Real-World Reproduction of Evolved Robot Morphologies: Automated Categorization and Evaluation

Eivind Samuelsen and Kyrre Glette

Department of Informatics, University of Oslo, Norway
{eivinsam,kyrrehg}@ifi.uio.no

Abstract. This paper describes the real-world reproduction of a handful of robots selected from a larger sample of simulated models previously generated by an evolutionary algorithm. The five robots, which are selected by automatic clustering to be representative of different morphological niches present in the sample, are constructed in the real world using off-the-shelf motor components, combined with 3D printed structural parts that were automatically generated based on the simulator models. A lab setup, involving evolution of turning gaits for each robot, is used to automate the experiments. The forward walking speeds of the constructed robots are measured, and compared with the simulated speeds. While some of the robots achieve near-identical results, some show a large performance loss compared to their simulated prototypes, underlining the reality gap issue seen in similar previous works.

1 Introduction

There is a high need for autonomously operating robotic systems in remote, hostile, or otherwise isolated environments, such as remote planets, disaster areas, deep mines, or subsea installations. At the same time, human intervention is difficult, time-consuming, costly, or at worst impossible, and thus robots which are able to automatically repair or adapt themselves to new situations would be a great advantage. With the recent and frequent advances in 3D printing technology, such as an increasing number of materials, higher speeds, and portability, new possibilities open up for the design or repair of robotic systems. For example, one could imagine a team of robots, including a mobile 3D printer, capable of repairing or producing new robot morphologies in situ [1].

Evolutionary robotics (ER) approaches the challenge of automatic design and adaptation of robotic systems through the use of evolutionary algorithms. While ER research has mainly concentrated on optimization of robotic control systems, e.g. for legged robots [2], using software simulations it is also possible to address the challenge of simultaneously optimizing robot morphology and control [3].

Automated robot design without a fixed topology introduces an encoding challenge, as more complex data structures are needed to describe the space of possible solutions. Thus, ER research has produced a wide variety of coding schemes for describing morphology. Examples include directly encoded module-based [4] or graph-based [3] approaches, and an approach using LEGO bricks and servo motors as modular building blocks [5]. Generative and developmental methods, where complex phenotypes are generated from simpler genotypes through some sort of indirect encoding, have been successfully applied to evolve morphology, for example by using cyclic graphs [6], gene regulatory network-inspired encodings [7], and scalar-field generating methods based on compositional pattern producing networks both with rigid [8] and soft [9] bodies. These generative methods can produce highly complex and structured results from very terse information [10] and are known to outperform direct encoding in simple morphology optimization tasks [11]. Robots designed by some of these methods have also been reproduced in real life [3,10].

ER in general faces a reality gap challenge, that is, controllers evolved in simulation often perform much worse on the real robot [12]. This is due to inaccuracies in simulation combined with evolution's tendency to produce overfit solutions. Thus, one may end up with solutions which exploit features nonexistent in the real world. While this is already a significant challenge for controllers transferred to robust engineer-designed robots, the additional freedom in the morphological dimension only increases the potential size of a reality gap. There are only a few examples of evolved robot morphologies and control systems resulting in physical, actuated robots [3,5,10], and the main challenges for these have been a lack of complexity or regularity in the structures, or low real life performance due to reality gap issues. In order for morphology-based ER to be a convincing approach, there is a need for demonstration of robust, high-performing approaches and more thorough performance evaluations.

In this paper we pursue this goal by building on our previously proposed generative encoding scheme for evolution of robot morphology and control [13] to present a full construction process for physical robots from the simulation results. The approach represents a flexible way to create relatively lightweight and regular robots of controllable complexity, and thus should have a potential of resulting in well-performing robots. With the help of a clustering algorithm working on morphological distance, we pick 5 different robots representative of the different morphologies evolved in simulation, construct them, and demonstrate a thorough way of evaluating their real-world performance. The evaluation setup is able to automatically perform a high number of evaluations and is thus suitable for future real-world learning experiments and investigations into transferability from simulation to the real world.

