

Evolutionary Synthesis of Sensing Controllers for Voxel-based Soft Robots

Jacopo Talamini², Eric Medvet^{1,2}, Alberto Bartoli¹, and Andrea De Lorenzo¹

¹Machine Learning Lab, Department of Engineering and Architecture, Università di Trieste, Italy

²Evolutionary Robotics and Artificial Life Lab, Department of Engineering and Architecture, Università di Trieste, Italy
emedvet@units.it

Abstract

Soft robots allow for interesting morphological and behavioral designs because they exhibit more degrees of freedom than robots composed of rigid parts. In particular, *voxel-based soft robots (VSRs)*—aggregations of elastic cubic building blocks—have attracted the interest of Robotics and Artificial Life researchers. VSRs can be controlled by changing the volume of individual blocks: simple, yet effective controllers that do not exploit the feedback of the environment, have been automatically designed by means of Evolutionary Algorithms (EAs).

In this work we explore the possibility of evolving *sensing controllers* in the form of artificial neural networks: we hence allow the robot to sense the environment in which it moves. Although the search space for a sensing controller is larger than its non-sensing counterpart, we show that effective sensing controllers can be evolved which realize interesting locomotion behaviors. We also experimentally investigate the impact of the VSR morphology on the effectiveness of the search and verify that the sensing controllers are indeed able to exploit their sensing ability for better solving the locomotion task.

Introduction

Traditionally, robots have been made using rigid parts connected by joints. This allowed engineers to model robots behaviour and eased the design of body and controllers for the robots. On the other hand, creatures in nature are composed also, or mainly, of soft tissues and are quite effective in solving many complex tasks which are still utterly hard for robots (Kim et al., 2013). Inspired by nature (Lin et al., 2011), in the recent years many researchers focused on robots made on soft tissues, called *soft robots* (Rus and Tolley, 2015). The efforts concerned methods for the assisted or automated design of soft robot bodies (Cheney et al., 2013, 2014) and controllers (Braganza et al., 2007; Vaughan, 2018), often by means of simulation, and techniques for building actual soft robots (Iida and Laschi, 2011; Shepherd et al., 2011).

Voxel-based Soft Robots (VSRs) are a particular category of soft robots. They are aggregations of small elastic cubic building blocks called *voxels* (Hiller and Lipson, 2012).

VSRs have been important for the raise of the *embodied cognition paradigm* according to which the complexity of behavior of a (virtual) creature depends on both its brain and its body (Pfeifer and Bongard, 2006). According to this paradigm, a robot should be designed by considering brain and body together rather than by focusing only on its brain, i.e., on its controller. This research path has been particularly significant for VSRs, within a common framework in which the ability of the VSR to interact with the environment derived mainly from its body (Cheney et al., 2013, 2014).

In this paper, we explore the possibility of automatically synthesizing *sensing controllers* for simple VSRs, i.e., controllers which can sense the environment and exploit the gathered information for guiding the robot movements. We consider VSRs in which the sensing is distributed across the full body, i.e., on each voxel composing the VSR. In other words, we consider a VSR as an aggregation of simple parts that can be used both as actuators and as sensors.

We consider three different VSRs, i.e., with different bodies, and synthesize the corresponding controllers for solving a locomotion task. For each VSR we evolved a sensing controller and a more traditional, non-sensing controller. We represent sensing controllers as artificial neural networks (ANNs) whose topology is determined by the body of the robot, while for non-sensing controller we use a simpler representation which has already been successfully adopted (Kriegman et al., 2018). We synthesize both kinds of controllers with the same EA where, as we will show, the sensing controller corresponds to a larger search space than the non-sensing one, having more parameters. We evolved each VSR in two different environments, i.e. an even surface and an uneven surface.

Our experimental results, obtained by simulation, show that sensing controllers are always more effective than non-sensing ones, regardless of the body of the VSR and of the environment in which they evolved. Moreover, we also find that sensing controllers exhibit behaviors that are more heterogeneous than those of their non-sensing counterparts. Most importantly, we also assess the behavior of controllers in environments *different* from those in which they were

evolved and found that sensing controllers are more effective even in such scenarios. This result suggests that sensing controllers are indeed able to exploit their peculiar ability to sense the environment in which they are immersed.

Related work

The idea of evolving the body and the controller of simulated creatures dates back to '90s (Sims, 1994). In the cited work, the creature's body is modular, and the controller, in the form of an ANN, is distributed among their body components, capable of sensing the environment.

Other attempts to optimize ANNs controlling soft robots have been done later. For instance, Braganza et al. (2007) consider a tentacle-like manipulator which is controlled by an ANN, since the design of a traditional closed-loop controller for this specific robot was considered unfeasible. Another example is the optimization of a locomotion controller in the form of an ANN for a quadruped simulated creature (Vaughan, 2018).

On the other hand, control strategies different than ANNs have led to interesting results. Bruder et al. (2019), for example, have recently designed a linear dynamical model for controlling soft robots, based on a data-driven model: the authors claim that the proposed method, being more traditional and control-oriented, avoids issues of the ANNs acting as the black-boxes.

