

# Evolution of Functional Specialization in a Morphologically Homogeneous Robot

Joshua Auerbach  
Morphology, Evolution and Cognition Lab  
Department of Computer Science  
University of Vermont  
Burlington, VT 05401  
[joshua.auerbach@uvm.edu](mailto:joshua.auerbach@uvm.edu)

Josh C. Bongard  
Morphology, Evolution and Cognition Lab  
Department of Computer Science  
University of Vermont  
Burlington, VT 05401  
[jbongard@uvm.edu](mailto:jbongard@uvm.edu)

## ABSTRACT

A central tenet of embodied artificial intelligence is that intelligent behavior arises out of the coupled dynamics between an agent's body, brain and environment. It follows that the complexity of an agent's controller and morphology must match the complexity of a given task. However, more complex task environments require the agent to exhibit different behaviors, which raises the question as to how to distribute responsibility for these behaviors across the agent's controller and morphology. In this work a robot is trained to locomote and manipulate an object, but the assumption of functional specialization is relaxed: the robot has a segmented body plan in which the front segment may participate in locomotion and object manipulation, or it may specialize to only participate in object manipulation. In this way, selection pressure dictates the presence and degree of functional specialization rather than such specialization being enforced *a priori*. It is shown that for the given task, evolution tends to produce functionally specialized controllers, even though successful generalized controllers can also be evolved. Moreover, the robot's initial conditions and training order have little effect on the frequency of finding specialized controllers, while the inclusion of additional proprioceptive feedback increases this frequency.

## Categories and Subject Descriptors

I.2.9 [Computing Methodologies]: Artificial Intelligence—Robotics

## General Terms

Experimentation

## Keywords

Evolutionary robotics, embodied cognition, artificial intelligence

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '09, July 8–12, 2009, Montréal Québec, Canada.  
Copyright 2009 ACM 978-1-60558-325-9/09/07 ...\$5.00.

## 1. INTRODUCTION

Proponents of embodied artificial intelligence argue that intelligent behavior arises out of the coupled dynamics between an agent's body, brain and environment [8, 1, 16, 5]. One corollary of this view is that the complexity of the agent's controller and morphology must match the complexity of the task at hand. However, more complex task environments require the agent to exhibit different behaviors, which raises the question as to how to distribute responsibility for these behaviors across the agent's controller and morphology. It has been argued [7, 10] that controllers should be organized in a modular fashion such that different control components are responsible for different behaviors, but others have shown that such structural modularity is not always necessary [6, 12, 2].

In addition to modularity in structure, modularity can be thought of in terms of the functions that an agent performs. Moreover, this separation of function can be 'proximal', that is as seen from the point of view of the system itself, i.e. a description from the point of view of a robot's sensory-motor system that accounts for how the agent reacts to different sensory stimulation. It can also be 'distal', i.e. a high level description from the point of view of an independent observer that describes the behavior of an entire sequence of sensory-motor steps [9].

When constructing a system to solve a given problem either through engineering or evolution a mapping is created from a functional space (objectives) to a physical space (how to achieve them). Specifically, the objectives are defined in terms of functional requirements in the functional space and the physical embodiment is defined in terms of design parameters in the physical space. A design is a mapping from the functional requirements to the design parameters. This mapping is not unique and often there are infinitely many viable solutions, but a specific solution is found through the creative process of a human engineer or through an automated process such as evolution [20].

Partly due to the human bias that favors breaking a problem down into separable, simpler sub-problems, roboticists often implicitly design such mappings to be functionally modular in the 'distal' sense: different parts of the robot's body are responsible for different behaviors. For example, wheels or legs may allow for movement while a separate gripper allows for object manipulation. In this work we investigate a robot trained to locomote and manipulate an object, but in which this assumption of functional modularity or specialization of different body parts is relaxed: the robot

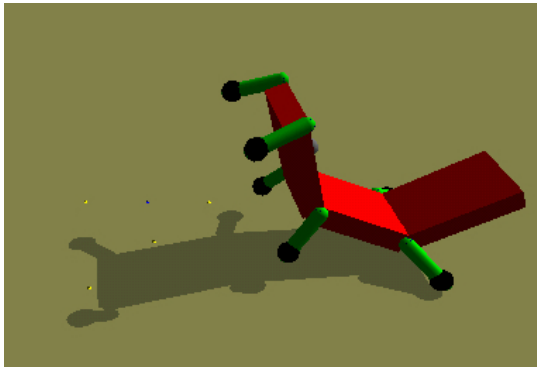


Figure 1: The virtual hexapod robot used in this work.

has a segmented body plan in which the front segment may participate in locomotion and object manipulation, or it may be specialized such that it only participates in object manipulation. In this way, selection pressure dictates the presence and degree of functional specialization rather than enforcing such specialization *a priori*.

In the next section the virtual robot and the incremental shaping method used for training the robot are introduced. The following section reports results demonstrating how changes in initial conditions, training order, and the inclusion of additional proprioceptive feedback affect the success of the evolved controllers and the frequency of evolution discovering functionally specialized controllers. The final section provides some discussion of the observed results, discusses multiple hypotheses that could explain the variability observed in the degree of specialization of evolved controllers across several different experimental regimes, and presents directions for potential future work.

## 2. METHODS

This section first describes the virtual robot used in this work followed by a description of its controller. Next the incremental shaping algorithm used for training the robot is presented. The section concludes with a description of the metrics used to evaluate the evolved controllers.

