Robotics and Autonomous Systems 63 (2015) 195–205

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

An evolution friendly modular architecture to produce feasible robots

A. Faiña [a], F. Bellas [b,*], F. Orjales [b], D. Souto [b], R.J. Duro [b]

[a] IT University of Copenhagen, Denmark

[b] Integrated Group for Engineering Research, Universidade da Coruña, Spain

HIGHLIGHTS

- We present a modular architecture to produce feasible robots through evolution.
- The architecture is based on a set of a heterogeneous modules.
- The modules contain a large number of connection faces per module.
- The design and the implementation of prototype modules is described in detail.
- Different experiments show its potential for evolving robot morphologies and control.

ARTICLE INFO

Article history:
Available online 7 August 2014

Keywords:
Modular robots
Evolutionary design
Evolvability

ABSTRACT

This paper proposes the use of a modular robotic architecture in order to produce feasible robots through evolution. To this end, the main requirements the architecture must fulfill are analyzed and a top-down methodology is employed to obtain the different types of modules that make it up. Specifically, the problem of how to increase the evolvability or evolution friendliness of the system is addressed by considering a heterogeneous modular architecture with a large number of connection faces per module. Afterwards, a prototypical implementation of these modules with the required features is described and different experiments provide an indication of how versatile the architecture is for evolving robot morphologies and control for specific tasks and how easy it is to build them.

1. Introduction

Designing a robot for a specific task and environment is a complex process that relies heavily on the expertise of designers. It usually involves two isolated steps where the morphology of the robot is first selected according to the environment and task features and, afterwards, the controller is programmed. These two isolated phases for designing the morphology of the robot and its behavior do not exploit the fact that morphology, controller, and environment are highly interdependent. In fact, several authors have pointed out that embodiment is a key aspect for developing really intelligent robots [1,2]. Complex, robust, and well adapted behaviors can be obtained with simpler controllers by exploiting the "morphological intelligence" of the robots [3–5].

One of the main approaches to address this complex design process is based on evolution. In fact, in the last three decades, several examples can be found where evolutionary algorithms have been successfully employed to design controllers for robots. On one hand, some authors have resorted to fixed morphologies [6–8]. These embodiment approaches use the interrelations between the morphology of the robot and its environment to find a suitable controller which provides the desired behavior. On the other hand, in order to automate the whole robot design process for a task and environment, some authors have introduced the morphology of the robot as part of the search space and they simultaneously co-evolve morphology and control. As commented above, the main advantage of this approach is that it exploits the interrelations between the morphology, the control and the environment for a specific task.

Following this second approach, authors such as Sims, in their seminal work, coevolved the morphology and the control of virtual creatures for tasks such as walking, swimming or jumping [9]. His work was based on cubic bodies joined by hinges, neural networks as a control system and a generative encoding. After his work, other authors have coevolved robots for different tasks in a similar way [10–12]. Nevertheless, these approaches only produce virtual creatures that cannot be transferred to reality without an ad hoc adaptation. Furthermore, as the algorithms do not take into

* Correspondence to: Escuela Politécnica Superior, Universidade da Coruña, Mendizábal s/n, 15403 Ferrol (A Coruña), Spain. Tel.: +34 981 337400x3886; fax: +34 981 337410.

E-mail addresses: anfv@itu.dk (A. Faiña), francisco.bellas@udc.es, fran@udc.es, francisco.bellas@gmail.com (F. Bellas), felix.orjales@udc.es (F. Orjales), dsouto@udc.es (D. Souto), richard@udc.es (R.J. Duro).

http://dx.doi.org/10.1016/j.robot.2014.07.014
0921-8890/

account any physical constraint, most of the solutions are unfeasible. For example, most of these approaches employ joints with high torque motors, very light weight structures and dynamic engines configured to achieve fast evaluations, but with low accuracy.

Other authors have studied how to achieve feasible morphological designs automatically. To this end, one interesting approach was employed in the Golem project [13]. Here, the morphological shape and the control parameters of virtual structures based on bars (some of them with telescopic actuators) and ball–socket joints were coevolved. Afterwards, some selected morphologies were processed by an algorithm to obtain feasible designs which could be built using a 3D printer. After manually removing the support material and installing the motors, these robots were able to operate using an external control structure. More recently, Cheney et al. [14] have coevolved soft robots using generative encodings. These robots can be built using an ad hoc process and they can be actuated varying the external pressure [15].

A different approach to automatically obtain feasible robots is to make use of some kind of blocks as a basic set of elementary building parts for the morphological evolution. This approach implies a discrete search space, but it guarantees that all solutions can be built. In this line, some authors employ Lego bricks as the basic element for morphological construction. A point in case is [16], which is more focused on morphology than on control or, more recently [17], that employs three different types of parts (a hinge joint, a controller block and several Lego bricks) for evolving the robots. Apart from Lego bricks [18], proposed using bars and circular sockets (actuated or fixed) as basic elements to design robots using a generative encoding based on Lindenmayer systems.

The main drawback of these approaches is that, although the robots they produce are feasible and can be built, they are only a proof of concept to show that coevolution can achieve successful robots for simple tasks like locomotion on flat surfaces in laboratory environments. Furthermore, most of the robots need a laborious building process to obtain the desired morphologies and, obviously, Lego bricks are not a suitable architecture to generate useful robots in any real industrial environment.

This paper shares with the previous approach its use of predefined blocks, but with a different perspective. The work presented here is based on robotic modules, which allow us to quickly deploy useful modular robots for complex tasks and environments. Modular robots are built by joining some relatively simple devices called modules. They are autonomous devices with a few actuators, sensors, communications, and some computational capabilities. Complex robots with different morphologies can be created by combining a small set of predesigned modules. There are a lot of different modular architectures that have shown high versatility for building different morphologies [19–23]. The use of modular robots to coevolve the morphology and control guarantees that all the solutions obtained are feasible and they make building the robots easier and faster. Nevertheless, compared to using simple blocks such as Lego parts, the new search space these modules induce makes evolution harder due to the increased deceptiveness. Evolution is still discrete but its resolution is decreased as the blocks are larger. Consequently, the addition or deletion of one of the modules generates more pronounced changes in the behavior of the robot.

The first attempts to design modular robots tried to obtain the configuration by only evaluating the morphological features of serial manipulators [24–26]. That is, the controller and dynamic properties of the system were not taken into account and were replaced by an analysis of the kinematics of the system. Similarly, Farritor and Dubowsky [27] explored this approach to develop robots for industrial tasks based on a kinematic analysis and a set of features of the task like, for example, the tallest obstacle that the robot had to go over. Leger [28] resorted to kinematic and dynamic analyses to generate field robots based on a base element with several serial manipulators, where the paths for the end effectors of the manipulator were predefined as a part of the task. Similar to this work, Chocron coevolved modular robots for rough explorations using dynamic simulators [29].

Despite the fact that all of these approaches generate feasible robots, they are mostly based on simulated modular architectures and they lack a physical implementation. Only a few authors have experimented with designing modular robots using real modular architectures. In this line, Lund coevolved the morphology and control of line-follower robots based on modules built using assemblies of Lego parts [30] (in this case they use modules with sensing and acting capabilities, and not only Lego blocks). Also, Marbach and Ijspeert coevolved simulated virtual robots using central pattern generators and Yamor modules, an homogeneous architecture based on hinge joints [31].

