Synthesizing Physically-Realistic Environmental Models from Robot Exploration

Josh Bongard

Department of Computer Science, University of Vermont
33 Colchester Ave., Burlington, VT 05405
josh.bongard@uvm.edu `www.cs.uvm.edu/~jbongard`

Abstract. In previous work [4] a framework was demonstrated that allows an autonomous robot to automatically synthesize physically-realistic models of its own body. Here it is demonstrated how the same approach can be applied to empower a robot to synthesize physically-realistic models of its surroundings. Robots which build numerical or other non-physical models of their environments are limited in the kinds of predictions they can make about the repercussions of future actions. In this paper it is shown that a robot equipped with a self-made, physically-realistic model can extrapolate: a slow-moving robot consistently predicts the much faster top speed at which it can safely drive across a terrain.

1 Introduction

Modeling has long played a part in robotics research, but has been plagued by two main challenges: models must either be laboriously created by hand, or (whether manually or automatically created) they are of limited use for making predictions about the outcome of future actions. Models have also been used extensively in evolutionary robotics [13], where the goal is to use evolutionary algorithms to automate the creation of behaviors. There are three main approaches to evolutionary robotics: controllers are either evolved directly on the physical device, requiring thousands of evaluations [6][8]; controllers are adapted from an existing, hand-designed controller [17]; or a hand-designed simulator is used to evolve controllers in simulation before transferal to the physical device [10][14]. The first approach is infeasible for continuous, rapid adaptation; the second approach requires a human to create the starting behavior; and the third approach requires a human to craft a simulation of the robot.

In previous work [2][4] a fourth method was introduced that overcomes these obstacles by allowing the robot to evolve simulations of itself, and then use the best of the evolved simulations to internally rehearse behaviors before attempting them in reality. It was found that the models allowed the robot to make successful predictions regarding future actions: it first created a self-model using only gentle rocking motions; it then optimized a locomotion pattern using the self-model; and finally it executed that pattern successfully in reality.

Here it is demonstrated that the same technique can allow a robot to model its surroundings, and then make successful predictions about how it can interact with that environment. Specifically, it is shown that a robot can learn the topological properties of a terrain and then predict the maximum speed at which it can traverse that terrain without tipping over. Many techniques exist for allowing a robot to model its environment (or the nature of its interaction with that environment), dating back to the first research with autonomous robots [12]. More recent approaches have embraced probabilistic models (for an overview see [18]). Other approaches include compiling a database of past experiences [11] or training neural networks to reproduce sensor-motor correlations [9]. However, the models from all of these approaches are limited, in that they only allow the robot to make predictions about future actions that have either been performed before, or are some aggregate of those actions. For example in [9], the robot can predict future change in its position given the current acceleration, as long as that acceleration falls within the space of past accelerations that were used to train the model.

In the next section the robot, its environment, and the technique it uses to make successful predictions about new behaviors in that environment is described. In section 3 results generated using this approach are presented, and in the final section some discussion and concluding remarks are provided.

2 Methods

Here it is assumed that a robot wishes to reconstruct the topological features of its surrounding terrain, in order to better understand how it can interact with it: for instance, what is the top speed at which it can drive across the terrain without tipping. Clearly, the robot should not simply drive fast to determine this threshold, as it may actually tip over. Rather it should move slowly, synthesize an accurate topological model of that terrain, and then drive a virtual copy of itself across that terrain to determine its safest top speed.

This process is executed as follows. First the robot slowly traverses the terrain it wishes to model, known as the *target terrain*. In this work, a virtual robot and environment are used in lieu of a physical robot and environment: in future work a physical robot will be used. Two examples of the robot and its environment are shown in Fig. 1.

This traversal generates sensor data, which the robot stores. The robot then generates a series of environmental simulations, known as *model terrains*. It drives a virtual copy of itself across each model terrain, which again generates sensor data. It then compares the model-generated sensor data against the sensor data from the target terrain. The closer the match, the more accurate the current model terrain must be. The robot then uses a hill climber [15] to optimize the model terrains by minimizing the distance between the model-generated and target-generated sensor data.

