



HAL
open science

Evolutionary synthesis of structure and control of locomotion systems

Olivier Chocron, Nicolas Brener, Philippe Bidaud, Faiz Ben Amar

► **To cite this version:**

Olivier Chocron, Nicolas Brener, Philippe Bidaud, Faiz Ben Amar. Evolutionary synthesis of structure and control of locomotion systems. Climbing and Walking Robots CLAWAR, 2004, Madrid, Spain. <hal-03178088>

HAL Id: hal-03178088

<https://hal.science/hal-03178088v1>

Submitted on 23 Mar 2021

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

Evolutionary Synthesis of Structure and Control for Locomotion Systems

O Chocron*, N. Brener**, Ph. Bidaud** , F.B Amar**

* Laboratoire d'Ingénierie Informatique, ENIB/CERV, Brest

** Laboratoire de Robotique de Paris, Université Paris 6 / CNRS 18

Abstract

This paper presents a global design approach for self-reconfigurable locomotion systems based on evaluation by dynamic simulation and optimization by artificial evolution. The main objective of this approach is to obtain fully integrated robotic solutions in term of morphology (topology and kinematics) and control (architecture and command). To achieve this goal, a modular robotic system is designed, including modules design and topology representation. Both topology and control are to be co-evolved through an evolutionary algorithm that accounts for the technological constraints (with precise module design) and multi-objective robotic missions (with detailed realistic simulation). Several design examples of modular systems are studied (rover and snake) and the autonomous reconfiguration ability is shown through simulation from rover to snake. The snake is simulated using the ODE tool showing how the robot actually walks (or crawls). The robot is feedback controlled in velocity to perform a snake-like crawling. Results of behavior and desired and simulated velocities validate the design method for locomotion systems.

1. Introduction

Many efforts have been made since a decade to obtain self-reconfigurable robotic systems (SRS) that are able to adapt their morphology to the task being undertaken. This need in adaptation has been empathized particularly in locomotion systems because many obstacles and uncertainties may be encountered while walking on rough terrains [1]. The reconfiguration possibilities can obviously be highly improved with modular robotic systems (MRS) that can change their topology by simply connect or disconnect a large number of modules that are identical one to another. The challenges of such systems are fundamental as well as technological [2]. The advantages of using MRS in association with the modular approach to achieve adaptation by reconfiguration are many:

- Each module being previously designed, the properties of the system are easily obtained from extrapolation of the modules ones.

- A deficient module can be straightforwardly replaced by another available module. This brings fault tolerant capacities for out of reach environments.
- The number of different assemblies (i.e. topologies) is exponential to the modules number, enabling a very basic functional design of modules.
- Distribution of critical resources (like power, actuation, sensors and CPU) through many modules brings robustness to the overall robot in operation.
- Mass production of a few different types of modules allows for a low cost design and production of a great number of modular robotic systems.

This promising approach has motivated researchers to develop modular SRS for critical robotic applications like space exploration [3], locomotion [4] and micro-robotics [5]. Some concepts have been thoroughly explored and led to interesting high potential systems [6]. The way and means the modular system has to adopt a new configuration according to the task considered is yet a whole new field in robotics that we could call: Modular Adaptation in Robotic Systems.

The adaptation is modular, not only because it applies to MRS, but also because of the task modularity itself. In locomotion task, we know that several objectives or constraints must be optimized such as speed, energy consumption, stability, safety and reliability [7]. Each objective can be seen as a part of the behavioral solution that could be solved separately. Meanwhile, the interdependency of the objectives (through their involved solutions) makes the design process a truly global optimization problem.

The use of highly mobile robots has been proposed for locomotion over natural terrains in planetary exploration or in military applications. Many research works have focused on behavior adaptation using advanced control methods [8][9] or artificial evolution [10], but very few of them have concerned the evolution of the mechanical structure and its behavior simultaneously. What we have learned from the different design experiences of such complex systems is that a co-evolution of the structure and its behavior, demands a realistic simulation to be relevant [11][12].

To answer these three issues (optimal module combination, adaptive behavior and relevant evaluation) we proposed an evolutionary design process in an earlier work [13]. A more advanced design of modules and a more powerful simulation tool has been proposed in a later work [14]. In this paper, we extend our previous works on evolutionary optimization for modular mobile robots to the design of integrated and controllable modular locomotion systems and present first results.