### 2.1 The robot

In this work a virtual hexapod robot is used (Fig. 1). The robot is composed of three homogeneous body segments attached to each other with one degree of freedom joints that rotate through the robot’s sagittal plane. At the outset of an evaluation period, the segments are arranged horizontally (Fig.3a). The intersegmental joints may rotate neighboring segments toward one another up to  $90^\circ$ . Two legs are attached to the anterior edge of each segment, one on each side. Each leg is attached to its segment with a universal joint that rotates through the sagittal plane with a range of  $[-45^\circ, 45^\circ]$  and through the coronal plane with a range of  $[-45^\circ, 45^\circ]$ . A joint angle of  $0^\circ$  for both degrees of freedom maintains the leg perpendicular to its segment. Each leg is capped with a spherical foot.

Twelve motors actuate the six legs, and another two motors actuate the joints between body segments for a total of 14 motors. A touch sensor and distance sensor reside in each of the two front feet, and a distance sensor is em-

bedded in the robot’s back, for a total of five sensors. The touch sensors return a value of one when the corresponding body part touches another object and zero otherwise. The distance sensors return a value commensurate with the sensor’s distance from the target object: they return zero if they are greater than five meters from the target object and a value near one when touching the target object. Object occlusion is not simulated here; the target object can be considered to be emitting a sound, and the distance sensors respond commensurately to volume.

The robot’s controller is evolved such that the robot locomotes toward, grasps and lifts a rectangular target object placed in its environment.

### 2.2 The controller

The robot is controlled by a continuous time recurrent neural network [4]. The CTRNN is composed of eight motor neurons. Each pair of legs shares two motor neurons: one motor neuron controls rotation through the sagittal plane for both legs, while the other motor neuron controls rotation through the coronal plane for both legs. Sharing motor neurons ensures that when grasping the object the front legs close symmetrically, while also reducing the size of the controller and therefore the dimensionality of the search space. The remaining two motors control the joints between body segments and each receive commands from their own motor neuron. The value of each motor neuron is updated according to

$$\dot{y}_i = \frac{1}{\tau_i} \left( -y_i + \sum_{j=1}^8 w_{ji} \sigma(y_j + \theta_i) + \sum_{j=1}^5 n_{ji} s_j \right) \quad (1)$$

for  $1 \leq i \leq 8$

where  $y_i$  is the state of neuron  $i$ ,  $w_{ji}$  is the weight of the connection from neuron  $j$  to neuron  $i$ ,  $\tau_i$  is the time constant of neuron  $i$ ,  $\theta_i$  is the bias of neuron  $i$ ,  $n_{ji}$  is the weight of the connection from sensor  $j$  to neuron  $i$ ,  $s_j$  is the value of sensor  $j$  and  $\sigma(x) = 1/(1 + e^{-x})$  is the logistic activation function.

The virtual robot with a given CTRNN controller is evaluated over a set number of simulation steps in a physical simulator<sup>1</sup>. For each simulation step, using a step size of 0.0005, the sensors, CTRNN, joint torques and resulting motion of the robot are updated.

### 2.3 Training

The same incremental shaping [19, 11, 18] algorithm presented in [6, 2] is used for dynamically tuning the robot’s task environment to facilitate learning. This method is outlined in Fig. 2. In short, the target object is initially placed in front of the robot such that it learns to grasp and lift the object. Once it does, the target object is moved slightly further away from the robot and training recommences. This process is repeated such that the robot must eventually learn locomotion as well as object manipulation in order to grasp and lift distantly-located objects.

More specifically, a random CTRNN is initially created by choosing all  $\tau$  from the range  $[0.1, 0.5]$ , all  $w$  from  $[-16, 16]$ , all  $\theta$  from  $[-1, 1]$ , and all  $n$  from  $[-16, 16]$ ; these ranges were found useful in previous work [6]. This gives a total

<sup>1</sup>Open Dynamics Engine: [www.opende.org](http://www.opende.org)

of  $8 + 8 * 8 + 8 + 5 * 8 = 120$  evolvable parameters. The robot is then equipped with this controller and allowed to behave in a task environment for 100 time steps in which the target object is placed directly in front of the robot. After evaluation the fitness of the controller is computed as

$$f = \begin{cases} \max_{k=1}^t (D(\text{LFF}, k) * D(\text{RFF}, k)), & \text{if } !g(k) \\ 1 + \max_{k=1}^t (H(\text{TarObj}, k)), & \text{if } g(k) \end{cases} \quad (2)$$

where  $t$  is the number of time steps during the evaluation,  $T(x, k)$  indicates that the touch sensor in body part  $x$  fired during time step  $k$ ,  $D(x, k)$  returns the value of the distance sensor in body part  $x$  during time step  $k$ , and  $H(\text{TarObj}, k)$  indicates the height of the target object from the ground plane. The fitness awarded is therefore conditional on whether the robot has successfully grasped the object, which is defined as

$$g(k) = (T(\text{LFF}, k) == 1) \& (T(\text{RFF}, k) == 1) \& (D(\text{LFF}, k) > 0.89) \& (D(\text{RFF}, k) > 0.89) \quad (3)$$

which ensures grasping is only indicated when both touch sensors in the front feet fire during some time step in the evaluation period, and that both distance sensors in the front feet are sufficiently close to the target object during the same time step. This latter condition allows the robot to distinguish between touching the ground with both feet and touching the object.

If the robot has not yet learned to grasp the object, the upper condition in Eqn. 2 determines fitness, which rewards the robot for minimizing the distance between its front feet and the object. Once it learns to successfully grasp the object the lower condition in Eqn. 2 determines fitness, which rewards the robot for lifting the object as high as possible.

