Chapter 9

Totems, Toys — Or Tools?

9.0 CHAPTER OVERVIEW

When Grey Walter reflected upon his robots, one question that he asked was whether they were totems, toys, or tools (Grey Walter, 1963). The robots that we have described in this book also require that this issue be explored. Some of these machines, such as Vehicle 2, the passive dynamic walker, the Strandbeest, and the Tortoise, are modern variants of historically important machines. All of our robots are constructed from toy parts. Are our machines more than just replicas of other devices, or more than mechanical toys? We answer these questions by making a point similar to that made by Grey Walter: our robots are more than totems, and more than toys—they are tools that can be used to explore many of the fundamental ideas of embodied cognitive science. Furthermore, they are tools that can be used to contribute new insight into current research issues. We demonstrate this with a final robot, called antiSLAM. This robot was initially designed to freely explore its environment, following walls and avoiding obstacles. However, we realized that it could generate some of the important regularities in a popular paradigm, called the reorientation task, used to study spatial cognition. Traditional theories of this task are strongly representational. AntiSLAM is of interest because it can produce many key reorientation task behaviours, but does so without any spatial representations or cognitive maps. This chapter provides some background on the reorientation task, and details the construction and programming of antiSLAM. It then describes antiSLAM’s behaviour in several different versions of the reorientation paradigm. The implications of these results clearly indicate that our LEGO robots provide both “hard fun” and “hard science.”
9.1 ARE OUR ROBOTS MORE THAN TOTEMS?

9.1.1 Uncanny Machines

The robot Tortoises provided “mimicry of life” (Grey Walter, 1963, p. 114). Grey Walter’s worry was that they were merely totems, and that his investigations of them were without meaning. “We are daily reminded how readily living and even divine properties are projected into inanimate things by hopeful but bewildered men and women; and the scientist cannot escape the suspicion that his projections may be psychologically the substitutes and manifestations of his own hope and bewilderment” (p. 115). Such a concern arises whenever one simulates the real (Baudrillard, 1994). While a symbol or simulation begins as a reflection of reality, it evolves into having “no relation to any reality whatsoever: it is its own pure simulacrum” (Baudrillard, 1994, p. 6). Grey Walter’s concern was that Tortoise behaviour might be meaningless because it would not refer to the behaviour of the real.

This issue is rooted in the seventeenth-century comparison of man and machine (Grenville, 2001; Wood, 2002). The view that man was a machine governed by universal, mechanistic principles was a central tenet of Cartesian philosophy (Descartes, 1637/1960). Eighteenth-century applications of this philosophy appeared in the form of elaborate, life-mimicking, clockwork automata, such as the androids constructed by Pierre and Henri-Louis Jaquet-Droz that wrote, sketched, or played the harpsichord (Wood, 2002). More famous automata of Jacques de Vaucanson, —including a flute player and a food-digesting duck— were in circulation for a century.

In their day, clockwork automata raised serious tensions between science and religion. In 1727, androids of Vaucanson’s that served dinner and cleared tables were deemed profane, and his workshop was ordered destroyed (Wood, 2002). Pierre Jaquet-Droz was imprisoned by the Spanish Inquisition — along with his writing automaton! Such tensions were salved by Descartes’ dualism: animals were machines; men were machines that also had souls, machines that also thought (see also Chapter 1).

Modern machines that mimic life still raise serious questions about what it is to be human. The emotions that they can provoke have long been exploited in literature and film. Freud examined how the feeling of the uncanny was used as a literary device (Freud, 1919/1976). He noted that the uncanny requires that the familiar be presented in unfamiliar form. The source of the uncanny that Freud identified mirrors Baudrillard’s analysis of symbols. “The cyborg is uncanny not because it is unfamiliar or alien, but rather because it is all too familiar. It is the body doubled —doubled by the machine that is so common, so familiar, so
ubiquitous, and so essential that it threatens to consume us, to destroy our links to nature and history” (Grenville, 2001, pp. 20–21).

