figure 5.9(c). Dans le meilleur des cas, les quatre mécanismes normalement locomoteurs seront en contact avec le sol. Dans le cas général, ce nombre descend à trois.

L'idéal serait que chaque mécanisme locomoteur soit lié à un sol plat par l'intermédiaire d'une liaison linéaire rectiligne et ce quelque soit le nombre de modules composant le robot. Ainsi, nous avons été amenés à réfléchir sur :

- la configuration et le nombre de mécanismes d'un module,
- la nature des articulations inter-modulaires.

### 5.2.2 Module

Un module à 8 mécanismes présente une certaine stabilité intrinsèque. Néanmoins, l'argument de stabilité intrinsèque n'est pas justifiable car le robot de type serpent sera obligatoirement une structure composée par plusieurs modules articulés, ne serait-ce que pour disposer de capacités de changement de direction. De plus, nous avons soulevé précédemment les problèmes que peut poser l'utilisation d'un module dont les mécanismes locomoteurs fonctionneraient par paire.

Par conséquent, notre proposition repose sur l'utilisation de modules à 4 mécanismes locomoteurs. La définition de la configuration angulaire des mécanismes est conservée et respecte toujours la loi  $q_i = \omega t + \frac{i\pi}{2}$ . Dans ce cas, un module ne sera plus intrinsèquement stable mais verra sa longueur diminuer presque de moitié. De plus, le fait de diminuer la longueur d'un mécanisme a une influence positive sur la capacité du robot à changer de cap. En effet, en considérant deux modules, reliés entre eux par une liaison pivot d'axe vertical, en interaction avec un sol plan, la grandeur critique est la courbure maximale admissible

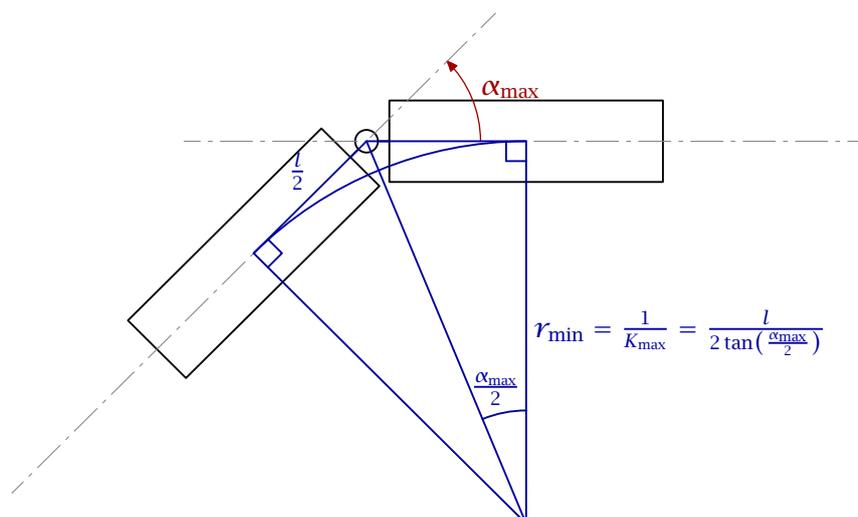


FIGURE 5.10. Courbure maximale admissible

pour la limite angulaire articulaire. Pour une limite articulaire donnée, la courbure dépend uniquement de la distance entre le centre géométrique du module et le centre de la liaison comme le représente la figure 5.10. Par conséquent, si cette distance diminue de moitié, le rayon de courbure minimum diminue de moitié, ce qui double la valeur de la courbure maximale.

Enfin, l'utilisation de modules à 4 mécanismes permet, en fonction du choix de la liaison inter-modulaire, d'optimiser le nombre de mécanisme en contact avec le sol.

Du point de vue de la locomotion, nous négligerons le fait qu'un module à 8 mécanismes soit susceptible d'embarquer une charge utile supérieure à celle d'un module à 4 mécanismes.

### 5.2.3 Articulations inter-modulaire

La liaison inter-modulaire du prototype VIPeRe est une liaison pivot d'axe vertical. En effet, même si elle dispose de possibilités de débattements angulaires, ces débattements admissibles sont relativement faibles. De plus, seule la rotation suivant l'axe de la liaison pivot peut être actionnée et, en l'état, les débattements ne peuvent en aucun cas être pilotés. Néanmoins, le problème posé par cette articulation ne survient pas si l'on se restreint à des déplacements sur des surfaces planes ou régulières, relativement aux grandeurs du robot, et si l'on modifie la conception d'un module. Nous ne présenterons pas ici la conception mécanique de cette liaison.

Pour le modèle utilisé en simulation, nous utiliserons une liaison rotule. Dans le cas d'une utilisation sur un sol très irrégulier présentant des discontinuités, il est nécessaire d'actionner les trois axes en rotation de la liaison. Ceci dans le but d'utiliser la redondance cinématique pour le franchissement d'obstacle. Dans le cadre de nos travaux, nous nous sommes limités à une gamme de substrats rigides allant du sol plat au sol présentant des variations régulières.

Ainsi, notre proposition d'articulation inter-modulaire est une liaison rotule avec :

- une rotation en lacet actionnée afin de permettre la changement de cap,
- une rotation en tangage libre pour que la structure complète puisse s'adapter au profil du terrain si celui-ci présente des variations régulières de niveau,
- une rotation en roulis libre afin d'optimiser le nombre de points de contact.

Nous limiterons les débattements de la rotation en lacet à  $\pm 45^\circ$ . La rotation en tangage sera limitée physiquement par le contact entre deux modules et la rotation en roulis n'aura pas de limite.

### 5.2.4 Conclusion

Le choix du nombre de modules dépend de contraintes liées à la commande du robot. Ainsi, ce point sera traité dans le chapitre suivant, lors de la proposition d'un algorithme de commande. Néanmoins, il est évident que la structure ne pourra pas changer de cap si l'on ne dispose pas d'au moins deux modules. La figure 5.11 représente la vue de dessus d'une structure composée de 4 modules à 4 mécanismes ainsi que son paramétrage complet. Les mécanismes ne sont pas représentés sur cette figure. Les modules du robot sont numérotés de 1 à  $N = 4$  en partant du module de tête. De plus, l'articulation  $i$  est l'articulation reliant

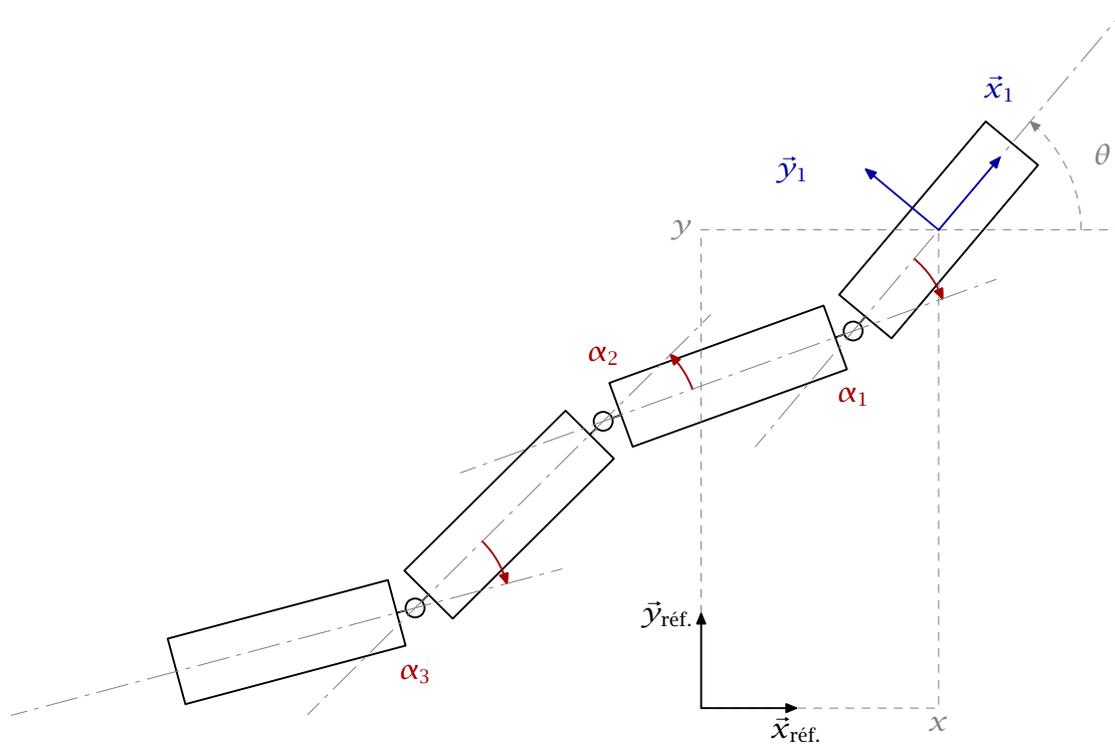


FIGURE 5.11. Vue de dessus de la structure proposée

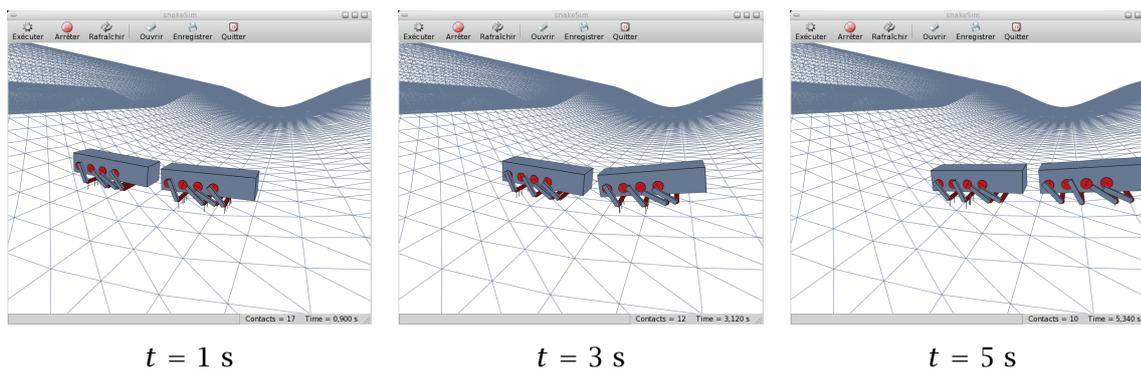
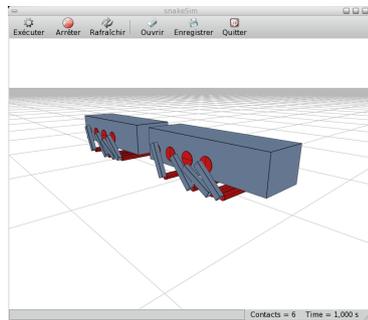
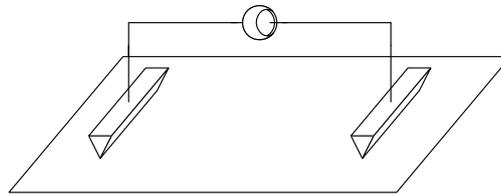


FIGURE 5.12. Simulation d'un robot constitué de 2 modules à 4 mécanismes



(a)



(b)

FIGURE 5.13. Modélisation cinématique de l'interaction d'un robot composé de 2 modules à 4 mécanismes avec un sol plan au repos

le module  $i$  au module  $i+1$ . Nous utiliserons toujours cette convention dans la suite de ce manuscrit.

L'utilisation d'un robot constitué de 2 modules à 4 mécanismes et d'une liaison intermodulaire de type rotule permet d'éviter la situation de blocage représentée à la figure 5.8. La figure 5.12 montre le déroulement de la simulation d'un robot à 2 modules basé sur notre proposition pour des conditions identiques à celles conduisant à la situation de blocage précédemment évoquée.

Enfin, notre proposition permet une meilleure utilisation des mécanismes de locomotion comme le montre la figure 5.13. En effet, chaque interaction d'un module avec le sol sera modélisable par une liaison linéaire rectiligne.



# Chapitre 6

## Utilisation de la simulation pour le développement d'algorithmes de commande

---

### Sommaire

---

6.1	Approche géométrique en boucle ouverte .....	81
6.1.1	Définition de la commande pour une tâche de suivi de trajectoire simple sur sol plat .....	82
6.1.2	Résultats de simulation .....	83
6.1.3	Conclusion .....	83
6.2	Approche cinématique .....	86
6.2.1	Description de la méthode .....	86
6.2.2	Trajectoires .....	88
6.3	Application de l'approche cinématique .....	88
6.3.1	Suivi de trajectoire sur sol plat avec 4 modules .....	88
6.3.1.1	Description de la tâche .....	88
6.3.1.2	Résultats .....	89
6.3.2	Suivi de trajectoire sur sol plat avec 6 modules .....	89
6.3.2.1	Description de la tâche .....	89
6.3.2.2	Résultats .....	93
6.3.3	Suivi de ligne droite sur sol plat avec 4 modules et état initial perturbé ..	93
6.3.3.1	Description de la tâche .....	93
6.3.3.2	Résultats .....	93
6.4	Conclusion .....	93

---

Maintenant que nous disposons d'un modèle de notre structure robotique articulée utilisable pour la simulation d'un déplacement, nous allons nous intéresser à la réalisation d'algorithmes de commande pour le suivi de trajectoire. L'évaluation de la performance ainsi que l'élaboration de la commande devront prendre en compte le fait que le robot est constitué de plusieurs modules articulés répartis dans le sens de la longueur.

Ce chapitre présente une approche géométrique puis une approche cinématique pour la commande de la structure robotique proposée. L'évaluation de l'approche cinématique s'appuie sur l'analyse des résultats de plusieurs simulations.

### 6.1 Approche géométrique en boucle ouverte

Au début de cette recherche, la nécessité de tester l'environnement de simulation nous a conduit à proposer un premier algorithme de commande [AUBIN *et al.*, 2005] basé sur

une approche géométrique simple en boucle ouverte. L'algorithme résultant utilise une trajectoire très simple constituée de deux segments de droite et d'un quart de cercle. La portée de cet algorithme est très limitée du point de vue de la commande. Néanmoins, son implémentation a été riche en enseignements.

Le robot utilisé est composé de plusieurs modules à 8 mécanismes articulés les uns par rapport aux autres par des liaisons pivot d'axe vertical.

### 6.1.1 Définition de la commande pour une tâche de suivi de trajectoire simple sur sol plat

Cet algorithme de commande est complètement dépendant de la trajectoire que nous nous proposons de suivre. Ainsi, la définition de cette trajectoire fait partie intégrante de l'élaboration de l'algorithme de commande.

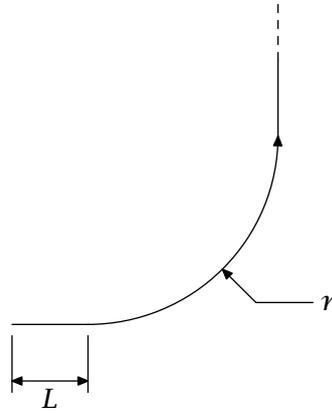


FIGURE 6.1. Trajectoire de référence (approche géométrique)

On utilise un robot constitué de  $N$  modules à 8 mécanismes reliés entre eux par une liaison pivot d'axe vertical. La vitesse d'entraînement des mécanismes est constante et identique pour tous les modules.

À l'état initial, le robot est aligné dans l'axe du premier segment constitutif de la trajectoire à suivre, le centre du premier module étant localisé à l'extrémité de la trajectoire.

On évalue le trajet parcouru par les modules par la somme des déplacements retournés par la simulation (on prend ainsi en compte les glissements de chacun des modules par rapport au sol). Soit  $s_i$  l'évaluation du trajet parcouru par le module  $i$  et  $M_i$  le centre géométrique du corps principal du module  $i$ . Pour un pas de simulation  $\Delta t$ , on définit  $s_i(k\Delta t) = s_i^k$  par

$$s_i^k = \sum_{j=1}^k \sqrt{\left( \left( \overrightarrow{OM}_i^j - \overrightarrow{OM}_i^{j-1} \right) \cdot \vec{x}_i \right)^2 + \left( \left( \overrightarrow{OM}_i^j - \overrightarrow{OM}_i^{j-1} \right) \cdot \vec{y}_i \right)^2}$$

On estime ensuite la position angulaire absolue d'un module en fonction de l'estimation de son abscisse curviligne et de la trajectoire à suivre. L'orientation absolue du module  $i$  est alors définie par

$$\theta_i = \begin{cases} 0 & \text{si } s_i < L + il, \\ \frac{\pi}{2} & \text{si } s_i \geq L + il + \frac{R\pi}{2}, \\ s_i - \frac{L+il}{R} & \text{si } L + il \leq s_i < L + il + \frac{R\pi}{2}. \end{cases}$$

On calcule la différence angulaire relative entre deux modules reliés par une liaison pivot puis on commande les positions angulaires de chaque articulation. Soit  $\alpha_i$  la position angulaire relative désirée entre deux modules telle que décrite par la figure 5.11, les asservissements en position des articulations sont obtenus avec la commande, proportionnelle, en vitesse suivante :

$$\dot{\alpha}_i = k (\alpha_{Mi} - \alpha_{Di})$$

avec  $\alpha_{Mi}$ , la mesure issue de la simulation et  $\alpha_{Di}$  la position désirée définie par

$$\alpha_{Di} = \theta_i - \theta_{i+1}.$$

Cet algorithme a été implémenté dans l'environnement de simulation. Il repose sur les hypothèses suivantes :

- le sol avec lequel interagit le robot est plat,
- le robot démarre en ligne droite,
- le nombre de modules doit être suffisant afin que la structure complète présente une certaine inertie relativement à l'actionnement d'une liaison.

### 6.1.2 Résultats de simulation

On doit disposer d'au minimum 3 modules pour que l'algorithme puisse fonctionner.