Both the target and model environments are simulated using physical simulation[1]. In a physical simulation, all objects have physical properties such as mass, velocity, and friction. Objects are connected to one another using joints,

[1] www.ode.org

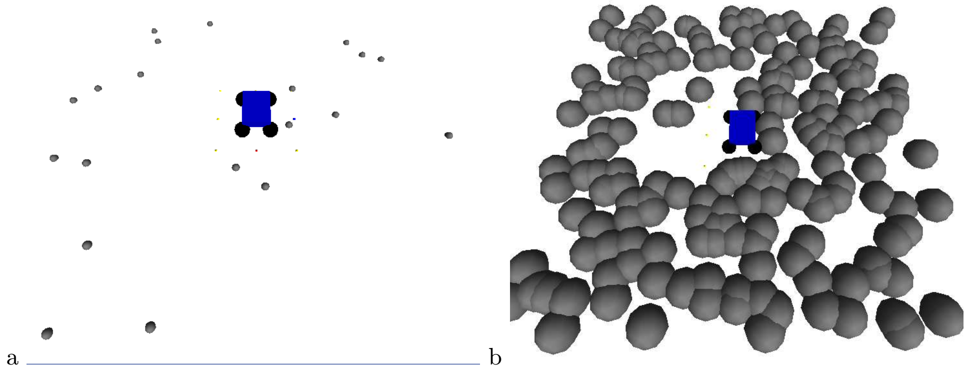

Fig. 1. The robot, and two sample environments. The wheeled robot is shown at center. **a:** The robot driving across a terrain with 10 boulders each with a radius of 10cm. **b:** The same robot traversing a terrain with 200 boulders with radius 45cm.

which may be powered. At each time step of the simulation, the positions of the objects are updated based on their current acceleration, and the internal and external forces acting on them.

The robot used here is made up of five objects: a rectangular body and four spherical wheels. The wheels are driven at a constant rotational velocity. The wheels are connected directly to the body: turning is accomplished by driving the left and right wheels at slightly different velocities. Every traversal of a terrain is accomplished as follows. The robot begins driving forward, with the left wheels moving at 0.6 revolutions per second and the right wheels at 0.4 rev/sec (or vice versa). The robot can sense, in degrees, how much its main body tilts to the left or right, and how much it tilts forward and backward. If the robot moves outside the terrain, it will not tilt for a period of time as it is on flat ground. This acts as a signal to the robot to reverse direction, and change the velocity differential on its left and right wheels.

Environments are created by altering two parameters: the number of boulders within the terrain, and the radius of those boulders. The boulders are embedded within the (flat) ground, and cannot be moved by the robot. It is assumed that the robot knows that the boulders are embedded and immovable, but not how many and how large they are. Boulders are distributed randomly, using a uniform distribution, across the terrain.

3 Results

The robot first traversed three environments with 10 boulders of radius 10cm, 100 boulders of radius 10cm, and 200 boulders of radius 10cm. Fig. 2a,c and e reports the time series of the resulting tilt sensor data collected by the robot from these three environments.

It was found (data not shown) that there was no correlation between the distance between two given environments (as characterized by the number of

