



HAL
open science

Robots reconfigurables : Étude de la convergence dans le problème du déplacement vers un but

Eliau Carrillo

► **To cite this version:**

Eliau Carrillo. Robots reconfigurables : Étude de la convergence dans le problème du déplacement vers un but. Autre [cs.OH]. Université de Bretagne Sud, 2007. Français. NNT : . tel-00517181

HAL Id: tel-00517181

<https://theses.hal.science/tel-00517181v1>

Submitted on 13 Sep 2010

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.



Université de Bretagne Sud
et
Université Nationale Expérimentale
Rómulo Gallegos
Vénézuéla



École doctorale

Doctorat en sciences de l'informatique

Thèse

ELIAN CARRILLO

Robots reconfigurables :

**Étude de la convergence dans le problème
du déplacement vers un but**

Thèse dirigée par Dominique DUHAUT,

Professeur à l'Université de Bretagne Sud
En partenariat avec l'Université Rómulo Gallegos

Lorient, le 7 juin 2007

Jury

Alexis DROGOUL, Professeur à l'Université de Paris 6, Rapporteur
Josep Lluís DE LA ROSA, Professeur à l'Université de Gironne (Espagne), Rapporteur
Laurent NANA, Professeur à l'Université de Bretagne Occidentale, Examineur
Dominique DUHAUT, Professeur à l'Université de Bretagne Sud, Directeur

A Elena, Jaime, Aimée, Aliana et Cécilia

Remerciements

Je voudrais, avant toute chose, exprimer ma plus grande reconnaissance à Monsieur Dominique Duhaut, qui a encadré ce travail avec efficacité, patience et rigueur. Sous sa direction j'ai pu développer mes idées avec de très grandes libertés.

Les travaux présentés dans ce mémoire ont été effectués au sein du laboratoire VALORIA, à l'Université de Bretagne Sud et en collaboration avec l'Université Experimental Romulo Gallegos (UNERG) au Venezuela. Je tiens à remercier Luis Gallardo, président de l'UNERG pour m'avoir apporté son soutien inconditionnel depuis le début de la thèse. Son apport a été essentiel dans les projets de coopérations bilatéral entre l'UBS et l'UNERG.

Je tiens également à exprimer ma gratitude à Josep Lluís de la Rosa pour avoir accepté d'être rapporteur de ce travail. L'intérêt qu'il a porté à la lecture du manuscrit me touche particulièrement. J'ai grandement apprécié sa gentillesse au cours de nos différentes rencontres et j'ai le grand souhait de continuer à collaborer avec lui dans la recherche au tour de systèmes robotiques.

Je remercie Alexis Drogoul d'avoir accepté de m'accorder une partie de son temps précieux pour être rapporteur de cette thèse. Ses questions et remarques ont été pour moi une grande source de réflexion et m'a permis d'améliorer l'ensemble de mon travail.

Je tiens aussi à remercier à Monsieur Laurent Nana, professeur des Universités à l'Université de Bretagne Occidentale qui m'a fait l'honneur de présider mon jury de thèse.

Je voudrais aussi remercier Jeanne Villaneau pour ses remarques pertinentes et constructives, ses conseils avisés et son soutien.

J'adresse toute ma sympathie à l'équipe du VALORIA, qui m'a accueilli si généreusement particulière à certain d'entre eux : Pierre-François Marteau, Gildas Ménier, Franck Poirier... Aussi à toute l'équipe de chercheur et doctorants. L'entente cordiale entre le l'équipe

J'ai grandement apprécié la gentillesse de Karen Le Dessert et de son efficacité, qui m'a facilité l'interaction avec les institutions universitaires.

Je tiens à exprimer tout ma gratitude en vers tous mes amis : Mauricio, Yoann, Claudio, Ivan, Martin, Evelyn, Natalia, Laure, Amira, Saadia, Kods et tant d'autres qui m'ont soutenue et encourager le long de mon travaille et de la réalisation de ce mémoire.

Je remercie de tout mon cœur le soutien que m'a apporté ma famille au Venezuela. Grâce à l'amour qu'elle m'a apportée j'ai pue faire une évolution interne et intellectuelle en toute sérénité. C'est à elle à qui je dédie ce mémoire de thèse.

Table des matières

Table des matières	7
Table des figures des tableaux et des équations	13
Introduction.	17
Chapitre 1. Introduction à la robotique et systèmes multi-agents	19
Origine de la robotique	19
1.1 Introduction à la robotique	19
<i>1.1.1 La robotique industrielle</i>	19
<i>1.1.2 La robotique intelligente</i>	21
<i>A. Aibo : robot de divertissement</i>	21
<i>B. QRIO</i>	23
<i>C. Asimo</i>	24
<i>D. HRP2</i>	25
<i>E. Repliee robot d'interaction</i>	26
<i>1.1.3 Les drones</i>	26
<i>A. La libellule</i>	27
<i>B. Des ballons dirigeables</i>	27
<i>1.1.4 Aide à la personne</i>	28
<i>A. Aide à la personne : interface Tactile Portable pour le Handicap Visuel</i>	28
<i>B. Robotique médicale et chirurgicale</i>	29
<i>C. Assistance à la conduite automobile</i>	30
<i>1.1.5 Les nouveaux défis : la robotique collective et multi-robots</i>	31
<i>A. La RoboCup</i>	31
<i>B. Les travaux de Mataric</i>	32
<i>C. CEBOT</i>	33

1.1.6 Les nouveaux défis : la robotique modulaire	34
A. Enjeux de la modularité	34
B. RMMS	35
C. DIGITAL Clay	36
1.1.7 Les nouveaux défis : la robotique modulaire reconfigurable	37
A. SWARM-BOT	37
B. I-Cubes	39
C. ATRON	40
D. POLYBOT	41
E. PROTEO	41
F. Système de transfert flexible (FTS) à travers l'auto-reconfiguration	42
G. Le projet M-TRAN	43
H. MAAM	43
1.1.8 Système complexe	45
1.2 IA et Système multi-agents	45
1.2.1 Définition des systèmes multi-agents	45
1.2.2 Des catégories d'agents	46
1.2.3 Les systèmes multi-agents	46
A. L'approche réactive	46
B. Emergence de comportement	47
C. Le principe de protection	47
Chapitre 2. Déplacement collectif : problématique et hypothèses	49
2.1 Problématique	49
A. La connexion	49
B. Le temps raisonnable	50
C. L'approche proposée : l'émergence	50
D. Les compagnons de l'émergence : blocage & oscillation et une solution	51
2.2 Formalisation du problème	51

2.2.1 Modélisation du monde du robot : attracteur & gradient	51
<i>A. Potentiel régulier</i>	52
<i>B. Potentiel double</i>	52
<i>C. Champ de potentiel déformé</i>	53
2.2.2 La perception des agents	54
2.2.3 Déplacement des atomes	54
<i>A. Déplacements élémentaires</i>	54
<i>B. Contrainte sur le déplacement</i>	55
2.2.4 L'ordonnancement des calculs dans les agents	56
<i>A. Ordonnancement séquentiel</i>	56
<i>B. Ordonnancement aléatoire</i>	57
2.3. Algorithme réactif	57
<i>A. Principe de l'algorithme réactif direct</i>	57
Algorithme "forward total"	57
Algorithme "forward un coup"	57
<i>B. Principe de l'algorithme réactif avec retour arrière</i>	58
Algorithme "backward"	58
Algorithme reabsorbtion	58
Sensibilité des algorithmes au type de gradient	59
<i>Descente de gradient simple</i>	59
<i>Descente d'un gradient double</i>	59
2.4. Synthèse	60
Chapitre 3. Déplacement collectif : simulations, erreurs, solution	61
3.1. Introduction	61
3.2. Algorithme réactif	61

<i>Simulation 1 : Potentiel régulier, ordonnancement séquentiel, algorithme type « forward total »</i>	62
<i>Simulation 2 : Potentiel déformé, ordonnancement séquentiel, algorithme type « forward total »</i>	63
<i>Simulation 3 : Potentiel non déformé avec obstacle, ordonnancement séquentiel, algorithme type « forward total »</i>	64
<i>Simulation 4 : Potentiel non déformé, ordonnancement séquentiel, algorithme type backward</i>	65
3.3 Méthodes connexionnistes	67
<i>3.3.1 Apprentissage supervisé</i>	67
<i>3.3.2 Apprentissage à partir d'observations</i>	68
<i>3.3.3 Structure générale du réseau de neurones</i>	69
<i>A. Les données du réseau</i>	69
<i>B. Les sorties du réseau</i>	69
<i>C. Structure du réseau</i>	70
<i>D. Fonctions de transfert</i>	70
<i>3.3.4 Méthode d'apprentissage</i>	70
<i>A. Initialisation des poids</i>	71
<i>3.3.5 Simulation sur plusieurs types de réseaux</i>	71
<i>3.3.2 Evaluation de l'apprentissage</i>	74
3.4 Des problèmes liés aux hypothèses	75
<i>3.4.1 Les problèmes intrinsèques au modèle</i>	76
<i>A. Le « U » avec un potentiel régulier</i>	76
<i>B. Le « O » avec un potentiel quelconque</i>	76
<i>3.4.2 Les problèmes dus à la dynamique</i>	77
<i>A. L'émergence du «U»</i>	77
Exemples	78

Simulation 1 : Emergence de U sous un ordonnancement séquentiel (<i>sans blocage</i>) Potentiel régulier + Forward total + ordonnancement séquentiel	78
Simulation 1 : Emergence de U sous un ordonnancement séquentiel Potentiel régulier + Forward total + ordonnancement séquentiel	79
Simulation 2 : Emergence de U sous un ordonnancement aléatoire Potentiel simple + Forward total + ordonnancement aléatoire	80
Simulation 3 : Emergence de U sous un ordonnancement aléatoire « doigts » multiples Potentiel simple + Forward total + ordonnancement aléatoire	81
<i>B. L'émergence du «O» sur algorithme forward</i>	82
<i>C- L'émergence du «O» sur algorithme backward</i>	82
Exemples	83
Simulation 1 : Emergence de O sous un ordonnancement aléatoire Potentiel simple + Forward + ordonnancement aléatoire	83
Simulation 2 : Emergence de O sous un ordonnancement séquentiel Potentiel simple + Backward + ordonnancement séquentiel	84
Simulation 3 : Potentiel double + Forward + ordonnancement aléatoire	85
<i>Cas de convergence de la simulation</i>	85
<i>Cas d'émergence de O et de U</i>	85
3.5 Synthèse	86
Chapitre 4. Vers une méthodologie de déplacement collectif	87
4.1 Analyse des simulations	87
4.2. Unicité de la position à atteindre	87
4.3. Une méthodologie pour garantir la convergence	89
Propriété 1: Unicité d'un point minimal	89
Propriété 2: atomes actifs	89
Propriété 3: chemin décroissant au point minimal	90
Propriété 4 : progression	90
Lemme 1	91
Lemme 2	91

Théorème	92
4.4 Comment forcer un ordonnancement	94
<i>A- Ajouter un état</i>	94
<i>B- Ajouter une perception</i>	94
4.5 Algorithme	95
Algorithme “forward un coup avec perception d'état”	95
4.6 Simulations avec perception d'état	96
<i>Simulation 1 : émergence du comportement de déplacement ver l'attracteur sous un ordonnancement aléatoire</i>	96
<i>Simulation 2 : émergence d'un comportement de ligne avec un atome temporellement bloqué</i>	97
4.7 Généralisation de l'approche	98
Chapitre 5. Réalisation matérielle	99
5.1 Mise en oeuvre d'un potentiel double	99
5.2. Reconstruction de la carte du potentiel	100
5.3. Mise en oeuvre d'un variable d'état	101
Conclusion	103
Bibliographie	105
Références Internet	111
Annexe 1.	113
A.1 Les actions possibles dans le voisinages d'un atome	113

Table des figures des tableaux et des équations

Figure 1.1 Robots industriels	20
Figure 1.2 Estimation annuelle d'installations de robots industrielles	20
Figure 1.3 Différents modèles du robot AIBO	21
Figure 1.4 Interaction du robot AIBO	22
Figure 1.5 Modularité du modèle AIBO ERS-210	22
Figure 1.6 QRIO de Sony	23
Figure 1.7 Evolution d'Asimo	24
Figure 1.8 HPR2 et Hoap3	25
Figure 1.9 Des actroïdes à l'image des humains	26
Figure 1.10 La libellule	27
Figure 1.11 Illustrations du CargoLifter et du Skycat d'ATG	27
Figure 1.12 Interface tactile portable pour le handicap visuel	28
Figure 1.13 Application de l'interface tactile pour toucher avec les doigts une représentation des obstacles de l'environnement	28
Figure 1.14 Systèmes passifs pour la comanipulation en neurochirurgie	29
Figure 1.15 Le robot DA VINCI et immersion 3D	29
Figure 1.16 Interface d'assistance au maintien de voie par vision frontale, et celle d'affichage de vitesse excessive	30
Figure 1.17 Le prototype MTRC	30
Figure 1.18 Compétition, Robocup	31
Figure 1.19 Robots footballeurs	32
Figure 1.20-a Système multi-robot et des simulation utiliser par M. Mataric	32
Figure 1.20-b Système multi-robot et des simulation utiliser par M. Mataric	33
Figure 1.21 Le système CEBOT et possible application	33
Figure 1.22 Modularité et interaction par G. S. Virk	34
Figure 1.23 Séquence de montage d'un bras du système RMMS	35
Figure 1.24 Le RMMS configuré en bras robotique et son simulateur	36
Figure 1.25 Prototype du système Digital Clay	36
Figure 1.26 Le S-bot, réalisation matériel et sont modèle CAO	37
Figure 1.27 Le Swarm-bot franchissant un fosse	37
Figure 1.28 Séquence d'un s-bot avec un obstacle infranchissable, reconfiguration et franchissement de l'obstacle	38
Figure 1.29 Système I-Cubes	39
Figure 1.30 Système reconfigurable : ATRON	40
Figure 1.31 Système reconfigurable ATRON et simulation	40
Figure 1.32 Séquence de reconfiguration système PolyBot	41
Figure 1.33 Séquence simulée de système PROTEO	41
Figure 1.34 FST, un système de transfère flexible	42
Figure 1.35 Le système M-TRAN	43
Figure 1.36 Le système MAAM	44
Figure 1.37 Réalisation d'un atome	44
Figure 2.1 Déplacement reconfigurable	49

Figure 2.2 Molécule et Atome	50
Figure 2.3 Champ de potentiel régulier	52
Figure 2.4 Champ de potentiel double	52
Figure 2.5 Champ de potentiel déformé	53
Figure 2.6 Perception des atomes	54
Figure 2.7 Exemple de reconfiguration successives d'un atome	54
Figure 2.8 (a) Les 4 connectivités de Von Newman. (b) Le voisinage de Moore	55
Figure 2.9 Directions et les vecteurs du voisinage correspondants	55
Figure 2.10 Exemples de déplacement autorisé ou non autorisé	56
Figure 2.11 Distance à l'attracteur (D) et Distance à la ligne (H)	59
Figure 3.1 Images de la simulation 1	62
Figure 3.2 Images de la simulation 2	63
Figure 3.3 Images de la simulation 3	64
Figure 3.4 Images de la simulation 4	65
Figure 3.5 Le réseau	67
Figure 3.6 Simulation de l'apprentissage supervisé	68
Figure 3.7 Temps en nombre de pas pour atteindre la cible, algorithme réactif et apprentissage supervisé	68
Figure 3.8 Architecture du réseau de neurone retenu	73
Figure 3.9 Evaluation des apprentissages	74
Figure 3.10 Exécution d'un réseau appris	75
Figure 3.11 Formation de « U »	76
Figure 3.12 Formation de « O »	76
Figure 3.13 Formation de « U »	77
Figure 3.14 Emergence du comportement désiré avec un potentiel régulier et ordonnancement séquentiel	78
Figure 3.15 Emergence du U avec un potentiel régulier et ordonnancement séquentiel	79
Figure 3.16 Emergence du U avec un potentiel régulier et ordonnancement aléatoire	80
Figure 3.17 Emergence du U avec un potentiel régulier et ordonnancement aléatoire	81
Figure 3.18 Dynamique de formation du O	82
Figure 3.19 Cas d'émergence sous ordonnancement séquentiel	83
Figure 3.20 Emergence du O avec un potentiel régulier et ordonnancement aléatoire	84
Figure 3.21 Formation qui marche	85
Figure 3.22 Formation de « U »	85
Figure 4.1 Le potentiel double	88
Figure 4.2 Point minimum au sens de l'ordre défini	88
Figure 4.3 Atomes aux mouvements autorisés	89
Figure 4.4 Chemins décroissants	90
Figure 4.5 Chemins décroissants	91
Figure 4.6 Simulation avec un double potentiel et perception d'état	96
Figure 4.7 Simulation avec un double potentiel, perception d'état et un Atome bloqué	97
Figure 5.1 Lecture d'une diode par un récepteur à différentes distances	99
Figure 5.2 L'atome MAAM et son projet de système d'attache	100
Figure 5.3 Lecture d'un signal IR par le capteur au bout de la patte	101
Figure 5.4 Un prototype d'attache de modules MAAM	102

Tableau 1.1 Degrés de liberté d'Asimo	24
Tableau 1.2 Les différents types d'agents	46
Tableau 3.1 Nombre de combinaisons et exemples retenus	69
Tableau 3.2 Fonctions de transferts les plus fréquentes	70
Tableau 3.3 Architecture des réseaux de neurones et leurs évaluation	72
Tableau 3.4 Variantes des réseaux de neurones	73
Equation 2.1 Gaussienne	52
Equation 3.1 La somme des carrés des déviations	71

Introduction.

Déplacement collectif en robotique auto reconfigurable

La miniaturisation des systèmes électroniques et l'augmentation de leurs performances ont eu un effet immédiat sur les objectifs de la robotique. En effet, il est maintenant envisageable de fabriquer à des coûts raisonnables des systèmes mécatroniques incluant des capteurs et des actionneurs avec une capacité de décision et de communication dans des volumes proches de la taille d'un téléphone portable.

Dans cet esprit, le projet MAAM a pour ambition de fabriquer des modules robotiques capables de s'assembler en des structures complexes pour effectuer des tâches de transports ou de manipulations d'objets. Ce projet, décrit plus complètement au chapitre 1, devait résoudre un certain nombre de problèmes de conception mécanique et électronique, d'intégration et de programmation.

Du point de vue de la programmation le problème pourrait se résumer à : « une fois les modules fabriqués, comment leur expliquer ce qu'on veut leur faire faire ? ». C'est dans ce cadre que le travail présenté ici a été fait.

Deux approches à ce problème sont réalisées. L'une qui s'intéresse à la conception d'un langage de descriptions de tâches pour les modules robotiques fait l'objet du travail de thèse de Michel Dubois. L'autre s'interroge sur les programmes à mettre dans un module pour obtenir un comportement global du système conforme à un objectif. Ce mémoire décrit cette étude.

Ces deux approches sont complémentaires. L'une permet à une entité extérieure de décrire les comportements de chaque entité. L'autre par d'une capacité de décision locale et cherche comment par le biais des interactions le comportement globale du système va résoudre la tâche voulue.

Notre sujet de recherche est : « comment obtenir la suite des reconfigurations que le système robotique doit faire pour qu'il se déplace globalement vers une destination ? ». Comme nous le montrerons ce problème est très complexe. D'autres partenaires du projet ont tenté des approches à ce problème avec des outils issus du domaine d'intelligence artificielle, aucune n'a donné de résultats concluants. L'approche proposée ici est une approche par émergence où tous les modules exécutent le même programme et ne peuvent résoudre seul le problème posé. Nous verrons que ce travail qui a vite donné de bons résultats nous a permis de tester plusieurs méthodes : algorithmes réactifs, apprentissage supervisé et apprentissage à partir d'exemple. Nous montrerons également qu'au cours de ces simulations des problèmes inattendus sont apparus. Des blocages sont survenus avec tous les algorithmes. Les méthodes utilisées pour faire les simulations étaient à discuter. Enfin nous avons pu proposer une méthodologie pour résoudre ces problèmes de blocage.

Ce mémoire se décompose en 5 chapitres plus une conclusion.

Le chapitre 1 situe le projet Maam dans le contexte de la robotique d'aujourd'hui. Après une brève description de la robotique, nous présenterons quelques réalisations de robots pour différents types d'activités. Nous focaliserons notre état de l'art sur le cas plus particulier des systèmes multi-robots. Nous y distinguerons la robotique collective, la robotique modulaire et enfin la robotique modulaire et reconfigurable. Ceci nous amènera à une présentation du projet MAAM ainsi que d'autres projets connexes.

Le chapitre 2 décrira la problématique adressée dans ce mémoire : le déplacement collectif d'un robot reconfigurable. Nous définirons d'abord les propriétés que doit vérifier le système. Ensuite nous décrirons les hypothèses retenues pour les simulations. Ainsi nous montrerons la modélisation du monde retenu pour cette étude. Nous étudierons également les capacités de perceptions et les capacités de déplacements que possède le robot. Puis, nous examinerons plusieurs types de méthodes d'ordonnement de calcul dans les différents modules. Enfin nous étudierons les comportements réactifs des modules.

Le chapitre 3 retracera la chronologie de nos travaux. D'abord des simulations réactives dans un environnement libre et encombré. Puis nous présenterons les travaux effectués sur l'apprentissage supervisé ou pas.

C'est grâce à ces dernières simulations que nous avons vu émerger deux types de problèmes. L'un lié à la nature de l'attracteur utilisé, l'autre lié à l'ordonnement et à la dynamique des interactions. Ces problèmes se situent à la base de blocages, nous les définirons et expliquerons comment et pourquoi ils émergent.

Le chapitre 4 proposera des solutions pour éviter les problèmes décrits au chapitre 3. Deux axes seront proposés. L'un s'intéressera au type d'attracteur utilisé et proposera un système d'attraction aux propriétés différentes pour garantir un ordre sur l'espace. L'autre proposera une méthode pour permettre de garantir une politique d'ordonnement quelque soit l'ordre dans lequel les calculs sont effectués dans les différents modules. L'ensemble de ces deux propriétés nous permettra de prouver la convergence du système robotique reconfigurable quelque soit la position de l'attracteur et quelque soit la configuration initiale du robot.

Le chapitre 5 s'intéressera à la faisabilité des techniques proposées au chapitre 4. On montrera que des mesures effectuées sur les capteurs, actionneurs infrarouges offrent les propriétés proposées. Par ailleurs, nous montrerons que les modules du projet MAAM permettent de fournir les informations nécessaires à la réalisation des techniques requises par la politique d'ordonnement.

Enfin nous donnerons quelques conclusions et perspectives à ce travail.

Chapitre 1. Introduction à la robotique et systèmes multi-agents

Origine de la robotique

Le nom de robot est introduit dans l'histoire avec la pièce de théâtre Rossum's Universal Robots écrit par Karel Čapek [ČAP 20]. C'est à partir des premières décennies du 20^{ème} siècle que l'on emploie pour la première fois le mot robot dérivé du mot tchèque robota pour désigner la servitude. Dès lors plusieurs séries de romans sont apparues telle que le best seller d'Asimov I, Robot [ASI 50].

Un des précurseurs des robots est apparu au Japon au XVIII siècle avec des automates mécaniques appelés "karakuri-ningyô" (poupées mécaniques). Le plus connu d'entre eux est l'automate serveur de thé. Cet automate d'intérieur s'avance jusqu'à l'invité en lui apportant un bol de thé. Une fois le bol bu et reposé sur le plateau qu'il tient entre ses mains, il fait demi-tour et revient à son point de départ. Ces automates sont des ancêtres des robots.

Depuis les romans de science fiction, les robots ont évolué dans notre société. Aujourd'hui, les robots qui font l'actualité des médias et qui paraissent les plus spectaculaires viennent majoritairement du Japon, comme par exemple Aibo, Qrio, Asimo et les robots anthropomorphiques.

1.1 Introduction à la robotique

1.1.1 La robotique industrielle

A la fin des années 60, Kawasaki Heavy industries Ltd réalise le premier robot japonais de production, Unimate, conçu pour la soudure des automobiles. La production de robots prend son essor à partir de 1970 ; ils sont surtout destinés aux USA. Dans les années 1980, le Japon devient le géant du robot avec 150 fabricants, il fabrique 60% des robots dans le monde.

Dans les années 1985 l'association d'industrie robotique américaine définit le robot comme « une machine reprogrammable, multifonction, manipulateur conçu pour déplacer le matériel, pièces, ou les mouvements programmés variables spécialisés à travers de dispositifs pour l'exécution d'une variété de tâches ». Cette définition est assez restrictive et actuellement dépassée.



Figure 1.1 Robots industriels

Les robots se multiplient dans les usines du monde entier. La Commission Economique des Nations Unies pour l'Europe [unece] estime leur nombre à 757 000 à la fin 2001. Quant à la population des robots européens, celle-ci s'élève à 219 000 individus et celle des Américains à 97 000. Actuellement, la population mondiale pourrait atteindre 965 000 robots, selon les estimations de la Commission.

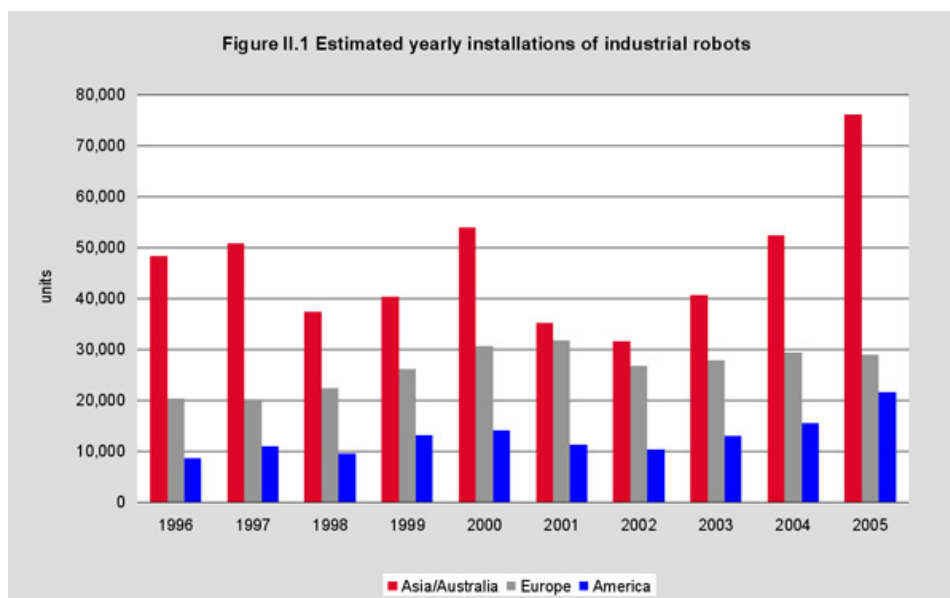


Figure 1.2 Estimation annuelle d'installations de robots industrielles

En 1999 et en 2000, les ventes de robots industriels en France ont augmenté de façon vertigineuse, soit une augmentation annuelle de 3800 unités. Ce fut de loin la plus forte croissance jamais enregistrée en France pour les ventes de robots.

En 2001, par contre, les ventes ont diminué de 8 % par rapport à 2000 pour atteindre un niveau de 3500 unités. En 2002, le marché a de nouveau enregistré une baisse de 14 % (voir figure 1.2). A titre de comparaison, la valeur du marché français des machines-outils est tombée de 27 % en 2002. Les investissements dans le secteur des robots semblent être moins sensibles à la récession que les autres types de biens d'investissement. Toutefois, le marché a repris en 2003 entraînant une augmentation de 4 % sur 2002.

Fin 2003, le stock des robots utilisés en France était estimé à environ 26 100 unités, soit une augmentation de 8% par rapport à 2002. Les prévisions pour la période 2004-2007 donnent une hausse de 6 % par an du marché des robots, ce qui équivaut à un stock de près de 36 000 unités fin 2007.

La robotique industrielle représente 4.2 milliards de dollars pour 1998, et le parc mondial serait estimé fin 1998 à plus de 700.000 robots dans le monde, selon l'International Federation of Robotics [ifr] .

1.1.2 La robotique intelligente

Pour R. Arkin un robot intelligent est : « *une machine capable d'extraire l'information à partir de son environnement et de les employer d'après ses connaissances sur le monde et de s'y déplacer sans risque, et d'une façon autonome...* », « *des robots peuvent être différenciés en terme de leur taille, du matériel dont ils sont faits, du type de joints qui les unis ensemble, de leurs actionneurs, du type des systèmes des capteurs qu'ils possèdent, leur système de locomotion, et leur système informatique de bord* ».

Évidemment la structure physique seule n'est pas suffisante pour obtenir de bons résultats. Le système de contrôle des robots est fondamental pour les faire évoluer d'une manière coordonnée.

En marge de ces définitions, on voit apparaître ces dernières années la notion de robot de divertissement, d'aide à la personne sur un plan professionnel (médecine), personnel (domicile) ou de distraction.

A. Aibo : robot de divertissement [sony]

Une grande communauté s'est développée autour de l'apparition des robots programmables pour le grand public, avec la possibilité de créer ses propres programmes ou bien d'en télécharger sur Internet.



Figure 1.3 Différents modèles du robot AIBO

Au fur et à mesure le chien Aibo est devenu de plus en plus complexe. Dans sa dernière version il est capable de parler (1.000 mots), d'interagir avec les ordres de son propriétaire, de prendre des photos grâce à la caméra présente dans la tête et enfin de jouer de la musique.



Figure 1.4 Interaction du robot AIBO

Aibo est un robot d'entretien très remarquable. Il a plusieurs actionneurs : 3 par patte et 3 pour la tête, lui donnant à chacune de ses parties 3 ddl. Il a aussi plusieurs capteurs de toucher : à l'extrémité de ses pattes, sur son dos, sur sa tête et dans son menton. La liste de capteurs ne s'arrête pas là. Aibo est équipé de 3 gyroscopes, un capteur de vibrations, un émetteur récepteur infrarouge, une caméra couleur et des micros en stéréo. Il possède un processeur lui permettant d'exécuter un ou plusieurs programmes avec système opératif qui tourne sous OPEN-R, et un lecteur memory-stick. Le CPU embarqué lui donne la possibilité de communiquer en réseau grâce à son interface wifi.

On peut se servir de la plate-forme Urbi [urbi] pour programmer le chien sur le memory-stick ou en ligne, changer ses comportements et obtenir les données concernant ses capteurs.

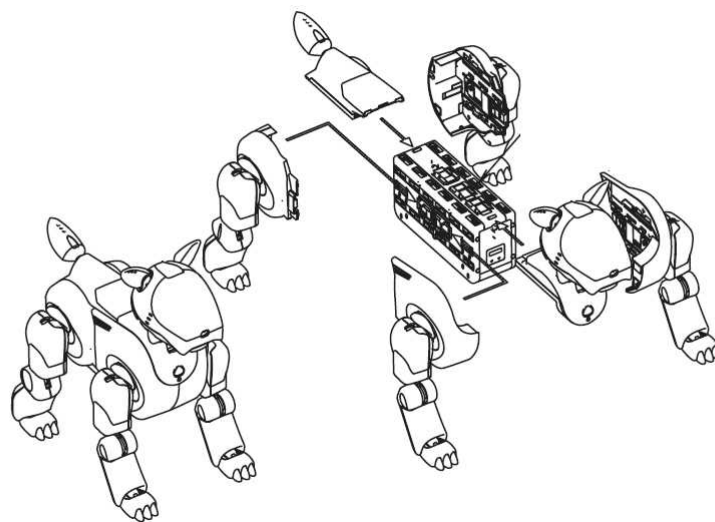


Figure 1.5 Modularité du modèle AIBO ERS-210

Sony a annoncé l'arrêt de sa production en dépit de 150 000 exemplaires vendus au prix de vente unitaire de 185.000 yens (1.500 euros).

B. QRIO [sony]

Sony a dévoilé, le 18 décembre 2003, QRIO, le premier robot capable de courir. Ce robot humanoïde dont Sony n'a pas l'intention de commercialiser peut atteindre une vitesse de 14 mètres par minute.

Sony a mis trois ans pour le fabriquer sans toutefois chiffrer le coût de sa construction. Le prototype métallique QRIO a prouvé sa stabilité dans la dynamique en perdant contact avec le sol. Il a aussi montré qu'il pouvait frapper une balle de base-ball et danser avec des drapeaux dans les mains.

Il possède plusieurs degrés de liberté (dof) : 4 pour le cou, 2 pour le corps, 5 dans chaque bras, 6 par pied, ce qui fait un total de 28 ddl plus les 5 doigts de chaque main.

Il est doté de plusieurs capteurs : des gyroscopes pour les trois dimensions pour le tronc et en XY pour les pieds (selon l'axe), des capteurs de température, d'appui et de toucher au niveau de ses bras, sa tête et ses épaules.

Il est équipé d'une vision stéréo avec des caméras CCD de 110.000 pixels de résolution et de 7 microphones.

Sa vitesse de marche est approximativement de 6 mètres par minute et de 20 mètres par minute quand la surface du sol est lisse et régulière. Son poids est de 6,5 Kg avec batterie et mémoire et ses dimensions sont de 580x260x190 mm.



Figure 1.6 QRIO de Sony

C. Asimo [honda]

Asimo est un robot humanoïde développé par Honda. Il est l'acronyme de « Advanced Step in Innovative MObility » ; prononcé « ashimo », qui veut également dire jambe en japonais. Asimo est le résultat de dix-sept ans de recherches de Honda. Il est capable de marcher, courir et grimper aisément des escaliers.

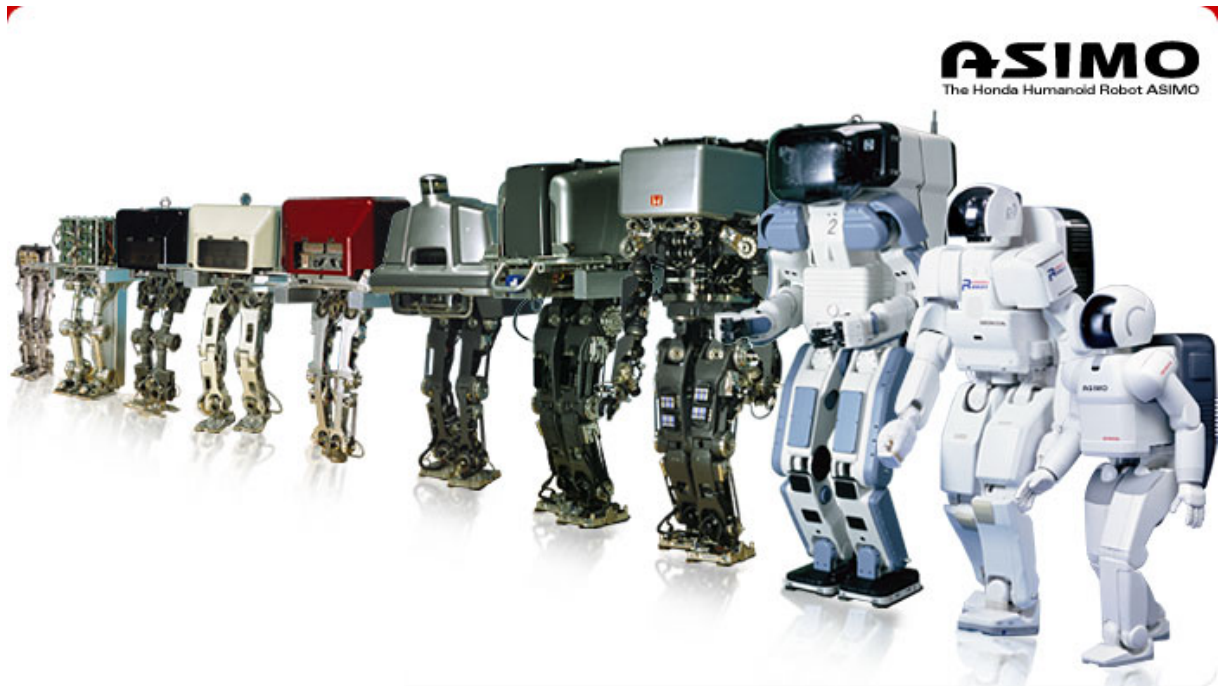


Figure 1.7 Evolution d'Asimo

A partir de 1986, Honda commence à développer les premiers prototypes de robots bipèdes avec une marche statique et une vitesse de pointe de 0.25 Km/h. La première marche dynamique est obtenue avec le modèle E2 à une vitesse de 1.2 Km/h. Avec le modèle P1 Honda réalise son premier humanoïde. C'est en 2000 que le robot prend le nom d'Asimo et son apparence actuelle.

