



HAL
open science

Adaptive Learning Control in Evolutionary Design of Mobile Robots,

Patrick Henaff, Olivier Chocron

► **To cite this version:**

Patrick Henaff, Olivier Chocron. Adaptive Learning Control in Evolutionary Design of Mobile Robots,. IEEE International Conference on Systems, Man and Cybernetics, Oct 2002, Hammamet, Tunisia. pp.345-349, <10.1109/ICSMC.2002.1176065>. <hal-00523329>

HAL Id: hal-00523329

<https://hal.science/hal-00523329v1>

Submitted on 21 Nov 2024

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.



HAL Authorization

Adaptive Learning Control in Evolutionary Design of Mobile Robots

P. Hénaff - O. Chocron

Laboratoire d'Instrumentation et de Relation Individu Système
CNRS - Université Versailles
10-12 Avenue de l'Europe, 78140, Vélizy, FRANCE
henaфф@robot.uvsq.fr

Abstract— This is a study of an application of neural technics to the learning of control laws within the framework of the evolutionary design of robotics systems. The present paper proposes the replacement of the evolutionary synthesis of the individual's control law by its learning. The learning of neural controller is carried out on-line when the robot undergoes evaluation tests. Thus, a robot that is a priori inadequate to solve a task can, thanks to the training it goes through, improve its performance. It participates then to the global improvement of the population while it would have been eliminated without learning. A mobile robot that could be equipped with up to 4 independent driving wheels and that must attain a given configuration will be taken as an example. The whole unit uses a simulation of the robot and its environment in which all dynamic effects are taken into account. Results show the accuracy and strength of the method since even the structures which would have been in fact eliminated to carry out this kind of task, are controlled with reasonable efficiency.

Keywords— Evolving, Adaptive Neural Control, adaptive behaviour, Mobile Robot.

I. EVOLVING SYNTHESIS OF ROBOTS

The use of genetic algorithms allows the generation of robotic mechanisms (mechanical structure and actuators) to carry on a specific task : to reach point or a configuration in the workspace, climb stairs... Thus, when a displacement task is defined for a mobile robot (reach a point with an given orientation), artificial evolution can converge into a platform equipped with wheels and/or legs. In the case of evolutionary robotics every individual robot (phenotype) is characterized by its genotype. The genotype usually gathers all the information about the robot morphology (number of legs, degrees of freedom...), electro-mechanical features, type of joints (revolute, linear) or actuators. The individual robot so made is assessed by a test it undergoes, and its fitness is used as a parameters in the population evolution. An evolutionary algorithm achieves this evolution by applying genetic operators (reproduction, mutation, crossover) on the robots topology and on the control law of its actuators.

A. Limits of the global approach

The aim of a global evolutionary algorithm (GEA) is to find the robot's topology and its associated control. The control law of each joint L_j can be represented by

a primitive law such as a temporal circular periodical function : $U_j = \alpha_j * \cos(\omega_j * t + \phi_j)$ where α_j is its amplitude, ω_j its pulsation, ϕ_j its phase and t the simulation time. The resulting genotype is constituted of the Incidence Matrix MI of the robot topology and of the three float vectors (α, ω, ϕ) for the control (see 1 and [2] for more details). This mixed genotype must, through

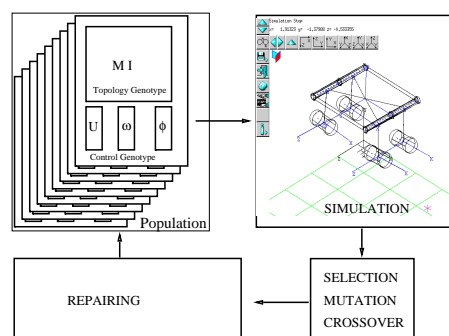


Fig. 1. Global Evolutionary Algorithm

assessment, show its capacities at fulfilling of the given locomotion task. To illustrate this approach, we have chosen to evolve rovers constituted by a rigid platform and a variable number of independent driving wheels (from 0 to 4 wheels). The evaluation of the rovers is performed through a dynamic simulation accounting for inertial, control and friction forces on a flat ground. For simulations, mechanical features of the robot are :

- size of the platform: 1.2mx0.7 m
- mass of the platform : 20 kg
- mass of one wheel: 1 kg
- maximum robot velocity 12.5 m/s

We show the limits of this global approach through a simple locomotion task that is to reach a position in front of the robot (Fig.2). When observing the behavior of the robot, one can notice that the front wheel produces an important force while the 2 back wheels regulate the friction to control the trajectory. The GEA suggests thus a solution for control which consists in exercising an important traction with the front wheel (locates on the opposite side of the goal so as to drive the robot towards this goal), and with the 2 back wheels applying only breaking forces to regulate the robot's trajectory. Indeed, the control law works in open loop and

receive no feedback from the robot, the environment or even the task to be carried out.

