

Conception évolutionniste de réseaux de neurones pour le contrôle de robots mobiles

Conference Paper

Author(s):

Mondada, Francesco; Floreano, Dario

Publication date:

1994

Permanent link:

<https://doi.org/10.3929/ethz-a-010111555>

Rights / license:

[In Copyright - Non-Commercial Use Permitted](#)

Conception évolutionniste de réseaux de neurones pour le contrôle de robots mobiles

Francesco Mondada et Dario Floreano
Laboratoire de microinformatique
Ecole polytechnique fédérale de Lausanne

Introduction

L'approche classique de conception est essentiellement basée sur la compréhension du système à gérer, sa modélisation à l'aide de techniques plus ou moins classiques et le dimensionnement de ses composants selon les règles de l'art de l'ingénieur. Cette approche comporte, entre autres, le découpage du système en blocs fonctionnels ou structurels auxquels on attribue des entrées et des sorties qui ont des caractéristiques connues. De plus, dans la plupart des cas, l'ingénieur s'arrange pour que ces caractéristiques soient linéaires ou avec des non-linéarités simples et bien conformes à sa façon de penser.

Cette méthode de conception s'est montrée très efficace dans beaucoup de domaines de complexité limitée. Les essais dans la direction d'une plus grande complexité, comme l'a fait l'intelligence artificielle classique, ont montré les limitations de cette approche. Les systèmes de contrôle de robots mobiles en sont un exemple frappant: dans un monde réel difficilement modélisable la modularisation du système devient difficile, les symboles internes deviennent moins sûrs, les règles de gestion très complexes et les failles du système plus vulnérables. Les méthodes classiques posent donc des problèmes de base. Des approches corrigées ont été proposées par Brooks [Brooks 1991], qui essaie d'éliminer les représentations internes de trop haut niveau. Malgré cela son approche reste liée à une modélisation du système et à son interaction avec l'environnement. Une autre direction qui a rencontré un succès surtout commercial est celle de la logique floue [Zimmermann 1990], qui vise à rendre moins rigide la symbolique du système et permet l'accès à quelques problèmes non-linéaires. Cette approche ne reste toutefois applicable que dans des cas très simples. Les réseaux de neurones ont aussi fait leur apparition dans le domaine, mais sont souvent utilisés comme un module d'un système classique [Berns et al. 1991]. Avec leur introduction on vise à rendre plus robuste la totalité de la construction, mais la structure rigide de l'ensemble limite beaucoup le succès de cette approche.

Or la nature, dans ses exemples de systèmes autonomes bien réussis que sont les animaux, nous montre des mécanismes dans lesquels les structures se mélangent, les fonctionnalités sont réparties, les représentations internes minimalistes sont bien adaptées à la tâche visée et le résultat émerge de l'interaction continue de processus simples qui effectuent des traitements parallèles. Deneubourg lors de ses études sur les fourmis et ses simulations [Deneubourg et al. 1991] et Gaussier lors d'expérimentations similaires avec des robots réels [Gaussier et al. 1994] montrent, par exemple, que dans des groupes de fourmis ou de robots très peu intelligents peuvent apparaître des comportements collectifs qui semblent être intelligents et comporter des mécanismes en réalité inexistantes. Dans l'expérience de ramassage d'objets faite par Gaussier, l'échange d'objets entre robots émerge d'un mécanisme d'interaction simple, alors qu'il aurait demandé un effort important de développement chez un ingénieur classique. Le même phénomène est illustré par Braitenberg [Braitenberg 1984] en ce qui concerne le mécanisme "neuronal" de contrôle de ses véhicules.

Malheureusement la maîtrise de cette émergence est rarement à la portée des techniques de conception classiques. Dans cet article nous montrons comment on peut aller dans cette direction à l'aide des algorithmes génétiques appliqués à des réseaux de neurones pour le contrôle d'un robot mobile réel.

L'outil d'expérimentation, Khepera

Khepera (Figure 1a) est un robot mobile développé dans notre laboratoire comme outil d'expérimentation [Mondada et al. 1993]. Sa taille (55 mm de diamètre et 60 mm de hauteur) en font une base d'expérimentation très maniable. Malgré cette caractéristique physique, le robot a les fonctionnalités d'un robot mobile de taille normale: il dispose en effet d'un processeur 32 bits avec une horloge à 16MHz, de 256 kbytes de RAM et 512 kbytes de ROM.

a.