Le 13 décembre 2005, Honda lance une nouvelle génération de son robot Asimo. Dans cette troisième version, Asimo peut atteindre les 6 km/h en quittant le sol pendant des intervalles de 0,08 secondes, et qu'aucune de ses jambes ne touche le sol alors qu'il avance sur une longueur de 50 mm. Il lui est aussi désormais possible de courir à 5 km/h en suivant une trajectoire circulaire de rayon 2,5 m.

Asimo a une hauteur de 130 cm, une largeur de 45 cm et une longueur 37 cm, avec un poids de 54 kg. Sa vitesse de marche normale est de 2,7 km/h et en portant un objet d'1kg il atteint 1,6 km/h. En marchant il a une autonomie de 40 minutes. Dans cette nouvelle génération, il a 34 degrés de liberté de mouvements :

Tête	3
Tronc	1
Bras	14
Jambes	12
Mains	4

Tableau 1.1 Degrés de liberté d'Asimo

D. HRP2 [hrp2]

Du haut de ses 1 mètre 54 et de ses 58 Kg, HRP-2, robot d'apparence humaine est équipé de caméras pour la vision et de capteurs d'efforts, d'attitudes pour la gestion de son équilibre, la planification et le contrôle de ses actions.

HRP-2 est sans équivalent dans la diversité des tâches dont il est capable d'effectuer. Le défi se trouve dans l'étude de l'autonomie des systèmes avec ses trois composantes : la perception, la décision et l'action.

Les activités de recherche dans le domaine de la robotique humanoïde sont en plein essor en France. Après le LAAS de Toulouse, au sein duquel est implanté HRP2 depuis l'automne dernier, le LIRMM (Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier), est en effet le deuxième laboratoire français à accueillir un robot humanoïde.

L'acquisition d'HOAP3, robot haut de 60 cm pour un poids de 8,8 kg, s'inscrit dans le cadre du JRL (Joint Japanese-French Robotics Laboratory), basé à la fois au Japon, à Tsukuba, et en France, à Toulouse et Montpellier. Disposant de 28 articulations motorisées, ce robot doté de nombreux capteurs (accéléromètres, gyromètres, télémètre infra-rouge, capteurs de pression, deux caméras), est conçu autour d'une plateforme logicielle entièrement ouverte (RTLinux) qui donne ainsi la possibilité à l'ensemble des chercheurs intéressés d'évaluer et valider librement leurs nouveaux développements théoriques concernant la modélisation, la commande, la vision ou l'apprentissage de ces robots. Complétant HRP2 déjà installée au LAAS, une plateforme plus réaliste car de "taille humaine" mais aussi plus complexe, HOAP3 va permettre aux chercheurs d'avancer très rapidement, sa mise en oeuvre étant simple et ne nécessitant pas de validations préalables sur simulateur.

Précisons que HOAP3 dispose d'un lien de communication sans fil grâce auquel il est possible d'effectuer des travaux de types téléopérations ou collaborations de robots mobiles. A terme, les chercheurs du LIRMM espèrent pouvoir disposer de plusieurs types de robots humanoïdes, leur objectif étant d'étudier la coopération entre robots.

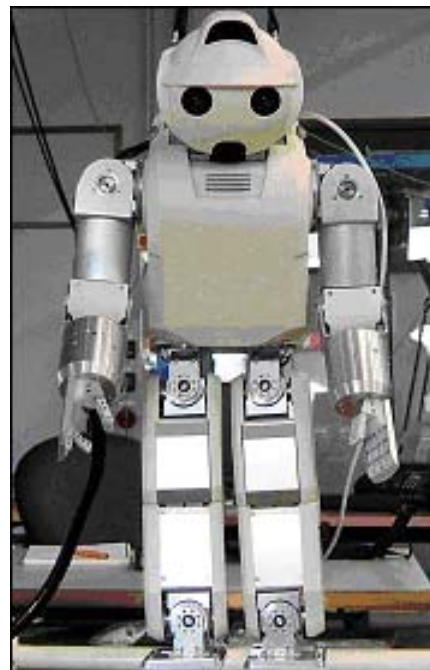


Figure 1.8. HRP2 et Hoap3

E. Repliee robot d'interaction

Repliee est un actroïde (contraction des mots acteurs et androïde) issu des travaux de l'équipe du professeur Hiroshi Ishiguro de l'Intelligent Robotic Laboratory - université d'Osaka. Il est fabriqué en collaboration avec les sociétés Kokoro et Advanced Media. Les actroïdes sont des robots réceptionnistes dont le rôle est de répondre aux questions des visiteurs et de les aider à organiser leur visite, actuellement en quatre langues : japonais, coréen, anglais et chinois.



Figure 1.9 Des actroïdes à l'image des humains

« *Un jour, les robots pourront nous duper en nous faisant croire qu'ils sont humains* », affirme le professeur Hiroshi Ishiguro [SHI 06], qui mène depuis plus de trois ans des recherches sur la communication avec les androïdes. Il est à la base des développements des prototypes d'Actroïd.

Il y a d'abord eu le "Repliee R1", prototype d'androïde enfant. Ensuite le Repliee Q1, androïde adulte, et aujourd'hui sa version améliorée le modèle Q2 qui présente 16 degrés de liberté rien que pour sa tête (3 pour les yeux 1 pour les sourcils, 1 pour les paupières, 1 pour les joues, 7 pour la bouche et 3 pour le cou). Aujourd'hui il est possible de louer des actroïds chez Kokoro Company ([kokoro]a Sanrio Group).

1.1.3 Les drones

Depuis une dizaine d'années, les avancées technologiques ont permis de concevoir et de construire des mini-avions ou mini-hélicoptères avec des capacités toujours plus développées pour réaliser des vols autonomes. Ces appareils sont connus sous le nom de drones. La variété des drones est large et n'est pas exclusive des systèmes volants traditionnelles, ils peuvent aller du micro aux super-structures telle que la libellule et les ballons dirigeables.

A. La libellule [Silmach]

Les ailes battantes de la libellule constituent son système de propulsion alternatif pour les mini et micro-aéronefs. La recherche dans ces domaines se base sur la reproduction du battement des ailes des oiseaux ou des insectes. En France, le laboratoire Femto de Besançon et la Société Silmach travaillent sur un projet de libellule artificielle d'une centaine de grammes.

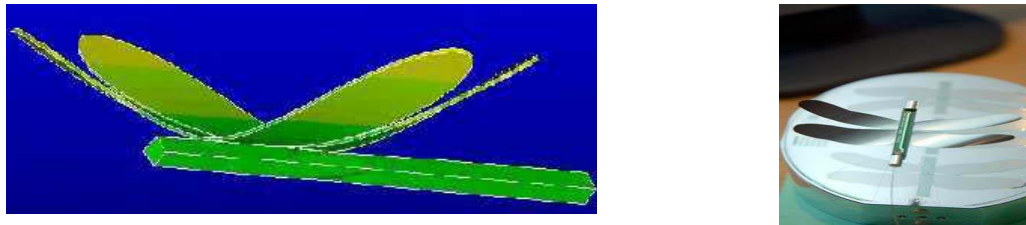


Figure 1.10 La libellule

B. Des ballons dirigeables [BAL, LAC 05, LOZ 05]

De très nombreux projets de ballons dirigeables de grandes tailles dédiés au transport de charges ont été initiés (ainsi le projet Titan, ballon lenticulaire de 188 mètres de diamètre pour un volume 900.000 m³ mené au CNRS pendant les années 1970). Ce type d'engin pourrait permettre de transporter des charges pesant jusqu'à plusieurs dizaines de tonnes entre le départ et l'arrivée, et sans besoin d'infrastructure spécifique. Le ballon ne doit pas nécessairement atterrir pour prendre et déposer la charge. Le projet industriel CargoLifter initié en Allemagne en 1997 est sans doute un des projets qui a le plus marqué les esprits, tant l'ambition et les investissements financiers étaient importants. L'objectif était de construire un engin de 260 mètres de long, capable d'embarquer des charges de 160 tonnes. Mais le projet a malheureusement fait faillite en 2002, après que de nombreux retards aient été pris dans le développement.



Figure 1.11 Illustrations du CargoLifter et du Skycat d'ATG

Les illustrations de la figure 1.11 ne sont que des prototypes en conception, aucun modèle d'une aussi grande taille n'a encore vu le jour.

1.1.4 Aide à la personne

A. Aide à la personne : interface Tactile Portable pour le Handicap Visuel [VEL 05]

Les interfaces à stimulation tactile sont utilisées principalement pour la transmission de l'information aux déficients visuels. Mais leurs vraies possibilités ont été relevées récemment lorsque le sens du toucher est devenu l'une des modalités d'interaction avec les ordinateurs : réalité virtuelle, robotique, télémanipulations et divertissements. Depuis peu, on envisage leur utilisation pour la réhabilitation et à des fins thérapeutiques.

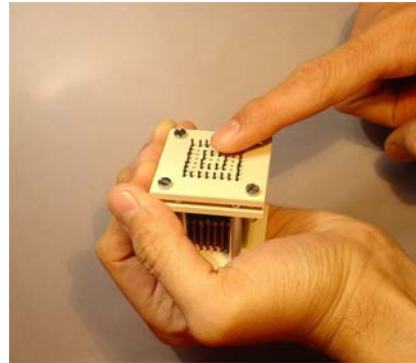
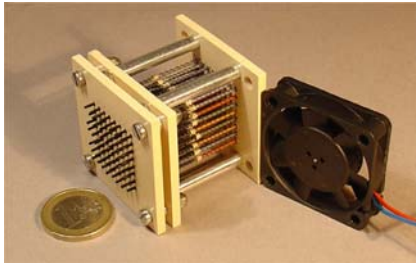


Figure 1.12 Interface tactile portable pour le handicap visuel

Interface tactile portable basée sur des microactionneurs en alliage à mémoire de forme

Projet : Lunettes Intelligentes du Laboratoire de Robotique de Paris, Université Pierre et Marie Curie – CNRS, a plusieurs avantages :

- Conception et réalisation simple
- Haut niveau d'intégration
- Portabilité
- Bon rapport force/poids du matériau
- Prix raisonnable

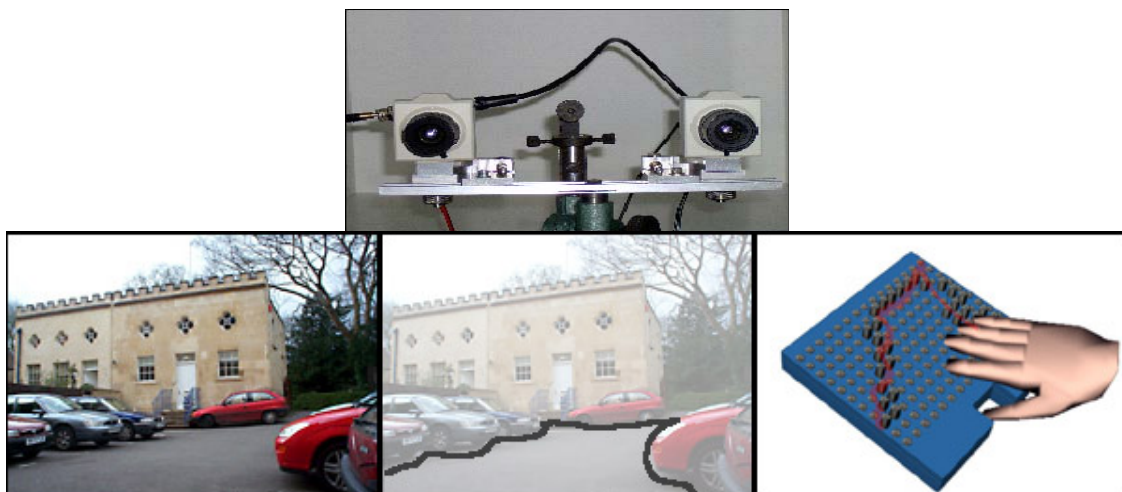


Figure 1.13 Application de l'interface tactile pour toucher avec les doigts une représentation des obstacles de l'environnement

B . Robotique médicale et chirurgicale [DAVINCI, MOR05]

Le corps médical est traditionnellement à l'écoute des évolutions technologiques. Celui-ci a d'ailleurs entrepris d'utiliser des systèmes robotiques pour des interventions chirurgicales. La précision et la dextérité des robots suscitent l'intérêt de cet outils. Cela a conduit au développement d'un certain nombre de dispositifs robotiques.

La commande référencée capteur est une modalité permettant à un système robotique de prendre en compte une interaction avec son environnement pour adapter son comportement en temps réel. Cette technique trouve de nombreuses applications dans les domaines de la médecine, de la chirurgie et de l'environnement. Un patient recevant des soins présente des caractéristiques géométriques et mécaniques variables dans l'espace et dans le temps.



Figure 1.14 Systèmes passifs pour la comanipulation en neurochirurgie

Dans le domaine médical, les systèmes de télémanipulation sont utilisés avec deux principaux objectifs :

- permettre des soins et examens à longue distance.
- permettre l'immersion du chirurgien dans le champ opératoire

Le principe de la comanipulation pour la chirurgie endoscopique permet un ressenti des efforts appliqués par l'instrument sur les organes.

Le robot DA VINCI est un de ces robots utilisés actuellement dans la chirurgie mini invasive. Il permet une immersion 3D dans le champ opératoire. Il donne aussi une bonne précision avec une réduction des mouvements et son système de commande élimine les tremblements.

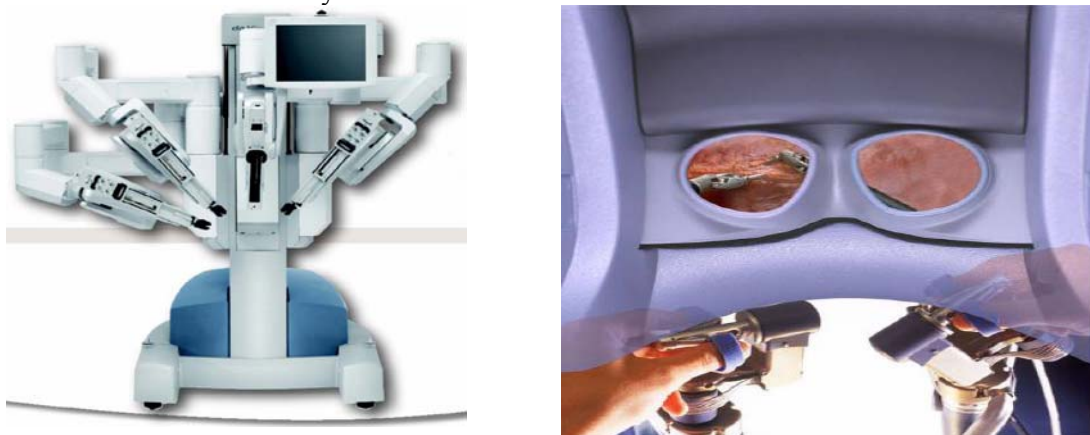


Figure 1.15 Le robot DA VINCI et immersion 3D

C. Assistance à la conduite automobile [MAM 05]

Des méthodes de commandes robustes appliquées au développement d'assistance au contrôle latéral et longitudinal des véhicules sont conçues pour une assistance active aux conducteurs en utilisant des moyens de perception, de localisation ou de communication entre les véhicules ou avec l'infrastructure.

A titre d'exemple, une caméra frontale détectant les lignes blanches permet de développer une fonction d'assistance au maintien de voie. Un radar fournit une mesure d'interdistance et de vitesse relative et permet d'envisager le développement d'un système de gestion des inter-distances et d'anti-collision. De même, une localisation à l'aide d'un GPS peu précis associée à une cartographie suffit pour limiter la vitesse à l'approche d'un virage.

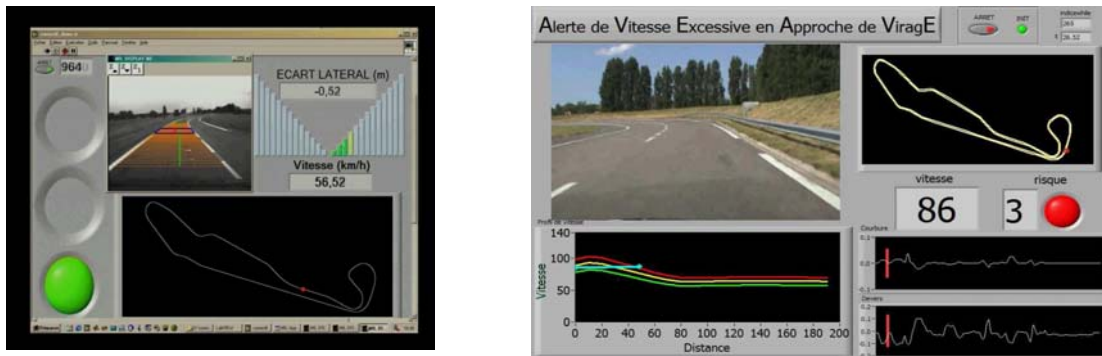


Figure 1.16 Interface d'assistance au maintien de voie par vision frontale, et celle d'affichage de vitesse excessive

Des grands constructeurs automobiles s'impliquent dans la recherche de la robotique et de l'intelligence artificielle. On assiste actuellement à un tournant dans la recherche de voitures consommant moins d'énergie et plus performantes.



Figure 1.17 Le prototype MTRC

MTRC, voiture de course à pile à combustible, est le prototype du groupe Toyota. Cette voiture utilise un système de 4WD (4 roues motrices) dont le moteur est dans les roues, c'est-à-dire qu'un moteur électrique est intégré dans chacune des roues. En utilisant sa pile à combustible comme source d'énergie il ne rejette que de l'eau. La voiture est équipée d'un système de suspension à contrôle électronique qui mesure constamment l'état de la route et ajuste la hauteur de roulement ainsi que les paramètres en fonction de l'environnement. Ses pneus intelligents s'adaptent également aux conditions de la route grâce à des capteurs intégrés qui assurent une adhérence adéquate et optimale [Toyota].

1.1.5 Les nouveaux défis : la robotique collective et multi-robots

Les nouveaux défis exposés dans ce paragraphe ne correspondent pas à une liste exhaustive mais plutôt aux limites du champs du travail que nous présentons.

A. La RoboCup [robocup]

RoboCup a été créé pour favoriser la recherche dans le domaine de la robotique et de l'intelligence artificielle. L'objectif de la RoboCup est de construire une équipe de robots capable de concurrencer la meilleure équipe humaine en 2050. Pour ce faire une feuille de route a été déterminée qui fixe tous les objectifs scientifiques et technologiques devant être résolus [BUR 02].



Figure 1.18 Compétition, Robocup

De rapides progrès ont été faits. En 1997, les robots ne se déplaçaient pas très bien et avaient des difficultés à trouver la balle, maintenant les robots la trouvent aisément. Ils se déplacent très rapidement et commencent à montrer des comportements de travail d'équipe.



Figure 1.19 Robots footballeurs

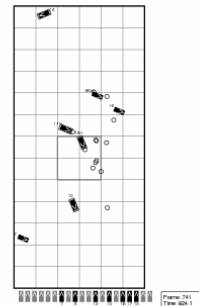
Robocup a favoriser l'échange entre chercheurs. Le savoir-faire de ces derniers a un impact dans différents domaines tels que dans le divertissement, l'aide à domicile ou au bureau.

B. Les travaux de Mataric [MAT 92, MAT 95]

En prenant le travail de C. Reynolds [bois] et en les adaptant à de vrais robots, M. Mataric, du laboratoire de R. Brooks au MIT, a créé une véritable équipe robotique. La coordination de ses applications est très difficile à mettre en œuvre sur les robots à cause de leurs mécanismes, la limitation de leurs capteurs. L'objectif est de s'assurer que les robots ne vont pas aller trop vite et s'écartent entre eux, ou inversement, que les groupes qu'ils forment ne soient pas trop compacts et entravent leurs mouvements.



Flocking



Foraging

Figure 1.20-a Système multi-robot et des simulation utiliser par M. Mataric

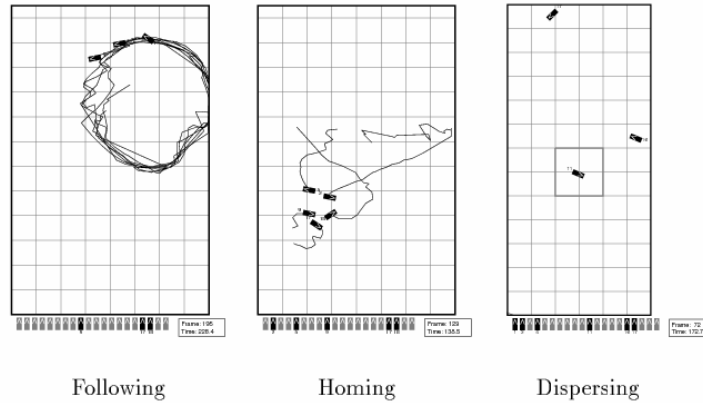


Figure 1.20-b Système multi-robot et des simulation utiliser par M. Mataric

En ajustant les robots avec de bons paramètres internes, M. Mataric a démontré et vérifié qu'à condition que ces facteurs soient bien ajustés, il est possible de faire émerger des comportements en groupe. Les robots ont prouvé leurs capacités d'évitement de collision dans un groupe, de suivie, d'agrégation, de dispersion, de flocking (mouvement de foules), d'errance et ont appris à faire des forages.

C. CEBOT [cebot]

Le domaine de la robotique mobile multi-robot remonte à plus d'une dizaine d'années de travail expérimental dans ces systèmes. Les aspects théoriques importants des mécanismes de coordination de multi-robot ont jusqu'ici été en grande partie non traités. Cependant, en dépit de la recherche faite, le champ manque jusqu'ici d'une base théorique pouvant expliquer ou prévoir le comportement d'un système de multi-robot Fukuda Cebot [GRE et MAT 2004].

CEBOT est une abréviation de « celular robotics system » qui se compose de plusieurs unités robotiques avec une fonction simple appelée cellule. Le système CEBOT peut se reconfigurer entièrement selon des tâches données et de l'environnement. Le concept du CEBOT est basé sur l'organisation biologique construite par d'énormes cellules normalisées. CEBOT a une stratégie de vie artificielle pour reconfigurer ses actions, comme la modélisation à partir des comportements coopératifs comme ceux des fourmis.

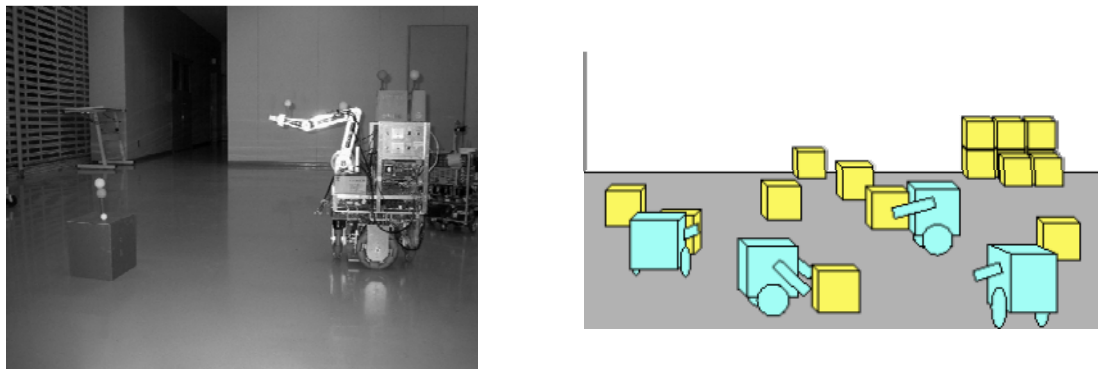


Figure 1.21 Le système CEBOT et possible application

Ceci nous renvoie à d'intéressants problèmes de recherches tels que la planification décentralisée dynamique, la distribution dynamique et le système de commande coordonné. Plusieurs applications peuvent être envisagées dans le cadre du système CEBOT telles que des applications dans l'espace, agricoles, médicales, et de construction, y compris l'inspection, la surveillance et le système réparti de surveillance.

1.1.6 Les nouveaux défis : la robotique modulaire

La robotique modulaire est un domaine où le robot n'est pas considéré comme un ensemble fixe mais comme un assemblage de plusieurs parties indépendantes. Ces éléments ou modules peuvent accomplir des tâches différentes et interagir pour former un organisme complexe.

A. Enjeux de la modularité [VIR 02]

Les composantes de la robotique modulaires sont des questions essentielles et largement reconnues en tant que tels. Beaucoup de groupes de chercheurs ont développé des méthodes innovatrices pour les composants d'intégrations, mais ces approches individualistes n'ont pas semblé être acceptable pour la communauté scientifique. Par conséquent, aucun progrès significatif n'a été accompli.

G. S. Virk propose d'adopter une approche à la modularité afin de subdiviser le processus de conception de robot dans un format modulaire où les différents composants peuvent être liés à d'autres modules pour former un système global. Ceci conduit à déterminer la manière dont les modules doivent être liés les uns aux autres. Après recherches et discussions, on a établi que six interactions existent pour cette interconnectivité : 1) puissance 2) bus de données d'ordinateur, 3) compatibilité mécanique, 4) signaux analogiques, 5) signaux numériques, et 6) l'environnement. Le schéma de la figure 1.22 montre plus clairement le concept ; les fils montrent les contraintes requises pour les modules spécifiques.

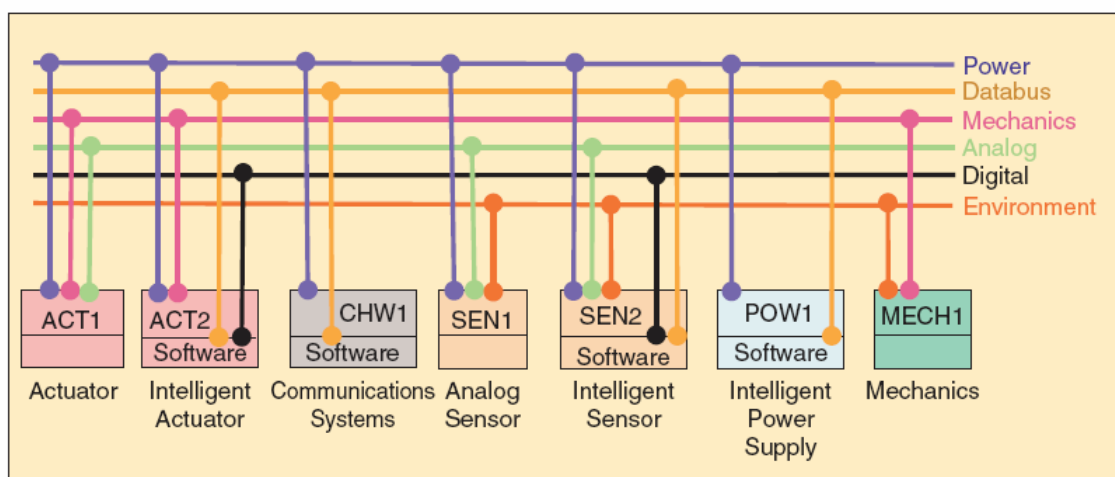


Figure 1.22 Modularité et interaction par G. S. Virk

B. RMMS [CHR 07, RAJ 00]

En théorie, les tâches qu'un robot doit accomplir peuvent être modifiées simplement avec le chargement d'un nouveau programme dans son contrôleur. Cependant, dans la pratique c'est rarement le cas. Le système modulaire de manipulateur de CMU reconfigurable (RMMS) a été conçu pour adresser les problèmes liés aux manipulateurs conventionnels de configuration fixe.

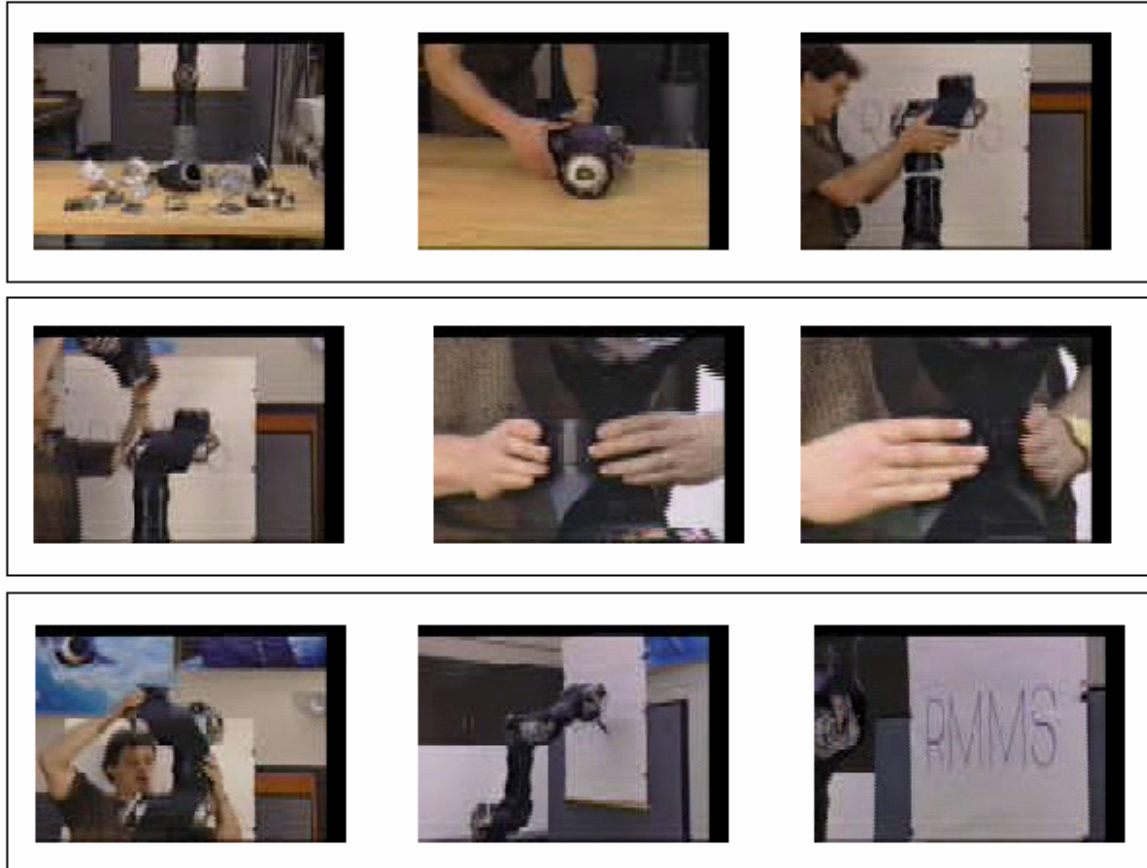


Figure 1.23 Séquence de montage d'un bras du système RMMS

Le RMMS utilise des actions des modules interchangeables de joints (actionneurs) et de liens de différentes caractéristiques de taille et d'exécution. La modularité dans la conception mécanique, électrique et électronique permet à l'utilisateur de concevoir le manipulateur optimal et de le changer manuellement. Le RMMS prolonge également le concept de la modularité pour inclure les algorithmes de commandes et pour charger le logiciel de planification.

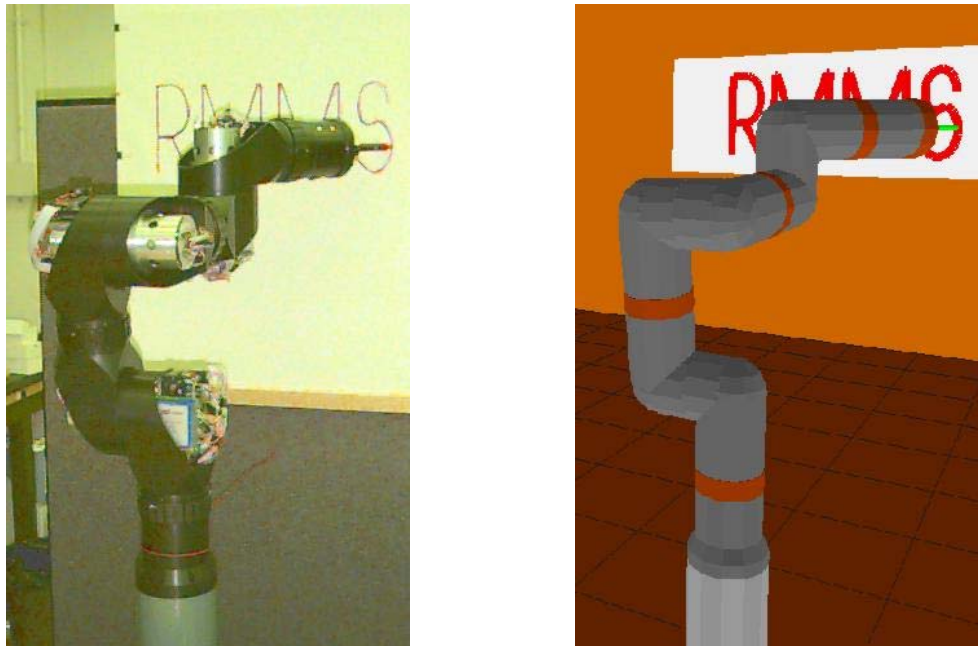


Figure 1.24 Le RMMS configuré en bras robotique et son simulateur

En assemblant les composantes du dispositif des degrés de liberté se créent dans les joints permettant le mouvement relatif entre les pièces. Dans des assemblages complexes les interactions entre différents degrés de liberté peuvent entraîner un comportement différent de celui prévu.

C. DIGITAL Clay [digital]

Ces modules sont en forme de dodécaèdre rhombique et n'ont pas d'actionneurs. Les modules sont déplacés par les hommes comme une interface 3D de manipulation.



Figure 1.25 Prototype du système Digital Clay

Ce projet adresse la question « quelles sont les limitations sur le nombre de modules pour qu'un système de robotique reconfigurable modulaire soit utile ? ».

Comment le nombre de modules affecte :

- La polyvalence du système, selon les différentes formes qu'il peut assumer.
- La robustesse et comment gérer l'auto-réparation et la redondance.
- Le coût de production et les facteurs d'économies dans la production.

1.1.7 Les nouveaux défis : la robotique modulaire reconfigurable

A. SWARM-BOT [swarm, GRO 06, TRI 05]

Un système robotique d'essaim est normalement caractérisé par beaucoup d'individus, chacun avec une connaissance partielle/limitée du modèle global dont il constitue un élément. Dans de tels systèmes les décisions collectives posent de grandes problématiques.

Un swarm-bot comporte des robots mobiles autonomes appelés les s-bots. Les s-bots peuvent agir de façon individuelle ou sous la forme d'un collectif auto reconfigurable dans un swarm-bot en utilisant leurs pinces.

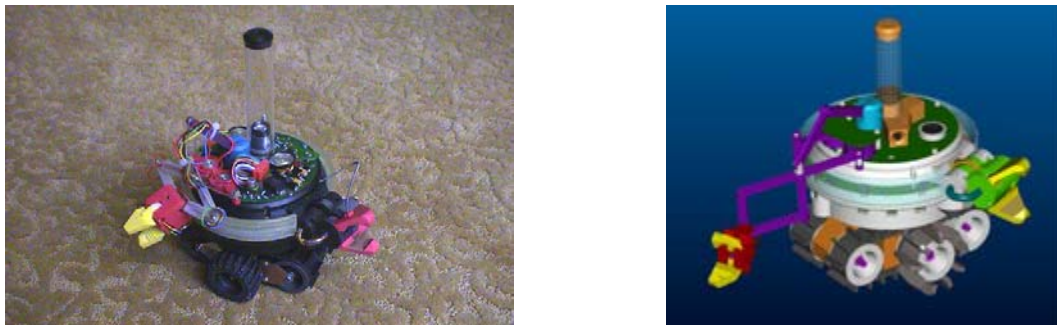


Figure 1.26 Le S-bot, réalisation matériel et sont modèle CAO

Les swarm-bot sont des robots ayant comme particularité l'auto-assemblage de ses modules. C'est un concept partant de la robotique distribuée et qui se trouve à l'intersection entre la robotique collective et reconfigurable.