The remainder of this paper is organized as follows: In Section 2, the methods used to evolve, select, construct, and test the robots are presented. Section 3 presents the resulting robots and their performance. Section 4 discusses the results, before a conclusion and pointers for future work are given in Section 5.

2 Methods

This section describes the methods used to perform the experiments in this paper. First, we describe the previous results these experiments are based on, before we present the selection method used to pick representative robots, and details about the hardware reproduction. Then we describe the hardware testing methods and performance evaluation procedures.

2.1 Evolving Robots Through Simulation

A large number of robots were evolved in a simulated environment in [14]. The evolutionary algorithm uses a genotype inspired by the high-level genetic coding used in nature to specify the shape of different body segments to encode robots with a various number of symmetric limbs around a central spine structure. All sets of limbs are coded with the same program, but variation based on a number of evolved parameters that are specific to each section along the spine.

The robots were evaluated with three objective functions: maximizing forward movement in a simulated environment, minimizing the estimated weight of the robot, and maximizing diversity. The weight of the simulation robot model was used as the weight estimate. Diversity was measured as the mean morphological distance to all other robots in both parent and offspring generations.

The simulation was done using the PhysX simulation engine. The robot was first simulated for 1 s in order to let it settle on the ground, before the robot's position was reset. The evaluation was then done by letting the robot move freely in the environment for 7 s and then measuring the displacement of the "head" of the robot in a predefined forward direction, so that the robots most proficient in moving in that direction would survive. A custom set of joint constraints were used to simulate the motors in the robot joints. Additionally, the control system was designed with a torque limit limiting the actual applied torque to 1 Nm. The environment itself consisted of an infinite ground plane and long, low obstacles at regular intervals in the forward direction. Parameters for the simulation and environment are summarized in Table 1.

2.2 Selecting Representative Robots

The experiments in [14] produced 42000 solutions across 210 runs, most of which had differing morphologies to some extent. Only five of these could be printed given the available time and resources; selecting these five purely at random would most likely give a poor sample of the different morphologies, so a method was used in order to select robots who are each representative of a larger group of robots:

Evolutionary runs with the same control system and diversity measure were selected, so that they differed only in the random seed used. The distance measure was an approximation of the graph edit distance (GED) introduced in the same article. From these runs, 12 were arbitrarily selected, maintaining some of the variety between runs while reducing the number of candidates further to

Table 1. Simulation parameters

General	PhysX version	3.3 beta-2		
	Timestep	$^1/_{128}$ s		
Friction	Env. static	0.20	Robot static	0.30
	Env. dynamic	0.15	Robot dynamic	0.30
	Env. restitution	0.40	Robot restitution	0.30
Motor	Static friction	0.15 Nm	Dynamic friction	$\frac{1.65\,\text{Nm}}{97\,\text{rpm}}$
	Appliable torque	1.8 Nm		
Obstacle	Width	0.02 m	Height	0.02 m
	Length	0.02 m	Spacing	0.5 m

2400. Candidates with a forward movement performance lower than a certain threshold value were then filtered out, reducing the number of candidates further down to 949.

The remaining candidates were then grouped into six clusters using hierarchical clustering. The distance measure used for clustering was the same as the one used in the diversity measure they were evolved with. The additional sixth cluster was a "trap cluster" intended to collect a specific kind of robot that was known to appear in the runs, with only one set of limbs and no tail, which moved by rolling around. Although very interesting, this locomotion method was impossible to reconcile with the laboratory setup described in Subsection 2.5.

2.3 Constructing the Robots

From the beginning, the genetic encoding used in [14] was designed to produce robots that are possible to construct in the real world. The scale of the robots and the joint simulation parameters were designed to fit the modular servo motors Dynamixel AX series produced by Robotis. Specifically, the simulated joint parameters, as well as the actual servo motors used on the constructed robots were of the AX-18 model.

Each link in the produced robots consists of one variable-length capsule combined with sockets to attach the servo motors for each joint. In order to minimize weight the capsules were made hollow, with a thickness of 1.5 mm, and evenly spaced holes throughout in order to keep wires inside the capsule and also to make it easier to clean away the support material used by the printer.

