Choice of robot morphology can prohibit modular control and disrupt evolution

Anton Bernatskiy[1] and Josh Bongard[1]

[1]Department of Computer Science, University of Vermont
abernats@uvm.edu

Abstract

In evolutionary robotics, controllers are often represented as networks. Modularity is a desirable trait of such networks because modular networks are resistant to catastrophic forgetting and tend to have less connections than nonmodular ones. However, these advantages can only be realized if the control task is solvable by a modular network, and for any given practical task the control task depends on the choice of the robot's morphology. Here we provide an example of a task solvable by robots with two different morphologies. We consider the most extreme kind of modularity – disconnectedness – and show that with the first morphology the task can be solved by a disconnected controller with few connections. On the other hand, the second morphology makes the task provably impossible for disconnected controllers and requires about three times more connections. For this morphology, most controllers that partially solve the task constitute local optima, forming an extremely deceptive fitness landscape. We show empirically that in this case a connection cost-based evolutionary algorithm for evolving modular controllers is greatly slowed down compared to the first morphology's case. Finally, this performance gap increases as the task is scaled up. These results show that the morphology may be a major factor determining the performance of controller optimization. Although in our task the optimal morphology is obvious to a human designer, we hypothesize that as evolutionary robotics is scaled to more sophisticated tasks the optimization of morphology alongside the control might become a requirement for evolving modular controllers.

Frequently used symbols:
T_i – target orientation of ith segment;
A_i – absolute orientation of ith segment;
r_i – relative orientation of ith segment (defined in eq. (2));
s_i – reading of the target orientation sensor measuring the orientation of the ith segment (5);
f_i – motor output for ith segment (6);
$\boldsymbol{T}, \boldsymbol{A}, \boldsymbol{r}, \mathbf{s}, \mathbf{f}$ – corresponding N-dimensional vectors, where
N – is a number of segments; J – sensor attachment matrix (5); K – constant $N \times N$ matrix such filled with ones on and below the main diagonal and zeros everywhere else (4).

Introduction

Evolutionary computation and particularly evolutionary robotics are important research tools in the area of artificial life (Langton (1997); Lipson and Pollack (2000); Rohde (2010)). A lot of research effort concerning evolutionary computation is dedicated to the evolution of networks. Network representation has several advantages. First, it can describe many kinds of systems, including controllers and morphologies of artificial agents (e.g. Sims (1994)). Second, it is relatively straightforward to design genetic operators such as mutation and crossover for networks. Last but not least, a lot of models in biology are network-based, making it easier to draw inspiration from natural evolution.

One characteristic property of biological networks that attracts a lot of attention from evolutionary computation community is modularity (e.g. Girvan and Newman (2002)). A network is structurally modular if its nodes can be divided into subsets (modules) that are connected more tightly within themselves than with the rest of the network. In computational (e.g. neural, genetic regulatory) networks structural modularity often leads to weak or absent functional dependence. Consequences of such weak dependence include resistance to catastrophic forgetting both in neuroevolutionary setting (Kashtan and Alon (2005); Espinosa-Soto and Wagner (2010); Clune et al. (2013)) and in learning (Ellefsen et al. (2015)). Such resistance allows for a reduced number of training examples (Cappelle et al. (2016)). In addition, modular networks tend to contain less connections, which further simplifies their optimization (Clune et al. (2013); Bernatskiy and Bongard (2015)).

Although some techniques for evolving modular computational networks have been developed (e.g. Kashtan and Alon (2005); Espinosa-Soto and Wagner (2010); Clune et al. (2013); Bernatskiy and Bongard (2015)), they mostly focus on finding nearly-optimal modular solutions that are assumed to be exist. In addition, to harness the full power of the approach an appropriate modular variation of the task is desirable, either among the training examples (Cappelle et al. (2016)) or over the evolutionary time (Kashtan and Alon (2005); Espinosa-Soto and Wagner (2010); Clune et al.

(2013)). However, in practice, it cannot be assumed that modular solutions for a given task exist, nor that the task itself is modularly varying. Indeed, in the work presented here, we demonstrate that there is a robotic task and a robot morphology for which even the former assumption does not hold.

In evolutionary robotics, controllers are often represented as computational networks. Properties of modular networks make modularity a desirable property of such controllers. However, previous work (Bongard et al. (2015); Cappelle et al. (2016)) suggests that performance of modular network evolution techniques in this setting can vary depending on the choice of the robot's body.

In Bongard et al. (2015) evolution produced more modular controllers if the morphology was under the evolutionary control. It was observed that certain morphologies enable modular control while others do not, but it was not clear which mechanism might be responsible for that.

In Cappelle et al. (2016), a morphology is defined to be modular iff activation of less than all of robot's motors results in a change of less than all of its sensors. Similarly, a control system is modular iff a change in less than all of the sensors induces a change in less than all of the motor neurons. It is shown that the number of environments in which the robot needs to be evaluated can be reduced significantly if both morphology and control are fixed to be modular.

Despite these findings, many things remain unclear regarding the relationship between the morphology and modularity. In particular, while it is known that certain morphologies are beneficial for the evolution of modular controllers, it is not known why. Another open question is how much worse can the performance of evolution be if a less appropriate morphology is chosen. Here we fill these gaps by introducing a task and a family of robot morphologies with two extremities. The first is a morphology for which some optimal controllers consist of multiple disconnected modules. For the second morphology of interest it is provably impossible for any optimal controller to be disconnected. Additionally, we show the that the latter morphology induces an extremely deceptive fitness landscape in the space of possible controllers.