We remark that the works cited above face the problem of sensing, but are not based on VSRs. Research on VSRs focused more on how to design (often by means of evolutionary computation) the body of the robot: when the controller was of a non-trivial complexity, it had no sensing ability. Nevertheless, interesting behaviors have been found.

First attempts of morphological optimization of VSRs were done by Hiller and Lipson (2012) and, later, by Cheney et al. (2013). In the latter work, the novelty was mainly in the representation of the morphology and in the corresponding EA, both achieved with CPPN-NEAT (Neuroevolution of Augmented Topologies applied to Compositional Pattern-Producing Networks, Stanley (2007)): because of their ability to compactly describe patterns with repetitions and symmetries (which resemble nature), CPPN proved to be useful for evolving effective VSR morphologies. In that case, the task was locomotion and the controller was actually determined by the morphology, since different materials statically corresponded to different actuations. A similar approach has been applied later by Cheney et al. (2015) for evolving VSRs able to escape from a tight space.

A different kind of control of the VSR, but still not able to sense the environment, has been studied by Cheney et al. (2014). The authors proposed to define materials for the voxels in terms of their ability to propagate and react to an activation signal, inspired by properties of real, biological tissues. Morphologies were then evolved with CPPN-NEAT for the locomotion task.

Materials composing VSRs, in particular soft vs. stiff ones, are also the focus of (Bongard et al., 2016). The authors implemented a distributed growth mechanism, in place of actuation by oscillating global signals. The development of VSRs is allowed during their entire life span, acting at a lower time scale than the oscillation. The task is inspired by plants, and consists in growing towards static (possibly multiple) source of light in the environment, thus allowing the VSRs the ability to sense to a certain extent.

VSRs have been used as a case of study also for reasoning about the evolution in different environment (Corucci et al., 2018). The authors of the cited work evolved morphologies on a land environment in comparison with the ones in a water environment. Subsequently, they investigated the effects of an environmental transition, from land to water and the opposite, during the evolution, and they try to explain morphological results. To some degree, we too experiment with VSRs facing different environment: we assess their ability to move in environments which were not seen during the evolution and we show that sensing is beneficial in this scenario.

Scenario: controlling VSRs

Voxel-based soft robots (VSRs)

A *voxel-based soft robot* (VSR) is an assembly of one or more *voxels*, i.e., cubic building blocks, each linked to up to 6 neighbour voxels. Voxels are also elastic in the sense that their volume may either contract or expand with respect to the resting volume; the volume of each voxel may vary independently of the volume of any other voxel. We consider VSRs composed of a predefined number of voxels n . The *morphology* of a VSR is the way in which its voxels are linked.

We assume a discrete-time physics model in which scale values are set at regular intervals $t = k\Delta t$, $k \in \mathbb{N}$, where Δt is a parameter.

At any time, each voxel is defined by $s, \mathbf{x}, \mathbf{v}, \mathbf{v}'$, where: s is the *scale*, i.e., the ratio between the current and resting volume of the voxel; \mathbf{x} , \mathbf{v} , and \mathbf{v}' are the position, velocity, and acceleration of its center.

The behavior of the robot can be determined by imposing a value for the scale of each of its voxels. By varying the scale for each voxel over time, the corresponding positions, velocities and accelerations will vary over time as well depending on how voxels are linked together. In this work we use the physics model presented by Kriegman et al. (2017). The behavior of the VSR derives hence from the positions, velocities, and acceleration of its composing voxels, which themselves derive from the values imposed to the scale. We call *controller* of the VSR the way in which scale values are set over the time.

In general, a controller may set the values of the scale over the time basing on external input related to the interactions of the VSR with the environment; or, it may set the

scale regardless of those interactions. We call the two approaches *sensing* and *non-sensing* controllers, respectively. In the next sections we describe the two specific controllers that we consider in this work.

Non-sensing controller

We consider the simple non-sensing controller proposed by Kriegman et al. (2018) in which the scale s_i of the i -th voxel varies over time according to a sinusoidal signal, which determines the relative scale with respect to a resting value:

$$s_i(k) = s_i^0 + a \sin(2\pi f k \Delta t + \phi_i) \quad (1)$$

Frequency f and amplitude a are predefined and identical for all the voxels. Phase ϕ_i and resting value s_i^0 are instead defined separately for each voxel and constitute the parameters $\theta_{\text{NS}} = (s_1^0, \phi_1, \dots, s_n^0, \phi_n)$ of the controller. It can be seen, hence, that the number of parameters of this non-sensing controller, and therefore the size of the space of the corresponding controller instances, grows linearly with the number n of voxels in the VSR, i.e., $|\theta_{\text{NS}}| = 2n \sim O(n)$.

