From Virtual Creatures to Feasible Robots

Andrés Faíña
IT University of Copenhagen
anfv@itu.dk

Daniel Souto
Universidade da Coruña
dsouto@udc.es

Felix Orjales
Universidade da Coruña
felix.orjales@udc.es

Francisco Bellas
Universidade da Coruña
fran@udc.es

Richard J. Duro
Universidade da Coruña
richard@udc.es

ABSTRACT

This paper provides a brief description of the robots obtained using the evolutionary design system called EDHMoR (Evolutionary Designer of Heterogeneous Modular Robots) that are displayed in the corresponding video [1]. This system is based on the coevolution of morphology and control with the main objective of obtaining feasible and manufacturable robots. To this end, a modular architecture has been defined and implemented [2], which faces real hardware issues and promotes the evolvability of the robotic structures by considering heterogeneous modules with a large number of connection faces per module. These modules constitute the building blocks the EDHMoR system uses to design the robots. Moreover, an evaluation methodology is proposed as a key element of EDMHoR, which is based on modifications in the environment that can produce more useful and realistic robots without limiting the search space. The video shows some of the resulting robots for two different tasks, painting a surface and walking, and the influence of these modifications. Finally, some real tests of these morphologies are presented.

1. INTRODUCTION

Several authors have evolved virtual creatures since Sim's work [3]. Nevertheless, most approaches are only based on simulations without taking into account physical constraints and without considering the feasibility of the solutions. Only a few of them address these issues, such as [4], by using a standard set of parts for the evolution. Similarly, the work presented here employs a modular robotic architecture to obtain feasible virtual creatures, but the architecture is designed to increase the evolvability of the system [2]. To this end, the architecture is based on four different actuator modules, each of them with a high number of connection faces. These features provide for a high level of morphological diversity in the population and they allow building well adapted robots by mutations and using a small number of modules.

On the other hand, the coevolution of morphology and control for manufacturable systems can provide unfeasible behaviors for real robots. In these cases, the designer typically restricts the morphological search space and explicitly defines a fitness function that guides evolution towards an objective while satisfying different real world constraints. This procedure involves a high degree of designer intervention that is undesirable because it is non-generalizable and it constrains the search space even more, preventing the emergence of more flexible and original solutions. To avoid these limitations, we employ a very simple fitness function to allow emergent solutions, but we introduce some modifications of the simulated environment that guide the evolutionary process towards more realistic solutions.

In this paper, we will describe the EDHMoR system and we will illustrate the importance of the application with two illustrative examples: the first one is focused on designing a moving robot and the second one on designing a robotic arm for painting. Finally, as the main objective is to obtain feasible robots, some real tests are performed.

2. EDHMOR

A detailed description of EDHMOR can be found in [5]. It is made up of three main blocks: algorithm, evaluation, and management. 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 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, 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. Regarding the control, all the modules are controlled using a sinusoidal function that provides the module position (displacement distance or angle between the two parts of the module) using the amplitude, angular velocity and phase parameters.

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.

3. WALKING TASK

This experiment has been organized as three stages of incremental complexity. The first case simply considers a flat surface where the robot has to move. In the second one, a rugged surface is contemplated. Finally, the third scenario considers the rugged surface and adds a payload to the base of the robot. The morphology is based on the modules defined by the architecture, but the initial module is always the base module. The control consists of only the phase φ of each module for this task. And, the objective of the robot to be designed is to move forward as far as possible minimizing energy consumption. Therefore, the fitness function is defined as:

$$fitness = \begin{cases} dist, & if\ dist < dist_{th} \\ dist + \frac{f_{max}}{N} \cdot (N - n), & if\ dist > dist_{th} \end{cases}$$

where $dist$ is the distance travelled by the robot, $dist_{th}$ is a distance threshold value, f_{max} is the fitness reward of a robot with zero modules, N is the maximum number of modules allowed, 16 in these experiments, and n is the current number of modules of the individual that is being evaluated. The idea behind this function is rewarding those individuals that cover a minimum distance $dist_{\text{th}}$ using a low number of modules, which is directly related with low energy consumption. The control of the number of modules is not considered until the individual reaches a minimum fitness to make the search space easier.

First, the video shows some robots obtained in a flat surface, most of them drag the base or other modules and use slider and telescopic modules to achieve the movement. This is a clear example of a robot that is feasible but not very practical if we are seeking a robot able to move in a general horizontal surface, because dragging the base would be impossible in many cases. The problem with this solution comes from the evaluation definition that does not consider any physical constraint, leading to solutions that are too “lax”.