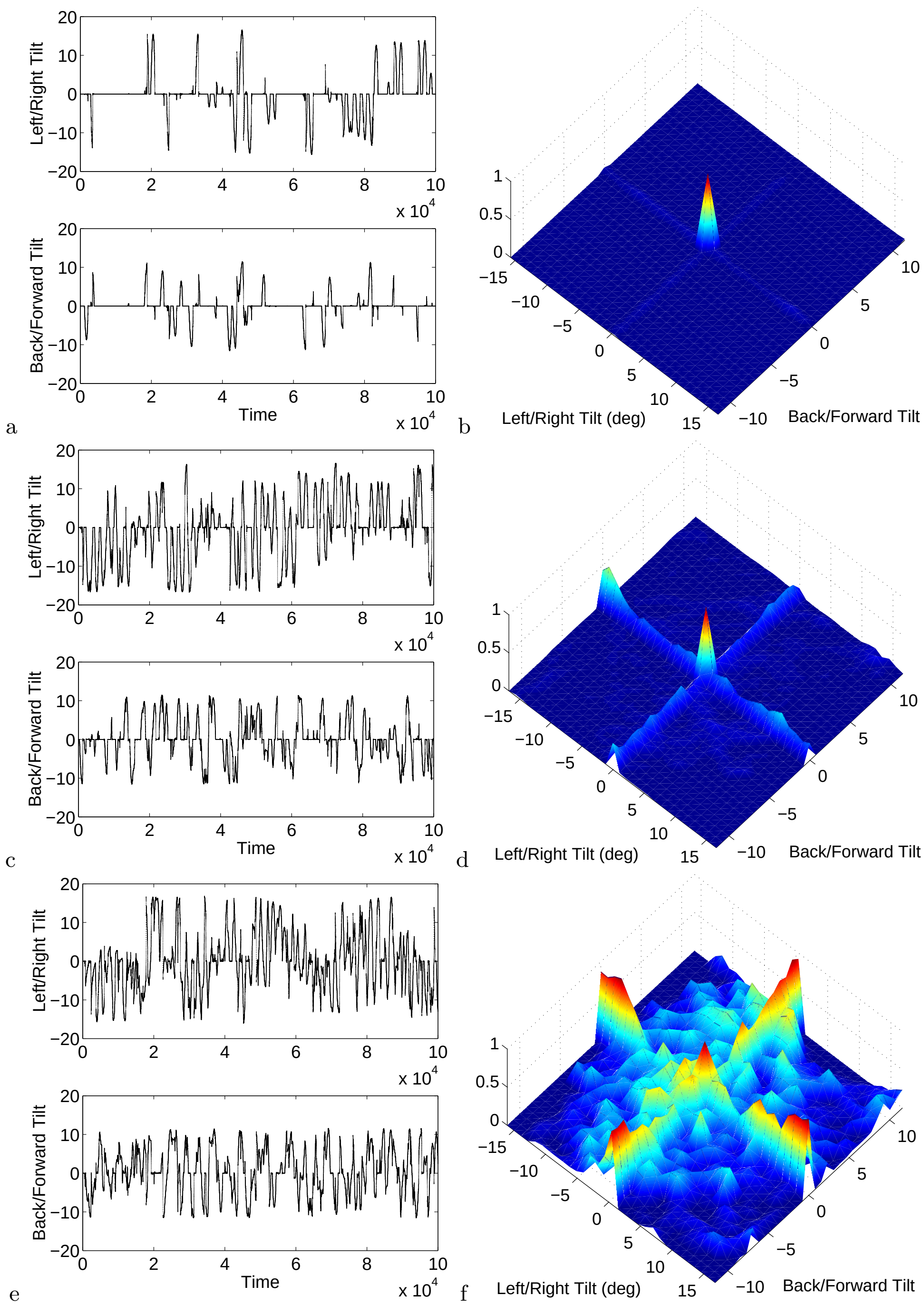

Fig. 2. Robot experience when traversing three different terrains. The robot sensor time series (**a**) and binned data (**b**) for environment 1. The time series (**c**) and binned data (**d**) for environment 2. The time series (**e**) and binned data (**f**) for environment 3.

boulders), and the difference between time series sensor data from those environments. For instance for the three environments described above, environments 1 and 3 are more different from each other than 1 is from 2, or 2 is from 3. Yet the distance between the sensor data from environments 1 and 3 is no larger than the distance between the sensor data from environments 1 and 2, or from environments 2 and 3.

However, a correlation *was* found when the sensor data was collected into bins. Each bin denotes the amount of time (during a traversal) that the robot's main body maintained a certain orientation. This data is shown for the robot traversing environment 1 in Fig. 2a, in which the main body is mostly flat. For environment 2 (Fig. 2d), the larger number of boulders caused the robot to list onto its left wheel pair, right wheel pair, back wheel pair, or forward wheel pair (represented by the four spines in the figure). In environment 3, the many boulders caused the robot to jostle around, therefore spending time in many more orientation regimes. Taking the absolute difference between each bin across two binned data sets, and summing those differences gives a single value indicating the distance between these two data sets. This signal was found to differ more between the data sets from environments 1 and 3 than either the distances between the data sets from environments 1 and 2, or environments 2 and 3.