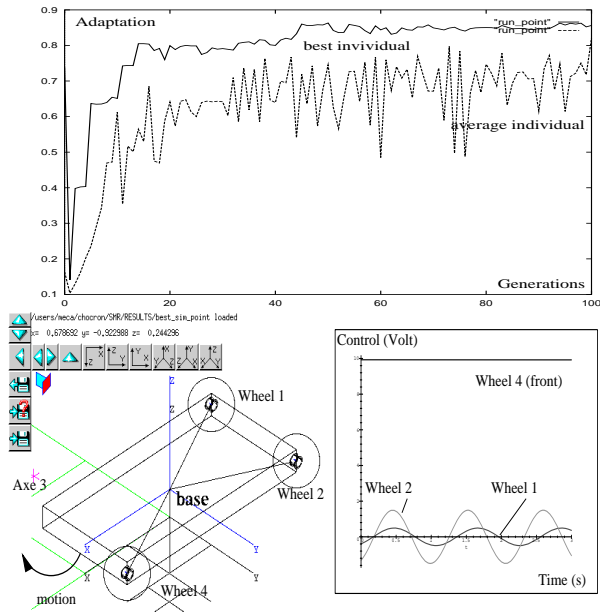


Fig. 2. Example of evolution results to reach a position (50 individuals/generation): fitness curves (Top), the robotic solution and its control (bottom).

II. EVOLUTION AND LEARNING

The above results show the limits of the global method for evolutionary synthesis of a control law. Furthermore, a robot individual even if disabled with regard to its task, must be able to learn on line how to improve its performance in order to carry out the task assigned to it. So, we propose to give each created individual a capacity of learning based on the use of a neural controller which will learn how to control the robot. When following this principle, the genotype, which does not code the robot's control law anymore, becomes more simple and artificial evolution simulation time is greatly reduces. The integration of a neural control structure through learning allows for increasing of every robot's potentiality, and the individual who will be able to learn well then have every chance of participating in a progressive evolution of the population even if it is initially poorly adapted.

How interesting it is to use ANN for mobile robots has already been shown at different levels of control architecture [1][4][6][7][5][9]. The use of ANN in evolutionary synthesis of robots can be seen in 2 ways. First way, to consider neural control as an integrated part of the genotype in the evolutionary process [7][5] which determine its parameters in relation to the performances reached by the robot. This technique is very efficient even if the algorithm convergence time can become too important for a embedded application. It allows, for instance, the generation of gait for moving an hexapod robot and the acquisition of a reflex behaviour to avoid

obstacles. Second way, to consider the neural controller as a fixed structure during evolution. The individual selected for evolution, will have, during its "life time", its neural controller learning how to control better and better the robot in a succession of trial an error test.

A. Learning during assessment

These two methods are complementary and nothing excludes their simultaneous application on the topology of the controller during the population evolution, then by the controller' learning during the individual's evaluation. In this case, computation time can be prohibitive in obtaining a correct robot. That is why, following the second way, our aim is to give every newly created robot an opportunity to adapt its behaviour to its structure with an on-line learning process. Its learning process is carried on during the evolution (Fig. 3), in which a period is dedicated to learning. This period means that each individual is given a chance to learn its task before it is passed to the evaluation process. Thus each robot goes through an life cycle: birth (robot creation), learning stage and a reproduction phase (through a process of genetic operators). The learning phase is a test that a robot undergoes and that allows it to learn how to carry out its task. It's only after this phase that the robot will be assessed.