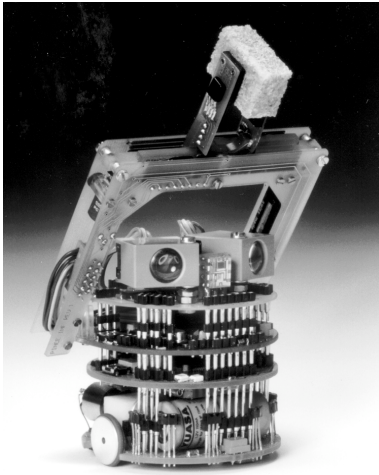


Photo A. Herzog

b.

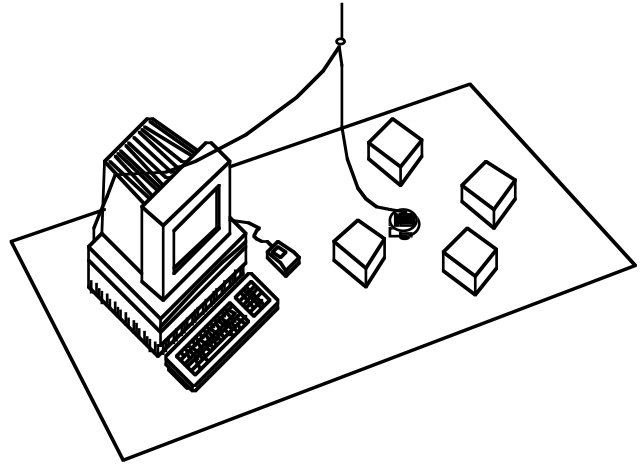


Figure 1. Le robot mobile miniature Khepera (a) et l'environnement de développement.

Le robot est matériellement modulaire. Aux modules de base, comportant l'étage avec les roues et les capteurs de proximité et l'étage comprenant l'unité de calcul, on peut en effet ajouter des modules permettant la préhension ou une vision simplifiée, comme le montre la figure 1a. Dans les expériences présentées dans cet article nous avons utilisé uniquement les modules de base. Cette configuration comporte le processeur principal, la mémoire, les interfaces pour la liaison avec un ordinateur hôte, les deux roues, les accumulateurs pour une activité indépendante et huit capteurs de proximité, six à l'avant et deux à l'arrière. Ces capteurs permettent de mesurer, avec une assez mauvaise précision, la distance d'objets environnants le robot. Les obstacles de couleur claire sont perçus à une distance maximale d'environ 50 mm.

La taille de Khepera permet surtout de faire des expérimentations dans une configuration agréable: on peut en effet placer l'environnement du robot sur une table à côté d'une station de travail et piloter le robot depuis n'importe quel programme (Figure 1b). Ceci permet d'utiliser au mieux les possibilités graphiques et de développement logiciel de la station de travail tout en travaillant avec un robot mobile réel.

Les algorithmes génétiques

Les algorithmes génétiques sont un outil d'optimisation d'inspiration biologique proposés par Holland [Holland 1975]. Leur fonctionnement est basé sur des mécanismes d'évolution. En effet on applique normalement l'algorithme à une population d'individus qui évolue le long de plusieurs générations. Chaque individu est caractérisé par un génotype, qui définit la structure de l'individu ou une partie de celle-ci. Au début des opérations, une population de génotypes est générée aléatoirement (Figure 2a). Chaque génotype permet de constituer un individu. On parle alors de population de la première génération. Lors du déroulement de l'algorithme on laisse vivre chaque individu de la population et on évalue ses performances selon un critère prédéfini. Cette évaluation donne lieu à une note de qualité (Figure 2b). Une fois que tous les individus de la population ont été évalués, on les reproduit avec une probabilité de reproduction qui est fonction de leur note. Ceci donne lieu à une nouvelle population (Figure 2c), normalement de même taille que la précédente. Sur cette population on applique des croisements: des paires de génotypes sont choisis au hasard, sont coupés à un emplacement aléatoire et recombinés entre eux (Figure 2d). Des mutations peuvent aussi être appliquées: une partie d'un génotype choisi au hasard est modifiée

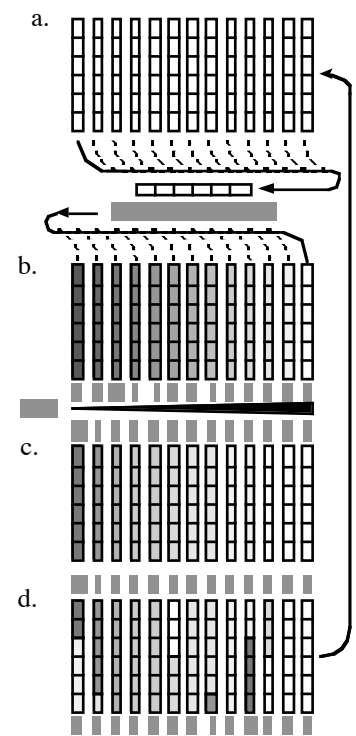


Figure 2. Opérations d'un algorithme génétique.

aléatoirement. Une fois ces opérations effectuées on obtient une nouvelle génération qui sera à son tour évaluée, reproduite, etc.