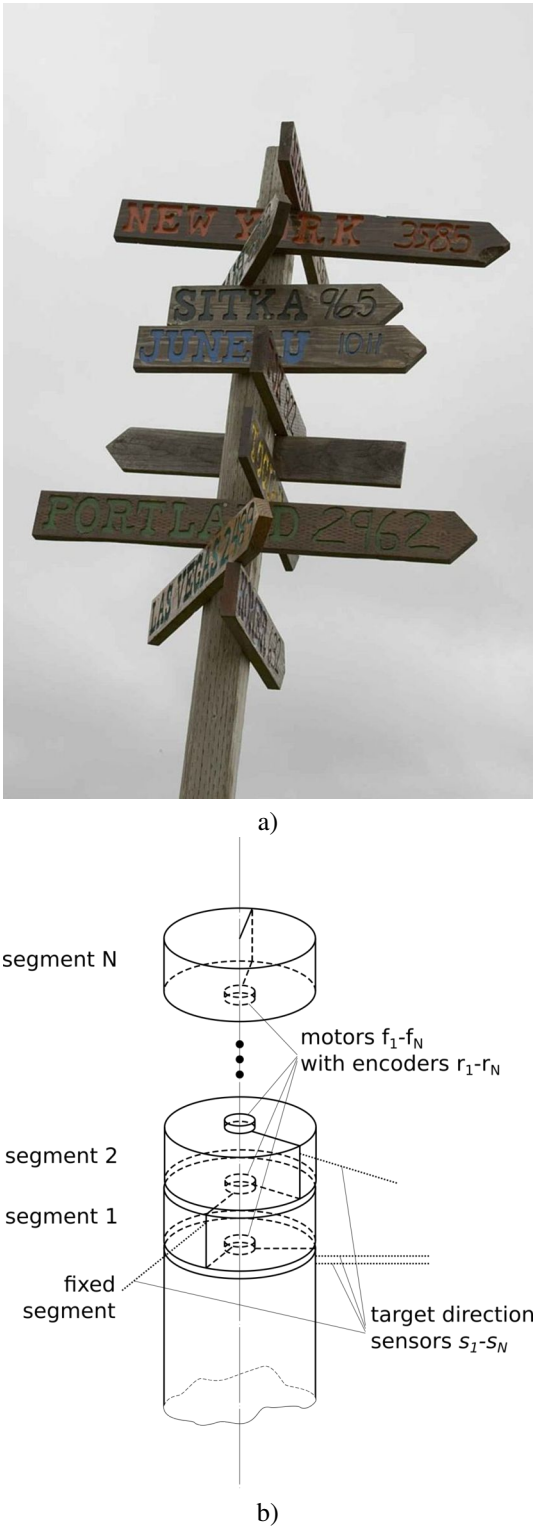

Figure 1: a) Regular road sign with multiple destinations. b) Arrowbot. Lines on the segments show orientations of segments' arrows.

System description

Robot and task

Robots of the family described in this paper are called Arrowbots. An Arrowbot (Fig. 1b) is a dynamic version of road sign with multiple destinations (Fig. 1a). The task of a road sign is to show the direction towards several fixed objects, such as cities or mountains. In contrast, the task of an Arrowbot is to track N objects that may move and point towards them.

To accomplish this task the robot's body is divided into N segments attached to each other in series by coaxial, actuated rotary joints, forming a stack (Fig. 1b). Segment #1

is located at the bottom of the stack. It is attached to a fixed base.

We define the orientation of the fixed base to be zero. For each segment i its absolute orientation angles A_i and target orientation T_i are defined relative to this reference (see Figure 2). If $A_i = T_i$, the segment points exactly in the target direction. Denoting $\boldsymbol{T} \equiv [T_1, T_2, ..., T_N]^T$, $\boldsymbol{A} \equiv [A_1, A_2, ..., A_N]^T$ we can reformulate the task as the minimization of

$$E \equiv |\boldsymbol{T} - \boldsymbol{A}|, \tag{1}$$

at $t \to \infty$ for some $\boldsymbol{T}(t)$ and initial condition $\boldsymbol{A}(t = 0)$. Throughout this work we assume constant target orientations, $\boldsymbol{T}(t) = const$.

Each segment i is associated with two sensors: a proprioceptive sensor and a target orientation sensor.

Each **proprioceptive sensor** measures the relative angle r_i between the orientation of its segment (ith) and the orientation of the $i-1$st segment below it (for segment #1, the fixed base). Readings of these sensors are tied to absolute orientations of segments:

$$\boldsymbol{A} = K\boldsymbol{r}, \tag{2}$$

or equivalently,

$$\boldsymbol{r} = K^{-1}\boldsymbol{A}. \tag{3}$$

Here $\boldsymbol{r} \equiv [r_1, r_2, ..., r_N]^T$ and K is defined to be the constant $N \times N$ matrix filled with ones on and below its main diagonal and zeros above it:

$$K_{ij} = 1 \text{ if } i \geqslant j \text{ else } 0. \tag{4}$$

The determinant of this matrix is 1 for any N, so the inverse always exists, hence the equivalence of (2) and (3).

Each **target orientation sensor** s_i outputs the angle between the orientation of whatever it is attached to and the target orientation of its segment. Each target orientation sensor can be attached to any segment or to the fixed base.

Different ways of attaching target direction sensors give rise to different Arrowbot morphologies. We describe them with **sensor attachment matrix** J: an $N \times N$ matrix for which any element J_{ij} is equal to 1 if sensor s_i is attached to the jth segment and 0 otherwise. There is always exactly one sensor for every target direction, so every row of J contains at most one unit entry. If ith sensor is attached to the fixed base, then it is not attached to any of the moving segments and all the elements of the ith row of J are zeros.

With J we can express the absolute orientations of the parts to which the target orientation sensors are attached as $J\boldsymbol{A}$. Then target orientations sensor readings $\mathbf{s} \equiv [s_1, s_2, ..., s_N]$ are

$$\mathbf{s} = \boldsymbol{T} - J\boldsymbol{A} = \boldsymbol{T} - JK\boldsymbol{r}. \tag{5}$$

Actuated joints between the segments are the only motors of the system. Arrowbot's inputs are joint rotational velocities of segments relative to the segments right below them, $\dot{r_i}$.

