Evolutionary Machine Learning in Robotics

Eric Medvet, Giorgia Nadizar, Federico Pigozzi, and Erica Salvato

University of Trieste, Trieste, Italy
emedvet@units.it, {giorgia.nadizar, federico.pigozzi}@phd.units.it,
erica.salvato@dia.units.it

Abstract. In this chapter, we survey the most significant applications of EML to robotics. We first highlight the salient characteristics of the field in terms of what can be optimized and with what aims and constraints. Then we survey the large literature concerning the optimization, by the means of evolutionary computation, of artificial neural networks, traditionally considered a form of machine learning, used for controlling the robots: for easing the comprehension, we categorize the various approaches along different axes, as, e.g., the robotic task, the representation of the solutions, the evolutionary algorithm being employed. We then survey the many usages of evolutionary computation for optimizing the morphology of the robots, including those that tackle the challenging task of optimizing the morphology and the controller at the same time. Finally, we discuss the reality gap problem, that consists in a potential mismatch between the quality of solutions found in simulations and their quality observed in reality.

1 Robot optimization and its peculiarities

The field of robotics involves the design, construction, and operation of *robots*. In this chapter, we define as robot any agent, be it real or simulated, which can interact with an *environment*. We require said interaction to be bidirectional, meaning that the agent can affect the environment with its actions, and is in turn affected by the environment, either through sensory perceptions or simple mechanical effects.

A robot is defined by its body and its controller, which are deeply interconnected. The *controller* (often called brain) is responsible for making decisions concerning the actions to be made. Oftentimes, such decisions consist of computing control values which are sent to the body, which then actuates them. The *body* of the agent involves all the physical aspects of the robot, such as its external aspect, its constituting materials, or its modules and connecting parts (joints). In addition, the body is responsible for actuating actions, and also for proprioception and environmental awareness.

In general, the actions of a robot are driven towards the achievement of a goal, which we define as the *task* of the agent. Within this paradigm, it is possible, and often desirable, to *optimize* the agent for a task, that is, achieving a design that enables the robot to most successfully accomplish its goal. However, due to the complexity of the design process, given the usually extremely vast search space, handcrafting satisfying solutions is practically unfeasible. Therefore, automatic optimization techniques play a fundamental role in assisting designers in the achievement of successful robots. Among those, evolutionary algorithms (EAs) stand out for their effectiveness and efficiency in exploring the solution space, yielding to extremely successful results with relatively little human effort. In particular, in most cases, they only require to define a suitable measure of quality (i.e., the *fitness*) of the solution under optimization (i.e., the robot): while defining such fitness is not always easy [37], it is in general much easier than attempting to manually design the solution.

The application of evolutionary computation (EC) to robotics constitutes the field of *evolutionary robotics* (ER) [39, 132]: in a broad sense, ER consists in optimizing the robot (or some of its parts) for a given task, using EC. In this chapter, we deal with EML in robotics, which can be considered the subfield of ER where ML is somehow involved. As a matter of fact, the vast majority of EML applications to robotics deal with the optimization of a robot controller which is based on an artificial neural network (ANN): indeed, ANNs are traditionally considered an artifact belonging to the field of machine learning (ML). As a consequence, the largest part of this chapter, namely Section 2, is devoted to surveying the body of literature dealing with evolutionary optimization of ANN-based controllers for robotic agents. There are, however, other artifacts that can be employed as controllers and can be optimized with EC: for example, behavioral trees can be evolved using genetic programming. From the point of view of the definition given in this book, they can be considered at the boundary of EML. We survey some of these approaches in Section 2.2.

From a broader point of view, robotics is a field that exhibits a few peculiarities that are relevant to the usage of EC as a form of optimization.

First, optimization can be performed at various levels, targeting different features of a robot. Namely, most applications of EML in robotics are either aimed at optimizing the controller, the body of the agent, or both, but there are also some more peculiar examples of EML in robotics, as we will see in Section 4. It is worth to note that, even when just the controller is subjected to optimization, the body still plays a relevant role, because it actively takes part in the way the robot interacts with the environment by processing perception to decide actions to be performed; in other words, the body is capable of performing some morphological computation [66]. This phenomenon is captured by the embodied cognition paradigm [143].

Second, robots are not static artifacts, but change over time. In the simplest case, the only thing that changes is their spatial configuration, i.e., their position within the environment or the relative position of their components. In more complex scenarios, they can suffer malfunctions [99, 108] or even grow over time [97, 123]. What is of key importance, however, is that a robot existence (or life) is not instantaneous: hence, the evolution time-scale may interact with the life time-scale and adaptation can occur at both levels. This opportunity has been exploited by researchers for combining EC with learning, morphological development, or other forms of adaptation in order to obtain robots that are eventually better in performing their task.

Third, robots, and the environment they are immersed in, are usually complex and sometimes dangerous. The straightforward application of the main EC steps, i.e., evaluation, selection, and variation, is often not feasible using the real robot in its real environment, because the evaluation stage is too costly, not scalable, or even dangerous (for the robot itself, the environment, or other human operators involved in the optimization process). For this reason, the optimization is often carried out in simulation, possibly exploiting computing machinery that allows fast computation and well fits the inherent feature of EAs to be massively parallelizable. However, the simulation is rarely capable of capturing each subtle aspect of the reality and this can eventually result in an optimized robot whose behavior in reality is different to the one observed in simulation. This problem is known as the reality gap problem: we discuss it in Section 5.

2 EML for controller optimization

The controller of a robot is responsible for deciding which actions need to be performed. In many practical cases, this boils down to computing the control values that are sent to the body to guide actuation. Since robots are *embodied* agents, it is often convenient to equip them with sensors of various kinds, whose readings can be used as feedback by the controller to effectively guide movements (closed loop controllers). For these controllers, the control values are, broadly speaking, a function of the sensor readings and possibly of past experience, in case there is some form of memory.

In the field of ML, ANNs are everywhere being deployed to approximate functions for solving a very diverse variety of tasks. The robotics domain is no exception, and ANNs are among the most used tool within the agent controller to compute its next actions. Since engineering ANNs is not an easy task, requiring a lot of domain expert knowledge, experience, and trial and error, EAs often come in handy for obtaining well-optimized ANNs that can successfully control artificial agents. More in details, neuroevolution (NE) has been beneficially applied both for neural architecture search (NAS) and for training various flavors of ANNs (see Section 2.1). In addition, EAs have also been combined with Reinforcement Learning (RL) to effectively train ANNs to solve robotics tasks.

Aside from ANNs, several other approaches have been proposed merging ML techniques with evolutionary optimization for robot control, as we will detail in Section 2.2. Among them, we can include behavior trees and more classical control theory approaches.

2.1 Neuroevolution

Neuroevolution (NE) is the sub-field of evolutionary computation that deals with ANNs. Here, NE has the objective of searching for a good robot controller, i.e., a good robot brain, meaning that it has to effectively search the space of ANNs, eventually finding the most suitable one for guiding an agent towards the accomplishment of its task. Several works exist in which NE has been applied to optimize robot controllers. Even though it is not feasible to analyze each of them in detail, we aim at providing an overview of the features of some relevant studies, together with a characterization along the following axes:

- task to be solved by the robot
- ANN model and architecture
- EA used for the optimization
- presence of another adaptation time-scale (e.g., learning)

Finally, we discuss the case where NE has been used for evolving one or more controllers that are shared by many robots that interact, more or less tightly, to achieve a common goal.

Tasks considered. For allowing NE to truly shine, the tasks which have been mostly taken into consideration in the examined literature are those which require advanced controllers, either because of the task difficulty itself or because of the complexity of the robot involved. Among the first ones, we can mention *navigation* [18, 19, 50, 62, 161, 162, 165] (see Figure 1), predator-prey tasks [31], or *locomotion* tasks in complex environments [157], for which the sensor-actuator mapping becomes non-trivial, even considering a simple agent with a restricted set of actions.

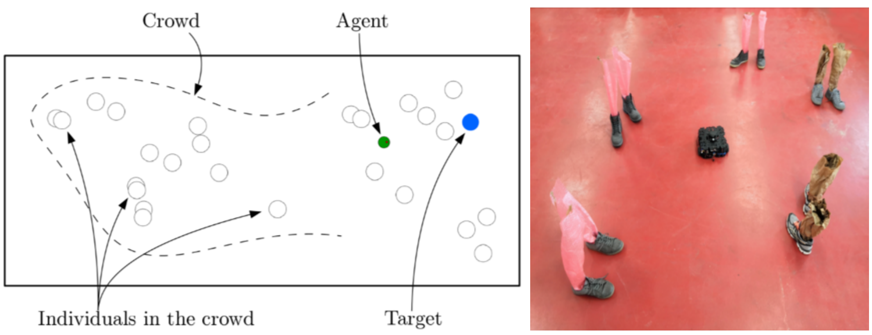


Fig. 1: A schematic view of a navigation task for a differential drive wheeled robot (left) and its realization with a few human mock-ups (right); both images are taken from [165]. The task has been solved by Seriani et al. [165] with NEAT.

Among the latter ones, locomotion is usually the go-to task, as for legged robots (see Figure 2), for instance, it already requires the control of several robotic parts [3, 57, 152]. In the case of legged robots, the task of obtaining a policy governing the movements of the legs is often referred to as *gait generation*.

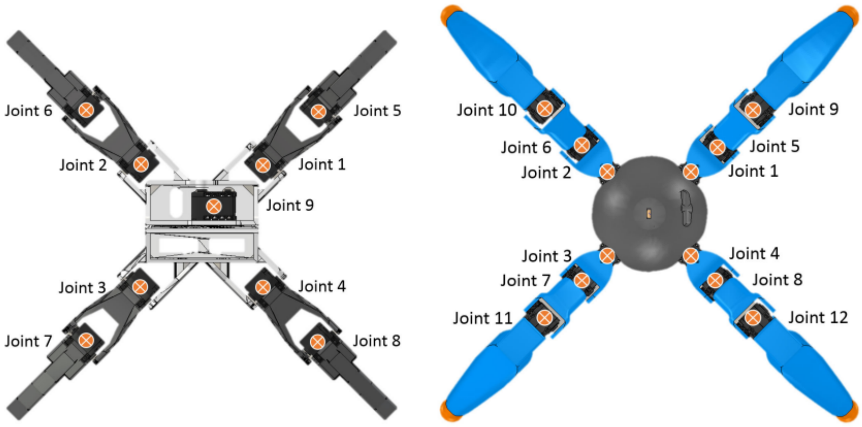


Fig. 2: Top view of two legged robots, each with four legs, for which the gait has been optimized by Reyes and Escobar [152] using a few variants of HyperNEAT. Image taken from [152].

Neuroevolution has been employed also for solving tasks related to industrial robots, among which the task of manipulating an object using a robotic arm with a suitable effector is a prominent example [71, 181] (see Figure 3).

Last, when dealing with modular robots or swarm robotics, self-assembly [88] or reconfiguration [194] tasks are also noteworthy test beds for evolutionary optimization: the use of ANNs for solving these tasks, and hence of NE for optimizing them, is still an open front.

ANN model and architecture. Focusing more on the NE aspect, the first features we concentrate on are the neuron model and the neural architecture employed. The McCulloch and Pitts neuron model (*perceptron*) is one of the most commonly used [109] for computing neurons outputs, thanks to its simplicity and computational efficiency [1, 176]. However, quite a few works have considered more biologically plausible models of neurons, such as the bio-inspired spiking neuron model [75, 178] (see Figure 4).

Concerning the architecture, again most of the works aim at simplicity and computational efficiency, choosing a fully-connected or a sparse feed-forward ANN. The latter ones in particular, can either result from a process of NAS (as we will see in the following paragraph) or can be obtained by pruning [122, 125], that is the removal of some neural structures (e.g., synapses) during the lifetime of the agent.

Some more sophisticated architectures of ANNs have also been used, namely Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM), in order to endow the controller with a form of memory, which is particularly useful for the accomplishment of tasks that benefit from tracking previous perceptions and/or actions [2, 196] such as navigation. In addition, recent advances in the

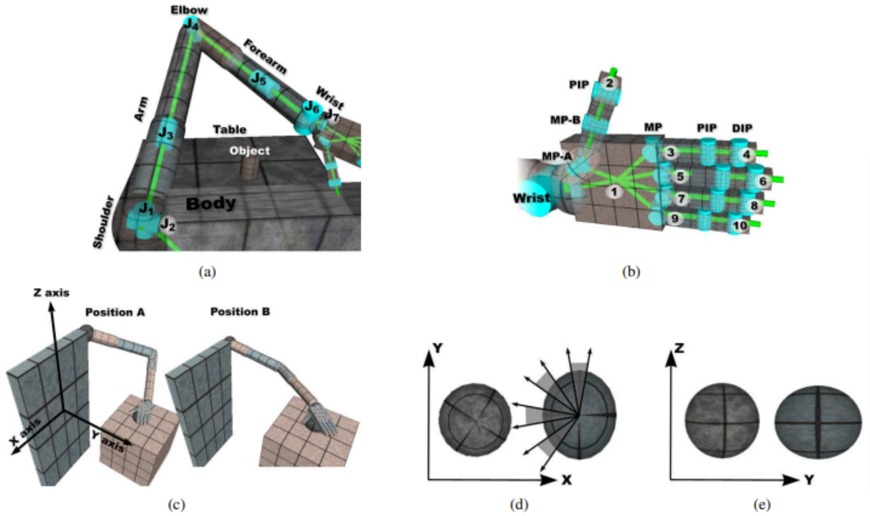


Fig. 3: An overview of the task tackled with neuroevolution by Tuci et al. [181]: the goal is to make the antropomorphic robotic arm (equipped with a human-like hand as effector) able to discriminate different kinds of objects using perception. The robot is controlled with an ANN with a single hidden layer of few nodes: the input layer is fed with readings coming from arm and hand proprio-sensors and tactile sensors; the output layer provides the arm and hand actuators values and the category of the object being manipulated as estimated by the robot. Image taken from [181].

field of deep learning (DL) have ignited the experimentation with deeper and more complex ANNs as robotic controllers, involving DL inspired elements such as self-attention modules [145, 177] (see Figure 5).

Last, a slightly separate role is played by Central Pattern Generators (CPGs), which are biological neural circuits that produce rhythmic outputs in the absence of rhythmic input [16], that have been successfully employed in those tasks (e.g., locomotion) where a periodic behavior is useful for achieving stability and high performance [81, 90, 105, 179] (see Figure 6).

Evolutionary algorithm. Another axis of categorization regards the evolutionary aspect, i.e., what is being evolved and what EA is being used to this extent. As seen in the previous paragraph, many works have relied on fixed neural architectures, be them feed-forward, recurrent, or more refined. In those cases, the most common approach to training involves evolving the parameters, i.e., the synaptic weights, of the ANN with EAs that are suitable for fixed-length numeric genotypes, such as evolutionary strategies (ES). Not only have ES been proved to achieve state-of-the-art performance on a wide set of tasks [157], but, as opposed to gradient-based optimization methods (e.g., backpropagation) they need not

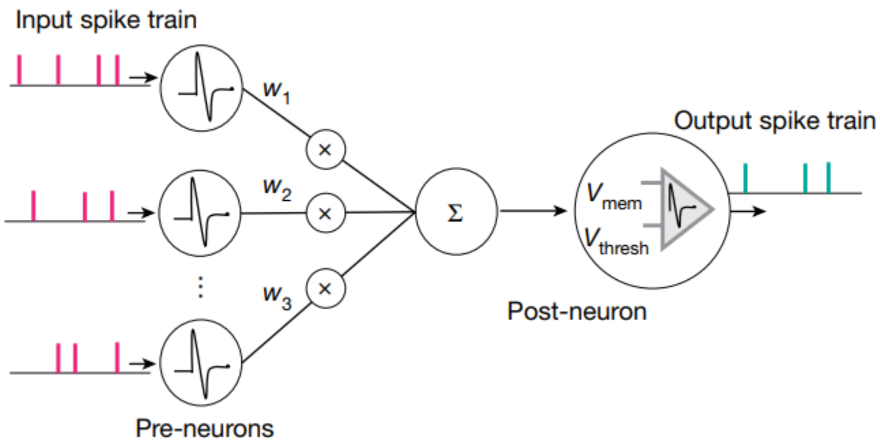


Fig. 4: A schematic view of a neuron model based on spikes, instead of continuous values; image taken from [156]. Nadizar et al. [124] optimized a controller for modular soft robots based on this model.

enforce any particular constraint on the outputs of neurons. As such, ES have been profitably used to overcome training issues like the non-differentiability in Spiking Neural Networks (SNNs) [32, 34, 50, 62, 124, 149, 161, 162]. In addition, within the domain of fixed architectures, quality-diversity (QD) approaches have also been explored, in order to avoid getting stuck in local optima [28, 49] (see Figure 7).