Modern robotics shifts the uncanny from fiction to fact, as did eighteenth-century automata. Current humanoid robots produce a phenomenon called the uncanny valley (MacDorman & Ishiguro, 2006; Mori, 1970): our acceptance of androids suddenly plummets when their appearance grows to be very lifelike, but can still be differentiated from biological humans. The uncanny valley concerns “what troubles us when we are faced with certain versions of ourselves — bionic men, speaking robots, intelligent machines, or even just a doll that moves” (Wood, 2002, p. xxvii). Wood notes that all automata are presumptions “that life can be simulated by art or science or magic. And embodied in each invention is a riddle, a fundamental challenge to our perception of what makes us human.” Usually such troubling riddles are addressed by discovering what differentiates us from machines.

This is the opposite of the problem that faced Grey Walter. For his robots to be accepted scientifically, they must at some level be equivalent to living organisms. Only if this were true could their mimicry advance the understanding of living beings as Grey Walter intended. Why should we believe that his robots — and the LEGO devices that have been described in preceding chapters — are not merely totems? Why should we believe that their mimicry can tell us something new about adaptive, biological agents?

9.2 ARE OUR ROBOTS MORE THAN TOYS?

9.2.1 The Tortoise as Toy

When one sees images of historical automata (Grenville, 2001; Standage, 2002; Wood, 2002), it is hard to imagine that they led their audience to the uncanny valley. While they often took humanoid form, it would be impossible to confuse these machines with living organisms on the basis of appearance. They were marvelous, not for their resemblance to life, but rather because their intricate actions seemed beyond the ability of ordinary machines. As living dolls, eighteenth-century automata defined “the golden age of the philosophical toy” (Wood, 2002, p. 17).

Toys are central to other devices that we have discussed in this book. For example, the passive dynamic walkers that were introduced in Chapter 5 were inspired by walking toys (McGeer, 1990a; Wisse & Linde, 2007), one of which was patented in 1888 by Faris, the other — the “Wilson Walkie” — in 1938 by Wilson.

Grey Walter argued that the scientific merit of his Tortoises required that they be more than totems. However, their intended usefulness also
required that they be more than toys. While he made this argument too (Grey Walter, 1963), his robots’ appearances were such that it was hard not to consider them to be children’s toys.

Grey Walter’s efforts to publicize his robotic research (Holland, 2003b) certainly did little to dispel this notion. Newspaper accounts of his work used the researcher’s own words to minimize its scientific intent: “Toys which feed themselves, sleep, think, walk, and do tricks like a domestic animal may go into Tommy’s Christmas stocking in 1950, said brain specialist Dr. Grey Walter in Bristol last night” (Holland, 2003b, p. 352). The Grey Walter Online (http://www.ias.uwe.ac.uk/Robots/gwonline/gwonline.html) website reports that one of Grey Walter’s robots was sent to an American company in 1953 with the express purpose of creating a new kind of toy. The 1972 documentary Future Shock, narrated by Orson Welles, includes footage of Grey Walter saying of one of his Tortoises, “This looks rather as though it was a child’s toy, and I suppose it might be.” While he goes on to say, “but in fact it is rather a serious model of my ideas about behavior,” the damage is done.

9.2.2 LEGO Is a Toy!

When the British Association of Toy Retailers chose its “Toy of the Century,” what lucky toy beat out the stiff competition provided by the teddy bear, Action man, and Barbie? It was LEGO, of course. One component of the LEGO world, the NXT Mindstorms robotics system, has itself been deemed an award-winning toy, receiving the Canadian Toy Testing Council Best Bet 2007 and the Oppenheim Toy Portfolio Platinum Award 2007.

To distinguish Tortoises from toys, Grey Walter could at least note that his earliest devices were constructed from war surplus parts. We are in a less enviable position with our robots in this book, because they are literally constructed from children’s playthings. The seeds that became LEGO Mindstorms were first sown as part of educational research at MIT in the 1960s (Martin, Mikhak, Resnick, Silverman, & Berg, 2000), and the programmable brick that emerged from this work is an educational toy (Resnick, Martin, Sargent, & Silverman, 1996). LEGO robots were designed for children. “Designing tools that allow children to add computation to traditional construction—and recognizing the learning opportunities afforded by this activity—has been the focus of our work over the last number of years” (Martin et al., 2000, p.10).