Les trajectoires des centres géométriques des corps principaux sont données pour chaque module à la figure 6.2. Ces trajectoires suivent relativement bien la trajectoire de référence. Néanmoins, on remarque la présence de perturbations de la trajectoire surtout pour le premier et le dernier module. Ceci s'explique par le fait que ces modules, du fait de leur positionnement à l'extrémité de la structure sont particulièrement sujets au glissement. Par exemple, à partir de l'instant où le premier module aborde la portion de cercle, la première articulation est commandée pour corriger son orientation tandis que le reste de la structure continue à progresser sur le premier segment de la trajectoire. L'inertie des trois derniers modules est plus importante que celle du premier pris isolément. Cette situation implique nécessairement une situation de glissement du premier module, la vitesse de glissement étant composée d'une part par la vitesse résultant de la rotation du module et d'autre part par la vitesse imposée par le reste de la structure. Enfin, on constate la présence d'une dérive pour le suivi du second segment de droite de la trajectoire de référence. Ce phénomène est évidemment induit par le glissement des modules que l'on vient d'évoquer. La figure 6.3 présente les étapes de la simulation d'un robot constitué de 5 modules.

### 6.1.3 Conclusion

La portée de l'approche géométrique présentée ici est très limitée du fait de l'utilisation d'hypothèses trop restrictives telles que la limitation à un sol plat. De plus, la méthode que nous utilisons n'est pas basée sur un calcul d'erreur de positionnement. Par conséquent,

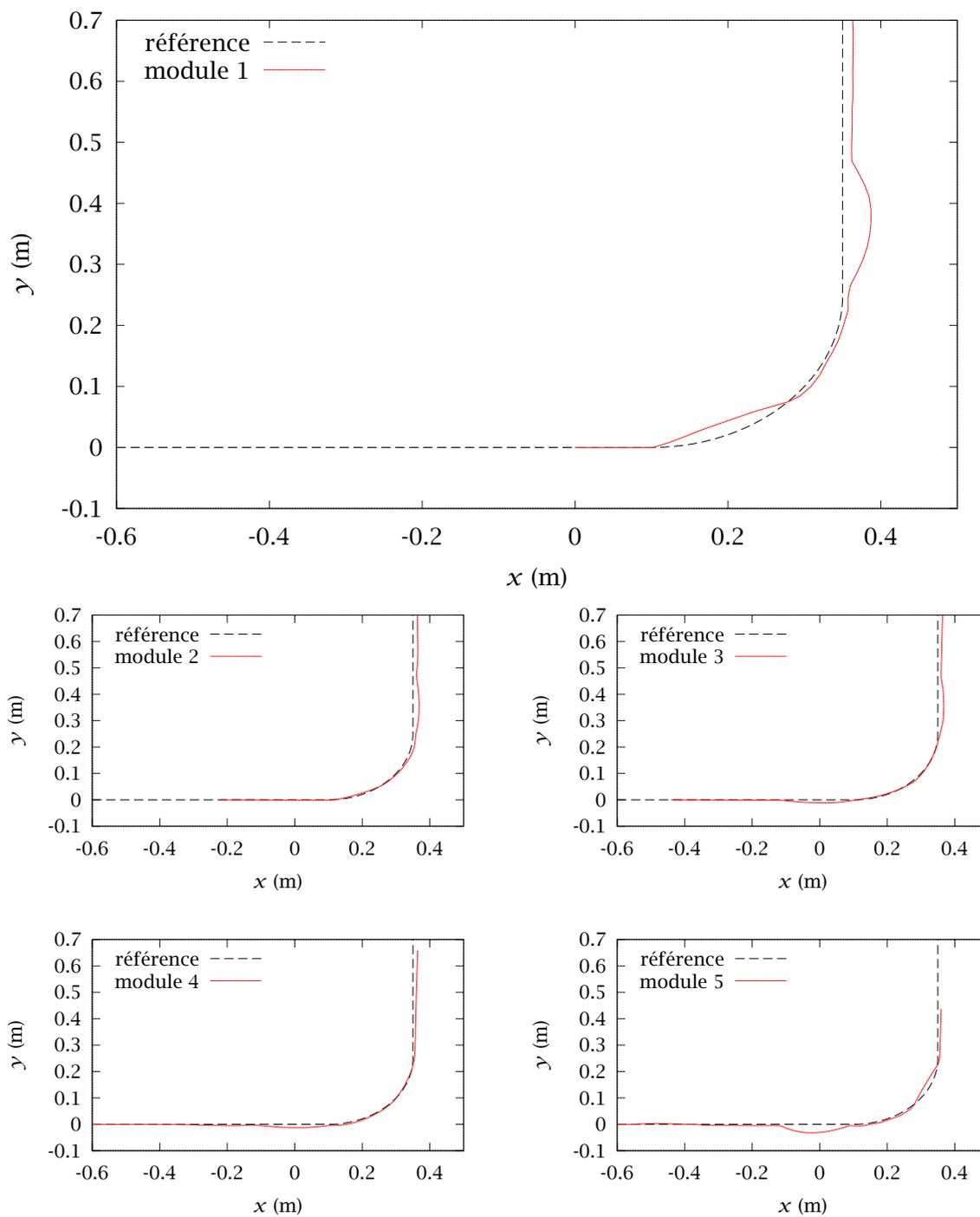


FIGURE 6.2. Trajectoires des centres géométriques des corps principaux des modules (approche géométrique)

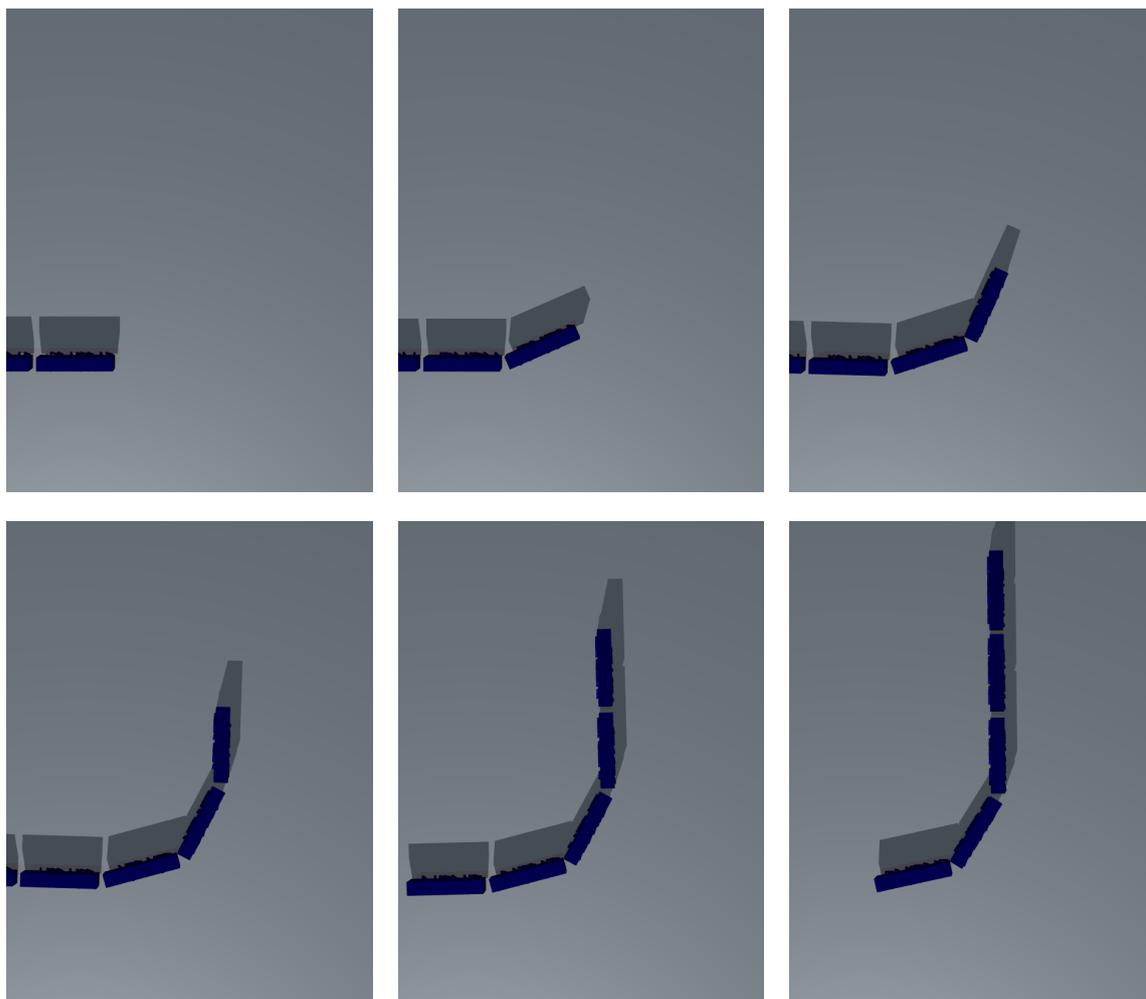


FIGURE 6.3. Étapes de simulation d'un robot constitué de 5 modules à 8 mécanismes (approche géométrique)

toute perturbation due à l'environnement est susceptible de fausser la trajectoire du robot. Néanmoins, l'implémentation de l'algorithme de commande correspondant et l'observation du comportement du robot simulé nous ont permis l'élaboration de premiers tests lors du développement de l'environnement de simulation.

Il est possible d'utiliser cet algorithme avec d'autres types de trajectoire mais cela ne représente qu'un intérêt minime.

## 6.2 Approche cinématique

### 6.2.1 Description de la méthode

La méthode que nous nous proposons de décrire dans cette section repose sur une hypothèse simplificatrice forte : pour la commande d'un module  $i$ , on suppose que l'inertie des modules de  $i + 1$  jusqu'à  $N$  est suffisamment importante pour que l'on puisse considérer que l'action de rotation autour de l'axe vertical local de l'articulation  $i$ , reliant le module  $i$  au module  $i + 1$ , n'a pour effet qu'une rotation du module  $i$  autour du centre de rotation de l'articulation.

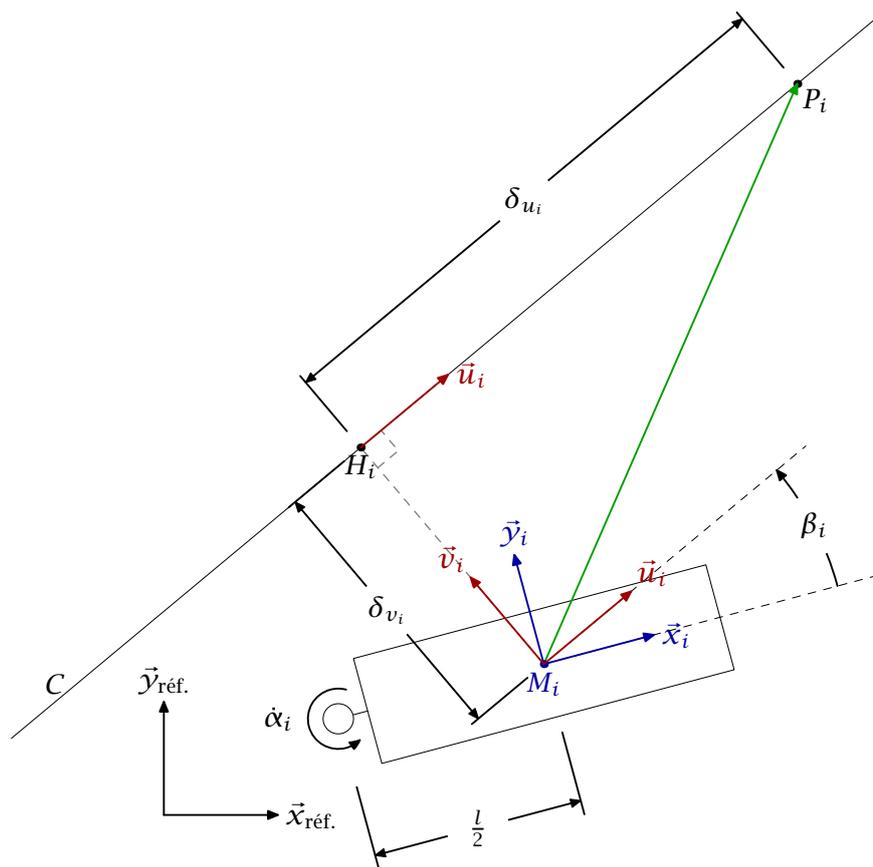


FIGURE 6.4. Module  $i$

La figure 6.4 représente la mise en situation d'un module  $i$  et d'une courbe  $C$ , la trajectoire de référence à parcourir. Le point  $H_i$  est le point le plus proche de  $M_i$  appartenant à la courbe  $C$ . Le vecteur  $\vec{u}_i$  est défini comme étant un vecteur unitaire colinéaire à la tangente à la courbe  $C$  au point  $H_i$ .

L'erreur en position du module  $i$  par rapport à la courbe  $C$  présente deux composantes :

- $\delta_{v_i}$ , l'erreur en distance ;
- $\beta_i$ , l'erreur angulaire entre l'axe longitudinal du module et la tangente à la courbe  $C$ .

L'instant de référence  $t$  est tel que le centre géométrique  $M_i$  du module  $i$  n'est pas sur la courbe  $C$ . On cherche à rejoindre la courbe à l'instant  $t + \Delta T$  en  $P_i$ . On fait l'hypothèse que le déplacement et les variations de la courbe  $C$  sont tels que  $H_iP_i$  est un segment de droite.

On peut par conséquent noter vectoriellement :

$$\overline{M_iP_i} = \overline{M_iH_i} + \overline{H_iP_i} = \delta_{v_i}\vec{v}_i + \delta_{u_i}\vec{u}_i$$

Pour un module  $i$ , les grandeurs de commandes sont :

-  $V_i$ , la vitesse de propulsion induite par l'association de plusieurs mécanismes de locomotion ;

-  $\dot{\alpha}_i$ , la vitesse de rotation induite par l'actionneur du mécanisme  $i + 1$ .

On considère que  $V_i$  est connu. Le problème se résume à la détermination de  $\dot{\alpha}_i$  pour que le module puisse parcourir  $\overline{M_iP_i}$  en un temps  $\Delta T$ .

$$\frac{\overline{M_iP_i}}{\Delta T} = \frac{\delta_{v_i}\vec{v}_i + \delta_{u_i}\vec{u}_i}{\Delta T} = \dot{\alpha}_i\vec{z}_i \times \frac{l}{2}\vec{x}_i + V_i\vec{x}_i = \frac{\dot{\alpha}_i l}{2}\vec{y}_i + V_i\vec{x}_i$$

On obtient alors

$$V_i = \frac{\delta_{v_i}\vec{v}_i \cdot \vec{x}_i + \delta_{u_i}\vec{u}_i \cdot \vec{x}_i}{\Delta T} = \frac{\delta_{u_i} \cos(\beta_i) - \delta_{v_i} \sin(\beta_i)}{\Delta T}$$

$$\dot{\alpha}_i \frac{l}{2} = \frac{\delta_{v_i}\vec{v}_i \cdot \vec{y}_i + \delta_{u_i}\vec{u}_i \cdot \vec{y}_i}{\Delta T} = \frac{\delta_{v_i} \cos(\beta_i) + \delta_{u_i} \sin(\beta_i)}{\Delta T}$$

L'expression du déplacement virtuel  $\delta_{u_i}$  est alors :

$$\delta_{u_i} = \frac{V_i \Delta T}{\cos(\beta_i)} + \delta_{v_i} \tan(\beta_i)$$

On arrive finalement à

$$\dot{\alpha}_i = \frac{2\delta_{v_i} [\cos(\beta_i) + \sin(\beta_i) \tan(\beta_i)]}{l\Delta T} + \frac{2V_i \tan(\beta_i)}{l}$$

$$\dot{\alpha}_i = \frac{2\delta_{v_i}}{l\Delta T \cos(\beta_i)} + \frac{2V_i \tan(\beta_i)}{l} \quad (6.1)$$

L'équation (6.1) se compose de deux termes distincts. Afin de pondérer chacun de ces termes on ajoute les gains  $k_1$  et  $k_2$  :

$$\dot{\alpha}_i = k_1 \frac{2\delta_{v_i}}{l\Delta T \cos(\beta_i)} + k_2 \frac{2V_i \tan(\beta_i)}{l} \quad (6.2)$$

La commande peut être paramétrée par l'intermédiaire des grandeurs  $\Delta T$ ,  $k_1$  et  $k_2$ . Nous ne proposons pas de méthode automatique pour le réglage de ces paramètres de commande. En fonction des valeurs affectées à ces paramètres, on renforce soit l'influence de l'erreur en distance, soit celle de l'erreur angulaire. L'intégration de la bibliothèque EO [EO, 2006 et KEIJZER *et al.*, 2001] dans notre environnement de simulation offre la possibilité d'utiliser des techniques d'apprentissage pour régler ces paramètres pour une tâche donnée. La faisabilité de la mise en place de ce type d'apprentissage pour la commande a été vérifiée. Des premiers tests d'apprentissage ont été effectués sur les paramètres  $k_1$  et  $k_2$  en considérant

un paramétrage identique de l'algorithme de commande pour tous les modules du robot. Néanmoins, le choix des paramètres d'apprentissage tels que le type d'opérateurs utilisé est complexe et ne relève pas de notre domaine d'expertise. Par conséquent, les résultats de ces expérimentations des techniques d'apprentissage ne figurent pas dans ce manuscrit.

## 6.2.2 Trajectoires

De notre point de vue, la tâche de locomotion consiste à suivre ou à rejoindre une trajectoire. Une méthode simple pour l'utilisateur est de définir une trajectoire de référence par la donnée de points de passage. La création d'une courbe interpolée à partir de ces points doit rester à la charge de l'environnement de simulation. Différents types d'interpolations existent mais elles ne sont pas tous adaptées à notre application de locomotion robotique. Néanmoins, notre structure robotique n'induit pas les mêmes contraintes quant au choix de la trajectoire que celles imposées par la reproduction du mode d'ondulation latérale avec des roues passives. Dans un souci d'efficacité et de simplicité, nous avons fait le choix d'utiliser les outils d'interpolation proposés par la bibliothèque de calcul GSL. Les types d'interpolation disponibles sont : linéaire, polynomiale, spline cubique naturelle, spline cubique périodique, akima et akima périodique. Nous aurions pu orienter notre choix vers d'autres types d'interpolation mais l'utilisation de splines cubiques permet d'obtenir une continuité en pente et en courbure et constitue un bon compromis entre complexité et précision.