This observation is shown more generally in Fig. 3. The robot was commanded to traverse 10 different environments, with an increasing number of boulders. For each environment, the sensor data was binned as explained above, and for each pair of environments, the difference between the binned data was calculated. As can be seen, a clear correlation exists: the greater the difference between two environments, the greater the distance between the corresponding pair of binned sensor data.

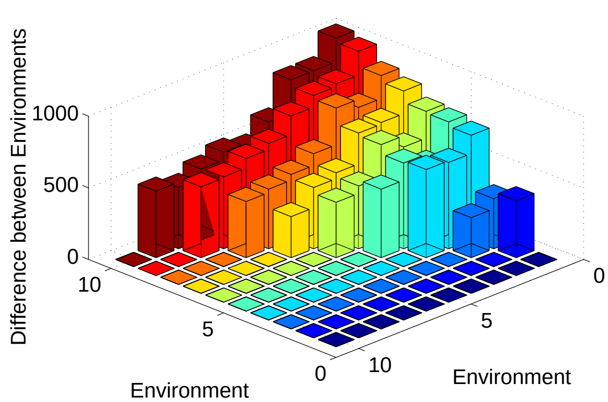

Fig. 3. Differences between sensor data collected from 10 different environments, with increasing numbers of boulders.

The robot was then commanded to use this indirect measure of environment similarity to infer the topology of a *hidden environment*: a target environment in which the number, distribution and radii of the boulders is unknown. For each trial, the robot began by traversing the hidden environment, recording the resulting sensor data, and binning it. The sensor values were stored in a 30×30 matrix of bins over the range $[-60^o, 60^o]^2$. Values beyond this range indicate that the robot is about to tip over, and the simulation was stopped prematurely. A random model environment was then created by choosing the number of boulders from the range $[1, 200]$ using a uniform distribution, and the radii of the boulders from the range $[0.05, 0.45]$ using a uniform distribution. (The ranges indicate the minimum and maximum possible values for these two environmental parameters.)

A model environment is then created from these two parameter values, traversed, and sensor data is collected and binned. Then, the distance between the model-generated and target-generated binned sensor data is computed. A child model environment is then created from the current model environment as follows. The two environmental parameters are copied, and with 50% probability the boulder number parameter is mutated; otherwise, the boulder radii parameter is mutated. If the boulder number parameter is mutated, a Gaussian value is chosen from $[-10, 10]$ and added to the current value. If the boulder radii is mutated, a Gaussian value is chosen from $[-0.1, 0.1]$ and added to the current value. A child model environment is then synthesized from these two new parameter values. (Note that as the distribution of boulders is still random, the same parameter values may generate different environments.)

The robot traverses the child model environment, and the sensor data is collected and binned. If the distance between the binned sensor data from the target and child model environments is less than the distance between the data from the target and parent model environments (in other words, if the child model environment is more accurate than the parent model environment) the parent model environment is discarded and replaced by the child model environment. Otherwise, the child model environment is discarded. This process is repeated for 100 generations.

Thirty independent trials were conducted using this process. For both the target and model environments, the robot was allowed to traverse them for 20,000 time steps of the simulation. Figure 4a reports the performance of these trials. As can be seen, the error (ie. the distance between the binned data from the target and model environment) gradually decreases over evolutionary time, indicating that this metric provides some gradient in the space of all possible model environments. However, the number of boulders encoded does not approach the true number of boulders in the target environment, which was fixed at 100. The models do however converge on the correct value for the boulder radii, which was fixed at 0.25, indicating that the robot successfully infers the size of the boulders in the environment indirectly, given only tilt sensor information.

[2] This differs from the 20×20 matrix of bins over the range $[-15^o, 15^o]$ shown in Fig. 2.