Les algorithmes génétiques sont particulièrement bien adaptés à des problèmes où l'espace de recherche est grand et contient de nombreux minima locaux. Ils demandent par contre un bon codage des paramètres dans le génotype et une fonction d'évaluation efficace.

Evolution d'une boucle sensori-motrice simple

Malgré sa simplicité, le comportement d'évitement d'obstacles reste très intéressant lorsqu'on aborde une nouvelle approche. Il existe en effet une solution à ce problème dans beaucoup de techniques de contrôle de robots mobiles, ce qui permet de comparer les différents aspects de la solution.

La solution de référence à ce problème est pour nous le véhicule 4 de Braitenberg (Figure 3a). Cette structure à deux neurones et à poids fixes résout très élégamment le problème et nécessite un temps de dimensionnement très limitée. Le seul problème de ce système réside dans le fait que le véhicule, lors de son fonctionnement, peut se trouver dans une situation symétrique de laquelle il n'arrive pas à sortir.

Le réseau que nous avons évolué par les algorithmes génétiques est présenté dans la figure 3b. Sa structure est similaire à celle du véhicule de Braitenberg, sauf en ce qui concerne les rebouclement sur les deux unités sommatrices et leur fonction de sortie. Dans le génotype nous avons codé simplement les poids du réseau.

L'évaluation de la population de réseaux a été faite sur un robot réel placé dans un environnement fixe. Chaque individu a été testé pendant environ 30 secondes et son évaluation a été faite selon la formule suivante, calculée toutes les 300ms et moyennée sur la durée de vie de l'individu:

$$\square = \bar{v} \left(1 - \sqrt{\square V} \right) \left(1 - A_{max} \right)$$

$$\begin{matrix} 0 & \square & \bar{v} & \square & 1 \\ 0 & \square & \square & V & \square & 1 \\ 0 & \square & A_{max} & \square & 1 \end{matrix}$$

Dans laquelle \square est la note d'évaluation, \bar{v} la vitesse moyenne absolue des deux roues, $\square V$ la différence en valeur absolue entre les vitesses des deux roues et A_{max} la valeur maximale des capteurs de proximité. A noter que ces capteurs fournissent un signal qui augmente exponentiellement avec le rapprochement des objets. Cette fonction d'évaluation pousse l'évolution à primer des robots qui vont vite (\bar{v}) qui ne tournent pas trop sur eux-mêmes ($1 - \sqrt{\square V}$) et qui évitent les obstacles ($1 - A_{max}$).

Résultats et commentaires

L'évolution du robot dans son environnement a été répétée plusieurs fois avec les mêmes résultats. Le comportement d'évitement d'obstacles a été obtenu toujours autour de la cinquantième génération, c'est à dire après l'évaluation d'environ 5000 individus (100 par génération), ce qui correspond à une durée de deux à trois jours. Les résultats plus intéressants concernent l'émergence de certains comportements qui permettent au réseau de neurones d'exploiter au mieux les caractéristiques du robot, de ses capteurs et actuateurs.

Bien que la fonction d'évaluation soit parfaitement symétrique par rapport à la direction d'avancement, dans toutes les expériences que nous avons effectuées, l'évolution a conduit à des individus qui vont en avant, c'est à dire dans la direction qui correspond au plus grand nombre de capteurs montés le robot (vers le haut dans la figure 3). Ce comportement émerge de l'interaction du robot avec son environnement et permet au réseau d'avoir une meilleure perception de