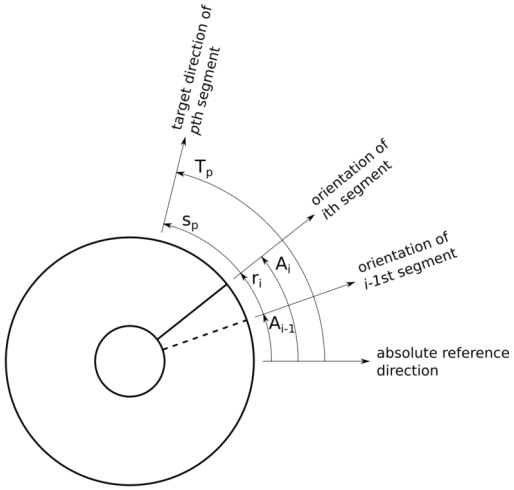

Figure 2: Arrowbot segment, top view. Solid radial line shows the orientation of the current (ith) segment. Dashed radial line shows the orientation of the segment right below the current one ($i-1$st). In this example, only one target direction sensor is attached to the segment. The sensor perceives the target direction of pth segment ($J_{pi} = 1$).

Control

A controlled Arrowbot is described by the following dynamical system:

$$\dot{\boldsymbol{r}} = \mathbf{f}(\boldsymbol{r}, \mathbf{s}(\boldsymbol{r}, \boldsymbol{T})), \tag{6}$$

where $\mathbf{f}(\boldsymbol{r}, \mathbf{s}) : \mathbb{R}^{2N} \to \mathbb{R}^N$ is the controller and $\mathbf{s}(\boldsymbol{r}, \boldsymbol{T})$ is given by (5). For the controller to solve the task (1), this dynamical system must have an isolated, asymptotically stable fixed point corresponding to $\boldsymbol{T} = \boldsymbol{A} = K\boldsymbol{r}$. This translates into several conditions, each of which must be met for every $\boldsymbol{T} \in \mathbb{R}^N$. Equilibrium at $\boldsymbol{T} = K\boldsymbol{r}$ translates to:

$$\mathbf{f}(\boldsymbol{r} = K^{-1}\boldsymbol{T}, \mathbf{s}(\boldsymbol{r} = K^{-1}\boldsymbol{T}, \boldsymbol{T})) = \mathbf{0}. \tag{7}$$

The equilibrium point must be of the attracting, or asymptotically stable, kind:

$$\exists \delta_0 > 0 \,\text{s.t.}\, |K^{-1}\boldsymbol{T} - \boldsymbol{r}(0)| < \delta_0 \Rightarrow \lim_{t\to\infty} \boldsymbol{r}(t) \to K^{-1}\boldsymbol{T}, \tag{8}$$

where $\boldsymbol{r}(t)$ denotes the trajectory of the dynamical system (6) given target orientations $\boldsymbol{T}$ and initial conditions $\boldsymbol{r}(0)$.

If some controller, in addition to satisfying these two necessary conditions, also ensures that the fixed point is unique, then the point $\boldsymbol{A} = \boldsymbol{T}$ will attract all trajectories regardless of the initial conditions $\boldsymbol{r}(0)$. Such controllers are globally optimal for the task (1).

We characterize the connectivity of the controller using the following formalized notion of dependence. In a system with n variables $V = \{x_1, x_2, ..., x_n\}$ subject to some constraints C, x_i is **dependent** on x_j iff for some setting of

the $n-2$ remaining variables $\hat{x}$ and some pair of settings $x'_j \neq x''_j$ the sets $X'_i \equiv \{x_i \text{ such that } \{x_i, x'_j, \hat{x}\} \text{ satisfies } C\}$ and $X''_i \equiv \{x_i \text{ such that } \{x_i, x''_j, \hat{x}\} \text{ satisfies } C\}$ do not coincide. Dependencies induced by the constraint C on the variables define an **undirected dependence graph** $G = (V, E)$ where $(x_i, x_j) \in E$ iff x_i depends on x_j or x_j depends on x_i.

If some variable x_i depends on some other variable x_j and the dependence is not satisfied by definition of these variables (as is the case, for example, for radius and diameter or $\boldsymbol{r}$ and $\boldsymbol{A}$), then in any implementation of the constraint C x_i is connected to x_j via some kind of channel transmitting the information about x_j to the process generating x_i. For example, in a system involving two coordinates x, y of some material body on a plane the constraint $x = -y$ can only be enforced by introducing some physical contraption (e.g. a diagonal rail) which ensures that whenever y changes, x changes accordingly. Due to this property, the connectivity of the undirected dependence graph is the same as the connectivity of any undirected graph representing the information channels in the implementation, except possibly for situations when the implementation involves hidden variables that are neither influenced by nor influence the variables in V.

To investigate the connectivity of optimal Arrowbot controllers we consider undirected dependence graphs on $V^* = \{\mathbf{f}, \boldsymbol{r}, \mathbf{s}\}$ subject to the constraint $\mathbf{f} = \mathbf{f}(\boldsymbol{r}, \mathbf{s}(\boldsymbol{r}, \boldsymbol{T}))$. As we will see, necessary conditions (7) and (8) restrict the connectivity of the graphs in a way that depends on the sensor attachment matrix J, i.e. on the robot's morphology. Since the definitions of all variables in V^* do not imply any automatically satisfied constraints, we can draw conclusions about the connectivity of arbitrary nonlinear controllers from the connectivity of undirected dependence graphs.

This approach is inspired by similar tools used to treat dynamical dependencies as well as constraints: diagrams of immediate effect (Ashby (1960)) and functional dependence graphs (Etesami and Kiyavash (2016)).

J=I admits disconnected optimal controllers

Attachment matrix $J = I$ corresponds to the morphology in which every sensor is attached to the segment for which it tracks the target orientation. In this case each sensor s_i measures its segment's signed pointing error, $T_i - A_i$.