On the other hand, many studies have encompassed non-fixed neural architectures, relying on the evolutionary process for obtaining the most suitable ANN structure for the task at hand. In fact, as recently shown by Gaier and Ha [53], often the architecture of the ANN plays such an important role, that even random weights could be used, as long as the architecture stays untouched. Along this line, many have resorted to EC applied to NAS, mainly applying the NeuroEvolution of Augmenting Topologies (NEAT) algorithm [173, 174] for obtaining well performing and robust robot controllers [18, 19, 165, 188].

A step further has been taken with Hyper-NEAT [172], a generative encoding which evolves large scale ANNs with the principles of NEAT. This approach has led to outstanding results for the control of robots with high symmetry, such as legged ones [3, 26, 31, 61, 152], since Hyper-NEAT can automatically identify and effectively exploit problem regularities (see Figure 8), yielding to emergent controller modularity and high coordination. As a side note, some older works have also focused on the importance of ANN modularity for control tasks [20], yet handcrafted solutions which tried to take advantage of symmetry and repetitions [57, 183] were in general less fortunate than those obtained through a generative encoding.

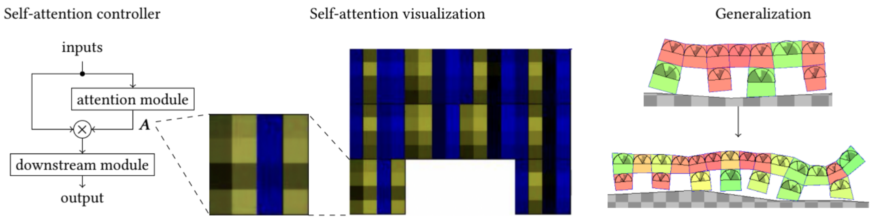


Fig. 5: A neural controller incorporating an attention module, optimized by Pigozzi et al. [145] for controlling a modular soft robot; image taken from [145]. Self-attention is a mechanism that allows the ANN to enhance some of the inputs with respect to other ones, resulting, in practice, in a form of auto-adaptation. Thanks to the attention module, the authors were able to make modules (colored squares in the images on the right) not needing any form of communication among them, still obtaining a collective behavior highly effective for the considered task (locomotion): specifically, the ANN in each module becomes more attentive to sensor readings that are more useful locally, hence performing a sort of specialization.

Adaptation time-scale. Another significant aspect to take into consideration when analyzing literature on ANNs for robot control regards the adaptation aspect. More precisely, most works focus on evolutionary adaptation, namely relying on the time-scale of different generations for improving the ANNs of the agents. However, some researchers have experimented with a shorter adaptation time-scale, that is life time learning, with the goal of fostering generalization to unforeseen circumstances, as for biological creatures [51, 133, 134]. In this context, the most fruitful results have been achieved with unsupervised Darwinian learning, i.e., learned traits are not transferred to the offspring, in the form of neural plasticity, i.e., Hebbian learning. These ideas have been ignited by Najarro and Risi [126] which have proposed to evolve the synapse-specific Hebbian learning rules instead of the synaptic weights, and have been productively applied for controlling different types of robots to enhance their resilience to body alterations and/or environmental changes [47, 141].

Modular robots and swarm robotics. The last view-point considers the amount of "independent" modules or robots involved. In the first case, we consider modular robots, for which the modules may be able to detach or at least to control themselves independently of the others (see Figure 9), whereas for the latter case we fall into the category of swarm robotics. In this context, we can put forward the concepts of monomorphic and polymorphic systems, where the first indicates modular agents or swarms were all components are alike, while the second refers to heterogeneous compounds. Some works have found monomorphism to be preferable [139, 144] for both modular robots with truly embodied controllers [111] and robot swarms [15, 88]. Intuitively, optimizing monomorphic components (i.e., modules or robots) is easier because the search space is smaller;

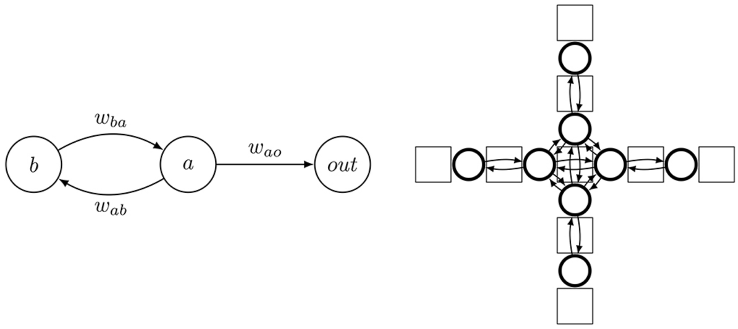


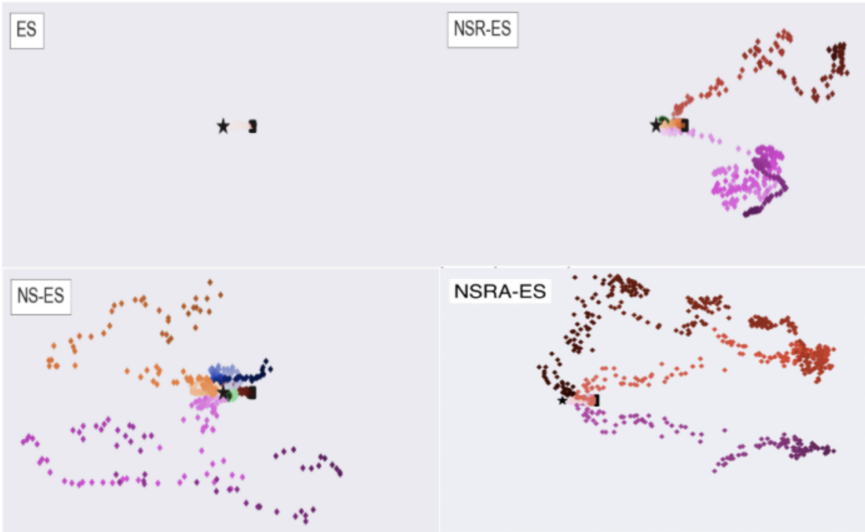
Fig. 6: The main component of a CPG (a differential oscillator, on left) and its usage within a specific morphology of a modular robot; both images are taken from [81]. In brief, a CPG constitutes a dynamical system whose evolution over time (i.e., the dynamics) is determined by a few numerical parameters. Jelisavcic et al. [81] used CPGs and CPPNs (see Section 3.1) for evolving both the morphology and the controller of simulated modular robots for the task of locomotion. Specifically, they used a compound of CPGs as an open-loop controller, i.e., one in which the CPGs are not fed with sensory feedback, since gait learning on a flat surface can be solved without such feedback. Moreover, the authors employed a Lamarckian approach, i.e., one in which some traits are developed after birth and are then inherited by the offspring.

moreover, the interaction between several indipendent components is facilitated when they are similar to each other. In addition, it has been observed that despite the independence of the agents, coordination can still emerge [15, 61, 124], even in the absence of explicit inter-agent communication [145].

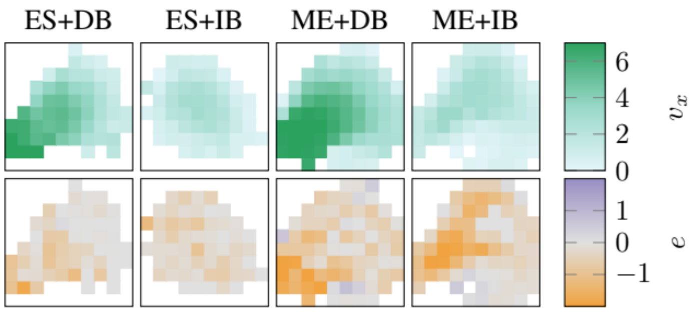
2.2 Other combinations of ML and EC for controller optimization

Besides ANNs, several other approaches to robot controller optimization have involved the application of EC. In this section we try to cover some relevant works within this broad area.

Combining evolution with Reinforcement Learning. Reinforcement Learning (RL) is the field of ML in which an agent learns the desired behavior through a trial-and-error process of interaction with an environment [175]. Due to its formulation, RL is naturally suited for the robotics context, in which it is applied to learn a controller, called *policy* in the RL context, which maximizes a certain reward signal during a given time span, e.g., during the life time of the agent. The core of RL, is hence the agent policy, which can be represented in several manners, and needs to be optimized through a policy search process [166] that can be conducted in various ways, depending mostly on the representation chosen and on the related search space.



(a) Four variants of ES, three of which employing a form of QD; image from [28]. Conti et al. [28] used ES for optimizing the neural network controlling a simulated humanoid for the task of locomotion. The plots show the behaviors of different agents in terms of position (top view): QD is clearly beneficial, as it allows the robot to overcome an obstacle being placed in front of it at its initial position (the star).



(b) Overview of the outcomes of several optimizations performed with ES or Map-Elites (ME, a form of QD evolution [121]) in terms of fitness (above) and evolvability (below), in the case of controller optimization for modular soft robots; image taken from [49]. x and y axes of each plot represent two behavioral descriptors. Ferigo et al. [49] compared different representations and EAs in terms of their ability to favor evolvability and exploring the space of solutions.

Fig. 7: Two examples of usage of QD EAs for optimizing the ANN-based controller of robotic agents.

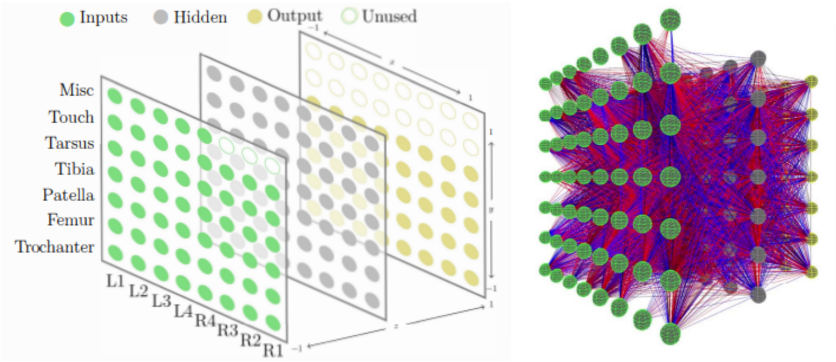


Fig. 8: Schematic view of the usage of Hyper-NEAT for optimizing the controller of a robotic-like eight-legged character: on the right, the mapping between inputs and output that exploits the physical structure of the robot; on the left, a representation of the best ANN obtained for the task of locomotion. Both images are taken from [3].

For instance, it is possible to rely only on EC for policy search with different EAs according to the specific representation chosen [28, 147, 157, 173, 190]. In this context, the dependency between the policy and the effectiveness of the corresponding behavior is not explicitly modeled; instead of trying to estimate the model, the search process only aims at moving solutions towards points in the search space which maximize the final objective, i.e., yield to a higher cumulative reward. Even though EC for policy search suffers from lower sample efficiency as compared to classical RL algorithms, there have been some attempts to tackle this issue [92, 147]. In addition, it has been shown how EC can be a scalable alternative to classical RL [157], achieving state-of-the-art performance on a benchmark of control tasks.

Other examples in which EC has been applied to RL, aim at leveraging the advantages of the two approaches. In particular, two mainstream ideas suggest to make use of EC to improve the outcome of RL techniques or to allow EC and RL to act as two concurrent forms of adaptation occurring along different time-scales, as in [63] (see Figure 10). Concerning the former idea, EC has been used for reward shaping, to discover a reward signal that allows the agent to efficiently learn [131].

Some have tried to achieve explainable policies, which are of primary importance in critical settings, relying on tree-based representations and genetic programming (GP) in combination with RL [30, 67, 193]. Last, EC and RL have often been combined, in and out of the robotic context for algorithm acceleration [41, 106, 107, 119, 140], using EC as bootstrapping for RL, or for hyper-parameters tuning [76].

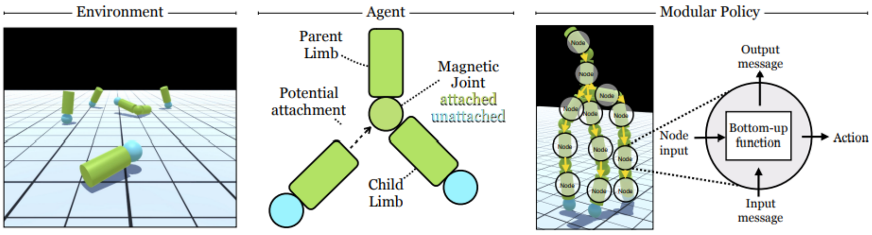


Fig. 9: An overview of an evolutionary approach for optimizing the controller of simple robotic modules that have the capacity of joining and hence forming complex morphologies; image taken from [139]. Single modules are governed by ANNs that determine the actions to be performed (i.e., how to control the actuator) and the messages to be sent to neighbor modules. Despite the approach is described by Pathak et al. [139] as a form of co-evolution of morphology and controller, the optimization is actually a form of RL.

Behavior trees. Behavior Trees (BTs) describe the controller of an agent by the means of a tree, which is meant to be traversed until reaching a terminal state (see Figure 11). The leaves in BTs correspond to executable behaviors, whereas internal nodes represent control flows. This representation was first invented to enable modularity in artificial intelligence for computer games, but has recently received an increasing amount of attention in the robotics community [73]. The main advantage of BTs lies in their interpretability, which comes in particularly handy when the considered task is critical or there is the need to adjust the learned behavior by hand to address reality gap issues [163] (see Section 5). Given their tree-based structure, GP is naturally suited for optimizing BTs [74], and it can also be boosted with grammatical evolution (GE) to enhance the final performance [64, 82]. BTs have also been profitably applied in various multi-agent contexts, to ease the understanding and prediction of the emergent swarm behavior [84, 128, 129].

Evolution in control systems. Despite not belonging fully to the ML context, several classical control theory approaches have benefited from EC techniques in robotics, so we briefly report some relevant works in the field here. For instance, EC has been used in combination with forward kinematics equations for trajectory planning in robotics [186], in order to reduce the impact of noisy environments [71]. In addition, GP has often been exploited to address the problem of inverse kinematics in manipulators (see Figure 12), aiming at obtaining precise yet interpretable results [21, 38, 89, 151]. Moreover, EC has been often applied in combination with control algorithms for parameter fine tuning in robots like Unmanned Aerial Vehicles (UAVs) [11] or walkers [69].

Last, we quickly mention some applications of EC on fuzzy control systems [45, 130, 148], that are those control systems based on fuzzy logic. In this context, EC has mostly been used for tuning the parameters of the fuzzy controller, in

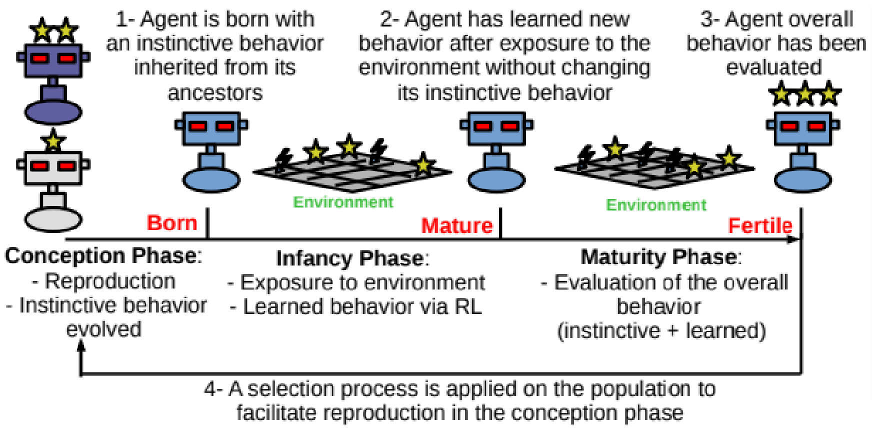


Fig. 10: The scheme principled by Hallawa et al. [63] for combining RL and evolutionary computation for ER; image taken from [63]. During the evolution, robotic agents are initially developed with an instinctive behavior dictated by a policy inherited from parents; their policy can change during their life based on their interactions with the environment, according to an RL approach; the selection phase of the evolutionary process takes finally into account the fitness resulting from the learned behavior, rather than just the instinctive one. The authors assessed their approach on a few (simulated) robotic tasks and found promising results.

several robotic scenarios. In particular, autonomous vehicles have been a widely used test-bed for deploying fuzzy controllers with feedback [58]. Various works have encompassed wheeled robots in navigation tasks [25, 85], and have used EC to achieve more interpretable results [86, 87].