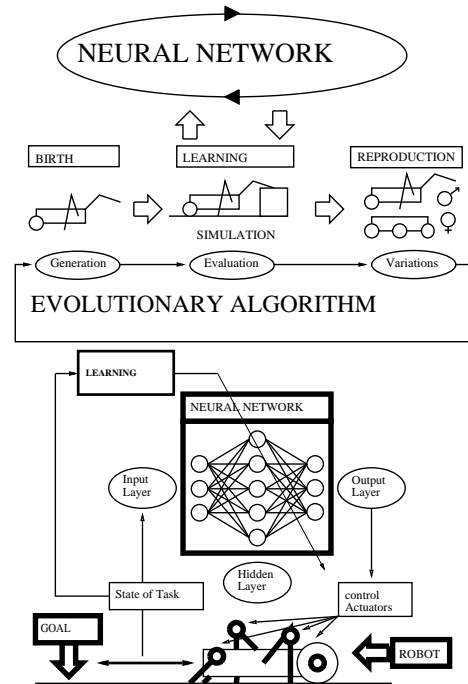


Fig. 3. Evolution scheme of a population with individual learning: global evolution and learning control

B. Learning algorithm

The neural network is used as a self-adaptive controller based on the minimization of a cost function (back-propagation algorithm) which uses an analytical

model of the robot dynamics and numerical data issued from the simulation. The input of neural controller is constituted by the cartesian and angular errors of the robot between its current position (at step k) of its center of gravity (X_G^k, Y_G^k, Θ_G^k) and a desired one (X_d^k, Y_d^k, Θ_d^k). Then, the goal of the learning is to compute on the net output the torque vector ($C_1^k, C_2^k, C_3^k, C_4^k$) to apply on wheels. In the present case, the cost function can be:

$$J^{k+1} = \alpha_1 [X_d - X_G^{k+1}]^2 + \alpha_2 [Y_d - Y_G^{k+1}]^2 + \alpha_3 [\theta_d - \theta_G^{k+1}]^2 \quad (1)$$

In order to obtain a global minimization of this fitness, we backpropagate at each step k on each output C_i the following gradient:

$$\frac{\partial J^{k+1}}{\partial C_i^k} = 2\alpha_1 [X_d - X_G^{k+1}] \frac{\partial X_G^{k+1}}{\partial C_i^k} + 2\alpha_2 [Y_d - Y_G^{k+1}] \frac{\partial Y_G^{k+1}}{\partial C_i^k} + 2\alpha_3 [\theta_d - \theta_G^{k+1}] \frac{\partial \theta_G^{k+1}}{\partial C_i^k}$$

Calculation of this gradient can be obtained using a dynamic model (equations 2, under hypothesis of rolling without sliding) and by a second order linearization (step time Δ):

$$\begin{cases} X_G^{k+1} = X_G^k + \Delta \dot{X}_G^k + \frac{\Delta^2}{2} \ddot{X}_G^k \\ Y_G^{k+1} = Y_G^k + \Delta \dot{Y}_G^k + \frac{\Delta^2}{2} \ddot{Y}_G^k \\ \theta_G^{k+1} = \theta_G^k + \Delta \dot{\theta}_G^k + \frac{\Delta^2}{2} \ddot{\theta}_G^k \end{cases}$$

$$\text{where } \begin{cases} \dot{X}_G^k = \frac{1}{rM} \sum_{i=0}^4 C_i^k \cos \theta_G^k \\ \dot{Y}_G^k = \frac{1}{rM} \sum_{i=0}^4 C_i^k \sin \theta_G^k \\ \ddot{\theta}_G^k = \frac{1}{I_z} \frac{R}{2r} \{C_1^k - C_2^k + C_3^k - C_4^k\} \end{cases} \quad (2)$$

M is the total mass of robot, r the radius of its wheel, R the half of axle length and I_z the inertial moment on z axis (Fig 2). This dynamic behaviour model can be considered as degraded since, during simulations and due to the control strategy imposed on the robot by neural control, the non-sliding rolling hypothesis will not be respected: the robot will skid while moving. However, the different learning experiments performed at the laboratory on fast and heavy real robots [3] have shown that this approximation has no effects on the precision and on the stability of the control loop because the learning is applied on line during the control.

C. Example of results of learning

This part shows few results, for a more complex locomotion task than in section I that is to reach a cartesian and an angular position on a flat ground. Each robot goes through 4 time limited (30 seconds) simulation phases for its assessment: 3 simulations with learning of the neural controller (“the robot is learning”), and one simulation without learning (the robot is tested). At the beginning of the first phase, the neural controller is initialised with random weights and has thus no a-priori knowledge of the task and on behavior

of the robot. During the next learning phases, neural controller which has learned during the previous phase increases its knowledge of the control and of the behavior of the robot. Then the robot is assessed in terms of the performance obtained during the test phase. As an illustration, a task consisting for the robot, in going from position $(0m, 0m, 0^\circ)$ to a desired configuration $(3m, -2m, 0^\circ)$ is suggested here. To move, it uses independent driving wheels that can be located only at the four corners of its platform. The evolution algorithm must thus find what the best structure is, that is to say, what the number of wheels and their position are. As for neural control, it has to find what the best control law is. The following curves present the results of learning and test on different types of robots in terms of cartesian trajectories, orientation and cost function (fitness).