2. Global evolutionary design process for adaptation

Evolutionary (or artificial evolution) algorithms are optimization methods inspired by natural evolution principles (see Fig.1). The candidate solutions of a given optimization problem are considered as individuals in a population that undergoes genetic operators. In that way, a population is bred under the *survival of the fittest* Darwinian law and will evolve toward fitter and fitter individuals.

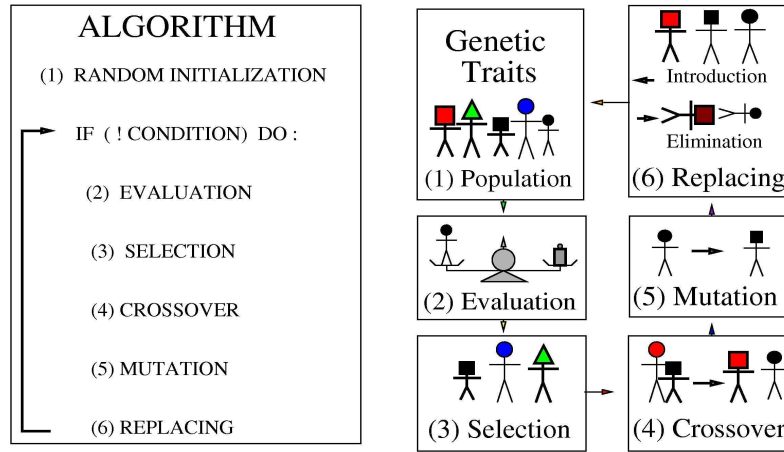


Fig. 1. Evolutionary Process over a symbolized population

Genotype Encoding

The use of a modular approach in robot design requires defining mechatronic modules as well as their assembly modes and their control system. Several kinds of modules and connecting ways may be considered. Actuators and sensors can also be modules since they can be combined together (motor with its gear or a photodiode with an optical lens). Some basic modules for locomotion have been proposed: **base** (payload), **segments**, **wheels** and **joints** including actuators (Fig. 2).

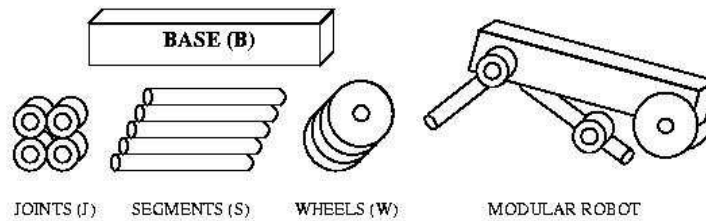


Fig. 2. Examples of basic modules for locomotion systems

All basic modules of the same kind are strictly identical. Revolute joints are used to link two bodies (base, segments or wheels). The modular system architecture results from connections between the modules. These connections are encoded using an incidence matrix \mathbf{M}_{ij} in which solid bodies \mathbf{S}_i are placed in rows and joints \mathbf{J}_j in columns. The integer value \mathbf{M}_{ij} at the matrix nodes determines the nature of the connection (i.e. the linkage) between a body and a joint (see Fig. 3).

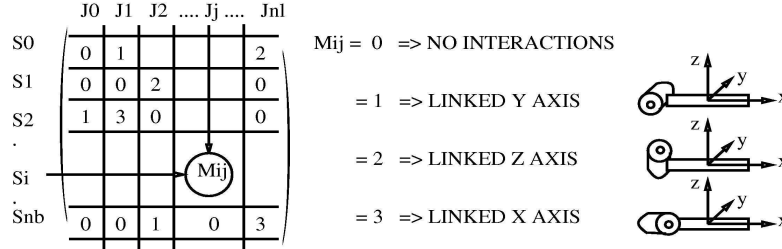


Fig. 3. Incidence matrix for mechanical system encoding

The platform (base) is represented by the first row and the position of attachment points for joints are regularly and symmetrically fixed on the platform sides. The segments are attached by both extremities and the wheels at their center. Since joints and segments are bivalent, only two slots at most in the associated column or row can be filled (number different from zero), only one for the wheels and as many as the number of joints for the base.