While simulated robotic modular architectures present interesting properties in order to increase their versatility, and in some cases, even their evolvability, real implemented modular architectures generally lack these features. In fact, almost every real modular architecture did not take evolvability as a design parameter when they were being designed. For example, most simulated architectures are heterogeneous or present a high number of connecting faces. However, most modular architectures implemented in the real-world rely on homogeneous modules with only a few connecting faces per module. On the other hand, simulated architectures do not address most of the issues that are faced in real-world architectures such as power transmission, computational and communication capabilities, structural stability, robustness, etc.

This work seeks to fill this gap by developing a modular architecture that is appropriate for building robots through evolution. Therefore, the architecture must face real hardware issues and, at the same time, it must provide a high level of versatility to build different robot morphologies and a series of characteristics that can help evolution. The architecture will provide a basic tool to generate feasible and useful robots with robust and well-adapted behaviors taking into account the interrelation between the environment, the morphology, and the controller of the robot.

The paper is structured as follows: Section 2 contains the initial requirements established for the architecture with the aim of promoting evolvability and feasibility and the design principles adopted to fulfil them. Section 3 is devoted to the details of the specific implementation of the architecture in a set of prototype modules. In Section 4, the capabilities of the implemented modules are shown through the construction of several real robotic structures. Section 5 contains a summary and discussion of the main results obtained when the architecture was applied to the evolutionary design of robots in linear and static missions. Finally, the main conclusions of this work and future directions in this line are commented in Section 6.

2. Requirements and design of the architecture

The requirements that a modular robotic architecture must fulfil in order to facilitate the evolution of real robots able to work in different and useful tasks were analyzed in depth. Whereas most modular architectures only take into account requirements related to the deployment and operation of modular robots, here the interest is also in making the architecture more evolution friendly. Thus, the combination of the two types of desires lead to the following requirements:

- Evolvability: in order to achieve successful morphologies for the robots, the architecture should allow for enough morphological variation within the population to prevent premature convergence and it should allow the generation of well-adapted robots by morphological mutations.

- Fast deployment: after a new morphology is obtained by evolution, it should be easy to assemble in a short time.
- Fault tolerance: a failure in one module should not affect the operation of the other modules. Thus, the robot should be able to continue with the task.
- Robustness: the modules must work in real environments and they should resist the external forces that are generated during their operation.
- Reduced cost: the modules must be cheap in order to manufacture several of them with a low budget.
- Scalability: the number of modules in one configuration should not affect the performance of the system. Modular robots have to be able to work with a lot of modules without performance degradation.

To fulfil the above requirements, a set of design decisions were made. First of all, a set of heterogeneous modules with a high number of anchor points were selected as the core elements of the architecture. These two features (heterogeneity and high number of anchor points) increase the evolvability of the system and, as pointed out earlier, they are not present in most real modular architectures. This high versatility of the modules exponentially increases the number of different morphologies that can be built. The large number of different phenotypes that can arise, even when using only a relatively small number of modules, favors the preservation of high morphological diversity in the population. On the other hand, a heterogeneous modular architecture allows solving a given task usually with a lower number of modules than a homogeneous one due to the different degrees of freedom they provide. Consider, as an example, a task involving painting a flat wall. Most real modular architectures are based on hinge modules, so they need to assemble a long string of them with a complex control structure to move the end effector in front of the wall. On the other hand, if one has a heterogeneous architecture with some linear actuators, an adequate robot can be obtained with only two modules and a very simple controller.

The second decision that has been made is that the architecture should have a low number of different modules so that the search space would not become huge. As commented above, this is aimed at promoting evolvability, but preserving feasibility and the advantages of modularity. Any modular architecture will present different types of modules depending on their general functionality within the robotic structures. These can be organized into five categories: actuators, end-effectors, expansion modules (computational capabilities, batteries, etc.), specialized sensors (like cameras or ultrasounds) and linkers. It must be pointed out that, as commented in the previous section, the modules are autonomous devices that contain their own processor, motor, power system, communications capabilities and sensors. However, from an evolutionary point of view, the most important groups are those of actuator modules and linker modules. They are the ones that determine the main characteristics of the morphology and motion capabilities of a robot. Consequently, from the point of view of evolution friendliness, those are the ones aimed here to keep to a minimum number without losing the advantages of heterogeneous modular systems.

With the two previous features in mind, a top-down methodology has been followed to specify a basic set of actuator modules for the robotic architecture. Inspired by the type of tasks that are performed in real environments, three general missions have been selected for the robots: linear, surface and static. The first ones consist in following a path across the environment such as moving a payload from one location to another. Surface missions are related with jobs in an area, like cleaning a room. Finally, static missions are those where the robots are fixed to one point and they move their end-effectors to carry out a task (for instance, painting an object). These basic missions were decomposed into tasks and subtasks as displayed in Fig. 1. Afterwards, the kinematic pairs required to accomplish the subtasks were analyzed and it was decided to employ only two of them with one degree of freedom: revolution and prismatic joints. This makes the modules cheaper and their simple mechanics allow for increased robustness. Nevertheless, to achieve basic motion primitives, two different modules were designed for each kinematic pair. The prismatic joint is implemented as a telescopic module, which increases or decreases its overall length, or as a slider version with a linear motion over its structure. Similarly, the revolution joint is implemented as a rotational version to be employed, for instance, in wheels, and as a hinge version. Finally, the lower layer in Fig. 1 shows some different end-effector modules that could be used.

A schematic view of these four actuator modules is displayed in Fig. 2. All the modules were designed to have a large number of connecting faces. These have been placed preferentially at the ends of the modules in cubic connection bays and they can join modules at 90° orientations. Also, this design permits building complex morphologies or closed kinematic chains and reduces the number of linkers that are needed to construct the robot structure. As an example, the different possibilities for connecting a slider module to a connection face are shown in Fig. 3.

In order to fulfil the feasibility requirements, the connectors of the modules of each face have been selected to allow for robust mechanical fixation that can be deployed or removed in seconds. In addition, these connection faces must allow for power sharing and communications. In terms of power, the modules were designed to be powered by an external power supply or through a small battery. The power provided by these inputs, which can be connected to one or more modules, is shared across the robot.

Regarding the communications capabilities, it has been decided to have three different channels. First, a wired and fast global bus as the main channel to coordinate the movements of the modules and share sensor information. In addition, a wireless communication channel is proposed to allow communications between modules that are not connected or supplement the capacities of the wired channel. Finally, the modules will have local communications with the neighbors they are connected to for recognizing their own position in the configuration of the robot. These three different paths guarantee robust communications even if a module completely fails.

3. Implementation of prototype modules

In the previous section, the main features and components of the modular architecture were presented. Here a description of the different solutions for the actuator modules that have been adopted is given through the presentation of a specific prototype implementation. Throughout this section, the design and morphology of the prototype modules will be explained as well as the different systems needed for the operation of the robot, such as the energy supply system, communications, control system, etc.

Fig. 4 displays the different modules (actuators, linkers and effectors) that have been fully designed and implemented in prototype form. They all comprise nodes built using fiber glass from milled printed circuit boards (PCBs). These parts are soldered to achieve a solid but lightweight structure. Each module can have one or more nodes, which act as connection bays. The shape of the nodes can vary depending on the type of module. Nonetheless, all of them provide one or more connection bays on their free sides. Their size without the connection mechanism is 48 mm × 48 mm × 48 mm and 54 mm × 54 mm × 54 mm including the connectors.