Consider a family of controllers

$$\mathbf{f}(\boldsymbol{r}, \mathbf{s}) = W\mathbf{s}, \tag{9}$$

where $W = \text{diag}[w_{11}, w_{22}, ..., w_{NN}]$ are diagonal matrices for which all w_{ii} are positive. Then the dynamical system (6) turns into

$$\dot{\boldsymbol{r}} = W\boldsymbol{T} - WK\boldsymbol{r}. \tag{10}$$

Right hand side of this equation turns into 0 iff $\boldsymbol{T} = \boldsymbol{A} = Kr$, making sure that (7) and (15) are satisfied. Also, it

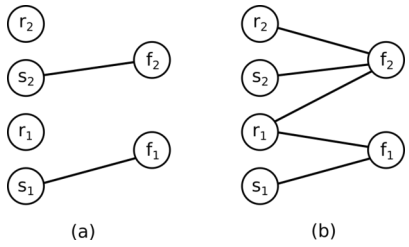

Figure 3: Examples of undirected dependency graphs (UDGs). (a) UDG of one of the controllers of the family (9) optimal for $J = I$; (b) UDG of one possible optimal controller for $J = 0$.

shows that the fixed point $\boldsymbol{T} = \boldsymbol{A}$ is unique. The Jacobian

$$-WK = \begin{bmatrix} -w_{11} & 0 & ... & 0 \\ 0 & -w_{22} & ... & 0 \\ & & \dot{} & \\ 0 & 0 & ... & -w_{NN} \end{bmatrix} \begin{bmatrix} 1 & 0 & ... & 0 \\ 1 & 1 & ... & 0 \\ & & \dot{} & \\ 1 & 1 & ... & 1 \end{bmatrix} = \\ = \begin{bmatrix} -w_{11} & 0 & ... & 0 \\ -w_{22} & -w_{22} & ... & 0 \\ & & \dot{} & \\ -w_{NN} & -w_{NN} & ... & -w_{NN} \end{bmatrix}. \tag{11}$$

is a triangular matrix, therefore its eigenvalues are the values at the diagonal, $-w_{11}, -w_{22}, ..., -w_{NN}$. All of them are negative, so the stability condition (8) is also satisfied.

Therefore, all controllers of the family (9) are globally optimal for the task (1).

Every controller of this family is a disconnected network of $2N$ independent modules. Half of the modules connect motors $\dot{r_i}$ to target orientation sensors s_i associated with and attached to their segments. Another half are the proprioceptive sensors, which are, in these controllers, not connected to any other nodes.

There is no disconnected optimal controller for *J=0*

$J = 0$ corresponds to the case when all the target orientation sensors are attached to the fixed base of the robot and are directly measuring the target orientations, $\mathbf{s} = \boldsymbol{T}$. The dependence of $\mathbf{s}$ on $\boldsymbol{r}$ disappears, so in this case $\mathbf{s}$ can be treated as a constant vector of parameters. The dynamical system (6) then has a fixed point whenever $\boldsymbol{r} = K^{-1}\mathbf{s}$:

$$\begin{aligned} &\forall \mathbf{s} \in \mathbb{R}^N \\ &[r_1, r_2, .., r_N] = [s_1, s_2 - s_1, .., s_N - s_{N-1}] \Rightarrow \\ &\mathbf{f}(\boldsymbol{r}, \mathbf{s}) = \mathbf{0} \end{aligned} \tag{12}$$

This also implies that for every point $\boldsymbol{r} \in \mathbb{R}^N$ there is a set of parameters $\mathbf{s} = K\boldsymbol{r}$ such that $\mathbf{f}(\boldsymbol{r}, \mathbf{s}) = \mathbf{0}$.

Theorem 1. Any controller $\mathbf{f}(\boldsymbol{r}, \mathbf{s})$ inducing an asymptotically stable fixed point in (6) for any $\mathbf{s} \in \mathbb{R}^N$ and $\boldsymbol{r} = K^{-1}\mathbf{s}$ has a connected undirected dependency graph.

See Appendix A for the proof.

Theorem 1 implies that for $J = 0$ any optimal controller has only one connected component that is not isolated from

sensors and motors. All sensors and motors, a total of $3N$ nodes, participate in this component. Such a component cannot be connected unless there are at least $3N-1$, which is about three times more than what controllers for $J=0$ require.

Evolution

We have shown that there are disconnected controllers among the globally optimal ones if the robot's morphology is defined by the sensor attachment matrix $J=I$ and that there are none if $J=0$. However, it is not clear if this fact influences the complexity of optimizing the controller for the two morphologies, particularly if the optimization algorithm is designed to find modular solutions efficiently. We investigate this using a biobjective error-connection cost optimization Clune et al. (2013).

Each controller is represented by a vector of $2N^2$ connection weights encoding two $N \times N$ matrices (Y, W) such that

$$\dot{\boldsymbol{r}} = Y\boldsymbol{r} + W\mathbf{s}. \tag{13}$$

Connection weights can take values in $\{-1, 0, 1\}$.

Controllers are evaluated in a number of environments characterized by initial conditions $\boldsymbol{r}(t=0)$ and target orientations $\boldsymbol{T}$. In each environment, each controller is evaluated by substituting (5) into (13) and integrating the resulting linear ODE system with fourth order Runge-Kutta method over a fixed time span $[0, 10]$ with a fixed timestep of 0.1. Pointing errors (1) are computed for each environment at $t=10$ and averaged to obtain the final evaluation e.

Following the connection cost method for evolving modular networks Clune et al. (2013), we simultaneously minimize pointing error and connection cost, defined as number of connections with nonzero weight. We use evolutionary algorithm identical to the one described in Bernatskiy and Bongard (2015). At each generation increment we select the pointing error – connection cost Pareto front of the current population and copy it into the new population. Then we proceed to add mutated copies of networks randomly chosen from Pareto front to the new population until it has the same size as the old one.