A construction algorithm as been designed to interpret the incidence matrix. In such a way, we can describe any topological configuration for a given set of modules. The advantage of such a representation is its compactness and its generality (every modular robot can be encoded in such a way). The control laws applied on the joints have to be task-based and dependent of the robot state with regard to the task and the environment. We propose to include the control system into the genotype and to let the genetic process search for an adapted command in parallel with the topology. The control law is defined as the input voltage of the actuators (modeled as DC-motors with gears) associated to the joint (see [13] for more details).

$$U_j = U \max_j * \cos(\omega_j * t + \phi_j)$$

- U_{max} : Max. voltage on joint j
- ω_j : Pulse of signal on joint j
- ϕ_j : Phase of signal on joint j
- t : Simulated time

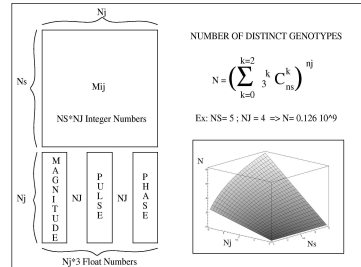


Fig. 4. Topology/Behavior genotype

This control law allows only to generate periodic functions and to command the system in an open loop way without any feedback. The goal is to test the adaptation power of artificial evolution for ill conditions. Every control law parameters (U_{max} , ω_j and ϕ_j) are normalized (from 0 to 1) and placed in three float vectors which constitute the behavior genotype. The resulting genotype is constituted by the integer incidence matrix (encoding the structure) and three float vectors (encoding the behavior) as shown in figure (Fig. 4). The number N of possible distinct genotypes for the structure grows exponentially with the number of bodies.

Phenotype Evaluation

We have to deal with two distinct genetic entities; **Topology** and **Behavior**. Both are evolved simultaneously in the same global evolutionary algorithm (See Fig.5). The evaluation process is done by an approximated dynamic simulation of the robot in interaction with its evolution environment (with choc and friction).

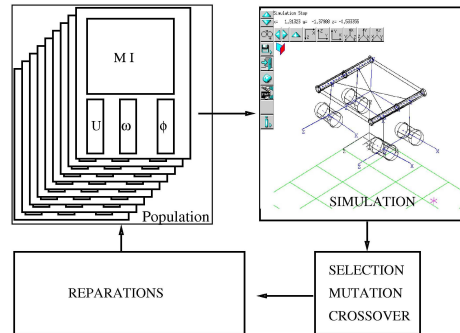


Fig. 5. Global Evolutionary Design Algorithm

The evolutionary algorithm calls upon the simulation each time it needs to evaluate a robot for completing the specified task. The simulation being the most time consuming stage, it has to be used sparingly. We use a hierarchical evaluation (mathematical evaluation, short and full simulation) to assess the robot fitness.

Simulation results

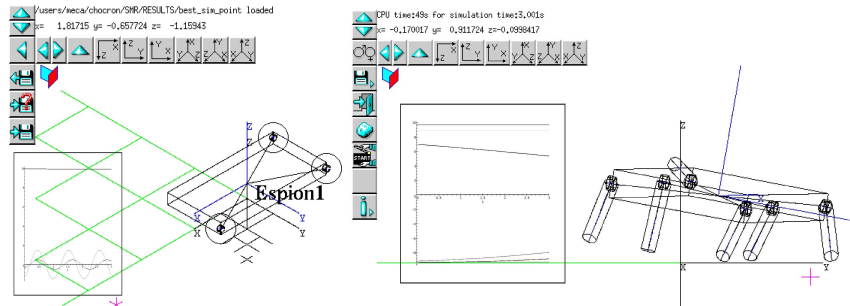


Fig. 6. Results for the reaching goal (left) and the getting altitude tasks (right)

The population has been evolved to perform four tasks on flat terrain: maximizing speed, reaching a goal (for a fixed time), getting altitude and fast spinning. The results turned out original and worthy in spite of the open loop control architecture. Complete solutions (topology and behavior) for the tasks of reaching a goal on the ground and getting as higher as possible are presented (Fig. 6). These results prove that the evolutionary and simulation approach proposed is efficient since different types of robots (wheeled and legged) as arisen from a strictly task-based criterion.

3. Design of a Self Reconfigurable Robot.

In [14] we proposed an original design of a Self Reconfigurable System, which consists of several mechatronic modules with connecting plates. By manipulating itself the system can dynamically change its topology depending on the situation. For a review on already existing or prototyped systems see [1 to 5, 15 and 16].