3.1. Actuator modules

According to the specifications of Section 2, four different types of actuator modules have been built: two linear actuators (slider and telescopic modules) and two rotational actuators (rotational

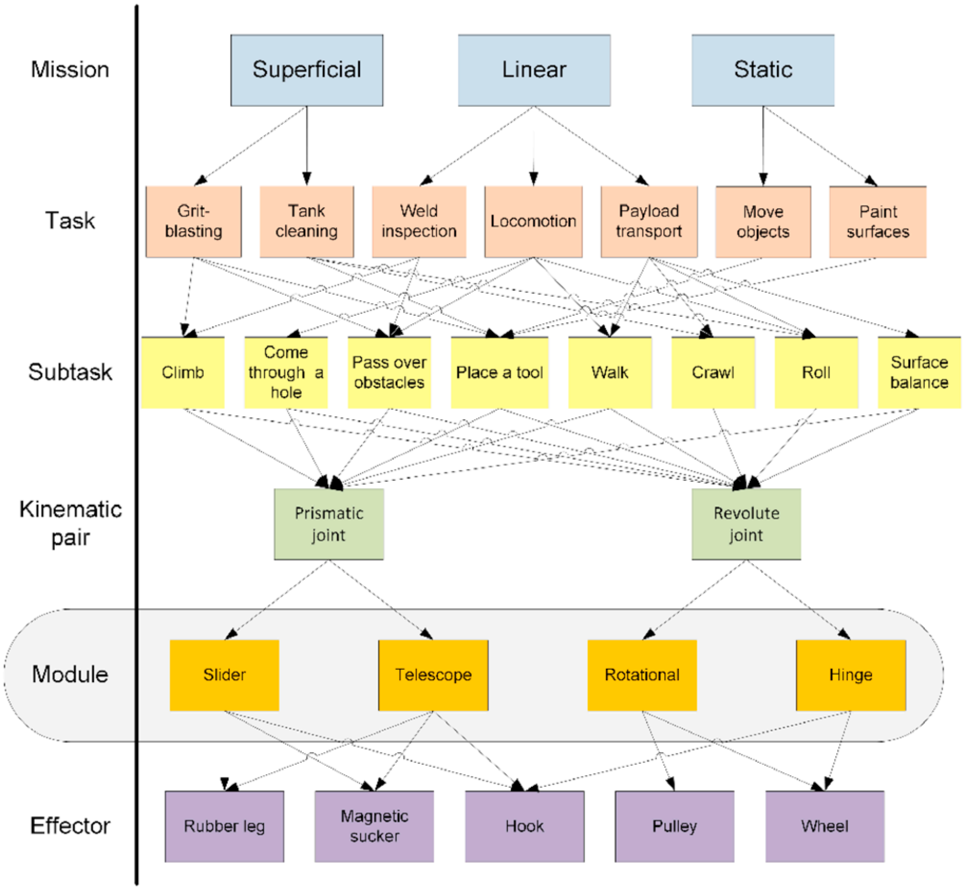

Fig. 1. Diagram of the missions, tasks and sub-tasks considered, and the required actuators and effectors.

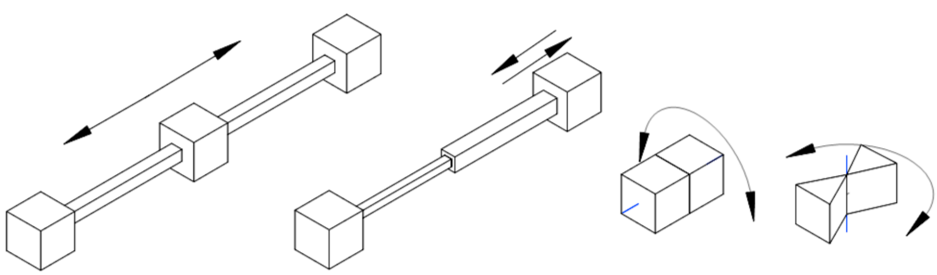

Fig. 2. Schematic representation of the four actuator modules.

and hinge modules). They only provide one degree of freedom in order to increase robustness and they contain different types of joints so that it is easy to build most of the kinematic chains used by real robotic systems. The main features of the actuator modules are summarized in Table 1.

3.1.1. Slider module

The slider module (the one on the right in Fig. 4) is made up of two cubic nodes, with five connection bays each, joined together using three carbon fiber tubes. Another node, the slider, slides along the tubes between the end nodes. The distance between the end nodes is 249 mm and the stroke of the slider node is 189 mm. Its motion is achieved through a servo with a pulley in one of the end nodes that moves a drive belt to which the slider node is fixed through a return pulley in the other end node. The sliding node contains the electronics of the module.

3.1.2. Telescopic module

The telescopic module (the one on the left of the slider module in Fig. 4) has two cubic nodes on its ends, being the distance between them variable. One node has the electronic board with all the control, communications and sensing elements of the module. The other contains a servo with a drive pulley that allows the contraction and extension of the module.

3.1.3. Rotational module

This module (center of Fig. 4, below the linker) has two nodes that can rotate with respect to each other. A low friction washer between the nodes and a shaft prevents misalignments. It is configured so as to permit 360° rotations of the nodes with respect to each other. Among its sensing elements, it can provide the relative rotation between nodes.

3.1.4. Hinge module

The hinge module (below the rotational module in Fig. 4) does not have cubic node in its structure, only one connection bay in each main block. A shaft joins two main parts built from milled PCBs. These parts rotate relative to each other as a hinge (with the rotation axis in one of their ends).

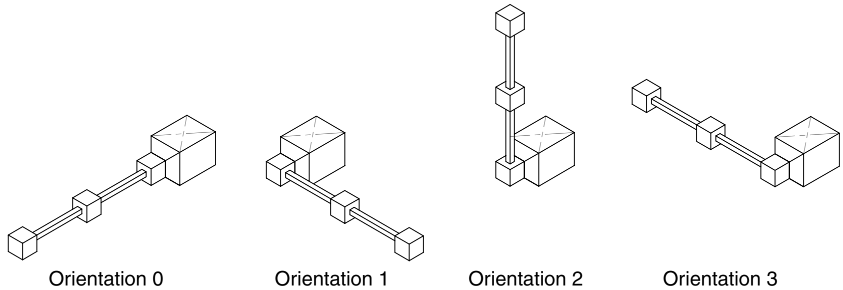

Fig. 3. Different possibilities to connect a slider module.

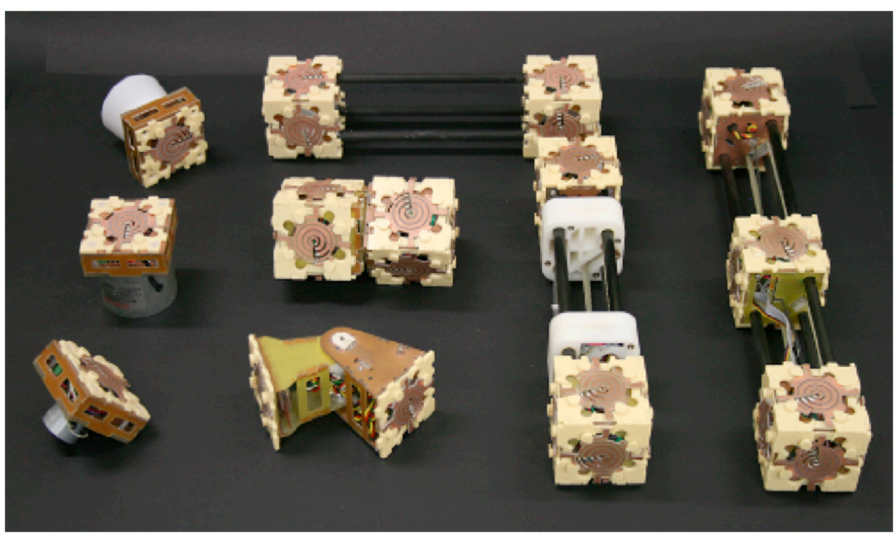

Fig. 4. Different types of modules developed: three effectors on the left side; rotational, hinge and telescopic modules in the middle; unactuated linker on the top and slider on the right.