A hill climber [17] is used to optimize the initial random CTRNN against this fitness function. At each generation a child CTRNN is created from the current best CTRNN and mutated. Mutation involves considering each  $\tau, w, \theta$  and  $n$  value in the child, and replacing it with a random value in its range with a probability of  $10/120 = 0.0833$ . This ensures that, on average, 10 mutations are incorporated into the child according to a normal distribution. If the fitness of the child CTRNN is equal to or greater than the fitness of the current best CTRNN, and the child CTRNN is either successful at picking up the target object in either the current or previous environment, then the best CTRNN is replaced by the child; otherwise the child is discarded. This ensures that the grasping behavior learned in previous environments is retained while the locomotion behavior is adapted to the current environment.

After each possible replacement, the current CTRNN is considered in order to determine whether a failure condition has occurred, or whether it has achieved the success criteria. In the present work the failure condition is defined as 100 generations of the hill climber elapsing before a successful CTRNN is found. A successful CTRNN is defined as one for which, at some time step during the current evaluation both front feet touch the target object and it is lifted off the ground above a certain threshold.

If the failure condition occurs, the task environment is eased; if the current CTRNN succeeds, the task environment is made more difficult. Easing the task environment

1. **IncrementalShaping()**
2.   Create and evaluate random parent  $p$
3.   **WHILE**  $\sim$ Done()
4.     Create child  $c$  from  $p$ , and evaluate
5.     **IF**  $\text{Fitness}(c) \geq \text{Fitness}(p)$  **AND**  
      (  $\text{PreviousSuccess}(c)$  **OR**  $\text{Success}(c)$  )  
      [see Eqns. 2,3]
6.        $p = c$
7.     **IF** Failure()
8.       EaseEnvironment()
9.       Re-evaluate  $p$
10.    **WHILE**  $\text{Success}(p)$
11.     HardenEnvironment()
12.     Re-evaluate  $p$

---

13. **Done()**
14.   30 hours of CPU time have elapsed **OR**  
      TargetDistance > 10m

---

15. **Failure()**
16.   100 generations since last success

---

17. **EaseEnvironment()**
18.   EvaluationTime  $\leftarrow$  EvaluationTime+100

---

19. **Success( $g$ )**
20.    $\exists k, k \in \{1, \dots, t\} |$
21.     $T(\text{LeftFrontFoot}, k)$ **AND**
22.     $T(\text{RightFrontFoot}, k)$ **AND**
23.    (  $\min(D(\text{LeftFrontFoot}, k), D(\text{RightFrontFoot}, k))$   
       $\geq 0.89$  ) **AND**
24.     $H(\text{TargetObject}, k) > 1.5$

---

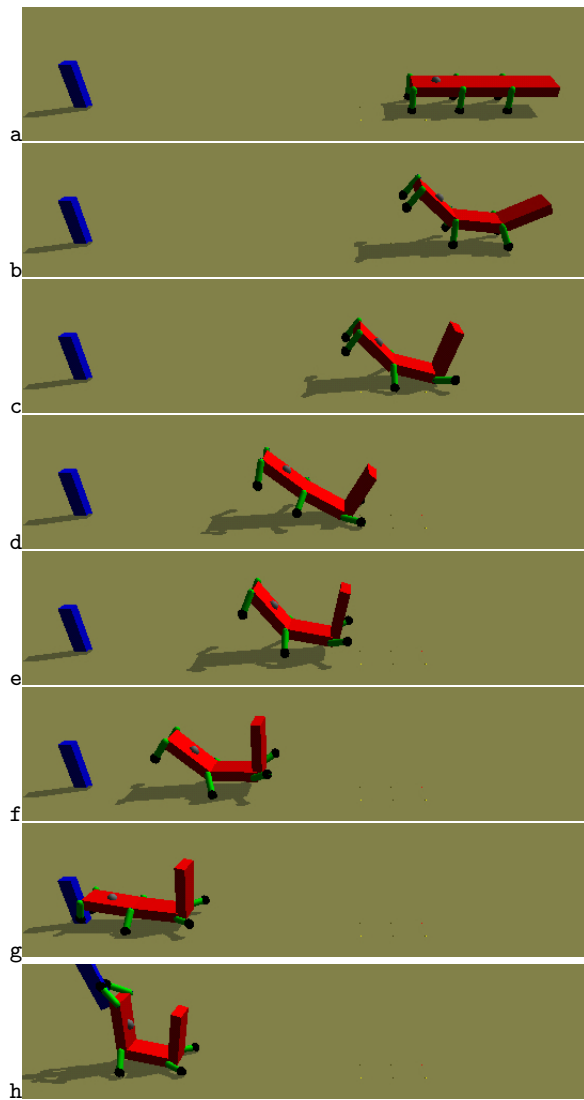
25. **PreviousSuccess( $g$ )**
26.   TargetDistance  $\leftarrow$  TargetDistance-0.01m
27.   success =  $\text{Success}(g)$
28.   TargetDistance  $\leftarrow$  TargetDistance+0.01m
29.   **RETURN** success;

---

30. **HardenEnvironment()**
31.   TargetDistance  $\leftarrow$  TargetDistance+0.01m

**Figure 2: Incremental shaping pseudocode. The algorithm executes a hill climber [1-14] (see text for description). If the current genome fails [15,16], the task environment is eased [17,18]; while it is successful [19-24], the task environment is made more difficult [30,31].  $T(x, k)$  returns 1 if body part  $x$  is in contact with another object and zero otherwise at time step  $k$ .  $D(x, k)$  returns the value of the distance sensor located at body part  $x$  at time step  $k$ .  $H(x, k)$  returns the height of object  $x$  at time step  $k$**

involves increasing the current evaluation period by 10 time steps. This has the effect of giving the robot more time to succeed at the current task if it fails. Making the task environment more difficult involves moving the target object further away from the robot. This has the effect of teaching the robot to grasp and lift the target object when it is close, and learning to locomote toward the target object, followed