The sockets are hand-designed in a CAD program to fit the servos correctly, including matching holes for easy assembly. In order to attach the sockets to the central capsule, which must be custom-generated for each link, the fact that intersecting meshes will be merged into one solid object implicitly by the 3D printer was exploited. Thus one only needs to have the capsule and socket meshes positioned correctly and contained in the same data file. To attach side sockets, which can have an arbitrary rotation relative to the capsule, an additional set of four supporting bars were generated that each aligned with one end of a socket in each end and with the surface of the capsule in the middle.

The parts were printed in an Objet Connex 500 multi-material 3D printer, using the DurusIvory digital material, a predefined mixture of VeroWhitePlus (a rigid but somewhat brittle) and DurusWhite (a softer, polypropylene-like material). It was soon discovered that this material, and in fact, most plastic materials, had significantly lower friction against the floor carpet in the leg than the friction values modeled in simulation. In order to increase the friction between robot and floor, hockey tape was added to the endpoints of each limb on the robots.

2.4 Hardware Testing Setup

In order to perform measurements with the hardware efficiently, the experiments were automated by programming the robots to turn back when they go out of bounds; when the robot reaches a certain distance from the center of the floor, the control system is replaced with one that makes the robot turn either left or right until it is approximately headed for the center of the floor. When the robot has finished turning, it is then switched back to its original control system and the evaluation resumes by restarting the unfinished period. In order to avoid the measurements being affected by the turning motion, the first period after resuming evaluation is discarded.

For these experiments, turning control systems were generated by running an evolutionary algorithm with two objective functions: maximizing turning velocity and minimizing positional displacement. The population was seeded with 60 mutations of the original forward movement control system and run for 60 generations. The evolution used uniform Gaussian mutation and whole arithmetic recombination. All the evolved turning systems worked satisfactory in real hardware, except for those for robot 2, who had too large positional displacement. This was remedied by further parameter tuning on the real robot.

The robots contain no power source, so they need to be wired to a power source and controller located elsewhere. In order to let the robot move freely without tripping in the wire, the wire was put through a pulley in the center of the ceiling to a winch that is programmed to let out approximately enough wire to let the robot move freely based on the robots position relative to the center of the floor.

The position and orientation measurements were done using a motion capture system. This enables very accurate measurements with minimal modification of the robots. Three reflex balls were attached to one of the central parts of the robot, enabling the motion capture system to identify the robot as a rigid body with a position and orientation. The motion capture system used was a NaturalPoint OptiTrack with 8 infrared Flex 3 cameras. The cameras were positioned in the lab ceiling, at the corners and midpoints of the walls.

2.5 Performance Evaluation

In order to provide a fair comparison of the simulated performance of the robots and the performance in hardware, the same evaluation procedure is used in both cases. Since the robots were originally evolved to maximize forward movement, this has been used as the performance metric in these experiments as well.

At the end of each period the displacement of the robot during the period, in the direction it was heading at the start of the period, is measured. The period length was set to 1 s to align with the control system cycle length. After every fourth period the mean displacement per second over the last four periods is stored as a single evaluation. 50 evaluations were done on each robot both in simulation and in hardware in order to get a good statistical sample of the performance.

3 Results

This section will present the results of the experiments described in the section above. First, a description of the selected robots is given in the subsection below, before the performance measurements are reported in Subsection 3.2.

3.1 The Selected Robots

The twelve runs selected had a total of 2400 candidate robots. After removing the robots that did not meet the performance criteria, this was reduced to 949 robots. The distribution of these in the six clusters are shown in Table 2. As expected, one cluster contained only robots with rolling behavior. One other cluster (number 1) contained mainly rolling robots as well, but this cluster contained two distinct sub-groups, where a second, smaller group of around 19 robots had a tail one or two sections long that hindered the rolling motion, and which had developed a dragging gait instead.

The distribution of the clusters in objective space is shown in Fig. 1, showing that the clusters roughly match up with different trade-offs between forward

Table 2. The morphological clusters.