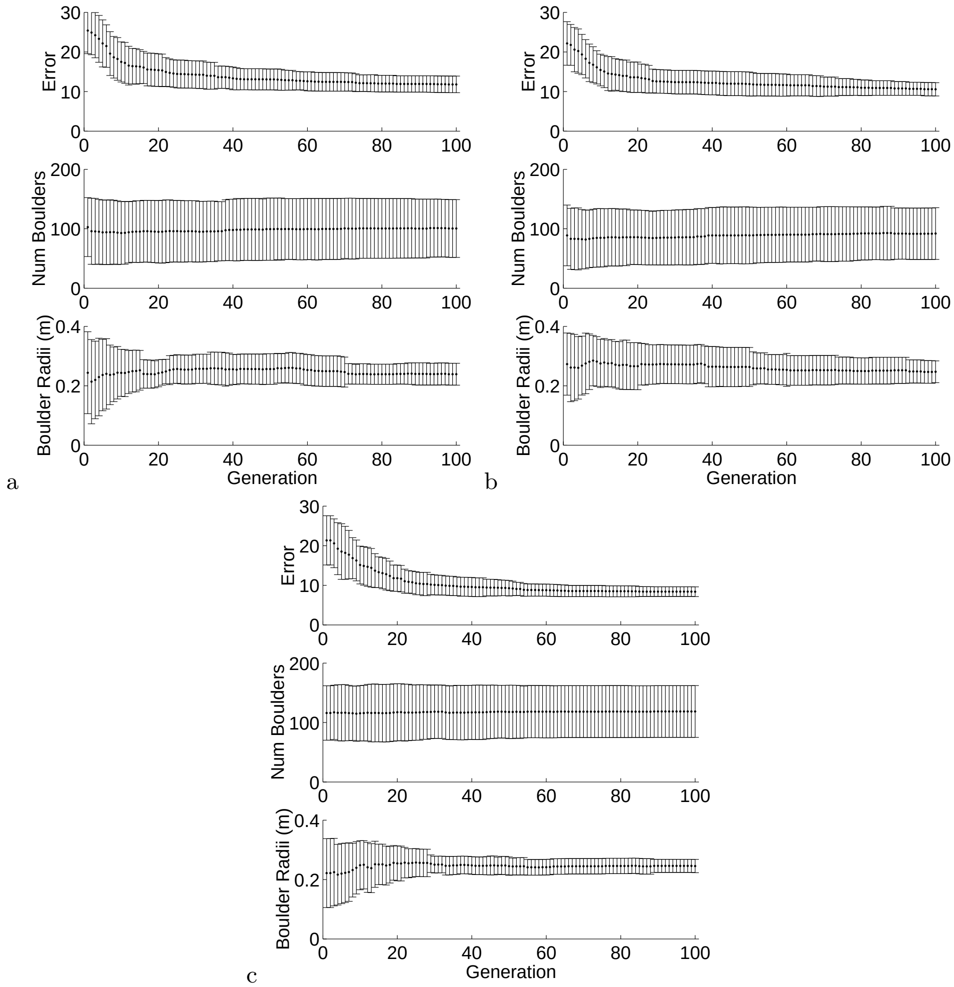

Fig. 4. Relative performance of the three experiments. a: The robot in the first experiment was allowed to traverse the target and model environments for 20,000 simulation time steps. **b:** Another thirty trials were conducted in which traversal time was extended to 40,000 time steps. **c:** A third set of thirty trials were conducted in which traversal time was extended to 60,000 time steps.

A further 30 trials were conducted in which the traversal time for both the target and model environments was extended to 40,000 time steps (Fig. 4b), and finally a third set of 30 trials were conducted in which the robot was allowed to traverse the environments for 60,000 time steps (Fig. 4c).

In order to determine whether the robot could use the synthesized models to make successful predictions about the result of future actions, two trials were randomly selected from the third experimental regime, and their best model environments were extracted. The robot was then commanded to traverse these two models at increasing speed, for 5000 time steps. Sixteen speeds were used: [5rev/sec, 6rev/sec, ... 20rev/sec]. For each speed, the robot traversed each of the two model environments 30 times. During each traversal, the simulation was stopped if the robot tipped over. If the simulation was repeatedly stopped prematurely for a given speed, this indicates that this is probably an unsafe speed at which to traverse not only the model environment, but also the target environment that it represents.