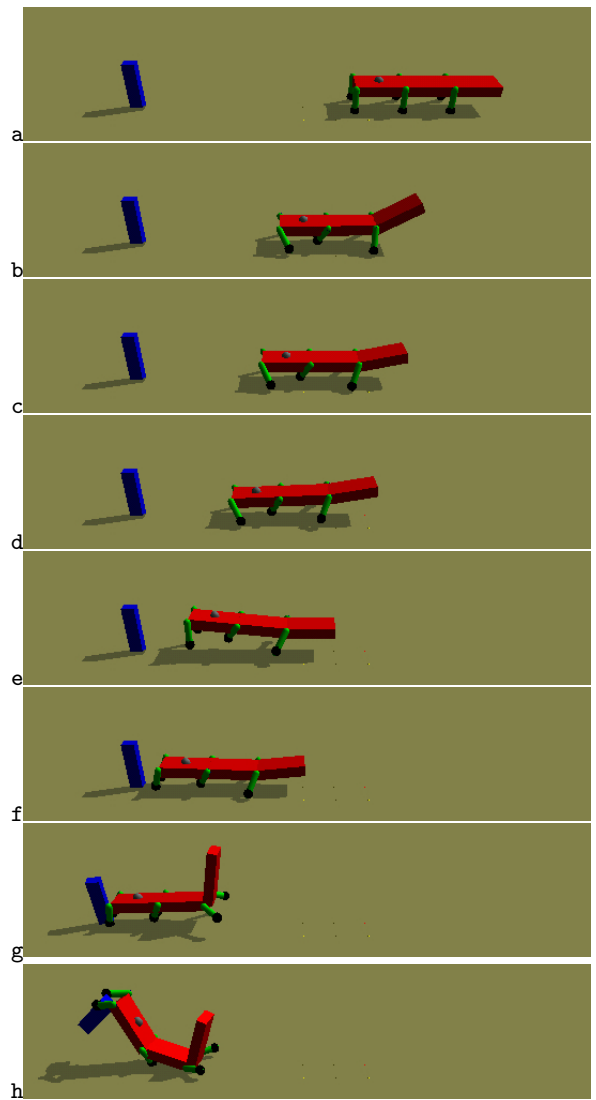


**Figure 3: Sample functionally specialized controller.** The robot’s front body segment is raised and the front feet are kept off the ground during locomotion, i.e. they are only used for grasping the target object.

by grasping and lifting it, when it is placed further away. As some CTRNNs that succeeded for a given target object distance also succeed when the target object is moved further away, the target object is continually moved until the current CTRNN no longer succeeds, at which time hill climbing recommences. In order to further speed the algorithm an individual evaluation is terminated early if the robot ceases to move before succeeding at the task.

## 2.4 Evaluating functional specialization

The two main questions of interest in the current work are (1) whether a single CTRNN acting as a monolithic controller for this robot can evolve to successfully locomote toward, grasp and lift the target object, and (2) if so whether the evolved controllers are functionally specialized in the ‘distal’ sense or not. To answer the first question it is sufficient to consider the distance of the target object from the



**Figure 4: Sample functionally generalized controller.** This controller uses the robot’s front legs for propulsion during locomotion and for grasping and lifting of the target object.

robot at the end of training. The greater this distance, the more simulations were performed in which the robot was considered to be successful, and the more rapidly the controller was able to adapt to changing environmental conditions. This metric will be referred to as the **adaptation rate**.

To investigate the second question one must consider that the robot’s serially homogeneous body plan was designed such that it may locomote using all six legs or, alternatively, may rotate the anterior (or posterior) segment upward to locomote using the four middle and posterior (or anterior and middle) legs. A controller may then involve the front legs in both locomotion and grasping by keeping the front segment horizontal or, alternatively, it may restrict the front legs such that they only contribute to object manipulation.

These latter controllers would realize functional specialization if locomotion and object manipulation are considered as two separate functions.

In order to evaluate whether a given successful controller is functionally specialized or not the simulation is run until the controller grasps the target object, while recording the sensor values during each time step. At the completion of this simulation the percent of total time steps during which both front feet touch sensors fire is calculated. Controllers with low values for this metric are considered to be functionally specialized because the robot rarely touches its front feet to the ground during locomotion. Conversely, controllers that use their front feet both for locomotion (either for propulsion, balance or both) and grasping are not functionally specialized and will receive higher values from this test. See Fig. 3 for an example of a functionally specialized controller (feet only touch in 0.076% of time steps), and Fig. 4 for an example of a functionally generalized controller (feet touch in 48.693% of time steps).

### 3. RESULTS

Using the above methods four different experimental regimes were investigated and their results compared. Each regime consisted of running 100 independent trials of the incremental shaping algorithm (Fig. 2) with identical initial environmental conditions but different randomly-generated controllers. In the first experiment (**regime 1**) the front body segment joint was rotated upward  $90^\circ$  such that it was perpendicular to the ground with the front feet pointing forward and the target object was initially placed directly in front of the robot. All of the runs from this regime can be considered successful in the sense that they were able to adapt to target objects placed at distances greater than three meters (a distance that requires locomotion), grasp, and lift up the object (see Fig. 5). Additionally, many of the runs from this regime resulted in functionally specialized controllers (black bars in Fig. 6).