Martin et al. (2000, p.10) ask, “when does something stop being a machine and start being a creature?” It is exactly this question that Grey Walter pondered when he argued that the Tortoises were more than...
totems; perhaps this question is even more telling when the machine in question is literally a children’s toy, such as the LEGO robots that we have been investigating. What arguments did Grey Walter provide that permit us to separate the Tortoises — and our LEGO creations — from totems and toys?

9.3 FROM TOTEMS AND TOYS TO TOOLS

9.3.1 Tortoise as Tool

Why did Grey Walter (1963) believe that his machines were more than totems? He argued that totems became potent symbols because they resembled that which they represented. “Until the scientific era, what seemed most alive to people was what most looked like a living being. The vitality accorded to an object was a function primarily of its form” (Grey Walter, 1963, p. 115). In contrast, his robots were not concerned with reproducing appearances, but instead with imitating behaviour and performance.

What classes of behaviour would be the target of scientific imitation? Grey Walter (1963, p. 120) provided an intimidating list: “exploration, curiosity, free-will in the sense of unpredictability, goal-seeking, self-regulation, avoidance of dilemmas, foresight, memory, learning, forgetting, association of ideas, form recognition, and the elements of social accommodation. Such is life.”

Grey Walter’s (1963) Tortoises demonstrated many of these characteristics (see also Chapters 6 and 7). They were more than toys, because (in his view) toys could not produce any of these behaviours. “The technical genius of the Swiss watchmakers was really wasted on their delicate clockwork automata; they arouse only a passing interest because they are neither sacred nor, like life, unpredictable, their performance being limited to a planned series of motions, be it a boy actually writing a letter or a girl playing a real keyboard instrument” (p. 115).

By situating his embodied machines, Grey Walter (1963) moved them beyond totems and toys. They produced behaviours that were creative and unpredictable because they were governed by the relationships between their internal mechanisms and the surrounding, dynamic world. “The important feature of the effect is the establishment of a feedback loop in which the environment is a component” (p. 130).

9.3.2 Pedagogical and Scientific Tools

We believe that Grey Walter’s arguments that his machines were tools apply equally well to our LEGO creations because of their varying degrees of situatedness and embodiment. Their ease of construction, and
the popularity and wide availability of their components, mean that they can also serve admirably as another kind of tool: a pedagogical tool that permits students to explore some of the key ideas emerging in embodied cognitive science.

The issues raised by embodied cognitive science are not easily dealt with by more traditional approaches (Clark, 1997, 1999, 2003, 2008). As a result, Clark (1997, p. 103) asks, “What kind of tools are required to make sense of real-time, embodied, embedded cognition?”

Programmable bricks are devices that serve as “things to think with.” (Resnick et al., 1996, p. 450). Mindstorms bricks present computation in a new light—not as the programming of a disembodied, stationary desktop computer, but as the development of sense–act relations in creatures that move freely in the world (Resnick et al., 1996). LEGO robots, then, appear to have been designed to provide a rich medium for exploring embodied cognitive science. Students can use them to “realize that sophisticated behaviours can emerge from interactions of rules with a complex world, but at the same time, are still captivated by the wonder of a machine acting like a pet” (Martin et al., 2000, p. 10).

However, can LEGO robots be more than pedagogical tools? Grey Walter (1963, p. 132) noted of his machines that “as tools they are trustworthy instruments of exploration and frequent unexpected enlightenment.” When used to illustrate historically important robots, LEGO robots are instruments of exploration. But retracing the steps of others limits their possibility for enlightenment. We need to use our LEGO devices to explore new ideas in embodied cognitive science. The remainder of this chapter illustrates this possibility, by showing how LEGO robots can contribute to current debates arising in the study of human and animal navigation.