Fig. 5a reports the mean time to stopping for these two model environments. As can be seen, both models predict that the robot will begin to tip over at 9 rev/sec. The robot was then driven over the target environment at these same speeds, and the mean time until the robot tipped over was again calculated. For the target environment, the robot indeed begins to tip over at 9 rev/sec, indicating that the robot was successfully able to use the models to predict, in advance, the threshold at which behaviors become dangerous.

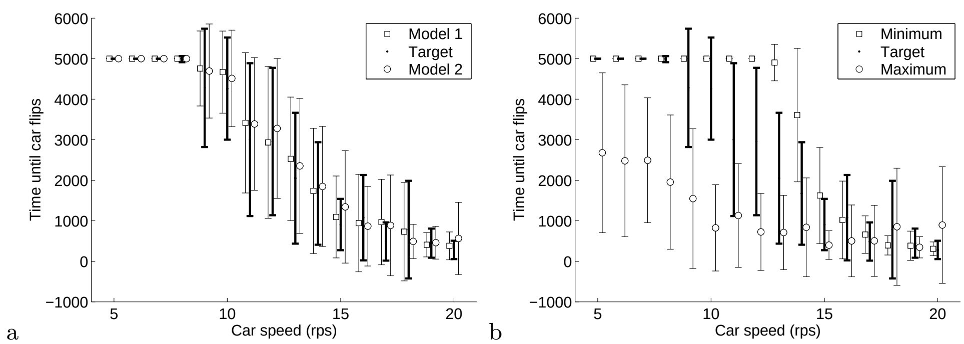

Fig. 5. Predictions of future action made by various models. a: Two of the optimized models predict that the robot will begin flipping over when it drives over the target environment at wheel speeds of 9 revolutions/second and higher, which matches that actual behavior of the robot in the target environment. **b:** Two other inaccurate models (at the minimum and maximum extremes of the allowed environment parameters) incorrectly predict that the robot will begin flipping over at 13rev/sec (the minimal model) or at speeds less than 5rev/sec (the maximal model).

In comparison, two other inaccurate models were used to predict the result of future action: a minimal model containing only one boulder with a radius of 5cm, and a maximal model containing 200 boulders with radii of 45cm. Fig. 5b reports the poor predictions generated by these models. Not surprisingly, the minimal model incorrectly predicts flipping over commences at higher speeds (13 rev/sec). The maximal model incorrectly predicts flipping over commences at lower speeds (5 rev/sec).

4 Discussion and Conclusions

In this work it has been shown that minimal sensor feedback (two tilt sensors) is required to indirectly infer and synthesize accurate, physically-realistic models. This finding supports the findings in [4], in which only two tilt sensors were used to indirectly recover the topology of a legged robot. Despite this minimal sensor requirement, sensor feedback must be processed correctly in order to induce a search gradient in the space of possible models (Fig. 3). The sensor processing introduced here in effect removes the time component from the sensor data. This reflects the observation that physical environments are extremely noisy and nonlinear, and it is well known that even accurate models will begin to diverge from nonlinear systems after a very short time interval [16][7]. In effect, the binning process described here compares the phase portraits of nonlinear signals, in which the dimensions of the portraits correspond to the number of sensors. Further investigation into the most appropriate method of signal comparison across a wide range of robots, sensors and environments is warranted.

This method of processing has the added benefit that the statistical behavior of the robot over time can be inferred by a human observer through visual inspection of the binned data (Fig. 2). Indeed visual inspection by human experts is a priority in this stream of research: processed sensor data should clarify robot behavior; three-dimensional physical simulations reveal robot [4] and environment structure; and automated compression of mathematical models improves both intelligibility and predictive ability [3].