In the second regime (**regime 2**) the robot was initialized with both body segments horizontal so that all six feet started on the ground and again the target object was initially placed directly in front of the robot. It was assumed that starting the robot flat would bias evolution to initially discover and retain locomotion involving all six legs, and therefore not specialize the front legs only for grasping. However, while still finding successful controllers in the majority of trials, the number of controllers resulting from this regime that developed functionally specialized controllers was similar to regime 1, and in fact counter to intuition more controllers from this regime caused the robot to touch their feet to the ground in less than 5% of time steps as compared with regime 1 (red bars in Fig. 6).

In the third experiment (**regime 3**) the body segments started horizontally, but in this case the target object was initially placed two meters away from the robot, so that before learning to grasp the target object the robot would first be forced to learn to move toward it. Without initial evolutionary pressure to involve the front legs in grasping it was assumed that the controllers to evolve in this experiment would be more likely to include them in locomotion, but once again a similar number of controllers resulting from this experiment developed functionally specialized controllers as compared with regimes 1 and 2 (yellow bars in Fig. 6).

The fourth regime (**regime 4**) was identical to regime 2 in that the body segments were started parallel to the ground with the target object initially directly in front of the robot. However, for this experiment two additional sensors were added to the robot: joint angle sensors for the two joints connecting the body segments, and these were wired to the controller. The controllers that evolved in this regime not only performed better in the sense that they adapted more rapidly to changes in the target object’s position during training as compared to regime 2 (Fig. 7), but also were more likely to be functionally specialized when compared to the other three regimes (blue bars in Fig. 6).

### 4. DISCUSSION AND CONCLUSIONS

After noting that all four regimes were able to successfully learn both locomotion and object manipulation in the majority of trials the question arises as to why evolution tends to converge on functionally specialized behaviors, and why the inclusion of additional sensors causes an increase in the frequency of converging on such behaviors. Three possible hypotheses are: (1) functionally specialized controllers are more evolvable, and therefore supplant less specialized controllers during an evolutionary run, (2) evolution initially discovers a specialized or generalized controller, and subsequently improves on that behavior but does not increase or decrease specialization, and (3) functionally specialized behaviors more easily allow for active perception [15].

Hypothesis (1) is supported by previous work, which has indicated that modularity can increase evolvability [21], but only under certain environmental conditions [13, 14]. However, Fig. 5 indicates that for two of the four regimes (regimes 2 and 4) studied here, adaptation rate is similar between those runs that converged on functionally specialized behaviors and those that converged on generalized behaviors, and in fact adaptation rate was lower within runs containing specialists compared to generalists in the other two regimes (regimes 1 and 3). This suggests that functionally specialized behaviors do not arise because they are more evolvable, but for some other reason.

Hypothesis (2) suggests that evolution may become ‘locked in’ to a specialized or generalized strategy, depending on which type it discovers at the outset: it may be difficult to subsequently evolve the robot’s controller to selectively tune the amount of behavioral specialization of one part of the body. It follows from this that the amount of specialization may be biased by the initial conditions of the robot during shaping. If scaffolding teaches grasping before locomotion or, more strongly, begins with the front segment raised vertically, controllers may converge on behaviors that allow the front legs to grasp the object, but evolution cannot subsequently co-opt those legs to participate in locomotion as well. However, this hypothesis is contradicted by Fig. 5, which indicates that changing the initial conditions to favor usage of the front legs in locomotion (regimes 2 and 3) do not produce more generalized controllers: these regimes also converge in the majority of runs on functionally specialized controllers. Hypothesis (2) is further invalidated by the run illustrated in Fig. 8, which shows that evolution may in some cases co-opt the front legs for increased participation in locomotion.

According to hypothesis (3), it may be that the robot is better able to actively perceive the proximity of the object—and therefore determine desirable conditions for lifting—if