3.2. Connection mechanism

As commented in Section 2, a mechanical connection has been designed to be able to easily join the modules. In addition to providing mechanical support, the mechanism transmits power and communications. The connector design can be seen in the nodes displayed in Fig. 4 and it has two main parts: a printed circuit board and a resin structure. The resin structure has four pins and four sockets to allow four connections at angles that are multiples of 90°. Inside the resin structure there is a PCB that can rotate 15° in order to be able to latch it to the connector it faces. This PCB has four concentric copper tracks on the top side that provide electrical contact between nodes. Two of the tracks are employed to transmit power (GND and +24 V) and the other two are used to transmit data: a CAN bus and local asynchronous communications. The local asynchronous communications track in each connector is directly connected to the microcontroller while the other tracks are shared for all the connectors of the module.

3.3. Energy

To improve the autonomy of the resulting robots, the use of wires or tethers to provide power should be avoided. Consequently, each module should include a battery and, if the robot requires more power, expansion modules with additional batteries should be attached. However, some real tasks may imply the use of tools that require cables and hoses to feed them, such as electromagnets and, for the sake of simplicity and length of time the robot can operate, it makes a lot of sense to use external power supplies. For this reason, the architecture allows for tethered operation, making sure that the power line reaches just one of the modules and is then internally distributed among the rest of the modules. In the case of the modules shown here, power was provided at 24 V and each module uses a DC converter to reduce this voltage to 5 V required to power the servomotors and the different electronic systems it contains.

Table 1
Main features of the actuator modules.

	Slider	Telescopic	Rotational	Hinge
Type of movement	Linear	Linear	Rotational	Rotational
Stroke	189 mm	98 mm	360° (1 turn)	200°
Num nodes	3	2	2	2
Num connection faces per node	5-4-5	5-5	5-5	1-1
Weight	360 g	345 g	250 g	140 g

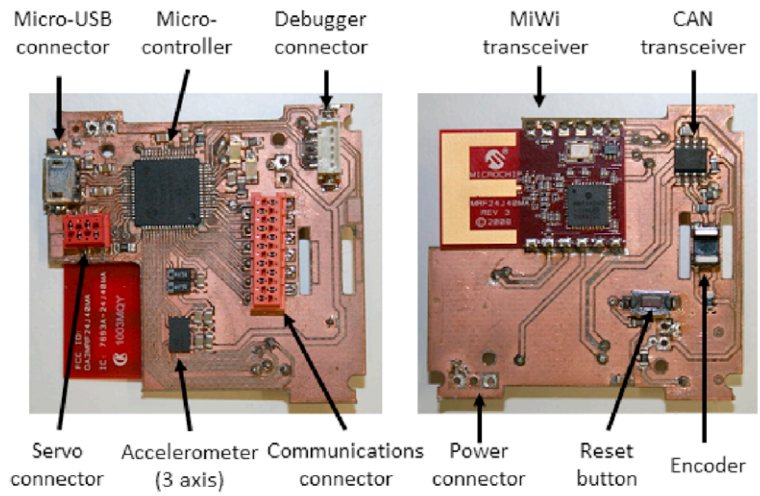

Fig. 5. Control board for the slider module and its main components.

3.4. Sensors

Every module is able to measure the position of its actuator. Thus, the linear modules have a quadrature encoder with 0.32 mm accuracy in their position. The rotational modules are servo controlled, so, in principle their position is already known. However, to improve precision a circuit has been added to sense the value of the potentiometer after applying a low pass filter. Additionally, an accelerometer has been incorporated to every module in order to provide their spatial orientation. This accelerometer, combined with the local communications between adjacent modules that is established in each attachment face, and that permit identifying the type and the face of the neighboring module, permits the determination of the morphology and attitude of the robot without any external help.

When specific sensors such as cameras, ultrasound sensors or whatever are required in order to perform a particular task (welding, inspection, measuring, etc.), they are included in specific sensor modules that are attached to one of the free bays of the actuator module that requires it.

3.5. Communications

Communications systems in modular robotics need to ensure the adequate coordination between modules as well as to respond quickly to possible changes in robot morphology. This is the reason why different communications channels are used. In this case two global and one local communications channels are considered. In terms of wired global communications, a CAN bus has been chosen as the main communications channel. This bus permits carrying out tasks requiring critical temporal coordination between remote modules. As a complement to the CAN bus and in order to consider cases of isolated modules or CAN bus saturation, a MiWi wireless communications system is implemented.

On the other hand, for local communications, that is, communications that are aimed at the transfer of information between a node and specific connection bays in its neighbors, an asynchronous local communications line has been implemented, which is mainly used for inter-module identification and to preserve some communications when the global communications systems fail. As commented above, this allows the detection of the robot's general morphology through the aggregation of the values of the local sensing elements in each module as well as the information they have on the modules they are linked to.

Finally, and for programming and debugging purposes, all the modules except the rotational one have a micro-USB connection to allow communications to an external computer. Fig. 5 shows the printed circuit board (PCB) of the slider module containing all the communications elements.

3.6. Control

In order to be able to implement the control systems within each module, they all carry their own electronics board with a micro-controller (PIC32MX575F512) and a DC/DC converter for power supply. The micro-controller is responsible of the low-level tasks of the module: controlling the actuator, managing the communications stacks and capturing the values of its sensors. As each actuator module has its own characteristics (number of connection faces, encoder type, etc.) and the available space inside the modules is very limited, a specific PCB has been developed for each kind of actuator module. As an example, Fig. 5 shows the top and bottom sides of the control board for the slider module.

In addition to the low-level tasks the control elements have to perform, this solution permits choosing the type of control to be implemented: centralized or distributed. While in a distributed control scheme, each of the modules contributes to the final behavior through the control of its own actions depending on its sensors or communications to other modules. In a centralized control scheme, one of the modules would be in charge of controlling the actions of all the other modules, with the advantage of having redundant units in case of failure. Additionally, all modules employ the CAN bus to coordinate their actions and to synchronize their clocks. Obviously, this architecture allows for any intermediate type of control scheme.

4. Some modular robots for useful tasks

The aim of this section is to show how useful robots can be easily built for different tasks by assembling the implemented modules in ad hoc configurations. In all of these cases the robots were powered through an external power supply attached by a cable to one of the modules and orders were given to them through a USB cable connected to the same module as the power supply. This module was taken as the master module and by means of messages through the CAN bus it discovers the other modules that make up the robot and provides commands to them. It can also discover the morphology of the robot by employing the local communications and the orientation of the modules given by their sensors.