3 EML for body and body-and-brain optimization

The *embodied cognition* paradigm [143] posits that the intelligence of an agent (natural or artificial it may be) emerges from the complex interactions between the body, the brain, and the environment. Intelligence is not only rational, but is also embedded in the body. The paradigm stands in contrast with the traditional view inside the computer science community of intelligence being abstract, purely Platonic “reason” [191]. Indeed, bodies co-evolved with brains and shaped them, inasmuch as the body defines the actions that the brain can “afford” [54]. As a matter of fact, there exist organisms capable of performing complex computations by the means of their bodies only. Individuals of the genus *Planaria* regrow amputated limbs by virtue of biological processes localized in the severed portion of their bodies [101]. Individuals of the species *Trichoplax adhaerens* self-organize their cilia to locomote in the absence of any nervous system [17]. As a notable instance in the artificial domain, passive walkers locomote by the

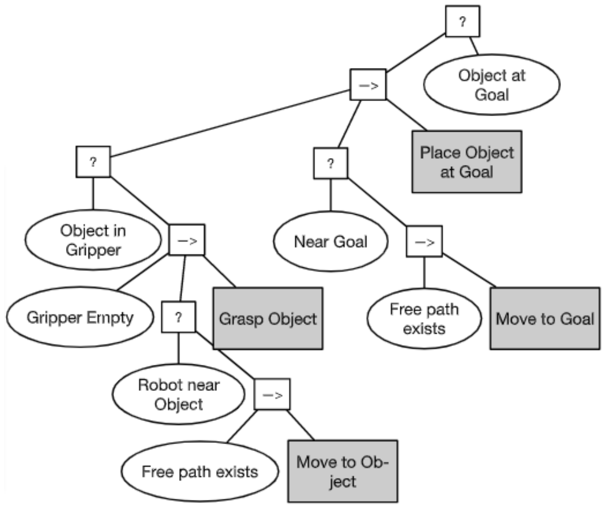


Fig. 11: An example of a BT for a mobile manipulator, i.e., a robot composed of a manipulator mounted on a wheeled platform; image taken from [73].

means of body dynamics only [110] and are already a mature field of study within robotics [27], to the point that artist Theo Jansen applied EC to design the *Strandbeests* (see Figure 13), passive walkers that rely on wind energy only, to raise environmental awareness [80].

To no surprise, researchers delved into the issue of agent body optimization; such topic is relevant for EML as it is deals with problems of RL, itself a part of ML. With respect to controller optimization, the body places an additional challenge: most agent bodies are not differentiable and thus cannot be optimized by the means of gradient-based optimization methods. Ha [60] explored policy search to jointly optimize the control and body parameters of simulated bipedal walkers, but restricted the search to different configurations of the same morphology (e.g., length of the pedals). To fully unleash its potential, body optimization must be capable of open-ended complexity. EAs fit this goal, since they can evolve any artifact given an appropriate representation and a fitness function.

Starting from the seminal work of Sims [168]—the "father" of ER—who first unveiled the power of evolution for optimizing the bodies of virtual creatures, researchers have increasingly relied on EC also for body optimization and body-brain optimization. Both of them pose unique challenges when compared to controller optimization, as we shall see in the following.

3.1 EML for body optimization

As representing a body is non-trivial, body optimization propelled researchers to put effort into conceiving representations that are: (a) scalable, in terms of the

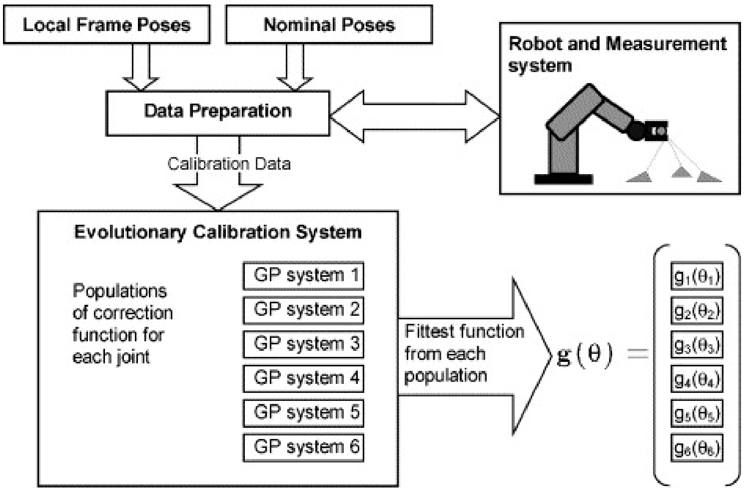


Fig. 12: An overview of the approach of Dolinsky et al. [38] for performing the inverse static kinematic calibration of industrial robots using GP; image taken from [38]. The authors used the capability of GP of optimizing both the structure and parameters of a regression model for building concurrently six models for the six joints of the robot.

solution complexity, and (b) effective, in terms of the quality of the solutions. The literature usually groups body representations into two categories: *direct* and *indirect* (sometimes also called *generative*). Direct representations provide a one-to-one mapping between genotype and solution, whereas indirect representations, on the other hand, evolve a data structure that we can map to a solution in a non-trivial procedure. In a certain sense, indirect representations evolve the blueprint that can generate a solution by decoding the instructions contained in the genotype (hence the name "generative").

Direct body representations. Direct representations provide a one-to-one mapping between genotype and solution. They are, in general, easy to craft, but potentially less scalable than indirect ones, as the complexity of the genotype directly depends on the complexity of the solution. Examples include graphs and numeric vectors.

Numeric vectors are the most direct of the representations. In that case, every gene (i.e., scalar) encodes a specific trait of the solution. For example, [144] evolved the morphologies of 2D voxel-based soft robots encoding each voxel (given a maximum enclosing grid) with a real number in a vector (see Figure 14). Albeit simple, this representation proved effective to evolve a variety of shapes and behaviors, also using a speciation mechanism [174].

Sims [168] first evolved virtual creatures by encoding their morphology as a directed graph, with nodes being the rigid parts and the edges being the connec-

Fig. 13: One of the Strandbeests realized by the dutch artist Theo Jansen, named by the creator Animaris Percipiere; image taken from [80].

tions among those. The resulting creatures not only evolved life-like patterns for locomotion, but also for competitive co-evolution, where individuals belonging to different populations (forming species) evolve to win a one-on-one fight for the control of a resource (a cube in the arena where the simulated creatures fight) [167]. As testified by adoption of open-source platforms, graphs (including trees [72]) have proved an intuitive representation for modular robots [4, 195].

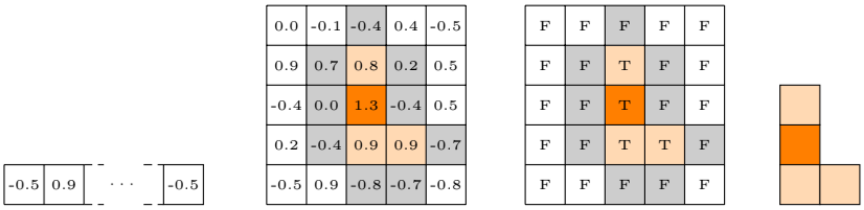
Indirect body representations. Indirect representations evolve a data structure that we can map to a solution in a non-trivial procedure.

Indirect representations attracted the most interest in the community. In fact, they bear similarities to what happens in biological organisms, whose complexity and diversity are the result of an astonishing data compression feat, the "genomic bottleneck" [153]: just a handful of genes in the DNA encode the instructions to build a complex organism (humans, with 30 trillion cells and all their cerebral complexity, just have 30 000 genes in their genome [189]). As a result, indirect representations might provide a bridge toward more complex robotic systems.

Examples of indirect representations include:

- Compositional Pattern Producing Networks (CPPNs);
- sequences mapped to strings belonging to languages defined by grammars;
- Gene Regulatory Networks (GRNs).

Indirect representations allow great freedom to the designer and are suitable to evolve properties we know are beneficial a priori. As a matter of fact, CPPNs [171]—univariate function networks resulting in multivariate functions expressing repetitions and symmetries—evolved to express multi-material 3-D morphologies with life-like patterns, e.g., symmetry [5] (see Figure 15). On



(a) Real vec. $\boldsymbol{v}$. (b) Real matrix $\boldsymbol{V}$. (c) Bool. mat. $\boldsymbol{M}$. (d) VSR.

Fig. 14: A schematic view of the grid-based, fixed-length direct representation for evolving the body of 2D voxel-based soft robots used by Nadizar et al. [123]; image taken from [123]. A fixed-length real vector is first reshaped in a matrix; then, starting from the cell with the largest values, subsequent voxels are added in correspondence with cell with next largest values that are adjenct to already visited cells. Similar representations have been used, e.g., in [49, 144, 145].

the same research line, evolved CPPNs expressed effective morphologies for 3-D voxel-based soft robots for squeezing through tight spaces [22], underwater locomotion [29], and recovering from damages [99]. Most notably, evolved CPPNs expressed—*in silico*—the morphologies for organisms to be assembled—*in vivo*—from cells of *Xenopus laevis* (a frog); such organisms (the *xenobots*) can locomote by themselves [98] as well as self-replicate [96] and are, to date, one of the most impressive simulation-to-reality feats within the robotics EML community.

Similarly to CPPNs, grammars allow the designer to bias the search space, but in a different way: grammars specify the building blocks of a solution and what relationships are admissible among those blocks. There exists a rich literature on the evolution of solutions defined in languages described by grammars, starting from Grammatical Evolution (GE), a form of GP [95], and its more recent variants, as Probabilistic Structured GE [112] or Weighted Hierarchical GE [8]. Lindenmayer systems (L-systems) [102]—first developed to model the biological development of multi-cellular organisms—consist of an alphabet of symbols to be replaced with replacement rules (see Figure 16): symbols can be robot modules and instructions on how to assemble them [68]. As a result, L-systems have succeeded at evolving modular robots [116, 117] on the Revolve framework [72].

When evolving embodied agents, the effect of mutations is non-trivial and hampers evolvability [44]. To this end, Bongard and Pfeifer [14] conceived GRNs as linear genotypes that, through differential gene expression and the diffusion of gene products, transform a single structural unit into a complete robot morphology via a development process: mutations expressed earlier in development tend to have a more variable effect than mutations expressed later, lending to the evolvability of the representation.

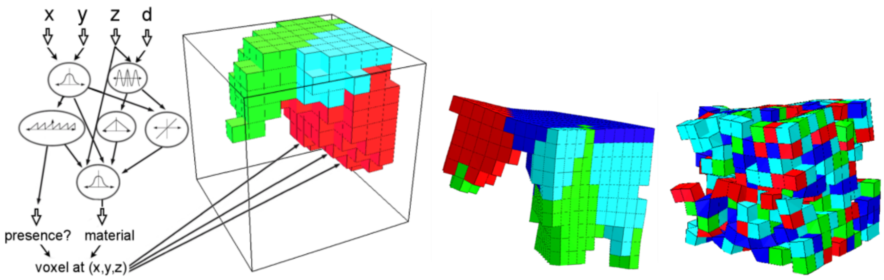


Fig. 15: Schematic view (on the left) of an indirect representation based on CPPNs for optimizing the body of 3D voxel-based soft robots; in the middle, an example of an evolved body that is effective in locomotion and exhibits life-like patterns as repetitions and simmetries; on the right, a robot evolved with a direct representation (conceptually similar to the one of Figure 14)—all the three images are taken from [24]. Cheney et al. [24] used NEAT as the EA driving the evolution of the CPPNs describing the robot bodies. The controller here simply consists in different phases of the expansion-contraction cycle of the voxel composing the body, here visually encoded with different colors.

Comparisons among representations. Considering how important the choice of the representation is, some works did consider comparing different representations; most of them considered the scenario of modular robots, because they allow for great expressive power in describing the bodies, due to their inherent nature of systems composed of many components (see Figure 17).

Extending the work of Auerbach and Bongard [5], Cheney et al. [24] showed that evolution of CPPNs is more effective than a direct representation at expressing morphologies for multi-material voxel-based agents, as it produces regular patterns as seen in nature (see Figure 15). Also using modular robots, Veenstra et al. [185] argued that indirect representations are more suitable for evolving small-sized robots, a conclusion that we find counter-intuitive. De Carlo et al. [33] found an indirect L-system representation to underperform a tree-based direct one in terms of locality of the representation [154], since many genes needed to align to create a positive effect on the solution (epistatic effects).

In the context of voxel-based soft robots, both Nadizar et al. [123] and Ferigo et al. [49] compared different representations (a direct and indirect ones) for the body of the robots. In the former study, the authors considered the case of morphological development of robots, i.e., the addition of new voxels to the body during the life of the individual, as dictated by the genotype according to the chosen representation. In [49] (see Figure 17c), the aim was to compare the representations in terms of their ability to favor the evolvability, i.e., the possibility to produce fit offspring, of found solutions.

Despite the efforts devoted by the authors of the studies surveyed above, a comprehensive comparison among representations is arduous, given the abundance of them; even indirect ones can differ wildly in terms of properties of the

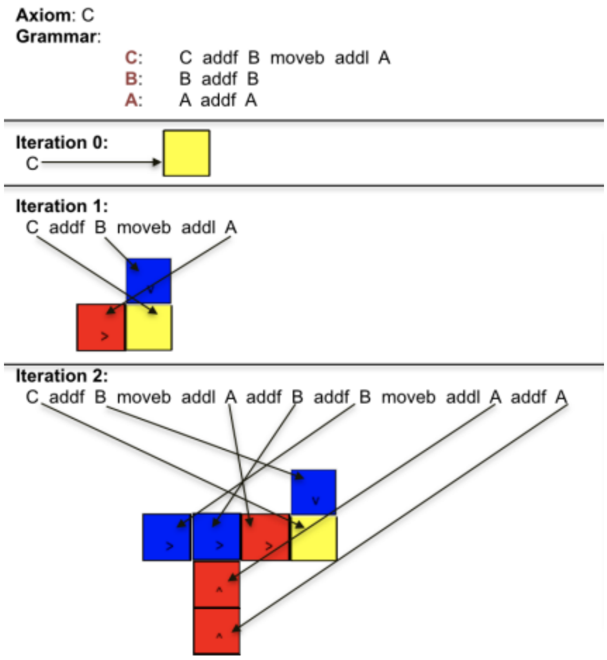


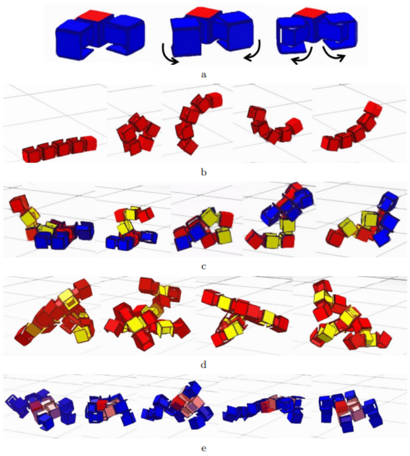
Fig. 16: A schematic overview of an indirect body representation for modular robots based on L-systems; image taken from [118]. There is a symbol in the grammar for each available module and a few symbols for operating on the current expression, by replacing symbols, and for changing the reference point. Miras et al. [118] used this representation for evolving (simulated) modular robots composed of three possible modules: a controller, a joint actuated by a servomotor, and a passive structural module.

representation [155]. Moreover, the works mentioned so far consider different robot types and frameworks. Thus, the community still misses a full understanding of the trade-offs among representations.

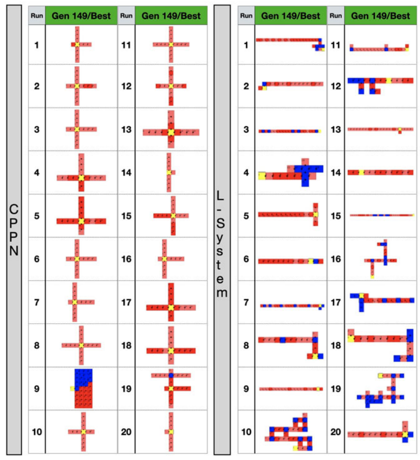
3.2 EML for body-and-brain optimization

Body-and-brain optimization has historically been difficult [83], due to the deep entanglement between the brain and the body. The seminal work of Lipson et al. [103] suggested the reason to be the ruggedness of the fitness landscape: they proved that evolving the morphology of voxel-based agents for a fixed controller results in more premature convergence to local minima than evolving the controller for a fixed morphology. Controllers need to adjust to their morphology [43], as the body modulates the communication (motor and sensory) between the brain and the environment.