C.1 case of a one-wheeled robot

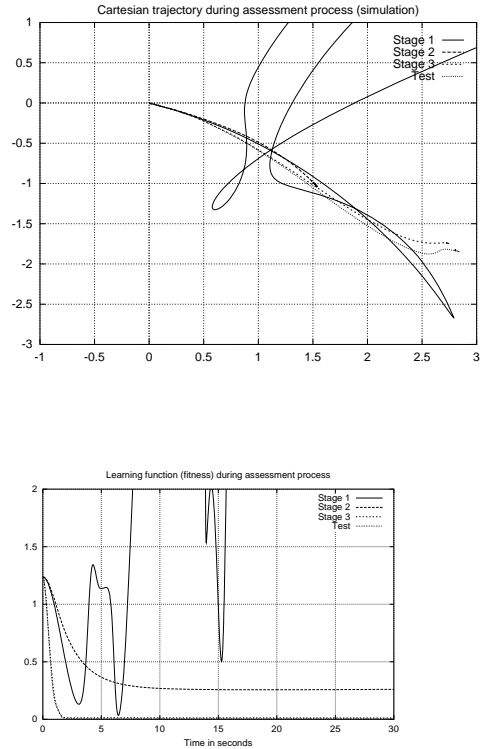


Fig. 4. Learning process for a one-wheeled robot: cartesian trajectory (top) and learning function (bottom)

The case of a one-wheeled robot is the most unfavourable configuration as the friction of the platform against the ground is important and there’s no symmetry in the kinematics of the robot. The orientation gets uncontrollable except by imposing sufficient acceleration to make the robot skid around the contact point. It can be seen Fig 4 that during the first learning stage, control becomes unstable : the robot goes away from the goal by turning on itself (20 meters at the end of stage).

Despite this failure, the second stage allows the robot to get close to the objective with an error around 1.5 m for position and 40 deg for orientation. These errors decrease considerably at the end of the last stage. The neural network has thus been able to identify a control strategy that allows it to take into account the bad behavior of the robot. However, for the one-wheeled the learning convergence of the robot can be obtained only when the objective is located at the opposite side of the wheel. Then, among the 4 one-wheeled robot individuals, only one reaches the goal with satisfactory error: when the robot has its wheel located on the back left side.

C.2 Case of a 2-wheeled robot at the front or at the back

For the 2-wheeled robot at the front or at the back, the inertia momentum is maximum and its platform is in contact with the ground at the opposite side of the wheels. This disturbs particularly the first learning stage for which a great oscillation of the trajectory can be noticed (see Fig. 5): the robot skids to right then to the left towards the goal and torque's motors are oscillating too. This control strategy is not effi-

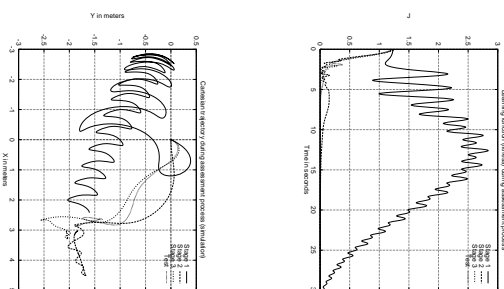


Fig. 5. Learning process for a 2-wheeled robot at the front : cartesian trajectory (top) and learning function (bottom)

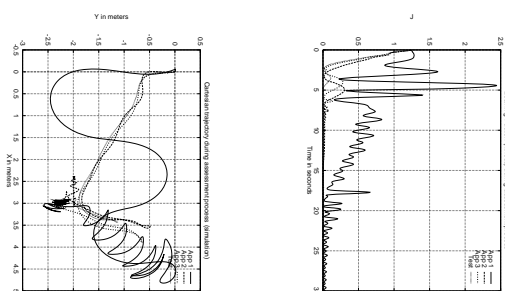


Fig. 6. Learning process for a 2-wheeled robot at the back : cartesian trajectory (top) and learning function (bottom)

cient as the robot cannot stabilize in 30 seconds to the desired configuration. However, on the second stage, it

can be noticed that the oscillation is strongly reduced: there is less skidding and the torque of the motors converge to zero at the end of learning. This can perfectly be seen in the cost function evolution (compare to the stage 1). During the last stage performances are improved, and the test phase is almost perfect: the robot stabilize rapidly with an error below to 10 cm and 4 degrees. For a 2-wheeled robot at the back (see 6), learning stages are similar and as well efficient. Others simulations (not detailed here) show that for a 2 diagonally wheeled robot or 3 wheeled robot, performances are as best than these ones.