9.4 ANIMAL NAVIGATION AND REPRESENTATION

9.4.1 Navigational Organisms

“Navigation is the process of determining and maintaining a course or trajectory from one place to another” (Gallistel, 1990, p. 35). There are many examples of extraordinary navigational feats. One is the small blackpoll warbler’s 1,575-km nonstop flight over the western Atlantic as it migrates from its New England staging grounds to South America or the West Indies (Baird, 1999; Drury & Keith, 1962). Another example is provided by the long-distance navigators of Micronesia who use seagoing canoes to routinely complete voyages of 150 miles or more without sight of land, and without the use of charts or instruments (Finney, 1976; Gladwin, 1970; Hutchins, 1995). However, many more mundane
tasks depend on navigation, and are critical to the survival of most organisms. As a result, there has been an intensive study of navigation that has revealed a tremendous amount of information about the spatial information that animals exploit to move about in their environment.

For instance, consider one basic element of navigation, determining a heading, which is a direction of movement relative to some external coordinate system (Gallistel, 1990). There is an abundance of evidence that many animals are sensitive to direction. The tail-wagging dance of bees communicates the location of food sources using directional information (Frisch, 1966, 1967). In particular, the dance specifies the angle between the food source and the azimuth of the sun, with the beehive at the origin of the angle. Birds like Clark’s nutcrackers and pigeons can localize food caches by encoding directional relationships amongst multiple landmarks (Jones & Kamil, 2001; Kamil & Cheng, 2001; Kamil & Jones, 1997, 2000; Spetch, Rust, Kamil, & Jones, 2003). The rat’s hippocampus contains head direction neurons that respond strongly when the rat’s head points in the cell’s preferred direction (Redish, 1999; Sharp, Blair, & Cho, 2001; Taube & Muller, 1998).

9.4.2 Sense–Think–Navigate
At the heart of most research on animal navigation is the notion that it is representational, in the strong sense proposed by classical cognitive science (see Chapters 2 and 3). That is, navigation is typically viewed as a sense–think–act process, where the thinking involves the use of various representations of space. As a result, it is not surprising that one particular interest of scientists is determining the properties of spatial representations in animals and humans (Healy, 1998).

For instance, many theories of bird navigation—concerning both local homing and long-distance migration—assume a map-and-compass model, in which birds use some form of navigational map to determine their current location relative to a goal, and then use a celestial or magnetic compass to set their heading toward that goal (Mouritsen, 2001; Wiltschko & Wiltschko, 2003). Similarly, there is a great deal of evidence that a cognitive map of the world (Tolman, 1948), specifying both locations and directions, is encoded in the rat hippocampus (O’Keefe & Nadel, 1978), although there is considerable debate about its specific properties (Burgess, Jeffery, & O’Keefe, 1999; Dawson et al., 2000; McNaughton et al., 1996; Redish, 1999; Redish & Touretzky, 1999; Touretzky & Redish, 1995; Touretzky, Wan, & Redish, 1994). Micronesian navigators do so by superimposing a number of mental images that are anchored by the rising and setting points of various stars (Hutchins,
1995). Gallistel (1990, p. 121) notes that “orienting towards points in the environment by virtue of the position the point occupies in the larger environmental framework is the rule rather than the exception and, thus, cognitive maps are ubiquitous.”

The dominance of sense–think–act theories of navigation has two implications for the current chapter. First, biologically inspired robotic models of navigation are frequently representational, as the next section shows. Second, given the themes developed in this book, we can wonder whether sense–act theories of navigation are possible, and whether such theories could be explored using synthetic methodologies to construct different kinds of robots.

**9.5 REPRESENTATION AND ROBOT NAVIGATION**

**9.5.1 Animals to Animats**

The intensive study of human and animal navigation has led to theories that are predominantly representational. For the most part, these theories are tested and refined using traditional methodologies from experimental psychology and neuroscience. However, in a growing number of cases these theories are also tested by using them to develop autonomous, navigating robots. These biologically inspired robots can be thought of as artificial creatures, often called animats, that have been used to test the strengths and weaknesses of various representational theories of navigation.

For example, the existence of place and head-direction cells in the rat’s hippocampus has inspired a number of different navigational robots (Arleo & Gerstner, 2000; Burgess, Donnett, Jeffery, & O’Keefe, 1997; Milford, 2008). The Burgess et al. (1997) robot employs a control system that includes “sensory cells” that encode a robot’s distance from walls via infrared sensing, and “place cells” that are used to encode the robot’s location when it is reinforced. Simple learning routines are used to modify connection weights between the various components of the control system. Tests of the robot demonstrated that it could use this modelled cognitive map to localize its position and direction when it moved around a rectangular environment. It remembered locations where it was reinforced at a particular location, and could return to them even when started from different locations.