Nous ne présenterons pas ici la méthode de calcul des polynômes d'interpolation par splines cubiques naturelles que nous utilisons.

## 6.3 Application de l'approche cinématique

### 6.3.1 Suivi de trajectoire sur sol plat avec 4 modules

#### 6.3.1.1 Description de la tâche

On simule une structure composée de 4 modules à 4 mécanismes. La trajectoire de référence est calculée à partir de la donnée des coordonnées de 9 points de passage. Ces coordonnées sont données dans le tableau 6.1.

Les vitesses d'entraînement des mécanismes sont identiques pour tous les modules. Leurs valeurs sont fixées à  $\pi$  rad/s et sont constantes durant toute la durée de la simulation. Les articulations inter-modulaires sont commandées en lacet suivant la méthode décrite à la section 6.2.1 tandis que les rotations en roulis et en tangage sont libres. Les moteurs pilotant

TABLEAU 6.1. Coordonnées des points de passage utilisés pour le calcul de la trajectoire de référence

$x$ (m)	-1,0	0,0	0,1	0,6	1,5	2,0	2,3	4,0	6,0
$y$ (m)	0,0	0,0	0,0	-0,1	-0,3	0,4	0,5	0,0	0,0

les rotations en lacet sont paramétrés avec une valeur raisonnable de couple maximal de 0,5 Nm.

Les valeurs des paramètres de commande sont réglées empiriquement et sont communes à toutes les articulations. Leurs valeurs sont :  $k_1 = 0,2$ ,  $k_2 = 2$  et  $\Delta T = 1$  s.

À l'instant  $t = 0$  s, le robot est situé sur la trajectoire de référence et le module de tête est centré sur l'origine. On utilise un pas de temps de simulation de 10 ms.

### 6.3.1.2 Résultats

La figure 6.5 représente les trajectoires des 4 modules en réponse aux consignes de vitesse des articulations telles que représentées par la figure 6.6.

On constate que les consignes représentées à la figure 6.6 sont bruitées. Ce phénomène est une conséquence de l'utilisation de  $V_i$  pour le calcul de la commande. En effet, le terme  $V_i$  de l'équation (6.2) est calculé à partir de la mesure à l'instant  $t$  de la vitesse linéaire du module  $i$ . Ainsi, le bruit que l'on retrouve au niveau des consignes articulaires est issu des perturbations numériques de la vitesse linéaire du module, induites par l'utilisation d'un pas de temps relativement grand.

En comparant les figures 6.8 et 6.6 on remarque que, pour la tâche simulée, la prise en compte des variations des erreurs en orientation sont prédominantes sur celles des erreurs en distance.

Les figures 6.7 et 6.8 montrent l'évolution au cours du temps des erreurs en position  $\delta_{v_i}$  et  $\beta_i$ . On constate que les modules commandés ne s'éloignent pas à plus de 2 cm du point le plus proche de la trajectoire. De plus, les erreurs d'orientation des modules par rapport à celles données par la tangente au point le plus proche de la trajectoire ne dépassent pas 0,15 rad, soit moins de 9 °.

Des captures d'écrans du déroulement complet de la tâche simulée sont données par la figure 6.9.

On constate que les courbes d'erreurs, et par conséquent de commandes articulaires, reproduisent presque à l'identique, avec un retard, les formes d'ondes de la première articulation. Cette approche permet au système robotique de progresser dans la trace du module de tête.

## 6.3.2 Suivi de trajectoire sur sol plat avec 6 modules

### 6.3.2.1 Description de la tâche

TABLEAU 6.2. Coordonnées des points de passage utilisés pour le calcul de la trajectoire de référence

$x$ (m)	-1,0	0,0	0,1	0,2	1,0	1,8	1,9	2,0	2,1	2,2	3,0	3,5	4,0	4,5
$y$ (m)	0,0	0,0	0,0	0,0	0,0	-1,0	-1,0	-1,0	-1,0	-1,0	-0,2	-0,2	-0,2	-0,2

Les paramètres de cette simulation sont identiques à ceux présentés à la section 6.3.1.1 à l'exception du nombre de modules et de la trajectoire de référence. Le robot simulé est constitué de 6 modules et la trajectoire de référence est calculée à partir des coordonnées

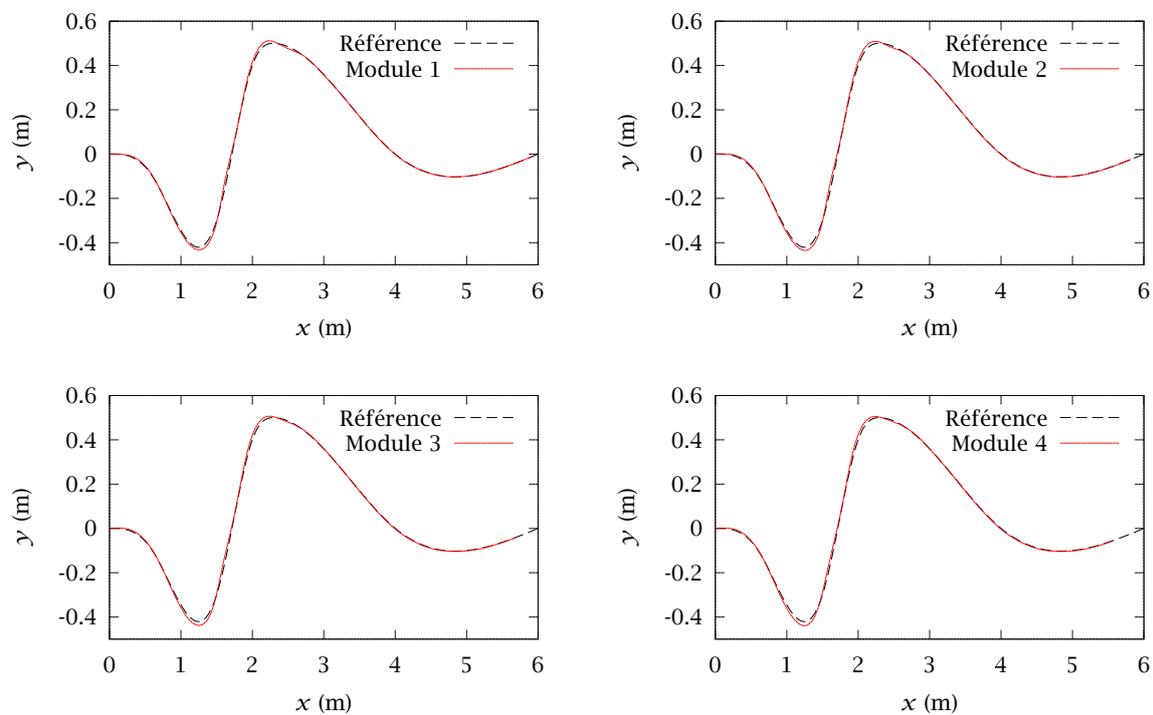


FIGURE 6.5. Trajectoires des 4 modules pour une tâche de suivi de courbe interpolée

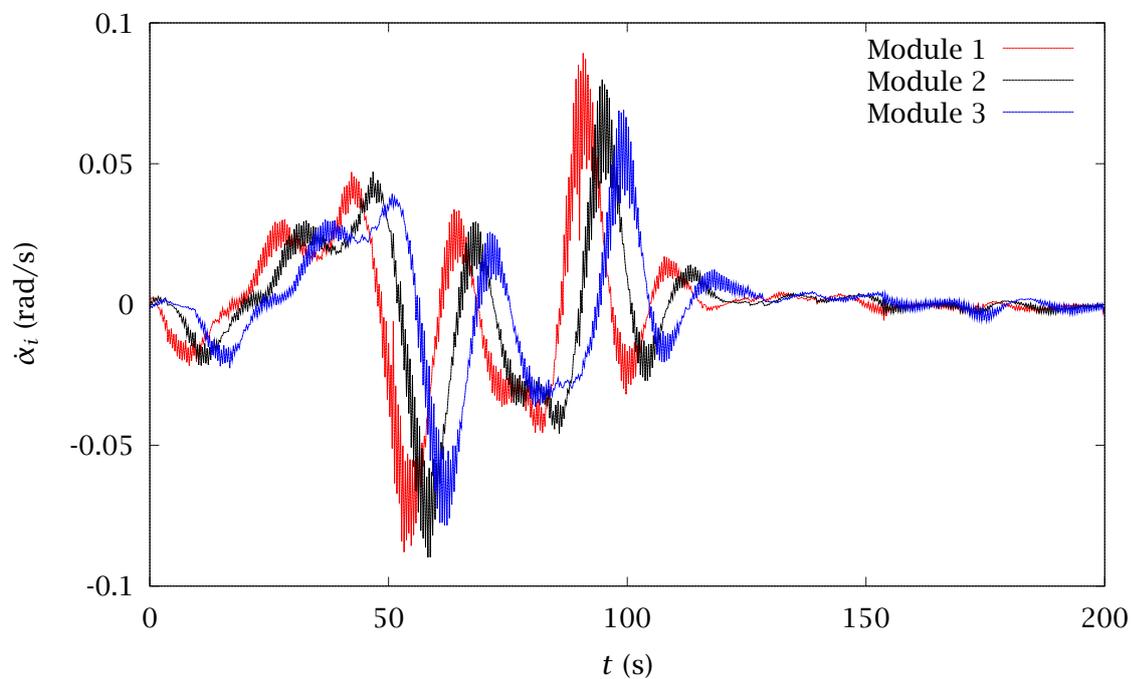


FIGURE 6.6. Évolution des consignes de vitesses articulaires au cours du temps

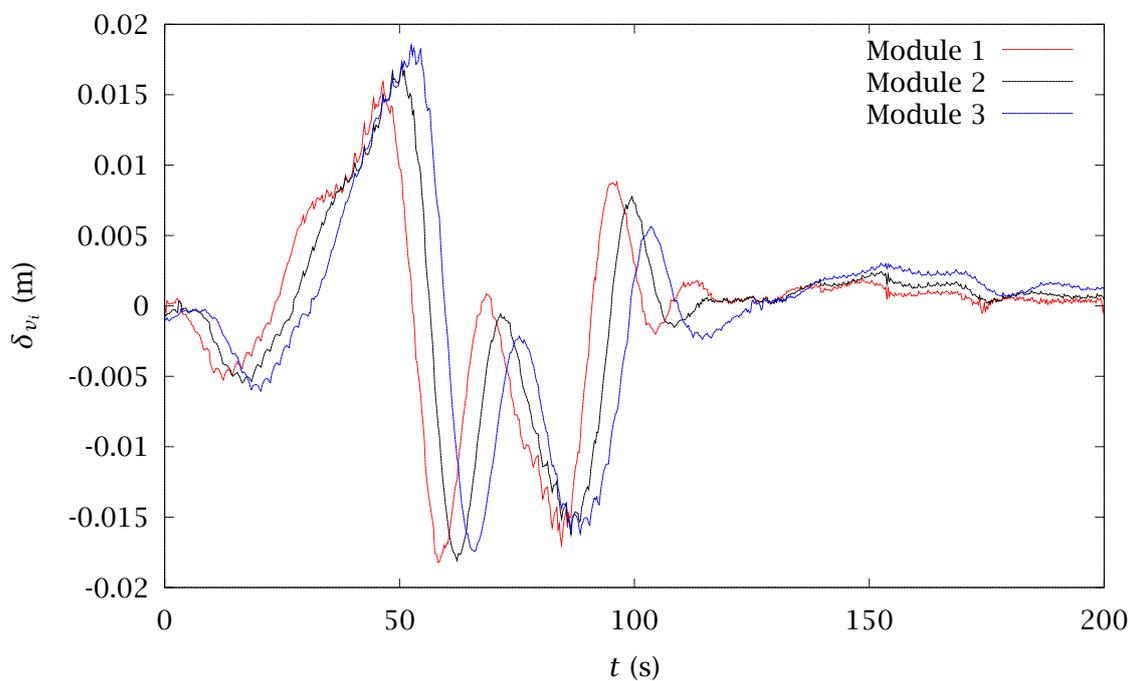


FIGURE 6.7. Évolution de l'erreur en distance  $\delta_{v_i}$  au cours du temps

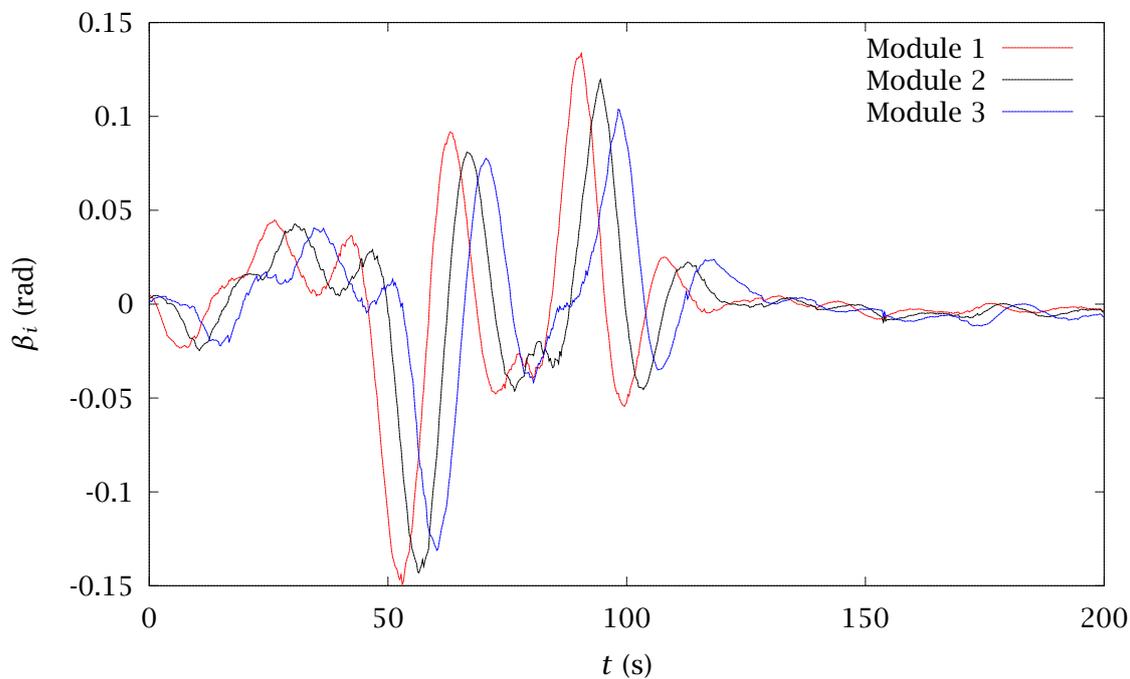


FIGURE 6.8. Évolution de l'erreur en orientation  $\beta_i$  au cours du temps

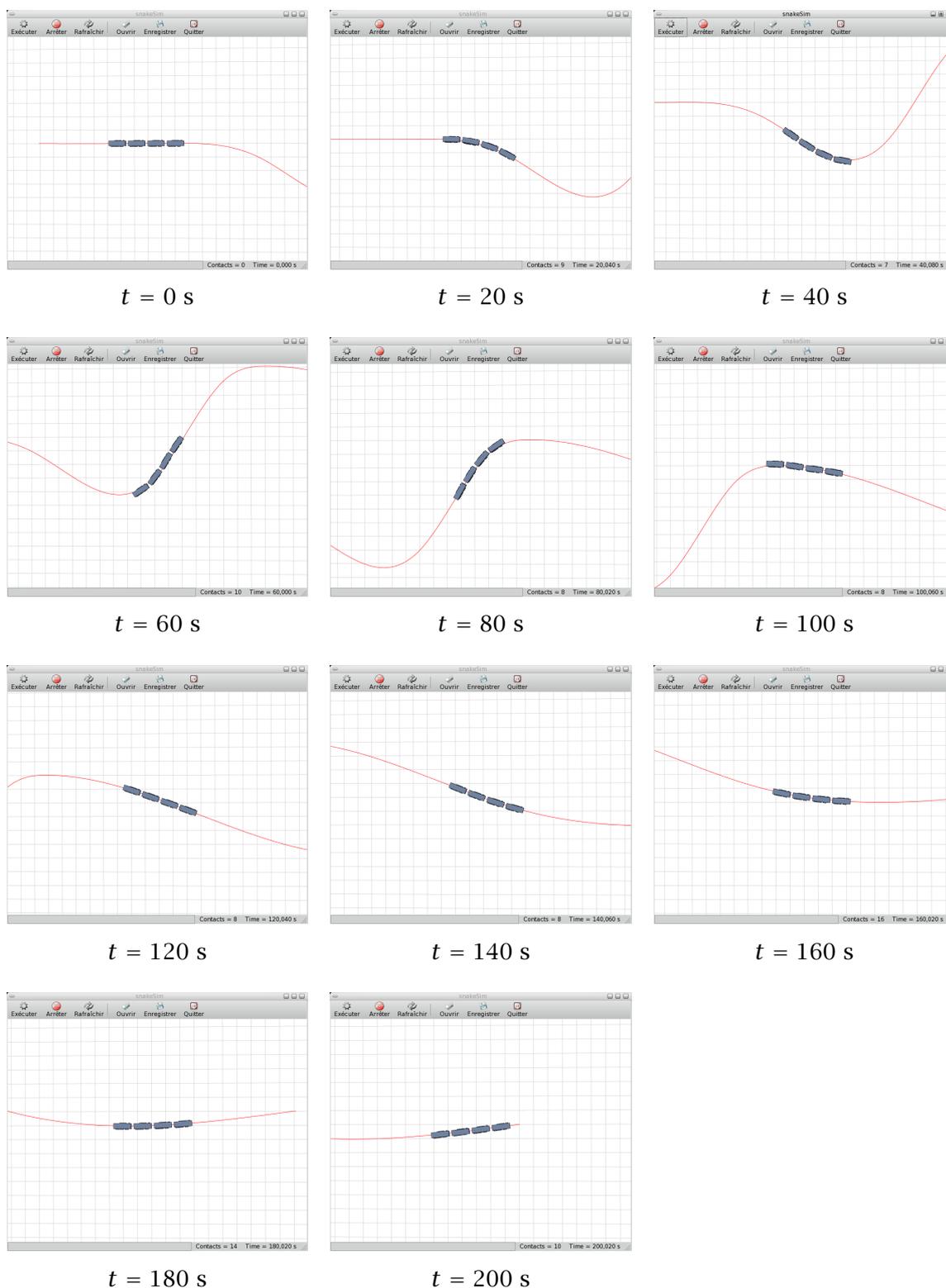


FIGURE 6.9. Déroulement de la simulation d'une tâche de suivi de trajectoire pour un robot à 4 modules

de 14 points de passage. Ces coordonnées sont données dans le tableau 6.2. À l'instant  $t = 0$  s, le robot est situé sur la trajectoire de référence et le module de tête est centré sur l'origine.