C.3 Case of a 4-wheeled robot

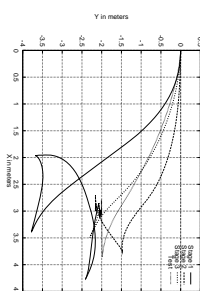


Fig. 7. Learning process for a 4-wheeled robot: cartesian trajectory

In the case of a 4-wheeled robot, learning control does not a priori present any difficulty since it is the ideal case (Fig. 7 and 8) in spite of the number of control parameters is most important and controller convergence is not faster than in others cases. However, it can be said that control is almost perfect since static error after learning is null.

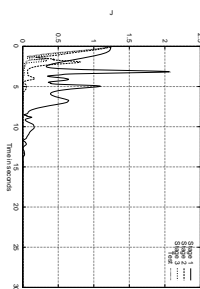


Fig. 8. Learning process for a 4-wheeled robot: learning function

D. Evolution result

What is pointed out through these results is the interest in using the neural controller for robots during the assessment of their capabilities at carrying out tasks.

Each ill-adapted individual is thus given a means to reach or get close to the objective. These results therefore allow to ensure a reasonable opportunity for the robot to express at the best their potentialities before assessment. This is confirmed on Fig. 9 where we can see that the evolution process converges rapidly (compare to Fig. 2) to best solutions : 4, 3 or 2 diagonally wheeled robots. Furthermore, the performances of each generation are better than those of global evolution, because the on-line learning control allows great accuracy.

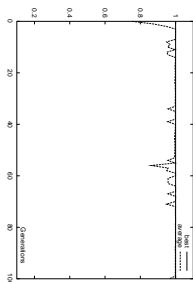


Fig. 9. Evolution result with learning abilities (50 individuals/generation)

III. CONCLUSION

We propose here to give learning abilities during the evaluation of a robots population which have to perform a task of positioning. The evolution results show that only good solutions are quickly found, but also ill-conditioned solutions (a one-wheeled robot for instance) that perform fairly well on the task, remain in the population and thus, contribute to the optimization of the whole population. This approach of evolutionary robotics allows for a much better evolvability level of artificial creatures since they can not only propose new morphologies, but also new control strategies that fit together.

REFERENCES

- [1] M.I. Jordan, "Constrained Supervised Learning", Journal of Mathematical Psychology, Vol 36, pp. 396-425, 1992
- [2] O. Chocron, *Conception Evolutionnaire de systèmes Robotiques*, thèse de l'Université Pierre et Marie Curie, 2000.
- [3] P. Hénaff, G. Bourdon "Fuzzy and Neural Control for Mobile Robotics Experimentations", IEEE/KES'97 Int. Conf. on Knowledge-Based Intelligent Systems, Adelaide, mai 1997
- [4] S. Sehad and C. Touzet, "Neural reinforcement path planning for the miniature robot khepera", World Congress of Neural Network, Volume 2, pp 350-354, Washington, 1995.
- [5] J. Kodjabachian "Développement et évolution de réseaux de neurones artificiels. Application au contrôle d'un animal hexapode", thèse de l'Université Pierre et Marie Curie, 1998
- [6] J. Kodjabachian and J.-A Meyer "Evolution and Development of Control Architectures in Animals", Robotics and Autonomous systems 16(2-4), December 1995, , 161-182
- [7] D. Filliat, J. Kodjabachian, J.-A Meyer "Evolution of Neural Controllers for locomotion and obstacle-avoidance in a 6-legged robot", Connection Sciences, 11, 223-240
- [8] O. Michel "Expériences en Neuroéthologie Artificielle EVOTS : Une méthode évolutionniste appliquée en robotique mobile", Thèse de l'Université de Nice Sophia Antipolis, 1996
- [9] Arnaud Revel, "Contrôle d'un robot mobile autonome par une approche neuromimétique", Thèse d'Université de Cergy-Pontoise, 1997