As indicated in the Introduction section, two prototypical tasks in modular robots are locomotion and manipulation. Therefore, some configurations for these tasks will be shown here. Regarding manipulators, the current architecture allows us to build different morphologies like Cartesian, cylindrical and spherical configurations or even more complex morphologies such as parallel manipulators. For the sake of clarity, a small spherical manipulator has been built using five modules (see Fig. 6). This robot is fixed to the ground using a magnetic end effector module (a permanent magnet with a coil inside). This allows the robot to be fixed without employing energy and it can be unfixed when required. A rotational module, a hinge module and a telescopic module are assembled in a serial chain with a small electromagnet module as an end effector. As shown in the sequence of pictures, the robot is able to pick up a metal part from one place and carry it to a different location in around 10 s.

Another prototypical configuration in modular robots is the snake or worm morphology, a serial chain of hinge modules. This allows the robot to pass through narrow passages such as pipes. Fig. 7 displays a snake robot configuration, which is built with only two hinge modules. The control parameters allow controlling the direction of the robot (forward or backwards) and its speed. In this case, the movement takes around 8 s but it could be increased building a snake configuration with more modules.

Finally, in Fig. 8 a bipedal robot built with a more complex morphology and greater locomotion capabilities is shown. It is based on a rotational module with two hinge modules attached, each one of them with a magnetic module in its end. Thus, the robot is able to walk over ferromagnetic ground and go over obstacles.

This section has briefly shown the capabilities of the architecture in terms of being able to build useful robots with different morphologies adapted to specific tasks. This versatility is mainly achieved by using four different types of actuator modules with a high number of connection faces. The next section will show how these features can be exploited by an evolutionary algorithm to achieve well-adapted solutions for different tasks.

5. Evolving robots using this architecture

One of the design requirements of this modular architecture was its adequateness for use within evolutionary processes. Consequently, a verification of whether it meets this requirement is necessary. It can be postulated that an architecture is amenable to evolution when it allows for easier and more successful evolutionary processes. This implies being able to provide for a large number of possible variations in the resulting robots when they are evolved to solve a given task and ensuring that any of these combinations of modules produce feasible structures from an operational point of view so that the need for constraints in the fitness function is avoided.

Feasibility, as indicated in previous sections, is intrinsic to the way the modules were designed. That is, these modules are completely autonomous and they provide enough power to chain several together so that any combination of modules up to a certain size will produce feasible structures. This improves evolution as there is no need to establish structural constraints which would make it much harder.

To address the issue of evolvability, two examples of evolutionary designs based on the architecture are described in this section. In both, the EDHMOR (Evolutionary Designer of Heterogeneous MOdular Robots) system is applied [32]. It is an automatic design system that includes all the elements involved in the process of evolving robotic structures to solve a target task proposed by the designer. It was specifically adapted to be able to handle structures made up of the heterogeneous modules defined in the previous sections.

A detailed description of EDHMOR can be found in [32], only its main elements will be briefly described here. Three main blocks, algorithm, evaluation, and management make up the system. The first one is in charge of encoding the morphology and control in a chromosome to be evolved. A direct tree-like encoding of individuals is used, with nodes, links between them and control parameters in each node. The chromosomes are evolved using a constructive evolutionary strategy [32] developed to deal with the high deceptiveness of the search space, derived from tree based encoding schemes, and to cope with the different time scales involved in the evolution of morphology and control. The second block of the EDHMOR system, evaluation, includes the definition of the fitness function. It is based on an implicit evaluation methodology that allows the emergence of original solutions while preserving their feasibility by means of a realistic simulator where the physical constraints can be easily incorporated and where the main features of the environment can be properly varied during the evaluation phase. Specifically, simulation models of the modules have been created in the Gazebo 3D dynamic simulator. Finally, the third block consists in the configuration elements and the graphical user interface, which allow setting up the experiments, storing the results for statistical analysis and evaluating the robot's behavior in a graphical way [32].

5.1. Linear mission

The first evolutionary design experiment consists in obtaining a robot capable of moving through rough uneven surfaces carrying a payload. Details about the specific experimental setup are described in [32]. Here the focus will be on the evolution characteristics obtained. The environment contains only an uneven surface and the objective is to evolve a robot capable of moving through this surface the longest distance possible without dropping a payload that is placed on top of it. Regarding the robots, the only element that is common to all of them is a square base to which the modules are initially attached and that carries the payload. The fitness function is directly the traveled distance, although with a reward for those individuals that cover a minimum threshold distance and use a low number of modules.

One way of determining how evolvable the architecture is, is to carry out several runs of the evolutionary process and determine how many of those produce feasible robots that perform the task assigned, and how many of those do not achieve the objective. On the other hand, one would expect that, being evolution a stochastic process, different runs would produce very different individuals in a problem that is as openly specified as the one proposed here. In other words, there are many structures that should be able to perform the task and, if the architecture has good properties for evolution, there should be a level of variability in the solutions produced by different runs.

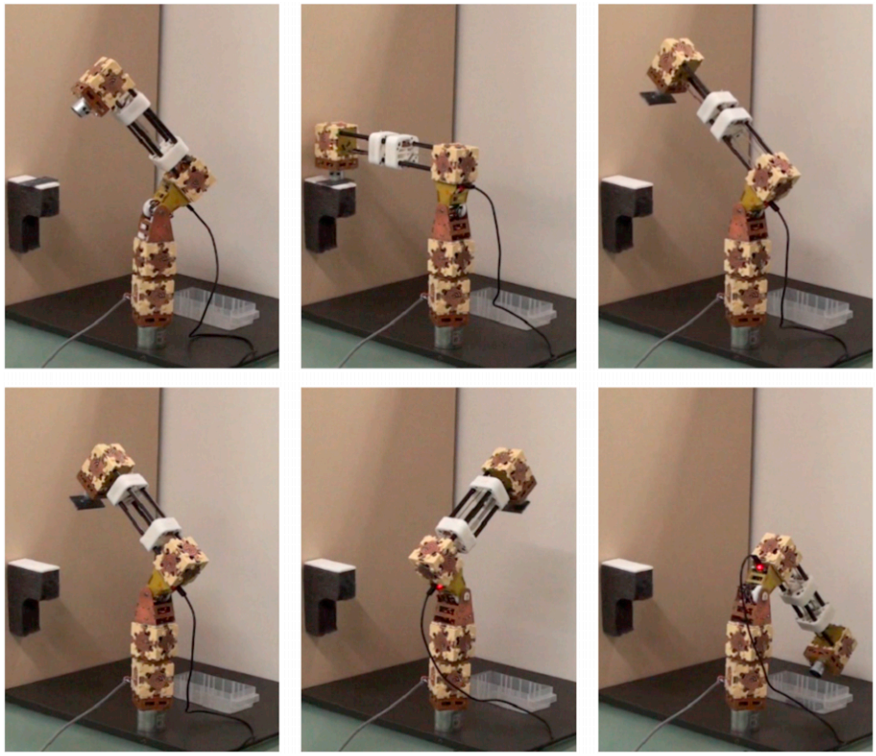

Fig. 6. Spherical manipulator made up of 3 actuator modules and two end-effectors carrying a part.

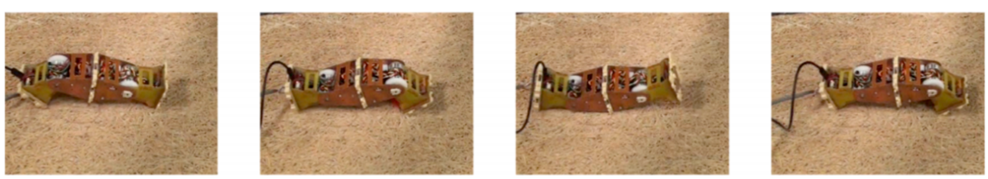

Fig. 7. A snake robot morphology built with two hinge modules.