Mutation operator either (1) replaces a value of one nonzero weight with another (with probability of $p_1 = 0.5$ in all our experiments), or (2) adds a nonzero weight ($p_2 = 0.25$), or (3) removes a nonzero weight ($p_3 = 0.25$). If an impossible operation is attempted (e.g. nonzero weight has to be removed from an network with no such weights), the mutation is attempted repeatedly until it happens to perform a possible operation.

Since the Pareto front is copied into the new population without modification regardless of its size, it can potentially fill the population completely and cause variation to cease. To prevent this, a sufficiently large population size must be chosen. Preliminary runs pointed to a population size of 50, and that was sufficient to avoid such variation cessation in any of the runs mentioned in this paper (see Table 1 for details).

We explored two types of initial populations – a population of networks generated by randomly choosing weights from $\{-1, 0, 1\}$ (**random** setup) and a population of networks generated by mutating an empty network once (**sparse** setup, Bernatskiy and Bongard (2015)). The latter setting trades off some initial variation to make the convergence faster if the task can be solved by a sparse and/or modular network. As we will see shortly, this leads to severe performance penalties if the task cannot be solved by such networks (in our case for $J=0$).

Results and discussion

We investigated performance of the evolution for two types of initial populations (random and sparse) and three values of the number of segments parameter $N = 3, 5, 10$. Since each genome encoded a linear controller in form of two NxN matrices, the length of the genome has grown quadratically with N and the corresponding genome sizes were $18, 50, 200$. We ran batches of 50 evolutionary runs, 500 generations each, with populations of 50 individuals.

A set of 3 environments was used to evaluate controllers, with target orientations

$$\boldsymbol{T}^1 = [1, 0, 0], \boldsymbol{T}^2 = [0, 1, 0], \boldsymbol{T}^3 = [0, 0, 1]. \tag{14}$$

where the upper index indicates the number of the environment. Initial conditions were $\boldsymbol{r} = [0, 0, 0]$ in all three environments.

The results are shown in Figure 4 and 1. It can be seen that for $J=0$ the performance of evolution is severely impaired, especially for the sparse initial population setting. In this case the error level is not improved relative to the initial conditions in any of 50 runs, while for $J=I$ a controller with near-zero error is found within 30 generations in all runs.

This result can be explained by considering the difference in response to mutation in non-optimal controllers between the two morphologies.

For $J=I$ the modules in the global optimum are disconnected and any controller that solves the task partially can retain the partial solution after mutation. This is a phenomenon called serial adaptation (Ashby (1960)). In this case fitness landscape is convex.

The other morphology, $J=0$, induces a more complicated landscape. Suppose, for this morphology, that at some point of the evolution there is a controller that successfully reduces the pointing error of the top $N-n$ segments of the Arrowbot, but not for the bottom n segments. Absolute orientation of nth segment A_n is required to compute $\dot{r}_i$ for $i = n+1..N$. Since the task for the lower segments is not solved yet, $A_n = \sum_{i=1}^{n} r_i$ will have random dynamics,

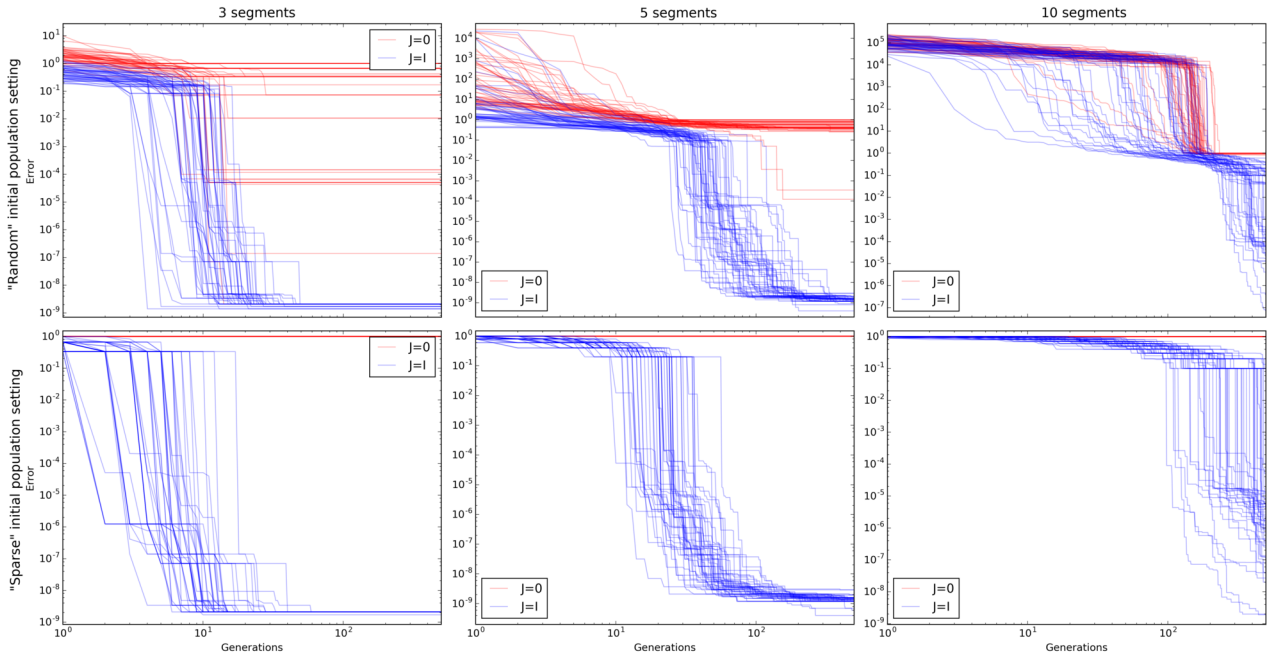

Figure 4: Time series of the smallest error for evolution of Arrowbots with two different morphologies $J = 0$ and $J = I$. Columns correspond to the three settings of Arrowbots' size (3,5 and 10 segments). For the runs in the top row, the evolution was initialized with a population of random networks; bottom row shows the performance if the initial population consists of sparse networks. Each of 50 trajectories is plotted in a semi-transparent line. Initial conditions in all cases are such that in every environment the error is initially equal to 1. It can be seen that for $J = 0$ evolution shows poor performance with random initial population and is completely disrupted for sparse initial population, even though for $J = I$ the optimum is found more rapidly in this setting.

which must be exploited by the partial solution. Any mutation which improves the pointing error of any of the lower n segments will change this dynamics and likely break the partial solution. Thus, the fitness landscape is deceptive.