Cluster	Robots	Description
1	138	Two limbs with two joints each, mostly rolling behavior
2	86	Two active limbs with one joint each, midsection either limbless or with nonfunctional limbs
3	254	Four limbs with two joints each
4	4	Four limbs with two joints each plus limbless waist section
5	458	Six limbs with two joints each
(6)	9	Two limbs with one joint each, no tail, typically with rolling behavior

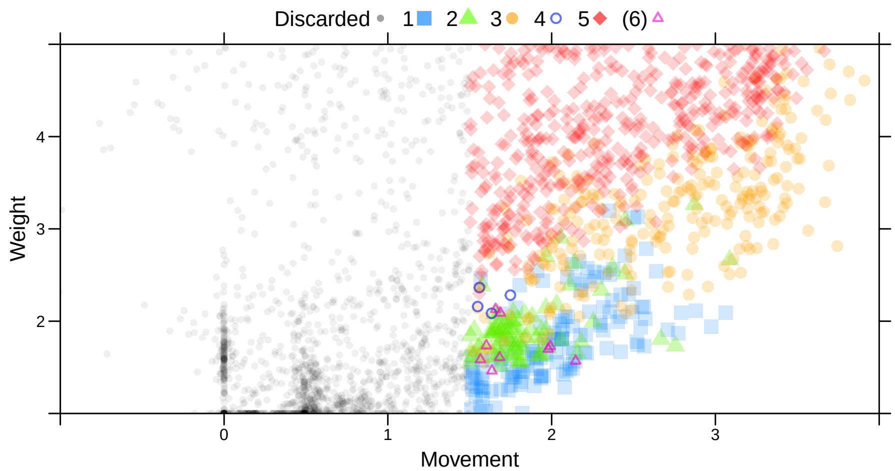

Fig. 1. Objective-space distribution of the clusters. The diversity dimension is omitted, but generally speaking solutions far from the movement-weight pareto front will have a higher diversity score. The small dots are the individuals that were discarded before clustering because of low fitness. (In grayscale, cluster 2 is difficult to see, but occupies most of the area in between clusters 4 and 6.)

movement ability, weight and diversity. It can be seen that cluster 5, while largest in number of individuals, consists mostly of robots that survived due to having distinct morphologies rather than being light or efficient.

In three of the five clusters (clusters 2, 4 and 5) the centroid robot was picked as its representative. In cluster number 1 a representative of the dragging subgroup was selected by hand. In cluster 3 a representative with a more interesting gait than the centroid, but close to it (rank 13 in closeness to the centroid), was chosen to improve the gait variation between the representatives. Body schematics for the five selected robots and photographs of the finished robots are shown in Fig. 3.

Robot 1 has two limbs and a tail, each with two joints, and drags itself forward with the limbs. Robot 2 had four limbs and two joints in the waist, but only one joint in each limb. It achieves a trotting gait (neighboring limbs are in opposite phase) by having a large synchronized swing in the waist joints. Robots 3 and 4 both have four limbs with two joints per limb. Robot 3, which has one waist joint, exhibits a bounding gait (front and rear limbs are out of phase and the robot makes a small leap each time the back limbs pushes the robot forward). Robot 4 has two waist joints and a pronk-like gait (all limbs are approximately in sync).

Robot 5 has six limbs, with two joints in each limb. The two rear limbs are considerably larger than the other two pairs. This robot has a bounding-like gait, where each limb pair is in sync; the front and middle limbs have a small phase difference between them, and the rear limbs are approximately in opposite