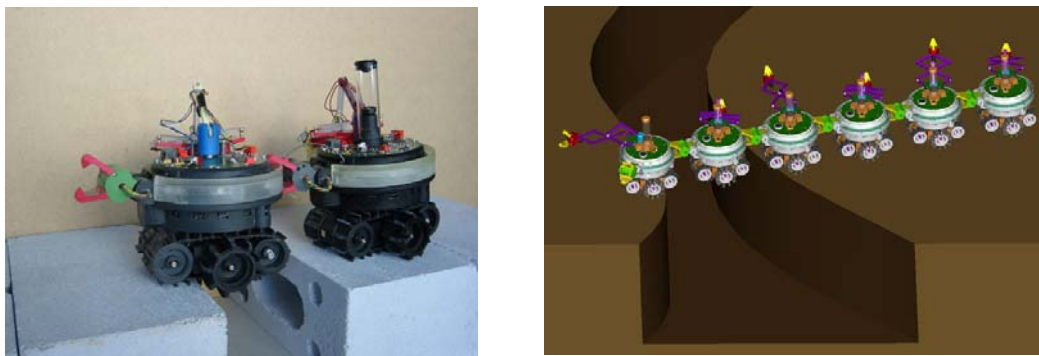


Figure 1.27 Le Swarm-bot franchissant un fosse

L'objectif scientifique principal du projet de swarm-bot est d'étudier une approche de la conception et de l'exécution des objets partant de l'auto-organisation et de l'auto-reconfiguration. Cette approche trouve ses racines théoriques dans des études récentes sur l'intelligence d'essaim montrées par les insectes sociaux et d'autres sociétés animales [REM 82] [KON 84].



Figure 1.28 Séquence d'un s-bot avec un obstacle infranchissable, reconfiguration et franchissement de l'obstacle

La séquence de la figure 1.28 montre qu'un s-bots seul ne peut pas franchir un obstacle, mais après une reconfiguration en Swarm-bot de 2 s-bots en ligne l'obstacle est franchissable.

Le travail est sous l'effet coordonné de quatre activités principales : la conception et l'exécution du matériel, la conception et l'exécution du simulateur, la conception et l'exécution du système de commande du Swarm-Bot, l'intégration, l'essai et l'évaluation des résultats de ces activités. Basés sur de telles caractéristiques des prototypes seront développés, évalués et documentés pour chacun des trois composants: s-bots (matériel), simulation (logiciel), et mécanismes de commande basée sur l'intelligence collective (logiciel). Les mécanismes de commande sont testé sur le simulateur et les réalisations de matériel.

B. I-Cubes [I-cubes]

I-Cubes fait partie des systèmes robotiques modulaires auto-reconfigurables. Dans ce système, chaque module est commandé indépendamment avec un lien ayant la capacité de se relier et de se décrocher de la face d'un autre cube.

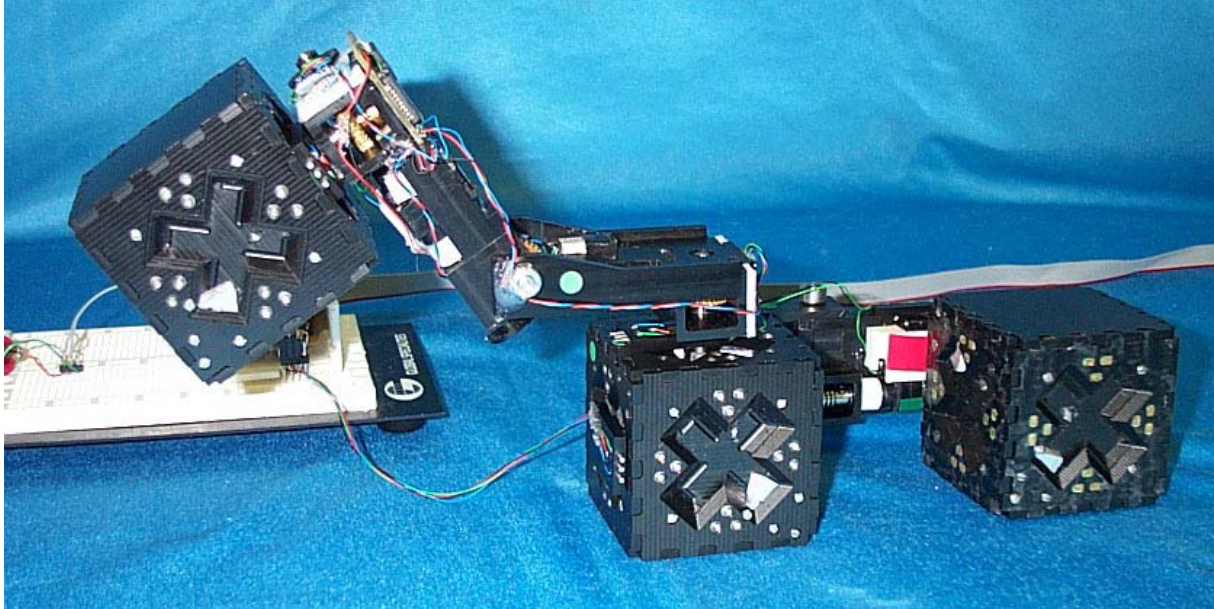


Figure 1.29 Système I-Cubes

Tandis qu'attachés à un cube sur une extrémité les liens sont également capables de se déplacer avec un cube différent attaché à l'autre extrémité. Nous supposons que tous les modules actifs (liens) et passifs (de cube) sont capables de communiquer localement l'information à leurs modules voisins.

C. ATRON [CHR 05, hydra]

Le projet HYDRA propose de concevoir de nouveaux objets élaborés de façon facile en employant une architecture standard ouverte composée de modules simples, de raccords simples et d'interactions simples. Ceci signifie que les systèmes comportant des centaines de blocs fonctionnels de base peuvent devenir auto-reconfigurables pour la réparation automatique.

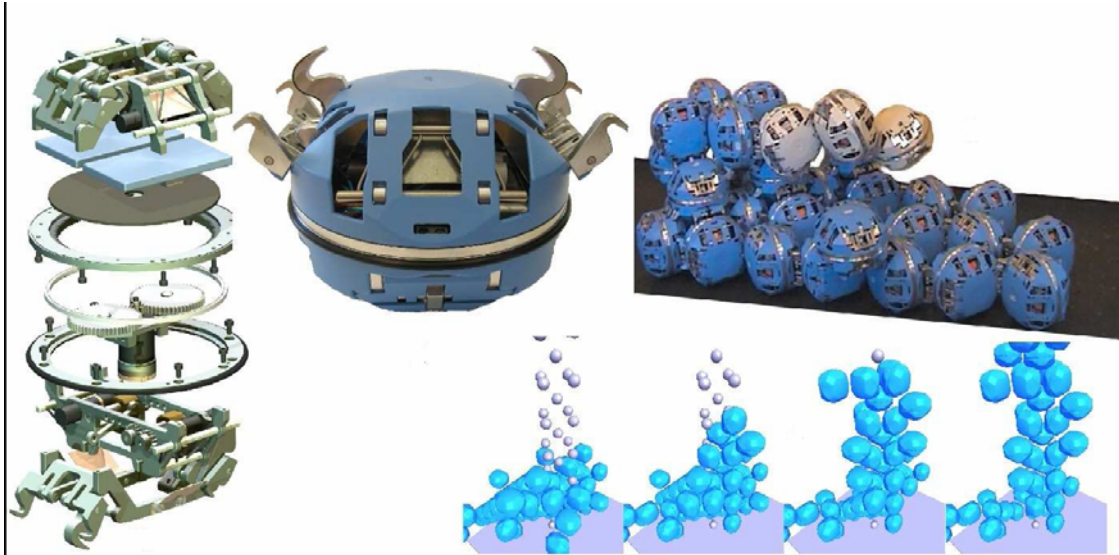


Figure 1.30 Système reconfigurable : ATRON

En dépit de leurs avantages, les méthodes de génie industriel traditionnelles sont inflexibles et incertaines une fois comparée aux organisations biologiques. Les chercheurs d'HYDRA projettent de développer une nouvelle architecture pour la conception d'objets inspirés des cellules biologiques basées sur des modules simples. Le potentiel de la nouvelle conception sera exprimé par la construction de systèmes robotiques qui peuvent changer leurs morphologies, résultant des objets de grandes flexibilités, robustes et applicables au-delà des perspectives conventionnelles de notre technologie actuelle.

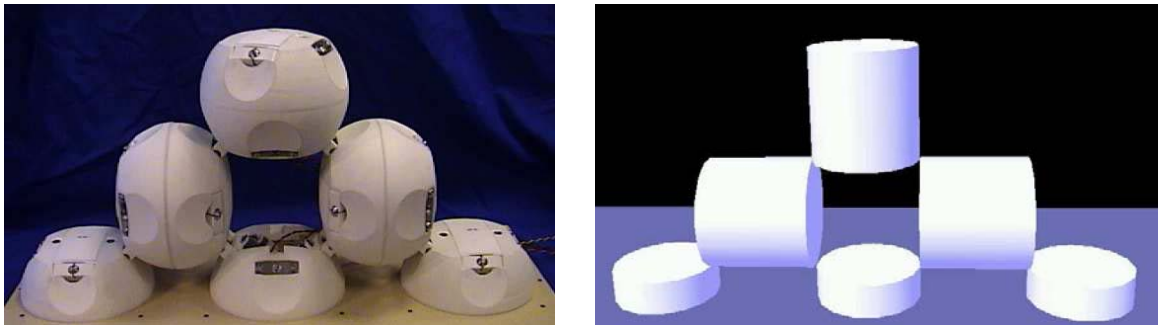


Figure 1.31 Système reconfigurable ATRON et simulation

Le consortium d'HYDRA se compose :

- Groupe d'AdapTronics, institut de Maersk, université du Danemark méridional
- Groupe mobile de robots, université d'Edimbourg
- Laboratoire d'AI, université de Zurich
- Développement de plateforme de LEGO

D. POLYBOT [Polybot]

Conçu pour des reconfigurations successives, PolyBot est un robot composé de nombreux modules répétés. Chaque module est pratiquement un robot seul, ayant un ordinateur, un moteur, des sondes et une capacité à s'attacher à d'autres modules. Ces modules s'attachent ensemble en forme de chaînes pouvant être employées comme un bras, une jambe ou un doigt selon le programme chargé dans son CPU.

Les séquences d'accouplement requièrent encore une aide manuelle. La reconfiguration du robot est montré ci-dessous:

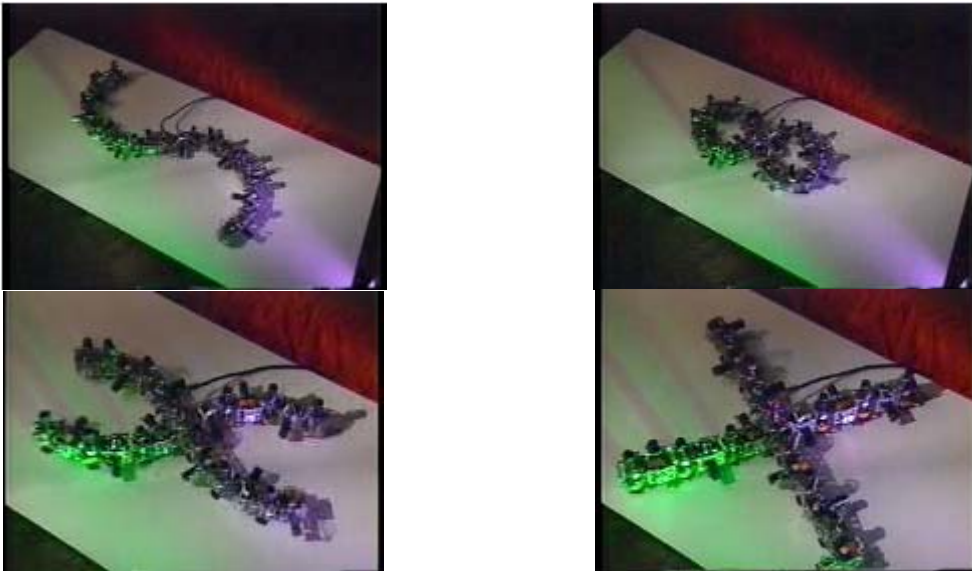


Figure 1.32 Séquence de reconfiguration système PolyBot

La reconfiguration dynamique permet au robot d'être très souple en changeant sa forme pour atteindre une meilleure configuration dans sa tâche attribuée.

E. PROTEO [proteo, BOJ 01]

Les modules en forme de dodécaèdre rhombique sont conçus pour se reconfigurer par le déplacement entre ses arêtes. Ce système a été seulement jusqu'ici simulé.

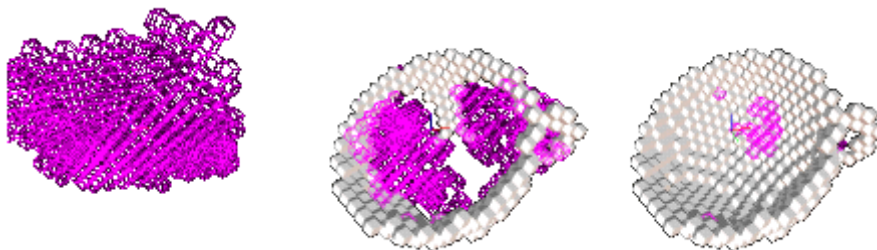


Figure 1.33 Séquence simulée de système PROTEO

La conception des modules limite Proteo au niveau de ses actionnements. Un module peut seulement

s'arrêter avec une configuration définie de son voisinage. Un module peut se déplacer d'une position à une autre dans son voisinage à travers une reconfiguration qui requière qu'un degré de liberté. Ces contraintes réduisent la complexité du calcul informatique.

Les modules de modules peuvent seulement communiquer localement avec leurs voisins.

F. Système de transfert flexible (FTS) à travers l'auto-reconfiguration [TAK 03]

FTS est un type de système modulaire de transfert bidimensionnel composé de modules dotés d'une capacité d'auto-reconfiguration. L'auto-organisation des chemins de transferts appropriés pour la reconfiguration physique des modules est basée sur des informations locales.

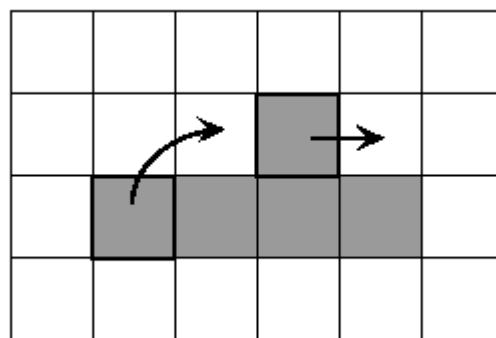
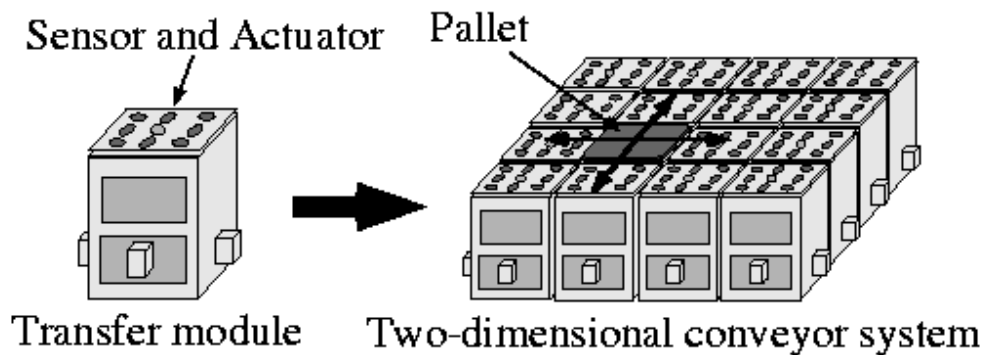


Figure 1. 34 FST, un système de transfère flexible

Les simulations numériques ont révélé que la configuration désirée des modules peut se construire par une méthode employant seulement l'information locale des modules voisins.

Le système flexible de transfert (FST) est une approche de la robotique modulaire reconfigurable proposée par le Fukuda Laboratory.

G. Le projet M-TRAN [m-tran]

Le système de M-TRAN peut changer sa structure, à trois dimensions, et son mouvement afin de s'adapter à l'environnement. Dans la configuration de petite taille, il marche sous une forme de robot à jambes, puis se métamorphose en un serpent pour, par exemple, passer dans les espaces étroits. Une grande structure peut graduellement changer sa configuration. Elle peut produire des marcheurs à multiples jambes. Les applications possibles du M-TRAN sont l'exploration autonome dans l'environnement inconnu telles que des explorations planétaires, la recherche et les opérations de secours dans des secteurs touchés par des désastres.

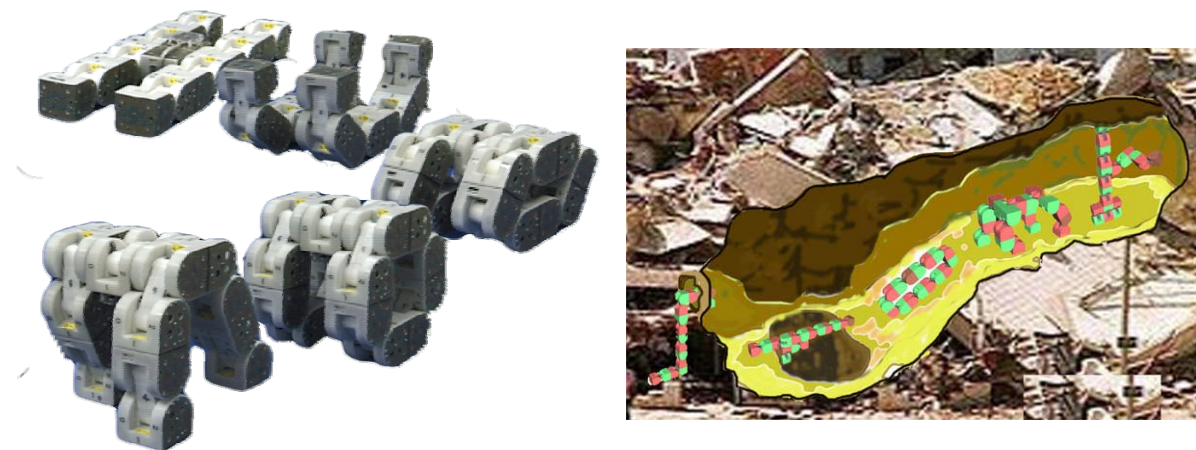


Figure 1.35 Le système M-TRAN

H. MAAM [maam]

Le VALORIA, laboratoire de recherche en informatique de l'Université de Bretagne Sud, développe ses activités sur la communication homme-machine, la maîtrise du développement logiciel, la conception d'application à base de composants et la robotique. C'est dans ce dernier cas que s'est développé une partie du projet MAAM.

Projet MAAM. Son nom est un acronyme récursif pour Molécule = Atome | (Atome + Molécule). Plusieurs équipes de recherche en électronique, robotique et informatique y participent :

- Le LESTER et le VALORIA de l'Université de Bretagne Sud,
- Le LIP6 et LRP de l'Université Paris VI,
- Le GREYC de l'Université de Caen,
- Le LIST du CEA.

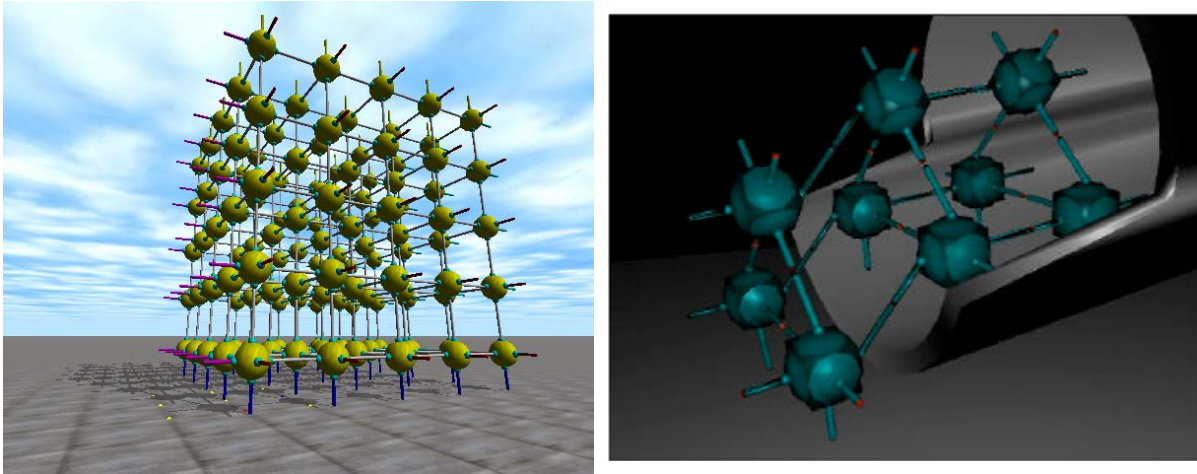


Figure 1.36 Le système MAAM

Le projet MAAM a pour objectif de définir, spécifier, concevoir et réaliser un ensemble d'atomes robotiques capables de s'assembler en une molécule qui pourra, par reconfiguration successive, réaliser une tâche donnée. Dans le cas du projet MAAM, un atome sera une structure mécanique à six pattes, chacune d'elles pouvant se solidariser deux à deux avec d'autres atomes. Les six pattes sont réparties autour d'un noyau suivant les trois directions de l'espace comme sur la figure ci-dessous.

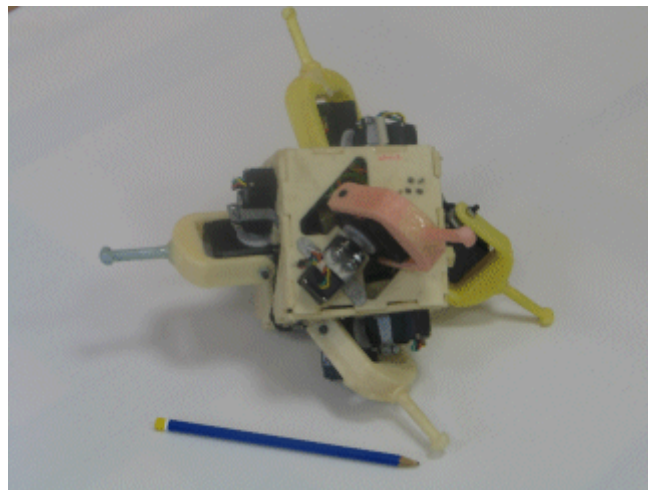


Figure 1.37 Réalisation d'un atome

Les exemples les plus évidents de systèmes auto-organisés sont issus de la physique. C'est d'ailleurs dans ce domaine que le terme est apparu pour la première fois. L'auto-organisation est aussi présente en chimie où elle a souvent été synonyme d'auto-assemblage. Le concept d'auto-organisation est aussi central dans les systèmes biologiques, que ce soit au niveau cellulaire ou social.

Un des buts de la robotique reconfigurable du projet MAAM est d'arriver à obtenir des robots composés de plusieurs unités qui par le biais de reconfigurations dynamiques peuvent évoluer à l'instar des cellules vivantes s'auto-organisant pour former des organes, des tissus, etc. De même, la reconfiguration des modules du robot permettrait l'adaptation à l'environnement et de varier sa forme selon les tâches qu'on lui attribue.

Dans cette étude le comportement retenu pour le déplacement collectif du robot reconfigurable se fait par de reconfigurations successives de ses modules. Bien entendu cette contrainte n'est pas absolue et

d'autres études ont été menées en la relâchant, comme le travail de V.Montreuil [MON 03]. On pourrait envisager de relâcher cette contrainte mais la reconstruction physique d'une molécule par reconnexion entre deux atomes robots distant est une opération longue car elle exige de la communication, de la localisation et de la coopération.

1.1.8 Système complexe

De nombreux systèmes sont constitués d'un grand nombre d'entités en interaction. On les qualifie de complexes lorsqu'un observateur ne peut prévoir le comportement ou l'évolution d'un tel système par un raccourci de calcul. Les cellules nerveuses de notre cerveau, ou une colonie de fourmis sont des systèmes complexes car le seul moyen de connaître l'évolution du système est de faire l'expérience.

Les systèmes complexes sont donc un contre-exemple au réductionnisme. Malgré une connaissance parfaite des composants élémentaires d'un système, il est impossible de prévoir son comportement autrement que par l'expérience ou la simulation. Cet écueil ne vient pas seulement de nos limites de calcul, mais plus généralement de l'impossibilité de mettre le système en équations.

A l'opposer les systèmes compliqués sont décomposables en entité individuellement compréhensible et linéaire, ayant comme résultat un comportement prévisible.

1.2 IA et Système multi-agents

Les systèmes multi-agents correspondent à la programmation informatique des systèmes complexes. Le génie logiciel a pour objectif de programmer des systèmes compliqués.

1.2.1 Définition des systèmes multi-agents

Le déplacement collectif peut être envisagé à travers la conceptualisation de systèmes multi-agents résultant de l'ensemble d'agents situés dans un environnement donné et en interagissant selon une certaine organisation. Jacques Ferber [FER 95] a précisé la définition d'agent comme étant une entité physique ou virtuelle :

- capable d'agir dans un environnement,
- communicant directement avec d'autres agents,
- muant par un ensemble de tendances (sous forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser),
- détenant des ressources propres,
- capable de percevoir (mais de manière limitée) son environnement,
- ne disposant que d'une représentation partielle de cet environnement (et éventuellement aucune),
- possédant des compétences et offre des services,
- pouvant éventuellement se reproduire,
- dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.

Dans notre cas nous allons retenir certaines de ces connotations : la capacité d'agir dans un environnement, de se muer pour satisfaire ses objectifs individuels ayant une fonction à optimiser et de percevoir de manière limitée son environnement. C'est avec ces conditions que nous devons avancer

vers l'intégrité du robot reconfigurable dans le déplacement collectif de ses modules.

1.2.2 Des catégories d'agents

J. Ferber [FEB 95] poursuit dans la précision des agents et établit une classification des agents selon deux critères : les agents cognitifs ou réactifs d'une part, et comportement téléonomique ou réflexe d'autre part.

La distinction qu'il fait entre cognitif et réactif tient essentiellement de la représentation du monde dont dispose l'agent. Si l'individu est doté d'une représentation symbolique du monde à partir de laquelle il est capable de formuler des raisonnements, on parlera d'agent cognitif. Tandis que s'il ne dispose que d'une représentation sub-symbolique, c'est à dire limitée à ses perceptions, on parlera d'agent réactif. Cette distinction cognitif/réactif correspond à deux écoles de pensée des systèmes multi-agents. La première soutient une approche de famille d'agents "intelligents" devant collaborer, avec une perspective plus sociologique. La deuxième étudie la possibilité de l'émergence d'un comportement "intelligent" d'un ensemble d'agents non-intelligents (type fourmis).

La distinction entre comportement téléonomique et réflexe sépare les comportements intentionnels dans l'explicite de la poursuite de buts des comportement, liés à des perceptions. Les tendances des agents peuvent ainsi être exprimées explicitement dans les agents ou au contraire provenir de l'environnement. Il construit un tableau regroupant les différents types d'agents :

	Agents cognitifs	Agents réactifs
Comportement téléonomique	Agents intentionnels	Agents pulsionnels
Comportement réflexe	Agents "modules"	Agents tropiques

Tableau 1.2 Les différents types d'agents

Nous nous plaçons dans la perspective d'utiliser des agents réactifs avec un comportement réflexe pour la modélisation de l'intégrité de robots modulaires reconfigurables puisque il nous permet d'assumer avec plénitude la particularité de sa simplicité. La suite chronologique de notre étude nous fera prendre des cheminements de modélisations qui diffèrent de ce concept, mais le principe restera de garder de la simplicité dans la commande des actionnements du système.

1.2.3 Les systèmes multi-agents

A. L'approche réactive

Les modèles d'agents réactifs étant particulièrement nombreux nous n'en retiendrons que deux traits distinctifs. Un agent réactif ne possède pas de représentation symbolique de son environnement et de lui-même. Par conséquent, il n'est pas censé pouvoir contrôler son propre comportement. Son contrôle est entièrement pris en charge par un mécanisme rigide qui se sert du résultat de la perception pour déclencher une action (automate cellulaire). Ce mécanisme n'est modifiable ni par l'action ni par la perception de l'agent. En ce sens, le contrôle du comportement est donc en partie effectué par l'environnement (c'est la non moins fameuse "physical grounding hypothesis" [BRO 91]).

B. Emergence de comportement

Le principe d'intelligence artificielle est pour ainsi dire appliqué aux systèmes reconfigurables comme le principe de protection dans l'émergence

Le principe de l'émergence est basé dans l'apparition d'une propriété qui n'a pas été précédemment observé comme caractéristique fonctionnelle du système. Généralement, les propriétés de plus haut niveau sont considérées comme émergentes. Une automobile et ses propriétés sont la résultante de l'émergence de l'assemblage de ses pièces. Cette propriété disparaît si les pièces sont démontées et juste placées dans un tas. On peut énumérer trois aspects impliqués dans l'émergence.

Tout d'abord, les propriétés émergentes n'existeront plus si le niveau plus bas est enlevé. Ensuite, les nouvelles propriétés ne sont pas des agrégats et ne sont pas simplement les résultats prévisibles additionnés des propriétés de ses parties. Enfin, il devrait y avoir de la causalité, ainsi les propriétés émergentes ne sont pas epiphenomenal. Ceci signifie que les propriétés de plus haut niveau devraient avoir des effets causals sur les plus bas.

C. Le principe de protection

Les lois émergentes sont dans une large mesure découplées de leurs bases présumées. Elles restent stables sous variations considérables de leurs bases.

En éthologie et en entomologie, un comportement émergent est un effet global qui résulte de l'application de règles locales, généralement très simples. Par exemple, il est possible de simuler l'activité très complexe d'une fourmilière à l'aide de règles très simples concernant chaque fourmi [manta].

En science des systèmes comme la cybernétique, l'accent est mis sur le phénomène d'émergence. Ce n'est plus une somme de comportements simples mais le résultat d'une interaction entre ces comportements et la complexité du système.

Nous proposerons donc une approche distribuée à ce problème qui s'appuie sur la coopération d'atomes réactifs permettant d'obtenir la solution par émergence du comportement au niveau du robot reconfigurable.

A ce titre nous proposons de définir un programme unique pour chaque module robotique dans le système multi-agents. Celui-ci sera le comportement individuel dans chaque module du robot reconfigurable et prendra une décision locale en fonction des capacités de perceptions de l'environnement immédiat. La solution deviendra donc hautement parallèle puisque tous les atomes feront leurs calculs en même temps. Le fait de limiter la prise de décisions à une situation purement locale permet également de réduire la complexité de la solution. Toutefois, il faut pouvoir garantir que la tâche globale pour le robot reconfigurable sera bien atteinte par l'émergence d'un comportement global sans perte de ses modules.

Chapitre 2. Déplacement collectif : problématique et hypothèses

2.1 Problématique

L'étude du déplacement d'un robot dans un univers inconnu est un sujet classique de la robotique. Nous étudions ici une variante de ce problème en nous intéressant au déplacement d'un robot reconfigurable. Nous cherchons à définir le programme devant être chargé dans chaque module du robot pour qu'il se déplace de façon autonome vers un but en maintenant la cohésion du groupe de modules. Le modèle de robot que l'on étudie est inspiré du projet MAAM.

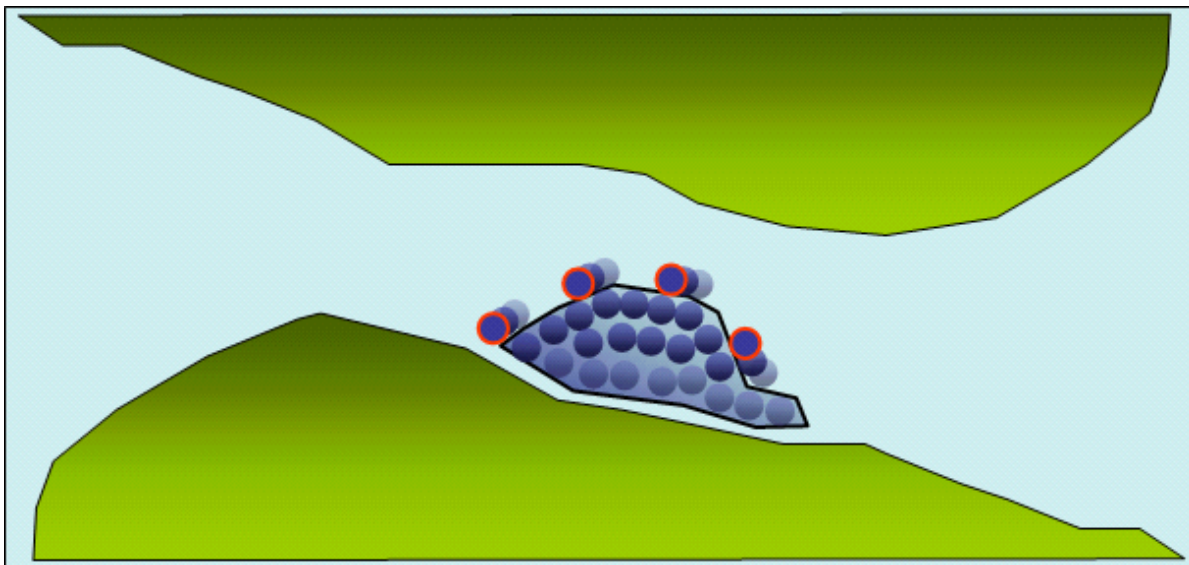


Figure 2.1 Déplacement reconfigurable

Les deux principales difficultés de ce problème sont de ne pas perdre la **connexion** entre les éléments du robot et trouver une méthode assurant d'obtenir une solution dans un **temps raisonnable**.

A. La connexion

La connexion est une contrainte liée à la nature des robots utilisés. Un robot reconfigurable est constitué d'un ensemble de modules connectés entre eux. Dans notre cas, chaque module est un robot muni d'actionneurs, de capteurs et d'une capacité de décision. Pour simplifier, nous appellerons **molécule** le robot reconfigurable et **atome** chacun des modules. Lors d'un déplacement de la molécule on cherche à maintenir la connexion des atomes entre eux de façon à ce que la molécule ne se « casse » jamais.

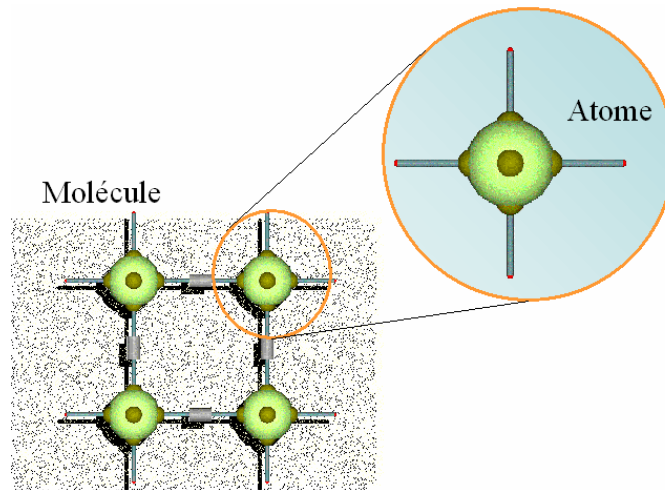


Figure 2.2 Molécule et Atome

B. Le temps raisonnable

Un rapide calcul de complexité montre le niveau de difficultés de ce problème. En effet, si nous avons N atomes et que chaque atome peut faire 8 mouvements différents alors à un instant donné on devra envisager 8^N possibilités de mouvements. Si maintenant on considère que chaque mouvement se rapproche du but d'une unité et que le but se trouve à une distance P alors le nombre total de mouvements possibles est de $(8^N) * P$. A titre d'exemple avec une dizaine d'atomes et une centaine de pas de reconfiguration, le calcul devient $(8^{10}) * 100$ soit un nombre de possibilités qui grandie exponentiellement en fonction du nombre d'atomes. C'est bien un système où le temps raisonnable de calcul pour toute généralisation reste hors d'atteinte de tout ordinateur actuel.

L'importance du nombre de possibilités montre qu'une planification centralisée des mouvements est un problème difficile. Néanmoins, des études menées dans le cadre du projet MAAM [LIV 03] ont montré les limites des approches proposées. Dans le projet M-Tran de robotique reconfigurable une autre approche globale est proposée, en particulier [YOS 04], mais comme l'explique ce travail le sujet reste d'une grande complexité.

C. L'approche proposée : l'émergence

Nous proposerons donc une approche distribuée à ce problème qui se fonde sur la coopération d'atomes réactifs permettant d'obtenir la solution par émergence du comportement au niveau de la molécule.

A ce titre nous proposons de définir un programme unique pour chaque atome. Ce programme sera le comportement individuel de chaque atome. Ce comportement prendra une décision locale en fonction des capacités de perception de l'environnement immédiat. La solution deviendra donc hautement parallèle puisque tous les atomes feront leurs calculs en même temps.