Sensing controller

We consider a sensing controller in which the VSR senses the environment in terms of the actual scale, velocity, and acceleration of each of its voxels: since these figures are determined also by how the VSR interacts with the environment, e.g., by pushing on the floor, they correspond to sensing the environment. These inputs, along with a single sinusoidal signal $\sin(2\pi f k \Delta t)$, are fed to a feed-forward ANN whose output layer determines the values of the scale to be set for each of the voxels.

More in detail, the ANN is composed of an input layer of $3n + 1$ neurons (the $+1$ being fed with the sinusoidal signal), an hidden layer of h neurons, and an output layer of n neurons. The activation function is the *Rectified Linear Unit* (ReLU). The input layer is fed with the values $s_1(k-1)$, $\|\mathbf{v}_1(k-1)\|$, $\|\mathbf{v}'_1(k-1)\|$ of each voxel. Each output neuron emits a value $o_i \in [-1, 1]$ which is then mapped to $[s^0 - \Delta s, s^0 + \Delta s]$, where s_0 and Δs are pre-defined values which are the same for all the voxels. The output of the i -th neuron at time $k\Delta t$ determines the scale $s_i(k)$ of the i -th voxel:

$$s_i(k) = s^0 + \Delta s o_i \quad (2)$$

$$o_i = \mathbf{f}_i(s_1(k-1), \|\mathbf{v}_1(k-1)\|, \|\mathbf{v}'_1(k-1)\|, \dots, s_n(k-1), \|\mathbf{v}_n(k-1)\|, \|\mathbf{v}'_n(k-1)\|); \theta_S) \quad (3)$$

where $\|\mathbf{v}_i(k-1)\|$ is the norm of the velocity of the i -th voxel at time $(k-1)\Delta t$, $\mathbf{f} : \mathbb{R}^{3n+1} \rightarrow [0, 1]^n$ represents the ANN, and θ_S are the ANN parameters (i.e., weights).

Concerning the number of neurons in the hidden layer, we set $h = 0.65n$. It can be seen that the number of parameters of this sensing controller grows with n^2 , i.e., $|\theta_S| = 3(n+1)h + hn \sim O(n^2)$.

Instantiating the controller

We instantiate the two controllers, i.e., we determine the values for their parameters θ_{NS} and θ_S , by means of evolutionary computation. To this end, we use for both controllers the Evolutionary Algorithm (EA) shown in Algorithm 1, already used by Kriegman et al. (2018) for evolving a non-sensing controller. This EA evolves a fixed size of n_{pop} individuals for n_{gen} generations, each individual being a vector θ of values ($\theta = \theta_{\text{NS}}$ and $\theta = \theta_S$ for the non-sensing and for the sensing controller, respectively). Only a unary genetic operator (mutation) is used: the mutation consists in perturbing each parameter in θ with probability p_{mut} , the amount of perturbation being with a random value randomly sampled from a normal distribution $N(0, \sigma_{\text{mut}})$. When evolving the non-sensing controller, we limit the values of each $s_i^0 \in \theta_{\text{NS}}$ parameter, after the mutation, to the interval $[s^0 - \Delta s, s^0 + \Delta s]$.

The generational model is a $n + m$ with overlapping and individuals are compared using Pareto dominance applied on their fitness and age: the age of the individual is incremented at each generation, whereas new individuals have the age set to 0. In case of tie in a selection (i.e., when one individual has to be selected from a set of individuals on the same Pareto front), individuals with the best fitness are preferred; in case of further tie, the individual is chosen at random. The same criterion is used to determine the *best individual* at the end of the evolution.

```

1  $P \leftarrow \emptyset$ 
2 foreach  $i \in \{1, \dots, n_{\text{pop}}\}$  do
3    $P \leftarrow P \cup (\text{random}(), 0)$ 
4 end
5 foreach  $i \in \{1, \dots, n_{\text{gen}}\}$  do
6    $P' \leftarrow \emptyset$ 
7   foreach  $(\theta, a) \in P$  do
8      $\theta' \leftarrow \text{mutate}(\theta)$ 
9      $P' \leftarrow P' \cup (\theta, a + 1)$ 
10     $P' \leftarrow P' \cup (\theta', a + 1)$ 
11  end
12   $P' \leftarrow P' \cup (\text{random}(), 0)$ 
13   $P \leftarrow \text{select}(P', n_{\text{pop}})$ 
14 end

```

Algorithm 1: The EA for evolving the controller.

The fitness of an individual θ , i.e., a controller for a VSR, measures its ability to perform a given task. In this work, we consider the locomotion task and set the fitness to the distance that the VSR corresponding to the individual travels along the x -axis during a simulation of a predefined amount of n_{sim} time steps. Despite its apparent simplicity, locomotion is considered a benchmark for VSRs (Cheney et al., 2013, 2014; Kriegman et al., 2018).