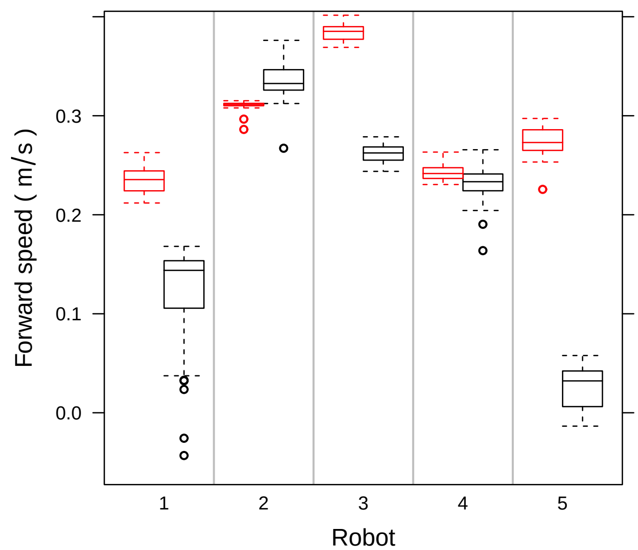

Fig. 2. Performance box plot of the five robots. The left column of each robot is the simulated measurements and the right box is the measurements in hardware. Each of the boxes represents 50 measurements.

Table 3. Mean performance summary.

Robot	1	2	3	4	5
Simulation (m/s)	0.235	0.311	0.384	0.242	0.274
Hardware (m/s)	0.120	0.335	0.262	0.231	0.026
Ratio	0.510	1.079	0.682	0.955	0.094

phase of the other pairs. The limbs touch the ground in order from front to back. As the middle limbs move backwards the rear limbs lift above the middle limbs, touching them near the knee joints during maximum forward extension of the rear limbs.

3.2 Performance

The distribution of the performance measurements is shown in Fig. 2. Table 3 show a summary of the mean performance for each robot. Robots 2 and 4 have overlapping performance ranges between simulation and hardware and mean performance ratio close to one, while the other three robots have significantly lower performance in hardware. Robot 5 has the largest performance loss: the mean performance in hardware is only 9.4% of the mean simulated performance. Second worst is robot 1 with a 49% performance loss, while robot 3 has 31.8% lower performance in hardware.