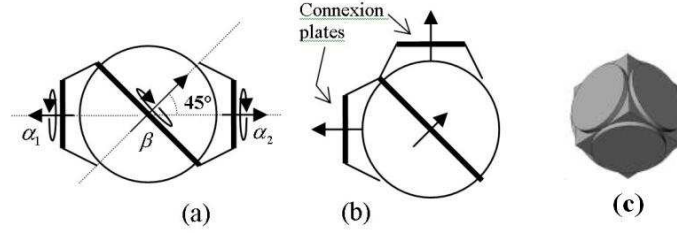


Fig. 7. Ball module configuration straight (a) and full (b) and cube module (c)

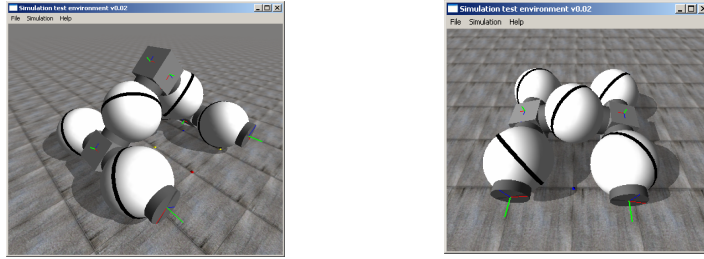


Fig. 8. Two rover topologies: independent steering (a) and linked steering (b)

This design provides wheeled locomotion when two balls are connected on opposite faces of a cube. Fig. 8 shows two rover topologies. By coordinating its three rotations the ball can provide in-plan bending or can orientate the bending plane. Let α_1 and α_2 be the rotation angles of the two plates and β be the middle axis angle, with zero values in configuration (a) of Fig.7. Let θ and ϕ be respectively the azimuth and the pitch angles of the ball, set to zero in configuration (a). The relations between inner angles (α_1, α_2 and β) and the outer angles (θ and ϕ) are given in (Eq.1). Pitch and azimuth provide easier control of module attitude since they are closer to the natural 3D space representation of motions.

$$\begin{cases} \alpha_1 = \arctan \frac{\cos(2\phi) - 1}{\sqrt{2} \sin(2\phi)} + \theta + \lambda\pi \\ \beta = 2\phi \\ \alpha_2 = \arctan \frac{\cos(2\phi) - 1}{\sqrt{2} \sin(2\phi)} - \theta + \lambda\pi \end{cases} \quad \text{with} \quad \begin{cases} \lambda = 1 \text{ if } \phi < -\pi/2 \\ \lambda = -1 \text{ if } \phi > \pi/2 \\ \lambda = 0 \text{ else} \end{cases} \quad (\text{Eq.1})$$

Topologies and reconfiguration

The Open Dynamics Engine library for C (ODE) has been used to study and simulate some robotic locomotion systems (see Fig. 9).

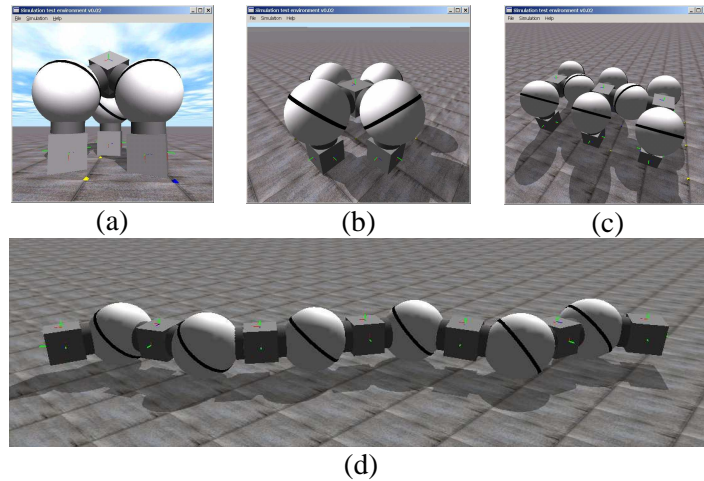


Fig. 9. Some topologies: (a) a tripod, (b) a quadruped, (c) a hexapod, (d) a snake.