### 6.3.2.2 Résultats

Les figures 6.10, 6.11 et 6.12 représentent, respectivement, les trajectoires des 6 modules, l'évolution des erreurs en distance  $\delta_{v_i}$  ainsi que celle des erreurs en orientation  $\beta_i$  pour les modules commandés. Les erreurs ne dépassent pas 4 cm et 0,4 rad (23 °). Cette simulation montre que la commande que nous proposons reste valable lorsque l'on augmente le nombre de modules ainsi que si l'on utilise une trajectoire de référence différente.

### 6.3.3 Suivi de ligne droite sur sol plat avec 4 modules et état initial perturbé

Avec cette simulation, on teste la capacité de l'algorithme de commande à faire converger le robot vers une trajectoire.

#### 6.3.3.1 Description de la tâche

Le robot et la commande sont identiques à celles de la tâche décrite à la section 6.3.1.1. La courbe utilisée est une ligne droite définie par une série de points d'ordonnées identiques  $y = 0,2$  m. À l'instant  $t = 0$  s, le robot est aligné le long de l'axe défini par  $y = 0$ . La commande proposée étant basée sur un calcul d'erreur, on cherche à vérifier que le comportement du robot sera de rejoindre la trajectoire de référence malgré un positionnement initial erroné.

#### 6.3.3.2 Résultats

Les trajectoires des modules sont représentées à la figure 6.13. Les consignes articulaires sont représentées à la figure 6.14. Le comportement du robot est cohérent par rapport au phénomène que l'on souhaite observer. Les modules rejoignent correctement la trajectoire de référence malgré un positionnement initial erroné de la structure. L'évolution au cours du temps des erreurs est représentée par les figures 6.15 et 6.16. Contrairement aux résultats décrits par la section 6.3.1.2, la comparaison des figures 6.15 et 6.14 montre que l'erreur qui prédomine pour l'élaboration de la commande est l'erreur en distance. Cette prédominance est effective tant que le robot n'a pas rejoint la trajectoire de référence.

## 6.4 Conclusion

L'algorithme de commande issu de l'approche géométrique, limité dans son utilisation et ses performances, est présenté dans ce manuscrit en tant que première approche de suivi de trajectoire.

L'algorithme de commande issu de l'approche cinématique donne des résultats satisfaisants avec des couples réalistes pour une tâche de suivi de trajectoire sur sol plat, et ce,

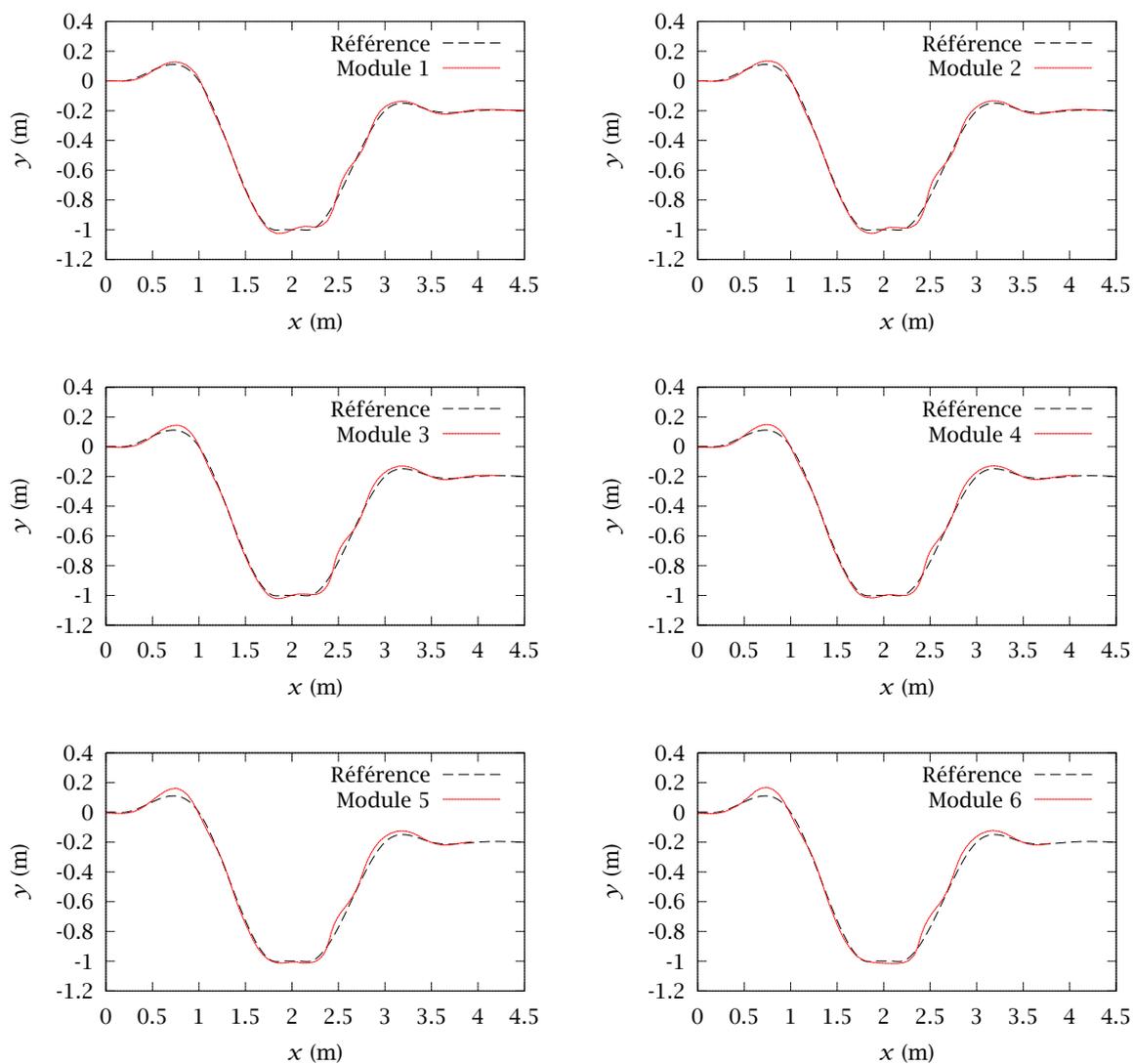
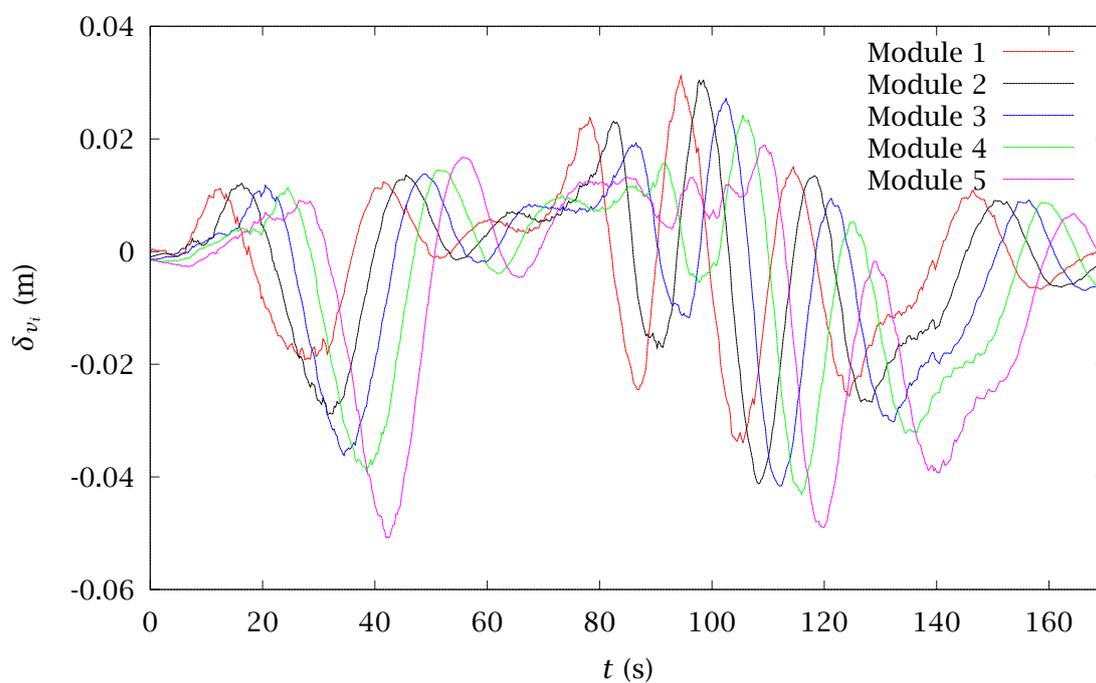
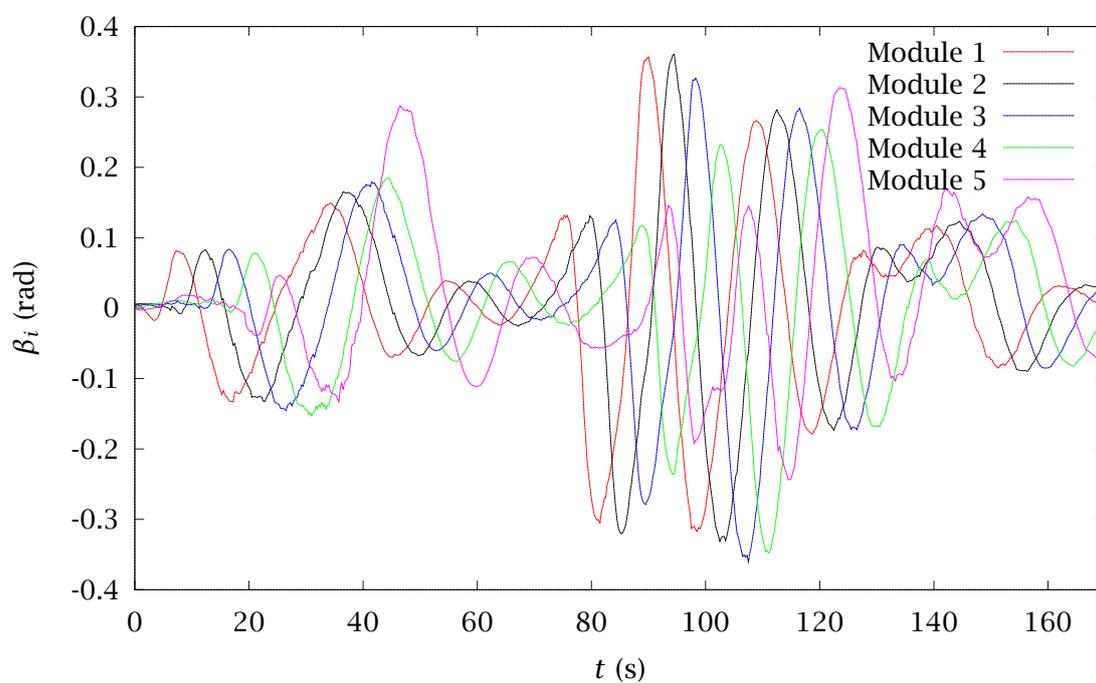


FIGURE 6.10. Trajectoires des 6 modules pour une tâche de suivi de courbe interpolée

FIGURE 6.11. Évolution de l'erreur en distance  $\delta_{v_i}$  au cours du tempsFIGURE 6.12. Évolution de l'erreur en orientation  $\beta_i$  au cours du temps

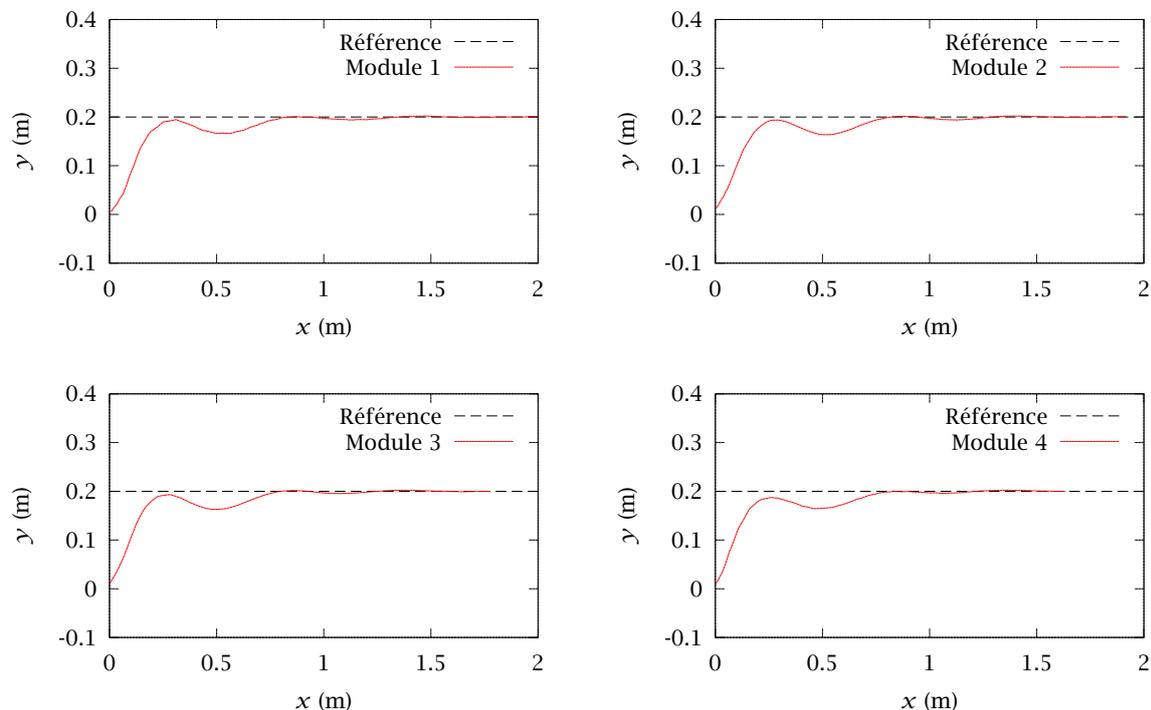


FIGURE 6.13. Trajectoires des 4 modules pour une tâche de suivi de ligne droite (état initial perturbé)

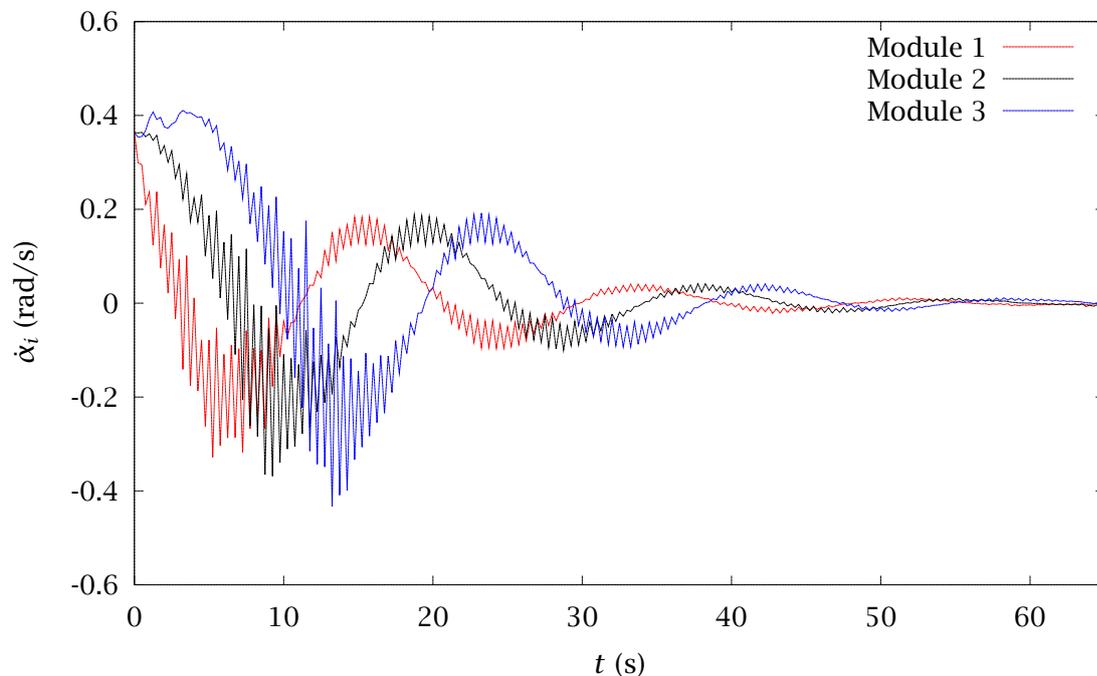


FIGURE 6.14. Évolution des consignes de vitesses articulaires au cours du temps

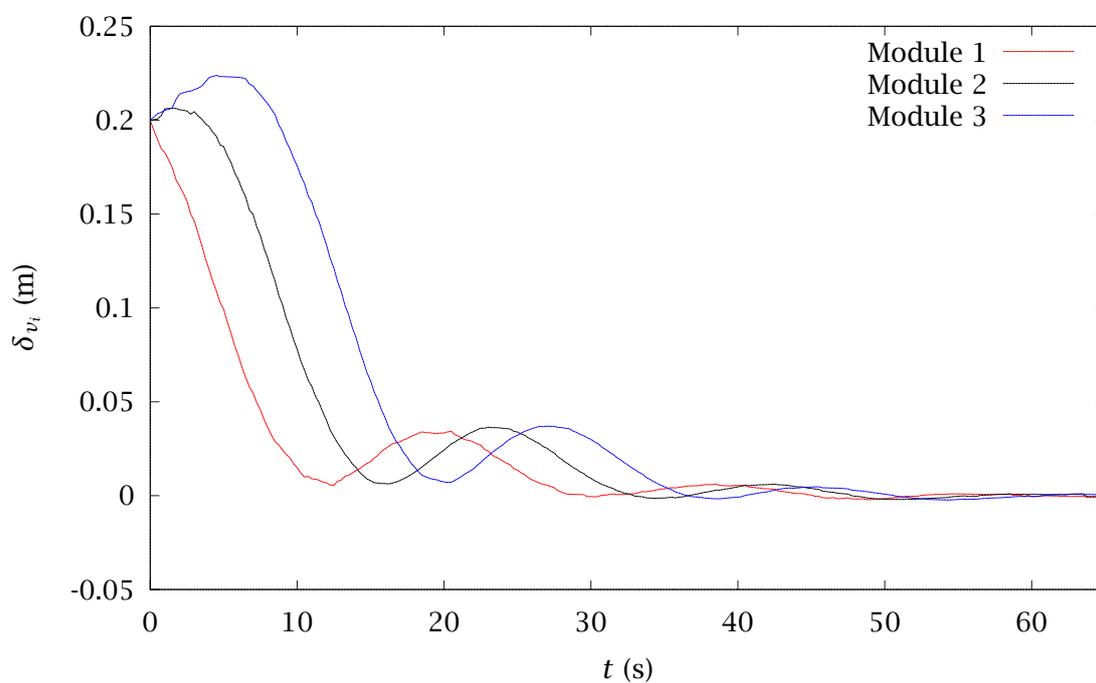


FIGURE 6.15. Évolution de l'erreur en distance  $\delta_{v_i}$  au cours du temps