An initial population of sparse networks and the pressure to minimize the number of connections reinforce such deceptiveness. Under those conditions the most likely random behavior of the lower segments is staying at rest. This makes exploiting it especially simple and gives the local optima

Initial population type	Number of segments		
	N=10	N=5	N=3
random	37/38.4	21/37.6	13/41.2
sparse	28/40.3	21/40.4	13/43.0

Table 1: Variation data for the evolutionary runs shown in Figure 4. Each cell shows maximum size of the error-connection cost Pareto front A across all generations and all runs and the average number of individuals mutated on each generation, B, in format A/B. Since the maximum Pareto front size never reaches the size of the population (50), the variation never ceases in any of the runs.

large basins of attraction with pronounced gradients, ultimately causing a complete disruption of the evolution.

Figure 5 shows how the performance of the evolution, measured by the final total square pointing error after 500 generations, varies with the number of segments N. Optimization task becomes more challenging as N grows. For the random initial population setting and $J = 0$, the evolution does not improve over the error of initial conditions at $N = 10$ in most runs, while for $J = I$ it does. For the sparse initial population setting, with $J = 0$ evolution does not improve in any of 3x50 runs, while for $J = I$ it achieves near zero pointing error in all 3x50 cases except for three evolutionary runs performed with $N = 10$.

Conclusions

We have introduced Arrowbots, a task and a family of scalable robot morphologies which exhibits strong dependence of control modularity on morphology. In particular, we have shown that within this family there is (1) a morphology $J = I$ for which there are optimal controllers consisting of multiple disconnected modules and (2) a morphology $J = 0$ for which any optimal controller is necessarily connected. We have demonstrated that the performance of

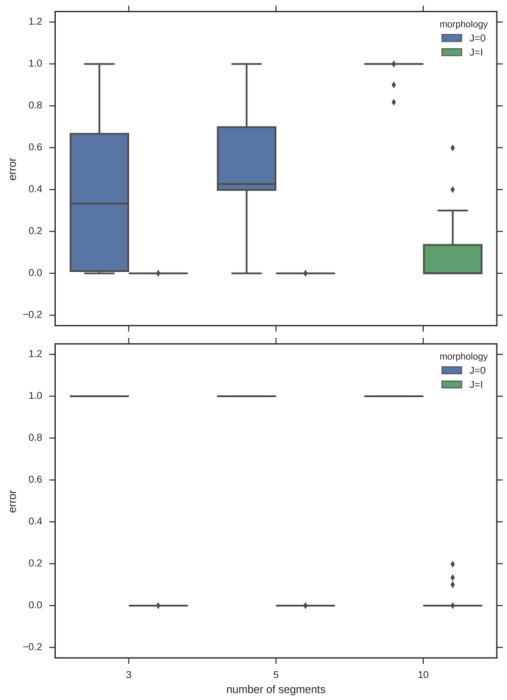

Figure 5: Final smallest errors for evolution of Arrowbots with different number of segments N after 500 generations. Top plot shows the performance if the evolution is initialized with a population of random networks; bottom plot shown the case of sparse initial population setting. It can be seen that the task becomes increasingly more challenging as N grows, especially for $J = 0$ and random initial population setting.

the evolution of Arrowbot control for the $J = 0$ morphology can be markedly worse than for the $J = 1$ and that the performance gap grows when Arrowbots with more segments are considered. We hypothesize that the difference in the performance of evolution is due to the extreme deceptiveness of the fitness landscape which arises if the robot has the $J = 0$ morphology and does not arise if $J = I$.

Thus, we have shown by construction that the choice of morphology can be the decisive factor in the evolution of modular controllers. The more aggressively the algorithm exploits the heuristic of modularity, the more important the morphology seems to become. Although for Arrowbots the morphology corresponding to most modular controllers is obvious to human designers, we hypothesize that in more sophisticated tasks this may not be the case. Optimization of the morphology alongside the control might be the solution of choice for those more sophisticated tasks.

Acknowledgments

We thank Elizaveta Guseva, Ken Livingston, John Long, Nick Livingston, Jodi Schwarz, Marc Smith and Collin Cappelle for fruitful discussions. This work was supported by NSF awards INSPIRE-1344227 and PECASE-0953837. The computational resources provided by the UVM's Vermont Advanced Computing Core (VACC) are gratefully acknowledged.

References

Ashby, W. R. (1960). *Design for a Brain: The origin of adaptive behavior*. Chapman & Hall Ltd, London, 2nd edition.

Bernatskiy, A. and Bongard, J. C. (2015). Exploiting the relationship between structural modularity and sparsity for faster network evolution. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 1173–1176. ACM.

Bongard, J. C., Bernatskiy, A., Livingston, K., Livingston, N., Long, J., and Smith, M. (2015). Evolving robot morphology facilitates the evolution of neural modularity and evolvability. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 129–136. ACM.

Cappelle, C. K., Bernatskiy, A., Livingston, K., Livingston, N., and Bongard, J. (2016). Morphological modularity can enable the evolution of robot behavior to scale linearly with the number of environmental features. *Frontiers in Robotics and AI*, 3:59.