Arleo and Gerstner (2000) have developed an animat that is similar to the Burgess et al. (1997) machine, but includes senses of self-motion. The robot associates locations in its “hippocampal map” with rewards, and will navigate to them. If it is not rewarded after it arrives at an intended location, the unrewarded location will be forgotten from its map.
9.5.2 SLAM and AntiSLAM

Because most theories of animal navigation are representational, it is not surprising that when they are transferred to animats, as in the preceding examples, the resulting machines are sense–think–act devices. However, robot navigation is often construed as necessarily being representational. “Low level robots may function quite adequately in their environment using simple reactive behaviours and random exploration, but more advanced capabilities require some type of mapping and navigation system” (Milford, 2008, p. 10).

Because of this assumption, one of the central problems being explored by roboticists is simultaneous localization and mapping (Jefferies & Yeap, 2008), or SLAM. Assume that robots find their place in the world by relating their current sensed location to some place on an internal map. However, if they are placed in a novel environment then no such map exists, and self-localization is impossible. Methods must be developed for the agent to simultaneously build a new map of the novel environment and locate itself using this map. This is a difficult problem, and robotics researchers are turning to studies of biological navigation to help solve it (Jefferies & Yeap, 2008). For example, Milford (2008) suggests that simultaneous localization and mapping can be accomplished by using a hippocampus-inspired model that uses both place cells and head-direction cells.

The SLAM problem is predicated upon the assumption that navigation involves representation. Some researchers who study animal navigation have begun to question aspects of this assumption (Alerstam, 2006). To what extent might a completely reactive, sense–act robot be capable of demonstrating interesting navigational behaviour? We now turn to exploring this question. First we will introduce a simple task that has been used to study navigation in local environments, and has inspired representational theories. Second, we will explore some recent concerns about such theories. Third, we will investigate non-representational theories of this task, which include some synthetic studies that use our own reactive LEGO robot, which we—for obvious reasons—call antiSLAM.

9.6 SPATIAL BEHAVIOUR AND THE REORIENTATION TASK

9.6.1 Navigational Cues

How do organisms find their place in the world? One approach to answering this question is to set up small, manageable indoor environments. These environments can be customized to provide a variety of different cues to animals that learn to navigate within them. For
instance, like some of the robots described in the preceding section, an animal might be reinforced for visiting a particular location. What information does an animal use to return to this location in the hope of receiving more rewards?

Studies of navigation in indoor environments have found that humans and animals exploit various geometric cues as well as feature cues (see Cheng & Newcombe [2005] for a recent review). Geometric cues are relational, while feature cues are not: “A geometric property of a surface, line, or point is a property it possesses by virtue of its position relative to other surfaces, lines, and points within the same space. A non-geometric property is any property that cannot be described by relative position alone” (Gallistel, 1990, p. 212). One question of considerable interest is the relative contributions of these different cues for navigation.

9.6.2 The Reorientation Task

One approach to answering this question is the reorientation task. In this paradigm, an agent is placed within an “arena” that is usually rectangular. Metric properties (wall lengths, angles between walls) combined with an agent’s distinction between left and right (e.g., the long wall is to the left of the short wall) provide geometric cues.

Other arena properties can provide feature cues. For example, Figure 9-1 illustrates an arena that has one blue wall, while all the other walls are black; the distinctive colour is a feature cue. Or, one could place unique objects at different locations in the arena. This is shown in Figure 9-2, where each letter in the figure stands for a unique object (e.g., a coloured panel) that distinguishes each corner from the others. These objects also provide feature cues.
In the reorientation task, an agent learns that a particular place—usually a corner of a rectangular arena—is a goal location. Imagine that when placed in either arena illustrated in Figure 9-1 or Figure 9-2, the agent is rewarded when it visits the corner labelled 4, but is not rewarded when it visits any other corner. The agent learns that corner 4 is the goal location.