Le fait de limiter la prise de décision à une situation purement locale permet également de réduire la complexité du programme. Toutefois, il faut pouvoir garantir que la tâche globale du robot reconfigurable (la molécule) sera bien atteinte par l'émergence d'un comportement global.

D. Les compagnons de l'émergence : blocage & oscillation et une solution

Dans l'étude que nous allons présenter nous montrerons que deux problèmes principaux peuvent apparaître lors de l'implémentation d'une telle approche. Nous montrerons en particulier que des simulations qui « marchent » cachent en fait des défauts lourds. Nous pourrions ainsi prouver que ce type d'approche est sensible à l'**ordonnement** des prises de décisions dans les modules.

Nous montrerons que les deux problèmes auxquels il faut faire front sont le **blocage** où un module bloque un autre dans sa progression et l'**oscillation** où un module bloqué depuis longtemps cherche une solution alternative en tentant de « revenir sur ses pas » mais se retrouve dans un environnement totalement différent le faisant osciller sans qu'il puisse le détecter.

Nous proposerons une méthode générale qui permet quelque soit l'ordonnement choisi de garantir la convergence de la solution émergente.

2.2 Formalisation du problème

2.2.1 Modélisation du monde du robot : attracteur & gradient

La molécule, système multi-atomes, est modélisée par un système multi-agents.

L'**environnement** de l'agent est discrétisé dans un modèle de grille en 2D représenté sous forme de matrice environnemental. Dans ce modèle, chaque case est associée à ses voisins par 8 connectivités. Un agent prend la place d'une case dans la matrice d'environnement.

La **destination** à atteindre pour le robot est donnée par un élément attracteur qui permet de définir un champ de potentiel. L'attracteur est un élément qui émet un signal pouvant être reconnu par les agents. Dans une première approche, le signal se distribue dans l'environnement avec une intensité qui varie en fonction de la distance. Plus un emplacement dans l'environnement sera éloigné de l'émetteur et plus le signal sera faible. Ce signal est le potentiel de l'attracteur qui produit le gradient de l'attracteur.

Comme nous le verrons au chapitre 5, ce choix est raisonnable car dans le projet Maam les atomes disposent aux extrémités de leurs « pattes » de récepteurs infrarouges qui peuvent permettre de lire la direction et la distance d'une source lumineuse.

Des obstacles peuvent être présents dans l'environnement. Un obstacle est un objet immobile qui n'est pas franchissable par les agents. Ils rempliront des cases dans la matrice d'environnement.

Les différents types de potentiels

A. Potentiel régulier

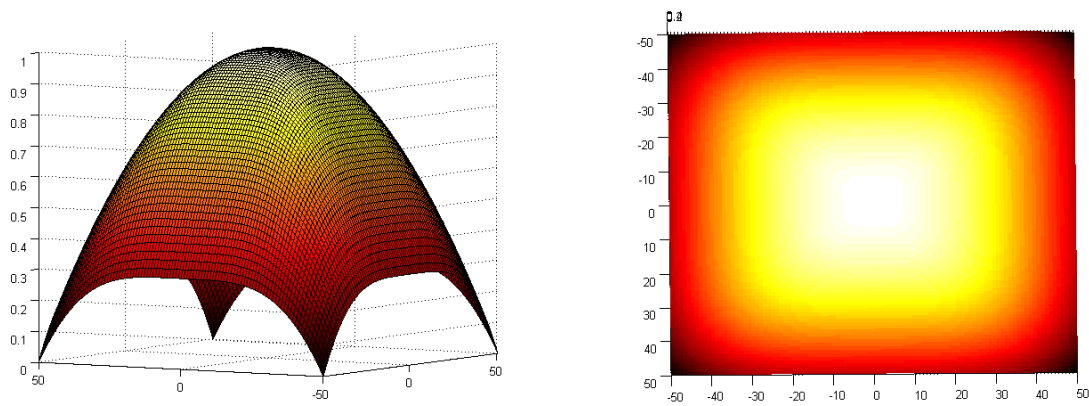


Figure 2.3 Champ de potentiel régulier

La propriété du potentiel régulier est que tous les points situés à la même distance de la cible mesurent la même valeur du potentiel. Ceci correspond par exemple à une lampe allumée.

B. Potentiel double

La propriété du potentiel double est qu'il décroît dans deux dimensions, avec une ligne de plus forte intensité qui correspond à un faisceau lumineux.

Il existe une droite D passant par la cible où l'intensité est maximale de part et d'autre de cette ligne. Le signal décroît régulièrement dans deux directions. La droite de plus forte intensité décroît en fonction de la distance et celle perpendiculaire décroît sous forme de gaussienne.

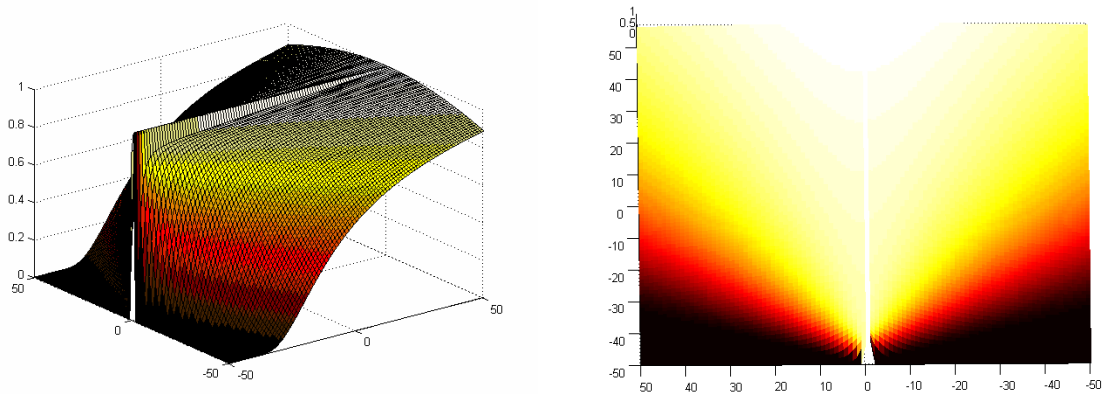


Figure 2.4 Champ de potentiel double

L'équation d'une gaussienne à un axe de symétrie peut s'écrire de la forme suivante :

$$f(x, \sigma, c) = e^{-\frac{(x-c)^2}{2\sigma^2}}$$

Equation 2.1 Gaussienne

Le x représente l'axe des abscisses, c l'axe des ordonnées et σ le paramètre modifiant la forme de la courbe.

Dans la figure 2.4, σ évolue en fonction des abscisses : plus x s'éloigne de l'attracteur et plus σ devient grand. Avec ce paramétrage et gardant c constante (0), on peut modéliser un potentiel prenant la forme d'une coque de bateau ou d'un cône de lumière et avec une ligne de plus haute intensité.

Avec un tel type de potentiel, la connaissance du potentiel d'un point M quelconque et la connaissance des valeurs du potentiel dans les cases voisines permet de définir de façon unique la position dans l'espace. On connaît donc la position de la cible et sa distance de M.

C. Champ de potentiel déformé

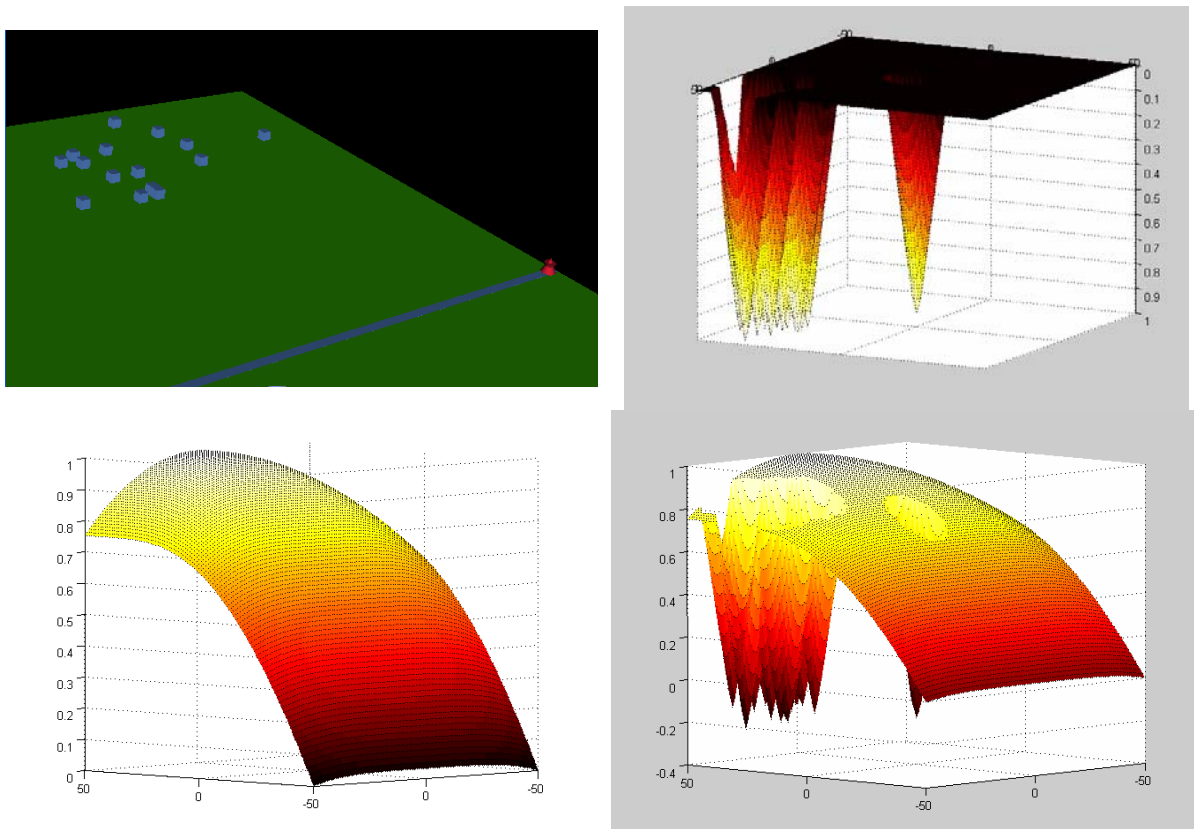


Figure 2.5 Champ de potentiel déformé

On peut penser que dans le cas où des obstacles sont présents dans l'environnement ils se comportent comme des cibles négatives qui perturbent le potentiel en l'augmentant à proximité des obstacles. Ceci correspond physiquement à des zones d'ombre où l'attracteur est progressivement masqué.

2.2.2 La perception des agents

Par hypothèse nous supposons que les organes de perception des atomes permettent à l'agent :

- de se faire une représentation sur la **présence ou l'absence d'agents** qui l'entourent dans la matrice de l'environnement (soit la détection de ses huit voisins potentiels).
- de **mesurer le gradient** issu du l'attracteur qui l'entoure

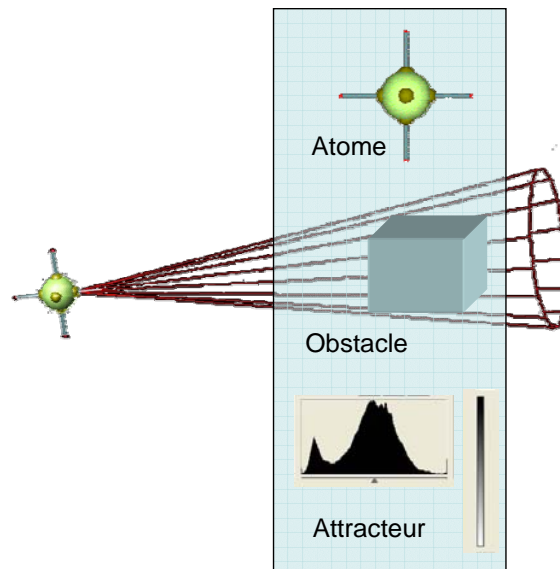


Figure 2.6 Perception des atomes

Les agents se construisent une représentation de l'environnement et de ce qui les entoure dans leurs 8 connectivités. Ils aperçoivent leurs environnements sous différents types d'entrées associés aux 8 directions possibles de déplacements dans le plan.

2.2.3 Déplacement des atomes

A. Déplacements élémentaires

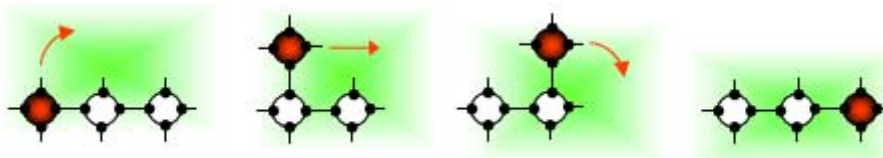


Figure 2.7 Exemple de reconfiguration successives d'un atome

Les agents sont contraints à se déplacer vers une direction où un emplacement vide est disponible (voir figure ci dessus). Le déplacement des agents est toujours fait par le déplacement d'une connexion ou d'un agent à un autre. Les connexions, entre les agents, se font à travers ses 4 principales connectivités: Nord, Sud, Est, Ouest ou les directions 2,4,6,8 dans la figure 2.8. Dans le déplacement d'une connexion à une autre, les agents n'ont pas la possibilité de déplacer d'autres agents (voir l'Annexe 1).

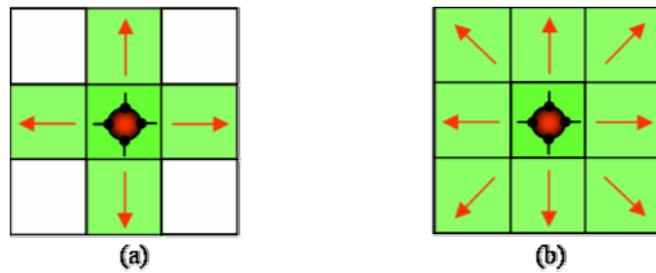


Figure 2.8 (a) Les 4 connectivités de Von Newman. (b) Le voisinage de Moore

On retrouve ici les idées développées dans la représentation de l'environnement pour les systèmes multi-agents plus connues sous le nom de connectivités de Von Neuman et de voisinage de Moore [AMB 06].

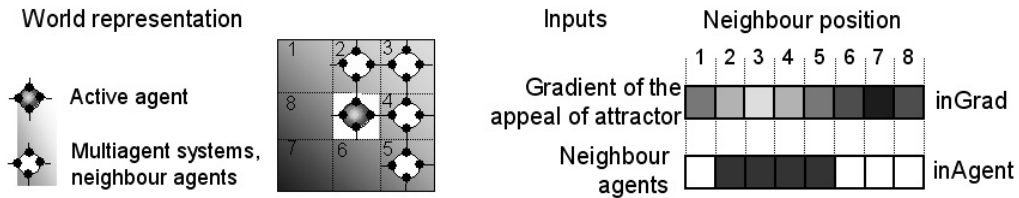


Figure 2.9 Directions et les vecteurs du voisinage correspondants

Dans la figure 2.9 nous proposons une numérotation pour différencier les positions du voisinage de l'atome.

B. Contrainte sur le déplacement

Les actions autorisées, pour le déplacement des agents, sont celles qui ne viennent pas à l'encontre de la cohésion globale du groupe.

Les contraintes de déplacements sont :

- garder au moins une connexion avec un autre agent.
- dans le cas où un agent garde une seule connexion avec la structure, il devient un agent clef pour la cohésion du groupe. Il ne lui est pas permis de se séparer. Pour bouger un agent ne doit pas s'avouer comme une pièce clef pour la structure.
- une place pour un déplacement possible doit être vide
- le chemin pour de déplacer vers la position de destination doit être libre

Pour réaliser un déplacement, tout en assurant la cohésion du groupe, l'agent doit vérifier que les agents qui sont autour sont dûment chaînés, et ainsi éviter la perte d'un maillon du groupe.

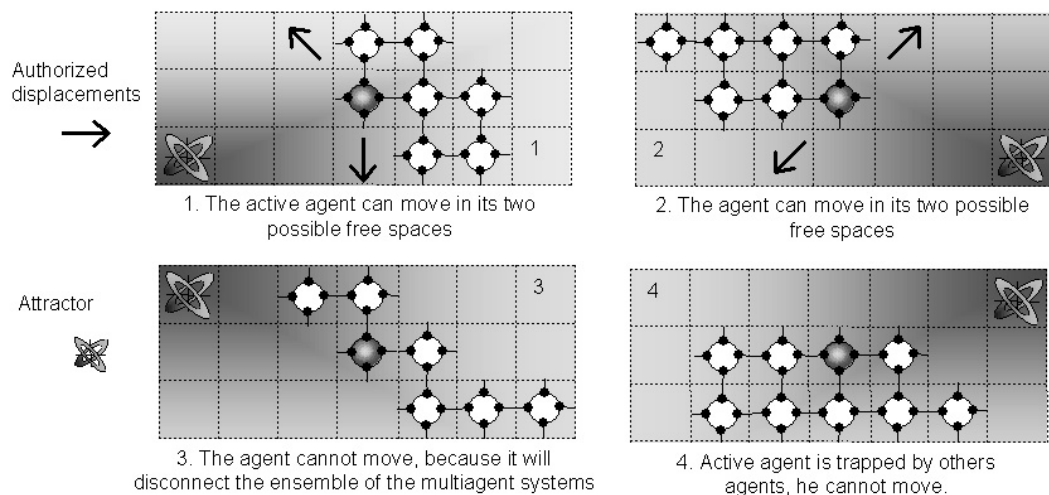


Figure 2.10 Exemples de déplacement autorisé ou non autorisé

Ainsi un agent qui n'a pas plus de 2 connexions avec d'autres agents peut se trouver dans le cas 1 ou 2 de la figure 2.10, auquel cas, grâce à son voisinage il sait qu'en se déplaçant il ne cassera pas la chaîne. Le mouvement est donc autorisé. Par contre dans le cas 3 le déplacement de l'atome couperait la molécule en deux, le mouvement est donc interdit. Dans le cas 4 le mouvement est en général interdit mais cette contrainte peut être levée dans certaines simulations. Ce type de mouvement est à l'origine de trous dans la structure ce qui génère beaucoup de problèmes.

2.2.4 L'ordonnancement des calculs dans les agents

L'ordonnancement des calculs correspond à l'ordre dans lequel seront exécutés les programmes dans les atomes de la molécule. On peut considérer que le modèle idéal est celui du parallélisme absolu : tous les calculs se font en même temps dans chaque module. Toutefois, un robot reconfigurable est de nature séquentiel. En effet, les déplacements ne peuvent se faire en parallèle que lorsque les deux atomes sont voisins à cause des risques de déconnexion que cela engendrerait. On supposera que l'ordre d'exécution des mouvements de deux atomes voisins est réglé au niveau mécanique. Le premier ne bouge seulement s'il est garanti que les autres ne bougent pas.

Cette contrainte montre qu'au niveau du programme, le calcul n'est pas le même si on fait le mouvement d'un atome après ou avant un autre. Pour tenir compte de ce problème nous supposons dans les simulations qu'il y a deux types d'ordonnements logiciels possibles.

A. Ordonnancement séquentiel

Dans ce cas les atomes reçoivent un quota de temps (supposé suffisant pour dérouler une itération complète du comportement) les uns après les autres suivant une liste fixe (mode round robin).

C'est souvent la solution retenue dans les simulations car cela correspond à une logique liée à une vue « macro » où l'on considère que tous les atomes prennent une décision par leur algorithme de comportement par unité de temps. Toutefois, cette régularité pose des problèmes dont nous discuterons par la suite et qui nous amènera à proposer un deuxième modèle d'ordonnement

B. Ordonnement aléatoire

Dans ce cas les atomes reçoivent leurs quotas de temps de façon aléatoire. Ceci permet de tenir compte d'éventuelles différences dans les temps de traitements des calculs, des capteurs, des actionneurs dans les atomes. Cette ordonnancement étant le plus « chaotique », il force donc à développer des méthodes sûres de construction des algorithmes émergents.

2.3. Algorithme réactif

Le principe de l'algorithme réactif est de diminuer la distance qui sépare les agents à la source de l'attracteur tout en gardant la connexion. Il s'agit pour les agents de prendre la place là où la distance à l'attracteur se réduit, c'est-à-dire que le gradient de l'attracteur diminue par rapport à l'emplacement où se trouve l'agent.

L'algorithme décide de l'action à entreprendre selon deux facteurs : le gradient de l'attracteur et la configuration des atomes qui sont dans son voisinage. Le déplacement est effectué vers une direction où la distance avec l'attracteur est réduite en respectant les contraintes pour la cohésion du système.

En considérant que les entrées sont modélisées sous forme de vecteurs, le pseudo code de l'algorithme réactif, pour atteindre l'attracteur, peut s'écrire de la manière suivante :

A. Principe de l'algorithme réactif direct

On va distinguer plusieurs types d'algorithmes qui ont été testés. Nous présentons ici leurs pseudo-codes.

Algorithme "forward total"

```
tant que <je suis actif>
  si il existe des positions maximisant le potentiel
  alors vérifier pour ces positions l'accessibilité;
          choisir la meilleure action;
          se déplacer
  sinon ne pas bouger <actif off>
fin tant que
```

Dans ce cas le module va faire une suite de déplacement le long de la molécule jusqu'à atteindre un position qui sera un minimum pour le potentiel.

Algorithme "forward un coup"

```
si il existe des positions maximisant le potentiel
alors vérifier pour ces positions l'accessibilité;
          choisir la meilleure action;
          se déplacer
sinon ne pas bouger
```

Dans ce cas le module va faire un seul déplacement le long de la molécule puis va s'arrêter et attendre

son prochain quota de temps. Pendant ce temps tous les autres modules peuvent avoir bougés.

Les comportements émergents sont évidemment différents avec l'algorithme "forward total" ou avec l'algorithme "forward un coup".

B. Principe de l'algorithme réactif avec retour arrière

Pour réagir à l'existence possible de minima locaux du potentiel entre plusieurs obstacles de l'environnement ou une configuration de blocages de la molécule. On peut être amené à programmer avec un algorithme de type backward. Dans ce cas, les atomes possèdent un chronomètre qui leur permettent d'estimer combien de temps il reste dans une position où ils peuvent bouger, sans pouvoir diminuer la distance avec l'attracteur.

Algorithme "backward"

```
si il existe des positions maximisant le potentiel
alors vérifier pour ces positions l'accessibilité;
        choisir la meilleure des actions;
        se déplacer
sinon si le chronomètre à dépasser le seuil
        alors « algorithme de réabsorption » ;
            armer le chronomètre
        sinon ne pas bouger
```

Algorithme reabsorbtion

```
pile <- vide
tant que je peux bouger
    place <- meilleure place où un déplacement est possible et potentiel plus
        élevé et qui n'est pas dans pile;
    si une place existe
    alors ajouter (place, pile);
        bouger en (place);
    sinon vider (pile);
        ne pas bouger;
fin tant que
```

On voit sur cet algorithme que l'atome va revenir en arrière sur ses mouvements jusqu'à ce qu'il rencontre:

- soit une possibilité de repartir dans une autre direction.
- soit qu'il soit bloqué sans pouvoir bouger.

Le retour arrière est fait pour sortir des minima locaux où peuvent être attrapé un groupe d'atomes dans la molécule.

Il faut noter que cet algorithme ne revient en arrière que pour trouver un nouveau chemin possible.

Rien ne permet ici de savoir si on n'a pas déjà testé ce chemin. Le fait que tout bouge dans la molécule rend impossible de construire une carte globale qui permettrait de résoudre ce type de prise de décision. A partir de cet algorithme on risque l'émergence de phénomènes d'oscillations. L'algorithme backward est donc à utiliser avec précaution même si on pourra vérifier de bons résultats en simulation.

Sensibilité des algorithmes au type de gradient

Descente de gradient simple

La descente de gradient simple correspond à la recherche de la direction du déplacement pour laquelle le gradient vers la cible va diminuer.

Descente d'un gradient double

La descente d'un gradient double doit être expliquée avec plus de précision. En effet il existe deux référentiels:

- Pour le robot : il existe les directions des mouvements possibles pour le module suivant axe X et Y.
- Dans l'environnement : il existe deux grandeurs, D celle qui mesure la distance de l'émetteur et H celle qui mesure la distance de la ligne de plus forte intensité.

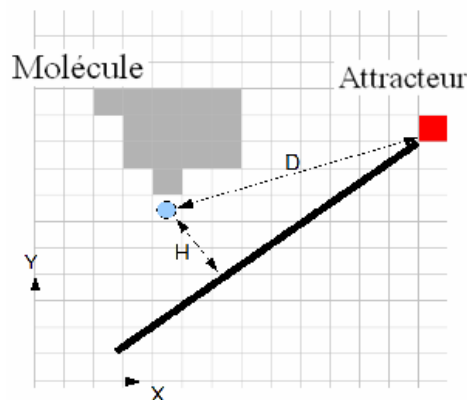


Figure 2.11 Distance à l'attracteur (D) et Distance à la ligne (H)

On utilise la descente du gradient basée sur un ordre total donné par un ordre lexicographique sur le couple (D,H). Dans ce cas un mouvement en X ou en Y sera considéré comme acceptable si la nouvelle position vérifie:

$$(D,H) > (D',H') \text{ si } (H > H') \text{ ou } (D > D' \text{ et } H \approx H')$$

La relation \approx entre H et H' est due à l'introduction de la tolérance de la largeur de la ligne de plus forte intensité.

On ne peut pas permuter l'ordre d'évaluation de (D,H) en (H,D). La formation de la structure des modules vers un attracteur radiale annihile l'effet introduit par la ligne de plus forte intensité.

2.4. Synthèse

La résolution du problème du déplacement collectif va donc se placer dans une combinaison des trois facteurs ci-dessous:

Choix du potentiel

- Régulier
- Potentiel double
- Potentiel déformé

Ordonnancement

- Séquentiel
- Aléatoire

Algorithme réactif

- type Forward total
- type Forward un coup
- type Backward

Chapitre 3. Déplacement collectif : simulations, erreurs, solution

3.1. Introduction

Dans ce chapitre nous allons suivre l'ordre dans lequel nous avons effectué nos travaux. Nous voulons montrer comment sont apparues les interrogations sur les simulations que nous faisons.

En effet, notre démarche donnant de bons résultats, nous avons poussé notre réflexion dans une voie que nous allons finalement abandonner, celle de l'apprentissage, pour reprendre sur des questions plus fondamentales de deadlock.

3.2. Algorithme réactif

Comme nous l'avons présenté au chapitre précédent les simulations vont s'appuyer sur les choix de trois types de propriétés :

Choix du potentiel

- Régulier
- Potentiel double
- Potentiel déformé

Ordonnancement

- Séquentiel
- Aléatoire

Algorithme réactif

- type Forward total
- type Forward un coup
- type Backward

Simulation 1 : Potentiel régulier, ordonnancement séquentiel, algorithme type « forward total »

Dans cette première simulation nous étudions un attracteur de type lampe qui crée un gradient régulier centré au point de l'attracteur. Les atomes avancent de façon régulière chacun à leur tour suivant un algorithme de round robin. Dans ce cas chaque atome descend le gradient jusqu'à atteindre une position où le gradient est minimal.

L'émergence est un comportement où les atomes se mettent en ligne, puis les atomes les plus en arrière avancent jusqu'à la position en tête de la file.

Sur la simulation ci-dessous la présence d'un obstacle, loin du chemin suivi par la molécule, n'engendre pas de déformations du champ de potentiel perceptible par les atomes.

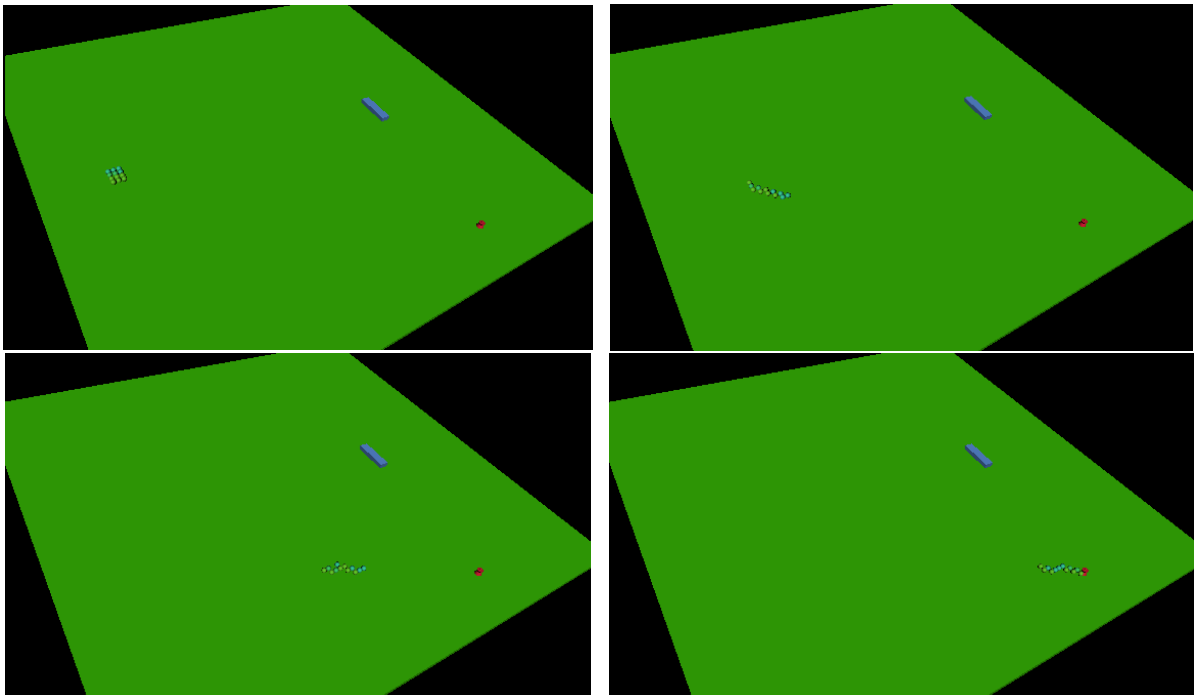


Figure 3.1 Images de la simulation 1

Sur cette simulation les résultats sont toujours satisfaisants, une ligne se forme et les atomes roulent les uns sur les autres en direction de la cible.

La descente du gradient s'effectue de façon continue car l'ordonnancement séquentiel permet à chaque atome d'avancer de tous les mouvements possibles à chaque itération.

C'est la simulation de référence qui marche toujours et qui sera le guide de la méthodologie proposée au chapitre suivant.

Simulation 2 : Potentiel déformé, ordonnancement séquentiel, algorithme type « forward total »

Par rapport à la première simulation l'obstacle déforme le champ de potentiel ce qui engendre un contournement de l'obstacle. La seule différence est donc l'utilisation d'un champ de potentiel déformé.

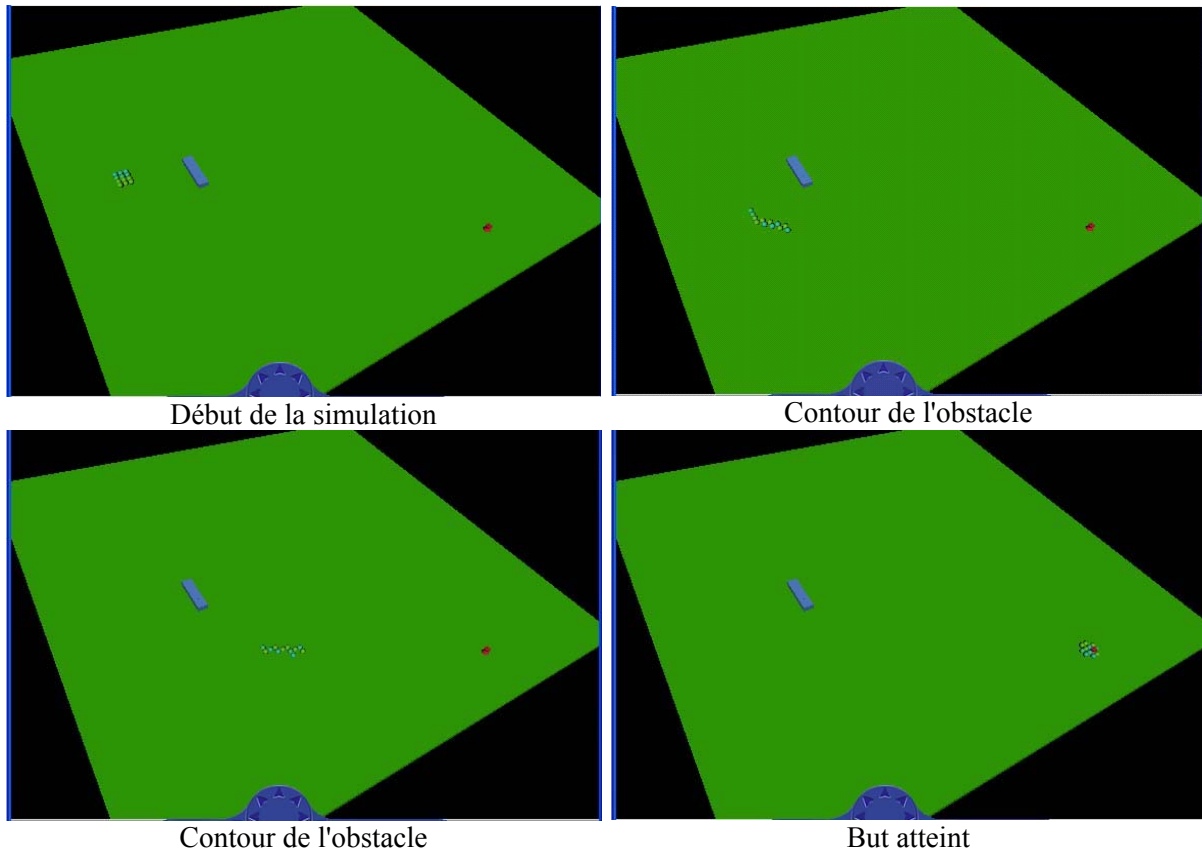


Figure 3.2 Images de la simulation 2

Avec cette hypothèse, le système converge de la même manière que précédemment. On voit que les atomes tournent autour des obstacles en suivant les lignes des gradients.

Ce résultat attendu est compatible avec le précédent.

Simulation 3 : Potentiel non déformé avec obstacle, ordonnancement séquentiel, algorithme type « forward total »

Sur cet exemple la molécule rencontre un obstacle sur le chemin vers l'attracteur. Cet obstacle lui coupe toutes les possibilités de déplacement en direction de l'attracteur. La molécule s'arrête sur le rebord de l'obstacle.

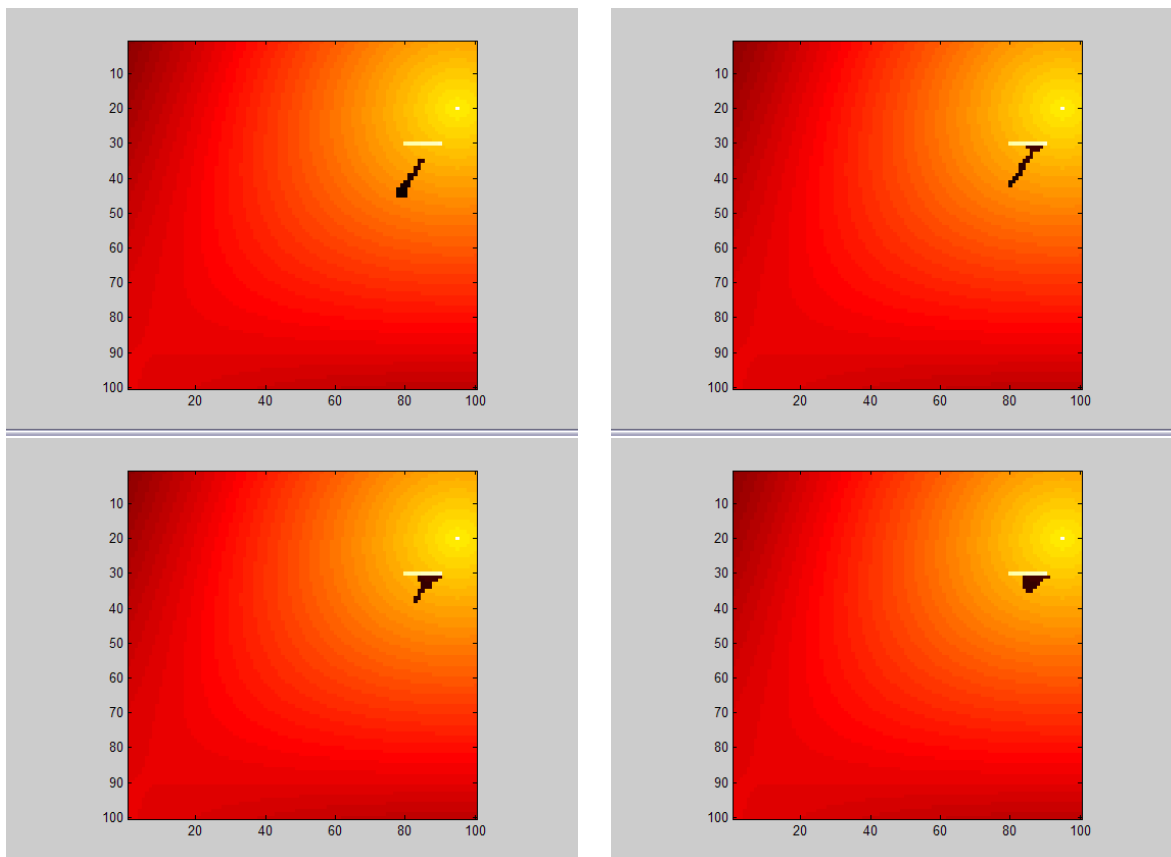


Figure 3.3 Images de la simulation 3

Ici, comme sur le simulation précédente le comportement obtenu est conforme aux prévisions que nous avions.