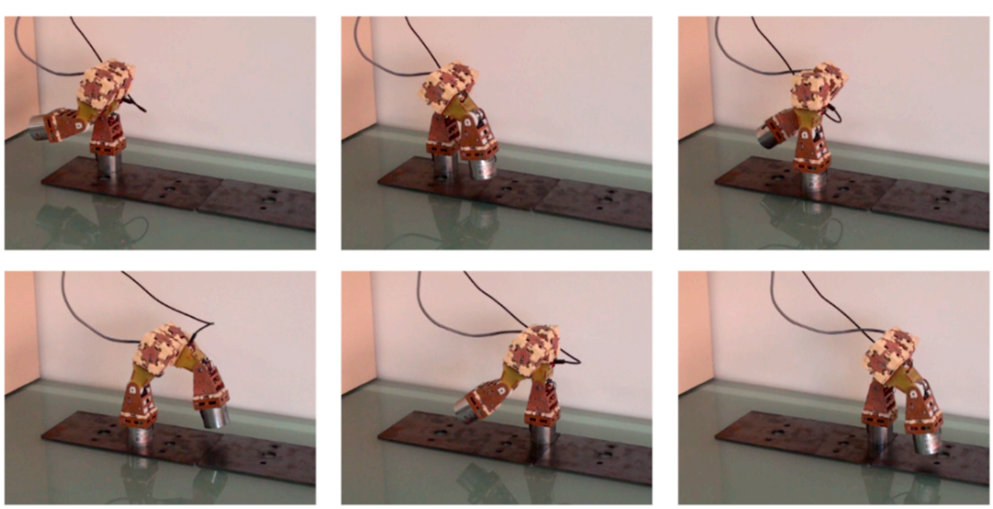

Fig. 8. A biped robot walking on a ferromagnetic surface.

In this case, the evolution process has been run 12 times and the results for the evolution of fitness are presented in Fig. 9. The top graph displays the fitness evolution for 80 generations using the EDHMOR system. There are 12 lines corresponding to the independent runs and one more corresponding to their average. It can be clearly seen that the fitness tendency is growing in all the cases and, what is more relevant here, the different runs follow different paths, that is, there is a high level of exploration of the fitness land-

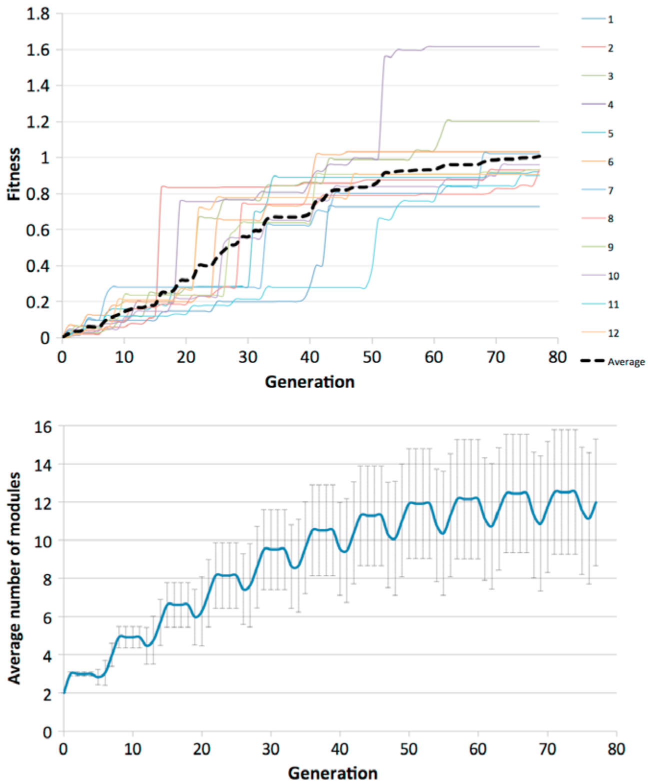

Fig. 9. Fitness evolution for 12 independent runs in the linear mission (top) and average number of modules (bottom).

scape allowed by using the architecture. This is confirmed by the bottom graph of Fig. 9, which shows the average number of modules obtained in the 12 runs and the deviation in each generation. It is clear that throughout the evolutionary process there are robots in the population with different numbers of modules and, as a consequence, with different morphologies. In fact, the best robots have about 12 modules on average, but there is a high diversity ranging from 8 to 16 module robots.

These 12 experiments have led to robots that were successful in solving the task, although most of them clearly different. As an example, Fig. 10 displays three robots obtained for three different runs and it can be observed how different they are. Obviously, if one desires less variability, a more constrained definition of the task must be provided and the number of different feasible solutions will be reduced. This was not the objective here as our aim was to show that the architecture allows for many evolutionary paths that reach feasible solutions complying with the task requirements, which is what makes it evolution friendly. The previous statement is quite important, as one could think, for instance, that a homogeneous modular architecture could also provide for many evolutionary paths. This is true, but in most cases these would not lead in 80 generations to robots that perform the task due to the fact that the evolutionary system would need to put together many more modules in order to be able to carry out the same actuations, often leading to unfeasible solutions due to power requirements, and, in general requiring many more generations.

Just to emphasize the fact that the architecture produces feasible robots, it must be pointed out that all the robots resulting from the 12 runs can be easily manufactured using the modular architecture. As an example, Fig. 11 displays a fully functional robotic structure that corresponds to the solution displayed in the middle image of Fig. 9.

5.2. Static mission

The second evolutionary design experiment consists in evolving a robot for a static mission where the robot is in a fixed position, in this case, painting a surface. Again, the specific details of the experimental setup are described in [32], and here the attention will be focused in what is relevant for the modular architecture. Fig. 13 contains screen captures of this second experiment, where the simulation environment is displayed. It consists of a small surface that is placed at different distances in front of the robot, which starts

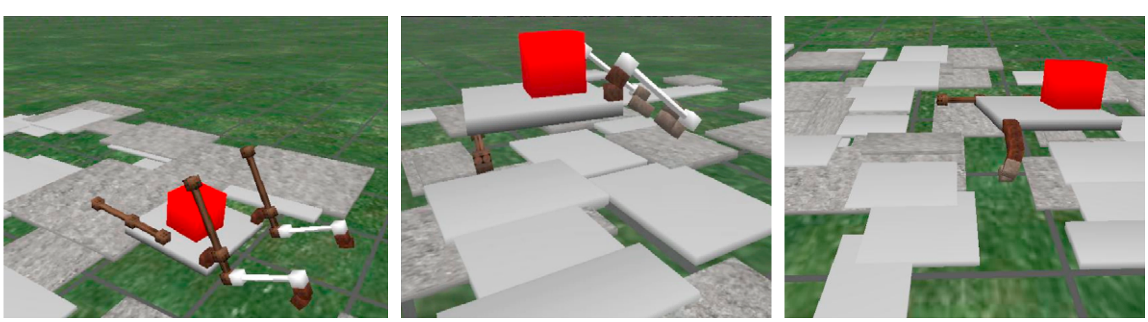

Fig. 10. Three different robots obtained using the modular architecture and the EDHMOR system.