The agent is then removed from the arena, disoriented, and returned to an arena, with the task of using the available cues to relocate the goal. Of particular interest are experimental conditions in which the arena has been altered from the one in which the agent was originally trained.

For example, in the new arena the feature cues might have been moved to different locations than was the case when the subject originally learned the goal location. This places feature cues in conflict with geometric cues. Will the agent move to a location defined by geometric information, or will it move to a different location indicated by feature information? Extensive use of the reorientation task has revealed some striking regularities.

**9.7 BASIC FINDINGS WITH THE REORIENTATION TASK**

*9.7.1 Rotational Error*

First, consider the case in which agents are trained that corner 4 is a goal location using an arena like Figure 9-1 or 9-2. Then, the agent must reorient itself in a new arena that only provides geometric cues. Such an arena has no local features that can be used to distinguish one location from another, as illustrated in Figure 9-3.

Geometric cues do not uniquely specify a target location in such an arena. For instance, the geometric cues available at Location 4 of Figure
9.3 are identical to those available at Location 2 of the same figure: 90° angle, longer wall to the left and shorter wall to the right. However, these geometric cues can distinguish Location 4 from either Location 1 or Location 3.

Under such conditions, one of the basic findings is rotational error (Cheng, 1986, 2005). When rotational error occurs, the trained animal goes to the goal location (e.g., Location 4 in Figure 9-3), as well as the corner that is geometrically identical to it (Location 2), which is located at a 180° rotation through the centre of the arena, at above chance levels. Rotational error is usually taken as evidence that the agent is relying upon the geometric properties of the environment.

9.7.2 Mandatory Geometry

The second main regularity that governs the reorientation task occurs when feature cues, such as the distinct objects illustrated in Figure 9-2, are available during training. Such feature cues uniquely identify a goal location — that is, it is possible for an agent to learn where the goal location is by only using these cues, and by ignoring geometric cues. However, the evidence suggests that agents still learn about the geometric properties during this training, even though these cues are irrelevant or unnecessary in this version of the task. That is, geometric cues still influence behaviour even when such cues are not required to solve the task. It as if the processing of geometry is mandatory.

This regularity is supported by several pieces of evidence. First, in some cases subjects continue to make some rotational errors even when a feature disambiguates the correct corner (Cheng, 1986; Hermer & Spelke, 1994).
Second, when features are removed following training, subjects typically revert to choosing both of the geometrically correct locations (Kelly, Spetch, & Heth, 1998; Sovrano, Bisazza, & Vallortigara, 2003).

Third, consider the case when features are moved after training—for instance, after being trained in the arena illustrated in Figure 9-2, the animal must reorient in the arena illustrated in Figure 9-4, where all of the objects have been moved in a clockwise direction. This produces a conflict between geometric and feature cues; control by both types of cues is often observed in such conditions (e.g., Brown, Spetch, & Hurd, 2007; Kelly, Spetch, & Heth, 1998). That is, there will be an increased tendency to visit Corner 1 than was the case during training, because it is now marked by the correct feature. However, Corner 4 will still be visited (because it still has the correct geometric cues), as will the geometrically equivalent Corner 2.

9.8 REPRESENTATIONAL THEORIES OF REORIENTATION

9.8.1 The Geometric Module

The reorientation task has inspired a number of different theories related to reorientation and navigation. For instance, Gallistel (1990) viewed the solution of the reorientation task as a two-stage process. The first stage occurs when an agent is first placed in an arena: it encodes the shape of the arena by attending to metric cues, such as wall lengths and angles between walls, as well as to sense cues (i.e., the distinction between left and right). The purpose of encoding the arena’s shape is that this information is then used by the agent to determine its heading: that is, the arena’s shape provides the reference frame for the agent’s ability to orient itself.
The second stage occurs when an agent is disoriented, and then placed in an arena once again. In this stage, the agent uses a representation of the shape of the previously encountered arena as a mental map. The agent “gets its heading and position on its map by finding the rotation and translation required to produce a congruence (shape match) between the currently perceived shape of the environment and a corresponding region of its map” (Gallistel, 1990, p. 220). If the only sources of information used to create such maps are sense and geometric cues, one consequence of this theory is rotational error in rectangular arenas.