Simulation 4 : Potentiel non déformé, ordonnancement séquentiel, algorithme type backward

En utilisant l'algorithme backward, la molécule va chercher des solutions pour contourner l'obstacle. Et après un départ vers la gauche elle va finalement trouver une solution vers la droite.

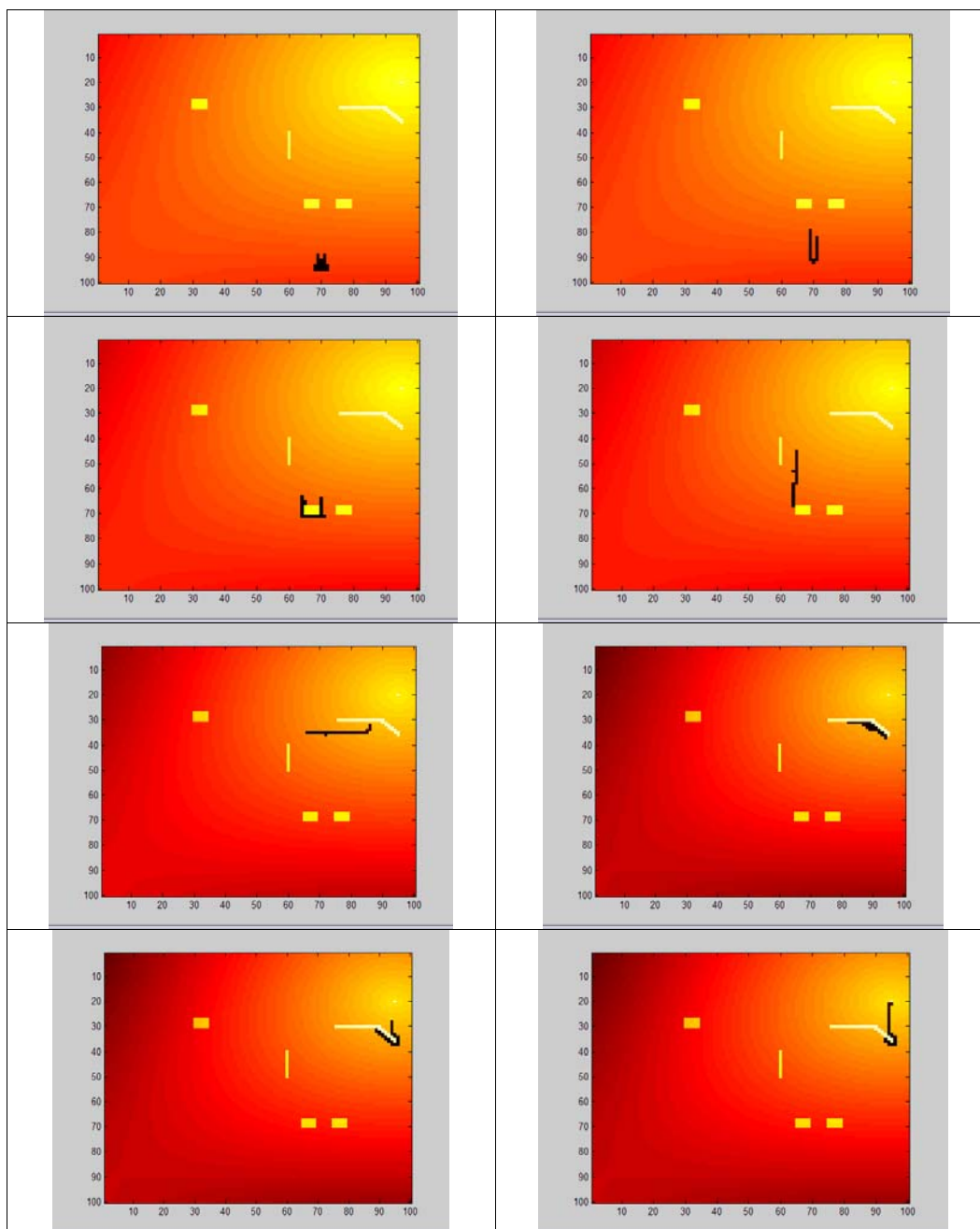


Figure 3.4 Images de la simulation 4

Cette simulation donne les résultats attendus.

Les algorithmes réactifs donnent de bons résultats. Nous avons alors voulu nous intéresser aux

capacités d'apprentissage des modules. L'objectif était alors de voir si ces algorithmes de décisions pouvaient être trouvés par les atomes (on pouvait alors accepter que l'apprentissage des atomes ait des programmes différents)

3.3 Méthodes connexionnistes

Fort de bons résultats obtenus avec les algorithmes précédents, nous avons étudié la possibilité qu'un atome découvre le comportement élémentaire qui permet au groupe de rejoindre la cible.

Plusieurs possibilités pour effectuer ce travail étaient possibles. Nous avons retenu deux approches:

- l'une classique, le réseau apprend son comportement par une mesure de récompense basée sur la distance à l'attracteur. Nous avons utilisé un réseau de neurones hybride avec des algorithmes génétiques dans un apprentissage supervisé.
- l'autre est de partir d'une observation partielle du comportement collectif pour essayer de déterminer le comportement individuel de chaque module. La problématique est ici de se demander si à partir de l'observation d'un groupe, pendant quelques cycles, il est possible de faire une généralisation.

3.3.1 Apprentissage supervisé

Dans ce cas nous avons utilisé un réseau avec deux couches cachées.

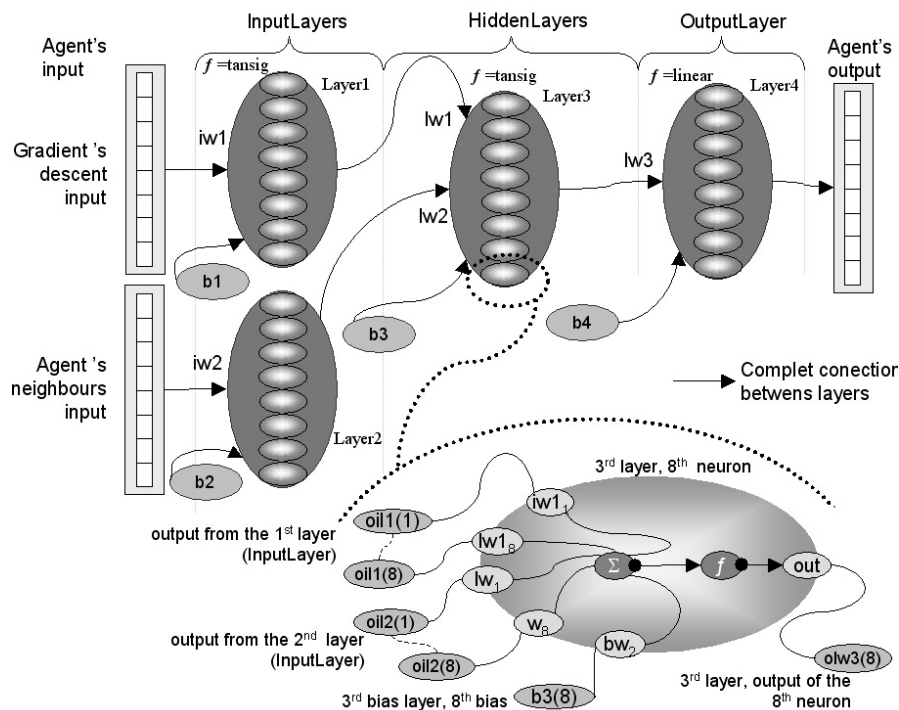


Figure 3.5 Le réseau

Le poids des connexions sera modifié en utilisant des algorithmes génétiques. La fonction d'évaluation de la performance du réseau est calculée en fonction de la distance à la cible et de la connexion ou de la déconnexion du module dans la molécule.

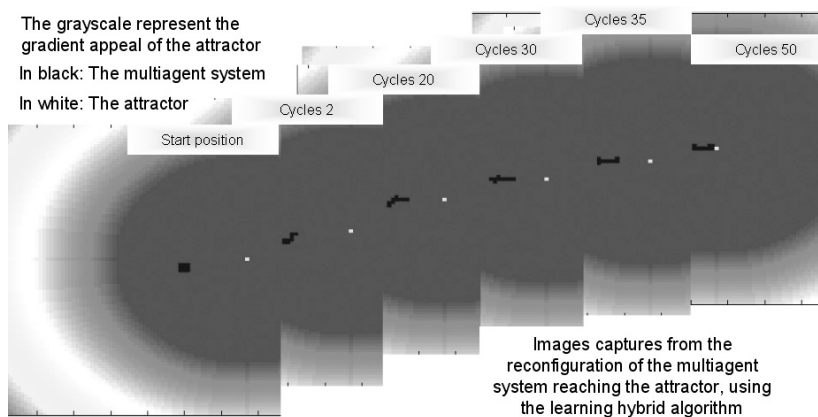


Figure 3.6 Simulation de l'apprentissage supervisé

Avec cette approche on obtient de nouveau de très bons résultats puisque la molécule suit le gradient jusqu'à la cible sans perdre aucun module. Par rapport à l'algorithme réactif, le nombre de mouvements pour atteindre la cible est cinq fois plus important.

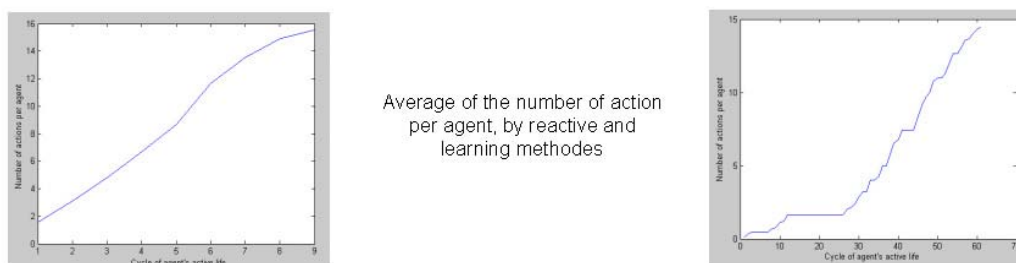


Figure 3.7 Temps en nombre de pas pour atteindre la cible, algorithme réactif et apprentissage supervisé

Il faut noter ici que l'algorithme utilisé est pour l'ordonnancement est le même que celui utilisé lors des simulations avec l'algorithme réactif : un ordonnancement séquentiel.

3.3.2 Apprentissage à partir d'observations

Comme l'apprentissage précédent a donné de bons résultats, nous avons voulu étudier la possibilité de faire apprendre un réseau sans faire référence à la cible (par un calcul de distance) mais en partant de l'observation d'une partie de l'exécution d'un algorithme réactif. L'idée est de se demander s'il est possible d'observer un phénomène réel (par exemple des comportements dans la nature) et de les faire apprendre.

Nous utiliserons l'algorithme réactif défini en 2.2.1 pour construire la base d'apprentissage. L'objectif est de regarder les déplacements des robots réactifs durant une petite période de temps et de choisir une série d'exemples parmi eux. La question scientifiquement intéressante est : « A partir de combien d'exemples de déplacements a-t-on un apprentissage du comportement du robot ? »

On va mémoriser les déplacements (couple (V1, V2) en entrée et S en sortie) pour 12 robots pendant leurs déplacements sur 5 cycles. Un cycle est la période pendant laquelle tous les robots ont tourné leur algorithme réactif une fois. Il faut rappeler ici que l'algorithme réactif génère pendant son cycle tous les déplacements successifs possibles pour le robot. Nous exécuterons cette mémorisation sur 20 positions différentes de l'attracteur. Pour la constitution de cette base de déplacements nous avons recueilli 4179 couples d'entrées-sorties.

A partir de cette base de déplacements nous extrayons les exemples les plus significatifs en utilisant une fonction de distance sur l'espace des données (V1,V2). Cette fonction commence par trier les sorties S afin de choisir un nombre équitable de couples de données (V1,V2) par déplacement possible. Puis, pour un même déplacement on choisi parmi les données triés (V1,V2) le nombre voulu de couples en les équirépartissant. Ainsi nous choisissons plusieurs séries d'exemples pour l'apprentissage 3752, 2056 et 1031.

Nombre total de combinaison V1,V2, S	10240
Base de déplacements observées	4179
Nombre d'exemples retenus	3752 ou 2056 ou 1031

Tableau 3.1 Nombre de combinaisons et exemples retenus

3.3.3 Structure générale du réseau de neurones

A. Les données du réseau

Comme pour l'algorithme réactif nous utiliserons comme données du réseau de neurones deux vecteurs de dimension 8 :

Le premier vecteur V1 est un vecteur de booléen qui code la présence ou non de robot dans le voisinage immédiat de l'agent (cf §2.2.3). V1 possède donc $2^8 = 256$ combinaisons possibles.

Le deuxième vecteur V2 code le gradient qui entoure le robot par rapport à son attracteur. La mesure est normalisée pour ne pas tenir compte de la distance à l'attracteur. Comme il y a 8 cases différentes la normalisation comprendra au plus 8 valeurs différentes.

Pour tenir compte de la réalité expérimentale des mesures de capteurs (lecture par un capteur infra rouge dans le cas du projet Maam), nous considérons durant la simulation que ce codage dans V2 est légèrement bruité. Ceci signifie que la direction dans laquelle se trouve l'attracteur ne détermine pas obligatoirement un maximum unique dans cette direction. En tenant compte de cette hypothèse, on calcule le nombre de combinaisons pour le gradient comme étant égal à 8 configurations possibles pour une direction donnée de l'attracteur par 5 directions possibles pour tenir compte de la symétrie du problème soit 40 combinaisons.

Le nombre total d'entrées est donc de $40 * 256 = 10240$.

B. Les sorties du réseau

Au final, le réseau doit déterminer le mouvement à effectuer en fonction de la valeur des entrées. Le codage attendu est donc un vecteur de 8 valeurs booléennes parmi lesquelles une seule d'entre elles (au plus) sera égal à 1, celle qui sera quelle direction de déplacement à faire.

Pour obtenir ce résultat les sorties internes du réseau qui sont des valeurs réelles dans l'intervalle [0,1] sont seuillée.

Une fonction SeuilMax est utilisée celle-ci est telle que pour chaque sortie S_i on calcule :

$$S_i (S_i > \text{seuil}) \text{ et } (S_i = \text{Max}(S_1, \dots, S_8)) \text{ Alors } 1 \text{ sinon } 0$$

On constate que si aucune sortie n'a atteint le seuil le robot ne bougera pas.

C. Structure du réseau

Tout d'abord nous allons travailler avec des réseaux à une ou deux couches cachées, puis avec des couches séparées, enfin, tirer des conclusions respectives à l'adéquation pour un meilleur réseau de neurones.

Le nombre de neurones dans les couches cachées joue un rôle très important dans les performances des réseaux. Ces dernières peuvent varier selon les différentes tailles des couches cachées. Pour trouver les tailles adéquates il faut tester les réseaux de neurones avec différents nombres des couches cachées.

D. Fonctions de transfert

Les fonction de transfert les plus utiliser sont:

La sigmoïde qui retourne des valeurs entre 0 et 1	$f(n) = \frac{1}{1 + e^{-n}}$		
La tangente sigmoïde, qui prend des valeurs entre -1 et +1	$f(n) = \frac{2}{(1 + e^{-2n})} - 1$		
La fonction d'échelon, où sortie logique	$f(n) = \begin{cases} 0 & \text{si } v < \text{seuil} \\ 1 & \text{si } v \geq \text{seuil} \end{cases}$		
La fonction de transfert linéaire	$f(n) = (n)$		

Tableau 3.2 Fonctions de transferts les plus fréquentes

3.3.4 Méthode d'apprentissage

La méthode utilisée pour l'apprentissage est le rétro propagation de Levenberg-Marquardt [LEV 44, MAR 63].

L'algorithme de Levenberg-Marquardt ou algorithme LM permet d'obtenir une solution numérique au

problème de minimisation d'une fonction, souvent non-linéaire et dépendant de plusieurs variables.

Il consiste à interpoler l'algorithme de Gauss-Newton et la méthode de descente de gradient. L'algorithme est plus stable que le premier, c'est à dire qu'il trouve une solution même s'il est démarré très loin d'un minimum. Cependant, pour certaines fonctions très régulières il peut converger légèrement moins vite. L'algorithme fut découvert par Donald Marquardt puis publié par Kenneth Levenberg.

Son application principale est la régression linéaire au travers de la méthode des moindres carrés. Etant donné un certain nombre de paires de données (t_i, y_i) , on cherche le paramètre \mathbf{p} de la fonction $f(t|\mathbf{p})$ de sorte que la somme des carrés des déviations soit minimale (eq 3.1).

$$S(\mathbf{p}) = \sum_{i=1}^m [y_i - f(t_i | \mathbf{p})]^2$$

Equation 3.1 La somme des carrés des déviations

La procédure de l'algorithme est itérative. On part d'un paramètre initial que l'on suppose « assez proche » d'un minimum, et qui constituera le vecteur \mathbf{p} de départ.

A. Initialisation des poids

Avec le souci de trouver des réseaux de neurones les plus performants possibles, nous avons procédé à plusieurs initialisations de ses poids. La première initialisation des poids se réalise avant l'entraînement et sous forme aléatoire, dans un rang entre 0 et 1. Après l'aboutissement de l'apprentissage des réseaux avec des initialisations aléatoires, nous avons remarqué dans des suites d'expérimentations que plusieurs de ses réseaux adoptent un comportement très différent selon la direction de l'apprentissage. Ceux qui possèdent de bonnes performances ont généralement une configuration des poids relativement similaire. C'est pourquoi nous avons continué avec des expérimentations sur les mêmes réseaux avec différentes initialisations des poids des couches cachées. Nous avons utilisé plusieurs sortes de combinaisons pour différencier qu'elles étaient les plus performantes.

3.3.5 Simulation sur plusieurs types de réseaux

Dans cette section nous rendons compte des différents types de réseaux testés ainsi que des performances obtenues par rapport aux critères de performance définis ci-dessous. Nous avons également essayé beaucoup d'autres types de combinaisons.

Dans la recherche de l'implémentation d'un réseau de neurones pour obtenir une commande de l'atome pour cela on a utilisé plusieurs types de réseaux. Nous présentons un tableau comparatif des systèmes de commandes neurales introduit dans chaque atome (tableau 3.3).

A la différence du choix de l'apprentissage supervisé (c.f. 3.3.1), on a voulu que l'architecture de base du réseau de neurones soit simple et performante. Nous avons donc utilisé un réseau à une couche cachée avec 2 types d'entrées (atome et gradient), chacun de ces types ayant 8 (le voisinage de l'atome) entrée.

Ce choix correspond aussi à une étude préliminaire sur le comportement de plusieurs architectures de réseaux où on a essayé plusieurs types de réseaux.

Les graphes d'évaluation correspondent à l'appréciation des caractéristiques de **Performance**, **Rapidité** et **Généralisation** des réseaux entre les architectures proposer. La notation RNC (A,B,C) décrit la structure du réseau de neurone avec A entrées, B couche cachée et C sortie.

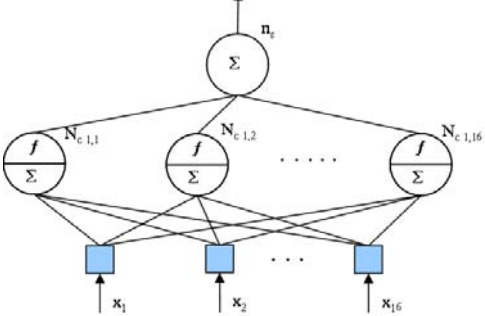
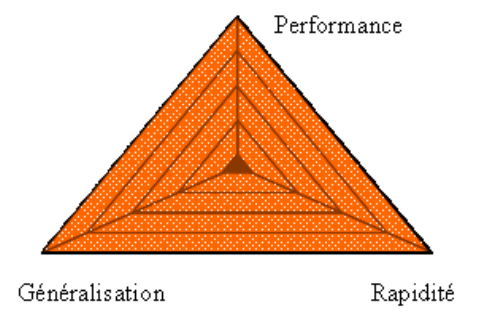
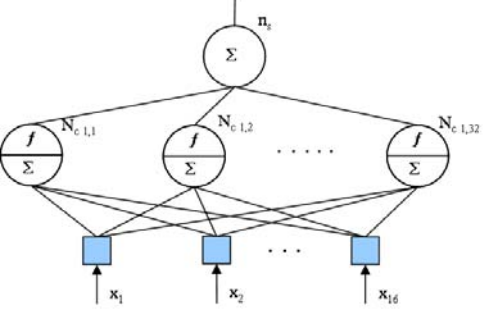
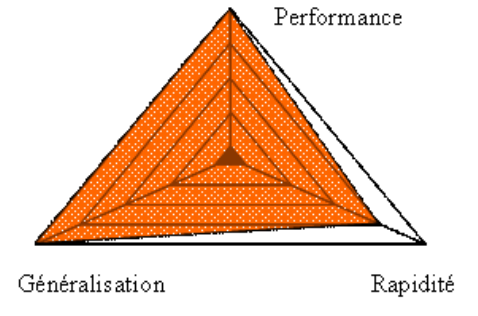
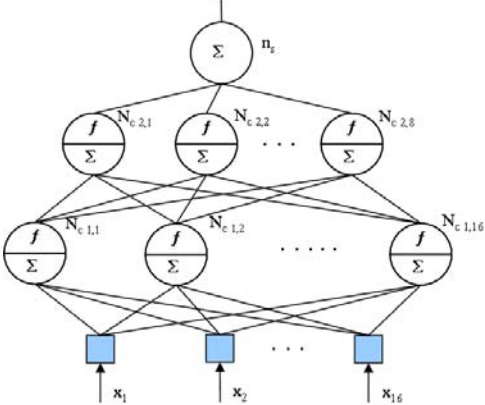
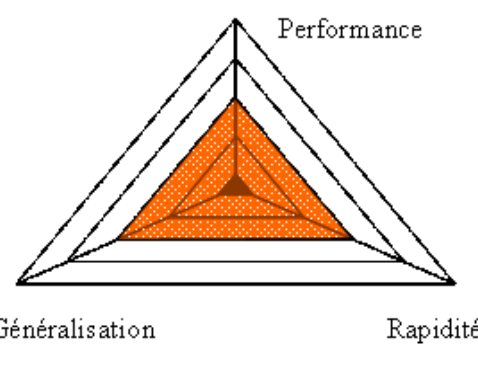
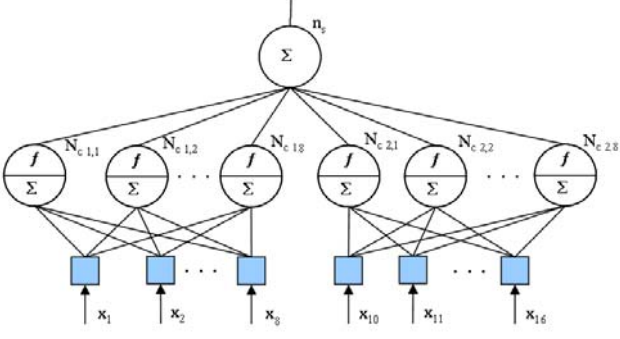
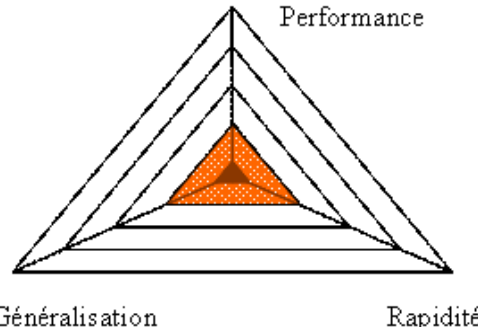
 <p style="text-align: center;">RNC(16,16,1)</p>	
 <p style="text-align: center;">RNC(16,32,1)</p>	
 <p style="text-align: center;">RNC(16, 16, 8, 1)</p>	
 <p style="text-align: center;">RNC((RNC(8,8) + RNC(8,8)),0,1)</p>	

Tableau 3.3 Architecture des réseaux de neurones et leurs évaluations

L'évaluation suivante se fait en fonction des architectures utilisées. La **Performance** est le résultat de la performance au moment de l'apprentissage. La **Rapidité** est relatif au temps de calcul que l'ordinateur a utilisé pour arriver à la convergence de l'algorithme de minimisation lors de l'apprentissage. La **Généralisation** est la capacité du réseau appris à être implémenté dans le choix de l'action pour les huit voisinages sous une base de données différente à celle de l'apprentissage.

On a choisi de travailler avec 3 fonctions de transfert dans 2 configurations d'architectures de réseaux. Utiliser une fonction de transfert non-linéaire combiné avec une autre linéaire permet l'apprentissage des relations entrées-sorties linéaires et non linéaires.

Réseaux de neurones	Fonction de transfert	
	Couche cachée	Couche de sortie
Variante 1	tangente sigmoïde	linéaire
Variante 2	sigmoïde	linéaire

Tableau 3.4 Variantes des réseaux de neurones

On a voulu savoir sous quelle initialisation on pouvait obtenir les meilleurs résultats. C'est pour cela que nous avons utilisé une batterie de systèmes neurales sur chaque type d'architecture.

L'architecture du réseau retenue correspond donc à un réseau ayant 16 entrées, avec 1 couche cachée partiellement connectée avec des fonctions de transferts tangente sigmoïde ou sigmoïde et une sortie linéaire (fig. 3.8).

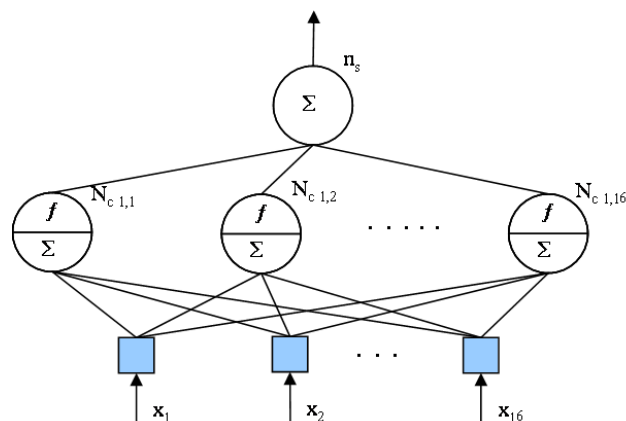


Figure 3.8 Architecture du réseau de neurone retenu

Les modèles d'architecture de réseaux de neurones retenus pour la suite sont 6. Les différents types de réseaux varient selon le nombre de neurones de la couche cachée avec 2, 3, 4, 8, 12, 16. Ceci est pour savoir jusqu'à quelle diminution du nombre de neurones, dans la couche cachée, le réseau arrive à faire une bonne classification.

Nous avons utilisé 16 à 2 neurones dans la couche cachée pour déterminer le nombre de neurones nécessaires pour que le réseau apprenne correctement la configuration de ses poids.

Type de réseau	Types d'initialisation	Types d'erreurs	Sortie des réseaux de neurones par direction							
			1	2	3	4	5	6	7	8
Réseaux de neurones type5	Aléatoire	Déconnection	0	0	0	0	0	0	0	0
		Erreur d'action	2	0	19	0	235	0	42	5
		Erreur du réseau	1,4305e-007	9,5864e-011	0,0010749	1,8667e-009	0,020407	2,6482e-011	0,0068776	0,0013438
		Déconnection	0	0	0	0	0	6	0	0
		Erreur d'action	202	0	4	0	9	11	18	0
		Erreur du réseau	0,013714	4,0485e-007	6,0868e-009	1,8866e-011	0,00047225	0,00171	0,0017375	5,8207e-011
	Gaussienne	Déconnection	0	0	0	0	0	0	0	0
		Erreur d'action	2	23	2	0	1	0	1	0
		Erreur du réseau	1,0393e-005	0,0010381	3,071e-008	8,0004e-011	1,9271e-005	9,9079e-011	4,4988e-007	9,9731e-011
		Déconnection	0	0	0	0	0	0	0	7
		Erreur d'action	8	0	97	0	5	0	1	9
		Erreur du réseau	0,0015926	8,2889e-011	0,0091763	2,757e-011	3,2136e-011	8,0744e-006	0,00059182	0,0022018
	Binaire1 (triple)	Déconnection	0	0	0	0	0	0	0	3
		Erreur d'action	2	1	23	0	8	246	3	8
		Erreur du réseau	4,4752e-005	0,00047067	0,0017693	2,5694e-011	0,00037378	0,018574	0,00098303	0,001221
		Déconnection	0	0	0	0	0	1	0	0
		Erreur d'action	5	10	1	0	15	6	1	0
		Erreur du réseau	0,0011135	0,00094199	2,3924e-010	2,3645e-011	0,0024249	0,0013472	3,9295e-008	3,6936e-006
	Binaire2 (direction)	Déconnection	0	0	0	0	0	0	0	0
		Erreur d'action	3	0	8	0	7	0	1	280
		Erreur du réseau	0,00021469	5,403e-011	0,00054008	1,6406e-011	1,922e-006	1,6629e-011	0,00030193	0,025969
		Déconnection	0	0	0	0	0	0	0	0
		Erreur d'action	6	15	300	0	9	0	5	0
		Erreur du réseau	4,4112e-007	0,0010068	0,019211	3,289e-011	0,00066849	5,0308e-011	0,00083872	6,892e-008
Zeros	Déconnection	0	0	0	0	0	0	0	0	
	Erreur d'action	2	0	5	0	9	0	11	0	
	Erreur du réseau	0,00010668	5,2428e-009	0,00026316	1,7537e-007	0,00031078	1,0719e-007	0,00032226	2,1418e-008	
	Déconnection	0	0	0	0	0	0	0	0	
	Erreur d'action	3	0	4	0	10	0	12	0	
	Erreur du réseau	0,00019678	1,0658e-006	8,5933e-005	1,7414e-007	0,00053335	6,1134e-008	0,00086812	1,7255e-006	

Figure 3.9 Evaluation des apprentissages

La figure 3.9 montre un des exemples réalisés pour l'ensemble des réseaux de neurones testés. La base de données utilisée est celle de 2056 exemples. L'architecture type 5 correspond à un réseau de 12 neurones dans la couche cachée de Variante 2 (fonction de transfert sigmoïde). Il est aussi montré les différentes initialisations des poids du réseau : aléatoire, zéro, gaussienne, binaire 1 et binaire 2 en fonction de la direction. Dans la colonne suivante on constate les erreurs de déconnexions et d'actions obtenues lors de la simulation du réseau, et l'erreur du réseau correspond au résultat de l'apprentissage. Les suivantes colonnes correspondent aux directions apprises par les réseaux respectifs et à leurs résultats.

Pour évaluer chaque type de réseau on observe trois de ses résultantes : la performance de l'apprentissage, les erreurs d'action et de déconnexion commissent lors de son implémentation dans le système multi-agents.

Les résultats obtenus avec les différentes initialisations sont variés. Cependant on constate que ceux qui sont initiés avec les poids des couches cachées à zéro ont une meilleure performance. En moyenne ils réalisent l'apprentissage plus vite dans toutes les architectures de réseaux confondus, ils apportent aussi moins d'erreurs de sorties.

3.3.2 Evaluation de l'apprentissage

Grâce à ce travail nous avons remis en question les règles de reconfiguration. C'est dans cette mesure qu'il nous a sembler raisonnable de réaliser l'annexe 1 : réviser à la main les 256 différents cas de voisinage d'atomes. De cette révision a émergé un cas de configuration pas pris en compte (d'ailleurs pas nécessaire dans la démarche de nos hypothèses). Il s'agit des atomes seuls qui ne sont pas intégrés dans la molécule, pour exemple voir Annexe 1 AV vs. AP N°21. Nous sommes bien face à une configuration dépassant les limites des hypothèses, mais avec une caractéristique émergente très

intéressante : une façon de réaliser un regroupement des modules isolés.

Dans *l'apprentissage supervisé* nous avons vérifié que sous certaines conditions le système est capable d'apprendre à exécuter des déplacements collectifs sans perdre la connexion avec le groupe (voir § 2.1). Dans *l'apprentissage à partir d'observations* nous avons découvert que des situations de blocage émergent des comportements appris, même si la molécule commence à avancer vers la cible.

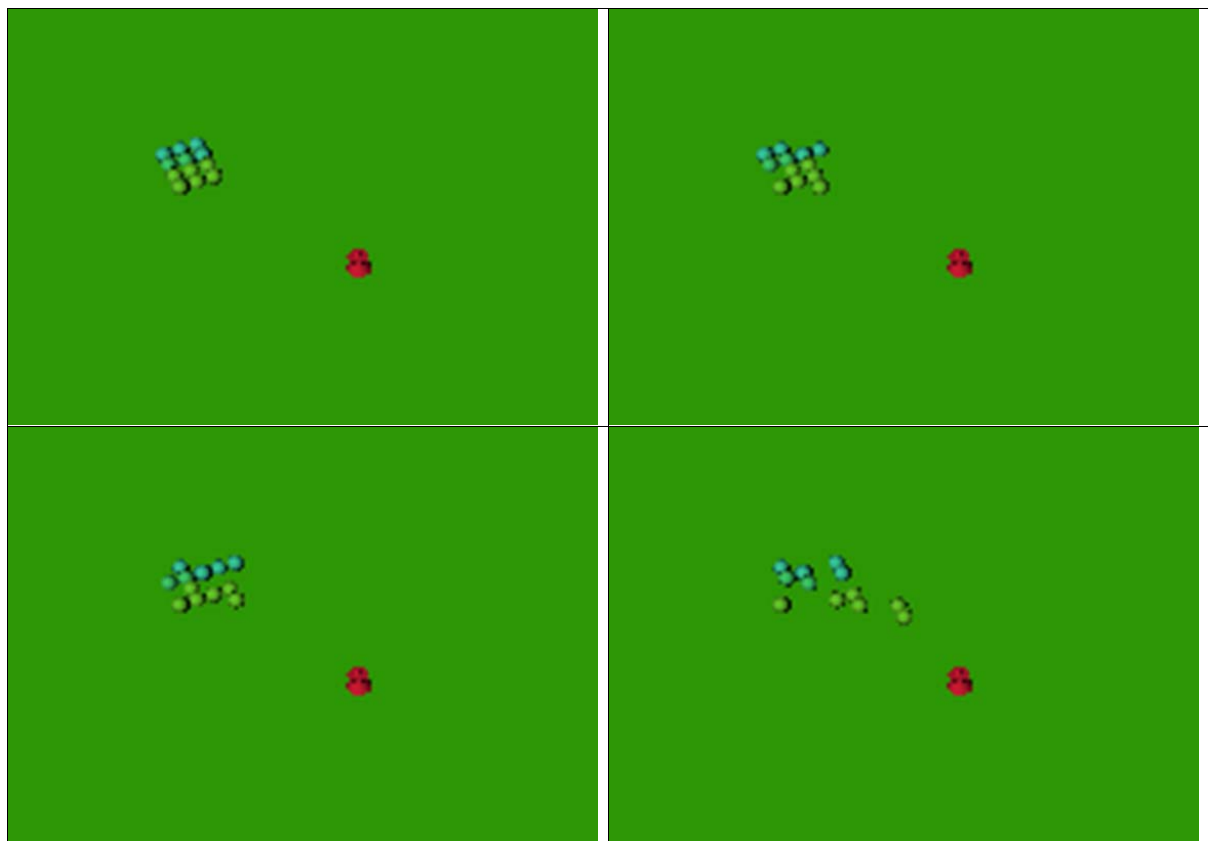


Figure 3.10 Exécution d'un réseau appris

Avec le dernier cas d'apprentissage, nous avons constaté aussi qu'un passage à l'échelle de la molécule pouvait changer radicalement de comportement, ce que nous n'avions pas systématiquement testé sur les méthodes réactives. C'est à dire que la molécule se déconnecte dès les toutes premières actions faites par les atomes si le nombre d'atomes est important.