Fig. 4 indicates that increasing the amount of traversal time on an environment allows a robot to better infer the environment's topology: the mean error of the trials drops from regime 1 (20,000 time steps) to regime 2 (40,000 time steps) to regime 3 (60,000 time steps). This can be explained by the observation that longer traversal times leads to more balanced sensor data: rare sensor data and biases from short traversals (such as only tipping to the left by encountering a series of boulders on the right) is damped out in longer traversals.

Similarly, the accuracy of the inferred size of the boulders improves in the long traversal experimental regime (Fig. 4c): the trials converge more rapidly and more tightly to the actual boulder size of 25cm. On the other hand, the number of boulders is rarely inferred correctly (Fig. 4). This is presumably due to the fact that certain random distributions of boulders (such as many boulders together) may fool the robot into believing there are more (or less) boulders than there actually are. Surprisingly however, even with this particular handicap, the robot is still able to correctly predict the outcome of future actions.

This is an encouraging result, as one of the main arguments against using models in robotics [5][1] is that a perfect model can never be created (or at least not in a finite amount of time). The results here indicate that a robot does not need to create a perfect model in order to correctly predict the result of future action. Rather, it merely needs to create a model that is appropriate for considering the outcomes of certain types of future action. For instance, because the robot does not model boulder distribution, it cannot plot a path through the boulder field. This result indicates that rapid synthesis of multiple, approximate models in which each is tuned for considering the results of a particular kind of future action may be more adaptable than a robot that attempts to create a single, perfect model.

References

1. R. Arkin. *Behavior-based Robotics*. MIT Press, 1998.
2. J. Bongard and H. Lipson. Nonlinear system identification using coevolution of models and tests. *IEEE Transactions on Evolutionary Computation*, 9(4):361–384, 2005.
3. J. Bongard and H. Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Science*, 2007. to appear.
4. J. Bongard, V. Zykov, and H. Lipson. Resilient machines through continuous self-modeling. *Science*, 314:1118–1121, 2006.
5. R. A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–160, 1991.
6. D. Cliff, P. Husbands, and I. Harvey. Evolving visually guided robots. In J.-A. Meyer, H. Roitblat, and S. Wilson, editors, *Proceedings of the Second International Conference on the Simulation of Adaptive Behaviour*, Boston, MA, 1993. MIT Press.
7. C. Danforth and J. Yorke. Making Forecasts for Chaotic Physical Processes. *Physical Review Letters*, 96(14):144102, 2006.
8. D. Floreano and F. Mondada. Hardware solutions for evolutionary robotics. In P. Husbands and J.-A. Meyer, editors, *EvoRobots*, pages 137–151, 1998.
9. A. Gloye, F. Wiesel, O. Tenchio, and M. Simon. Reinforcing the driving quality of soccer playing robots by anticipation. *IT - Information Technology*, 47(5):250–257, 2005.
10. N. Jakobi. Evolutionary robotics and the radical envelope of noise hypothesis. *Adaptive Behavior*, 6(1):131–174, 1997.
11. D. Keymeulen, M. Iwata, Y. Kuniyoshi, and T. Higuchi. Online evolution for a self-adapting robotics navigation system using evolvable hardware. *Artificial Life*, 4:359–393, 1998.
12. N. Nilsson. *Shakey the Robot*. SRI International, 1984.
13. S. Nolfi and D. Floreano. *Evolutionary Robotics*. MIT Press, Boston, MA, 2000.
14. J. B. Pollack, H. Lipson, S. Ficici, P. Funes, G. Hornby, and R. Watson. Evolutionary techniques in physical robotics. In J. Miller, editor, *Evolvable Systems: from biology to hardware*, pages 175–186. Springer-Verlag, 2000.
15. S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Upper Saddle River, NJ, 1995.
16. S. Strogatz. *Nonlinear Dynamics and Chaos: with applications to physics, biology, chemistry, and engineering*. Perseus Books, 1994.
17. R. Tedrake, T. Zhang, and H. Seung. Learning to walk in 20 minutes. In *Proceedings of the Fourteenth Yale Workshop on Adaptive and Learning Systems*, Yale University, New Haven, CT, 2005.
18. S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2005.