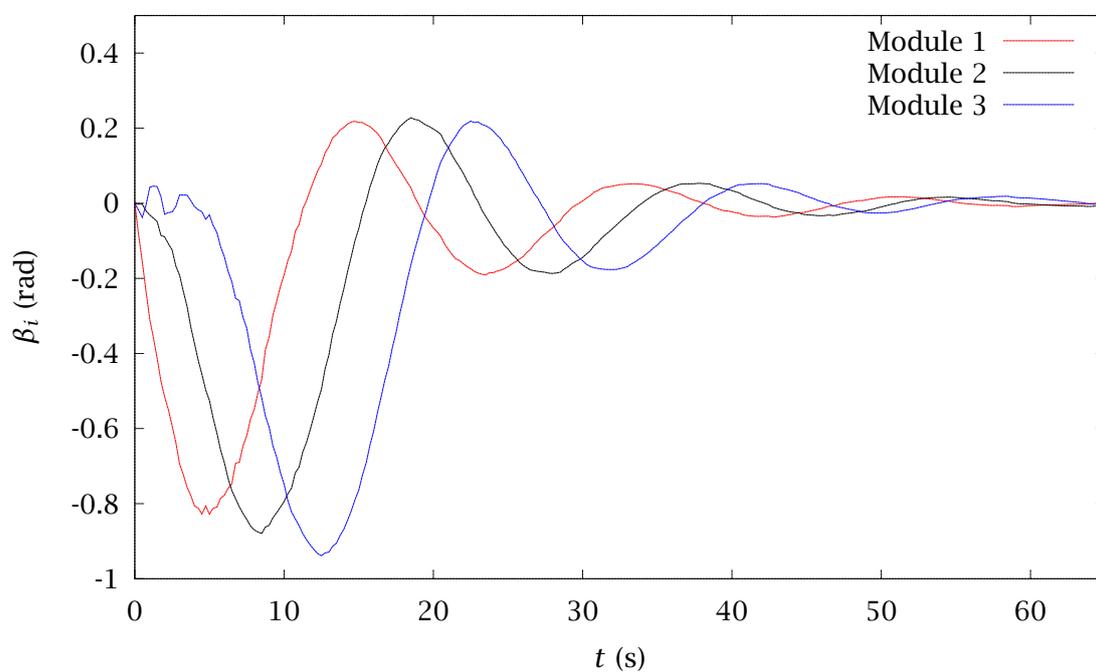


FIGURE 6.16. Évolution de l'erreur en orientation  $\beta_i$  au cours du temps

malgré des hypothèses fausses pour les derniers modules de la structure. Il est toutefois nécessaire de valider ces premiers résultats de commande sur un prototype réel afin d'affiner les paramètres de simulation que nous avons utilisés. Lorsque l'expérimentation aura permis de calibrer notre outil de simulation, d'autres algorithmes de commande pourront être évalués.

À ce stade, nous n'avons pas étudié d'approches dynamiques pour la commande en raison de la complexité de la modélisation des comportements aux points de contact entre les modules et le sol.

# Chapitre 7

## Conclusions et perspectives

---

Sommaire
----------

---

7.1	Conclusions .....	99
7.2	Perspectives .....	100

---

### 7.1 Conclusions

Ce travail a été basé sur l'analyse d'un système existant à partir duquel un ensemble d'outils de simulation robotique ont été développés et ont aboutis à la proposition d'une nouvelle structure robotique.

Ce manuscrit propose tout d'abord l'analyse géométrique et cinématique d'un mécanisme à 5 barres dans le cadre d'une utilisation pour la locomotion d'une structure robotique. Ces modèles ont été établis en premier lieu pour analyser le principe de locomotion reposant sur l'association de plusieurs mécanismes. Ils ont également été utilisés au début du développement de l'environnement de simulation afin de le valider, au moins du point de vue du comportement local du mécanisme. L'obtention des trajectoires et des profils de vitesse de l'organe terminal de plusieurs mécanismes adjacents ont permis d'établir des critères d'optimisation en fonction de la valeur du déphasage interne liant les deux coordonnées généralisées d'un mécanisme.

L'absence d'une plate-forme expérimentale a motivé le développement d'un environnement de simulation. Cet environnement de simulation robotique a été conçu à partir de bibliothèques à code source ouvert en utilisant le langage C++. Le choix du langage s'est imposé de lui-même dans un souci d'inter-opérabilité avec un grand nombre de bibliothèques, de facilité de maintenance et de portabilité. Il met à disposition du chercheur en robotique une palette complète d'outils de calcul scientifique, d'apprentissage et de visualisation. Il fournit des capacités de configuration de simulation sans qu'il soit nécessaire de recompiler le programme. De plus, il permet la sauvegarde et la restitution des données de simulation. Enfin, une option permet d'exécuter à distance le programme de simulation sans interface graphique pour déporter la consommation de ressources de calcul, à la condition que le système d'exploitation de la machine distante soit une distribution Linux.

L'analyse du mécanisme de locomotion et l'exécution de premières simulations lors de la phase de développement de l'environnement de simulation ont abouti à une analyse critique de la structure TDA VIPeRe. Cette analyse critique s'appuie également sur l'étude cinématique d'un robot constitué de 2 modules, à 8 mécanismes, reliés entre eux par une liaison pivot d'axe vertical. Cette analyse critique a permis de démontrer une utilisation non optimale des mécanismes locomoteurs conduisant à des situations de blocage dans le cas

d'une utilisation sur sols complexes ou à l'inefficacité d'un mécanisme devant normalement contribuer à la propulsion de la structure. Afin de palier les problèmes soulevés par cette analyse, nous proposons une nouvelle structure robotique basée sur l'utilisation de modules à 4 mécanismes. Notre proposition repose également sur l'utilisation d'articulations inter-modulaires de type rotule. Les rotations de ces articulations sont commandées ou non selon l'axe considéré. Enfin, le nouveau robot a été étudié dans le souci de le doter de capacités de progression sur un sol rigide présentant de légères variations régulières.

En dernier lieu, la simulation de la nouvelle structure a permis d'élaborer d'un premier algorithme de suivi de trajectoire sur sol plat. Deux approches ont été étudiées. La première approche est une approche géométrique en boucle ouverte dont la portée est limitée par une contrainte très forte. Cette contrainte provient du fait que l'algorithme proposé n'est pas basé sur un calcul d'erreur mais uniquement sur les caractéristiques géométriques de la courbe. En cas de perturbations extérieures, la tâche de suivi de trajectoire n'est plus valable. Les limitations de cette première approche nous ont conduit à proposer un algorithme de commande basé sur une approche cinématique. Bien que reposant également sur des hypothèses simplificatrices fortes, cet algorithme semble donner des résultats satisfaisants. Les trajectoires utilisées dans le cadre de la proposition du second algorithme de commande sont des courbes d'interpolation cubique naturelle calculées à partir de la donnée d'une série de points de passage obligatoire.

Le système étudié est complexe. En effet, sa propulsion est induite par les efforts de réaction du sol sur les mécanismes. Ces efforts résultent des mouvements des mécanismes locomoteurs ainsi que de la nature des interactions avec le substrat. Par conséquent, ils ne sont pas directement commandés et il est difficile d'utiliser l'approche dynamique pour la commande.

Les travaux présentés dans ce manuscrit sont parmi les premiers à utiliser une association de plusieurs mécanismes à barres pour la reproduction du mode de locomotion par progression rectilinéaire.

## **7.2 Perspectives**

Tout au long de nos travaux, nous avons discuté de la pertinence d'équiper notre robot d'une peau synthétique. L'intérêt de l'utilisation d'une peau est double. Dans un premier temps, la peau protégerait la structure des agressions extérieures susceptibles de nuire au fonctionnement correct des mécanismes de locomotion, des actionneurs ainsi que des composants électroniques. Dans un second temps, la modification des propriétés de la peau serait susceptible de modifier celles des interactions avec l'environnement. Ainsi, une étude des propriétés d'adhérence de différents matériaux conduirait à une optimisation partielle de la locomotion en accroissant son efficacité. Le modèle géométrique donne une première information sur la nature du matériau à employer pour l'élaboration d'une peau synthétique puisqu'il permet de calculer l'élasticité requise pour le fonctionnement d'un robot équipé d'une telle peau. En effet, il apparaît à la lumière de discussions avec des spécialistes du textile de l'ENSAIT de Roubaix qu'il n'existe pas de textile, défini par sa matière ou la nature de son tissage, présentant les capacités d'extensions requises. La solution réside donc dans l'utilisation d'une peau présentant des plis à la manière d'un soufflet.

Le simulateur utilisé pour notre étude dispose de la capacité de mettre en place et d'évaluer des stratégies d'apprentissage pour l'élaboration ou l'optimisation de la commande.

Dans ce manuscrit, nous ne traitons pas le problème de la conception mécanique et électronique du robot que nous proposons. Un travail de conception important reste à faire du point de vue des actionneurs et des capteurs. Par conséquent, la construction d'un prototype viable repose sur une étude préalable de conception.

Les simulations présentées dans ce manuscrit n'ont pas pu être calibrées par des expérimentations. L'utilisation d'un premier prototype simple devrait permettre ce travail nécessaire de calibration dans le but de vérifier expérimentalement la validité des premiers algorithmes de commande proposés dans ce manuscrit.

Enfin, l'environnement de simulation que nous avons développé a été conçu dans un souci de réutilisation pour des applications de robotique variées. Des premières simulations d'un bras manipulateur redondant ainsi que d'un robot de type serpent différent de celui que nous proposons ont été conduites afin de vérifier leur faisabilité.



# Abréviations

ACM	<i>Active Chord Mechanism</i>
CMU	<i>Carnegie Mellon University</i>
DGA	Délégation Générale pour l'Armement
ELROB	<i>European Land-Robot Trial</i>
ENIB	École Nationale d'Ingénieurs de Brest
ENSAIT	École Nationale Supérieure des Arts et Industrie Textiles de Roubaix
ENSMM	École Nationale Supérieure de Mécanique et des Microtechniques de Besançon
EO	Librairie de calcul évolutionnaire <i>Evolving Objects</i>
EPFL	École Polytechnique Fédérale de Lausanne
ETAS	Établissement Technique d'Angers (DGA)
GIMP	<i>The GNU Image Manipulation Program</i>
GSL	Librairie scientifique <i>GNU Scientific Library</i>
GTK+	<i>The GIMP Toolkit</i> , [GTK+, 2006] disposant d'une interface C++ [GTKMM, 2006].
HUT	<i>Helsinki University of Technology</i>
ITÜ	<i>Istanbul Teknik Üniversitesi</i>
KIST	<i>Korean Institute of Science and Technology</i>
LISV	Laboratoire d'Ingénierie des Systèmes de Versailles
LRP6	Laboratoire de Robotique de Paris 6
MNHN	Muséum National d'Histoire Naturelle
ODE	Librairie de simulation de solides rigides <i>Open Dynamics Engine</i>
OpenGL	Librairie graphique 2D/3D <i>OpenGL</i>
OpenMASK	<i>Open Modular Animation and Simulation Kit</i>
POV-Ray	<i>Persistence of Vision Raytracer</i>
PPU	<i>Physics Processing Unit</i> , processeurs dédiés aux calculs physiques
SIAMES	Projet Synthèse d'images, animation, modélisation et simulation
SSR	<i>Slim Slime Robot</i>
TDA	Cet acronyme n'a pas de signification particulière. Il désigne l'entreprise TDA Armements, anciennement Thomson Brandt Armements. Filiale de Thalès à 100% depuis 2005.
UML	Langage graphique de modélisation des données et traitements informatiques <i>Unified Modeling Language</i>
UNSW	<i>University of New South Wales</i>
UVSQ	Université de Versailles-Saint-Quentin-en-Yvelines
VIPeRe	Véhicule Intelligent de Pénétration Par Reptation



# Notations

.	opérateur binaire de produit scalaire
$\times$	opérateur binaire de produit vectoriel
*	opérateur binaire de multiplication
$\ \vec{AB}\ $	norme du vecteur $\vec{AB}$
$\vec{AB}$	vecteur
$\vec{a}$	vecteur d'accélération du point $G$ appartenant à $\Sigma$ relativement à $\mathcal{R}$
$\alpha_i$	angle relatif entre le module $i$ et le module $i+1$ du robot
$\dot{\alpha}_i$	vitesse angulaire entre le module $i$ et le module $i+1$ du robot
$\vec{\alpha}$	vecteur d'accélération angulaire de $\Sigma$ relativement à $\mathcal{R}$
$\beta_i$	erreur de positionnement en orientation du module $i$ relativement à l'orientation de la tangente à la courbe suivie au point le plus proche
$d$	constante géométrique du mécanisme correspondant à la longueur du bras fixe
$\frac{df}{dX}$	dérivée de la fonction $f$ par rapport à la variable $X$
$\delta_{v_i}$	erreur de positionnement en distance du module $i$ relativement au point le plus proche appartenant à la trajectoire suivie
$\Delta_z$	amplitude verticale, critère d'optimisation du mécanisme
$\Delta T$	pas de temps
$\delta_x$	performance locale de progression dans l'axe longitudinal, critère d'optimisation du mécanisme
$\delta_z$	variation d'altitude de la plateforme, critère d'optimisation du mécanisme
$\vec{F}$	résultante vectorielle des efforts extérieurs appliquée au solide $\Sigma$
$f^k$	valeur discrète de la fonction $f$ à l'instant $t = k\Delta T$ , $k \in \mathbb{N}$
$\varphi$	déphasage angulaire interne du mécanisme de locomotion
$\vec{H}$	moment cinétique
$\mathbf{I}$	matrice d'inertie d'un solide $\Sigma$ exprimée dans le repère absolu $\mathcal{R}$
$\mathbf{I}_{\text{corps}}$	matrice d'inertie du solide $\Sigma$ exprimée dans le repère local $\mathcal{R}_{\Sigma}$
$\theta_i$	angle entre le module $i$ et le repère absolu
$K$	courbure
$k, k_{\Delta_z}, k_{\delta_x}, k_{\delta_z}, k_1$ et $k_2$	gains proportionnels
$\vec{L}$	résultante cinétique
$l$	longueur d'un module
$M$	point de l'espace 3D
$\mathbf{M}$	matrice
$\mathbf{M}^{-1}$	inverse de la matrice $\mathbf{M}$
$\mathbf{M}^T$	transposée de la matrice $\mathbf{M}$

$m$	masse
$\omega$	vitesse de rotation en entrée du mécanisme
$\vec{\omega}$	vecteur vitesse de rotation de $\Sigma$ relativement à $\mathcal{R}$
$\vec{\omega}$	représentation matricielle du produit vectoriel construite à partir des composantes du vecteur $\vec{\omega}$
$\vec{p}$	vecteur position d'un point quelconque du solide $\Sigma$ relativement à $\mathcal{R}$
$\vec{p}_1$	vecteur position d'un point quelconque du solide $\Sigma$ relativement à $\mathcal{R}_\Sigma$
$q, q_1, q_2$	coordonnées généralisées d'un mécanisme de locomotion
$R$	constante géométrique du mécanisme correspondant à la longueur des bras entraînés
$\mathbf{R}$	matrice de passage de $\mathcal{R}_\Sigma$ vers $\mathcal{R}$ , la transformation étant une rotation
$\vec{r}$	vecteur position du centre de gravité du solide $\Sigma$ relativement à $\mathcal{R}$
$\rho$	constante géométrique du mécanisme correspondant à la longueur des bras actionnés
$r$	rayon de courbure
$s$	abscisse curviligne
$\vec{\tau}$	résultante vectorielle des couples extérieurs appliqués au solide $\Sigma$ au point $G$
$(\vec{x}, \vec{y}, \vec{z})$	base de vecteurs unitaires
$(\vec{x}_i, \vec{y}_i, \vec{z}_i)$	base locale de vecteurs unitaires attachée au module $i$ du robot
$w$	nombre de degrés de liberté du mécanisme
$G$	centre de gravité
$V_i$	amplitude du vecteur vitesse linéaire du module $i$ projeté sur l'axe longitudinal local $\vec{x}_i$
$\vec{V}_{\text{solide}}^{\text{référence}}$	vecteur vitesse linéaire du centre de gravité d'un solide par rapport à un repère de référence
$\vec{v}$	vecteur vitesse du point $G$ relativement à $\mathcal{R}$