Clune, J., Mouret, J.-B., and Lipson, H. (2013). The evolutionary origins of modularity. In *Proc. R. Soc. B*, volume 280, page 20122863. The Royal Society.

Ellefsen, K. O., Mouret, J.-B., and Clune, J. (2015). Neural modularity helps organisms evolve to learn new skills without forgetting old skills. *PLoS Comput Biol*, 11(4):e1004128.

Espinosa-Soto, C. and Wagner, A. (2010). Specialization can drive the evolution of modularity. *PLoS Comput Biol*, 6(3):e1000719.

Etesami, J. and Kiyavash, N. (2016). Measuring causal relationships in dynamical systems through recovery of functional dependencies. *IEEE Transactions on Signal and Information Processing over Networks*.

Girvan, M. and Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826.

Kashtan, N. and Alon, U. (2005). Spontaneous evolution of modularity and network motifs. *Proceedings of the National Academy of Sciences of the United States of America*, 102(39):13773–13778.

Langton, C. G. (1997). *Artificial life: An overview*. Mit Press.

Lipson, H. and Pollack, J. B. (2000). Automatic design and manufacture of robotic lifeforms. *Nature*, 406(6799):974–978.

Rohde, M. (2010). *Enaction, embodiment, evolutionary robotics: simulation models for a post-cognitivist science of mind*, volume 1. Springer Science & Business Media.

Sims, K. (1994). Evolving 3d morphology and behavior by competition. *Artificial life*, 1(4):353–372.

Appendix A. Proof of Theorem 1.

Isolation: asymptotic stability (8) implies that the fixed point $\boldsymbol{r} = K^{-1}\boldsymbol{T}$ is isolated:

$$\exists \epsilon > 0 : 0 < |\boldsymbol{r} - K^{-1}\boldsymbol{T}| \leqslant \epsilon \Rightarrow \mathbf{f}(\boldsymbol{r}, \mathbf{s}(\boldsymbol{r}, \boldsymbol{T})) \neq \mathbf{0}. \quad (15)$$

The statement seems to be well known in the dynamical systems community, so I'll only provide a sketch of the proof. Suppose the opposite, then a fixed point $\boldsymbol{r}'$ can be found arbitrarily close to $K^{-1}\boldsymbol{T}$. Trajectories starting at $\boldsymbol{r}'$ do not approach $K^{-1}\boldsymbol{T}$, which contradict the asymptotic stability of that point.■

We begin by using this property, together with the asymptotic isolation itself, to constrain necessary dependencies of the involved variables. Then we show that under these constraints the undirected dependency network must be connected.

Lemma 1. In any controller $\mathbf{f}(\boldsymbol{r}, \mathbf{s})$ inducing a stable fixed point in (6) for any $\mathbf{s} \in \mathbb{R}^N$ and $\boldsymbol{r} = K^{-1}\mathbf{s}$ **(part 1)** any motor output $f_i(\boldsymbol{r}, \mathbf{s})$ depends on the readings of at least one proprioceptive sensor r_j and **(part 2)** for any proprioceptive sensor r_i, there is a motor output $f_j(\boldsymbol{r}, \mathbf{s})$ that depends on it.

Proof. Part 1: Suppose some motor output f_i is independent of all proprioceptive sensors $\boldsymbol{r}$. Then f_i is the same for all $\boldsymbol{r} \in \mathbb{R}^N$. Since we presupposed the existence of at least one fixed point, $f_i = 0$ at the point and therefore everywhere. It follows that $r_i = \text{const}$, which contradicts the asymptotic convergence.

Part 2: Suppose no motor output depends on some proprioceptive sensor r_i. For any $\mathbf{s}$ $\mathbf{f}(\boldsymbol{r}' \equiv K^{-1}\mathbf{s}, \mathbf{s}) = 0$. Now, consider a vector

$$\boldsymbol{r}'' \equiv K^{-1}\mathbf{s} + (0, .., \epsilon/2, .., 0) \quad (16)$$

where is the second term ϵ is at ith position in the vector. Since $\mathbf{f}$ is independent from r_i, $\mathbf{f}(\boldsymbol{r}'', \mathbf{s}) = 0$. Because ϵ can be chosen arbitrarily, that means that $\forall \epsilon > 0$ there is a fixed point $\boldsymbol{r}'' \neq \boldsymbol{r}'$ in the ϵ-neighborhood of $\boldsymbol{r}'$. This contradicts isolation. ■

Lemma 2. In any controller $\mathbf{f}(\boldsymbol{r}, \mathbf{s})$ inducing a stable fixed point in (6) at $\boldsymbol{T} = \boldsymbol{A}$ any target orientation sensor s has at least one motor output f that depends on it.

Proof. Suppose it is not, then there is a sensor s_i such that for any two values s_i and $s_i^* \in \mathbb{R}$ and any setting of remaining values $\hat{\mathbf{s}} \in \mathbb{R}^{N-1}$, $\boldsymbol{r} \in \mathbb{R}^N$ $\mathbf{f}(s_i, \hat{\mathbf{s}}, \boldsymbol{r}) = \mathbf{f}(s_i^*, \hat{\mathbf{s}}, \boldsymbol{r})$. Picking an arbitrary $\mathbf{s} \in \mathbb{R}^N$, we can choose its ith component as s_i, remaining values as $\hat{\mathbf{s}}$ and $K^{-1}\mathbf{s}$ as $\boldsymbol{r}$. By the conditions of Lemma 2, in this case the dynamical system (6) has a fixed point, so $\mathbf{f}(s_i, \hat{\mathbf{s}}, \boldsymbol{r}) = \mathbf{0}$.