A key assumption of the Gallistel (1990) model is that the processing of environmental shape is modular (Fodor, 1983). According to Fodor, a module is a neural substrate that is specialized for solving a particular information-processing problem. Modules operate in a fast, mandatory fashion; they exhibit characteristic breakdown patterns when they fail because of their specialized neural circuitry; and they operate independently of the influence of the contents of higher-order beliefs—that is, they are cognitively impenetrable (Pylyshyn, 1984). It has been argued (Cheng, 1986; Gallistel, 1990) that the geometric computations in Gallistel’s model are modular because they are mandatory and because they are not influenced by “information about surfaces other than their relative positions” (Gallistel, 1990, p. 208).

Why would there be a module for processing geometric cues? Gallistel (1990) proposes two reasons. First, reorientation can be accomplished by using a fairly simple algorithm for bringing the shape of the new arena, and the shape of the remembered arena, into register. Such an algorithm is not subject to combinatorial explosion when the shape of the arena changes (e.g., in size or complexity). These computational advantages are substantial, and therefore it may be important to ‘reify’ them as modular properties.

Second, from an evolutionary point of view, geometric modularity might take advantage of the fact that overall shape of an animal’s typical environment is not likely to change dramatically, even though many visual features within the environment might change from day to day. “In relying on overall shape alone, the nervous system finesses the problem of finding the optimal weights for mediating the trade-offs between changes occurring along incommensurable sensory dimensions” (Gallistel, 1990, p. 212). This is an example of modularity being used to solve the frame problem that can be frequently encountered by representational systems (Dawson, 1998; Pylyshyn, 1987).
9.8.2 Geometry and Representation

The geometric module is an influential theory designed to account for the regularities in the reorientation task. For the purpose of the current chapter, it is important to stress that it is a representational theory: “Rats have a representation of the shape of the environment that includes the uniquely metric relations and sense” (Gallistel, 1990, p. 219). Recently, though, questions have been raised about the nature—and even possible existence—of the geometric module. We shall see that such questions can inspire research that explores possible non-representational accounts of the reorientation task.

9.9 Whither the Geometric Module?

9.9.1 Modifying Modularity

Recently, some researchers have begun to question the geometric module. One reason for this is that the most compelling evidence for claims of modularity comes from neuroscience (Dawson, 1998; Fodor, 1983), but such evidence about the modularity of geometry in the reorientation task is admittedly sparse (Cheng & Newcombe, 2005). As a result, most arguments about modularity in this context are based on behavioural data. However, the data obtained from the reorientation task is consistent with many different notions of modularity (e.g., Cheng & Newcombe, 2005, Figure 3).

For this reason, some researchers have proposed alternative notions of modularity when explaining reorientation task regularities (e.g., Cheng, 2005; Cheng & Newcombe, 2005). Cheng (2005, p. 17), suggests that “geometric and feature information are encoded together in one record for localization. This process is non-modular.” Cheng then attempts to preserve modularity by arguing that different types of information might be stored in the same location, but when certain devices access this common store, they only access particular types of information, and are thus modular in nature. In short, Cheng conjoins “a modular process and a non-modular representational structure.” This approach is very similar to that exemplified in the production system architectures discussed in Chapter 3 (Anderson, 1983; Anderson et al., 2004; Newell, 1973, 1990) if one views working memory as a “non-modular representation structure” and individual productions as special purpose “devices.”

9.9.2 Non-modular Reorientation

While some theories (e.g., Cheng, 2005) are attempts to preserve geometric modularity by redefining it, others reflect more radical approaches
(Cheng, 2008b): several new theories of the reorientation task completely reject the existence of a geometric module.

For instance, one model of the reorientation task uses a general theory of associative learning in which geometric and feature cues are not treated differentially (Miller & Shettleworth, 2007, 2008). Organisms learn what cues are present at a reinforced location, and then are more likely to approach locations with similar cues at a later time. This model does not require the reorientation task to be solved by applying geometric operations to global representations of arena shape. Similarly, simple neural networks can generate reorientation task regularities by modifying associations involving locally available cues only — such as the length and colour of a particular wall — while at the same time having absolutely no representation of global arena shape (Dawson, Kelly, Spetch, & Dupuis, 2008).