# Références bibliographiques

- [ADAMS, 2006] Page web d'Adams. [En ligne]. Disponible : <http://www.mscsoftware.com/products/adams.cfm>
- [ANDRADE-BARROSO, 2000] Guillermo E. ANDRADE-BARROSO - *Modélisation et adaptation du mouvement de robots tout-terrain* - Thèse de doctorat, Université Pierre et Marie Curie, Paris 6, 2000.
- [AOKI *et al.* , 2004] Takeshi AOKI, Akiyoshi OCHIAI et Shigeo HIROSE - Study on slime robot. development of the mobile robot prototype model using bridle bellows - Dans *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2808-2813, 2004.
- [ARAI *et al.* , 2004] Masayuki ARAI, Toshio TAKAYAMA et Shigeo HIROSE - Development of "Souryu-III": Connected crawler vehicle for inspection inside narrow and winding spaces - Dans *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 52-57, 2004.
- [ARTOBOLEVSKI, 1977] I. ARTOBOLEVSKI - *Théorie des machines et des mécanismes* - Mir , 1977.
- [AUBIN *et al.* , 2006] Renaud AUBIN, Pierre BLAZEVIC, Benoît CLÉMENT et Jean-Paul GUYVARCH - Simulation and Design of a Snake-Like Robot Based on a Bio-Inspired Mechanism - Dans *Proc. of the IEEE/RAS-EMBS Int. Conf. on Biomedical Robotics and Biomechatronics (BioRob)*, 2006.
- [AUBIN *et al.* , 2005] Renaud AUBIN, Pierre BLAZEVIC et Jean-Paul GUYVARCH - Simulation of a Novel Snake-Like Robot - Dans *Proc. of CLAWAR 2005: Eight International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, pages 875-882, 2005. Springer.
- [BAYRAKTAROĞLU, 2002] Zeki Yağız BAYRAKTAROĞLU - *Locomotion apode: application à la robotique mobile avec un mécanisme de type serpent* - Thèse de doctorat, Université de Versailles Saint-Quentin-en-Yvelines, 2002.
- [BAYRAKTAROĞLU et BLAZEVIC, 2004] Zeki Yağız BAYRAKTAROĞLU et Pierre BLAZEVIC - A controllability criterion based on a physical analogy : limbless locomotion and object grasping in robotics - *Robotica*, 22 :493-503, 2004.
- [BDI, 2006] Page web du robot BigDog (BostonDynamics). [En ligne]. Disponible : <http://www.bdi.com/content/sec.php?section=BigDog>
- [BECK et ANDRES, 2004] Kent BECK et Cynthia ANDRES - *Extreme Programming Explained, Embrace Change* - Addison-Wesley, 2<sup>e</sup> édition , 2004.
- [BLENDER, 2006] Page web de Blender. [En ligne]. Disponible : <http://blender.org/>
- [BOGERT, 1947] Charles M. BOGERT - Rectilinear locomotion in snakes - *Copeia*, 4 :253-254, 1947.
- [BOYER *et al.* , 2006] Frédéric BOYER, Mazen ALAMIR, Damien CHABLAT, Wisama KHALIL, Alban LEROYER et Philippe LEMOINE - Robot anguille sous-marin en 3D - Dans *Techniques de l'Ingénieur*, dossier S 7 856 de Informatique Industrielle. Techniques de l'Ingénieur, 2006.

- [BULLET, 2006] Site web de la bibliothèque de simulation physique Bullet. [En ligne]. Disponible : <http://www.continuousphysics.com/Bullet/>
- [CHOCRON, 2000] Olivier CHOCRON - *Conception évolutionnaire de systèmes robotiques* - Thèse de doctorat, Université Pierre et Marie Curie, Paris 6, 2000.
- [COUTINHO, 2001] Murilo G. COUTINHO - *Dynamic Simulation of Multibody Systems* - Springer , 2001.
- [CRESPI *et al.* , 2005a] Alessandro CRESPI, André BADERTSCHER, André GUIGNARD et Auke JAN IJSPERT - Amphibot I: an amphibious snake-like robot - *Robotics and Autonomous Systems*, 50(4) :163-175, 2005.
- [CRESPI *et al.* , 2005b] Alessandro CRESPI, André BADERTSCHER, André GUIGNARD et Auke JAN IJSPERT - Swimming and crawling with an amphibious snake robot - Dans *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 3035-3039, 2005.
- [DATE *et al.* , 2001b] Hisashi DATE, Mitsuji SAMPEI et Shigeki NAKAURA - Control of a snake robot in consideration of constraint force - Dans *Proc. of the IEEE Int. Conf. on Control Applications*, pages 966-971, Mexico, 2001.
- [DOWLING, 1997] Kevin J. DOWLING - *Limbless Locomotion: Learning to Crawl with a Snake Robot* - Thèse de doctorat, Carnegie Mellon University, 1997.
- [DOWLING, 1999] Kevin J. DOWLING - Limbless locomotion: Learning to crawl - Dans *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 3001-3006, 1999.
- [ĐURIKOVIČ et NUMATA, 2004] Roman ĐURIKOVIČ et Katsuhiko NUMATA - Human hand model based on rigid body dynamics - Dans *Proc. of the IEEE Conf. on Information Visualization*, pages 853-857, London, UK, 2004.
- [ELROB, 2006] Page web de ELROB. [En ligne]. Disponible : <http://www.elrob.org/>
- [EO, 2006] Page web de la bibliothèque Evolving Objects. [En ligne]. Disponible : <http://eo-dev.sourceforge.net>
- [ERKMEN *et al.* , 2002] Ismet ERKMEN, Aydan M. ERKMEN, Fumitoshi MATSUNO, Ranajit CHATTERJEE et Tetsushi KAMEGAWA - Snake robots to the rescue! - *IEEE Robot. Automat. Mag.*, pages 17-25, 2002.
- [ERLEBEN, 2004] Kenny ERLEBEN - *Stable, Robust, and Versatile Multibody Dynamics Animation* - Thèse de doctorat, Université de Copenhague, 2004.
- [ESTIER *et al.* , 2000] T. ESTIER, Y. CRAUSAZ, B. MERMINOD, M. LAURIA, R. PIGUET et R. SIEGWART - An innovative space rover with extended climbing abilities - Dans *Proc. of Robotics 2000*, 2000.
- [EVOLVINGROBOTS, 2006] Page web de Evolving Robots. [En ligne]. Disponible : <http://www.erachampion.com/ai/>
- [FUJITA, 2001] Masahiro FUJITA - AIBO: Toward the Era of Digital Creatures - *The International Journal of Robotics Research*, 20(10) :781-794, 2001.
- [GANS, 1980] Carl GANS - *Biomechanics An Approach to Vertebrate Biology* - The University of Michigan Press, 2<sup>e</sup> édition , 1980.
- [GASC, 1974] Jean-Pierre GASC - *L'interprétation fonctionnelle de l'appareil musculo-squelettique de l'axe vertébral chez les serpents* -, volume 83 de *A Mémoires du Muséum National d'Histoire Naturelle*, Paris , 1974.

- 
- [GAZEBO, 2006] Page web de Gazebo. [En ligne]. Disponible : <http://playerstage.sourceforge.net/gazebo/gazebo.html>
- [GIMP, 2006] Page web de GIMP. [En ligne]. Disponible : <http://www.gimp.org/>
- [GRAND *et al.* , 2005] Christophe GRAND, Faïz BEN AMAR, Frédéric PLUMET et Philippe BIDAUD - Évaluation des modes de déplacement d'un robot à locomotion hybride roue-patte évoluant en milieux naturels - *Mécanique & Industries*, 6 :353-364, 2005.
- [GRANOSIK *et al.* , 2005] Grzegorz GRANOSIK, Malik G. HANSEN et Johann BORENSTEIN - The Omnitread serpentine robot for industrial inspection and surveillance - *International Journal on Industrial Robots, Special Issue on Mobile Robots*, IR32-2 :139-148, 2005.
- [GRAY, 1968] J. GRAY - *Animal Locomotion* - W.W. Norton Co. , 1968.
- [GTK+, 2006] Page web de GTK+. [En ligne]. Disponible : <http://www.gtk.org/>
- [GTKMM, 2006] Page web de l'interface C++ pour GTK+. [En ligne]. Disponible : <http://www.gtkmm.org/>
- [GUYVARCH, 1995] Jean-Paul GUYVARCH - Procédé de propulsion d'un véhicule à déplacement par reptation et véhicule pour sa mise en œuvre- 1995.French
- [GUYVARCH, 2001] Jean-Paul GUYVARCH - Véhicule à déplacement par reptation propulsé conjointement par ondulations longitudinales et latérales- 2001.French
- [HASHIMOTO *et al.* , 2002] Minoru HASHIMOTO, Takeshi YAMADA et Masaki YAMAKITA - Control of locomotion and head configuration for 3d snake robot - Dans *Proc. of the Int. SICE Conf.*, pages 1470-1475, 2002.
- [HIROSE, 1993] Shigeo HIROSE - *Biologically Inspired Robots - Snake-Like Locomotors and Manipulators* - Oxford University Press , 1993.
- [HUGEL, 1999] Vincent HUGEL - *Contribution à la commande de robots hexapode et quadrupède* - Thèse de doctorat, Université Pierre et Marie Curie, Paris 6, 1999.
- [ITO *et al.* , 2003] Kazuyuki ITO, Tetsushi KAMEGAWA et Fumitoshi MATSUNO - Extended QDSEGA for controlling real robots - Dans *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 791-796, 2003.
- [ITO et MATSUNO, 2002a] Kazuyuki ITO et Fumitoshi MATSUNO - Control of hyper-redundant robot using QDSEGA - Dans *Proc. of the Int. SICE Conf.*, pages 1499-1504, 2002.
- [KADOUS *et al.* , 2005] Mohammed Waleed KADOUS, Raymond Ka-Man SHEH et Claude SAMMUT - CASTER: A Robot for Urban Search and Rescue - Dans *Proc. of the Australasian Conference on Robotics & Automation (ACRA)*, 2005.
- [KAMEGAWA *et al.* , 2002] Tetsushi KAMEGAWA, Fumitoshi MATSUNO et Ranajit CHATTERJEE - Proposition of twisting mode of locomotion and GA based motion planning for transition of locomotion modes of 3-dimensional snake-like robot - Dans *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1507-1512, 2002.
- [KAMEGAWA *et al.* , 2004] Tetsushi KAMEGAWA, Tatsuhiro YAMASAKI, Hiroki IGARASHI et Fumitoshi MATSUNO - Development of the snake-like rescue robot "KOHGA" - Dans *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 5081-5086, 2004.
- [KEIJZER *et al.* , 2001] Maarten KEIJZER, Juan Julián MERELO, Gustavo ROMERO et Marc SCHOENAUER - Evolving Object: a general purpose evolutionary computation library - Dans *Proc. of the Int. Conf. on Artificial Evolution*, 2001.
-

- [KHALIL et DOMBRE, 1999] Wisama KHALIL et Étienne DOMBRE - *Modélisation, identification et commande des robots* - Hermès Science Publication, 2<sup>e</sup> édition , 1999.
- [KLAASSEN et PAAP, 1999] Bernhard KLAASSEN et Karl L. PAAP - GMD-SNAKE2: A snake-like robot driven by wheels and a method for motion control - Dans *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 3014-3019, 1999.
- [KOHL et STONE, 2004] Nate KOHL et Peter STONE - Policy Gradient Reinforcement Learning for Fast Quadrupedal Locomotion - Dans *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2004.
- [KOLOVSKY *et al.* , 2000] M.Z. KOLOVSKY, A.N. EVGRAFOV, Yu. A. SEMENOV et A.V. SLOUSCH - *Advanced Theory of Mechanisms and Machines* - Springer , 2000.
- [LEE *et al.* , 2005] Woosub LEE, Sungchul KANG, Munsang KIM et Kyungchul SHIN - Rough Terrain Negotiable Mobile Platform with Passively Adaptive Double-Tracks and Its Application to Rescue Missions and EOD Missions - Dans *Proc. of the Int. Conf. on Computer Applications in Shipbuilding (ICCAS)*, 2005.
- [LISSMAN, 1950] H.W. LISSMAN - Rectilinear locomotion in a snake (*Boa occidentalis*) - *Journal of Experimental Biology*, 26 :368-379, 1950.
- [MA, 1999] Shugen MA - Analysis of snake movement forms for realization of snake-like robots - Dans *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 3007-3013, 1999.
- [MA *et al.* , 2001b] Shugen MA, Wen J. LI et Yuechao WANG - A simulator to analyze creeping locomotion of a snake-like robot - Dans *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 3656-3661, 2001.
- [MATSUNO et SATO, 2005] Fumitoshi MATSUNO et Hiroki SATO - Trajectory tracking control of snake robots based on dynamic model - Dans *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 3040-3045, 2005.
- [MATSUNO et SUENAGA, 2002] Fumitoshi MATSUNO et Kentaro SUENAGA - Control of redundant snake robot based on kinematic model - Dans *Proc. of the Int. SICE Conf.*, pages 1482-1486, 2002.
- [MATSUNO et SUENAGA, 2003] Fumitoshi MATSUNO et Kentaro SUENAGA - Control of redundant 3d snake robot based on kinematic model - Dans *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2061-2066, 2003.
- [MICHEL, 2004] Olivier MICHEL - Cyberbotics Ltd. Webots: Professional Mobile Robot Simulation - *Int. Journal of Advanced Robotics Systems*, 1(1) :39-42, 2004.
- [MILLER, 1988] Gavin S. P. MILLER - The motion dynamics of snakes and worms - *Computer Graphics*, 22(4) :169-178, 1988.
- [MRS, 2006] Page web du Microsoft Robotics Studio. [En ligne]. Disponible : <http://msdn.microsoft.com/robotics/>
- [NEWTON, 2006] Site web de la bibliothèque de simulation physique Newton Dynamics. [En ligne]. Disponible : <http://www.newtondynamics.com/>
- [ODE, 2006] Site web de la bibliothèque de simulation physique open dynamics engine. [En ligne]. Disponible : <http://ode.org/>
- [OPENMASK, 2006] Page web de OpenMASK (Projet SIAMES). [En ligne]. Disponible : <http://www.irisa.fr/siames/softwares-fr/openmask>

- 
- [OPENSIM, 2006] Page web de OpenSim. [En ligne]. Disponible : <http://opensimulator.sourceforge.net/>
- [OPENTISSUE, 2006] Page web de la bibliothèque OpenTissue. [En ligne]. Disponible : <http://www.opentissue.org/>
- [PAAP *et al.* , 2000] K. L. PAAP, T. CHRISTALLER et F. KIRCHNER – A robot snake to inspect broken buildings – Dans *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2079-2082, 2000.
- [PACKBOT, 2006] Page web du robot PackBot. [En ligne]. Disponible : <http://www.irobot.com/sp.cfm?pageid=109>
- [PFEIFFER et GLOCKER, 2000] Friedrich PFEIFFER et Christoph GLOCKER – Contacts in multi-body systems – *Journal of Applied Mathematics and Mechanics*, 64(5) :773-782, 2000.
- [PHYSX, 2006] Page web du SDK AGEIA PhysX. [En ligne]. Disponible : <http://www.ageia.com/developers/downloads.html>
- [POV-RAY, 2006] Page web de POV-Ray. [En ligne]. Disponible : <http://www.povray.org>
- [ROBHAZ, 2006] Page web du robot ROBHAZ. [En ligne]. Disponible : <http://www.robhaz.com>
- [ROBOCUPRESCUE, 2006] Page web officielle de la RoboCup Rescue. [En ligne]. Disponible : <http://www.rescuesystem.org/robocuprescue/>
- [ROBUROC6, 2006] Page web du robot RobuROC 6. [En ligne]. Disponible : <http://www.robosoft.fr/robuROC6.html>
- [RÖFER, 2004] Thomas RÖFER – Evolutionary Gait-Optimization Using a Fitness Function Based on Proprioception – Dans *International Workshop on RoboCup*, 2004.
- [SAITO *et al.* , 2002] Masashi SAITO, Masakazu FUKAYA et Tetsuya IWASAKI – Serpentine locomotion with robotic snakes – *IEEE Control Syst. Mag.*, pages 64-81, 2002.
- [SATO *et al.* , 2004] Noritaka SATO, Fumitoshi MATSUNO, Tatsuhiko YAMASAKI, Tetsushi KAMEGAWA, Naoji SHIROMA et Hiroki IGARASHI – Cooperative task execution by a multiple robot team and its operators in search and rescue operations – Dans *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1083-1088, 2004.
- [SECOR *et al.* , 1992] Stephen M. SECOR, Bruce C. JAYNE et Albert F. BENNETT – Locomotor performance and energetic cost of sidewinding by the snake *crotalus cerastes* – *J. exp. Biol.*, 163 :1-14, 1992.
- [SHRIMP, 2006] Page web du robot Shrimp III. [En ligne]. Disponible : <http://www.bluebotics.com/solutions/Shrimp/>
- [SIMMECHANICS, 2006] Page web de SimMechanics. [En ligne]. Disponible : <http://www.mathworks.com/products/simmechanics/index.html>
- [SIMULATORBOB, 2006] Page web de Simulator Bob . [En ligne]. Disponible : <http://simbob.sourceforge.net/>
- [SMITH, 1998] Russell L. SMITH – *Intelligent Motion Control with an Artificial Cerebellum* – Thèse de doctorat, Université d'Auckland, Nouvelle-Zélande, 1998.
- [SMITH, 2006] Russell SMITH – Dynamics Simulation, A Whirlwind Tour – Présentation PARC (décembre 2004). [En ligne]. Disponible : <http://ode.org/slides/parc/index.html>
-