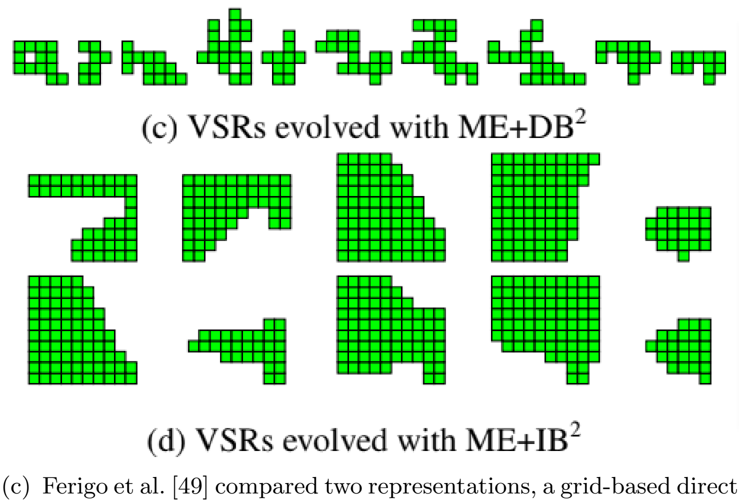
Researchers have since then tackled body-and-brain optimization either by *joining* the body and the brain in the same optimization or *decoupling* the two



(a) Veenstra et al. [185] considered modular rigid robots composed of up to 20 modules and found that, in particular at the initial stage of the evolution, a generative representation is better than a direct representation. Modules are of two kinds: a fixed cube and a cube-like one with a face that can be displaced with respect to the others by actuating a rotational joint. The generative representation is based on L-systems. Image taken from [185].



(b) Miras [114] compared a representation based on CPPN and one based on L-systems for evolving the body of modular rigid robots composed of four kinds of modules. The authors considered not only the impact on the effectiveness, i.e., the ability of the robot to move, but analyzed how the two representation bias and constrain the search space. They found that CPPN gives slower but more stable gaits. Image taken from [114].



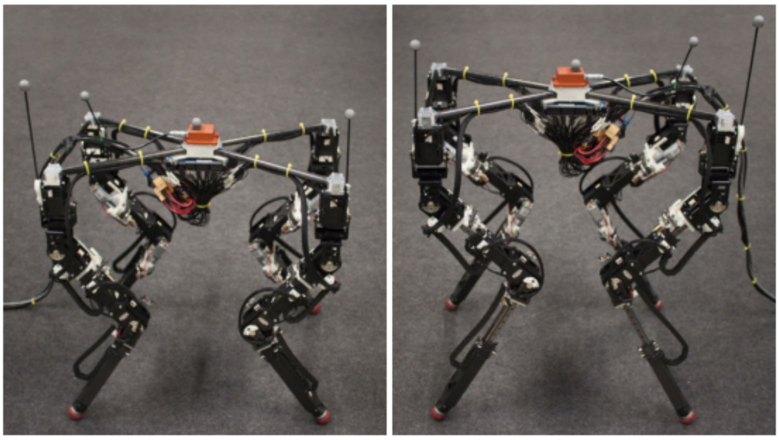
(c) Ferigo et al. [49] compared two representations, a grid-based direct one and an indirect based on Gaussian-mixtures, and two EAs in terms of their ability to foster the evolvability of found solutions, in the context of 2D voxel-based soft robots. The authors found that the evolvability is mostly influenced by the EA, while the representation largely affects the fitness. They also found an evident bias towards larger robots with the indirect representation (bottom in this figure). Image taken from [49].

Fig. 17: A visual summary of the outcome of three studies comparing different representations for the body of modular robots. In all the cases, the figure shows a few individuals obtained by the means of evolutionary optimization using one or two representations.

(e.g., with two nested optimization loops). Both approaches have benefits and costs: the former simplifies the optimization, while the latter explicitly takes into account how brains need to adapt to their bodies.

Joining body-and-brain optimization. Joining body-and-brain optimization allows to solve two (entangled) optimization problems as one single optimization problem.

When the representation is direct (as in Section 3.1), that amounts to joining the two representations (for brain and body) into one single representation [144]. For instance, Nygaard et al. [136] successfully evolved real four-legged robots (see Figure 18) using one single numerical genotype for both morphological (leg lengths) and control (gait) parameters and showed adaptation as physical conditions changed. Going back, Sims [168] evolved the neural network controllers of his virtual creatures as graphs embedded inside the nodes of the morphology graph (see Section 3.1), with mutations applying to both types of graphs.



(a) Shortest possible legs **(b) Longest possible legs**

Fig. 18: The four-legged robot used by Nygaard et al. [136] for the joint evolution of morphology and control; image taken from [136]. The robot can (slowly) change the length of the legs: since the time-scale of this variation is much longer than that of the gait, the change is made offline, i.e., between subsequent fitness evaluations, and is hence considered a form of morphological evolution.

Indirect representations (as in Section 3.1), on the other side, can potentially encode body and brain with the same data structure. Cheney et al. [23] evolved voxel-based agents with two CPPNs, one to express the morphology and one to express the distributed open-loop controller, and reducing selection pressure on individuals with recent morphological mutations: evolution was free to mutate one CPPN or the other, while having time to "readapt" to new morphologies. Their approach showed potential to avoid local optima. Finally, Pontes-Filho et al. [146] evolved one single neural cellular automaton for developing the body and functioning as the brain for multi-material car racers and is a promising line of research (see Figure 19).

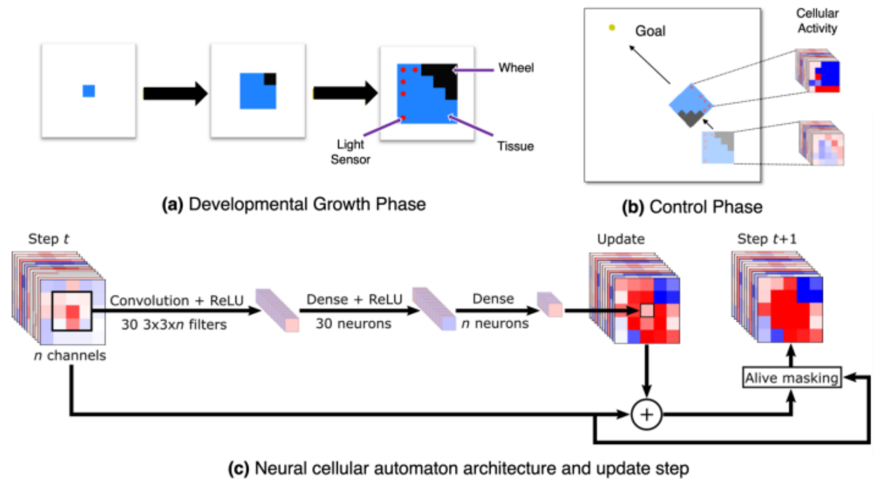


Fig. 19: A schematic overview of the approach proposed by Pontes-Filho et al. [146] for the concurrent evolution of the body and the brain of a simulated robotic mobile agent composed of a few modules with different roles. The key contribution of this work is in the neural cellular automaton used for determining both the way the body develops, in an initial instantaneous stage at the beginning of the agent life, and how it behaves during its life: in the latter stage, actuable modules (i.e., those equipped with wheels) are controlled based on the state of the automaton. Image taken from [146].

Decoupling body-and-brain optimization. Decoupling approaches usually cast the body-and-brain optimization problem as a nested optimization problem. An outer evolutionary loop searches in the space of bodies, while an inner optimization loop searches—for each body—in the space of brains; intuitively, that is how nature shaped animal life on Earth. Approaches then differ according to the algorithm employed at the inner loop. Interestingly, both evolution and learning (the slowest and the fastest time-scales of adaptation, respectively [169]) appear.

Some works adopted RL to learn the brains, as RL is a sample-efficient optimization over the lifetime of a robot. Most notably, Gupta et al. [59] evolved bodies with a genetic algorithm and learned their brains with RL to master a wide gamut of tasks for tree-based simulated robots, verifying that such environmental complexity fosters the ability of a body to facilitate learning of novel tasks. Additionally, bodies that are more physically stable and energy efficient facilitate learning the most.

On the other side, the inner loop need not consist of learning, but may be evolutionary just like the outer loop. In particular, every body comes with a population of brains that evolve and whose best individual determines the fitness to the body-brain pair. Using this approach, Jelisavcic et al. [81] compared a Darwinian and Lamarckian approach for the inner loop and found that the latter considerably reduces the time to learn a good solution. Similarly, Le Goff et al. [100] showed that initiating learning from a brain inherited from an archive of learned brains, rather than from a randomly initialised one, both the speed and magnitude of learning increased over time. Subsequently, Miras et al. [115] found that indeed the inner evolutionary loop produces robots that perform better on a given task and Luo et al. [105] quantified the *learning delta*—the performance difference between inherited and "learned" brains—in terms of morphological descriptors.

4 Other combined usages of EC and ML in robotics

There are many other cases in which EC has been successfully used for optimizing other components of scenarios involving robots than "just" body and brain. These cases include other parts of the robots, such as the sensory apparatus of voxel-based soft robots [46, 48] or the object recognition part of robot grippers [70], the way robot develops during their life [97, 123, 127], as well as the task itself [135, 187], or even the simulator used for optimizing, in simulation, a real robot [12]. While all these approaches can be encompassed in the field of ER, they hardly fit the definition of EML, since they do not directly and explicitly employ ML.

In a few other cases, ML and EC "met" in contexts where they both concurred in determining a solution for a broader problem. For example, Pigozzi et al. [144] studied what are the key evolutionary factors (namely, the employed EA) affecting the diversity of evolved voxel-based soft robots: the authors interpreted the notion of diversity taking the inspiration from biology, where a human observer categorizes individuals within a predefined structure of categories (species). In the cited work, the authors used a supervised ML pipeline to replace the human observer with a model learned from a few data points labeled by an actual human observer. The model is hence used for measuring the diversity of a large number of evolved populations of robots that, for the scale, could not have been manually inspected by an actual human. The authors found that those EAs that are designed to promote diversity are indeed able to achieve this goal, in particular when behavioral traits are used for telling apart individuals. However, they also found that external conditions (i.e., the environment) may have an even greater impact on observed diversity: e.g., robots evolved for doing locomotion on a downhill terrain very often ended up having a rounded shape which facilitated rolling.

A similar approach, i.e., one in which ML powers a component of a larger system in which at least another part is based on EC, has been adopted by Koos et al. [94]. For the aim of fighting the reality gap (see Section 5), the authors trained a ML model for estimating the transferability of robot controllers evolved in simulation, i.e., the degree to which their performance, as seen in simulation, do not change when moved in reality. The evolution itself was bi-objective, with one objective being the transferability and the other being the effectiveness. According to the authors, evolving for transferability and effectiveness together is practical way for fighting the reality gap problem.

5 The reality gap problem

As already mentioned in the previous sections of the chapter, in ER an EA operates on a population of robots plunged into an environment. The ultimate goal is that of designing a robot (its brain, body, or both) which maximizes all the desired performance when involved in a given task.

Since EAs are loosely inspired by natural evolution, which has been quite successful in evolving various kinds of biological agents, i.e., living creatures, they appear promising for optimizing artificial agents, i.e., robots, too [104]. However, their stochastic nature implies that multiple executions have to be performed to assess their outcome in practice, and calls for a solid statistical analysis for determining the best strategy to be adopted [9, 39]. Unfortunately, repeating several (often thousands) experiments on real robotic platforms can be hugely expensive, time consuming, and often potentially dangerous. For this reason, most ER applications focus on evolving robots using simulated robot-environment interactions, and then transfer the obtained results to the real robot-environment system (i.e., simulator-to-reality transfer, otherwise known as *sim-to-real* transfer) [94]. Only few works evolve controllers directly on the robot, and often optimize few individuals during few generations, which reduces the effectiveness of the evolutionary methods [40, 56, 104, 150].

Simulation is therefore considered a very powerful and useful tool in the context of evolution of robots, since a long time [184]. However, at the same time, it carries some disadvantages that often result in an under-performing, or, in the worst case, completely ineffective solutions when applied on the real system [42, 65]. This effect is often called *reality-gap* [79], i.e., the difference between the effectiveness measured in simulation and the one measured in reality of a solution that has been obtained in simulation (see Figure 20).

Although the term reality gap is widely used in the ER community, this problem is not limited only to applications of EAs in robotics. Instead, as a matter of fact, it can occur on any system designed and developed using simulators, and then implemented in reality. RL applications, for example, can also be affected by this problem when trained using simulators [158, 159]. More generally, in the control system community, where dynamic systems are controlled typically

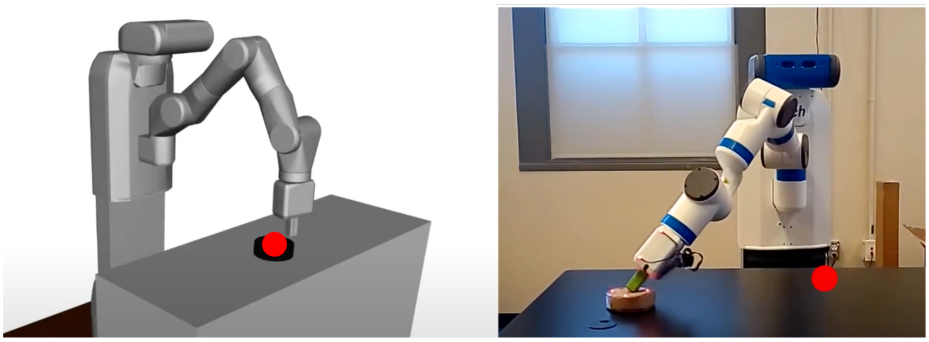


Fig. 20: Example of reality gap, from [142]. The robot effector reaches the target (the red circle) in simulation (left), but fails to reach it in the real application (right).

relying on model-based approaches—i.e., where the way the simulation models the real system is known and exploited in the control strategy—the reality gap problem is for example referred to as model-plant mismatch [6, 7, 91, 138, 164]. In this section, however, we focus our attention on the reality gap in ER.

The main questions one needs to ask when dealing with a reality gap are:

1. what are the differences between simulators and reality that affect the effectiveness of the solution found in the simulation?
2. how can the mismatch be reduced?

Of course, as one might expect, the above questions are interrelated, often dependent on the problem addressed, and thus have no one-size-fits-all answers.

From a practical point of view, the mismatches between simulator and reality, being the cause of the reality gap, can be classified into three macro-categories [137]:

- *robot-robot* correspondence,
- *robot-environment* correspondence,
- *environment-environment* correspondence.

The former refers to the physical differences between the simulated robot and the real one, such as morphological differences. Robot-environment differences refer to errors, or approximations, in the dynamic interactions between the robot and the environment, and include both perception and actuation. On the other hand, the latter, i.e., environment-environment correspondence, concerns the misrepresentation of significant features of the environment.

Clearly, once the source of such differences is recognized, there is a trivial solution to reduce the gap: improving the simulator by bringing it as close as possible to the real system. In [113], for example, simulator is integrated with real data, in [79] with noise on sensor level, while in [12, 192] the robot model is directly learned online. Klaus et al. [93] presented a broad comparison of model improvement approaches in dealing with the reality gap.

However, this general solution has a negative impact on the efficiency of EAs even when applied in simulation: the time or computational effort needed for finding a solution increases. Simulators with higher accuracy tend to be more demanding in terms of computational time, thus losing one of the main advantages of using simulators. Hence, a trade-off exists between the *effectiveness* of an optimization technique (i.e., the quality of the solution it finds for a given task), and its *efficiency*. Assuming, therefore, that a certain amount of reality gap must be tolerated, it is necessary to build or tune optimization techniques (including evolutionary ones) in such a way that they match the user expectation in terms of reality gap, effectiveness, and efficiency.