We remark that other techniques might be used for the purpose of instantiating a controller, given a morphology

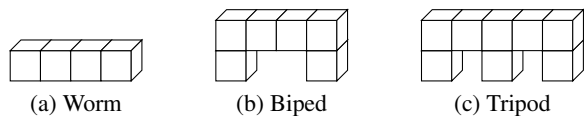


Figure 1: The three different VSR morphologies.

and a simulator. In particular, for learning the sensing-controller, which is based on ANN, EAs operating on ANNs might be used, e.g., NEAT (Stanley and Miikkulainen, 2002) or CPPN-NEAT (Stanley, 2007). Or, since the considered scenario consists in an autonomous agent that interacts with the environment trying to maximizing a reward (here, the traveled distance), Reinforcement Learning techniques might be used (Duan et al., 2016). However, we leave the exploration of these alternative options to future work, since here we are interested in comparing the nature of the controller, and the information it can exploit, rather than the learning technique.

Experiments and results

We performed an experimental evaluation aimed at investigating the effectiveness of the sensing controller with respect to the non-sensing one. In particular, we aimed at answering the following research questions: (RQ1) Is a sensing controller better than a non-sensing one? (RQ2) Does the larger size of the search space for the sensing controller affect the search effectiveness? (RQ3) Is a sensing controller actually able to exploit its ability to sense the environment? For answering these questions, we considered three different VSR *morphologies* and two different *environments*.

Morphologies are shown in Figure 1: we call the corresponding VSRs *worm*, *biped*, and *tripod*. They differ in the number of composing voxels ($n \in \{4, 6, 8\}$) and hence correspond to different numbers of parameters for defining the controllers.

Concerning the environment, we simulated the movement of the VSR on an *even* surface and on an *uneven* surface. In all cases, we performed 30 evolutionary runs (i.e., 30 independent executions of Algorithm 1) for each combination of morphology and environment. We used the implementation made available by Kriegman et al. (2018)¹, with the parameters of the physics model, morphologies, and EA shown in Table 1. We run the experiments using AWS EC2 on the c4.8xlarge EC2 instances, each equipped with 36 vCPU based on 2.9 GHz Intel Xeon E5-2666 and with 60 GB RAM; we distributed the fitness evaluation across the vCPUs and runs across instances.

In each run, the VSR was put in the environment with its main dimension laying on the x -axis, the same axis along

¹<https://github.com/skriegman/how-devo-can-guide-evo>

Param.	Value	Param.	Value
Δt	0.14 ms	n_{pop}	30
a	0.001 m ³	n_{gen}	200
f	40 Hz	p_{mut}	$\frac{1}{ \theta }$
s^0	0.01 m ³	σ_{mut}	1
Δs	0.001 m ³	n_{sim}	10 000

Table 1: Parameters of the physics model (top left), morphologies (bottom left), and EA (right) used in the experiments.

Table 2: Fitness (in mm, mean μ and standard deviation σ across the 30 runs) of the best individual at the end of the evolution in the environment with even surface. The p -value is computed with the Mann-Whitney U-test (see text).

Morph.	Non-sensing		Sensing		p -value [$\times 10^{-3}$]
	μ	σ	μ	σ	
Worm	146	8	3012	329	0.002
Biped	69	19	931	74	0.006
Tripod	550	26	636	76	0.024

which the traveled distance is measured for computing the fitness.

Environment: even surface

Table 2 presents the main results obtained in the environment with even surface, with the three morphologies. The table shows the mean μ and the standard deviation σ of the fitness of the best individual at the last generation across the 30 runs. The table also shows the p -values obtained with the Mann-Whitney U-test that we performed for each morphology in order to verify if the samples have the same median.

The foremost finding is that sensing controllers clearly outperform non-sensing ones. That is, a VSR controlled by a sensing controller is in general better in performing the locomotion task, regardless of the morphology. The difference is always statistically significant (with a significance level of $\alpha = 0.05$) and substantial in two on three cases, the worm and the biped.

Concerning the tripod, the sensing controller is still better, in terms of the final best fitness, than the non-sensing one, but the difference is lower (636 ± 76 vs. 550 ± 263) with respect to the worm and biped (for which traveled distance difference is of an order of magnitude). We interpret this finding as a consequence of the fact that the number of voxels in the tripod is larger: the complexity of the controller is $O(n)$ for the non-sensing case and $O(n^2)$ for the sensing case, and the same applies for the size of the search space. As a further evidence for this interpretation, we show in Figure 2 how the fitness of the best individual varies during the evolution (mean across the 30 runs) for the

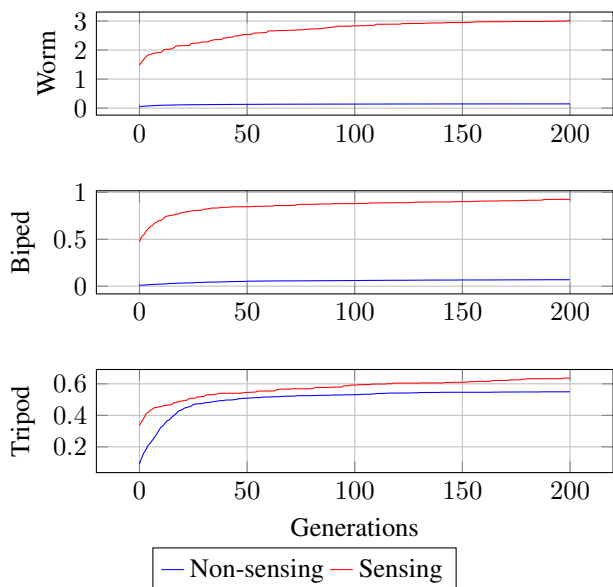


Figure 2: Fitness (in m, mean across the 30 runs) of the best individual during the evolution.