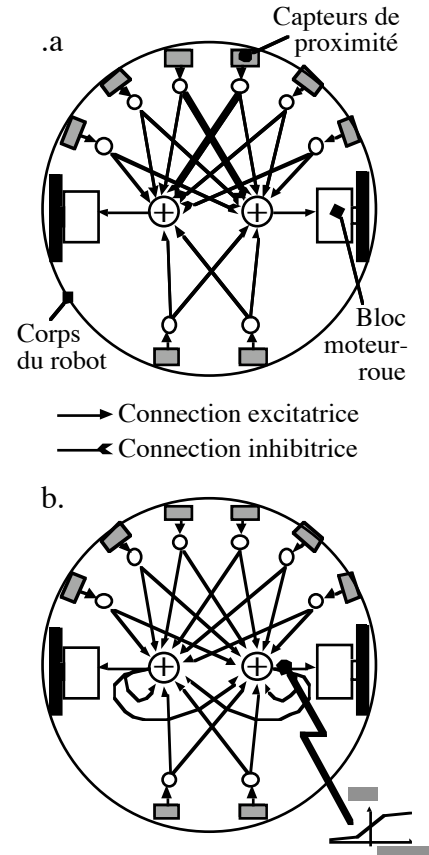


Figure 3. Structure du réseau de Braitenberg (a) et du réseau évolué (b).

l'environnement. Ce choix, qui est évident, montre bien l'aptitude des algorithmes génétiques à exploiter les caractéristiques du système.

Un autre choix peut-être moins évident consiste dans la vitesse d'avancement. Les meilleurs robots en fin d'évolution avancent à une vitesse qui n'est pas la plus grande qui leur est permise. Cette vitesse est en effet fonction de la distance à laquelle les robots arrivent à percevoir les obstacles. Si on évolue le robot dans un environnement composé d'obstacles plus sombres (donc moins visibles pour le robot), la vitesse des meilleurs robots est nettement réduite. Ceci permet au robot de réagir en temps utile lors de la perception d'un obstacle. Ce choix émerge donc aussi de l'interaction du robot avec son environnement et tend à optimiser l'efficacité du système.

Enfin, contrairement au véhicule de Braitenberg, les robots évolués ne sont pas bloqués par des situations symétriques. Ces situations étaient présentes dans l'environnement dans lequel ils ont évolués. La structure de contrôle a donc développé une technique pour ne pas avoir de point de symétrie. Ceci a été possible grâce au rebouclage sur les unités sommatrices, correctement ajustées pour que toute situation stable soit rendue instable. Cette solution, extrêmement simple mais efficace, montre aussi l'exploitation optimale de la structure faite par l'algorithme génétique.

Conclusion

L'expérience présentée dans ce document montre que les algorithmes génétiques permettent de faire émerger des comportements de l'interaction du robot avec son environnement, afin d'exploiter au mieux les caractéristiques de cette interaction et la structure de contrôle. Cette propriété est très intéressante et complémentaire aux techniques de conception classiques. Les algorithmes génétiques pourraient donc être un outil important d'aide à la conception.

Le codage des poids du réseau de neurones dans le génotype est une approche très simpliste et qui ne peut pas être étendue vers des tâches beaucoup plus complexes. Dans nos travaux futurs nous essayerons de combiner cette technique avec d'autres méthodes, comme l'apprentissage ou l'inspiration biologique. Dans la mesure du possible l'évolution sera menée sur une vraie population de robots.

Remerciements

Nous aimerions remercier Edoardo Franzi et André Guignard pour la réalisation de Khepera ainsi que le Fond National Suisse de Recherche qui a supporté ce travail dans le cadre du projet PNR23.

Bibliographie

- Berns, K., Dillmann, R. & Hofstetter, R. (1991). An Application of a Backpropagation Network for the Control of a Tracking Behavior. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, California, April, pages 2426-31.
- Braitenberg, V. (1984). *Vehicles: Experiments in Synthetic Psychology*. Cambridge, MIT Press.
- Brooks, R.A. (1991). Intelligence without representation. *Artificial Intelligence*, 47, pages 139-159.
- Deneubourg, J.C., Goss, S., Franks, N., Sendova, A., Franks, A., Detrin, C. & Chatier, L. (1991). The dynamics of collective sorting: Robot-like ant and ant-like robot. In *Simulation of Adaptive Behavior: From Animals to Animats*, J.A. Mayer and S.W. Wilson editors, pages 356-365. MIT Press.
- Holland, J.H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: The University of Michigan Press.
- Mondada, F., Franzi, E., & Jenne, P. (1993). Mobile robot miniaturisation: A tool for investigation in control algorithms. To appear in *Proceedings of the Third International Symposium on Experimental Robotics*, Kyoto, Japan, Oct. 28-30.
- Zimmermann, H.-J. (1990). *Fuzzy Sets Theory - and Its Applications*. Kluwer Academic Publishers.
- Gaussier, Ph. & Zrehen, S. (1994). A Constructivist Approach for Autonomous Agents. To appear in *Artificial Life in Virtual Realit*, Thalmann N. & D. editors, Addison Wesley.