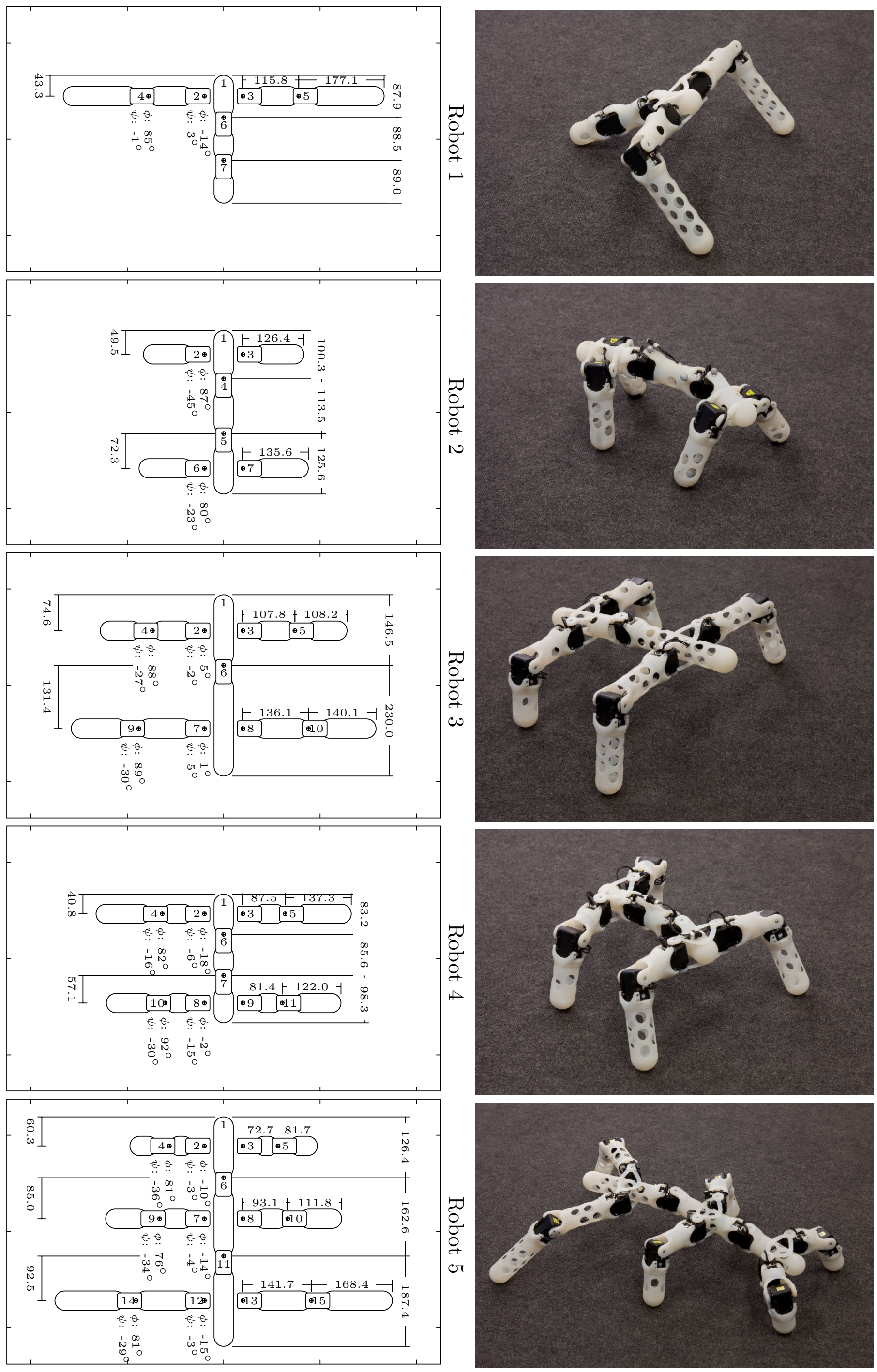

Fig. 3. Schematic view and photographs of the robots. All lengths are in millimeters.

4 Discussion

The hierarchical clustering produced clusters that seemed to match up reasonably well with what was expected, based on randomly sampling the populations by hand. It also managed to find a rare but interesting robot class, represented by robot 4, with a double-jointed waist, of which there were only four in the entire set of 949 robots that met the performance criteria.

While the structural strength of the printed parts generally proved to be sufficient for these experiments, in one instance (robot 4), one of the spine parts broke at the intersection between the capsule shape and the bars attaching the limbs. A new, stronger design was created and printed in order to complete the experiments.

The performance in hardware relative to simulation varies from robot to robot, with some robots having about the same performance, while others perform considerably worse in hardware. These results are in agreement with similar experiments such as [3] and [5].

The robots that had the lowest loss of performance in this experiment were also the least risky designs. Both the dragging motion of robot 1 and the bounding motion of robots 3 and 5 are dependent on friction being simulated correctly, and in this case they were evolved in an environment with larger friction than in the real-world lab. Robots 2 and 4, on the other hand, employed gaits that are less sensitive to changes in friction since they do not rely on pushing forward against the ground, but rather on alternating between having parts resting on the ground and moving them forward. One simple measure to avoid friction problems might be to underestimate friction in simulation, in order to give a conservative estimate of the performance of friction-dependent gaits.

One other cause for the simulation-hardware performance loss is limb collisions. Some of the robot gaits caused limbs to collide, which might not be modeled well enough in simulation. Robot 5 in particular had a gait that caused collision or at least near collision once every period. From a performance perspective, there seems to be little reason for gaits to cause self-collisions, so one could perhaps introduce some form of collision discouragement in evolution to mitigate this. Using a closed-loop control system might also help, because the control system would be better able to correct itself post-collision.

5 Conclusion & Future Work

This paper has documented the transferral of robots, that were automatically designed in simulation, into real hardware. From a large pool of robot designs drawn from previous experiments, five robots were constructed, each acting as a representative for a larger group of robots with similar morphologies. The representatives were found using a clustering algorithm, and then a functioning mechanical body was created from the design using modular servo motors combined with a combination of hand-designed and automatically generated plastic parts manufactured using a 3D-printer.

A lab setup was created that enabled thorough performance measurements both in simulation and hardware using the same measure. Measurements in the lab were compared to similar measurements in simulation, demonstrating that, at least for some of the robots, forward walking speeds in the same range as the well-studied hand-designed AIBO robot (see for example [15,2]), were achieved. However, there was large variation in how well the real-life reproduction performed compared to the simulated model.