There are different works in the literature proposing approaches able to mitigate the reality gap effect in ER. Although we cannot discuss each of them in-depth, we provide below a categorization of those which work in the optimization phase—i.e., the vast majority of them. A rather different approach is to try to fill the reality gap by building robots that can adapt to changes while they operate. For example, Song et al. [170] employed ES to realize a form of meta-learning that allows a real legged robot to adapt to changes in its body dynamics (due to robot load and battery voltage change). In principle, this kind of adaptation could be achieved also with controllers that exhibit plasticity, such as SNNs or ANNs with Hebbian learning.

Domain randomization. The idea of domain randomization is to improve the simulator robustness by providing sufficient simulated variability at the training time, such that the model is able to generalize to real-world data in test. The result is a more robust solution to model variation, and it can be achieved following different strategies: performing adaptive feedback control [35, 120, 149], i.e., closed-loop control strategies in which some measurements are fed back to the controller to properly adapt the control law to instantaneous changes in the environment or in the robot itself, developing robots within a variety of different environments [10, 13, 55], or randomly perturbing model parameters during the evolution [77, 78, 180].

Simulator flaws avoidance. Avoid the exploitation of simulator flaws is grounded on the idea that, if the solution search-space during evolution is such that it exploits the defects of the simulator, then the learned solution will necessarily be subject to the reality gap when transferred to the real robot. Therefore, the approaches that address the reality gap out of this point of view try to reduce the exploitation of simulator pitfalls automatically adjusting hyperparameters [50, 51, 182], or slowly increasing the representative power of the search space [52].

Fostering transferability. A promising idea is the one that consists in learning a model that can roughly quantify the reality gap between a simulator and a real robot for each investigated design, i.e., to estimate the design (non-)*transferability*, and use the estimate to constrain the design process. This is the main idea of [94], where authors formulated the transferability approach, i.e., a bi-objective algorithm in which a task-dependent performance metric is maximized, while a disparity measure between performance in simulation and in reality (i.e., a non-transferability measure) is simultaneously minimized (see Figure 21). The cited work is particularly relevant to this chapter because the authors use a classical ML pipeline for estimating the transferability based on some features that can computed for each possible solution being evolved.

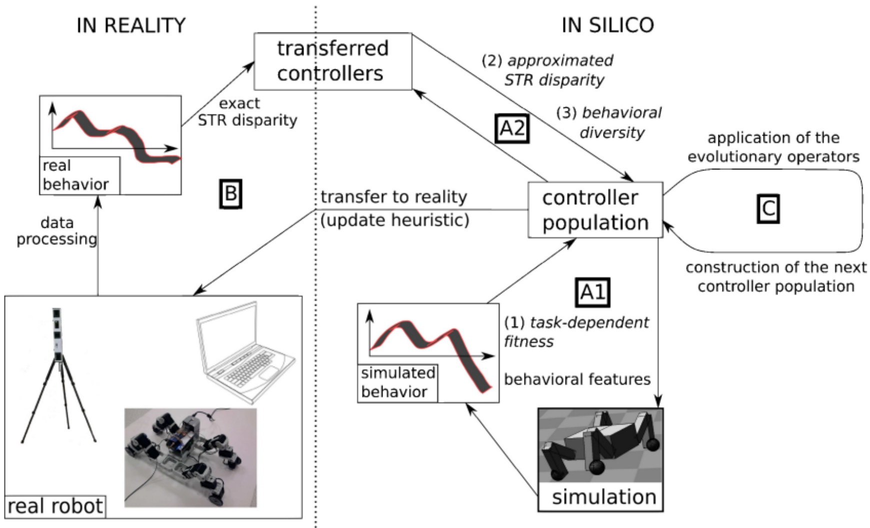


Fig. 21: An overview of the algorithm proposed by Koos et al. [94] for evolving robot controllers avoiding to fall in the reality gap problem. The designer needs to identify a few behavioral descriptors that can be computed both in simulation and reality; based on these descriptors, an estimation of the (non-)transferability (STR, i.e., sim-to-real, disparity) of each solution evaluated in simulation is produced using a supervised ML model. The evolution is bi-objective, with one objective being the non-transferability, to be minimized, and the other being the task-dependent effectiveness. The ML model for estimating the non-transferability is maintained updated by assessing in reality some of the solutions found in simulation at regular intervals. Image taken from [94].

The same idea has been subsequently pursuit in other works in which, for example, the discrepancy between simulator and real robot has been monitored in morphology [36, 98, 160].

6 Concluding remarks and open challenges

The field of robotics offers a plethora of opportunities for applying evolutionary optimization. Many components of robots and their tasks can be optimized, rather than manually designed, and the corresponding search spaces are often very large and hardly amenable to be searched with more traditional search methods: hence EAs can deploy their potential as universal search techniques. Moreover, the combination of EC and ML appears to be particularly effective when used to tackle different facets of a larger problem.

There are, however, some open challenges. We believe they are well captured by the recent work of Eiben [42], who suggested that future research in ER should attempt to:

1. Target more realistic robots and real tasks: this will require to (i) focus on robotic subsystems that are currently overlooked, such as sensors, (ii) consider more complex tasks, such as, e.g., object transportation instead of the simple locomotion, and (iii) make assessment more scalable, in such a way that it can actually be executed for many candidate solutions. We think that the latter point is a particularly fertile terrain for the combined use of EC and ML.
2. Increase sample efficiency, i.e., reduce the number of assessments that are needed to obtain a given solution quality upon optimization.
3. Provide a more solid formalization of the properties of a robotic system that can affect the success of evolutionary optimization. In particular, we think that a finer characterization of the body-brain duality, i.e., their ability to host the cognition that the robot needs to perform its task, would be beneficial.

Bibliography

[1] Shape Change and Control of Pressure-based Soft Agents, Artificial Life Conference Proceedings, vol. ALIFE 2022: The 2022 Conference on Artificial Life (07 2022), https://doi.org/10.1162/isal_a_00520, URL https://doi.org/10.1162/isal_a_00520, 37

[2] Akinci, K., Philippides, A.: Evolving Recurrent Neural Network Controllers by Incremental Fitness Shaping. Artificial Life Conference Proceedings, vol. ALIFE 2019: The 2019 Conference on Artificial Life, pp. 416–423 (07 2019), https://doi.org/10.1162/isal_a_00196, URL https://doi.org/10.1162/isal_a_00196

[3] Albrigtsen, S.I., Imenes, A., Goodwin, M., Jiao, L., Nunavath, V.: Neuroevolution of Actively Controlled Virtual Characters - An Experiment for an Eight-Legged Character. In: Pimenidis, E., Jayne, C. (eds.) Engineering Applications of Neural Networks, pp. 94–105, Springer International Publishing, Cham (2018), ISBN 978-3-319-98204-5

[4] Auerbach, J., Aydin, D., Maesani, A., Kornatowski, P., Cieslewski, T., Heitz, G., Fernando, P., Loshchilov, I., Daler, L., Floreano, D.: RoboGen: Robot Generation through Artificial Evolution. Artificial Life Conference Proceedings, vol. ALIFE 14: The Fourteenth International Conference on the Synthesis and Simulation of Living Systems, pp. 136–137 (07 2014), https://doi.org/10.1162/978-0-262-32621-6-ch022, URL https://doi.org/10.1162/978-0-262-32621-6-ch022

[5] Auerbach, J.E., Bongard, J.C.: Evolving CPPNs to Grow Three-Dimensional Physical Structures. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, p. 627–634, GECCO '10, Association for Computing Machinery, New York, NY, USA (2010), ISBN 9781450300728, https://doi.org/10.1145/1830483.1830597, URL https://doi.org/10.1145/1830483.1830597

[6] Badwe, A.S., Gudi, R.D., Patwardhan, R.S., Shah, S.L., Patwardhan, S.C.: Detection of model-plant mismatch in MPC applications. Journal of Process Control **19**(8), 1305–1313 (2009), ISSN 0959-1524, https://doi.org/https://doi.org/10.1016/j.jprocont.2009.04.007, URL https://www.sciencedirect.com/science/article/pii/S0959152409000572, special Section on Hybrid Systems: Modeling, Simulation and Optimization

[7] Badwe, A.S., Patwardhan, R.S., Shah, S.L., Patwardhan, S.C., Gudi, R.D.: Quantifying the impact of model-plant mismatch on controller performance. Journal of Process Control **20**(4), 408–425 (2010), ISSN 0959-1524, https://doi.org/https://doi.org/10.1016/j.jprocont.2009.12.006, URL https://www.sciencedirect.com/science/article/pii/S095915240900239X

[8] Bartoli, A., Castelli, M., Medvet, E.: Weighted Hierarchical Grammatical Evolution. IEEE Transactions on Cybernetics **50**(2), 476–488 (2020), https://doi.org/10.1109/TCYB.2018.2876563

[9] Bartz-Beielstein, T., Preuss, M.: Experimental Research in Evolutionary Computation. In: Proceedings of the 9th Annual Conference Companion on Genetic and Evolutionary Computation, p. 3001–3020, GECCO '07, Association for Computing Machinery, New York, NY, USA (2007), ISBN 9781595936981, https://doi.org/10.1145/1274000.1274102, URL https://doi.org/10.1145/1274000.1274102

[10] Boeing, A., Bräunl, T.: Leveraging multiple simulators for crossing the reality gap. In: 2012 12th International Conference on Control Automation Robotics & Vision (ICARCV), pp. 1113–1119 (2012), https://doi.org/10.1109/ICARCV.2012.6485313

[11] Bojeri, A., Iacca, G.: Evolutionary optimization of drone trajectories based on optimal reciprocal collision avoidance. In: 2020 27th Conference of Open Innovations Association (FRUCT), pp. 18–26 (2020), https://doi.org/10.23919/FRUCT49677.2020.9211037

[12] Bongard, J., Lipson, H.: Once more unto the breach: Co-evolving a robot and its simulator. In: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9), pp. 57–62 (2004), https://doi.org/10.7551/mitpress/1429.003.0011

[13] Bongard, J., Zykov, V., Lipson, H.: Resilient Machines Through Continuous Self-Modeling. Science **314**(5802), 1118–1121 (2006), https://doi.org/10.1126/science.1133687, URL https://www.science.org/doi/abs/10.1126/science.1133687

[14] Bongard, J.C., Pfeifer, R.: Evolving Complete Agents using Artificial Ontogeny. In: Hara, F., Pfeifer, R. (eds.) Morpho-functional Machines: The New Species, pp. 237–258, Springer Japan, Tokyo (2003), ISBN 978-4-431-67869-4

[15] Bredeche, N., Haasdijk, E., Prieto, A.: Embodied Evolution in Collective Robotics: A Review. Frontiers in Robotics and AI **5** (2018), ISSN 2296-9144, https://doi.org/10.3389/frobt.2018.00012, URL https://www.frontiersin.org/articles/10.3389/frobt.2018.00012

[16] Bucher, D., Haspel, G., Golowasch, J., Nadim, F.: Central Pattern Generators, pp. 1–12. John Wiley & Sons, Ltd (2015), ISBN 9780470015902, https://doi.org/https://doi.org/10.1002/9780470015902.a0000032.pub2, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470015902.a0000032.pub2

[17] Bull, M., Kroo, L.A., Prakash, M.: Excitable mechanics embodied in a walking cilium. In: APS March Meeting Abstracts, APS Meeting Abstracts, vol. 2022, p. Y04.002 (Jan 2022)

[18] Cáceres, C., Rosário, J.M., Amaya, D.: Approach of kinematic control for a nonholonomic wheeled robot using artificial neural networks and genetic algorithms. In: 2017 international conference and workshop on bioinspired intelligence (IWOBI), pp. 1–6, IEEE (2017), https://doi.org/10.1109/IWOBI.2017.7985533

[19] Cáceres Flórez, C.A., Rosário, J.M., Amaya, D.: Control structure for a car-like robot using artificial neural networks and genetic algorithms. Neural Computing and Applications **32**(20), 15771–15784 (Oct 2020), ISSN 1433-3058, https://doi.org/10.1007/s00521-018-3514-1, URL https://doi.org/10.1007/s00521-018-3514-1

[20] Cazenille, L., Bredeche, N., Hamann, H., Stradner, J.: Impact of Neuron Models and Network Structure on Evolving Modular Robot Neural Network Controllers. In: Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation, p. 89–96, GECCO '12, Association for Computing Machinery, New York, NY, USA (2012), ISBN 9781450311779, https://doi.org/10.1145/2330163.2330177, URL https://doi.org/10.1145/2330163.2330177

[21] Chapelle, F., Bidaud, P.: A closed form for inverse kinematics approximation of general 6R manipulators using genetic programming. In: Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164), vol. 4, pp. 3364–3369 (2001), https://doi.org/10.1109/ROBOT.2001.933137

[22] Cheney, N., Bongard, J., Lipson, H.: Evolving Soft Robots in Tight Spaces. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, p. 935–942, GECCO '15, Association for Computing Machinery, New York, NY, USA (2015), ISBN 9781450334723, https://doi.org/10.1145/2739480.2754662, URL https://doi.org/10.1145/2739480.2754662

[23] Cheney, N., Bongard, J., SunSpiral, V., Lipson, H.: Scalable co-optimization of morphology and control in embodied machines. Journal of The Royal Society Interface **15**(143), 20170937 (2018), https://doi.org/10.1098/rsif.2017.0937, URL https://royalsocietypublishing.org/doi/abs/10.1098/rsif.2017.0937

[24] Cheney, N., MacCurdy, R., Clune, J., Lipson, H.: Unshackling Evolution: Evolving Soft Robots with Multiple Materials and a Powerful Generative Encoding. In: Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, p. 167–174, GECCO '13, Association for Computing Machinery, New York, NY, USA (2013), ISBN 9781450319638, https://doi.org/10.1145/2463372.2463404, URL https://doi.org/10.1145/2463372.2463404

[25] Chou, C.Y., Juang, C.F.: Navigation of an Autonomous Wheeled Robot in Unknown Environments Based on Evolutionary Fuzzy Control. Inventions **3**(1) (2018), ISSN 2411-5134, https://doi.org/10.3390/inventions3010003, URL https://www.mdpi.com/2411-5134/3/1/3

[26] Clune, J., Beckmann, B.E., Ofria, C., Pennock, R.T.: Evolving coordinated quadruped gaits with the HyperNEAT generative encoding. In: 2009 iEEE congress on evolutionary computation, pp. 2764–2771 (2009), https://doi.org/10.1109/CEC.2009.4983289

[27] Collins, S., Ruina, A., Tedrake, R., Wisse, M.: Efficient Bipedal Robots Based on Passive-Dynamic Walkers. Science **307**(5712), 1082–

1085 (2005), https://doi.org/10.1126/science.1107799, URL https://www.science.org/doi/abs/10.1126/science.1107799

[28] Conti, E., Madhavan, V., Such, F.P., Lehman, J., Stanley, K.O., Clune, J.: Improving Exploration in Evolution Strategies for Deep Reinforcement Learning via a Population of Novelty-Seeking Agents. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, p. 5032–5043, NIPS'18, Curran Associates Inc., Red Hook, NY, USA (2018)

[29] Corucci, F., Cheney, N., Giorgio-Serchi, F., Bongard, J., Laschi, C.: Evolving Soft Locomotion in Aquatic and Terrestrial Environments: Effects of Material Properties and Environmental Transitions. Soft Robotics **5**(4), 475–495 (2018), https://doi.org/10.1089/soro.2017.0055, URL https://doi.org/10.1089/soro.2017.0055, pMID: 29985740

[30] Custode, L.L., Iacca, G.: A co-evolutionary approach to interpretable reinforcement learning in environments with continuous action spaces. In: 2021 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–8 (2021), https://doi.org/10.1109/SSCI50451.2021.9660048

[31] D'Ambrosio, D.B., Stanley, K.O.: Generative Encoding for Multiagent Learning. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, p. 819–826, GECCO '08, Association for Computing Machinery, New York, NY, USA (2008), ISBN 9781605581309, https://doi.org/10.1145/1389095.1389256, URL https://doi.org/10.1145/1389095.1389256

[32] Das, S., Shankar, A., Aggarwal, V.: Training Spiking Neural Networks with a Multi-Agent Evolutionary Robotics Framework. In: Proceedings of the Genetic and Evolutionary Computation Conference, p. 858–865, GECCO '21, Association for Computing Machinery, New York, NY, USA (2021), ISBN 9781450383509, https://doi.org/10.1145/3449639.3459329, URL https://doi.org/10.1145/3449639.3459329