three morphologies. Beyond highlighting the lower difference for the tripod, Figure 2 suggests that the evolution of a sensing controller has not yet stopped at the end of the evolution (200-th generation), for this case; on the other end, this does not occur with the non-sensing controller. In other words, the larger search space makes finding the optimum harder. We remark, however, that other techniques exist for evolving ANNs which are suitable for scenarios like the one considered in this work. In particular, we argue that NEAT (or its recent variants as, e.g., the one of Silva et al. (2015)) might be a way to address the issue of the large search space, thanks to its ability to progressively increase the expressiveness of the representation—i.e., complexification.

Analysis of the behaviors In order to further investigate the differences between the sensing and non-sensing controllers, we observed the resulting behaviors during the simulations: i.e., we looked at the way best evolved controllers moved and drawn qualitative reasoning (see Figure 3). We found that sensing controllers resulted, in general, in a broader set of behaviors, the difference being more apparent for the worm. Interestingly, for this morphology the behaviors exhibited by the sensing controllers often visually resembled those of the real biological counterpart.

In an attempt of quantifying the result of this qualitative analysis, we devised a way of systematically capturing and describing the behaviors of the VSR—similar procedures have been already used for analyzing the behavior of robots with evolved controllers, e.g., in Silva et al. (2017). We proceeded as follows. (1) For each morphology, we considered all and only the 60 best controllers (sensing and non-sens-

ing) obtained at the last generation. (2) We considered the discrete signals corresponding to the position $\mathbf{x}_{\text{CM}}(k)$ of the center of mass of the VSR during fitness evaluation. Figure 4 shows an example trajectory of one of the best sensing controllers for the worm morphology. (3) We computed the discrete Fourier transform (DFT) coefficients \mathbf{d}_x and \mathbf{d}_y , with $\mathbf{d}_x, \mathbf{d}_y \in \mathbb{R}^{n_{\text{sim}}}$, of the x - and z -components of $\mathbf{x}_{\text{CM}}(k)$; we did not consider the y -component since VSRs do not move significantly along that axis (see Figure 4). (4) We concatenated \mathbf{d}_x and \mathbf{d}_y , hence obtaining a vector $\mathbf{d} \in \mathbb{R}^{2n_{\text{sim}}}$ for each observed behavior. (5) Finally, we mapped all the behaviors from $\mathbb{R}^{2n_{\text{sim}}}$ to \mathbb{R}^2 using Multidimensional Scaling (MDS) (Cox and Cox, 2000). We explored different dimensionality reduction techniques, e.g., t-SNE (Maaten and Hinton, 2008): the qualitative observations presented below did not change.

Figure 5 shows the results of the analysis of the behaviors: for each morphology, the figure includes a plot where each behavior is a marker positioned according to the first two MDS coordinates. Three observations can be done based Figure 5. First, for the two simplest morphologies (worm and biped) the behaviors obtained with sensing and non-sensing controllers look clearly dissimilar: the red cloud is far from the blue cloud. Second, non-sensing controllers result in more homogeneous behaviors than sensing controllers: the red cloud is in general larger than the blue cloud. Third, the tripod case is, consistently with the previous findings, different from the other two cases: the difference of behaviors is fuzzier and similar behaviors can be found which are obtained with different controllers. We think that the motivation for this finding is twofold. On one hand, the larger complexity of the morphology may result in a larger set of interactions between the VSR and the environment, that is, in a larger expressiveness. On the other hand, as already observed above, the larger search space of this case may take longer to converge to a good controller; i.e., from another point of view, within the same number of generations, different evolutionary runs may follow different paths in the search space which do not end in the same “point”.

Environment: uneven surface

For the purpose of answering (RQ3), we considered a second case in which some aspect of the environment changes over the time. Differently than in the environment with even surface, variable environmental conditions constitute an opportunity for the sensing controller to exploit its peculiar ability of sensing the environment: that ability is instead not available for VSRs with the non-sensing controller.