To make things more useful, we complicated the scenario making the floor uneven or rugged, but leaving the rest of the parameters as they were in the previous one. Moreover, to obtain robots that are robust in general rugged terrain, five different floors have been designed in Gazebo and randomly used during evaluation. Some robots obtained in this case are shown in the video. As it can be observed, all of them raise the base from the ground, at least a few moments, in order to pass over the small irregularities of the ground. Nevertheless, some of them still drag the rear part of the base to provide stability. A very interesting solution is the last one, where motion over the rugged terrain is achieved by rotating the base, very much like a wheel. This solution is quite optimal for dealing with rugged surfaces and reinforces the idea of how a more flexible search space leads to more original solutions. To force the base to remain basically horizontal, we have changed the goal with respect to the previous case by making the robots have to carry a payload of unspecified weight without dropping it. The fitness function remains unchanged but the *dist* parameter is now the distance travelled by the payload. If the payload falls, the simulation is stopped and the final position of the payload is employed to calculate the travelled distance. In addition to the five rugged floors used in the previous setup, we include five different cubic payloads of different weights, sizes and friction coefficients to obtain robots that are robust with respect to the type of payload. The robots shown in the video correspond to some of the solutions obtained in this case. They can solve the task robustly by raising the base module to pass over the obstacles, but now the base is usually reasonably horizontal.

4. PAINTING TASK

This example has been organized in two stages of increasing complexity: the first only considers one large flat surface to paint placed at a fixed distance from the base of the robot. The last one limits the area where painting is allowed and seeks a robot to paint surfaces placed at different distances from its base. The morphology is based on the modules defined by the architecture, but the initial module is always a fixed base with only one face placed horizontally. The control consists in three parameters for each module: amplitude (α), angular velocity (ω) and phase(φ).

To calculate the fitness of the different solutions, we divide the surface into a matrix of (10x10cm) tiles, where $f_{ij} \in [0,1]$ represents the fitness for a tile. The painted portions of the surface, *paint*, and the *fitness* can be computed as:

$$paint = \sum_{i=0}^{R} \sum_{j=0}^{S} f_{ij}$$

$$fitness = \begin{cases} paint, & if\ paint < paint_{th} \\ paint + \frac{f_{max}}{N} \cdot (N - n), & if\ paint \geq paint_{th} \end{cases}$$

where $paint_{th}$ is a painted tiles threshold value, f_{max} is the fitness reward for a robot with zero modules, N is the maximum number of modules allowed (16 in this experiment) and n is the current number of modules of the individual that is being evaluated.

The first stage seeks to produce robots that paint as much of a large flat surface, placed 1.1m away from their base, as possible This large surface can be taken as infinite by the robot. Some of the robots obtained in this test are displayed in the video. Most of them employ a rotational module, which generates a circular motion of the rest of the robot branches. We have to indicate that the evolved robots exploit their morphologies to avoid the excess of momentum in the joints of the modules. An example of this can be seen in the second and the third robots shown in the video. These robots have two opposing branches allowing the rotational module, which has a limited torque, to move the rest of the structure.

In the second stage, the surface to paint is smaller (1x1m) and we generate two different worlds at different random distances between the base of the manipulator and the objective surface to obtain robots that are robust with respect to the distance to the surface. To this end, the robots are evaluated twice, once in each world, and the resulting fitness is the minimum obtained by the robot. Furthermore, as the robot must modify the way it controls its actions for each distance to the wall, we introduce two additional control parameters: an amplitude modulator (β) and an angular velocity modulator (ρ). These two parameters, combined with the distance to the surface, and the other three control parameters generate different behaviors for each distance to the wall. More details about this controller can be found in [3].

As shown in the video, this simple strategy allows finding new robot morphologies that can paint different surfaces without bare patches and that can adapt their end effectors to paint surfaces at different distances. Most of the robots obtained are based on sliders or telescopic modules to place the end effectors at different distances. A good number of them of them can even paint surfaces placed at distances not contemplated during evolution.

5. SOME REAL TESTS

As pointed out above, the main objective of this system is for the robots that are designed to be easily manufactured. To show that this is direct, different examples of prototype implementations have also been included in the video. Currently, we have only built a small number of modules and only simple morphologies can be tested, some of them with slight variations. Nevertheless, all of them can be easily assembled and they work.

6. REFERENCES

[1] http://vimeo.com/afaina/feasible-virtual-creatures

[2] A. Faiña, F. Orjales, F. Bellas, R. J. Duro, “First Steps towards a Heterogeneous Modular Robotic Architecture for Intelligent Industrial Operation”, Workshop on Reconfigurable Modular Robotics at the IROS 2011.

[3] T. Taylor, C. Massey, “Recent developments in the evolution of morphologies and controllers for physically simulated creatures”. Artificial Life, 7(1), 2001, pp.77-87.

[4] D. Marbach, A. J. Ijspeert, “Online optimization of modular robot locomotion” IEEE International Conference in Mechatronics and Automation, Vol. 1, 2005, pp. 248-253.

[5] A. Faíña, F. Bellas, F. López-Peña, R. J. Duro, “EDHMoR: Evolutionary designer of heterogeneous modular robots”, *Engineering Applications of Artificial Intelligence,* Volume 26, Issue 10, 2013, pp 2408-24.