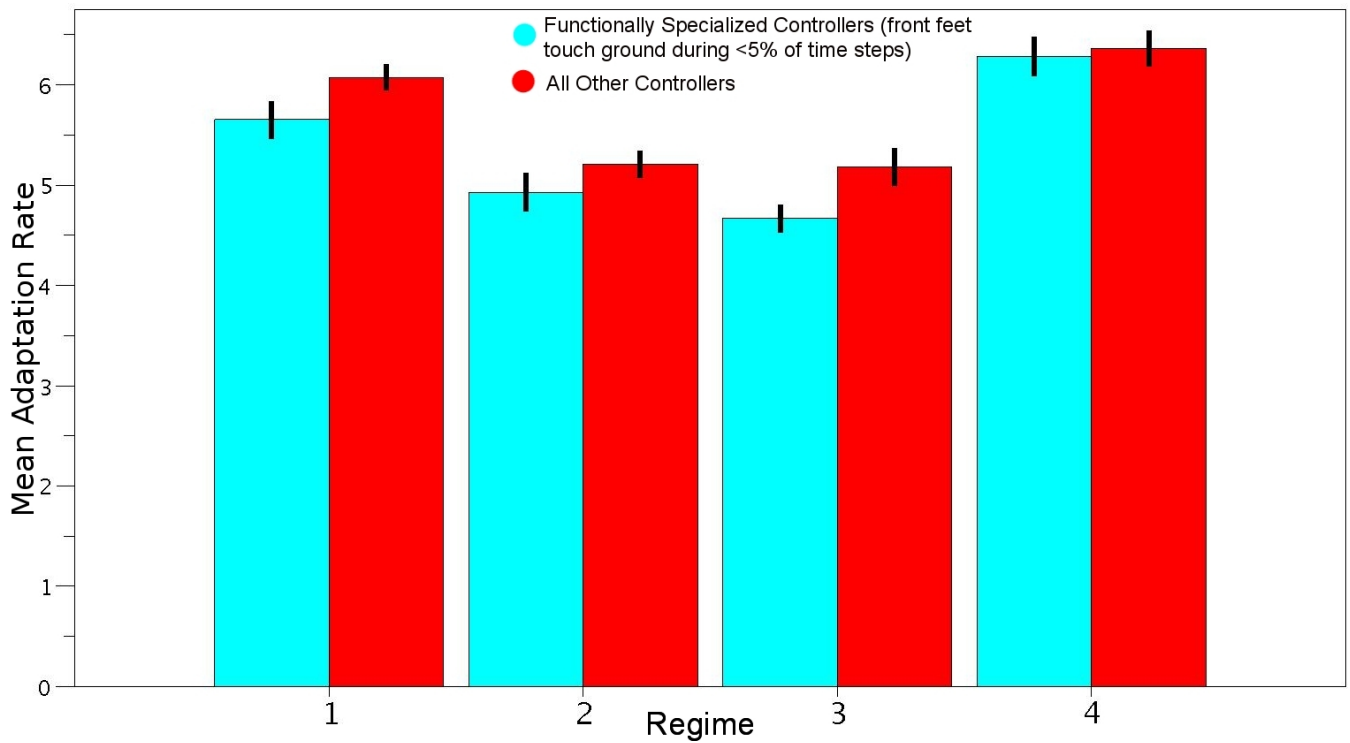


Figure 5: Plot of mean adaptation rate by regime with standard error bars shown. Data is split between those controllers that cause the robot's feet to touch the ground during less than 5% of time steps (leftmost grouping in Fig. 6) and all others.

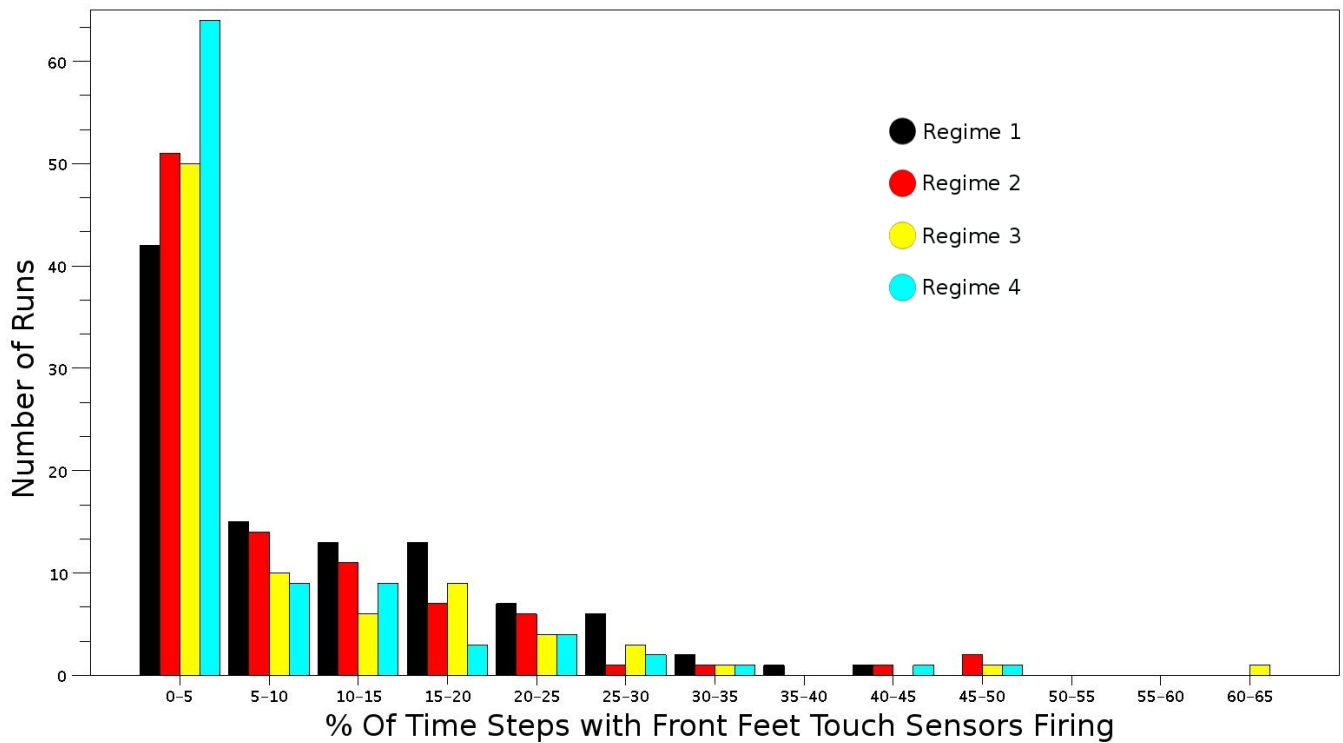
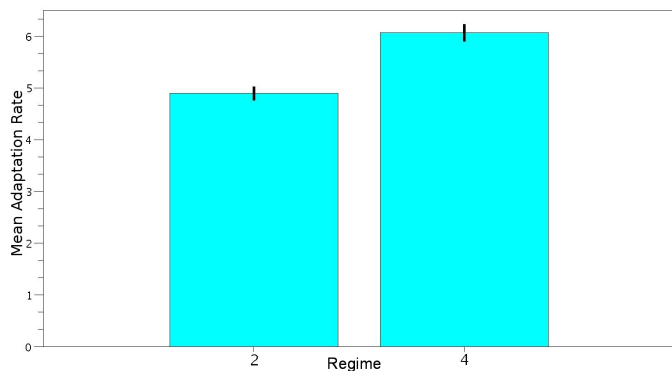