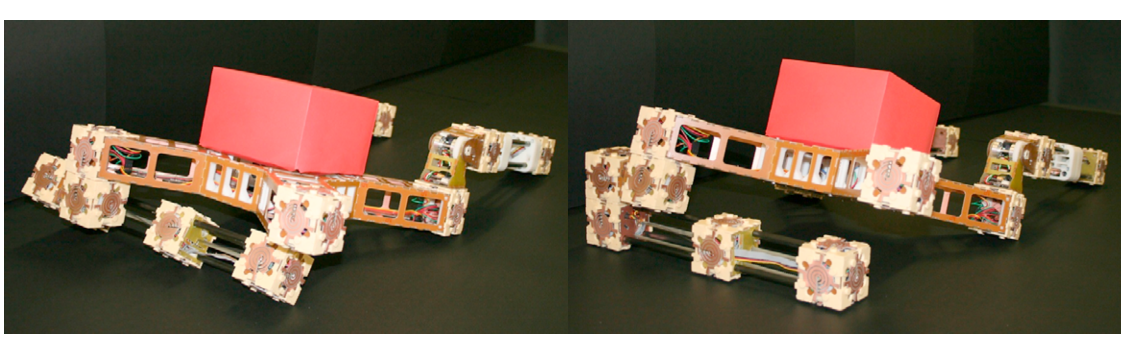

Fig. 11. Fully functional prototype robot obtained by evolution and constructed with the modular architecture.

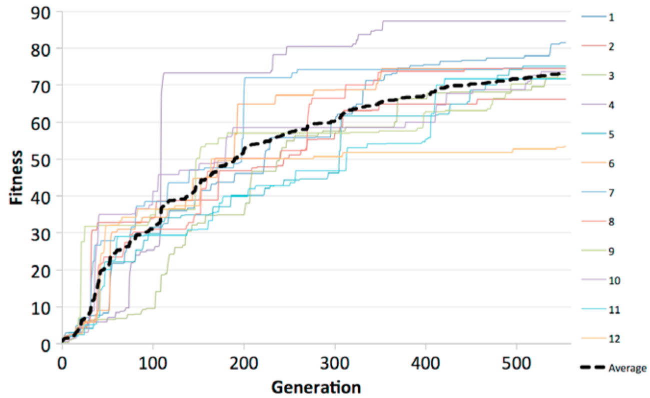

Fig. 12. Fitness evolution for 12 independent runs in the static mission.

from a passive module fixed to the floor (a stick). The objective is to evolve a robot capable of painting as much surface as possible with the minimum number of modules. As in the previous experiment, the fitness function is directly the size of the surface that has been painted, although with a reward for those individuals that paint a minimum threshold surface and use a low number of modules.

Fig. 12 shows the fitness evolution for 550 generations using the EDHMOR system. As in the previous case, it contains 12 lines corresponding to independent runs and one more corresponding to their average. Again, the fitness tendency is clearly growing in all cases and the high diversity of solutions is confirmed, because most of the runs reach different, although valid, robotic configurations. In Fig. 13, the robots resulting from three of these runs can be seen. All of them solve the task successfully, although they are morphologically different and their control structure is also different. Fig. 14 displays the prototype implementation of one of these evolved robots (rightmost one in Fig. 13), which again is fully functional and capable of painting a surface if it is provided with the adequate effector.

6. Conclusions

This work has presented a modular robotic architecture that can be employed as a basic element to build evolutionary robots. Unlike other approaches, this method guarantees feasibility and rapid deployment of the solutions. In addition, the requirements of the architecture have been analyzed taking into account the evolvability or evolution friendliness of the system. This has led to the determination that a heterogeneous architecture would be better from an evolutionary perspective due to the fact that it would, in general, lead to smaller search spaces and thus make it easier to construct platforms with complex motion patterns than homogeneous modular approaches. Additionally, and to allow for multiple evolutionary paths by introducing flexibility in the architecture, each module was endowed with a large number of connection faces. This feature increases the possible morphological diversity in the population and allows the generation of well adapted robots through mutations.

In terms of the types of modules in the architecture, a top down approach has been followed with the aim of designing the different types of modules based on the types of movements that are required in real missions with the objective of producing the smallest number of different types of modules that would cover the desired functions. Prototypical versions of these modules have been implemented.

Finally, and to demonstrate the versatility of the architecture for constructing quite different types of robots and its evolution friendliness, the modules were first used to build ad hoc morphologies for useful tasks showing the quick deployment of different

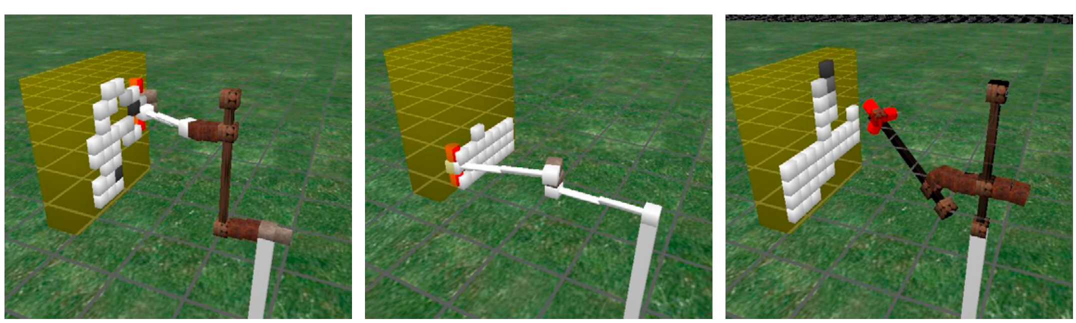

Fig. 13. Three different robots obtained using the modular architecture and the EDHMOR system in the static mission.

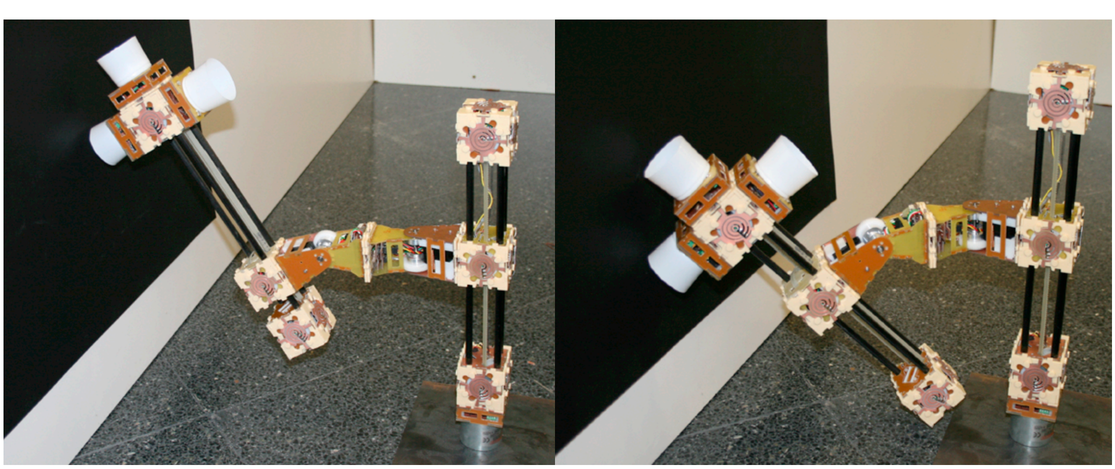

Fig. 14. Fully functional prototype robot obtained by evolution and constructed with the modular architecture.

robots. Then the morphology and control of several modular robots have been evolved for linear and static missions. These tests have shown that a large number of different robotic structures can easily be obtained that successfully carry out the proposed tasks, highlighting the evolvability of the architecture.