This confirms that some sort of measure is needed in order to make the products of simulated evolution correlate better with real hardware in terms of performance. On the simulation side, noise may be added to model external effects. Forces such as friction, where the conditions are expected to vary between environments in the real world anyway, should perhaps be underestimated in order to avoid having robots exploit effects that are not certain to appear in reality. On the hardware side, it might be possible to implement some sort of adaption technique on the control system in order to regain some of the performance seen in simulation.

The transferability approach [16] is a promising method for dealing with the reality gap in the evolution of controllers for fixed-morphology robots, by building a transferability model based on real-world samples. For such a method to work also with evolved morphologies, there would be a need for selecting a limited but representable set of morphologies for real-world sampling. As such, the methods presented in this paper could be a possible approach, and this application should be investigated in future work.

References

1. Revzen, S., Bhoite, M., Macasieb, A., Yim, M.: Structure synthesis on-the-fly in a modular robot. In: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, IEEE (2011) 4797–4802
2. Hornby, G., Takamura, S., Yokono, J., Hanagata, O., Yamamoto, T., Fujita, M.: Evolving robust gaits with aibo. In: Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on. Volume 3. (2000) 3040 –3045 vol.3
3. Lipson, H., Pollack, J.: Automatic design and manufacture of robotic lifeforms. Nature **406** (2000) 974–978
4. Leger, C.: Automated synthesis and optimization of robot configurations: an evolutionary approach. PhD thesis, Carnegie Mellon University (1999)
5. Macinnes, I., Di Paolo, E.: Crawling out of the simulation: Evolving real robot morphologies using cheap reusable modules. In Pollack, J., Bedau, M., Husbands, P., Ikegami, T., Watson, R., eds.: Artificial Life IX: Proceedings of the Ninth Interational Conference on the Simulation and Synthesis of Life, Cambridge, Massachusetts, MIT Press (2004) 94–99
6. Sims, K.: Evolving virtual creatures. In: Proceedings of the 21st annual conference on Computer graphics and interactive techniques. SIGGRAPH '94 (1994) 15–22
7. Bongard, J.: Evolving modular genetic regulatory networks. In: Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on. Volume 2., IEEE (2002) 1872–1877

8. Auerbach, J.E., Bongard, J.C.: Environmental influence on the evolution of morphological complexity in machines. PLoS computational biology **10**(1) (2014) e1003399
9. Hiller, J., Lipson, H.: Automatic design and manufacture of soft robots. Robotics, IEEE Transactions on **28**(2) (2012) 457–466
10. Hornby, G.S., Lipson, H., Pollack, J.B.: Generative representations for the automated design of modular physical robots. Robotics and Automation, IEEE Transactions on **19**(4) (2003) 703–719
11. Komosinski, M., Rotaru-Varga, A.: Comparison of different genotype encodings for simulated three-dimensional agents. Artificial Life **7**(4) (2001) 395–418
12. Jakobi, N., Husbands, P., Harvey, I.: Noise and the reality gap: The use of simulation in evolutionary robotics. In: Advances in Artificial Life: Proc. 3rd European Conference on Artificial Life, Springer-Verlag (1995) 704–720
13. Samuelsen, E., Glette, K., Torresen, J.: A hox gene inspired generative approach to evolving robot morphology. In: Proceedings of the 15th annual conference on Genetic and evolutionary computation, ACM (2013) 751–758
14. Samuelsen, E., Glette, K.: Some distance measures for morphological diversification in generative evolutionary robotics. In: Proceedings of the 16th annual conference on Genetic and evolutionary computation, ACM (2014) 721–728
15. Röfer, T.: Evolutionary gait-optimization using a fitness function based on proprioception. In: RoboCup 2004: Robot Soccer World Cup VIII. Springer (2005) 310–322
16. Koos, S., Mouret, J.B., Doncieux, S.: The transferability approach: Crossing the reality gap in evolutionary robotics. Evolutionary Computation, IEEE Transactions on **17**(1) (Feb 2013) 122–145