For easing the experimentation, we introduced the variable conditions as a varying vector for the gravity acceleration. In particular, we varied the direction of the gravity vector during the simulation and kept constant its norm $\|\mathbf{g}\| = 9.8 \text{ m s}^{-2}$. The condition can be expressed as a function describing the value of the x -component $g_x(k)$ of

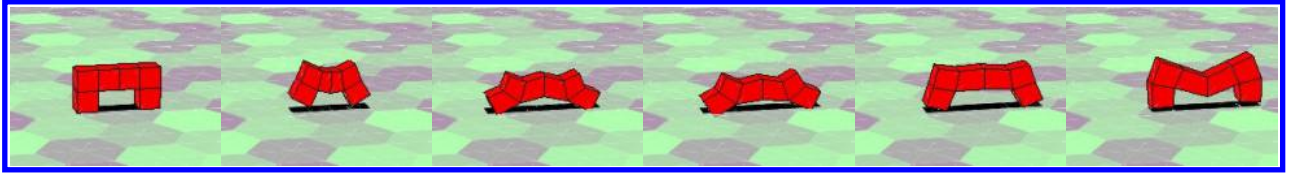


Figure 3: Frames capturing the behavior of a biped with one of the evolved sensing controller in the environment with even surface.

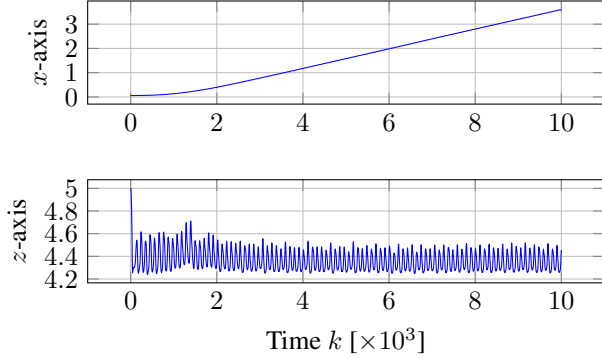


Figure 4: Trajectory $\mathbf{x}_{\text{CM}}(k)$ of the center of mass, shown separately for the two salient axes (scale is $\times 10^{-3}$ for the z -axis), of a worm with one of the evolved sensing controller in the environment with even surface.

the gravity vector \mathbf{g} over the time—assuming that the y -component is always equal to 0.

We proceeded as follows. First, we performed the evolutionary runs imposing a sinusoidal signal for the x -component of the gravity:

$$g_x^{\text{evo}} = \sin(2\pi f_{\mathbf{g}}^{\text{evo}} k \Delta t) \quad (4)$$

where $f_{\mathbf{g}}^{\text{evo}} = 2 \frac{1}{\Delta t n_{\text{sim}}} = 1.43$ Hz. Then, we assessed each evolved controller (i.e., the best individual at the last evolution) in three different *validation scenarios*:

$$g_x^{\text{flat}}(k) = 0 \quad (5)$$

$$g_x^{\text{step}}(k) = \begin{cases} 0 & \text{if } k \leq \frac{n_{\text{sim}}}{2} \\ 3 & \text{otherwise} \end{cases} \quad (6)$$

$$g_x^{\text{sin}}(k) = \sin(5\pi f_{\mathbf{g}}^{\text{evo}} k \Delta t) \quad (7)$$

By considering validation scenarios which are different from the one using during the evolution, we hence also assessed the generalization ability of the EA in evolving the VSR controllers. Note that varying the direction of the gravity vector basically corresponds to considering an uneven, instead of flat, surface on which the VSR moves.

Table 3 shows the results of the experiments in the uneven environment.

It can be seen that, also in this environment, the sensing controller is always more effective than the non-sensing one.

Table 3: Fitness of the best individual at the end of the evolution and its traveled distance in the validation scenarios (both in mm, mean μ and standard deviation σ across the 30 runs) in the uneven environment. ρ is the ratio between the traveled distance in the validation scenario and the fitness value.

	Morph.	Non-sensing			Sensing		
		μ	σ	ρ	μ	σ	ρ
Fitness	Worm	120	8		3050	358	
	Biped	78	19		873	180	
	Tripod	556	60		620	378	
Flat	Worm	138	5	1.15	3132	498	1.02
	Biped	73	30	0.93	2228	187	2.55
	Tripod	528	43	0.95	727	100	1.17
Step	Worm	111	56	0.92	3543	412	1.16
	Biped	69	20	0.88	1010	226	1.15
	Tripod	539	160	0.96	870	213	1.40
Sin	Worm	104	7	0.86	3194	281	1.04
	Biped	77	17	0.99	446	151	0.51
	Tripod	505	48	0.91	512	175	0.82

VSRs moved by the former travel a longer distance in any condition: both when computing the fitness (i.e., with g_x^{evo}) and in the validation scenarios (i.e., with g_x^{flat} , g_x^{step} , and g_x^{sin}). As for the even environment, differences are in general less apparent for the tripod than for the other two morphologies. All the differences are statistically significant according to the Mann-Whitney U-test ($\alpha = 0.05$): we do not show the values in the table.

Of more interest are the findings concerning the comparison between the fitness of the best individual and its performance in the validation scenario. Table 3 captures the outcome of this comparison in the two ρ columns: for a given morphology, controller, and validation scenario, ρ is the ratio of the distance traveled in the validation scenario and the fitness value, i.e., the one traveled with g_x^{evo} .