References

[1] R. Pfeifer, F. Iida, J. Bongard, New robotics: design principles for intelligent systems, Artif. Life 11 (1–2) (2005) 99–120.
[2] R. Pfeifer, J. Bongard, How the Body Shapes the Way We Think: A New View of Intelligence, MIT Press, 2007.
[3] R. Pfeifer, F. Iida, Morphological computation: connecting body, brain and environment, in: Creating Brain-Like Intelligence, 2005.
[4] K. Matsushita, M. Lungarella, C. Paul, Locomoting with less computation but more morphology, 2005, pp. 2008–2013.
[5] C. Paul, Morphological computation: a basis for the analysis of morphology and control requirements, Robot. Auton. Syst. 54 (8) (2006) 619–630.
[6] M. Matarić, D. Cliff, Challenges in evolving controllers for physical robots, Robot. Auton. Syst. 19 (1) (1996) 67–83.
[7] D. Floreano, F. Mondada, Evolution of homing navigation in a real mobile robot, IEEE Trans. Syst. Man Cybern. B 26 (3) (1996) 396–407.
[8] D. Floreano, L. Keller, Evolution of adaptive behaviour in robots by means of Darwinian selection, PLoS Biol. 8 (1) (2010) e1000292.
[9] K. Sims, Evolving virtual creatures, in: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques—SIGGRAPH'94, 1994, pp. 15–22.
[10] M. Komosin'ski, The Framsticks system: versatile simulator of 3D agents and their evolution, Kybernetes 32 (1–2) (2003) 156–173.
[11] N. Chaumont, R. Egli, C. Adami, Evolving virtual creatures and catapults, Artif. Life 13 (2) (2007) 139–157.
[12] M. Mazzapioda, A. Cangelosi, S. Nolfi, Evolving morphology and control: a distributed approach, in: IEEE Congress on Evolutionary Computation, No. IV, 2009, pp. 2217–2224.
[13] J.B. Pollack, H. Lipson, The GOLEM project: evolving hardware bodies and brains, in: Proceedings of the Second NASA/DoD Workshop on Evolvable Hardware, 2000, pp. 37–42.
[14] N. Cheney, R. MacCurdy, J. Clune, H. Lipson, Unshackling evolution, in: Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference—GECCO'13, 2013, p. 167.
[15] J. Hiller, H. Lipson, Automatic design and manufacture of soft robots, IEEE Trans. Robot. 28 (2) (2012) 457–466.
[16] P. Funes, J. Pollack, Evolutionary body building: adaptive physical designs for robots, Artif. Life 4 (4) (1998) 337–357.
[17] I. Macinnes, E. Di Paolo, Crawling out of the simulation?: Evolving real robot morphologies using cheap, reusable modules, 2003.
[18] G.S. Hornby, H. Lipson, J.B. Pollack, Evolution of generative design systems for modular physical robots, in: IEEE International Conference on Robotics and Automation, 2001, pp. 4146–4151.
[19] K. Stoy, D. Brandt, D.J. Christensen, Self-Reconfigurable Robots: An Introduction, Vol. 18, MIT Press, 2010, pp. 237–240.
[20] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, G.S. Chirikjian, Modular self-reconfigurable robot systems, in: IEEE Robotics and Automation Magazine, IEEE, 2007.
[21] T. Matsumaru, Design and control of the modular robot system: TOMMS, in: Proc. 1995 IEEE Int. Conf. Robot. Autom., vol. 2, 1965, pp. 2125–2131.
[22] W. Chen, E. Hui, L. Ha, I. Chen, Interactive-motion control of modular reconfigurable manipulators, no. October, 2003, pp. 1620–1625.
[23] S. Kernbach, F. Schlachter, R. Humza, J. Liedke, S. Popesku, S. Russo, T. Ranzani, L. Manfredi, C. Stefanini, R. Matthias, C. Schwarzer, B. Girault, P. Alschbach, E. Meister, O. Scholz, Heterogeneity for increasing performance and reliability of self-reconfigurable multi-robot organisms, in: Workshop on Reconfigurable Modular Robotics at IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011, pp. 1–8.
[24] I. Chen, J.W. Burdick, Determining task optimal modular robot assembly configurations, in: IEEE International Conference on Robotics and Automation, 1995, pp. 132–137.
[25] J. Han, K.W. Chung, Y. Youm, S.H. Kim, Task based design of modular robot manipulator using efficient genetic algorithms, in: IEEE International Conference on Robotics and Automation, 1997, pp. 507–512.
[26] O. Chocron, Evolutionary design of modular robotic arms, Robotica 26 (03) (2007) 323–330.
[27] S. Farritor, S. Dubowsky, On modular design of field robotic systems, Auton. Robots 10 (1) (2001) 57–65.
[28] C. Leger, Automated synthesis and optimization of robot configurations, in: Proceedings of the 1998 ASME Design Engineering Technical Conferences, 1999, pp. 13–16.
[29] O. Chocron, Evolving modular robots for rough terrain exploration, 46 (2007) 23–46.
[30] H.H. Lund, Co-evolving control and morphology with LEGO robots, in: H. Fumio, R. Pfeifer (Eds.), Morpho-Functional Machines: The New Species, Springer, Japan, 2003, pp. 59–79.
[31] A.J.I. Daniel Marbach, Co-evolution of configuration and control for homogeneous modular robots, in: Eighth Conference on Intelligent Autonomous Systems, 2004.
[32] Andrés Faíña, Francisco Bellas, Fernando López-Peña, Richard J. Duro, EDHMoR: Evolutionary designer of heterogeneous modular robots, Eng. Appl. Artif. Intell. 26 (10) (2013) 2408–2423.

Andrés Faíña received the M.S. degree in Industrial Engineering in 2006 and Ph.D. in 2011 from the University of A Coruña, Spain. He is currently working as a post-doctoral researcher in the IT University of Copenhagen. His main interests include modular and self-reconfigurable robotics, mobile robotics and electronic and mechanical design.

Francisco Bellas is a *Profesor Titular* at the University of A Coruña, Spain. He received the B.S. and M.S. degrees in Physics from the University of Santiago de Compostela, Spain, in 2001, and a Ph.D. in Computer Science from the University of A Coruña in 2003. He is a member of the Integrated Group for Engineering Research at the University of A Coruña. His current research activities are related to evolutionary algorithms applied to artificial neural networks, multiagent systems and robotics.

Félix Orjales received the M.S. degree in Industrial Engineering in 2010 from the University of León, Spain. He is currently working towards a Ph.D. degree in the Department of Industrial Engineering at the University of A Coruña. He is currently a researcher at the Integrated Group for Engineering Research. His research activities are related to electronics and unnamed aerial vehicles.

Daniel Souto received the M.S. degree in Industrial Engineering in 2007 from the University of A Coruña, Spain. He is currently working towards a Ph.D. degree in the Department of Industrial Engineering at the same University. He is currently a researcher at the Integrated Group for Engineering Research. His research activities are related to automatic design and mechanical design of robots.

Richard J. Duro received an M.S. degree in Physics from the University of Santiago de Compostela, Spain, in 1989, and a Ph.D. in Physics from the same University in 1992. He is currently a *Catedrático de Universidad* in the Department of Computer Science and Head of the Integrated Group for Engineering Research at the University of A Coruña. His research interests include higher order neural network structures, signal processing and autonomous and evolutionary robotics.