[33] De Carlo, M., Ferrante, E., Zeeuwe, D., Ellers, J., Meynen, G., Eiben, A.: Heritability in Morphological Robot Evolution. arXiv preprint arXiv:2110.11187 (2021)

[34] Di Paolo, E.: Spike-Timing Dependent Plasticity for Evolved Robots. Adapt Behav **10**(3–4), 243–263 (jul 2002), ISSN 1059-7123, https://doi.org/10.1177/1059712302919993006, URL https://doi.org/10.1177/1059712302919993006

[35] van Diggelen, F., Babuska, R., Eiben, A.: The effects of adaptive control on learning directed locomotion. In: 2020 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 2117–2124 (2020), https://doi.org/10.1109/SSCI47803.2020.9308557

[36] van Diggelen, F., Ferrante, E., Harrak, N., Luo, J., Zeeuwe, D., Eiben, A.: The Influence of Robot Traits and Evolutionary Dynamics on the Reality Gap. IEEE Transactions on Cognitive and Developmental Systems (2021), https://doi.org/10.1109/TCDS.2021.3112236

[37] Divband Soorati, M., Hamann, H.: The Effect of Fitness Function Design on Performance in Evolutionary Robotics: The Influence of a Priori

Knowledge. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, p. 153–160, GECCO '15, Association for Computing Machinery, New York, NY, USA (2015), ISBN 9781450334723, https://doi.org/10.1145/2739480.2754676, URL https://doi.org/10.1145/2739480.2754676

[38] Dolinsky, J.U., Jenkinson, I., Colquhoun, G.J.: Application of genetic programming to the calibration of industrial robots. Computers in Industry **58**(3), 255–264 (2007), ISSN 0166-3615, https://doi.org/https://doi.org/10.1016/j.compind.2006.06.003, URL https://www.sciencedirect.com/science/article/pii/S0166361506001102

[39] Doncieux, S., Bredeche, N., Mouret, J.B., Eiben, A.E.G.: Evolutionary Robotics: What, Why, and Where to. Frontiers in Robotics and AI **2** (2015), ISSN 2296-9144, https://doi.org/10.3389/frobt.2015.00004, URL https://www.frontiersin.org/articles/10.3389/frobt.2015.00004

[40] Doncieux, S., Mouret, J.B.: Behavioral diversity measures for evolutionary robotics. In: IEEE congress on evolutionary computation, pp. 1–8 (2010), https://doi.org/10.1109/CEC.2010.5586100

[41] Downing, K.L.: Adaptive Genetic Programs via Reinforcement Learning. In: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, p. 19–26, GECCO'01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001), ISBN 1558607749

[42] Eiben, A.: Real-World Robot Evolution: Why Would it (not) Work? Frontiers in Robotics and AI **8** (2021), ISSN 2296-9144, https://doi.org/10.3389/frobt.2021.696452, URL https://www.frontiersin.org/articles/10.3389/frobt.2021.696452

[43] Eiben, A.E., Hart, E.: If It Evolves It Needs to Learn. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, p. 1383–1384, GECCO '20, Association for Computing Machinery, New York, NY, USA (2020), ISBN 9781450371278, https://doi.org/10.1145/3377929.3398151, URL https://doi.org/10.1145/3377929.3398151

[44] Eiben, A.E., Smith, J.: From evolutionary computation to the evolution of things. Nature **521**(7553), 476–482 (May 2015), ISSN 1476-4687, https://doi.org/10.1038/nature14544, URL https://doi.org/10.1038/nature14544

[45] Feng, G.: A survey on analysis and design of model-based fuzzy control systems. IEEE Transactions on Fuzzy systems **14**(5), 676–697 (2006), https://doi.org/10.1109/TFUZZ.2006.883415

[46] Ferigo, A., Iacca, G., Medvet, E.: Beyond Body Shape and Brain: Evolving the Sensory Apparatus of Voxel-Based Soft Robots. In: Castillo, P.A., Jiménez Laredo, J.L. (eds.) Applications of Evolutionary Computation, pp. 210–226, Springer International Publishing, Cham (2021), ISBN 978-3-030-72699-7

[47] Ferigo, A., Iacca, G., Medvet, E., Pigozzi, F.: Evolving Hebbian Learning Rules in Voxel-based Soft Robots. IEEE Transactions on Cognitive

and Developmental Systems pp. 1–1 (2022), https://doi.org/10.1109/TCDS.2022.3226556
[48] Ferigo, A., Medvet, E., Iacca, G.: Optimizing the Sensory Apparatus of Voxel-Based Soft Robots Through Evolution and Babbling. SN Computer Science **3**(2), 109 (Dec 2021), ISSN 2661-8907, https://doi.org/10.1007/s42979-021-00987-w, URL https://doi.org/10.1007/s42979-021-00987-w
[49] Ferigo, A., Soros, L.B., Medvet, E., Iacca, G.: On the Entanglement between Evolvability and Fitness: an Experimental Study on Voxel-based Soft Robots. Artificial Life Conference Proceedings, vol. ALIFE 2022: The 2022 Conference on Artificial Life (07 2022), https://doi.org/10.1162/isal_a_00493, URL https://doi.org/10.1162/isal_a_00493, 15
[50] Floreano, D., Mattiussi, C.: Evolution of Spiking Neural Controllers for Autonomous Vision-Based Robots. In: Gomi, T. (ed.) Evolutionary Robotics. From Intelligent Robotics to Artificial Life, pp. 38–61, Springer Berlin Heidelberg, Berlin, Heidelberg (2001), ISBN 978-3-540-45502-8
[51] Floreano, D., Mondada, F.: Evolution of plastic neurocontrollers for situated agents. In: Proc. of The Fourth International Conference on Simulation of Adaptive Behavior (SAB), From Animals to Animats, ETH Zürich (1996)
[52] Francesca, G., Brambilla, M., Brutschy, A., Trianni, V., Birattari, M.: AutoMoDe: A novel approach to the automatic design of control software for robot swarms. Swarm Intelligence **8**(2), 89–112 (Jun 2014), ISSN 1935-3820, https://doi.org/10.1007/s11721-014-0092-4, URL https://doi.org/10.1007/s11721-014-0092-4
[53] Gaier, A., Ha, D.: Weight Agnostic Neural Networks. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA (2019)
[54] Gibson, J.J.: The ecological approach to visual perception: classic edition. Psychology press (2014)
[55] Glette, K., Johnsen, A.L., Samuelsen, E.: Filling the reality gap: Using obstacles to promote robust gaits in evolutionary robotics. In: 2014 IEEE International Conference on Evolvable Systems, pp. 181–186 (2014), https://doi.org/10.1109/ICES.2014.7008738
[56] Gongora, M.A., Passow, B.N., Hopgood, A.A.: Robustness analysis of evolutionary controller tuning using real systems. In: 2009 IEEE Congress on Evolutionary Computation, pp. 606–613 (2009), https://doi.org/10.1109/CEC.2009.4983001
[57] Gruau, F.: Automatic definition of modular neural networks. Adaptive behavior **3**(2), 151–183 (1994)
[58] Guo, J., Hu, P., Li, L., Wang, R.: Design of Automatic Steering Controller for Trajectory Tracking of Unmanned Vehicles Using Genetic Algorithms. IEEE Transactions on Vehicular Technology **61**(7), 2913–2924 (2012), https://doi.org/10.1109/TVT.2012.2201513
[59] Gupta, A., Savarese, S., Ganguli, S., Fei-Fei, L.: Embodied intelligence via learning and evolution. Nature Communications **12**(1), 5721 (Oct 2021),

ISSN 2041-1723, https://doi.org/10.1038/s41467-021-25874-z, URL https://doi.org/10.1038/s41467-021-25874-z

[60] Ha, D.: Reinforcement Learning for Improving Agent Design. Artificial Life **25**(4), 352–365 (11 2019), ISSN 1064-5462, https://doi.org/10.1162/artl_a_00301, URL https://doi.org/10.1162/artl_a_00301

[61] Haasdijk, E., Rusu, A.A., Eiben, A.E.: HyperNEAT for Locomotion Control in Modular Robots. In: Tempesti, G., Tyrrell, A.M., Miller, J.F. (eds.) Evolvable Systems: From Biology to Hardware, pp. 169–180, Springer Berlin Heidelberg, Berlin, Heidelberg (2010), ISBN 978-3-642-15323-5

[62] Hagras, H., Pounds-Cornish, A., Colley, M., Callaghan, V., Clarke, G.: Evolving spiking neural network controllers for autonomous robots. In: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004, vol. 5, pp. 4620–4626 (2004), https://doi.org/10.1109/ROBOT.2004.1302446

[63] Hallawa, A., Born, T., Schmeink, A., Dartmann, G., Peine, A., Martin, L., Iacca, G., Eiben, A.E., Ascheid, G.: Evo-RL: Evolutionary-Driven Reinforcement Learning. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion, p. 153–154, GECCO '21, Association for Computing Machinery, New York, NY, USA (2021), ISBN 9781450383516, https://doi.org/10.1145/3449726.3459475, URL https://doi.org/10.1145/3449726.3459475

[64] Hallawa, A., Schug, S., Iacca, G., Ascheid, G.: Evolving Instinctive Behaviour in Resource-Constrained Autonomous Agents Using Grammatical Evolution. In: Castillo, P.A., Jiménez Laredo, J.L., Fernández de Vega, F. (eds.) Applications of Evolutionary Computation, pp. 369–383, Springer International Publishing, Cham (2020), ISBN 978-3-030-43722-0

[65] Hasselmann, K., Ligot, A., Ruddick, J., Birattari, M.: Empirical assessment and comparison of neuro-evolutionary methods for the automatic off-line design of robot swarms. Nature Communications **12**(1), 4345 (Jul 2021), ISSN 2041-1723, https://doi.org/10.1038/s41467-021-24642-3, URL https://doi.org/10.1038/s41467-021-24642-3

[66] Hauser, H., Ijspeert, A.J., Füchslin, R.M., Pfeifer, R., Maass, W.: Towards a theoretical foundation for morphological computation with compliant bodies. Biological Cybernetics **105**(5), 355–370 (Dec 2011), ISSN 1432-0770, https://doi.org/10.1007/s00422-012-0471-0, URL https://doi.org/10.1007/s00422-012-0471-0

[67] Hein, D., Udluft, S., Runkler, T.A.: Interpretable policies for reinforcement learning by genetic programming. Engineering Applications of Artificial Intelligence **76**, 158–169 (2018), ISSN 0952-1976, https://doi.org/https://doi.org/10.1016/j.engappai.2018.09.007, URL https://www.sciencedirect.com/science/article/pii/S0952197618301933

[68] Hornby, G.S., Pollack, J.B.: Evolving L-systems to generate virtual creatures. Computers & Graphics **25**(6), 1041–1048 (2001), ISSN 0097-8493, https://doi.org/https://doi.org/10.1016/S0097-8493(01)

00157-1, URL https://www.sciencedirect.com/science/article/pii/S0097849301001571, artificial Life

[69] Hornby, G.S., Takamura, S., Yamamoto, T., Fujita, M.: Autonomous evolution of dynamic gaits with two quadruped robots. IEEE transactions on Robotics **21**(3), 402–410 (2005), https://doi.org/10.1109/TRO.2004.839222

[70] Hossain, D., Capi, G.: Multiobjective evolution of deep learning parameters for robot manipulator object recognition and grasping. Advanced Robotics **32**(20), 1090–1101 (2018), https://doi.org/10.1080/01691864.2018.1529620, URL https://doi.org/10.1080/01691864.2018.1529620

[71] Hu, Y., Wu, X., Geng, P., Li, Z.: Evolution strategies learning with variable impedance control for grasping under uncertainty. IEEE Transactions on Industrial electronics **66**(10), 7788–7799 (2018), https://doi.org/10.1109/TIE.2018.2884240

[72] Hupkes, E., Jelisavcic, M., Eiben, A.E.: Revolve: A Versatile Simulator for Online Robot Evolution. In: Sim, K., Kaufmann, P. (eds.) Applications of Evolutionary Computation, pp. 687–702, Springer International Publishing, Cham (2018), ISBN 978-3-319-77538-8

[73] Iovino, M., Scukins, E., Styrud, J., Ögren, P., Smith, C.: A survey of Behavior Trees in robotics and AI. Robotics and Autonomous Systems **154**, 104096 (2022), ISSN 0921-8890, https://doi.org/https://doi.org/10.1016/j.robot.2022.104096, URL https://www.sciencedirect.com/science/article/pii/S0921889022000513

[74] Iovino, M., Styrud, J., Falco, P., Smith, C.: Learning behavior trees with genetic programming in unpredictable environments. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 4591–4597 (2021), https://doi.org/10.1109/ICRA48506.2021.9562088

[75] Izhikevich, E.: Simple model of spiking neurons. IEEE Transactions on Neural Networks **14**(6), 1569–1572 (2003), https://doi.org/10.1109/TNN.2003.820440

[76] Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W.M., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., Simonyan, K., et al.: Population based training of neural networks. arXiv preprint arXiv:1711.09846 (2017)

[77] Jakobi, N.: Evolutionary robotics and the radical envelope-of-noise hypothesis. Adaptive behavior **6**(2), 325–368 (1997), ISSN 1059-7123, https://doi.org/10.1177/105971239700600205, URL https://doi.org/10.1177/105971239700600205

[78] Jakobi, N.: Minimal simulations for evolutionary robotics. Ph.D. thesis, University of Sussex (1998)

[79] Jakobi, N., Husbands, P., Harvey, I.: Noise and the reality gap: The use of simulation in evolutionary robotics. In: Morán, F., Moreno, A., Merelo, J.J., Chacón, P. (eds.) Advances in Artificial Life, pp. 704–720, Springer Berlin Heidelberg, Berlin, Heidelberg (1995), ISBN 978-3-540-49286-3

[80] Jansen, T.: Strandbeests. Architectural Design **78**(4), 22–27 (2008), https://doi.org/https://doi.org/10.1002/ad.701, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/ad.701
[81] Jelisavcic, M., Glette, K., Haasdijk, E., Eiben, A.E.: Lamarckian Evolution of Simulated Modular Robots. Frontiers in Robotics and AI **6** (2019), ISSN 2296-9144, https://doi.org/10.3389/frobt.2019.00009, URL https://www.frontiersin.org/articles/10.3389/frobt.2019.00009
[82] Jie, Y., Qi, Z., Junjie, Z., Quanjun, Y.: Survey of Evolutionary Behavior Tree Algorithm. Journal of System Simulation **33**(10), 2315 (2021)
[83] Joachimczak, M., Suzuki, R., Arita, T.: Artificial Metamorphosis: Evolutionary Design of Transforming, Soft-Bodied Robots. Artificial Life **22**(3), 271–298 (08 2016), ISSN 1064-5462, https://doi.org/10.1162/ARTL_a_00207, URL https://doi.org/10.1162/ARTL_a_00207
[84] Jones, S., Studley, M., Hauert, S., Winfield, A.: Evolving Behaviour Trees for Swarm Robotics. In: Groß, R., Kolling, A., Berman, S., Frazzoli, E., Martinoli, A., Matsuno, F., Gauci, M. (eds.) Distributed Autonomous Robotic Systems: The 13th International Symposium, pp. 487–501, Springer International Publishing, Cham (2018), ISBN 978-3-319-73008-0, https://doi.org/10.1007/978-3-319-73008-0_34, URL https://doi.org/10.1007/978-3-319-73008-0_34
[85] Juang, C.F., Chang, Y.C.: Evolutionary-Group-Based Particle-Swarm-Optimized Fuzzy Controller With Application to Mobile-Robot Navigation in Unknown Environments. IEEE Transactions on Fuzzy Systems **19**(2), 379–392 (2011), https://doi.org/10.1109/TFUZZ.2011.2104364
[86] Juang, C.F., Hsu, C.H.: Reinforcement Ant Optimized Fuzzy Controller for Mobile-Robot Wall-Following Control. IEEE Transactions on Industrial Electronics **56**(10), 3931–3940 (2009), https://doi.org/10.1109/TIE.2009.2017557
[87] Juang, C.F., Jeng, T.L., Chang, Y.C.: An Interpretable Fuzzy System Learned Through Online Rule Generation and Multiobjective ACO With a Mobile Robot Control Application. IEEE Transactions on Cybernetics **46**(12), 2706–2718 (2016), https://doi.org/10.1109/TCYB.2015.2486779
[88] Kaiser, T.K., Hamann, H.: Engineered self-organization for resilient robot self-assembly with minimal surprise. Robotics and Autonomous Systems **122**, 103293 (2019), ISSN 0921-8890, https://doi.org/https://doi.org/10.1016/j.robot.2019.103293, URL https://www.sciencedirect.com/science/article/pii/S0921889019300855
[89] Kalra, P., Mahapatra, P., Aggarwal, D.: An evolutionary approach for solving the multimodal inverse kinematics problem of industrial robots. Mechanism and Machine Theory **41**(10), 1213–1229 (2006), ISSN 0094-114X, https://doi.org/https://doi.org/10.1016/j.mechmachtheory.2005.11.005, URL https://www.sciencedirect.com/science/article/pii/S0094114X05002053
[90] Kamimura, A., Kurokawa, H., Yoshida, E., Tomita, K., Kokaji, S., Murata, S.: Distributed adaptive locomotion by a modular robotic system,