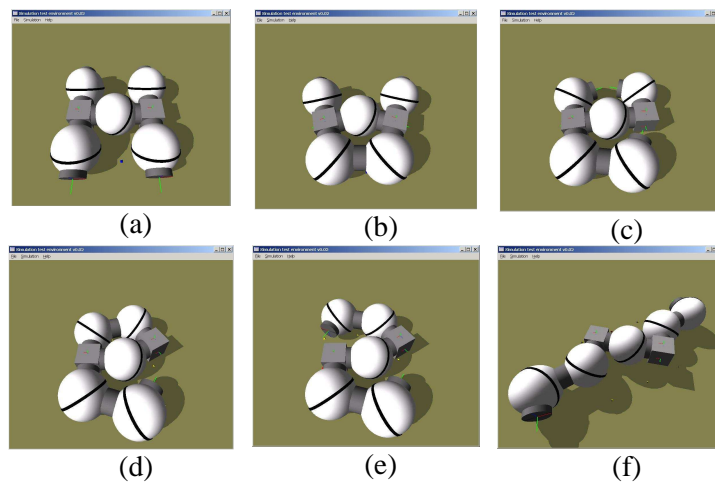


Fig. 10. Reconfiguration from rover to snake. (a) rover configuration, (b) connecting both south wheels, (c) disconnecting the southeast wheel from the east cube, (d) connecting both north wheels, (e) disconnecting the northwest wheel from the west cube, (f) expanding in snake configuration.

The robot can change its topology by moving, connecting and disconnecting modules. The reconfiguration stages from a rover to a snake are shown in Fig. 10.

To verify that ODE is suitable for the evaluation of performing robots, a locomotion simulation was run for the snake topology (Fig. 9d). We plotted the desired motor angular speed (empirically designed to obtain a crawling) versus the observed speed (Fig. 11). ODE computes the real speed according to the maximum actuator torque that is user defined. Here, we limited the actuation torque to 15 Nm.

- Cubes modules : mass = 1kg, edge=0.1m
- Ball modules: mass= 4.5kg radius = 0.1m

At each time step the pitch of all 6 ball modules is computed as below. A phase lag is set between each ball module. This phase lag has been introduced so that a motion wave propagates through the snake body. The azimuth angle is set to zero.

```
A:=1.7; step=0.04; phase_lag:=-3.9;
repeat
  for each i in [1..6] do
    ball(i).set_pitch(A* cos (inc + i * phase_lag))
  endFor
  inc := inc+step
endRepeat
```

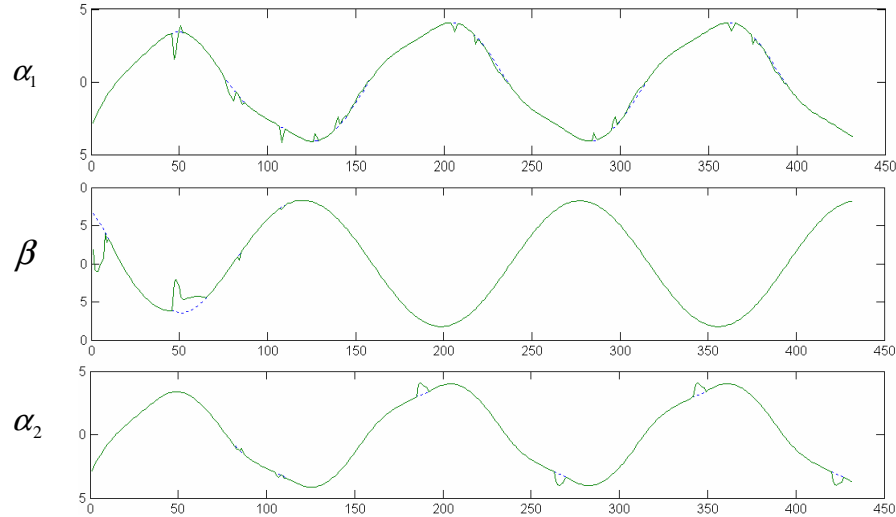


Fig. 11. Third ball module angular speeds versus time step: Dotted line shows the command speed and the solid line shows the simulated speed.

The inner angles are then computed using Eq1. The motor velocity command is computed by numerical derivation of the desired angle positions. The results are then passed to the module actuators. The ODE simulator takes into account the maximum defined torque for each joint and applies only acceptable torques. Hence, there are some discrepancies between the desired and observed joint angular speed caused by interactions of the robot modules with the ground (Fig. 11).