Le réseau de neurones finit toujours par se déconnecter. En analysant la situation juste avant la déconnexion on pensait trouver un faux mouvement parmi ceux qui étaient programmés dans l'algorithme réactif qui générait donc une erreur d'apprentissage. Or, en regardant de plus près nous avons constaté que ces formations de « doigts » étaient fréquentes et relevaient de plusieurs types de problèmes que nous allons détailler maintenant.

3.4 Des problèmes liés aux hypothèses

Nous allons distinguer deux types de problèmes qui engendrent un blocage. L'un est intrinsèque aux hypothèses retenues pour faire ces simulations de déplacements collectifs.

L'autre, plus général, est lié à la politique d'ordonnement des calculs dans les atomes ou aux conditions initiales de la molécule.

3.4.1 Les problèmes intrinsèques au modèle

L'expérimentation sur les réseaux de neurones nous a fait découvrir qu'il existe des configurations pour lesquelles la molécule ne peut pas bouger avec les hypothèses retenues. Nous présentons ici deux types d'organisations des atomes pour lesquels le blocage existe

A. Le « U » avec un potentiel régulier

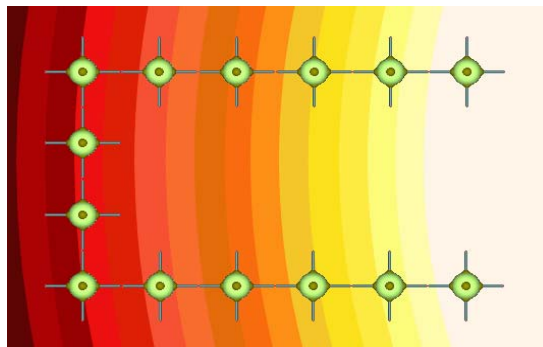


Figure 3.11 Formation de « U »

Dans ce cas la chaîne ouverte fait que les deux atomes en tête (à droite sur la figure ci-dessus) ont atteint la position de gradient minimale. Aucun des autres atomes dans la chaîne n'a le droit de bouger sous risque de détruire le connexion de la molécule.

Nous avons l'émergence d'un deadlock. Le cas présenté ci-dessus est un cas où deux « doigts » se forment. Il est possible d'avoir des configurations à plusieurs doigts.

B. Le « O » avec un potentiel quelconque

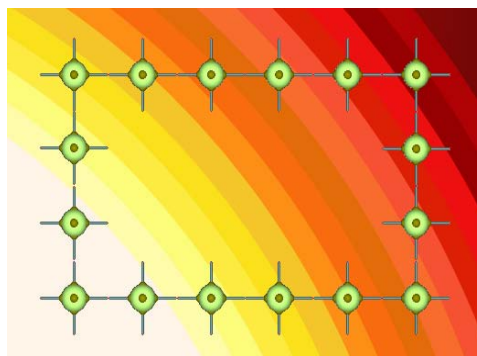


Figure 3.12 Formation de « O »

On remarque ici que la position en « O » pose le même problème quelque soit la position et le type d'attracteur. En effet aucun atome ne peut bouger sous risque de détruire la molécule.

Etant donné les hypothèses sous lesquelles nous avons placé cette étude, les atomes n'ont pas de moyen de communication. Comme aucun atome n'a de vision globale, il est impossible de détecter une chaîne fermée pour laquelle l'atome pourrait bouger sans déconnecter la molécule.

De plus, même en le sachant, il faudrait élire celui qui a le droit de se déplacer vers l'attracteur et comme il n'y a pas de communication entre les atomes ceci serait impossible.

Ce type de configuration semble impossible à atteindre « naturellement » avec un algorithme de descente de gradient. Pourtant, nous allons montrer comment ce type de configuration peut émerger.

3.4.2 Les problèmes dus à la dynamique

A. L'émergence du «U»

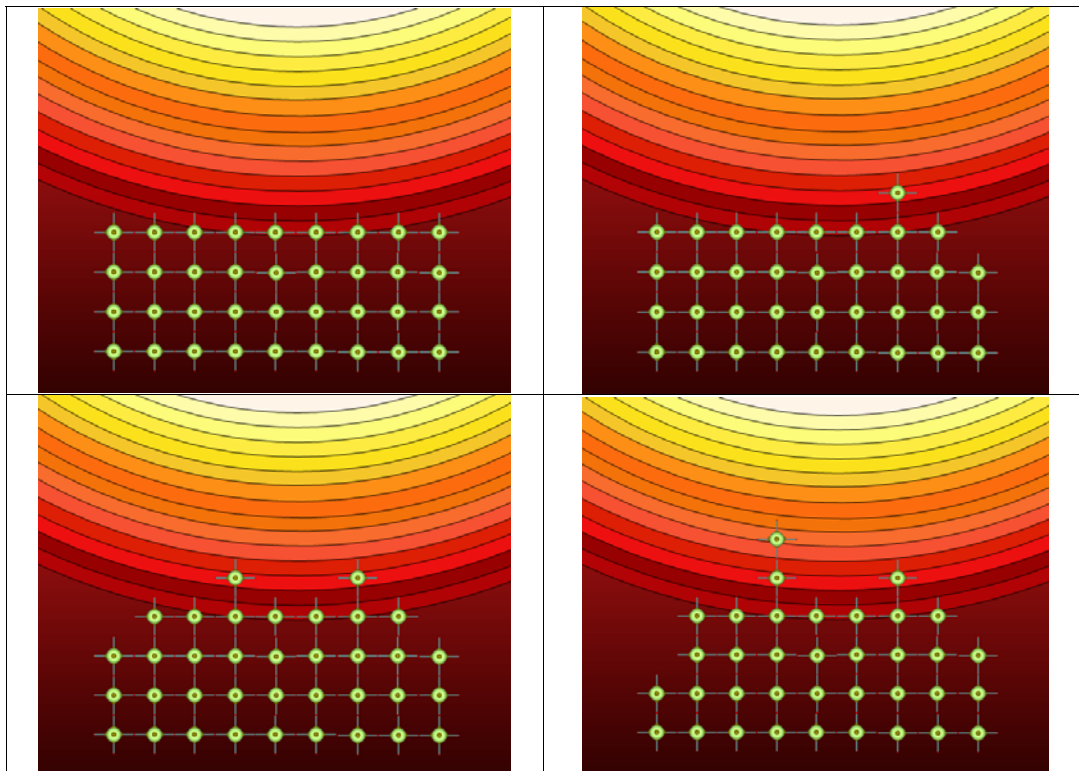


Figure 3.13 Formation de « U »

L'exemple ci-dessus est une illustration de la dynamique d'exécution pour laquelle la configuration en « U » émerge. Avec un nombre suffisamment grand d'atomes, les atomes situés à gauche et à droite descendent le potentiel jusqu'à une position où ils atteignent un minimum. A partir de la création de ces deux « doigts » tournés vers la cible tous les autres atomes vont venir se mettre au bout pour être encore plus proche de l'attracteur, la situation de blocage émerge.

Le système est bloqué puisqu'il existe plusieurs minimums. En effet les lignes concentriques de la figure 3.13 représentent des lignes équipotentielles.

Ce système est très sensible aux conditions initiales et à l'ordre dans lequel bougent les atomes.

Nous plaçons ici des simulations qui montrent différentes émergences de blocage ainsi que la cause de leurs apparitions.

Exemples

Simulation 1 : Emergence de U sous un ordonnancement séquentiel

Les conditions initiales sont un des facteurs qui influencent le déroulement des reconfigurations successives. Nous montrons ci-dessous deux simulations qui ne diffèrent que de la distance à la cible dans la configuration initiale et avec 25 atomes dans la molécule.

Simulation sans blocage

Potentiel régulier + Forward total + ordonnancement séquentiel

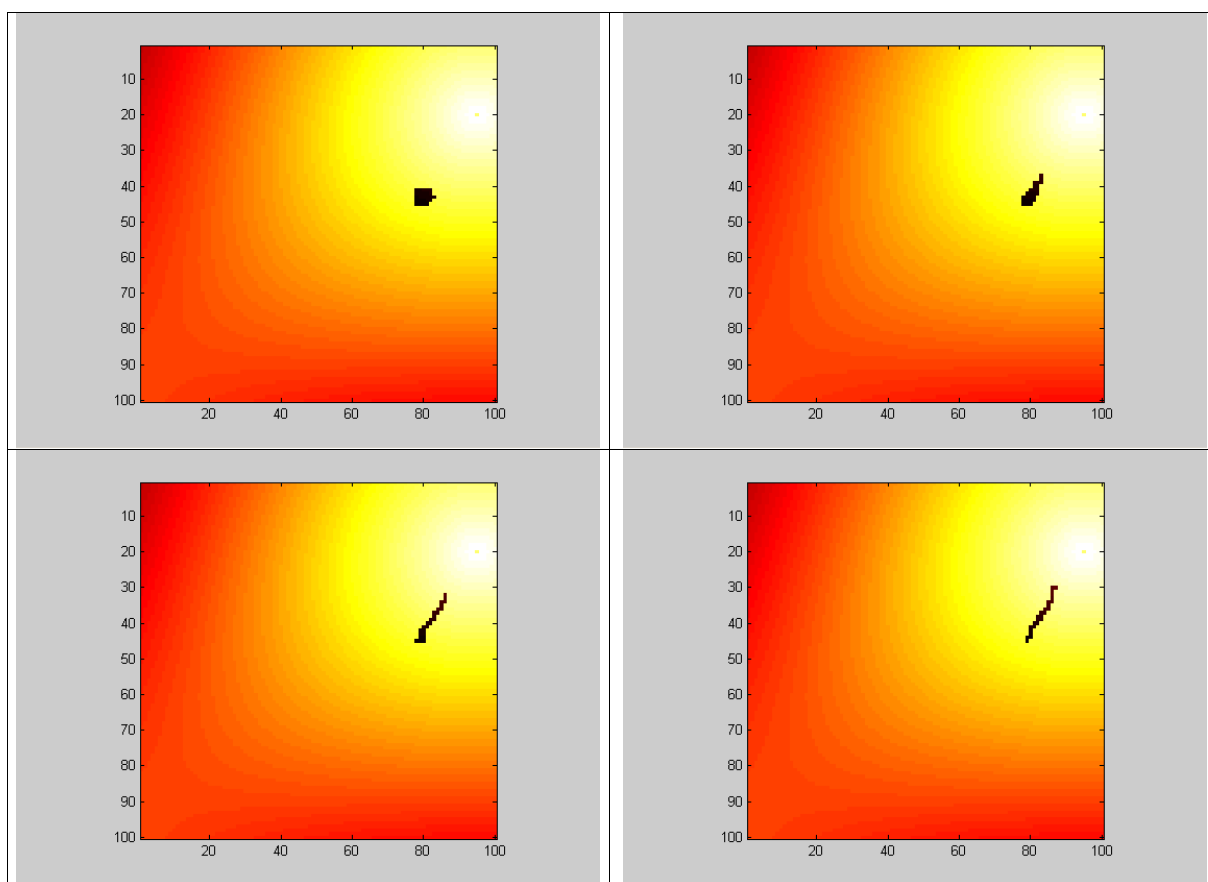


Figure 3.14 Emergence du comportement désiré avec un potentiel régulier et ordonnancement séquentiel

Dans ce cas le système converge vers la cible la ligne d'atomes se crée et à partir de là l'algorithme est stable et converge toujours.

Même simulation mais avec blocage

Mais dans ce cas, les conditions initiales sont différentes. La molécule est un peu plus loin de l'attracteur avec le même nombre d'atomes que précédemment.

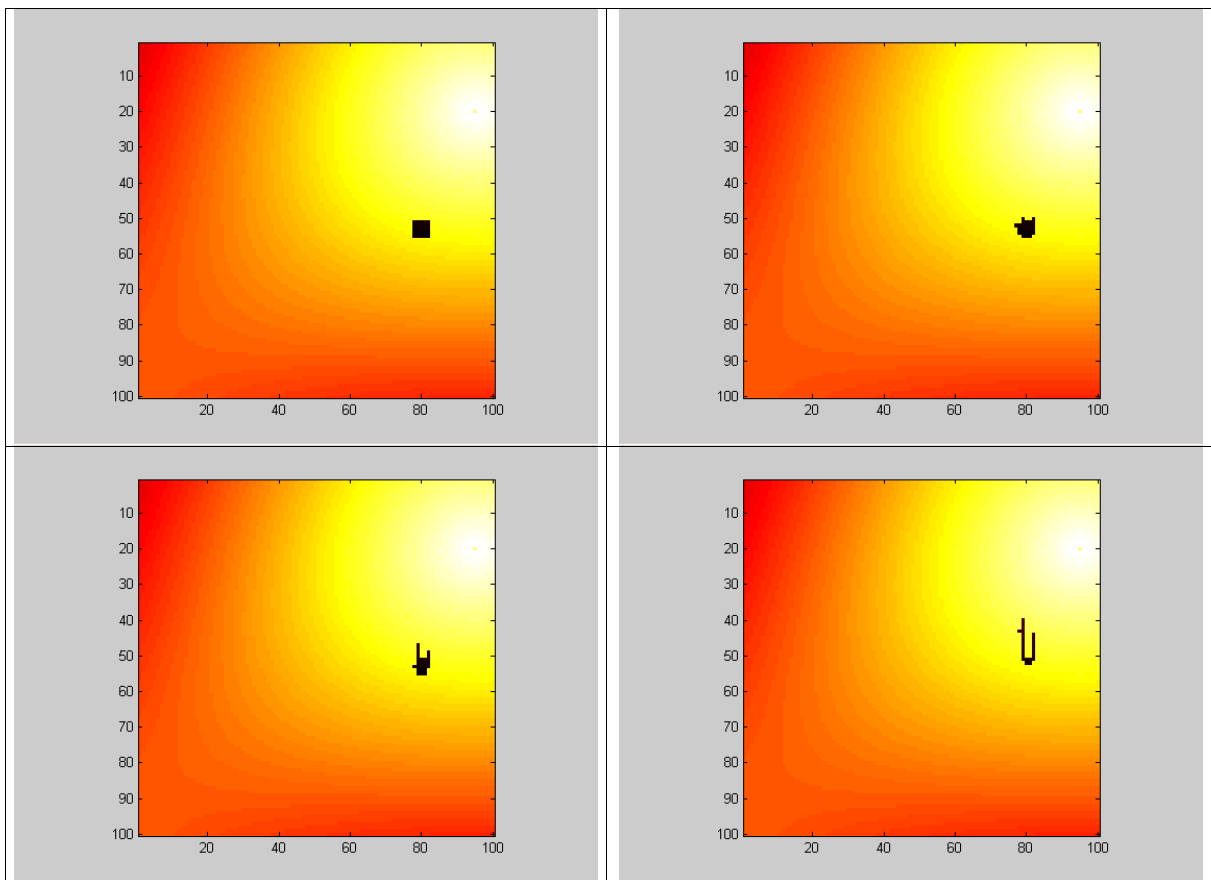


Figure 3.15 Emergence du U avec un potentiel régulier et ordonnancement séquentiel

En fait on constate que plus on s'éloigne de l'attracteur et plus la possibilité de voir émerger des « doigts » est importante.

Simulation 2 : Emergence de U sous un ordonnancement aléatoire

En général, on trouve aussi l'émergence du U avec ordonnancement aléatoire.

Dans les mêmes conditions initiales que celles pour lesquelles un ordonnancement séquentiel réussi, l'ordonnancement aléatoire se trouve face au blocage du U.

Potentiel simple + Forward total + ordonnancement aléatoire

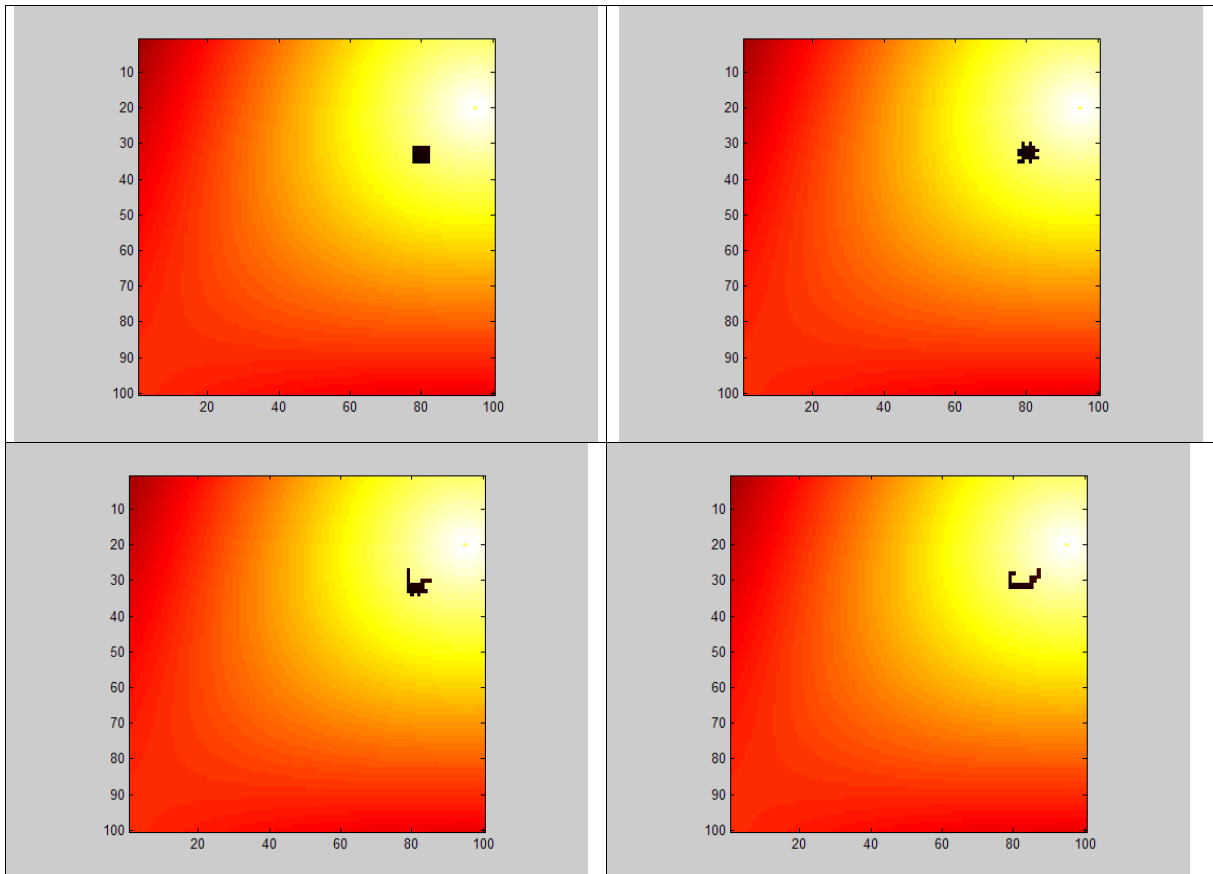


Figure 3.16 Emergence du U avec un potentiel régulier et ordonnancement aléatoire

L'apparition des « doigts » est plus rapide. L'ordonnancement régulier est en fait un cas particulier de l'ordonnancement aléatoire.

Simulation 3 : Emergence de U sous un ordonnancement aléatoire « doigts » multiples

On peut aussi avoir l'émergence de plusieurs doigts.

Potentiel simple + Forward total + ordonnancement aléatoire

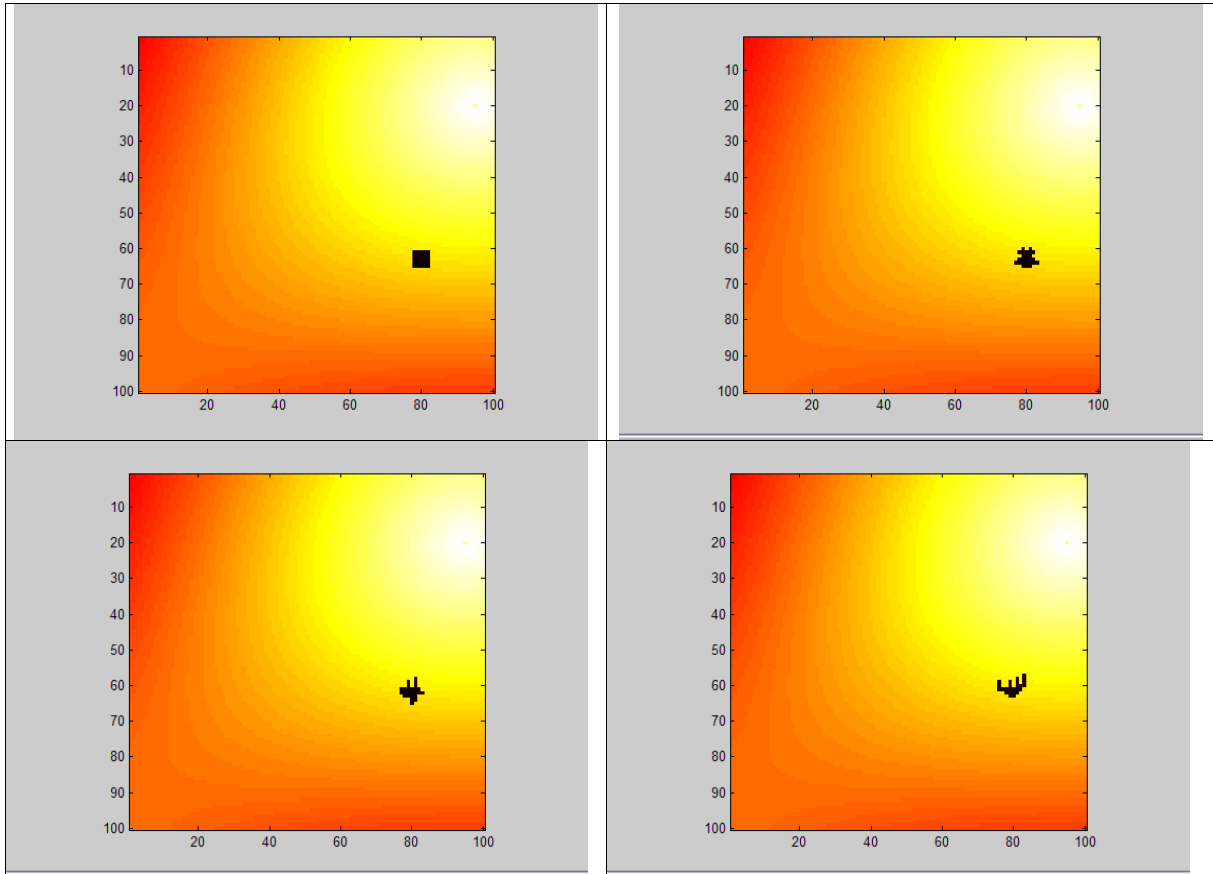


Figure 3.17 Emergence du U avec un potentiel régulier et ordonnancement aléatoire

En s'éloignant de la cible le nombre de doigts qui se forment est plus important.

B. L'émergence du «O» sur algorithme forward

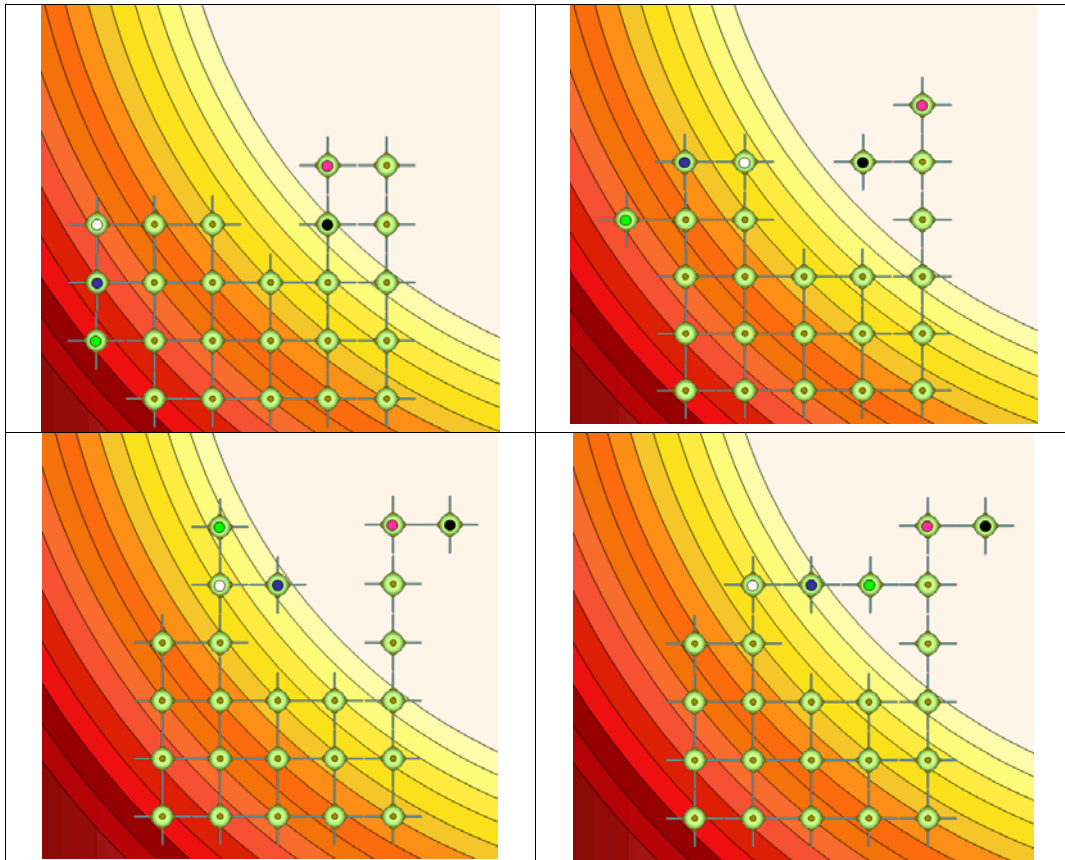


Figure 3.18 Dynamique de formation du O

Ici l'organisation qui émerge est de type « O ». Dans l'exemple ci-dessus, à partir de la configuration initiale, plusieurs solutions peuvent émerger. Soit la convergence vers la cible soit (c'est ce qui est représenté sur la figure ci-dessus) l'émergence d'un « O ».

L'atome marqué d'un point blanc se déplace de deux positions en suivant le gradient (il ne bougera plus par la suite). Puis, le noir va faire quatre déplacements pour atteindre la position qu'il gardera. Enfin, le vert va se déplacer en suivant le chemin qu'a emprunté le noir avant lui puis aller à la position finale où il va se reconnecter avec un autre atome de la ligne à droite. A ce moment le « O » est formé.

Suivant le type de programmation des mouvements élémentaires, il est possible que le trou se résorbe (si des mouvements de « glisseur » à l'intérieur sont autorisés). Néanmoins le « O » peut ne pas se résorber si tous les atomes en arrière du « O » passent en avant vers l'attracteur. La configuration atteinte est alors du type défini dans les contraintes structurelles § 3.4.1.

C- L'émergence du «O» sur algorithme backward

Lorsqu'un algorithme backward tente d'éviter des obstacles il est possible d'avoir émergence du « O ». Suite à un déplacement élémentaire de cet algorithme, un atome peut entrer en connexion avec deux atomes simultanément créant ainsi un « O » lorsque la réabsorption est utilisée.

Exemples

Simulation 1 : Emergence de O sous un ordonnancement aléatoire

Potentiel simple + Forward + ordonnancement aléatoire

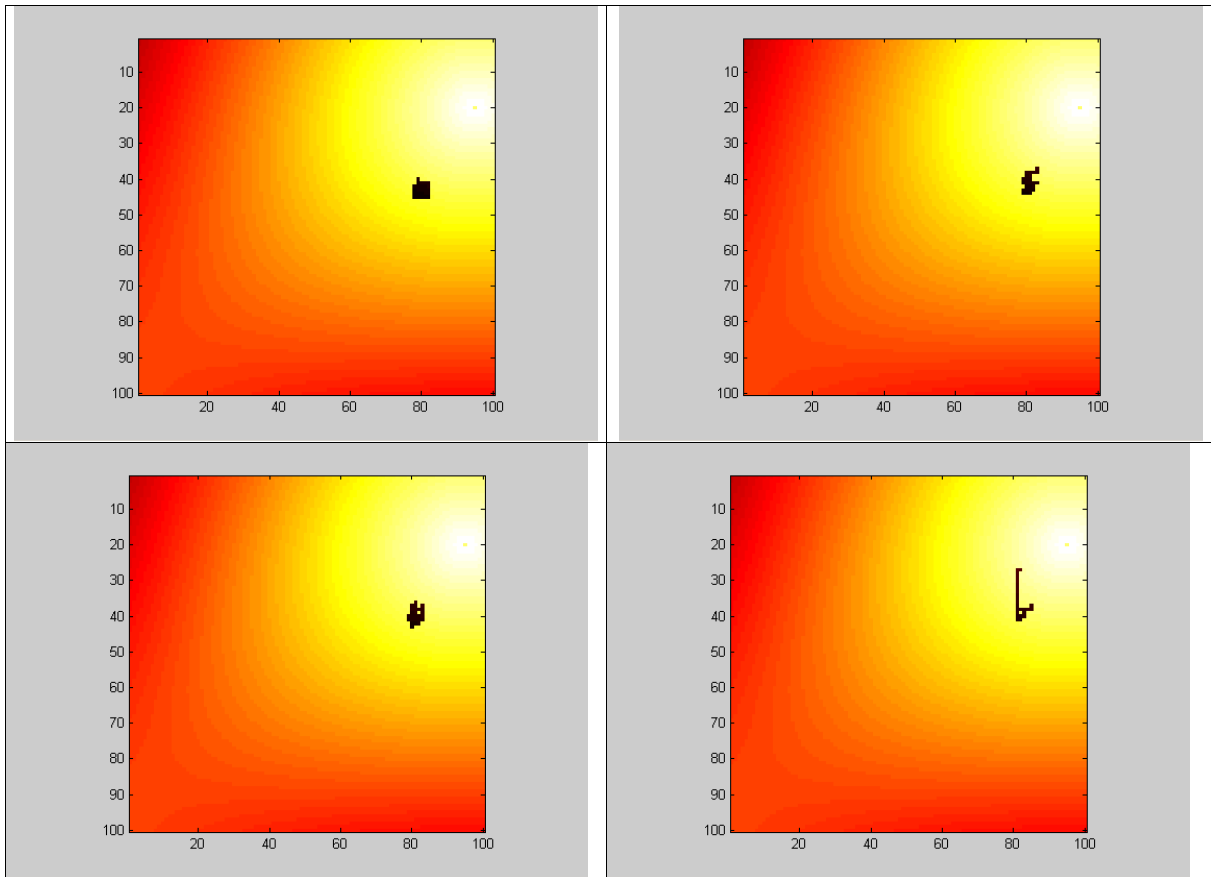


Figure 3.19 Cas d'émergence sous ordonnancement séquentiel

Nous avons ici une illustration de l'émergence d'un comportement en O sous un algorithme de type forward.

Simulation 2 : Emergence de O sous un ordonnancement séquentiel

Potentiel simple + Backward + ordonnancement séquentiel

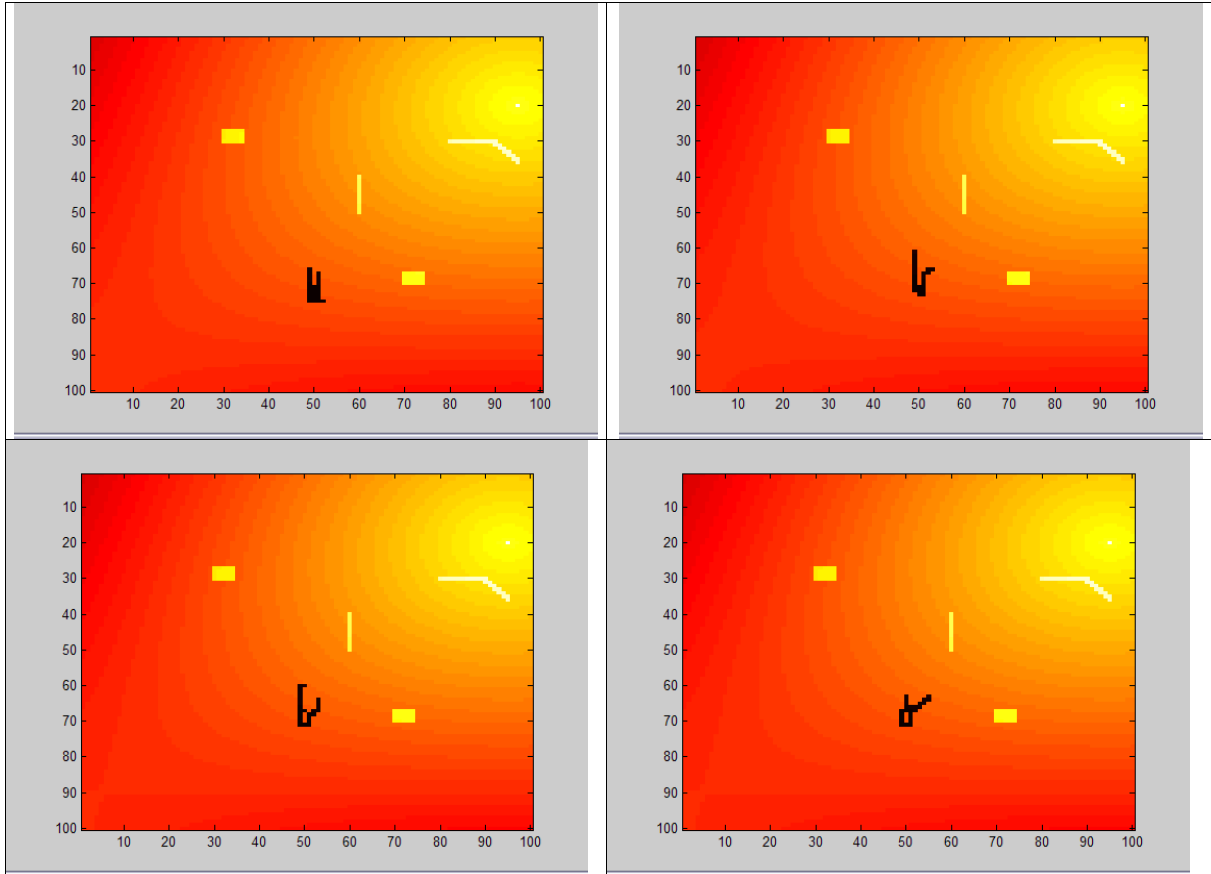


Figure 3.20 Emergence du O avec un potentiel régulier et ordonnancement aléatoire

Ce cas présente un algorithme backward qui tente de sortir de la création d'une « U ». En revenant le « O » se forme et les robots à partir de la ne pourront plus remplir le vide créé.

Simulation 3 : Potentiel double + Forward + ordonnancement aléatoire

Cette simulation est intéressante car suivant l'exécution il peut y avoir convergence ou pas avec les mêmes conditions initiales.