M-TRAN II. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), vol. 3, pp. 2370–2377, IEEE (2004), https://doi.org/10.1109/IROS.2004.1389763

[91] Kano, M., Shigi, Y., Hasebe, S., Ooyama, S.: Detection of Significant Model-Plant Mismatch from Routine Operation Data of Model Predictive Control System. IFAC Proceedings Volumes **43**(5), 685–690 (2010), ISSN 1474-6670, https://doi.org/https://doi.org/10.3182/20100705-3-BE-2011.00113, URL https://www.sciencedirect.com/science/article/pii/S1474667016301148, 9th IFAC Symposium on Dynamics and Control of Process Systems

[92] Khadka, S., Tumer, K.: Evolution-Guided Policy Gradient in Reinforcement Learning. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, p. 1196–1208, NIPS'18, Curran Associates Inc., Red Hook, NY, USA (2018)

[93] Klaus, G., Glette, K., Tørresen, J.: A Comparison of Sampling Strategies for Parameter Estimation of a Robot Simulator. In: Noda, I., Ando, N., Brugali, D., Kuffner, J.J. (eds.) Simulation, Modeling, and Programming for Autonomous Robots, pp. 173–184, Springer Berlin Heidelberg, Berlin, Heidelberg (2012), ISBN 978-3-642-34327-8

[94] Koos, S., Mouret, J.B., Doncieux, S.: The transferability approach: Crossing the reality gap in evolutionary robotics. IEEE Transactions on Evolutionary Computation **17**(1), 122–145 (2012), https://doi.org/10.1109/TEVC.2012.2185849

[95] Koza, J.R.: Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems, vol. 34. Stanford University, Department of Computer Science Stanford, CA (1990)

[96] Kriegman, S., Blackiston, D., Levin, M., Bongard, J.: Kinematic self-replication in reconfigurable organisms. Proceedings of the National Academy of Sciences **118**(49), e2112672118 (2021), https://doi.org/10.1073/pnas.2112672118, URL https://www.pnas.org/doi/abs/10.1073/pnas.2112672118

[97] Kriegman, S., Cheney, N., Bongard, J.: How morphological development can guide evolution. Scientific Reports **8**(1), 13934 (Sep 2018), ISSN 2045-2322, https://doi.org/10.1038/s41598-018-31868-7, URL https://doi.org/10.1038/s41598-018-31868-7

[98] Kriegman, S., Nasab, A.M., Shah, D., Steele, H., Branin, G., Levin, M., Bongard, J., Kramer-Bottiglio, R.: Scalable sim-to-real transfer of soft robot designs. In: 2020 3rd IEEE International Conference on Soft Robotics (RoboSoft), pp. 359–366 (2020), https://doi.org/10.1109/RoboSoft48309.2020.9116004

[99] Kriegman, S., Walker, S., Shah, D., Levin, M., Kramer-Bottiglio, R., Bongard, J.: Automated shapeshifting for function recovery in damaged robots. Proceedings of Robotics: Science and Systems (2019)

[100] Le Goff, L.K., Buchanan, E., Hart, E., Eiben, A.E., Li, W., De Carlo, M., Winfield, A.F., Hale, M.F., Woolley, R., Angus, M., et al.: Morpho-evolution with learning using a controller archive as an inheritance mecha-

nism. IEEE Transactions on Cognitive and Developmental Systems (2022), https://doi.org/10.1109/TCDS.2022.3148543

[101] Levin, M., Pietak, A.M., Bischof, J.: Planarian regeneration as a model of anatomical homeostasis: Recent progress in biophysical and computational approaches. Seminars in Cell & Developmental Biology **87**, 125–144 (2019), ISSN 1084-9521, https://doi.org/https://doi.org/10.1016/j.semcdb.2018.04.003, URL https://www.sciencedirect.com/science/article/pii/S1084952117301970

[102] Lindenmayer, A.: Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. Journal of Theoretical Biology **18**(3), 280–299 (1968), ISSN 0022-5193, https://doi.org/https://doi.org/10.1016/0022-5193(68)90079-9, URL https://www.sciencedirect.com/science/article/pii/0022519368900799

[103] Lipson, H., Sunspiral, V., Bongard, J., Cheney, N.: On the Difficulty of Co-Optimizing Morphology and Control in Evolved Virtual Creatures. Artificial Life Conference Proceedings, vol. ALIFE 2016, the Fifteenth International Conference on the Synthesis and Simulation of Living Systems, pp. 226–233 (07 2022), https://doi.org/10.1162/978-0-262-33936-0-ch042, URL https://doi.org/10.1162/978-0-262-33936-0-ch042

[104] Long, J.: Darwin's Devices: What Evolving Robots Can Teach Us About the History of Life and the Future of Technology. Hachette UK (2012)

[105] Luo, J., Stuurman, A.C., Tomczak, J.M., Ellers, J., Eiben, A.E.: The Effects of Learning in Morphologically Evolving Robot Systems. Frontiers in Robotics and AI **9** (2022), ISSN 2296-9144, https://doi.org/10.3389/frobt.2022.797393, URL https://www.frontiersin.org/articles/10.3389/frobt.2022.797393

[106] Mabu, S., Hirasawa, K., Hu, J.: Genetic Network Programming with Reinforcement Learning and Its Performance Evaluation. In: Deb, K. (ed.) Genetic and Evolutionary Computation – GECCO 2004, pp. 710–711, Springer Berlin Heidelberg, Berlin, Heidelberg (2004), ISBN 978-3-540-24855-2

[107] Mabu, S., Hirasawa, K., Hu, J.: A Graph-Based Evolutionary Algorithm: Genetic Network Programming (GNP) and Its Extension Using Reinforcement Learning. Evolutionary Computation **15**(3), 369–398 (09 2007), ISSN 1063-6560, https://doi.org/10.1162/evco.2007.15.3.369, URL https://doi.org/10.1162/evco.2007.15.3.369

[108] Mahdavi, S.H., Bentley, P.J.: An evolutionary approach to damage recovery of robot motion with muscles. In: European Conference on Artificial Life, pp. 248–255, Springer (2003), https://doi.org/doi.org/10.1007/b12035

[109] Marsland, S.: Machine learning: an algorithmic perspective. CRC press (2015)

[110] McGeer, T.: Passive Dynamic Walking. The International Journal of Robotics Research **9**(2), 62–82 (1990), https://doi.org/

10.1177/027836499000900206, URL https://doi.org/10.1177/027836499000900206

[111] Medvet, E., Bartoli, A., De Lorenzo, A., Fidel, G.: Evolution of Distributed Neural Controllers for Voxel-Based Soft Robots. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference, p. 112–120, GECCO '20, Association for Computing Machinery, New York, NY, USA (2020), ISBN 9781450371285, https://doi.org/10.1145/3377930.3390173, URL https://doi.org/10.1145/3377930.3390173

[112] Mégane, J., Lourenço, N., Machado, P.: Probabilistic Grammatical Evolution. In: Hu, T., Lourenço, N., Medvet, E. (eds.) Genetic Programming, pp. 198–213, Springer International Publishing, Cham (2021), ISBN 978-3-030-72812-0

[113] Miglino, O., Lund, H.H., Nolfi, S.: Evolving Mobile Robots in Simulated and Real Environments. Artificial Life **2**(4), 417–434 (07 1995), ISSN 1064-5462, https://doi.org/10.1162/artl.1995.2.4.417, URL https://doi.org/10.1162/artl.1995.2.4.417

[114] Miras, K.: Constrained by Design: Influence of Genetic Encodings on Evolved Traits of Robots. Frontiers in Robotics and AI **8** (2021), ISSN 2296-9144, https://doi.org/10.3389/frobt.2021.672379, URL https://www.frontiersin.org/articles/10.3389/frobt.2021.672379

[115] Miras, K., De Carlo, M., Akhatou, S., Eiben, A.E.: Evolving-Controllers Versus Learning-Controllers for Morphologically Evolvable Robots. In: Castillo, P.A., Jiménez Laredo, J.L., Fernández de Vega, F. (eds.) Applications of Evolutionary Computation, pp. 86–99, Springer International Publishing, Cham (2020), ISBN 978-3-030-43722-0

[116] Miras, K., Eiben, A.E.: Effects of Environmental Conditions on Evolved Robot Morphologies and Behavior. In: Proceedings of the Genetic and Evolutionary Computation Conference, p. 125–132, GECCO '19, Association for Computing Machinery, New York, NY, USA (2019), ISBN 9781450361118, https://doi.org/10.1145/3321707.3321811, URL https://doi.org/10.1145/3321707.3321811

[117] Miras, K., Ferrante, E., Eiben, A.E.: Environmental influences on evolvable robots. PLOS ONE **15**(5), 1–23 (05 2020), https://doi.org/10.1371/journal.pone.0233848, URL https://doi.org/10.1371/journal.pone.0233848

[118] Miras, K., Haasdijk, E., Glette, K., Eiben, A.E.: Search Space Analysis of Evolvable Robot Morphologies. In: Sim, K., Kaufmann, P. (eds.) Applications of Evolutionary Computation, pp. 703–718, Springer International Publishing, Cham (2018), ISBN 978-3-319-77538-8

[119] Möller, F.J.D., Bernardino, H.S., Gonçalves, L.B., Soares, S.S.R.F.: A Reinforcement Learning Based Adaptive Mutation for Cartesian Genetic Programming Applied to the Design of Combinational Logic Circuits. In: Cerri, R., Prati, R.C. (eds.) Intelligent Systems, pp. 18–32, Springer International Publishing, Cham (2020), ISBN 978-3-030-61380-8

[120] Montanier, J.M., Bredeche, N.: Embedded Evolutionary Robotics: The (1+1)-Restart-Online Adaptation Algorithm. In: Doncieux, S., Bredèche, N., Mouret, J.B. (eds.) New Horizons in Evolutionary Robotics, pp. 155–169, Springer Berlin Heidelberg, Berlin, Heidelberg (2011), ISBN 978-3-642-18272-3
[121] Mouret, J.B., Clune, J.: Illuminating search spaces by mapping elites. arXiv preprint arXiv:1504.04909 (2015)
[122] Nadizar, G., Medvet, E., Huse Ramstad, H., Nichele, S., Pellegrino, F.A., Zullich, M.: Merging pruning and neuroevolution: towards robust and efficient controllers for modular soft robots. The Knowledge Engineering Review **37**, e3 (2022), https://doi.org/10.1017/S0269888921000151
[123] Nadizar, G., Medvet, E., Miras, K.: On the Schedule for Morphological Development of Evolved Modular Soft Robots. In: Medvet, E., Pappa, G., Xue, B. (eds.) Genetic Programming, pp. 146–161, Springer International Publishing, Cham (2022), ISBN 978-3-031-02056-8
[124] Nadizar, G., Medvet, E., Nichele, S., Pontes-Filho, S.: Collective control of modular soft robots via embodied Spiking Neural Cellular Automata. arXiv preprint arXiv:2204.02099 (2022)
[125] Nadizar, G., Medvet, E., Pellegrino, F.A., Zullich, M., Nichele, S.: On the Effects of Pruning on Evolved Neural Controllers for Soft Robots. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion, p. 1744–1752, GECCO '21, Association for Computing Machinery, New York, NY, USA (2021), ISBN 9781450383516, https://doi.org/10.1145/3449726.3463161, URL https://doi.org/10.1145/3449726.3463161
[126] Najarro, E., Risi, S.: Meta-Learning through Hebbian Plasticity in Random Networks. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20, Curran Associates Inc., Red Hook, NY, USA (2020), ISBN 9781713829546
[127] Naya-Varela, M., Faina, A., Mallo, A., Duro, R.: A study of growth based morphological development in neural network controlled walkers. Neurocomputing **500**, 279–294 (2022), ISSN 0925-2312, https://doi.org/https://doi.org/10.1016/j.neucom.2021.09.082, URL https://www.sciencedirect.com/science/article/pii/S0925231222006385
[128] Neupane, A., Goodrich, M.: Learning Swarm Behaviors Using Grammatical Evolution and Behavior Trees. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, p. 513–520, IJCAI'19, AAAI Press (2019), ISBN 9780999241141
[129] Neupane, A., Goodrich, M.A.: Designing Emergent Swarm Behaviors Using Behavior Trees and Grammatical Evolution. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, p. 2138–2140, AAMAS '19, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2019), ISBN 9781450363099
[130] Nguyen, A.T., Taniguchi, T., Eciolaza, L., Campos, V., Palhares, R., Sugeno, M.: Fuzzy control systems: Past, present and future. IEEE Com-

putational Intelligence Magazine **14**(1), 56–68 (2019), https://doi.org/10.1109/MCI.2018.2881644

[131] Niekum, S., Barto, A.G., Spector, L.: Genetic Programming for Reward Function Search. IEEE Transactions on Autonomous Mental Development **2**(2), 83–90 (2010), https://doi.org/10.1109/TAMD.2010.2051436

[132] Nolfi, S., Bongard, J., Husbands, P., Floreano, D.: Evolutionary Robotics, pp. 2035–2068. Springer International Publishing, Cham (2016), ISBN 978-3-319-32552-1, https://doi.org/10.1007/978-3-319-32552-1_76, URL https://doi.org/10.1007/978-3-319-32552-1_76

[133] Nolfi, S., Floreano, D.: Learning and Evolution. Autonomous Robots **7**(1), 89–113 (Jul 1999), ISSN 1573-7527, https://doi.org/10.1023/A:1008973931182, URL https://doi.org/10.1023/A:1008973931182

[134] Nolfi, S., Parisi, D.: Learning to adapt to changing environments in evolving neural networks. Adaptive behavior **5**(1), 75–98 (1996)

[135] Norstein, E.S., Ellefsen, K.O., Glette, K.: Open-Ended Search for Environments and Adapted Agents Using MAP-Elites. In: Jiménez Laredo, J.L., Hidalgo, J.I., Babaagba, K.O. (eds.) Applications of Evolutionary Computation, pp. 651–666, Springer International Publishing, Cham (2022), ISBN 978-3-031-02462-7

[136] Nygaard, T.F., Martin, C.P., Samuelsen, E., Torresen, J., Glette, K.: Real-World Evolution Adapts Robot Morphology and Control to Hardware Limitations. In: Proceedings of the Genetic and Evolutionary Computation Conference, p. 125–132, GECCO '18, Association for Computing Machinery, New York, NY, USA (2018), ISBN 9781450356183, https://doi.org/10.1145/3205455.3205567, URL https://doi.org/10.1145/3205455.3205567

[137] O'Dowd, P.J., Winfield, A.F., Studley, M.: The distributed co-evolution of an embodied simulator and controller for swarm robot behaviours. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4995–5000 (2011), https://doi.org/10.1109/IROS.2011.6094600

[138] Olivier, L.E., Craig, I.K.: Model-plant mismatch detection and model update for a run-of-mine ore milling circuit under model predictive control. Journal of Process Control **23**(2), 100–107 (2013), ISSN 0959-1524, https://doi.org/https://doi.org/10.1016/j.jprocont.2012.09.002, URL https://www.sciencedirect.com/science/article/pii/S0959152412002193, iFAC World Congress Special Issue

[139] Pathak, D., Lu, C., Darrell, T., Isola, P., Efros, A.A.: Learning to Control Self-Assembling Morphologies: A Study of Generalization via Modularity. Curran Associates Inc., Red Hook, NY, USA (2019)

[140] Paul, S.K., Bhaumik, P.: A reinforcement learning agent based on genetic programming and universal search. In: 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 122–128 (2020), https://doi.org/10.1109/ICICCS48265.2020.9121014