We can also choose an arbitrary $\epsilon \in \mathbb{R}$ and set $s_i^* = s_i + \epsilon$, then

$$\mathbf{f}(s_i + \epsilon, \hat{\mathbf{s}}, \boldsymbol{r}) = \mathbf{f}(s_i, \hat{\mathbf{s}}, \boldsymbol{r}) = \mathbf{0}. \quad (17)$$

Denoting the vector $\mathbf{s}^*$ to have the same values as $\mathbf{s}$ except for $s_i^* = s_i + \epsilon$, we can conclude that the dynamical system $\dot{\boldsymbol{r}} = \mathbf{f}(\boldsymbol{r}, \mathbf{s}^*)$ has a fixed point at

$$\boldsymbol{r} = K^{-1}\mathbf{s} = [s_1, s_2 - s_1, ..., s_N - s_{N-1}]. \quad (18)$$

However, by Lemma conditions it also has an isolated fixed point at

$$\begin{aligned} \boldsymbol{r}^* \equiv K^{-1}\mathbf{s}^* = [s_1, s_2 - s_1, ..., \\ s_i + \epsilon - s_{i-1}, s_{i+1} - s_i - \epsilon, ..., s_N - s_{N-1}]. \end{aligned} \quad (19)$$

$\boldsymbol{r}$ and $\boldsymbol{r}^*$ are no further than 2ϵ from each other by any distance measure. By choosing appropriate $\epsilon = \epsilon'/3$, we can find a fixed point in any ϵ'-neighborhood of $\boldsymbol{r}^*$, which must be an isolated fixed point: contradiction. ■.

It follows from Lemma 1 that there is one-to-one correspondence between motors and proprioceptory sensors: they can be partitioned in N pairs (f_i, r_j) such that every f_i and every r_j participates in some pair and in each pair f_i depends on r_j. In each pair (f_i, r_j), f_i may also depend on some ss and rs other than r_j.

Theorem 1. Any controller $\mathbf{f}(\boldsymbol{r}, \mathbf{s})$ inducing an isolated, stable fixed point in (6) for any $\mathbf{s} \in \mathbb{R}^N$ and $\boldsymbol{r} = K^{-1}\mathbf{s}$ has a connected undirected dependencies graph.

Proof. Suppose there is a controller that has a disconnected undirected dependencies graph. In this case the variables $\mathbf{f}, \boldsymbol{r}, \mathbf{s}$ can be divided in two non-empty subsets α and β, such that no f in α depends on any r or s in β and vice versa. Let us denote the subset of all variables of type $B \in \{f, r, s\}$ in the subset $A \in \{\alpha, \beta\}$ as $\mathbf{B}^A$, e.g. a subset of all motors in α as $\mathbf{f}^\alpha$. Due to Lemma 1 each of the subsets will have an equal number m of motors f and proprioceptory sensors r: $|\mathbf{s}^A| = |\mathbf{f}^A| \equiv m^A$ for $A \in \{\alpha, \beta\}$. We will then have

$$m^\alpha + m^\beta = N. \quad (20)$$

By Lemma 2 neither subset can be composed only of proprioceptive sensors, so m^α and m^β are both greater than zero.

Since the motors in one set cannot depend on proprioceptive sensors in the other, sets α and β will each form its own dynamical system:

$$\begin{aligned} \dot{\boldsymbol{r}^\alpha} &= \mathbf{f}^\alpha(\boldsymbol{r}^\alpha, \mathbf{s}^\alpha), \\ \dot{\boldsymbol{r}^\beta} &= \mathbf{f}^\beta(\boldsymbol{r}^\beta, \mathbf{s}^\beta). \end{aligned} \quad (21)$$

The systems are completely isolated, so the position of fixed points of each only depends on its set of parameters, $\mathbf{s}^\alpha$ for the first dynamical system and $\mathbf{s}^\beta$ for the second. Let us investigate the minimal sizes of these sets. One of the subsets α, β will contain r_1; let us pick α to be definite. In this case the fixed point condition for the α system is

$$\begin{aligned} &f^\alpha_{i_1}(r^\alpha_1 = s_1, ..., \mathbf{s}^\alpha) = 0, \\ &f^\alpha_{i_2}(r^\alpha_{j_2} = s_{j_2} - s_{j_2-1}, ..., \mathbf{s}^\alpha) = 0, \\ &... \\ &f^\alpha_{i_{m^\alpha}}(r^\alpha_{j_{m^\alpha}} = s_{j_{m^\alpha}} - s_{j_{m^\alpha}-1}, ..., \mathbf{s}^\alpha) = 0. \end{aligned} \quad (22)$$

The number of parameters in $\mathbf{s}^\alpha$ should be at least the number of parameters through which $\boldsymbol{r}^\alpha$ is expressed; otherwise, due to the necessary dependence of fs on their corresponding rs, there will be a value of some s for which not all conditions (22) hold. Each condition adds at least one s into the expression for $\boldsymbol{r}^\alpha$, therefore

$$|\mathbf{s}^\alpha| \geqslant m^\alpha. \quad (23)$$

The other subset β does not contain r_1. Same reasoning applied to β leads to the following conditions:

$$\begin{aligned} &f^\beta_{i_1}(r^\beta_{j_1} = s_{j_1} - s_{j_1-1}, ..., \mathbf{s}^\beta) = 0, \\ &... \\ &f^\beta_{i_{m^\alpha}}(r^\beta_{j_{m^\beta}} = s_{j_{m^\beta}} - s_{j_{m^\beta}-1}, ..., \mathbf{s}^\beta) = 0. \end{aligned} \quad (24)$$

The first equation adds at least two proprioceptory sensors into the expression for $\boldsymbol{r}^\beta$ at the fixed point, and each subsequent condition adds at least one. This gives

$$|\mathbf{s}^\beta| \geqslant 1 + m^\beta. \quad (25)$$

Since $\mathbf{s}^\alpha \cap \mathbf{s}^\beta = \varnothing$ and $\mathbf{s}^\alpha \cup \mathbf{s}^\beta = \mathbf{s}$, equations (23), (25) and (20) can be combined to yield

$$|\mathbf{s}| = N \geqslant m^\alpha + m^\beta + 1 = N + 1. \quad (26)$$

Contradiction. ■