Cas de convergence de la simulation

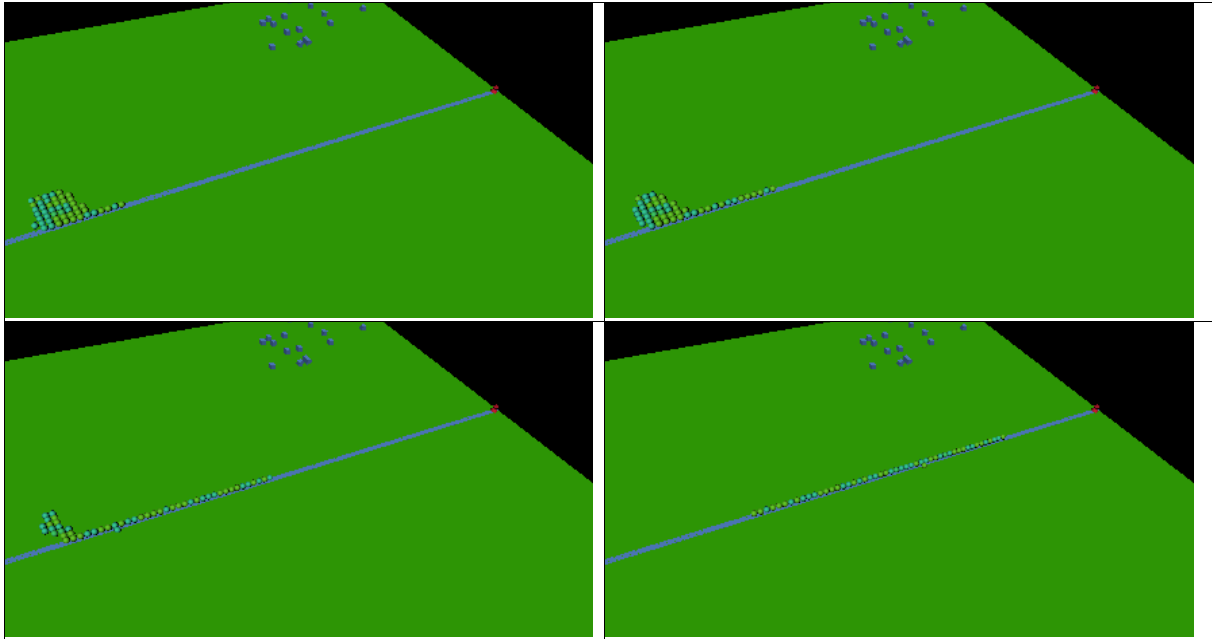


Figure 3.21 Formation qui marche

Cas d'émergence de O et de U

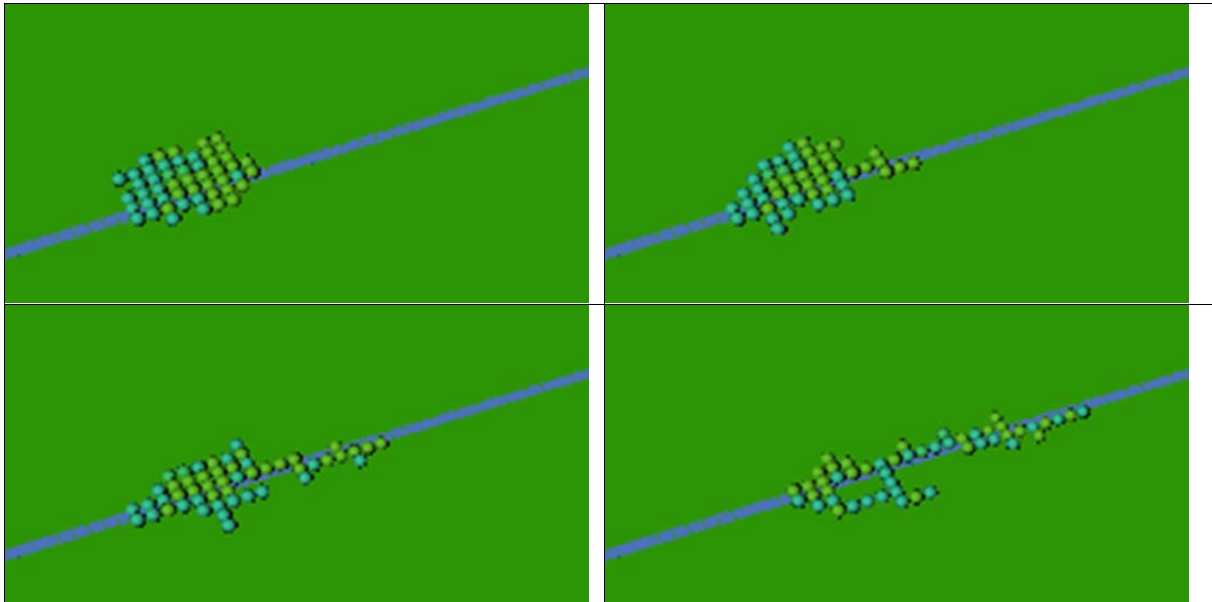


Figure 3.22 Formation de « U »

3.5 Synthèse

On peut donc résumer les différents types d'émergences constatés sous la forme de tableau :

Potentiel Régulier	Ordonnancement séquentiel	Algorithme Forward	Fonctionne sur configuration petites ou courtes distances Le passage à l'échelle est appréciable avec l'émergence de "U" dans les figures 3.14 - 3.15
		Algorithme Backward	Emergence de "U" dans la direction privilégiée, mais arrive se reconfigurer et continuer d'évoluer sous des actions bornées (fig. 3.4 - 3.20)
	Ordonnancement aléatoire	Algorithme Forward	Emergence de "U" avec un ou plusieurs doigts, voir figures 3.16 - 3.17 Emergence de "O" dans la figures 3.19
		Algorithme Backward	Emergence de "O" (fig. 3.20)
Potentiel Double	Ordonnancement séquentiel	Algorithme Forward	Fonctionne avec un potentiel double Emergence de "U" sous de mauvaises conditions initiales, voir figure 3.21
		Algorithme Backward	Emergence de "U et "O" testé à la mains
	Ordonnancement aléatoire	Algorithme Forward	Emergence de "U" en descente de gradient total sur potentiel double création O", voir figure 3.22
		Algorithme Backward	Emergence de "U et "O" testé à la mains

Chapitre 4. Vers une méthodologie de déplacement collectif

4.1 Analyse des simulations

L'étude des résultats obtenus au chapitre 3 montre qu'il existe deux types de problématiques à résoudre lors d'un déplacement collectif. L'un, de nature structurelle, l'autre, lié à la dynamique d'exécution. Résoudre un problème de déplacement collectif revient donc à garantir de résoudre ces deux types de problèmes.

Du point de vue structurel, on a pu montrer que dans le cas de systèmes reconfigurables l'absence d'unicité dans les forces d'attractions obtenues par le gradient, issu de l'attracteur, pouvait engendrer un blocage.

D'un autre côté, on a montré que la dynamique d'exécution pouvait engendrer des blocages si un module ne pouvait pas atteindre sa position de minimum de gradient dans la progression avant d'être bloqué par un autre module. D'ailleurs, on peut retrouver ses résultats réalisant une modélisation à la main avec des fiches et un peu d'entêtement pour aboutir sur une ordonnance explicite de blocage.

Dans ce chapitre nous proposons une approche qui va adresser ces deux problèmes et permettre un déplacement collectif de robots reconfigurables vers une cible.

Le Chapitre 5 montrera une possible mise en oeuvre d'une telle approche sur des robots physiques.

4.2. Unicité de la position à atteindre

On a vu au chapitre précédent qu'une chaîne ouverte de modules dont les deux extrémités sont sur le même potentiel engendre un blocage. Pour éviter ce problème, il convient donc de créer un gradient ne possédant autres types de propriétés.

Le potentiel double est une solution à ce problème. En effet, le potentiel double contient des points de même potentiel (voir figure) mais dont les valeurs du potentiel au voisinage ne sont pas identiques. Dans un tel potentiel, il existe une ligne de plus forte intensité dont le potentiel change uniformément en fonction de la distance à l'émetteur.

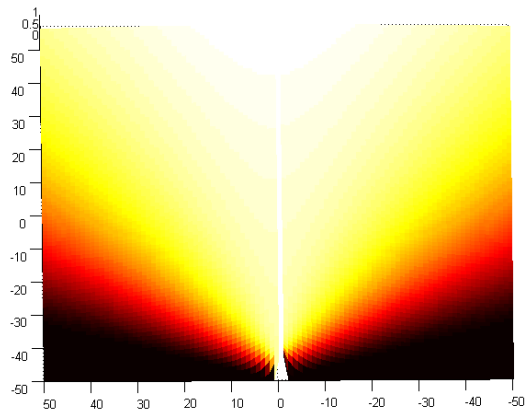


Figure 4.1 Le potentiel double

Chaque point est unique par son voisinage. Il est donc possible de créer un ordre partiel sur l'environnement comme se montre dans la figure 4.2.

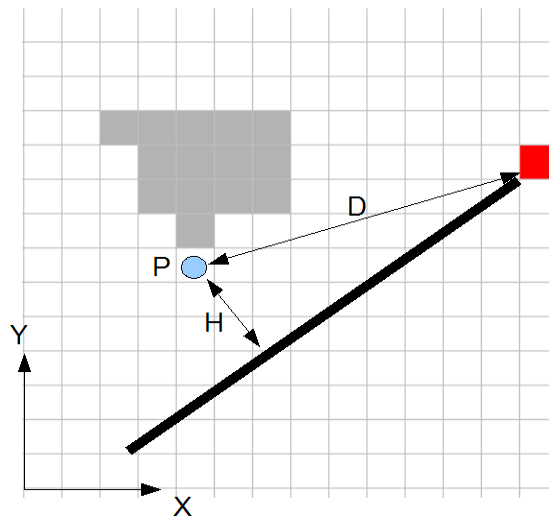


Figure 4.2 Point minimum au sens de l'ordre défini

-soit H la distance d'un point P à la ligne de plus forte intensité.

-soit D la distance à la cible.

On dira que $(H,D) < (H',D')$ si $(H < H')$ ou $[(H \approx H') \text{ et } D < D']$. La relation « $<$ » est une relation d'ordre. Comme proposé dans la descente d'un gradient double (cf 2.3), la relation \approx est due à la tolérance spécifique au maillage du système et la largeur de la ligne de plus forte intensité.

Il faut noter également qu'en regardant les projections sur l'axe des X et Y, il existe aussi un ordre en définissant $(X,Y) < (X',Y')$ en fonction de l'emplacement des composantes du système. On constate qu'un champ de potentiel double donne deux degrés de liberté pour l'information obtenue lors de la lecture du voisinage d'un point.

4.3. Une méthodologie pour garantir la convergence

Eviter l'apparition de doigts pendant l'exécution du déplacement de la molécule revient donc à éviter que deux positions soient minimales par rapport à une suite de déplacements élémentaires des modules. On a vu au chapitre précédent que les **conditions initiales** sont importantes. Nous supposons donc que les conditions initiales de la molécule vérifient les propriétés décrites ci-dessous.

Propriété 1: Unicité d'un point minimal

*Quelque soit la configuration connexe (c'est à dire sans trous dans la molécule donc sans « O ») et initialement compacte (en forme de carré, rectangle, losange...) de la molécule dans l'espace, pour un atome à la position M pris sur la périphérie de la molécule (donc capable de bouger) il existe pour l'atome une position P minimale à atteindre à partir de M . P est **unique**.*

Sur l'exemple la figure 4.2, le point P est le point minimum à atteindre pour tous les atomes capables de bouger de la molécule.

Cette propriété permet de ne pas avoir dans l'environnement initial de la molécule plusieurs buts locaux à atteindre. Elle est obtenue en utilisant un potentiel double.

Propriété 2: atomes actifs

Les mouvements élémentaires programmés dans l'algorithme réactif autorise le déplacement des modules que pour ce qui sont à l'extrémité d'une ligne et d'une colonne, montré dans l'annexe 1. Ces atomes sont appelés atomes actifs

Les mouvements autorisés de M à M' vérifient $M' < M$.

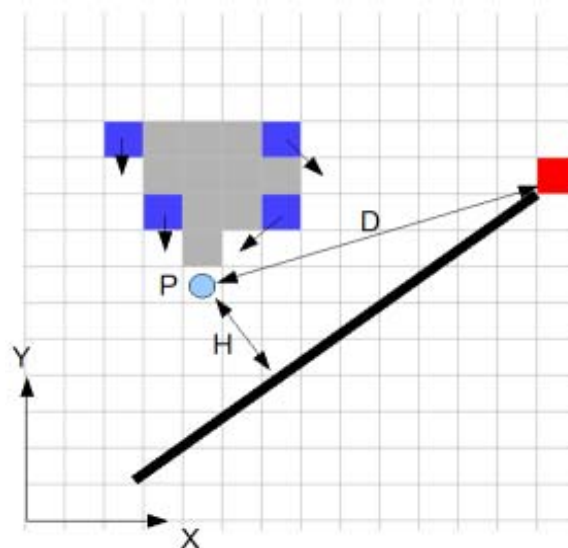


Figure 4.3 Atomes aux mouvements autorisés

Sur l'exemple on voit que seul quatre atomes vérifient les conditions ci-dessus et les flèches indiquent le mouvement autorisé.

Ceci permet d'éviter la formation de « trous » dans un ligne d'atomes. En fait certaines simulations menées ont montré qu'il pouvait y avoir convergence en relâchant cette contrainte. Mais l'objectif de ce chapitre est de garantir la convergence du système et donc cela oblige à durcir les conditions.

Propriété 3: chemin décroissant au point minimal

Pour tous les atomes vérifiant la propriété 2, il existe un chemin S unique décroissant de M à P . Le chemin est défini comme une suite finie de mouvements autorisés.

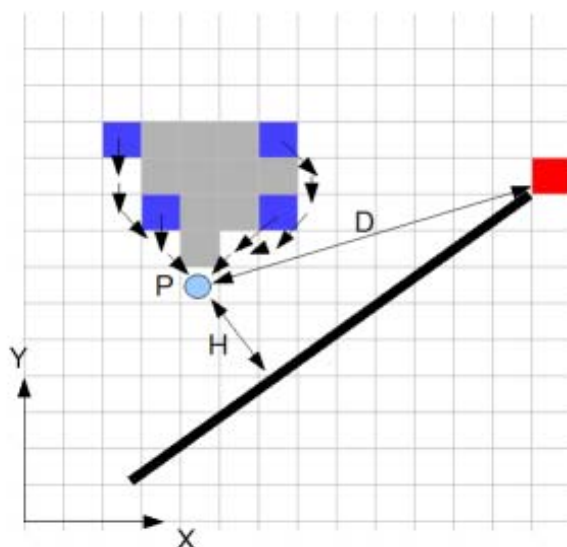


Figure 4.4 Chemins décroissants

Cette propriété de régularité de la structure permet d'éviter que la molécule ne contienne dans sa configuration une « bosse » qui pourrait engendrer la formation d'un doigt

Propriété 4 : progression

Un atome ne se déplace le long de son chemin décroissant de M à P que s'il ne bloque pas un autre atome actif qui possède un chemin jusqu'à P .

Cette propriété doit être vérifiée à l'initialisation du programme, nous montrerons qu'elle reste maintenue durant l'exécution du programme. La programmation des déplacements élémentaires permet aux atomes actifs qui sont en avant (au sens de l'ordre sur l'environnement) de bloquer l'avancement de ceux qui sont en arrière.

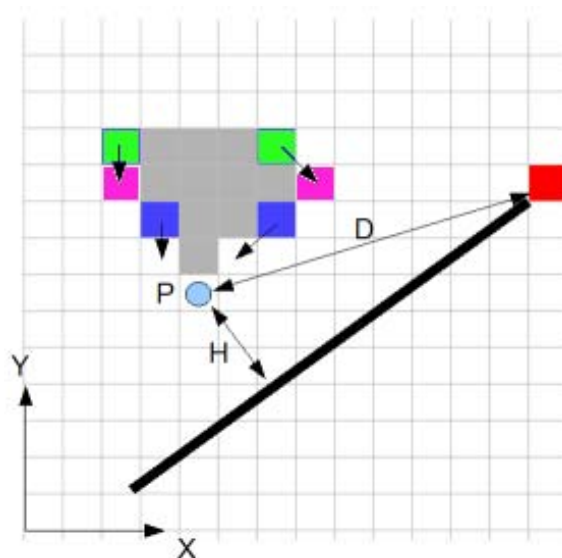


Figure 4.5 Chemins décroissants

Sur l'exemple ci-dessus les deux atomes en vert ne vont pas pouvoir dépasser les atomes bleus. Les atomes verts seront bloqués dans leurs progressions aux positions en rose.

Cette propriété force l'ordonnancement des mouvements des atomes. Elle sous-entend qu'un atome quelconque est capable de déterminer si les atomes avec lesquels il progresse possèdent ou non un chemin à la cible partielle P.

On voit que cela introduit une forme de communication. Comme nous le verrons au paragraphe suivant cela ne remet pas en cause les hypothèses de notre travail (chapitre 2) mais introduit une nouvelle capacité de perception dans les atomes.

Lemme 1

Si un atome actif atteint P alors le nouveau point minimal P' est unique.

Preuve :

Comme le point P est minimal alors le point P' est forcément voisin de P à cause de la régularité de l'ordre.

Ce point existe puisque l'environnement est uniformément décroissant (sauf sur la cible ou le programme s'arrête). Le point $P' < P$ est donc unique puisque P est unique et minimal par la propriété 1 et qu'il n'existe pas deux points identiques dans l'environnement..

CQFD

Lemme 2

Le nouveau point P' est accessible par tous les atomes actifs de la molécule.

Preuve :

Par la propriété 2 tous les atomes actifs ont la possibilité d'atteindre P par un chemin uniformément

décroissant. Comme P' est voisin de P , l'accès au point P' oblige donc à remplacer le dernier mouvement vers P par une suite de deux mouvements. Le premier pour contourner P , le second pour atteindre P' . On suppose ici que les mouvements élémentaires programmés dans les atomes permettent de résoudre toutes ces configurations de déplacement de P à P' (voir annexe 1).

La décroissance du chemin est évidente puisque P' est minimal par le lemme 1.

CQFD

Théorème

Si les quatre propriétés précédentes sont vérifiées alors la molécule convergera vers son but quel que soit l'ordre dans lequel les calculs seront effectués.

Preuve :

Faire cette preuve consiste à montrer que pour chaque but intermédiaire P :

- (1) il existe un atome actif M qui atteindra P .
- (2) tous les autres atomes actifs pourront accéder au but intermédiaire suivant P' .
- (3) les nouveaux atomes actifs qui vont se créer suite au déplacement de M auront les propriétés 3 & 4 et pourront accéder à P'

si tel est le cas alors comme il existe un chemin C décroissant du P initial jusqu'à l'attracteur par construction du potentiel, il existera toujours un atome pour atteindre les points intermédiaires de C sans blocage puisque la propriété 4 est vérifiée à chaque instant.

- (1) il existe un atome actif M qui atteindra P :

La position P sera nécessairement atteinte par un atome actif. A cause de la propriété 4 seul ceux qui ont un chemin libre jusqu'à P pourront l'atteindre. Si il y a 2 atomes dans l'espace 2D, l'un venant à P par la gauche et l'autre par la droite. Il n'est pas possible de prédire lequel atteindra P puisque cela dépend de l'ordonnancement. Par contre on sait que l'un des deux atteindra effectivement la cible puisque par la propriété 4 rien ne s'opposera à son déplacement.

Le (1) est vérifié.

- (2) tous les autres atomes actifs pourront accéder à P' :

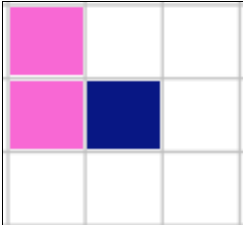
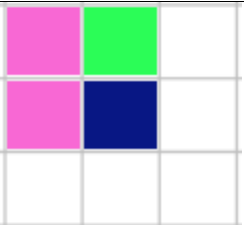
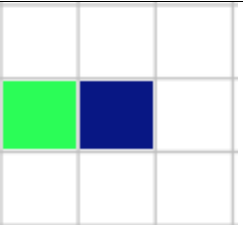
Tous ceux qui pouvaient atteindre P par le lemme 2 peuvent atteindre P' . Par la propriété 4 aucun d'eux n'est bloqué dans sa progression donc le (2) est vérifié.

- (3) les nouveaux atomes actifs qui vont se former par le déplacement de M auront les propriétés 3 & 4 et pourront accéder à P' :

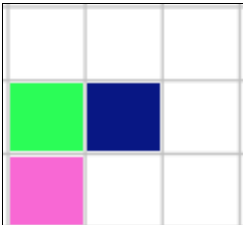
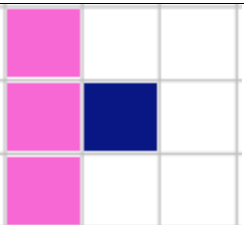

Comme l'atome M qui va atteindre P est un atome actif, alors par définition, il est extrémité d'un ligne et d'une colonne. Si on suppose que la position intermédiaire P est au Sud (en bas) on peut donc énumérer tous les cas de mouvements possibles pour M avec la création ou non de nouveaux atomes

actifs M' en direction du Sud. Ces déplacements sont dans trois directions possibles : sud ouest, sud, sud est (le raisonnement sur Sud peut s'appliquer de la même manière sur les trois autres directions à une symétrie prêt).

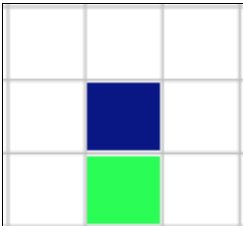
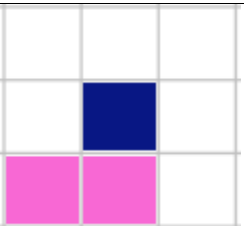
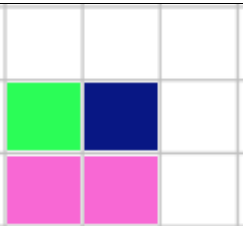
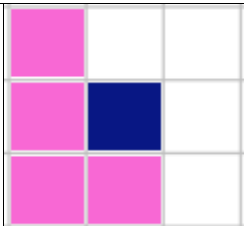
Déplacement Sud Ouest

		
Le déplacement de M ne crée pas d'atome actif	L'atome actif (vert) pourra prendre la place de M	L'atome actif (vert) pourra prendre la place de M au déplacement suivant

Déplacement Sud

		
L'atome actif (vert) pourra prendre la place de M	Le déplacement de M ne crée pas d'atome actif	L'atome actif (vert) pourra prendre la place de M

Déplacement Sud Est

			
Ce cas est plus délicat mais on peut vérifier qu'après deux mouvements de M l'atome actif (vert) pourra prendre la place de M	Le déplacement de M ne crée pas d'atome actif	L'atome actif (vert) pourra prendre la place de M	Le déplacement de M ne crée pas d'atome actif

Avec cette énumération de tous les cas possibles (à une symétrie prêt) on montre que les atomes actifs, M' créé par le déplacement de M, peuvent se positionner à la place de M ou en plusieurs déplacements. Puisqu'ils occupent la position de M les atomes M' vérifient la propriété 3.

Le lemme 2 permet également de savoir si ces nouveaux atomes actifs peuvent aussi atteindre P'.

Par programmation des comportements des atomes (cf §4.4) on supposera (comme précédemment)

que les nouveaux atomes actifs vérifient la propriété 4.

Le (3) est vérifié.

CQFD

Cette méthodologie contraignante permet de garantir la convergence de la molécule quelque soit le nombre d'atomes et quelque soit la position de la cible.

4.4 Comment forcer un ordonnancement

Comme on vient de le voir, il existe un atome (au moins) pour lequel il y a un chemin décroissant jusqu'à la cible. Les phénomènes issus de la dynamique d'exécution nous ont montré que des blocages arrivent parce que des atomes ne peuvent pas suivre leur chemin jusqu'à la position P parce qu'ils sont bloqués par d'autres atomes. Comme ils ont eu plus de temps de calcul (ordonnancement aléatoire), ils peuvent « doubler » les autres atomes actifs. Pour empêcher les dépassements et forcer l'ordonnancement des atomes dans le suivi de leurs chemins, nous proposons une méthode d'implémentation

La solution proposée ici consiste à ajouter une action et une perception dans les modules. Comme nous le montrerons au chapitre 5 cette solution est effectivement implémentable dans un système de robots modulaires.

A- Ajouter un état

Nous proposons ici d'ajouter une expression physique d'un état interne qui soit facilement détectable par les autres modules. Le module devra donc posséder une caractéristique détectable qui permet de lui donner un état à valeur booléenne. Cette valeur booléenne permettra d'exprimer au voisinage que l'on est « actif ou pas ». Tous les atomes actifs de la molécule seront dans l'état haut (1) alors que tous les autres seront dans l'état bas (0).

B- Ajouter une perception

Afin de pouvoir détecter le voisinage (cf 2.2.2) nous supposons qu'il existe un capteur (ou une méthode) permettant de connaître la présence ou l'absence d'un voisin. En plus de cette capacité de détection de présence, on ajoute ici la possibilité de déterminer l'état du voisin par la lecture d'une information booléenne. Une fois connu l'état des voisins on ne pourra progresser vers la cible que s'il n'y a pas d'atome à l'état actif dans la direction de choix.

Remarque

On peut constater que l'introduction de cette information sur l'état d'un module s'apparente à de la communication. En effet, un module peut « dire » à son entourage qu'il est « dans cet état ». Toutefois, (comme le sont les marques de phéromone) il s'agit d'une forme de marquage de l'environnement et d'une communication indirecte. Celle-ci n'est que détectable localement par un autre module.

4.5 Algorithme

Dans la suite nous utilisons la simplicité de l'algorithme « forward un coup » avec les nouvelles restrictions de mouvements pour éviter de le blocage d'un atome actif.

Algorithme “forward un coup avec perception d'état”

```
    si il existe des positions maximisant le potentiel
    alors vérifier pour ces positions l'accessibilité utilisant la perception d'état;
           choisir la meilleure action;
           se déplacer
    sinon ne pas bouger
```

Dans ce cas le module va faire un seul déplacement le long de la molécule puis va s'arrêter et attendre son prochain quota de temps. Pendant ce temps tous les autres modules peuvent avoir bougés. Si un atome est actif ses mouvements peuvent être retardés par un autre atome actif, mais cela ne signifie pas que ça sera la cause d'un changement d'état à passif.

Un atome actif ne va pas interférer avec un autre actif en le sollicitant pour une reconfiguration. Il ne va pas lui monter dessus, puisque cela le bloquerait dans la poursuite de son objectif.

4.6 Simulations avec perception d'état

Simulation 1 : émergence du comportement de déplacement vers l'attracteur sous un ordonnancement aléatoire

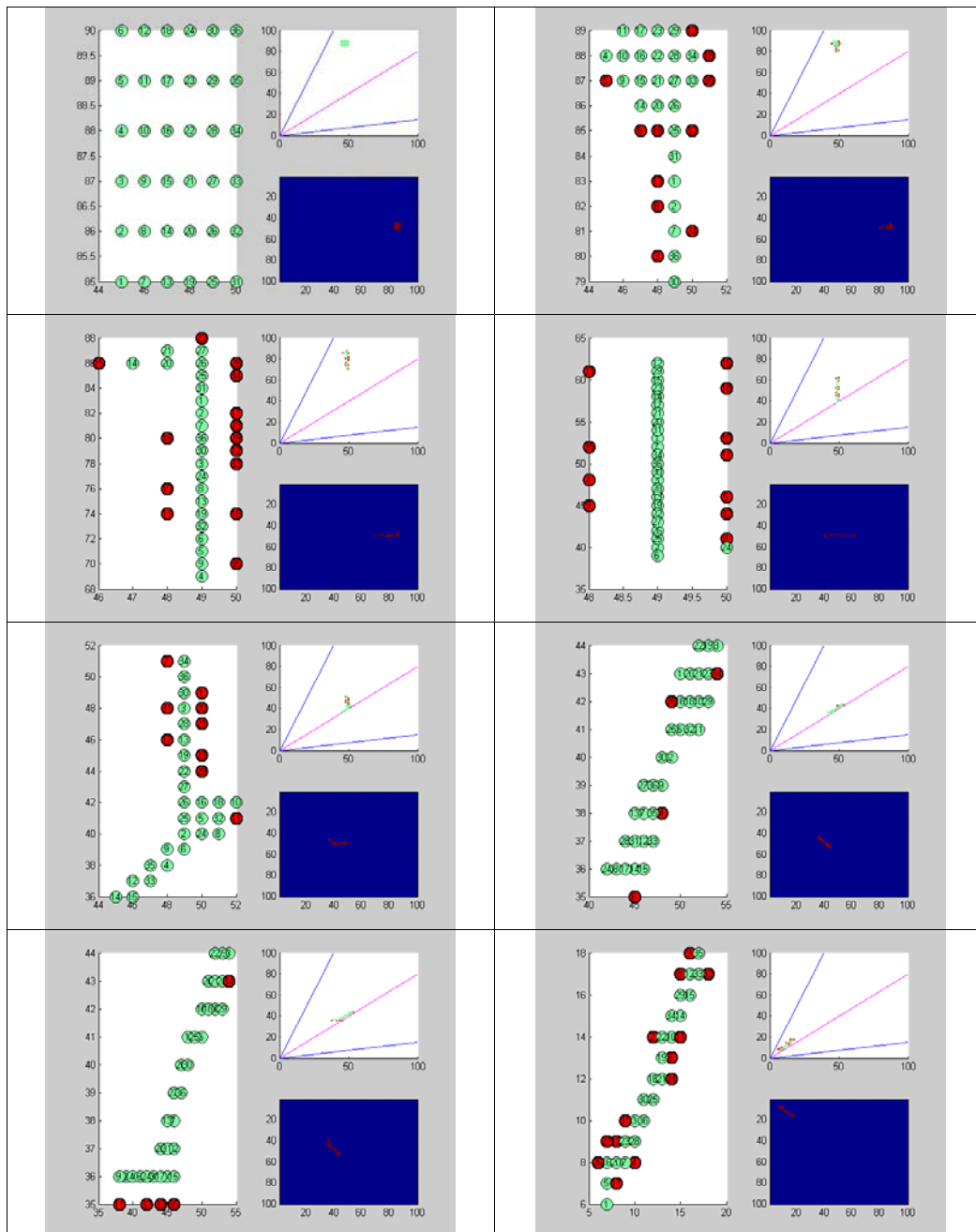


Figure 4.6 Simulation avec un double potentiel et perception d'état

Le système global du déplacement reconfigurable vers l'attracteur de la molécule émerge. Il lui faut une étape de transition et que tous les atomes changent de comportement une fois atteinte le ligne de plus forte intensité.

Simulation 2 : émergence d'un comportement de ligne avec un atome temporellement bloqué

Comme on aura pu le constater tout au long des simulations et expérimentations, les méthodes proposées possèdent des émergences caractéristiques. Les algorithmes proposés ont des comportements qui émergent lorsque le système est perturbé.

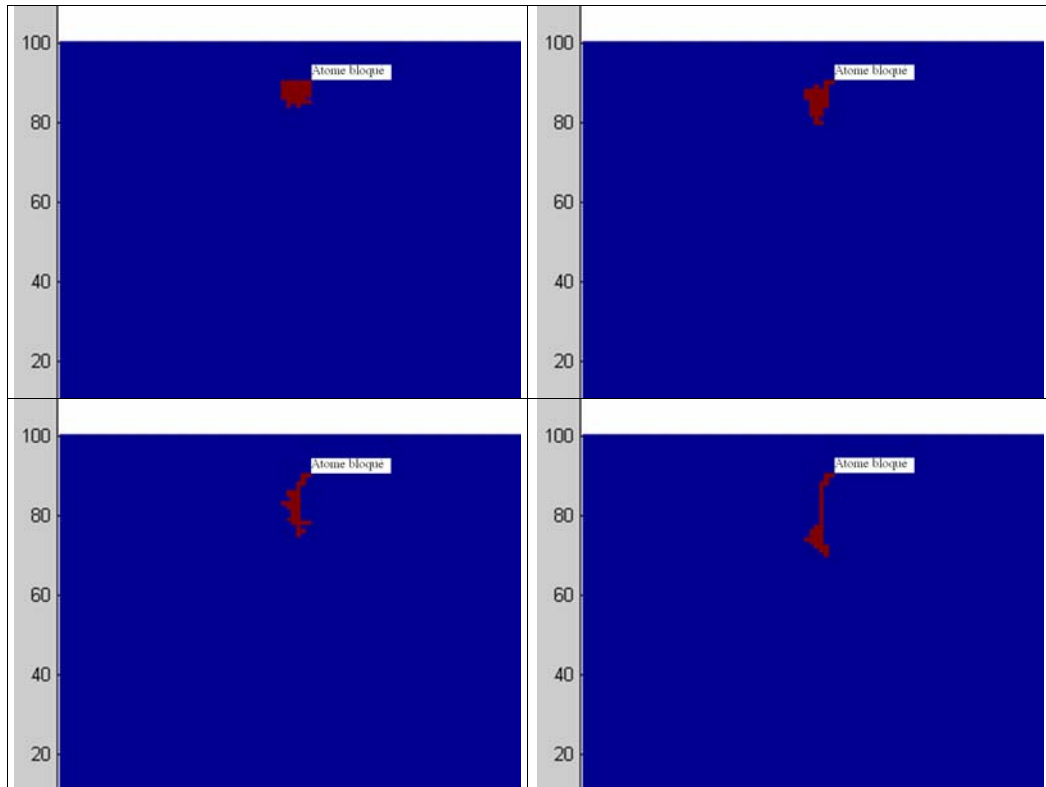


Figure 4.7 Simulation avec un double potentiel, perception d'état et un Atome bloqué

La molécule se déplace vers l'attracteur, mais le fait qu'un atome soit bloqué le robot reconfigurable s'allonge. Cette émergence ressemble à la déformation d'un chewing-gum quand on l'étire.

4.7 Généralisation de l'approche

On peut ici s'interroger sur la portabilité de cette méthode en particulier si on passe à l'espace à trois dimensions.

Il semble raisonnable de penser qu'en introduisant un ordre stricte et continu dans l'espace à 3 dimensions, la même méthode doit converger de la même manière. Bien entendu il faudra, comme c'est le cas ici, supposer que les mouvements élémentaires autorisés permettrait de suivre effectivement les chemins décroissants vers la cible.

Chapitre 5. Réalisation matérielle

Le chapitre 4 propose pour garantir la convergence de l'algorithme de déplacement vers la cible de considérer :

- que l'environnement soit ordonné par un potentiel double
- qu'il existe un état pour forcer l'ordonnement

Dans ce chapitre nous vérifierons qu'une telle hypothèse est effectivement implémentable sur des robots physiques tout en conservant les hypothèses du chapitre 2:

- pas de communication entre les atomes
- capacité à percevoir localement son environnement

5.1 Mise en oeuvre d'un potentiel double

L'étude des capteurs et émetteurs infra rouge effectuée dans la cadre du projet MAAM montre qu'il est possible de faire un potentiel double avec une diode.

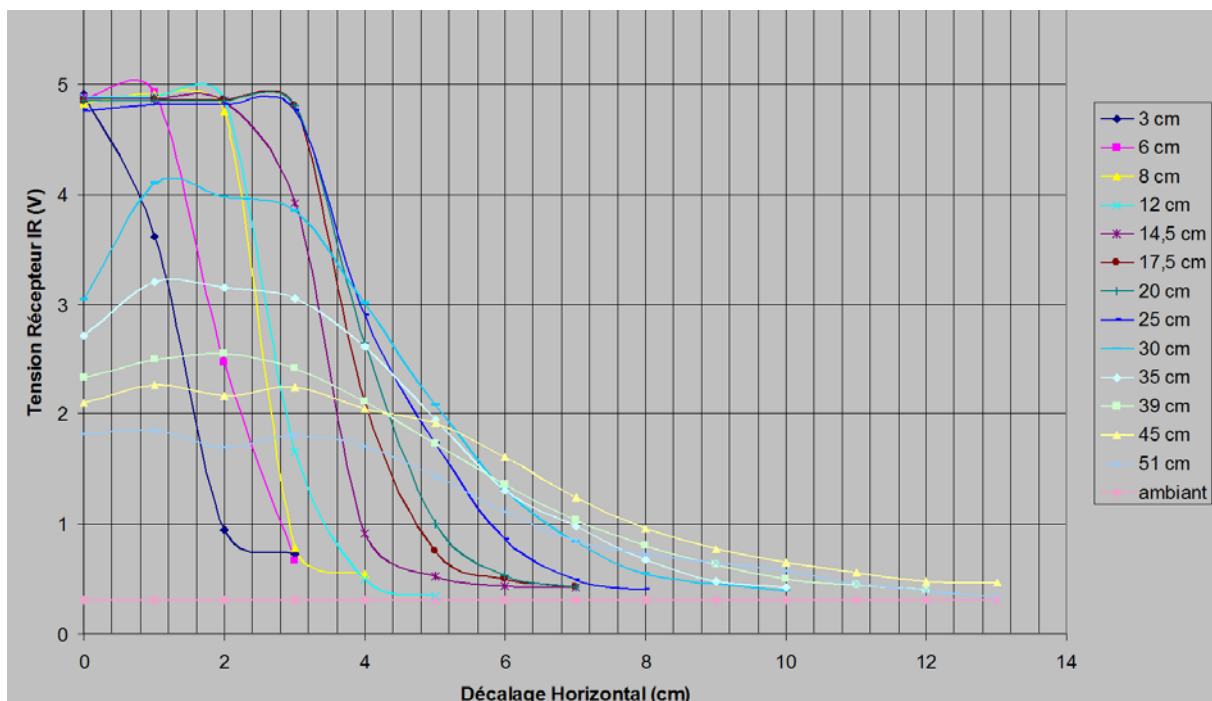


Figure 5.1 Lecture d'une diode par un récepteur à différentes distances

Cette courbe montre les mesures effectuées avec un couple de composant émetteur/récepteur infrarouge. Durant l'expérimentation le récepteur a été déplacé le long d'un axe perpendiculaire à la

direction de l'émission du faisceau lumineux, à des distances différentes.