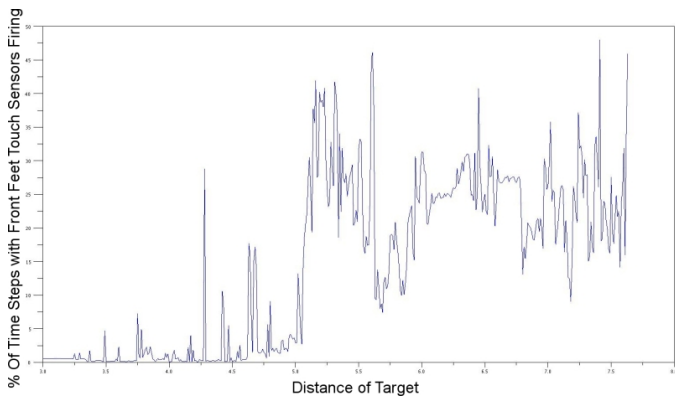
Figure 6: Histogram of the specialization metric for each of the four regimes. All runs in which the target object reached at least three meters are included (100 runs from regime 1, 94 runs from regime 2, 85 runs from regime 3, and 94 runs from regime 4).





**Figure 7: Plot of mean adaptation rate with standard error bars for regimes 2 and 4.**

the front legs do not participate in locomotion, because then the touch sensors will only fire when in contact with the target object. Such controllers may be easier for the evolutionary process to find and optimize. Indeed, it has been demonstrated in the literature that active categorical perception may evolve in learning agents [3]. Moreover, providing the robot with additional proprioceptive feedback in regime 4 not only increased the prevalence of functional specialization (as shown in Fig. 6), but also the adaptation rate within those runs that produced specialized controllers (as shown in Fig. 7). It is plausible that these added sensors allow for better active perception as the touch sensors and sensed body posture may together indicate appropriate conditions for object manipulation.

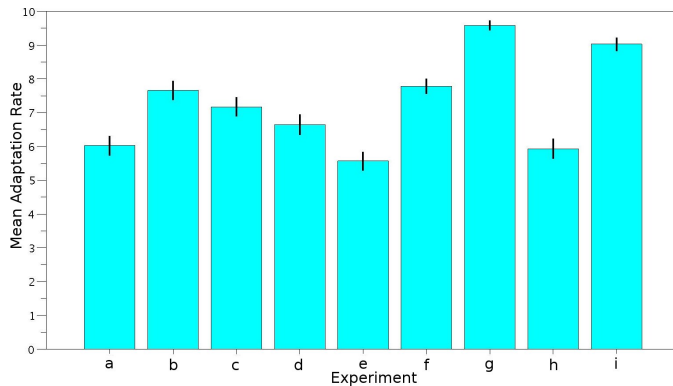


**Figure 8: Target object distance where controller was successful vs. % of time steps with front feet touch sensors firing from a single evolutionary run.**

Several additional experiments were designed to test this hypothesis. These experiments followed the theme of regimes 2 and 4. Specifically, in all cases the body segments were started parallel to the ground with the target object initially directly in front of the robot. What varied in these experiments were the sensors the robot was equipped with. Since a variable number of sensors results in a variable number of parameters under evolutionary control these experiments all used a fixed mutation rate of  $\frac{10}{288} \approx 0.035$ . Experiment **a** used the same sensors as regime 1 above, these sensors will be referred to as the base sensor set. Experiment **b**

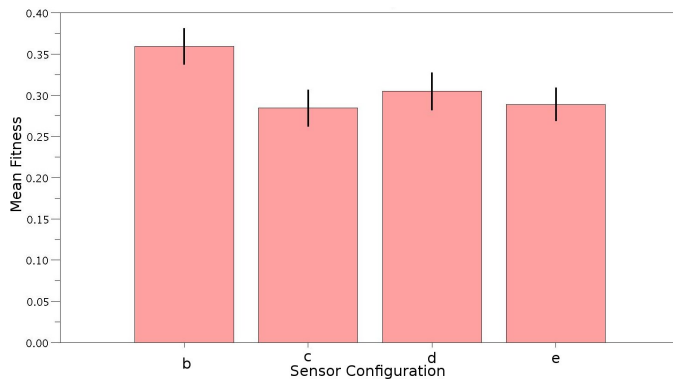
used the sensors of regime 4: the base sensor set with two joint angle sensors on the two joints connecting the main body segments added in. Experiment **c** used a robot with the base sensor set plus two more joint angle sensors: one apiece for the two degrees of freedom of the front left leg (just the left leg was used, because due to the construction of the controller the left and right legs operated symmetrically). Experiment **d** used a robot with the base sensor set plus two additional joint angle sensors on the middle left leg, and similarly experiment **e** used a robot with the base sensor set plus two additional joint angle sensors on the rear left leg. Experiment **f** used a robot with the base sensor set plus all the joint angle sensors featured in experiments **b-d**, while experiment **g** used a robot with the base sensor set plus touch sensors on the rear four feet. Experiment **h** used a robot with the base sensor set plus distance sensors on the rear four feet, and finally experiment **i** used a robot with all the sensors of experiment **f** plus the additional touch sensors and distance sensors on the rear four feet used in **g** and **h**.