- [TAKAYAMA et HIROSE, 2000] Toshio TAKAYAMA et Shigeo HIROSE - Development of souryu-I connected crawler vehicle for inspection of narrow and winding space - Dans *Proc. of the IEEE Annual Conf. of the Industrial Electronics Society*, pages 143-148, 2000.
- [TDA, 2006] Page web de l'entreprise TDA Armements. [En ligne]. Disponible : <http://www.tda-arm.com/>
- [TERRAFORM, 2006] Page web de Terraform. [En ligne]. Disponible : <http://terraform.sourceforge.net/>
- [TOKAMAK, 2006] Site web de la bibliothèque de simulation physique Tokamak. [En ligne]. Disponible : <http://www.tokamakphysics.com/>
- [VIDAL et HEDGES, 2004] Nicolas VIDAL et S. Blair HEDGES - Molecular evidence of terrestrial origin of snakes - *Proc. R. Soc. Lond. B (Suppl.)*, 271 :S226-S229, 2004.
- [VIDAL et HEDGES, 2005] Nicolas VIDAL et S. Blair HEDGES - The phylogeny of squamate reptiles (lizards, snakes, and amphisbaenians) inferred from nine nuclear protein-coding genes - *C. R. Biologies*, 328 :1000-1008, 2005.
- [VORTEX, 2006] Page web de Vortex. [En ligne]. Disponible : <http://www.cm-labs.com/products/vortex/>
- [WALTON *et al.* , 1990] Michael WALTON, Bruce C. JAYNE et Albert F. BENNETT - The energetic cost of limbless locomotion - *Science*, 249 :524-527, 1990.
- [WEBOTS, 2006] Page web de Webots. [En ligne]. Disponible : <http://www.cyberbotics.com/products/webots/>
- [YAMADA *et al.* , 2003] Takeshi YAMADA, Kenta TANAKA et Masaki YAMAKITA - Winding and task control of snake like robot - Dans *Proc. of the Int. SICE Conf.*, pages 2138-2142, 2003.
- [YAMAKITA *et al.* , 2003] Masaki YAMAKITA, Minoru HASHIMOTO et Takeshi YAMADA - Control of locomotion and head configuration of 3D snake robot (SMA) - Dans *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2055-2060, 2003.
- [YIM *et al.* , 2002] Mark YIM, Ying ZHANG et David DUFF - Modular robots - *IEEE Spectr.*, pages 30-34, 2002.
- [YLÖNEN et HALME, 2002] Sami J. YLÖNEN et Aarne J. HALME - WorkPartner - Centaur Like Service Robot - Dans *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2002.

# Annexe A

## Simulation dynamique de solides indéformables avec *Open Dynamics Engine* (ODE)

---

Sommaire

---

A.1	Équations de mouvement .....	113
A.2	Approximation numérique .....	117
A.3	Utilisation de contraintes en vitesse pour la résolution .....	118
A.3.1	Contraintes en vitesse entre deux solides .....	118
A.3.2	Exemple de contrainte : liaison rotule .....	119
A.3.3	Formulation globale des équations de contraintes .....	120
A.4	Contacts .....	121

---

[COUTINHO, 2001 et ERLEBEN, 2004] donnent une description assez complète des mécanismes internes et des bases théoriques utiles à la simulation des solides indéformables. [DŽURIKOVIČ et NUMATA, 2004] décrivent plus particulièrement les mécanismes propres à ODE. Nous présentons ici une partie de ces mécanismes afin de préciser le fonctionnement d'ODE.

### A.1 Équations de mouvement

Soit un solide rigide  $\Sigma$ , dont les propriétés de masse sont connues, de centre de gravité  $G$ , en mouvement par rapport à un repère  $\mathcal{R} (O, \vec{x}, \vec{y}, \vec{z})$ . Soit  $\mathcal{R}_\Sigma (G, \vec{x}_\Sigma, \vec{y}_\Sigma, \vec{z}_\Sigma)$  le repère local associé à ce solide. La base du repère  $\mathcal{R}_\Sigma$  est définie par rapport à la base du repère  $\mathcal{R}$  par la matrice de rotation, dépendante du temps,  $\mathbf{R}(t)$ . Dans le but d'alléger les notations, le tenseur d'inertie du solide en son centre de gravité sera noté  $\mathbf{I}_{\text{corps}}$  ou  $\mathbf{I}(t)$  suivant qu'il sera exprimé, respectivement, dans la base associée à  $\mathcal{R}_\Sigma$  ou dans celle associée à  $\mathcal{R}$ . La relation reliant ces tenseurs s'écrit

$$\mathbf{I}(t) = \mathbf{R}(t) \mathbf{I}_{\text{corps}} \mathbf{R}^T(t). \tag{A.1}$$

Soit  $\vec{p}_1$ , la position d'un point donné du solide exprimée dans le repère  $\mathcal{R}_\Sigma$ , l'expression de la position de ce point par rapport à  $\mathcal{R}$  sera définie par

$$\vec{p}(t) = \mathbf{R}(t) \vec{p}_1 + \vec{r}(t) \tag{A.2}$$

avec  $\vec{r}(t)$ , la position du centre de gravité  $G$  par rapport à  $\mathcal{R}$  à l'instant  $t$ . Le vecteur  $\vec{p}_1$  étant constant par rapport au temps, la vitesse du point défini par  $\vec{p}(t)$  est exprimée par

$$\frac{d\vec{p}(t)}{dt} = \frac{d\mathbf{R}(t)}{dt} \vec{p}_1 + \frac{d\vec{r}(t)}{dt}. \quad (\text{A.3})$$

Soit  $\vec{v}(t)$  le vecteur vitesse linéaire de  $G$  par rapport à  $\mathcal{R}$ , alors

$$\frac{d\vec{r}(t)}{dt} = \vec{v}(t). \quad (\text{A.4})$$

De la même manière, soit  $\vec{\omega}(t)$  le vecteur vitesse de rotation de  $\Sigma$  par rapport à  $\mathcal{R}$ . La dérivée temporelle de la matrice de rotation est donnée par

$$\frac{d\mathbf{R}(t)}{dt} = \vec{\omega} \times \mathbf{R}(t) = \tilde{\omega}(t) \mathbf{R}(t) \quad (\text{A.5})$$

avec  $\tilde{\omega}(t)$ , la représentation matricielle du produit vectoriel telle que

$$\text{si } \vec{\omega}(t) = \begin{pmatrix} \omega_x(t) \\ \omega_y(t) \\ \omega_z(t) \end{pmatrix} \text{ alors } \tilde{\omega}(t) = \begin{pmatrix} 0 & -\omega_z(t) & \omega_y(t) \\ \omega_z(t) & 0 & -\omega_x(t) \\ -\omega_y(t) & \omega_x(t) & 0 \end{pmatrix}. \quad (\text{A.6})$$

En remplaçant les équations (A.4) et (A.5) dans (A.3), on obtient

$$\frac{d\vec{p}(t)}{dt} = \tilde{\omega}(t) \mathbf{R}(t) \vec{p}_1 + \vec{v}(t).$$

En utilisant l'équation (A.2), on obtient

$$\begin{aligned} \frac{d\vec{p}}{dt} &= \tilde{\omega}(t) \mathbf{R}(t) \mathbf{R}(t)^{-1} (\vec{p}(t) - \vec{r}(t)) + \vec{v} \\ &= \tilde{\omega}(t) (\vec{p}(t) - \vec{r}(t)) + \vec{v}(t) \\ &= \vec{\omega}(t) \times (\vec{p}(t) - \vec{r}(t)) + \vec{v}(t). \end{aligned} \quad (\text{A.7})$$

L'équation (A.7) nous donne l'expression de la vitesse linéaire d'un point quelconque appartenant au solide  $\Sigma$ . On obtient l'accélération par dérivation temporelle

$$\frac{d^2\vec{p}(t)}{dt^2} = \frac{d\vec{\omega}(t)}{dt} \times (\vec{p}(t) - \vec{r}(t)) + \vec{\omega}(t) \times \left( \frac{d\vec{p}(t)}{dt} - \frac{d\vec{r}(t)}{dt} \right) + \frac{d\vec{v}(t)}{dt}. \quad (\text{A.8})$$

Soit  $\vec{a}(t)$  l'accélération linéaire du centre d'inertie de  $\Sigma$

$$\vec{a}(t) = \frac{d\vec{v}(t)}{dt}$$

et  $\vec{\alpha}(t)$  l'accélération angulaire de  $\Sigma$  définie par

$$\vec{\alpha}(t) = \frac{d\vec{\omega}(t)}{dt}.$$

Ces accélérations sont exprimées relativement au repère  $\mathcal{R}$ . On obtient alors

$$\frac{d^2\vec{p}(t)}{dt^2} = \vec{\alpha}(t) \times (\vec{p}(t) - \vec{r}(t)) + \vec{\omega}(t) \times (\vec{\omega}(t) \times (\vec{p}(t) - \vec{r}(t))) + \vec{a}(t). \quad (\text{A.9})$$

Soit  $\vec{F}(t)$  la résultante des efforts extérieurs appliqués au solide  $\Sigma$  en son centre d'inertie à l'instant  $t$ . En utilisant le théorème de la quantité de mouvement, on a

$$\vec{F}(t) = \frac{d\vec{L}(t)}{dt}. \quad (\text{A.10})$$

où  $\vec{L}(t)$  est la résultante cinétique du solide calculée à partir de l'équation

$$\vec{F}(t) = \frac{d m \vec{v}(t)}{dt} = m \frac{d \vec{v}(t)}{dt} = m \vec{a}(t) \quad (\text{A.11})$$

avec  $m$  et  $\vec{v}(t)$  la masse du solide et la vitesse linéaire de son centre d'inertie.

De la même manière, on peut exprimer  $\vec{\tau}(t)$ , le couple résultant appliqué au centre d'inertie du solide :

$$\vec{\tau}(t) = \frac{d \vec{H}(t)}{dt}, \quad (\text{A.12})$$

où  $\vec{H}(t)$  est le moment cinétique du solide en son centre d'inertie obtenu par l'expression

$$\vec{H}(t) = \mathbf{I}(t) \vec{\omega}(t) \quad (\text{A.13})$$

avec  $\mathbf{I}(t)$  le tenseur d'inertie du solide tel que défini par l'équation (A.1).

L'accélération angulaire est obtenue par dérivation de la vitesse angulaire. L'équation (A.13) reliant cette vitesse angulaire au moment cinétique, la relation entre l'accélération angulaire et le moment cinétique est

$$\vec{\alpha}(t) = \frac{d \vec{\omega}(t)}{dt} = \frac{d \mathbf{I}^{-1}(t) \vec{H}(t)}{dt} = \frac{d \mathbf{I}^{-1}(t)}{dt} \vec{H}(t) + \mathbf{I}^{-1}(t) \frac{d \vec{H}(t)}{dt}. \quad (\text{A.14})$$

L'équation (A.12) donne alors

$$\vec{\alpha}(t) = \frac{d \mathbf{I}^{-1}(t)}{dt} \vec{H}(t) + \mathbf{I}^{-1}(t) \vec{\tau}(t). \quad (\text{A.15})$$

L'inverse du tenseur d'inertie est obtenue à partir de l'équation (A.1)

$$\mathbf{I}^{-1}(t) = \mathbf{R}(t) \mathbf{I}_{\text{corps}}^{-1} \mathbf{R}^T(t). \quad (\text{A.16})$$

La dérivée par rapport au temps de l'inverse du tenseur d'inertie est alors

$$\begin{aligned} \frac{d \mathbf{I}^{-1}(t)}{dt} &= \frac{d \mathbf{R}(t)}{dt} \mathbf{I}_{\text{corps}}^{-1} \mathbf{R}^T(t) + \mathbf{R}(t) \mathbf{I}_{\text{corps}}^{-1} \frac{d \mathbf{R}^T(t)}{dt} \\ &= \vec{\omega}(t) \mathbf{R}(t) \mathbf{I}_{\text{corps}}^{-1} \mathbf{R}^T(t) + \mathbf{R}(t) \mathbf{I}_{\text{corps}}^{-1} (\vec{\omega}(t) \mathbf{R}(t))^T \\ &= \vec{\omega}(t) \mathbf{R}(t) \mathbf{I}_{\text{corps}}^{-1} \mathbf{R}^T(t) + \mathbf{R}(t) \mathbf{I}_{\text{corps}}^{-1} \mathbf{R}^T(t) \vec{\omega}^T(t) \\ &= \vec{\omega}(t) \mathbf{R}(t) \mathbf{I}_{\text{corps}}^{-1} \mathbf{R}^T(t) - \mathbf{R}(t) \mathbf{I}_{\text{corps}}^{-1} \mathbf{R}^T(t) \vec{\omega}(t) \end{aligned} \quad (\text{A.17})$$

puisque  $\vec{\omega}^t(t) = -\vec{\omega}(t)$ . En utilisant l'équation (A.1), on simplifie l'expression précédente :

$$\begin{aligned} \frac{d \mathbf{I}^{-1}(t)}{dt} &= \vec{\omega}(t) \mathbf{R}(t) \mathbf{I}_{\text{corps}}^{-1} \mathbf{R}^T(t) - \mathbf{R}(t) \mathbf{I}_{\text{corps}}^{-1} \mathbf{R}^T(t) \vec{\omega}(t) \\ &= \vec{\omega}(t) \mathbf{I}^{-1}(t) - \mathbf{I}^{-1}(t) \vec{\omega}(t) \end{aligned} \quad (\text{A.18})$$

En substituant l'équation (A.18) dans (A.15), nous obtenons

$$\begin{aligned}
 \vec{\alpha}(t) &= \left( \vec{\omega}(t) \mathbf{I}^{-1}(t) - \mathbf{I}^{-1}(t) \vec{\omega}(t) \right) \vec{H}(t) + \mathbf{I}^{-1}(t) \vec{\tau}(t) \\
 &= \vec{\omega}(t) \overbrace{\mathbf{I}^{-1}(t) \vec{H}(t)}^{\vec{\omega}(t)} - \mathbf{I}^{-1}(t) \vec{\omega}(t) \vec{H}(t) + \mathbf{I}^{-1}(t) \vec{\tau}(t) \\
 &= \overbrace{\vec{\omega}(t) \vec{\omega}(t)}^{\vec{\omega}(t) \times \vec{\omega}(t) = \vec{0}} - \mathbf{I}^{-1}(t) \vec{\omega}(t) \vec{H}(t) + \mathbf{I}^{-1}(t) \vec{\tau}(t) \\
 &= -\mathbf{I}^{-1}(t) \vec{\omega}(t) \times \vec{H}(t) + \mathbf{I}^{-1}(t) \vec{\tau}(t) \\
 &= \mathbf{I}^{-1}(t) \vec{H} \times \vec{\omega}(t)(t) + \mathbf{I}^{-1}(t) \vec{\tau}(t). \tag{A.19}
 \end{aligned}$$

Ainsi, la relation liant l'accélération angulaire et le moment cinétique s'exprime sous la forme

$$\vec{\alpha}(t) = \mathbf{I}^{-1}(t) \left( \vec{H}(t) \times \vec{\omega}(t) + \vec{\tau}(t) \right), \tag{A.20}$$

ou encore

$$\begin{aligned}
 \vec{\tau}(t) &= \mathbf{I}(t) \vec{\alpha}(t) - \vec{H}(t) \times \vec{\omega}(t) \\
 &= \mathbf{I}(t) \vec{\alpha}(t) + \vec{\omega}(t) \times \vec{H}(t). \tag{A.21}
 \end{aligned}$$

Soit  $\vec{y}(t)$ , l'état dynamique du solide  $\Sigma$  à l'instant  $t$  :

$$\vec{y}(t) = \begin{pmatrix} \vec{r}(t) \\ \mathbf{R}(t) \\ \vec{L}(t) \\ \vec{H}(t) \end{pmatrix},$$

l'état dynamique à l'instant  $t = t_0$  est défini par la position du centre d'inertie  $\vec{r}(t_0)$ , la matrice de rotation  $\mathbf{R}(t_0)$ , la résultante cinétique  $\vec{L}(t_0) = m\vec{v}(t_0)$  et le moment cinétique  $\vec{H}(t_0) = \mathbf{I}(t_0) \vec{\omega}(t_0)$ . La dérivée temporelle du vecteur d'état définit l'évolution au cours du temps de l'état dynamique du solide et est donnée par

$$\frac{d\vec{y}(t)}{dt} = \begin{pmatrix} \vec{v}(t) \\ \vec{\omega}(t) \mathbf{R}(t) \\ \vec{F}(t) \\ \vec{\tau}(t) \end{pmatrix}.$$

Soit la définition suivante de l'état d'un solide, légèrement différente de celle présentée plus haut :

$$\vec{s}(t) = \begin{pmatrix} \vec{r}(t) \\ \mathbf{R}(t) \\ \vec{v}(t) \\ \vec{\omega}(t) \end{pmatrix}.$$

Sa dérivée par rapport au temps est

$$\frac{d\vec{s}(t)}{dt} = \begin{pmatrix} \vec{v}(t) \\ \vec{\omega}(t) \mathbf{R}(t) \\ \frac{\vec{F}(t)}{m} \\ \mathbf{I}^{-1}(t) (\vec{H}(t) \times \vec{\omega}(t) + \vec{\tau}(t)) \end{pmatrix} = \begin{pmatrix} \vec{v}(t) \\ \vec{\omega}(t) \mathbf{R}(t) \\ \frac{\vec{F}(t)}{m} \\ \mathbf{I}^{-1}(t) (\mathbf{I}(t) \vec{\omega}(t) \times \vec{\omega}(t) + \vec{\tau}(t)) \end{pmatrix}.$$

La méthode de résolution présentée dans la suite de ce chapitre est basée sur cette représentation de l'état d'un solide. En effet, la connaissance de  $\vec{F}$  et de  $\vec{\tau}$  permet le calcul de la dérivée du vecteur d'état et, par conséquent, une approximation de l'état du solide au pas de temps suivant.