On constate que l'on retrouve une fonction gaussienne lorsque le récepteur se trouve à une distance de l'émetteur compris dans l'intervalle 50-30 cm.

Lorsque l'on se rapproche des effets de saturation du signal apparaissent. La réponse est plus proche d'une réponse binaire de présence ou absence du signal. Toutefois lors du déplacement on arrive à vérifier que les courbes ne se superposent pas.

5.2. Reconstruction de la carte du potentiel

Dans le cas des modules MAAM, chaque module possède 6 pattes capables de se déplacer autour d'une liaison rotule. Le système d'attache est conçu pour recevoir en son centre un émetteur et en récepteur infrarouge.

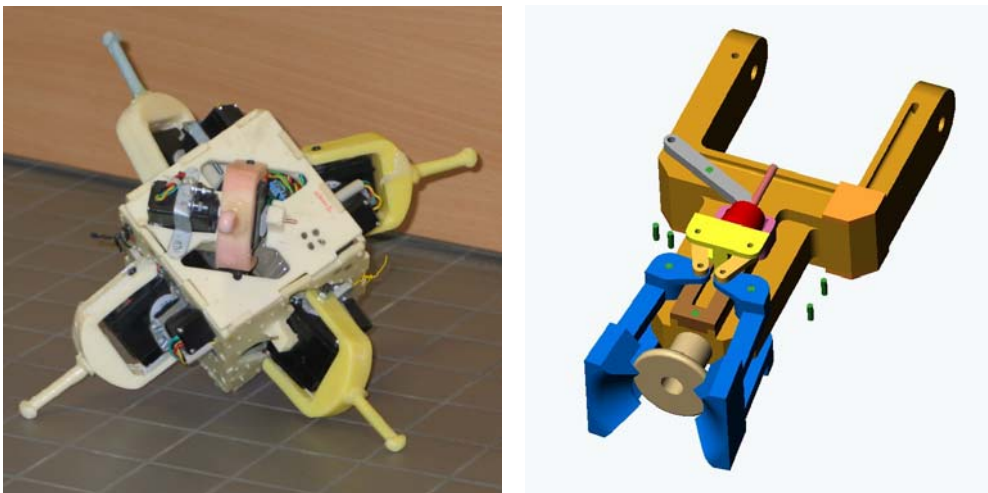


Figure 5.2 L'atome MAAM et son projet de système d'attache

Avec une patte il est donc possible de balayer l'environnement pour mesurer la position de l'émetteur.

Comme par ailleurs il est possible de mettre deux pattes dans la même direction, il est donc possible de faire d'utiliser une double information pour reconstituer l'environnement.



Figure 5.3 Lecture d'un signal IR par le capteur au bout de la patte

La géométrie des robots MAAM permet à deux pattes de scanner l'environnement dans une même direction. Il est donc possible de reconstituer le signal à partir de la fusion de ces deux capteurs. Comme l'information est stéréo on peut avoir des données précises sur la position et la distance à la cible.

Le potentiel double est clairement reconnaissable avec de telles reconstructions.

5.3. Mise en oeuvre d'un variable d'état

Au chapitre 4 on a proposé l'introduction d'une variable d'état pour forcer le séquençage de l'ordonnancement. Le paragraphe 4.4 explique ce qui est attendu d'une telle information.

Avec les atomes MAAM il sera possible de construire cette information.

Les atomes sont accrochés entre eux par les pattes. Chaque patte est pourvue d'un émetteur/récepteur infrarouge.

Une convention simple peut-être la suivante:

Lorsqu'un robot est connecté à un autre et qu'il laisse son émetteur allumé alors il est dans l'état « en mouvement ».

L'autre robot pourra avec la lecture sur son récepteur lire cette information.



Figure 5.4 Un prototype d'attache de modules MAAM

Cette solution ne nécessite pas d'actionneurs ni de capteurs supplémentaires que ceux déjà présents sur le robot et répond aux propriétés attendues.

Conclusion

Dans ce mémoire nous avons étudié le problème du déplacement d'un robot reconfigurable. Nous avons proposé des solutions réactives pour le résoudre. A partir de ces propositions nous avons constaté l'émergence de blocages ou d'oscillations dans les comportements globaux émergents. Les causes de ces blocages ou d'oscillations sont de deux types : d'une part de nature structurelle et environnemental, d'autre part liée à l'ordonnancement.

Nous avons pu prouver une méthode de programmation qui permet de garantir la convergence du déplacement d'un robot reconfigurable. Quelque soit la position de l'attracteur dans l'espace. Cette méthode suppose que la configuration initiale du robot respecte quatre propriétés et que la programmation de comportements élémentaires des modules garanti le suivi de chemin libre le long de l'atome jusqu'à un minimum local. Cette méthode a ensuite été validée en simulation et a montré des comportements émergents très intéressants, en particulier si on bloque momentanément un module dans sa progression.

Les perspectives que l'on peut donner à cette étude sont de plusieurs ordres.

Tout d'abord on peut s'interroger sur la généralisation de cette approche dans un environnement 3D. La question sous-jacente sera alors de savoir si les quatre propriétés proposées suffisent ou si la 3D introduit de nouvelles contraintes.

Une autre étude serait liée à la tolérance aux pannes. Les systèmes multi-agents ont comme propriété d'être résistant aux perturbations à cause du grand nombre d'agents qui coopèrent. La question qui poursuivrait notre étude serait de savoir comment on pourrait intégrer les pannes de modules au sein de la molécule sans perdre la propriété de convergence.

Le dernier axe est lié à la programmation du comportement émergent de la molécule. On a vu durant cette étude émerger des combinaisons en « U » avec un ou plusieurs « doigts ». La question que l'on pourrait se poser est comment se servir de ces propriétés émergentes pour programmer par l'environnement la structure pour réaliser une tâche donnée. En effet la formation de doigts pourrait être utile si l'on a besoin de fabriquer un manipulateur. Il faudrait donc pourvoir faire émerger cette forme à volonté.

L'émergence des systèmes multi-agents est de grande actualité même au sein des théories de physique fondamentale. Dans nos simulations de systèmes multi-robots nous avons trouver qu'il faut compter sur des lois de comportements simples. Le résultat de ses lois est observable dans le déplacement collectif. Les résultats obtenus avec une même loi de commande mais avec un système perturbé peut changer la formation finale du système. C'est pourquoi l'étude des systèmes multi-agents et leurs interactions reste de grande actualité dans des domaines de recherche scientifique.

Bibliographie

- [AMB 06] Frédéric Amblard, Denis Phan.
Modélisation et simulation multi-agents, applications pour les Sciences de l'Homme et de la Société.
Lavoisier, Hermes science, Paris 2006.
- [ARK 98] Ronakd C. Arkin,
Behavior-based robotics
The MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1998.
- [ARL 04] Angelo Arleo, Fabrizio Smeraldi, and Wulfram Gerstner,
Cognitive Navigation Based on Nonuniform Gabor Space Sampling, Unsupervised Growing Networks, and Reinforcement Learning
IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 15, NO. 3, pag 639, 2004
- [ARN 00] Pierre Arnaud,
Des moutons et des robots, architecture de contrôle réactive et déplacement collectif des robots
Presses Polytechniques et Universitaires Romandes, 2000.
- [ASI 50] Issac Asimov
Les Robots
1950
- [BAI 06] Jean-Christophe Baillie, Antoine Robin
URBI
Tutorial for urbi 0.9 Basé sur URBI révision 142
- [BOJ 01] Hristo Bojinov, Arancha Casal and Tad Hogg,
Multiagents control of self-reconfigurable robots
Xerox Palo Alto Research Center, 2001.
- [BON00] Eric Bonabeau, Sylvain Guérin, Dominique Snyers, Pascale Kuntz, Guy Theraulaz,
Three-dimensional architectures grown by simple 'stigmergic' agents
BioSystems, No 56, pages 13–32, 2000.
- [BRE04] Brener N., Ben Amar F., Bidaud P.
Analysis Of Self Reconfigurable Modular Systems A Design Proposal For Multi Modes Locomotion
Proceedings of the IEEE International Conference Robotics & Automation (ICRA'04), New Orleans, LA. 2004.
- [BRO 86] Rodney A. Brooks,
A robust layered control system for a mobile robot
IEEE J. Rob. Autom. 2 pages 14-23, 1986.
- [BRO 91] Rodney A. Brooks,
Intelligence without reason
MIT, Artificial Intelligence Laboratory, A.I. memo No. 1293, 1991.
- [BUR 02] H-D Burkhard, D.Duhaut, M.Fujita, P Lima, R. Murphy and R. Rojas.

The road to RoboCup 2050.

IEEE Robotics and Automation Magazine vol 9, 2 pages 31-38, june 2002

- [CAP 20] Karel Čapek.
Rossum's Universal Robots.
1920.
- [CAR 05] Elian Carrillo, Dominique Duhaut,
Methods for collective displacement of modular self-reconfigurable robots.
Proceedings of the CLAWAR05 Conference, London, UK. 2005.
- [CHA 82] Rémy Chauvin,
Les sociétés animales
Quadrige / Presses Universitaires de France, 1982
- [CHR 04] David Johan Christensen
Metamodule Control for the ATRON Self-Reconfigurable Robotic System
Proceedings of the The 8th Conference on Intelligent Autonomous Systems (IAS-8),
Pages 685-692, Amsterdam 2004.
- [CHR 05] David Johan Christensen
HYDRA From Cellular Biology to Shape-Changing Artefacts
Lecture Notes in Computer Science. Springer Berlin / Heidelberg, September 2005.
- [CHR 97] Christiaan J.J. Paredis, et Pradeep K. Khosla.
« *RMMS : Projet modulaire de système de manipulateur de Reconfigurable,* »
Dans les démarches visuelles de la conférence 1997 internationale d'IEEE sur la
robotique et l'automatisme, Albuquerque, nanomètre, 19 au 24 avril 1997.
- [COR 03] Antoine Cornuéjols, Laurent Miclet,
Apprentissage artificiel, concepts et algorithmes
Eyrolles 2003
- [COU 02] Remi Coulom.
*Apprentissage par renforcement utilisant des réseaux de neurones, avec des
applications au contrôle moteur.*
Institut National Polytechnique de Grenoble. Juin 2002.
- [DRO 1993] Alexis Drogoul,
*De la simulation multi-agent à la résolution collective de problèmes. Une étude de
l'émergence de structures d'organisation dans les systèmes multi-agents*
Thèse de Doctorat de L'université Paris VI. 1993.
- [DUT 01] Alain Dutech, Olivier Buffet, François Charpillet,
Multi-Agent Systems by Incremental Gradient Reinforcement Learning.
IJCAI 2001: 833-838
- [END 03] Yoichiro Endo, Ronald C. Arkin.
*Anticipatory Robot Navigation by Simultaneously Localizing and Building a Cognitive
Map.*
Georgia Institute of Technology, Atlanta, GA, 30332-0280? 2003
- [FER 95] Jacques Ferber,
Les systèmes multi-agents : Vers une intelligence collective
InterEditions, 1995
- [FRE 02] Jakob Fredslund, Maja J Mataric

A General, Local Algorithm for Robot Formations

IEEE Transactions on Robotics and Automation,, special issue on Advances in Multi-Robot Systems, 18(5), Oct 2002,

- [GIO 05] C. Giovannangeli, P Gaussier, J.P. Banquet,
De la modélisation des cellules de lieux à la navigation autonome
5e Journées Nationales de la Recherche en Robotique, Guidel, Morbihan, 2005.
- [GOL 98] Goldberg D. E.
From Genetic and Evolutionary Optimization to the Design of Conceptual Machines,
Department of General Engineering, University of Illinois at Urbana-Champaign. 1998.
- [GON 04] Théopile Gonos.
Apprentissage de la locomotion par système de classeurs pour les atomes robotiques du projet MAAM.
Université Pierre & Marie Curie. Juin 2004
- [GRE 04] Brian P. Gerkey Maja J Mataric
A formal analysis and taxonomy of task allocation in multi-robot systems. September 2004.
- [GRO 06] Roderich Groß, Michael Bonani, Francesco Mondada, and Marco Dorigo
Autonomous Self-assembly in Swarm-bots
In IEEE Transactions on Robotics, volume 22(6), 2006.
- [GUE 04] Gueganno C., Duhaut D.
A hardware/software architecture for self reconfigurable robots,
Proceedings of the 7th international Symposium on Distributed Autonomous Robotics Systems (DARS 04).
- [HAG 94] Hagan, M. T., Menhaj M.
Training feedforward networks with the Marquardt algorithm
IEEE Transactions on Neural Networks, vol. 5, no. 6, pp. 989-993. 1994.
- [HAW 05] Hawkins Jeff, Sandra Blakeslee
On Intelligence
Editeur : Owl Books (NY). Édition : Reprint (août 2005)
- [JOD 94] Jean-François Jodouin,
les réseaux neuromimétiques
Hermes 1994
- [KON 84] Konrad Lorenz,
Les Fondements de l'éthologie
Champs Flammarion 1984, Paris
- [LAC 05] Simon Lacroix
Ballons dirigeables autonomes
Journées Nationales de la Recherche en Robotique, Guidel, 5-7 octobre 2005.
- [LEV 44] Levenberg, K.
A Method for the Solution of Certain Problems in Least Squares
Quart. Appl. Math. 2, 164-168, 1944.
- [LIV 03] Erwan Livolant
Apprentissage Multi-Agents par Systèmes de Classeurs

- [LUC 03] Philippe Lucidarme.
Apprentissage et adaptation pour des ensembles de robots réactifs coopérants
Université Montpellier II. Novembre 2003
- [LUC 04] Lucidarme P,
An evolutionary algorithm for multi-robot unsupervised learning
Proceedings of the CEC, Portland, Oregon. 2004.
- [MAM 05] S. Mammar¹, P. Martinet², S. Glaser³, M. Netto³, L. Nouvelière¹, B. Thuilot²
Méthodes de l'automatique pour l'assistance et l'automatisation de la conduite automobile,
Journées Nationales de la Recherche en Robotique, Guidel, 5-7 octobre 2005.
- [MAR 63] Marquardt, D.
An Algorithm for Least-Squares Estimation of Nonlinear Parameters
SIAM J. Appl. Math. 11, 431-441, 1963.
- [MAT 92] Maja J Mataric,
Integration of Representation Into Goal-Driven Behavior-Based Robots
IEEE Transactions on Robotics and Automation, 8(3), Jun 1992, 304-312
- [MAT 95] Maja J Mataric',
Issues and Approaches in the Design of Collective Autonomous Agents
Robotics and Autonomous Systems, 16(2-4), Dec 1995, 321-331
- [MIL 93] Milgram M.
Reconnaissance des formes : méthodes numériques et connexionnistes", A. Colin,
Université Pierre et Marie Curie (Paris 6), 1993.
- [MON 05] Montreuil V., Duhaut D., Drogoul A.
A collective moving algorithm in modular robotics: contribution of communication capacities
Proceedings of the 6th IEEE International Symposium on Computational Intelligence in Robotics and Automation, Espoo, Finland, June 27-30 2005.
- [MON 95] D. Montana,
Neural Network Weight Selection Using Genetic Algorithms,
Intelligent Hybrid Systems, S. Goonatilake and S. Khebbal (eds.). 1995.
- [MUN 02] Angélica Muñoz-Meléndez.
Une approche constructiviste de la conception de colonies de robots. Université Pierre et Marie Curie. 2002.
- [PUR 03] Purves, Augustine, Fitzpatrick, Katz, Lamantia, McNamara, Williams,
Neurosciences, Neurosciences & cognition
De Boeck Diffusion S.A. 2003.
- [RAJ 02] Rajarishi Sinha^{1,*}, Satyandra K. Gupta², Christiaan J.J. Paredis¹, Pradeep K. Khosla¹
Extracting Articulation Models from CAD Models of Parts with Curved Surfac
Journal of Mechanical Design, pp. 106-114, March 2002.
- [REM 82] Rémy Chauvin,
Les sociétés animales
Quadrige / Presses Universitaires de France, Septembre 1982.

- [SCH 01] Bruno SCHERRER,
Auto-organisation modulaire d'une architecture intelligente.
LORIA-INRIA, Campus Scientifique, Vandoeuvre-les-Nancy, France, 2001.
- [SEL 01] Philippe Sellem.
Répartition de la perception dans un système distribué de robots autonomes. Université Pierre et Marie Curie Paris 6. Septembre 2001.
- [SEM 02] François Sempé.
Auto-organisation d'une collectivité de robots : Application à l'activité de pratouille en présence de perturbations.
Université Pierre et Marie Curie. 2002.
- [SHE 04] Xi Shen,
Degree, weights and dynamics in complex networks
Intelligent systems and network group, department of electrical and electronic engineering Imperial College of London, 2004.
- [SHI 06] Michihiro Shimada, Takashi Minato, Shoji Itakura, and Hiroshi Ishiguro
Evaluation of Android Using Unconscious Recognition
Proceedings of the 2006 6th IEEE-RAS International Conference on Humanoid Robots, pp.157-162, December 4-6, 2006 Genoa, Italy
- [STO 04] Stoy K., Nagpal R.
Self-Reconfiguration Using Directed Growth
7th International Symposium on Distributed Autonomous Robotic Systems (DARS), France, June 23-25 2004.
- [TAK 03] Isao Takagawa', Kosuke Sekiyama and Toshio Fukuda.
Coevolution of Physical Configuration and Control Strategy on Self-Reconfigurable Flexible transfer System
Proceedings of the 2003 IEEE WRSJ InU. Conference on Intelligent Robots and Systems Las Vegas. Nevada ' October 2003.
- [TUR 50] A. M. Turing
Computing Machinery and Intelligence.
Mind 49: 433-460. 1950
- [VEL 05] Ramiro Velázquez¹, Jérôme Szewczyk¹, Moustapha Hafez² et Edwige Pissaloux¹
Une Interface Tactile Portable à Bas Coût pour le Handicap Visuel
Journées Nationales de la Recherche en Robotique, Guidel, 5-7 octobre 2005.
- [VIR 02] G S Virk
CLAWAR Robot component modularity,
Proceedings of the CLAWAR02 Conference, Paris 2002.
- [VIT 05] Vito Trianni and Marco Dorigo
Emergent Collective Decisions in a Swarm of Robots
IRIDIA, Technical Report No. TR/IRIDIA/2005-021. Université Libre de Bruxelles, Belgium, October 2005
- [YOS 01] Yoshida E., Murata S., Kamimura A., Tomita K., Kurokawa H. and Kokaji S.
A Motion Planning Method for a Self-Reconfigurable Modular Robot Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems.

Références Internet

- [altivis] <http://www.altivis.fr/-Les-robots-japonais-.html>
- [bois] <http://www.red3d.com/cwr/boids/>
- [cebot] <http://www.mein.nagoya-u.ac.jp/>
- [da vinci] <http://www.intuitivesurgical.com/index.aspx>
- [digital] <http://www2.parc.com/spl/projects/modrobots/>
- [honda] <http://world.honda.com/ASIMO/>
- [hrp2] <http://www.is.aist.go.jp/humanoid/index.html>
- [hydra] <http://hydra.mip.sdu.dk/home.html>
- [i-cubes] <http://www.cs.cmu.edu/~unsal/research/ices/cubes/>
- [ifr] <http://www.ifr.org/>
- [jnrr] <http://jnrr05.irisa.fr/> (Journées Nationales de la Recherche en Robotique 2005)
- [kokoro] <http://www.kokoro-dreams.co.jp/english/index.html>
- [libellule] <http://www.silmach.com/>
- [m-tran] <http://unit.aist.go.jp/is/dsysd/mtran3/>
- [maam] www-valoria.univ-ubs.fr/Dominique.Duhaut/maam/publications.htm
- [manta] <http://www-poleia.lip6.fr/~drogoul/projects/fr/index.html>
- [polybot] <http://www2.parc.com/spl/projects/modrobots/>
- [proteo] <http://www2.parc.com/spl/projects/modrobots/>
- [robocup] <http://www.robocup.org/>
- [sony] <http://www.sony.net/SonyInfo>
<http://www.csl.sony.fr/index.html> (Sony Computer Science Laboratory Paris)
- [swarm] <http://www.swarm-bots.org/index>
- [toyota] <http://www.hybridsynergydrive.com/fr/future.html>
- [unece] http://www.unece.org/press/pr2004/04robots_index.htm
- [urbi] <http://www.urbiforge.com>

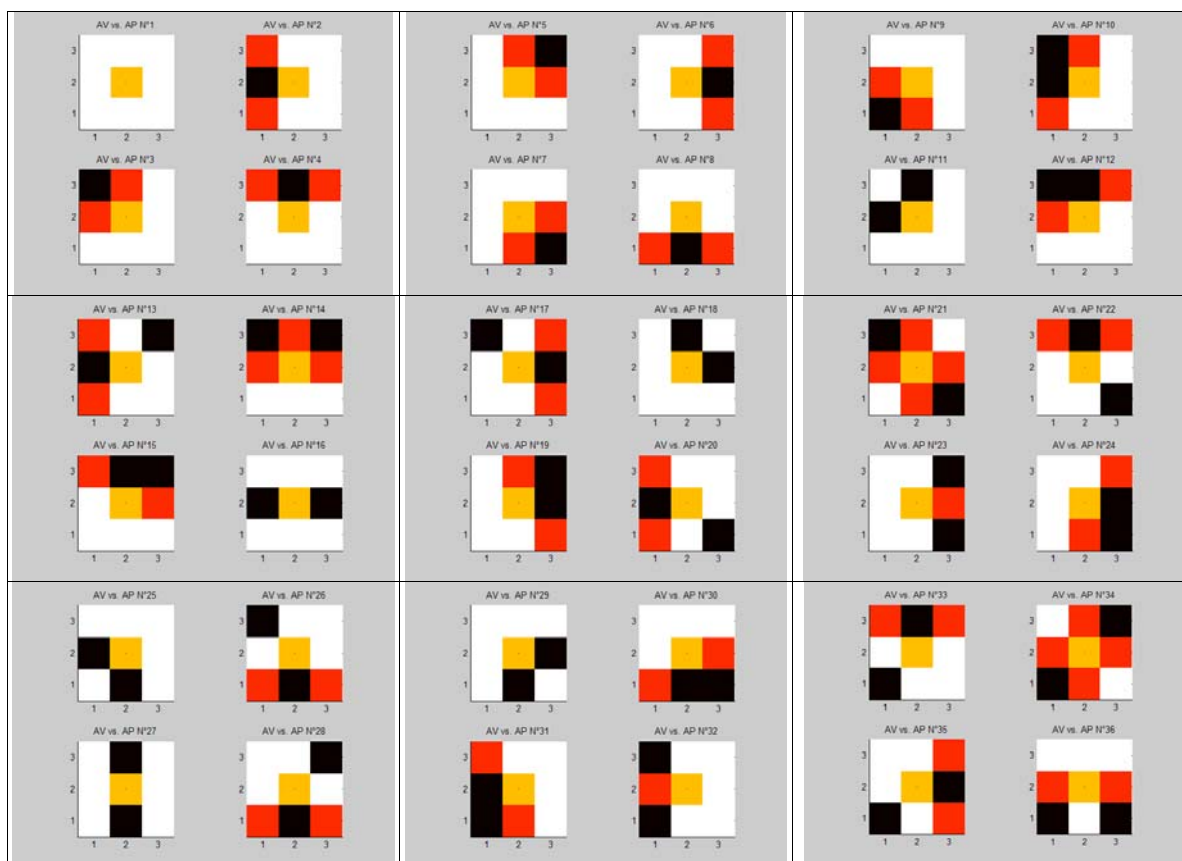
Annexe 1.

A.1 Les actions possibles dans le voisinages d'un atome

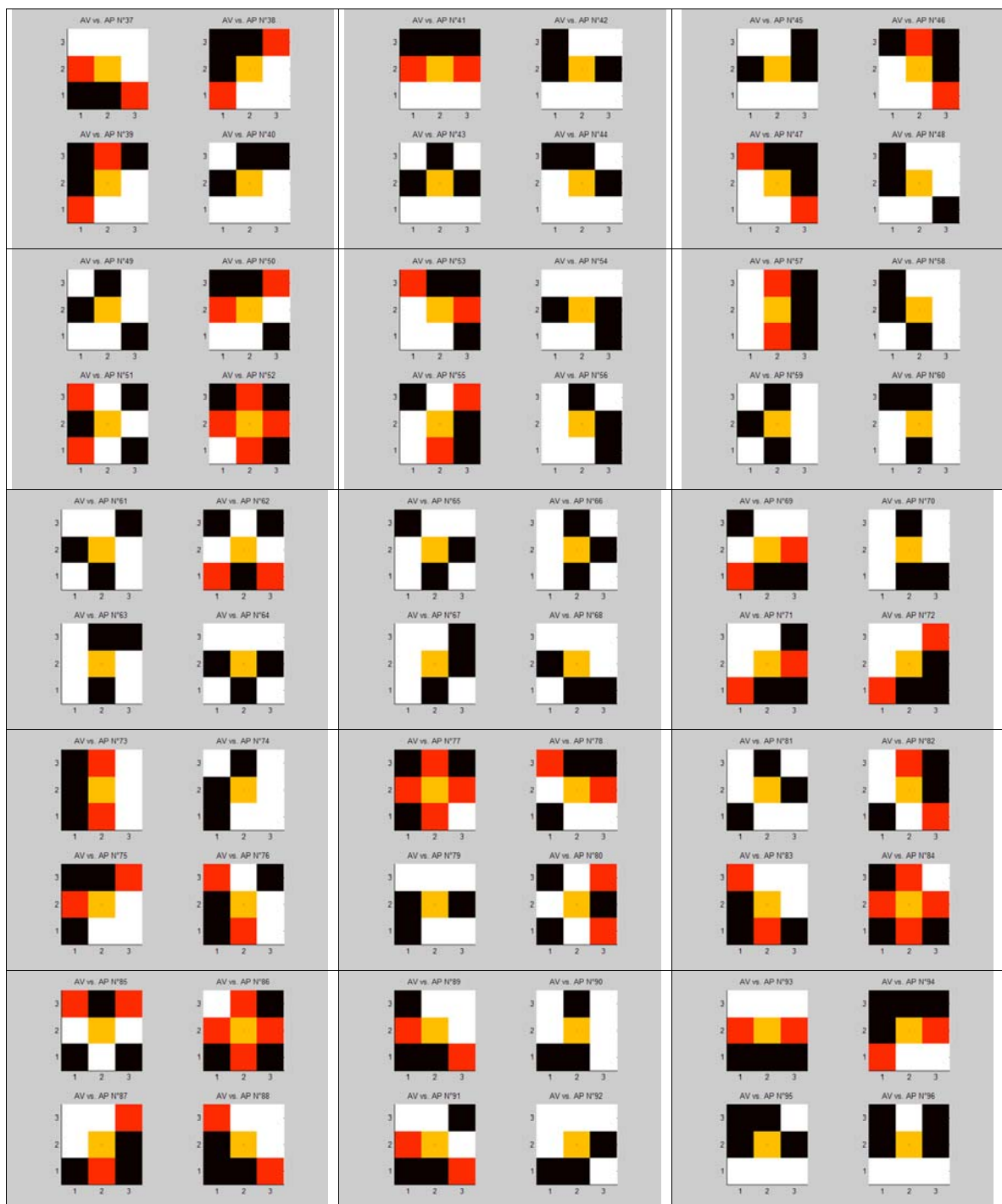
Cette annexe comporte les 256 combinaisons que peuvent prendre les atomes dans le voisinage. Il montre aussi les actions autorisées qui ne viennent pas à l'encontre des prémisses de connexions à d'autres modules que dans les 4 connectivités de Von Newman.

Il est certain qu'un nombre d'exemples se reproduit sur ses places symétriques, mais le nombre d'exemples changement en fonction du nombre d'atomes dans le voisinage. Les illustrations montrent l'évolution du cas avec un atome sans aucun autre dans son voisinages, passant par 1, 2, 3... et 8 possibles atomes voisins.

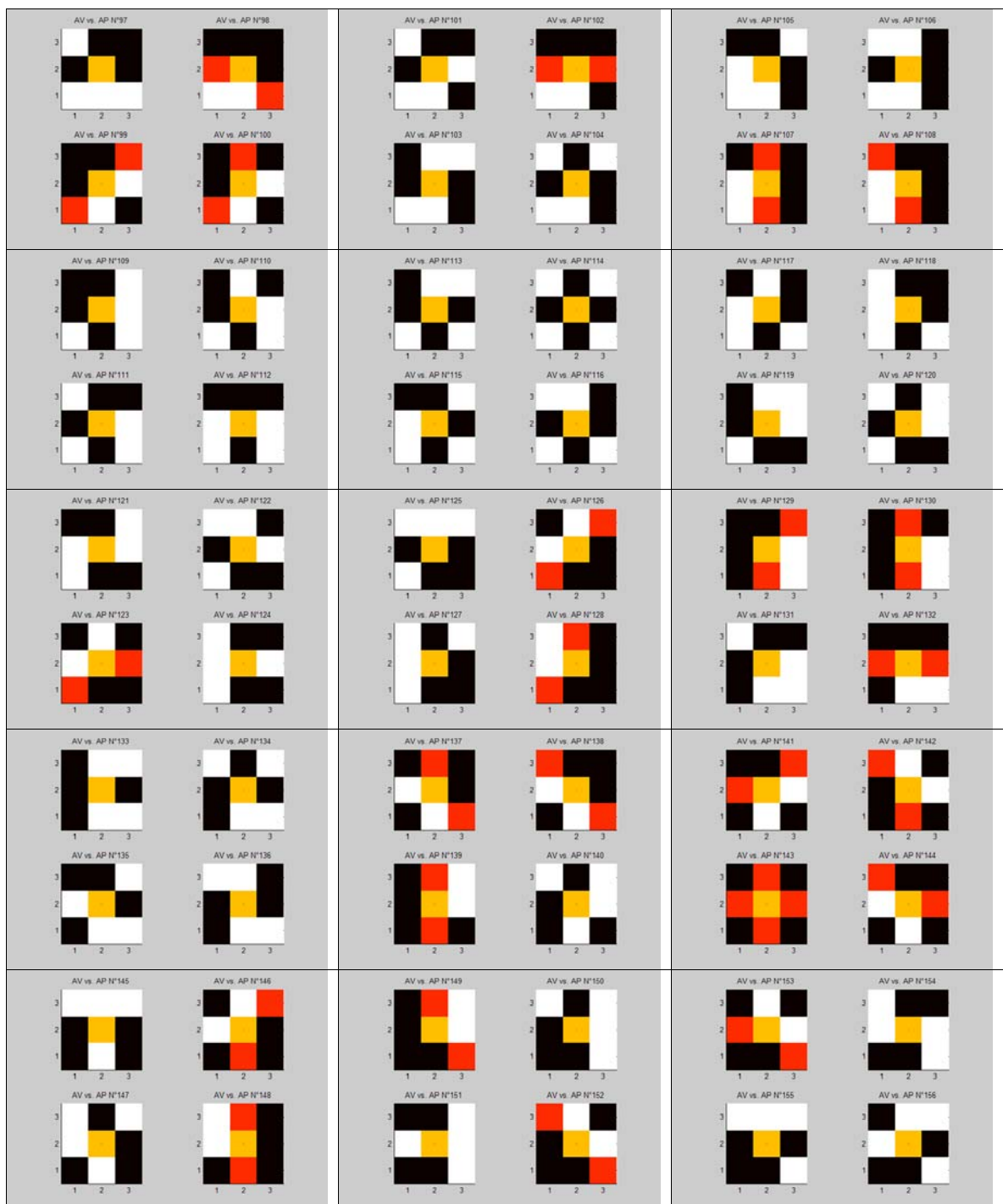
L'atome de référence est au centre en jaune, en noir les atomes du le voisinage et en rouge les actions autorisées. Le schéma ci-dessous montre la représentation des combinaison qu'un **Atomes** peut retrouver dans son **Voisinage** versus les **Actions** qui lui sont **Possibles**, de la 0 à la 36.



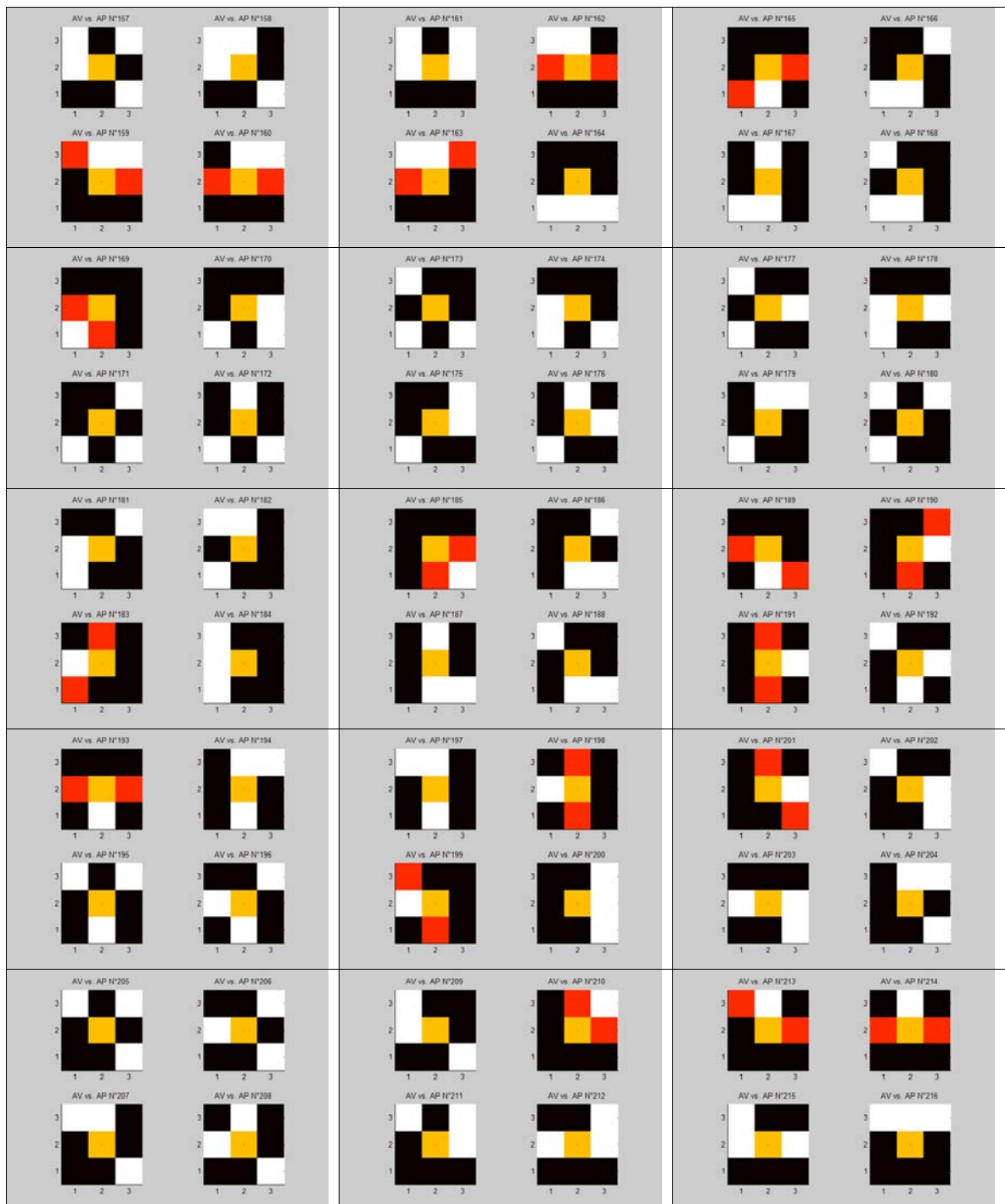
Les combinaison d'Atomes dans le Voisinage vs. les Actions qui lui sont Possibles, de la 37 au 96.



Les combinaison d'Atomes dans le Voisinage vs. les Actions qui lui sont Possibles, du 97 au 156.



Les combinaison d'Atomes dans le Voisinage vs. les Actions qui lui sont Possibles, du 157 au 216.



Les combinaison d'Atomes dans le Voisinage vs. les Actions qui lui sont Possibles, du 217 au 256.