5. Conclusion

We have implemented an evolutionary algorithm in order to evolve modular robotic systems for locomotion. The representation with incidence matrices is generic enough to obtain any type of mobile robot (wheeled, legged, crawling or hybrid). The association of evolutionary computations and dynamic simulation has yielded some consistent results proving the reliability of the method. The new design of modules (ball joints and cube modules) integrates structure and kinematics. The combination of these modules allows for any kind of poly-articulated mechanisms (as serial or parallel ones), actuation (redundant or not) and thus, many modes of locomotion that could be exploited by the search power of artificial evolution. For future works, we consider using a graph representation for connection performances and to integrate the control architecture and the topology. We could also exploit the *step fast* mode of ODE that approximates the solving of dynamic equations with a computation complexity proportional to the module number. This can be applied to obtain a hierarchical evaluation through progressive simulation.

References

- [1] T. Ueyama, T. Fukuda and F. Arai (1992), “*Structure Configuration using Genetic Algorithm for Cellular Robotic Systems*”, Proc. IEEE/RSJ ICIRS,
- [2] Farritor, S., Dubowsky, S., Rutman N., & Cole, J. (1996): “*A System-level Modular Design Approach to Field Robotics*”, Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA’96).
- [3] M. Yim, K. Roufas, D. Duff, Y. Zhang, S. Homans (1999), “*Modular reconfigurable robots in space applications*”, Autonomous Robot Journal, robots in space, Springer Verlag, 2003. II
- [4] Keith Kotay and Daniela Rus (1999), “*Locomotion versatility through self-reconfiguration*”, *Robotics and Autonomous Systems*, 26 (1999), 217-232.
- [5] E.Yoshida, S.Murata, S.Kokaji, K.Tomita, H. Kurokawa (2001), “*Micro selfreconfigurable modular robot using shape memory alloy*”, Journal of Robotics and Mechatronics, Vol.13, No.2 212
- [6] S. Murata, et al. (2002), “*M-TRAN: self-reconfigurable modular robotic system*”, IEEE/ASME Trans. Mech. Vol. 7, No. 4, pp. 431-441
- [7] Celaya E. and Albarral J. L. (2003), “*Implementation of a hierarchical walk controller for the LAURON III hexapod robot*”, CLAWAR 2003, Int. Conference on Climbing and Walking robots
- [8] Caballero R., Akinfiyev T., Armada M. (2002), “*Robust cascade controller for robicam biped robot: preliminary experiments*”, CLAWAR 2002, Madrid
- [9] Adouane L. and Le Fort-Piat N. (2004), “*Hybrid Behavioral Control Architecture for the Cooperation of Minimalist Mobile Robots*”, Proc. of the IEEE Int. Conf. on Robotics and Automation
- [10] Ziegler J., Barnholt J., Busch J. and Banzhaf W. (2002), “*Automatic evolution of control programs for a small humanoid walking robot*”, CLAWAR 2002
- [11] K. Sims (1994), “*Evolving 3D Morphology and Behavior by Competition*”, Artificial Life IV , Proc. ed by R. Brooks and P. Maes, MIT Press, pp.28-39
- [12] Nick Jakobi (1998), “*Running Across the Reality Gap: Octopod Locomotion Evolved in a Minimal Simulation*”, Lecture Notes in Computer Science 1468, Eds P.Husbands and J-A. Meyer.
- [13] O. Chocron and P. Bidaud (1999), “*Evolutionary Algorithm for Global Design of Locomotion Systems*”, Proc. IEEE of IROS’99, Kyongju (S. Korea)
- [14] N. Brener, F. Ben Amar, P. Bidaud (2004) “*Analysis of Self-Reconfigurable Modular Systems, A Design Proposal for Multi-Modes Locomotion*”, IEEE Int. Conf. on Robotics and Automation
- [15] Cem Ünsal and Pradeep K. Khosla (2000), “*Solutions for 3-D selfreconfiguration in a modular robotic system: implementation and motion planning*”, Sensor Fusion and Decentralized Control in Robotic Systems III.
- [16] Michel Dubois, Yann Le Guyadec, Dominique Duhaut, “*Control of interconnected homogeneous atoms: language and simulator*”, CLAWAR 03, Int. Conf. on Climbing and Walking Robots