Considérons maintenant le cas de  $N$  solides, la représentation de l'état des solides peut être mise sous la forme

$$\begin{aligned} \vec{S}_{\text{pos}}(t) &= (\vec{r}_1(t), \mathbf{R}_1(t), \dots, \vec{r}_i(t), \mathbf{R}_i(t), \dots, \vec{r}_N(t), \mathbf{R}_N(t))^T, \\ \vec{S}_{\text{vit}}(t) &= (\vec{v}_1(t), \vec{\omega}_1(t), \dots, \vec{v}_i(t), \vec{\omega}_i(t), \dots, \vec{v}_N(t), \vec{\omega}_N(t))^T. \end{aligned}$$

Les dérivées temporelles correspondantes sont

$$\begin{aligned} \frac{d\vec{S}_{\text{pos}}(t)}{dt} &= (\vec{v}_1(t), \vec{\omega}_1(t) \mathbf{R}_1(t), \dots, \vec{v}_i(t), \vec{\omega}_i(t) \mathbf{R}_i(t), \dots, \vec{v}_N(t), \vec{\omega}_N(t) \mathbf{R}_N(t))^T \\ \text{et } \frac{d\vec{S}_{\text{vit}}(t)}{dt} &= \mathbf{W}(t) \vec{Q}(t) \end{aligned}$$

avec

$$\begin{aligned} \vec{Q}(t) &= (\vec{F}_1(t), \vec{\tau}_1(t), \dots, \vec{F}_i(t), \vec{\tau}_i(t), \dots, \vec{F}_N(t), \vec{\tau}_N(t))^T \\ \text{et } \mathbf{W}(t) &= \text{diag} (m_1^{-1}, \mathbf{I}_1^{-1}(t), \dots, m_i^{-1}, \mathbf{I}_i^{-1}(t), \dots, m_N^{-1}, \mathbf{I}_N^{-1}(t)). \end{aligned}$$

## A.2 Approximation numérique

Dans le contexte de la simulation numérique,  $\vec{S}_{\text{pos}}(t)$  et  $\vec{S}_{\text{vit}}(t)$  sont évalués à chaque pas de temps. Ces évaluations reposent sur le calcul de  $\vec{S}_{\text{pos}}(t + \Delta t)$  et de  $\vec{S}_{\text{vit}}(t + \Delta t)$  par une méthode d'approximation numérique. ODE utilise la méthode EULER explicite. La description de cette méthode est illustrée par la figure A.1.

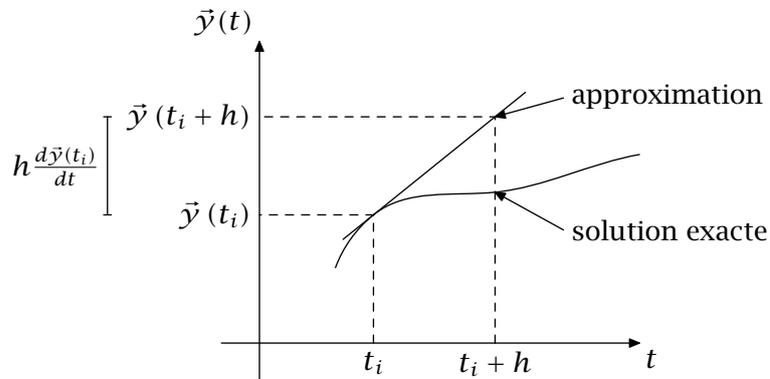


FIGURE A.1. Méthode d'approximation EULER explicite

On peut alors écrire

$$\begin{aligned}\vec{S}_{\text{pos}}(t + \Delta t) &= \vec{S}_{\text{pos}}(t) + \Delta t \frac{d\vec{S}_{\text{pos}}(t)}{dt} \\ \vec{S}_{\text{vit}}(t + \Delta t) &= \vec{S}_{\text{vit}}(t) + \Delta t \frac{d\vec{S}_{\text{vit}}(t)}{dt} \\ \vec{S}_{\text{vit}}(t + \Delta t) &= \vec{S}_{\text{vit}}(t) + \Delta t \mathbf{W}(t) \vec{Q}(t)\end{aligned}$$

Un pas de simulation respecte la séquence suivante :

1. calcul des efforts et des couples totaux appliqués à tous les solides (obtention de  $\vec{Q}$ ),
2. ajout de  $\mathbf{I}(t) \vec{\omega}(t) \times \vec{\omega}(t)$  à chaque composante de couple de  $\vec{Q}$ ,
3. mise à jour de  $\vec{S}_{\text{vit}}$ ,
4. mise à jour de  $\vec{S}_{\text{pos}}$ .

Le problème se résume au calcul des efforts et des couples.

### A.3 Utilisation de contraintes en vitesse pour la résolution

Une contrainte a pour but de restreindre les mouvements d'un solide. Du point de vue pratique, cette restriction de mouvement est obtenue par l'application d'efforts destinés à garantir la restriction. Ces efforts sont appelés efforts de contraintes afin de les distinguer des efforts extérieurs et doivent être calculés après ces derniers. En effet, les efforts de contraintes doivent pouvoir empêcher les mouvements d'un solide dues aux efforts extérieurs si ces mouvements sont les objets de la contrainte.

#### A.3.1 Contraintes en vitesse entre deux solides

Dans la méthode de résolution proposée par ODE, toutes les contraintes sont exprimées par des contraintes en vitesse. Dans un premier temps, considérons une contrainte en vitesse entre deux solides. Cette contrainte s'exprime par le système d'équations suivant :

$$\begin{aligned}\mathbf{J}_1 \vec{v}_1 + \mathbf{\Omega}_1 \vec{\omega}_1 + \mathbf{J}_2 \vec{v}_2 + \mathbf{\Omega}_2 \vec{\omega}_2 &= \vec{c}_l + \mathbf{C}_l \vec{\lambda}_l, \\ \vec{\lambda}_l &\geq \vec{l}_l, \\ \vec{\lambda}_l &\leq \vec{h}_l,\end{aligned}$$

avec

$\vec{v}_1, \vec{\omega}_1, \vec{v}_2, \vec{\omega}_2$	vecteurs de $\mathbb{R}^3$ , vitesses linéaires et angulaires des deux solides contraints
$\mathbf{J}_1, \mathbf{\Omega}_1, \mathbf{J}_2, \mathbf{\Omega}_2$	matrices jacobienne $\in \mathcal{M}_{(m,3)}(\mathbb{R})$ , composées des dérivées partielles des déviations de la contrainte par rapport aux composantes des vitesses
$\vec{c}_l$	vecteur de $\mathbb{R}^m$ , défini par la contrainte
$\mathbf{C}_l$	matrice diagonale $\in \mathcal{M}_{(m)}(\mathbb{R})$ , désignée en anglais par le terme <i>Constraint Force Mixing</i> , tous les éléments de la diagonale sont identiques et introduisent un réglage proportionnel de l'influence de l'effort de contrainte par l'utilisateur
$\vec{\lambda}_l$	vecteur de $\mathbb{R}^m$ , multiplicateurs de LAGRANGE
$\vec{l}_l$	vecteur de $\mathbb{R}^m$ , limites inférieures des valeurs de $\vec{\lambda}_l$
$\vec{h}_l$	vecteur de $\mathbb{R}^m$ , limites supérieures des valeurs de $\vec{\lambda}_l$

La dimension de la première équation est égale à  $m$ , le nombre de degrés de liberté supprimés par la liaison. La détermination de  $\vec{\lambda}_l$  permet l'obtention des efforts et couples de chacun des deux corps, soient  $\mathbf{J}_1^T \vec{\lambda}_l$ ,  $\mathbf{\Omega}_1^T \vec{\lambda}_l$ ,  $\mathbf{J}_2^T \vec{\lambda}_l$  et  $\mathbf{\Omega}_2^T \vec{\lambda}_l$  respectivement. La caractéristique principale de ces équations est que les variables de la première équation sont homogènes à des vitesses et que l'inconnue  $\vec{\lambda}_l$  possède des valeurs limites. L'exploitation de la dernière de ces caractéristiques implique qu'une contrainte peut disposer de l'équivalent d'un moteur. Nous pouvons, par exemple, spécifier une vitesse relative linéaire ou angulaire entre les deux corps et imposer des valeurs limites en efforts ou en couples par l'intermédiaire de  $\vec{h}_l$  et de  $\vec{l}_l$ .

Les définitions des matrices jacobiennes, de  $\vec{c}_l$ , de  $\vec{l}_l$  et de  $\vec{h}_l$  dépendent de la contrainte. La section suivante propose d'illustrer ce propos par un exemple.

### A.3.2 Exemple de contrainte : liaison rotule

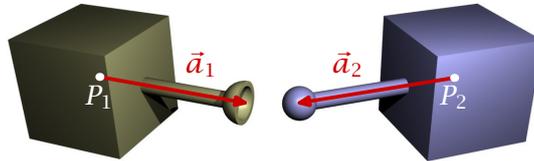


FIGURE A.2. Liaison rotule (avec représentation d'une erreur en position du centre de la liaison)

Soit une liaison rotule liant deux solides telle que représentée par la figure A.2. Ce type de liaison se caractérise par :

- la coïncidence de la position de deux points appartenant chacun à un des solides,
- l'égalité des vitesses linéaires de ces deux points.

L'égalité des vitesses linéaire se traduit par

$$\begin{aligned}\vec{v}_1 + \vec{\omega}_1 \times \vec{a}_1 &= \vec{v}_2 + \vec{\omega}_2 \times \vec{a}_2, \\ \vec{v}_1 + \vec{\omega}_1 \vec{a}_1 &= \vec{v}_2 + \vec{\omega}_2 \vec{a}_2, \\ \vec{v}_1 - \vec{a}_1 \vec{\omega}_1 &= \vec{v}_2 - \vec{a}_2 \vec{\omega}_2.\end{aligned}$$

Soit

$$\vec{v}_1 - \vec{a}_1 \vec{\omega}_1 - \vec{v}_2 + \vec{a}_2 \vec{\omega}_2 = \vec{0}$$

avec  $\vec{a}_1$ ,  $\vec{\omega}_1$ ,  $\vec{a}_2$  et  $\vec{\omega}_2$  définis suivant l'équation (A.6).

Par identification, on obtient les matrices jacobiennes

$$\mathbf{J}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{\Omega}_1 = -\vec{a}_1, \quad \mathbf{J}_2 = -\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{et} \quad \mathbf{\Omega}_2 = \vec{a}_2.$$

On ne limite pas l'effort de contrainte et la condition de contrainte est indépendante de l'effort de contrainte. Par conséquent,  $\vec{l}_l = -\vec{\infty}$ ,  $\vec{h}_l = +\vec{\infty}$  et  $\mathbf{C}_l = \mathbf{0}$ .

$\vec{c}_l$  est proportionnel à l'erreur en position entre les points de liaison de chaque solide

$$\vec{c}_i = \epsilon \frac{((\vec{p}_2 + \vec{a}_2) - (\vec{p}_1 + \vec{a}_1))}{\Delta t}$$

avec  $\Delta t$  le pas de temps et  $\epsilon$  un paramètre réglable de réduction d'erreur (*Error Reduction Parameter* de ODE) compris entre 0 et 1.

### A.3.3 Formulation globale des équations de contraintes

Les contraintes reliant les solides deux à deux peuvent être assemblées pour obtenir la formulation suivante :

$$\mathbf{J}\vec{q} = \vec{c} + \mathbf{C}\vec{\lambda}, \quad (\text{A.22a})$$

$$\vec{\lambda} \geq \vec{l}, \quad (\text{A.22b})$$

$$\vec{\lambda} \leq \vec{h}. \quad (\text{A.22c})$$

La matrice jacobienne globale est composée par les matrices jacobiennes issues des contraintes liant les solides deux à deux, relativement aux vitesses linéaires et angulaires. Chaque ligne de la première équation retire un degré de liberté au système. Les composants de  $\vec{\lambda}$  sont les multiplicateurs de LAGRANGE. L'effort global de contrainte est  $\mathbf{J}^T \vec{\lambda}$ .

On cherche la solution des équations de contraintes à l'instant  $t + \Delta t$ . En notant les efforts extérieurs  $\vec{Q}_{\text{ext}}$ , on obtient

$$\vec{Q} = \vec{Q}_{\text{ext}} + \mathbf{J}^T \vec{\lambda}$$

et l'équation (A.22a) devient

$$\mathbf{J} \left( \vec{q} + \mathbf{W} \left( \vec{Q}_{\text{ext}} + \mathbf{J}^T \vec{\lambda} \right) \Delta t \right) = \vec{c} + \mathbf{C}\vec{\lambda}. \quad (\text{A.23})$$

Afin de résoudre le système linéaire avec contraintes, l'équation (A.23) peut se mettre sous la forme

$$\mathbf{A}\vec{\lambda} = \vec{b} + \vec{\gamma} \quad (\text{A.24})$$

où

$$\mathbf{A} = \mathbf{J}\mathbf{W}\mathbf{J}^T - \frac{\mathbf{C}}{\Delta t},$$

$$\vec{b} = \frac{\vec{c}}{\Delta t} - \mathbf{J} \left( \frac{\vec{q}}{\Delta t} + \mathbf{W}\vec{Q}_{\text{ext}} \right),$$

et  $\vec{\gamma}$ , un vecteur résidu. Les éléments de  $\vec{\lambda}$  et de  $\vec{\gamma}$  respectent l'un des trois cas suivants :

$$\lambda_i = h_i, \quad \gamma_i \leq 0$$

$$\lambda_i = l_i, \quad \gamma_i \geq 0$$

$$l_i \leq \lambda_i \leq h_i, \quad \gamma_i = 0$$

L'équation (A.24) à deux inconnues  $\vec{\lambda}$  et  $\vec{\gamma}$  peut être résolue à l'aide de l'algorithme de DANTZIG tel que présenté dans [COUTINHO, 2001]. Cet algorithme décompose le problème

---

en trois groupes relativement aux trois cas énoncés ci-dessus. Après décomposition, des solutions intermédiaires sont obtenues en fonction du conditionnement propre à chacun des groupes.

## A.4 Contacts

Lorsqu'un solide entre en contact avec un autre, une contrainte de contact reliant les deux solides impliqués est ajoutée pour le point concerné.

Pour la modélisation du frottement aux points de contact, la bibliothèque ODE propose deux modèles de frottement dérivés du modèle de frottement de COULOMB ( $\|\vec{f}_t\| \leq \mu \|\vec{f}_n\|$ ).

1. Le premier modèle consiste à fixer la limite maximale de la norme de l'effort tangentiel au point de contact indépendamment de la norme de l'effort normal. On perd la signification physique du modèle de COULOMB mais on gagne en coût de calcul. Du point de vue de la notation utilisée,  $\mu$  correspond à une limite en effort et doit être adapté à la simulation. Pour un modèle de contact, ODE offre la possibilité de définir un axe de frottement principal, perpendiculaire à la normale du contact. La définition de ce premier axe entraîne la création d'un second, perpendiculaire au premier axe principal et à la normale. On dispose alors de la capacité de régler la limite en effort indépendamment suivant ces deux axes principaux de frottement. Ce comportement est optionnel.

2. Le second modèle est une approximation pyramidale du cône de frottement. La pyramide de frottement est alignée suivant les axes principaux de frottement. ODE pose une approximation supplémentaire en calculant d'abord tous les efforts normaux en considérant des contacts sans frottement. Ensuite, on calcule la norme limite des efforts tangentiels suivant la loi.

L'utilisation de l'un ou l'autre de ces modèles de frottement modifie la définition de la contrainte de contact.



---

---

---

## **Contribution à l'étude, à la simulation et à la commande d'un robot de type serpent**

### Résumé

Ce manuscrit porte sur l'étude d'une structure robotique originale dont le système de propulsion s'inspire du mode de locomotion par progression rectilinéaire des serpents. La structure proposée dérive d'un prototype développé par l'entreprise TDA et comprend plusieurs modules dotés de 4 mécanismes locomoteurs solidaires entraînés par un moteur. Une partie des travaux de thèse porte sur l'analyse et l'optimisation du mécanisme.

L'analyse du déplacement d'un modèle du prototype existant par un simulateur robotique intégré, réalisé à l'aide de bibliothèques à code source ouvert, a montré la nécessité de le modifier. Ainsi, le nombre de mécanismes locomoteurs par module passe de 8 à 4 et les liaisons inter-modulaires du nouveau robot sont des rotules actives ou passives selon les directions.

Un algorithme de suivi de trajectoire est proposé et des simulations ont permis de le valider.

Mots clés : Robotique mobile, robots de type serpent, simulation.

### **On the Study, the Control and the Simulation of a Snake-Like Robot**

#### Abstract

This dissertation is concerned with the study of an original robotic design whose propulsion system is inspired by the rectilinear mode of locomotion, as observed among real snakes. The proposed design is based on a prototype developed by the TDA company and consists of several modules. Each module is composed by a main body and 4 mechanisms driven by a motor. An analytic study is conducted and optimizations of the locomotor mechanism are introduced.

The analysis of the existing prototype's model within an integrated robotic simulator based on open source libraries has shown the necessity to improve the robotic design. Consequently, the number of mechanisms embedded in a module has been decreased from 8 to 4 and ball-and-socket joints are used as intermodular joints. The intermodular joints are actuated or passive depending on the considered axis.

A trajectory tracking algorithm is proposed. This algorithm has been validated by simulation.

Keywords : Mobile robotics, snake-like robots, simulation.