Fig. 9 shows the mean adaptation rates with standard error bars for each of these additional experiments. Note the steady decline in performance from experiment **b** through experiment **e**. This result provides further evidence for hypothesis (3) as it demonstrates that adaptation rate declines as the included sensors provide less information in regards to desirable conditions for lifting: the main body joints (**b**) are most informative as discussed above, while the front leg angles may provide some information about the relative position of the front feet. As the sensors are moved toward the rear of the body less of this relevant information is available. This is further demonstrated by experiment **f** which shows that including all of the joint angle sensors buys the robot very little above just including the most useful pair (**b**). Additionally it is seen from experiment **g** that additional touch sensors improve performance even more so than any angle sensors do, because touch sensors provide the most direct evidence as to which feet are on the ground and/or touching the target object.



**Figure 9: Mean adaptation rate with standard errors for additional experiments, see text for details.**

To verify that the additional sensors provide relevant information useful for the current task and do not merely aid in locomotion, virtual robots were instantiated with the sensor configurations of experiments **b-e** and were evolved for locomotion alone. This consisted of expanding the range of the robot’s distance sensors and placing the target object a large (100 m) distance away. Fitness was calculated as



**Figure 10: Mean fitness with standard errors when selecting for just locomotion with the four different pairs of joint angle sensors.**

the fraction of distance between the start location and the target object location that the robot was able to cover in a set amount of time. Fig. 10 shows the mean fitnesses along with standard error bars from these experiments grouped by sensor configuration. Note that while including the joint angle sensors on the joints connecting the main body segments (**b**) leads to improved locomotion performance, there is no significant difference between the performance of the other three sensor sets. This provides further evidence that the differences observed across these configurations above are due to active perception.

In conclusion, it was shown here that evolution can tune the amount of functional specialization of different parts of the body. In future work we plan to evolve morphology as well as control: it is predicted that evolution would then specialize both the morphology and function for different body parts as the task environment dictates. This may prove to be a more fruitful method for realizing robots capable of an increasing number of behaviors, rather than fixing the body plan and manually assigning function to structure.

## 5. REFERENCES

- [1] M. Anderson. Embodied Cognition: A field guide. *Artificial Intelligence*, 149(1):91–130, 2003.
- [2] J. Auerbach and J. C. Bongard. How robot morphology and training order affect the learning of multiple behaviors. In *Proceedings of the IEEE Congress on Evolutionary Computation*, 2009. To Appear.
- [3] R. Beer. The Dynamics of Active Categorical Perception in an Evolved Model Agent. *Adaptive Behavior*, 11(4):209, 2003.
- [4] R. D. Beer. Parameter space structure of continuous-time recurrent neural networks. *Neural Comp.*, 18(12):3009–3051, 2006.
- [5] R. D. Beer. The dynamics of brain-body-environment systems: A status report. In P. Calvo and A. Gomila, editors, *Handbook of Cognitive Science: An Embodied Approach*, pages 99–120. Elsevier, 2008.
- [6] J. Bongard. Behavior chaining: incremental behavioral integration for evolutionary robotics. In S. Bullock, J. Noble, R. Watson, and M. A. Bedau, editors, *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 64–71. MIT Press, Cambridge, MA, 2008.
- [7] R. Brooks. A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of [legacy, pre - 1988]*, 2(1):14–23, 1986.
- [8] R. Brooks. *Cambrian intelligence*. MIT Press Cambridge, Mass, 1999.
- [9] R. Calabretta, S. Nolfi, D. Parisi, and G. P. Wagner. Emergence of functional modularity in robots. In *Proceedings of the fifth international conference on simulation of adaptive behavior on From animals to animats 5*, pages 497–504, Cambridge, MA, USA, 1998. MIT Press.
- [10] R. Calabretta, S. Nolfi, D. Parisi, and G. P. Wagner. Duplication of modules facilitates the evolution of functional specialization. *Artificial Life*, 6(1):69–84, 2000.
- [11] M. Dorigo and M. Colombetti. Robot shaping: Developing situated agents through learning. *Artificial Intelligence*, 70(2):321–370, 1994.
- [12] E. Izquierdo and T. Buhrmann. Analysis of a dynamical recurrent neural network evolved for two qualitatively different tasks: walking and chemotaxis. In S. Bullock, J. Noble, R. Watson, and M. A. Bedau, editors, *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 257–264. MIT Press, Cambridge, MA, 2008.
- [13] N. Kashtan and U. Alon. Spontaneous evolution of modularity and network motifs. *Proc Natl Acad Sci U S A*, September 2005.
- [14] H. Lipson, J. Pollack, and N. Suh. On The Origin of Modular Variation. *Evolution*, 56(8):1549–1556, 2002.
- [15] A. Noë. *Action In Perception*. MIT Press, 2004.
- [16] R. Pfeifer and J. Bongard. *How the Body Shapes the Way We Think: A New View of Intelligence*. MIT Press, 2006.
- [17] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, second edition, December 2002.
- [18] L. Saksida, S. Raymond, and D. S. Touretzky. Shaping robot behavior using principles from instrumental conditioning. *Robotics and Autonomous Systems*, 22:231–249, 1997.
- [19] S. P. Singh. Transfer of learning across sequential tasks. *Machine Learning*, 8:323–339, 1992.
- [20] N. P. Suh. *The Principles of Design (Oxford Series on Advanced Manufacturing)*. Oxford University Press, March 1990.
- [21] G. Wagner and L. Altenberg. Perspective: Complex adaptations and the evolution of evolvability. *Evolution*, 50(3):967–976, 1996.