[141] Pedersen, J.W., Risi, S.: Evolving and Merging Hebbian Learning Rules: Increasing Generalization by Decreasing the Number of Rules. In: Proceedings of the Genetic and Evolutionary Computation Conference, p. 892–900,

GECCO '21, Association for Computing Machinery, New York, NY, USA (2021), ISBN 9781450383509, https://doi.org/10.1145/3449639.3459317, URL https://doi.org/10.1145/3449639.3459317

[142] Peng, X.B., Andrychowicz, M., Zaremba, W., Abbeel, P.: Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 3803–3810 (2018), https://doi.org/10.1109/ICRA.2018.8460528

[143] Pfeifer, R., Bongard, J.: How the body shapes the way we think: a new view of intelligence. MIT press (2006)

[144] Pigozzi, F., Medvet, E., Bartoli, A., Rochelli, M.: Factors Impacting Diversity and Effectiveness of Evolved Modular Robots. ACM Transactions on Evolutionary Learning **3**(1), 1–33 (2023)

[145] Pigozzi, F., Tang, Y., Medvet, E., Ha, D.: Evolving Modular Soft Robots without Explicit Inter-Module Communication Using Local Self-Attention. In: Proceedings of the Genetic and Evolutionary Computation Conference, p. 148–157, GECCO '22, Association for Computing Machinery, New York, NY, USA (2022), ISBN 9781450392372, https://doi.org/10.1145/3512290.3528762, URL https://doi.org/10.1145/3512290.3528762

[146] Pontes-Filho, S., Walker, K., Najarro, E., Nichele, S., Risi, S.: A Unified Substrate for Body-Brain Co-evolution. arXiv preprint arXiv:2203.12066 (2022)

[147] Pourchot, A., Perrin, N., Sigaud, O.: Importance mixing: Improving sample reuse in evolutionary policy search methods. arXiv preprint arXiv:1808.05832 (2018)

[148] Precup, R.E., Hellendoorn, H.: A survey on industrial applications of fuzzy control. Computers in Industry **62**(3), 213–226 (2011), ISSN 0166-3615, https://doi.org/https://doi.org/10.1016/j.compind.2010.10.001, URL https://www.sciencedirect.com/science/article/pii/S0166361510001363

[149] Qiu, H., Garratt, M., Howard, D., Anavatti, S.: Towards Crossing the Reality Gap with Evolved Plastic Neurocontrollers. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference, p. 130–138, GECCO '20, Association for Computing Machinery, New York, NY, USA (2020), ISBN 9781450371285, https://doi.org/10.1145/3377930.3389843, URL https://doi.org/10.1145/3377930.3389843

[150] Regan, W., Van Breugel, F., Lipson, H.: Towards evolvable hovering flight on a physical ornithopter. Alife X, Bloomington, USA (2006)

[151] Reuter, J., Steup, C., Mostaghim, S.: Genetic Programming-Based Inverse Kinematics for Robotic Manipulators. In: Medvet, E., Pappa, G., Xue, B. (eds.) Genetic Programming, pp. 130–145, Springer International Publishing, Cham (2022), ISBN 978-3-031-02056-8

[152] Reyes, P., Escobar, M.J.: Neuroevolutive Algorithms for Learning Gaits in Legged Robots. IEEE Access **7**, 142406–142420 (2019), https://doi.org/10.1109/ACCESS.2019.2944545

[153] Rosenzweig, M.R., Breedlove, S.M., Leiman, A.L.: Biological psychology: An introduction to behavioral, cognitive, and clinical neuroscience. Sinauer Associates (2002)

[154] Rothlauf, F.: On the Locality of Representations. In: Proceedings of the 2003 International Conference on Genetic and Evolutionary Computation: PartII, p. 1608–1609, GECCO'03, Springer-Verlag, Berlin, Heidelberg (2003), ISBN 3540406034

[155] Rothlauf, F.: Representations for Genetic and Evolutionary Algorithms, pp. 9–32. Springer Berlin Heidelberg, Berlin, Heidelberg (2006), ISBN 978-3-540-32444-7, https://doi.org/10.1007/3-540-32444-5_2, URL https://doi.org/10.1007/3-540-32444-5_2

[156] Roy, K., Jaiswal, A., Panda, P.: Towards spike-based machine intelligence with neuromorphic computing. Nature **575**(7784), 607–617 (Nov 2019), ISSN 1476-4687, https://doi.org/10.1038/s41586-019-1677-2, URL https://doi.org/10.1038/s41586-019-1677-2

[157] Salimans, T., Ho, J., Chen, X., Sidor, S., Sutskever, I.: Evolution strategies as a scalable alternative to reinforcement learning. arXiv preprint arXiv:1703.03864 (2017)

[158] Salvato, E., Fenu, G., Medvet, E., Pellegrino, F.A.: Characterization of Modeling Errors Affecting Performances of a Robotics Deep Reinforcement Learning Controller in a Sim-to-Real Transfer. In: 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO), pp. 1154–1159 (2021)

[159] Salvato, E., Fenu, G., Medvet, E., Pellegrino, F.A.: Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning. IEEE Access **9**, 153171–153187 (2021), https://doi.org/10.1109/ACCESS.2021.3126658

[160] Samuelsen, E., Glette, K.: Some Distance Measures for Morphological Diversification in Generative Evolutionary Robotics. In: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, p. 721–728, GECCO '14, Association for Computing Machinery, New York, NY, USA (2014), ISBN 9781450326629, https://doi.org/10.1145/2576768.2598325, URL https://doi.org/10.1145/2576768.2598325

[161] Sasaki, H., Kubota, N.: Virus-evolutionary genetic algorithm for fuzzy spiking neural network of a mobile robot in a dynamic environment. In: 2006 SICE-ICASE International Joint Conference, pp. 4214–4219 (2006), https://doi.org/10.1109/SICE.2006.314773

[162] Sasaki, H., Kubota, N.: Distributed behavior learning of multiple mobile robots based on spiking neural network and steady-state genetic algorithm. In: 2009 IEEE Workshop on Robotic Intelligence in Informationally Structured Space, pp. 73–78 (2009), https://doi.org/10.1109/RIISS.2009.4937909

[163] Scheper, K.Y.W., Tijmons, S., de Visser, C.C., de Croon, G.C.H.E.: Behavior Trees for Evolutionary Robotics†. Artificial Life **22**(1), 23–48 (02

2016), ISSN 1064-5462, https://doi.org/10.1162/ARTL_a_00192, URL https://doi.org/10.1162/ARTL_a_00192
[164] Selvanathan, S., Tangirala, A.K.: Diagnosis of Poor Control Loop Performance Due to Model-Plant Mismatch. Industrial & Engineering Chemistry Research **49**(9), 4210–4229 (May 2010), ISSN 0888-5885, https://doi.org/10.1021/ie900769v, URL https://doi.org/10.1021/ie900769v
[165] Seriani, S., Marcini, L., Caruso, M., Gallina, P., Medvet, E.: Crowded Environment Navigation with NEAT: Impact of Perception Resolution on Controller Optimization. Journal of Intelligent & Robotic Systems **101**(2), 36 (Feb 2021), ISSN 1573-0409, https://doi.org/10.1007/s10846-020-01308-8, URL https://doi.org/10.1007/s10846-020-01308-8
[166] Sigaud, O., Stulp, F.: Policy search in continuous action domains: An overview. Neural Networks **113**, 28–40 (2019), ISSN 0893-6080, https://doi.org/https://doi.org/10.1016/j.neunet.2019.01.011, URL https://www.sciencedirect.com/science/article/pii/S089360801930022X
[167] Sims, K.: Evolving 3D Morphology and Behavior by Competition. Artificial Life **1**(4), 353–372 (07 1994), ISSN 1064-5462, https://doi.org/10.1162/artl.1994.1.4.353, URL https://doi.org/10.1162/artl.1994.1.4.353
[168] Sims, K.: Evolving Virtual Creatures. In: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, p. 15–22, SIGGRAPH '94, Association for Computing Machinery, New York, NY, USA (1994), ISBN 0897916670, https://doi.org/10.1145/192161.192167, URL https://doi.org/10.1145/192161.192167
[169] Sipper, M., Sanchez, E., Mange, D., Tomassini, M., Perez-Uribe, A., Stauffer, A.: A phylogenetic, ontogenetic, and epigenetic view of bio-inspired hardware systems. IEEE Transactions on Evolutionary Computation **1**(1), 83–97 (1997), https://doi.org/10.1109/4235.585894
[170] Song, X., Yang, Y., Choromanski, K., Caluwaerts, K., Gao, W., Finn, C., Tan, J.: Rapidly adaptable legged robots via evolutionary meta-learning. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3769–3776 (2020), https://doi.org/10.1109/IROS45743.2020.9341571
[171] Stanley, K.O.: Compositional pattern producing networks: A novel abstraction of development. Genetic Programming and Evolvable Machines **8**(2), 131–162 (Jun 2007), ISSN 1573-7632, https://doi.org/10.1007/s10710-007-9028-8, URL https://doi.org/10.1007/s10710-007-9028-8
[172] Stanley, K.O., D'Ambrosio, D.B., Gauci, J.: A Hypercube-Based Encoding for Evolving Large-Scale Neural Networks. Artificial Life **15**(2), 185–212 (04 2009), ISSN 1064-5462, https://doi.org/10.1162/artl.2009.15.2.15202, URL https://doi.org/10.1162/artl.2009.15.2.15202
[173] Stanley, K.O., Miikkulainen, R.: Efficient Reinforcement Learning through Evolving Neural Network Topologies. In: Proceedings of the 4th An-

nual Conference on Genetic and Evolutionary Computation, p. 569–577, GECCO'02, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2002), ISBN 1558608788

[174] Stanley, K.O., Miikkulainen, R.: Evolving Neural Networks through Augmenting Topologies. Evolutionary Computation **10**(2), 99–127 (06 2002), ISSN 1063-6560, https://doi.org/10.1162/106365602320169811, URL https://doi.org/10.1162/106365602320169811

[175] Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)

[176] Talamini, J., Medvet, E., Bartoli, A., De Lorenzo, A.: Evolutionary Synthesis of Sensing Controllers for Voxel-based Soft Robots. Artificial Life Conference Proceedings, vol. ALIFE 2019: The 2019 Conference on Artificial Life, pp. 574–581 (07 2019), https://doi.org/10.1162/isal_a_00223, URL https://doi.org/10.1162/isal_a_00223

[177] Tang, Y., Nguyen, D., Ha, D.: Neuroevolution of Self-Interpretable Agents. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference, p. 414–424, GECCO '20, Association for Computing Machinery, New York, NY, USA (2020), ISBN 9781450371285, https://doi.org/10.1145/3377930.3389847, URL https://doi.org/10.1145/3377930.3389847

[178] Tavanaei, A., Ghodrati, M., Kheradpisheh, S.R., Masquelier, T., Maida, A.: Deep learning in spiking neural networks. Neural Networks **111**, 47–63 (2019), ISSN 0893-6080, https://doi.org/https://doi.org/10.1016/j.neunet.2018.12.002, URL https://www.sciencedirect.com/science/article/pii/S0893608018303332

[179] Téllez, R.A., Angulo, C., Pardo, D.E.: Evolving the Walking Behaviour of a 12 DOF Quadruped Using a Distributed Neural Architecture. In: Ijspeert, A.J., Masuzawa, T., Kusumoto, S. (eds.) Biologically Inspired Approaches to Advanced Information Technology, pp. 5–19, Springer Berlin Heidelberg, Berlin, Heidelberg (2006), ISBN 978-3-540-32438-6

[180] Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain randomization for transferring deep neural networks from simulation to the real world. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp. 23–30 (2017), https://doi.org/10.1109/IROS.2017.8202133

[181] Tuci, E., Massera, G., Nolfi, S.: Active categorical perception in an evolved anthropomorphic robotic arm. In: 2009 IEEE Congress on Evolutionary Computation, pp. 31–38 (2009), https://doi.org/10.1109/CEC.2009.4982927

[182] Urzelai, J., Floreano, D.: Evolutionary Robotics: Coping with Environmental Change. In: Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation, p. 941–948, GECCO'00, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2000), ISBN 1558607080

[183] Valsalam, V.K., Miikkulainen, R.: Modular Neuroevolution for Multilegged Locomotion. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, p. 265–272, GECCO '08, Association for

Computing Machinery, New York, NY, USA (2008), ISBN 9781605581309, https://doi.org/10.1145/1389095.1389136, URL https://doi.org/10.1145/1389095.1389136

[184] Varela, F.J., Bourgine, P.: Toward a practice of autonomous systems: Proceedings of the First European Conference on Artificial Life. MIT press (1992)

[185] Veenstra, F., Faina, A., Risi, S., Stoy, K.: Evolution and Morphogenesis of Simulated Modular Robots: A Comparison Between a Direct and Generative Encoding. In: Squillero, G., Sim, K. (eds.) Applications of Evolutionary Computation, pp. 870–885, Springer International Publishing, Cham (2017), ISBN 978-3-319-55849-3

[186] Wang, M., Luo, J., Fang, J., Yuan, J.: Optimal trajectory planning of free-floating space manipulator using differential evolution algorithm. Advances in Space Research **61**(6), 1525–1536 (2018), ISSN 0273-1177, https://doi.org/https://doi.org/10.1016/j.asr.2018.01.011, URL https://www.sciencedirect.com/science/article/pii/S0273117718300346

[187] Wang, R., Lehman, J., Clune, J., Stanley, K.O.: Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions. arXiv preprint arXiv:1901.01753 (2019)

[188] Wen, R., Guo, Z., Zhao, T., Ma, X., Wang, Q., Wu, Z.: Neuroevolution of augmenting topologies based musculor-skeletal arm neurocontroller. In: 2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), pp. 1–6 (2017), https://doi.org/10.1109/I2MTC.2017.7969727

[189] Willyard, C.: New human gene tally reignites debate. Nature **558**(7710), 354–356 (2018)

[190] Xu, S., Moriguch, H., Honiden, S.: Sample efficiency analysis of neuroevolution algorithms on a quadruped robot. In: 2013 IEEE Congress on Evolutionary Computation, pp. 2170–2177 (2013), https://doi.org/10.1109/CEC.2013.6557826

[191] Zador, A.M.: A critique of pure learning and what artificial neural networks can learn from animal brains. Nature Communications **10**(1), 3770 (Aug 2019), ISSN 2041-1723, https://doi.org/10.1038/s41467-019-11786-6, URL https://doi.org/10.1038/s41467-019-11786-6

[192] Zagal, J.C., del Solar, J.R., Vallejos, P.: Back to reality: Crossing the reality gap in evolutionary robotics. IFAC Proceedings Volumes **37**(8), 834–839 (2004), ISSN 1474-6670, https://doi.org/https://doi.org/10.1016/S1474-6670(17)32084-0, URL https://www.sciencedirect.com/science/article/pii/S1474667017320840, iFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal, 5-7 July 2004

[193] Zhang, H., Zhou, A., Lin, X.: Interpretable policy derivation for reinforcement learning based on evolutionary feature synthesis. Complex & Intelligent Systems **6**(3), 741–753 (Oct 2020), ISSN 2198-6053, https:

//doi.org/10.1007/s40747-020-00175-y, URL https://doi.org/10.1007/s40747-020-00175-y

[194] Zhang, T., Zhang, W., Gupta, M.M.: An underactuated self-reconfigurable robot and the reconfiguration evolution. Mechanism and Machine Theory **124**, 248–258 (2018), ISSN 0094-114X, https://doi.org/https://doi.org/10.1016/j.mechmachtheory.2018.03.004, URL https://www.sciencedirect.com/science/article/pii/S0094114X17314040

[195] Zhao, A., Xu, J., Konaković-Luković, M., Hughes, J., Spielberg, A., Rus, D., Matusik, W.: RoboGrammar: Graph Grammar for Terrain-Optimized Robot Design. ACM Trans. Graph. **39**(6) (nov 2020), ISSN 0730-0301, https://doi.org/10.1145/3414685.3417831, URL https://doi.org/10.1145/3414685.3417831

[196] Zou, X., Scott, E., Johnson, A., Chen, K., Nitz, D., De Jong, K., Krichmar, J.: Neuroevolution of a Recurrent Neural Network for Spatial and Working Memory in a Simulated Robotic Environment. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion, p. 289–290, GECCO '21, Association for Computing Machinery, New York, NY, USA (2021), ISBN 9781450383516, https://doi.org/10.1145/3449726.3459565, URL https://doi.org/10.1145/3449726.3459565