The key finding is that ρ is lower than 1 in most cases (8 on 9) for the non-sensing controller and greater than 1 in most cases for the sensing controller (7 on 9). VSRs equipped with the sensing controllers are hence able to move well on scenarios different than the one used for their evolu-

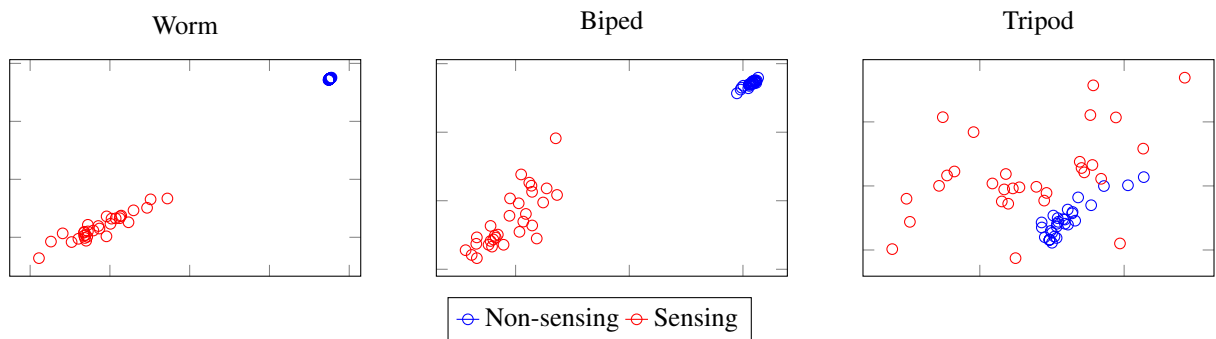


Figure 5: Behaviors resulting from the 60 best controllers evolved in the environment with even surface, with the three morphologies.

tion, whereas VSRs with non-sensing controller are not. We explain this clear difference with the fact that the sensing ability allows to react to an environment different from the one the controlled evolved and to adapt the VSR behavior.

Finally, Table 3 shows that, not surprisingly, the Sin validation scenario is the most difficult for all the VSRs: still, the worm equipped with a sensing controller is able to perform not worse on this scenario than on the one seen during the evolution ($\rho = 1.04$).

Analysis of the behaviors We performed the same analysis of the behaviors as for the environment with the even surface. The results are shown in Figure 6.

The findings are similar to the previous case. Sensing controllers exhibit, in general, more various behaviors and this difference is less apparent for the tripod than for the worm and the biped. However, Figure 6 also highlights that the behaviors resulting from sensing controllers differ among the three validation scenarios. The difference is more apparent for the biped. We motivate this latter finding with the fact that this morphology is a good trade-off in complexity: it is not too simple to prevent large variation in the behaviors (like the worm), nor too complex to make harder the evolution of controller able to exhibit a well-defined behavior (like the tripod).

Conclusions

Voxel-based soft robots are a promising framework in which the behavior of a robot is determined by both its brain, i.e., its controller, and its body. In this work we have explored a form of holistic design in which the controller is equipped with sensing capabilities distributed across the full body of the robot. We have considered a sensing controller represented as a neural network and have considered the problem of synthesizing such a controller automatically, by means of an Evolutionary Algorithm. We have exercised such an algorithm on three different bodies, each in two different environments, with the aim of solving a locomotion task. We have compared the resulting sensing controller to a more tra-

ditional one, also synthesized automatically with the same Evolutionary Algorithm, and we have found that the sensing controller is more effective than its non-sensing counterpart, also when immersed in an environment different from the one in which it evolved.

We believe these results are very promising and suggest that the shifting of complexity from the controller to the body intrinsic to voxel-based soft robots, should be carefully coupled with forms of distributed sensing. We intend to investigate the potential of sensing controllers on larger robots and more complex tasks. In order to cope with the resulting complexity of the search space, we plan to rely on a more efficient evolutionary framework, such as, e.g., CPPN-NEAT, as well as a modular design in which robots are assembled out of smaller (parts of) robots evolved separately.

Acknowledgments

The experimental evaluation of this work has been done on Amazon AWS within the “AWS Cloud Credits for Research” program. We thank Vittorio Casagrande and Erica Salvato for useful discussions about the design of the sensing controller.

References

- Bongard, J., Laschi, C., Lipson, H., Cheney, N., and Corucci, F. (2016). Material properties affect evolutions ability to exploit morphological computation in growing soft-bodied creatures. In *Artificial Life Conference Proceedings 13*, pages 234–241. MIT Press.
- Braganza, D., Dawson, D. M., Walker, I. D., and Nath, N. (2007). A neural network controller for continuum robots. *IEEE transactions on robotics*, 23(6):1270–1277.
- Bruder, D., Gillespie, B., Remy, C. D., and Vasudevan, R. (2019). Modeling and control of soft robots using the koopman operator and model predictive control. *arXiv preprint arXiv:1902.02827*.
- Cheney, N., Bongard, J., and Lipson, H. (2015). Evolving soft robots in tight spaces. In *Proceedings of the 2015 annual conference on Genetic and Evolutionary Computation*, pages 935–942. ACM.

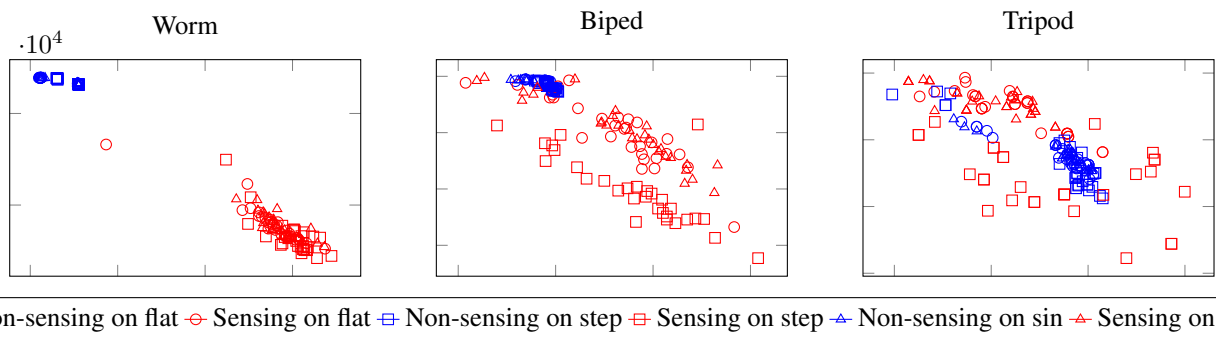


Figure 6: Behaviors resulting from the 60 best controllers evolved in the environment with uneven surface, with the three morphologies when testing them in the three validation scenarios.

- Cheney, N., Clune, J., and Lipson, H. (2014). Evolved electrophysiological soft robots. In *Artificial Life Conference Proceedings 14*, pages 222–229. MIT Press.
- Cheney, N., MacCurdy, R., Clune, J., and Lipson, H. (2013). 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*, pages 167–174. ACM.
- Corucci, F., Cheney, N., Giorgio-Serchi, F., Bongard, J., and Laschi, C. (2018). Evolving soft locomotion in aquatic and terrestrial environments: effects of material properties and environmental transitions. *Soft robotics*, 5(4):475–495.
- Cox, T. F. and Cox, M. A. (2000). *Multidimensional scaling*. Chapman and hall/CRC.
- Duan, Y., Chen, X., Houthoofd, R., Schulman, J., and Abbeel, P. (2016). Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*, pages 1329–1338.
- Hiller, J. and Lipson, H. (2012). Automatic design and manufacture of soft robots. *IEEE Transactions on Robotics*, 28(2):457–466.
- Iida, F. and Laschi, C. (2011). Soft robotics: challenges and perspectives. *Procedia Computer Science*, 7:99–102.
- Kim, S., Laschi, C., and Trimmer, B. (2013). Soft robotics: a bioinspired evolution in robotics. *Trends in biotechnology*, 31(5):287–294.
- Kriegman, S., Cappelle, C., Corucci, F., Bernatskiy, A., Cheney, N., and Bongard, J. C. (2017). Simulating the evolution of soft and rigid-body robots. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1117–1120. ACM.
- Kriegman, S., Cheney, N., and Bongard, J. (2018). How morphological development can guide evolution. *Scientific reports*, 8(1):13934.
- Lin, H.-T., Leisk, G. G., and Trimmer, B. (2011). Goqbot: a caterpillar-inspired soft-bodied rolling robot. *Bioinspiration & biomimetics*, 6(2):026007.
- Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.
- Pfeifer, R. and Bongard, J. (2006). *How the body shapes the way we think: a new view of intelligence*. MIT press.
- Rus, D. and Tolley, M. T. (2015). Design, fabrication and control of soft robots. *Nature*, 521(7553):467.
- Shepherd, R. F., Ilijevski, F., Choi, W., Morin, S. A., Stokes, A. A., Mazzeo, A. D., Chen, X., Wang, M., and Whitesides, G. M. (2011). Multigait soft robot. *Proceedings of the national academy of sciences*, 108(51):20400–20403.
- Silva, F., Correia, L., and Christensen, A. L. (2017). Hyper-learning algorithms for online evolution of robot controllers. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 12(3):14.
- Silva, F., Urbano, P., Correia, L., and Christensen, A. L. (2015). odneat: An algorithm for decentralised online evolution of robotic controllers. *Evolutionary Computation*, 23(3):421–449.
- Sims, K. (1994). Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 15–22. ACM.
- Stanley, K. O. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2):131–162.
- Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127.
- Vaughan, N. (2018). Evolution of neural networks for physically simulated evolved virtual quadruped creatures. In *Conference on Biomimetic and Biohybrid Systems*, pages 507–516. Springer.