Another theory assumes that agents reorient by maximizing the visual similarity (e.g., unprocessed pixilated images of locations) of locations in the new arena to the image of the goal location in the original arena (Cheung, Stuerzl, Zeil, & Cheng, 2008; Stuerzl, Cheung, Cheng, & Zeil, 2008). In this theory, the metric of visual similarity does not make explicit the geometric properties (i.e., arena shape) that were central to earlier theories of the task.

While these newer theories reject the geometric module, they still share Gallistel’s (1990) assumption that the reorientation task is solved by representational mechanisms. The Cheung et al. (2008) theory is obviously representational, because it involves matching visual input to remembered visual information. Associative models like Miller and Shettleworth’s (2007, 2008) or Dawson et al.’s (2008) are also representational because associative strengths or neural network connection weights are representations of previous experience (Dawson, 2004).

However, if representations of locally visible cues are sufficient to deal with the reorientation task, then it is a small step to ask whether non-representational, sense–act processes are also sufficient. Some recent robotic studies of the reorientation task have explored this very question (Nolfi, 2002; Nolfi & Floreano, 2000). Let us now consider some examples of this reactive research.

9.10 REACTIVE ROBOTS AND THEIR EVOLUTION

9.10.1 New Wave Robotics

The pioneering autonomous, self-navigating robots were sense–think–act devices, planning their future movements using some sort of internal map (Nilsson, 1984). As we have seen, this tradition persists in much of
modern research on robot navigation (Filliat & Meyer, 2003; Milford, 2008; Trullier, Wiener, Berthoz, & Meyer, 1997). However, there is an alternative trend in robotics, a movement that has been called “new wave” (Sharkey, 1997). New wave robotics strives to replace representation with reaction (Brooks, 1999); robots are created with direct links between sensors and actuators, so that they are better described as sense–act systems than as sense–think–act devices. All the LEGO NXT robots that we have described earlier in this book are simple examples of new wave, or reactive, robots.

While reactive roboticists do not necessarily deny the existence or importance of representations, they recognize that “embodied and situated systems can solve rather complicated tasks without requiring internal states or internal representations” (Nolfi & Floreano, 2000, p. 93). Of particular interest to us is the use of reactive robots to investigate behaviour in the reorientation task (Lund & Miglino, 1998). Lund and Miglino conducted a study in which their robots were evolved to accomplish the reorientation task, and to mimic animal behaviour when governed by the geometric cues of a rectangular arena. Before discussing their robot, let us briefly discuss the evolutionary approach that they adopted.

9.10.2 Evolving Robots

While we have been exploring LEGO Mindstorms robots, robotics researchers have conducted their work using a variety of different platforms. One of these is the Khepera miniature robot (Mondada & Floreano, 1995; Mondada, Franzi, & Ienne, 1994). This small robot has two motor-driven wheels, and 8 infrared proximity detectors arranged about its puck-shaped chassis. When used as a reactive robot, the speeds of its two motors are determined by the signals from the various proximity detectors. To set speed, the signals are weighted; the roboticist’s task is to find a set of signal weights that cause the robot to perform some task of interest.

Evolutionary robotics is one approach to discovering an appropriate set of signal weights to control robot behaviour (Nolfi & Floreano, 2000). Evolutionary robotics is inspired by a more general form of evolutionary computation, genetic algorithms (Holland, 1992; Mitchell, 1996).

In general, the evolution of a Khepera robot design begins by defining a fitness function that measures how well a robot is performing a task of interest (Nolfi & Floreano, 2000). An initial population of different control systems (e.g., different sets of sensor-to-motor weights) is then produced. Each of these control systems is evaluated using the fitness function, and the control systems that produce higher fitness
values are maintained. They are also used as templates for new control systems in the next generation of control systems, which are created by “mutating” the surviving control systems in prescribed ways. The whole process is then repeated with the next generation of controllers. It is expected that average fitness will improve with each new generation. The evolutionary process can be stopped when, for instance, fitness stops changing appreciably over a series of generations.

Of course, evolutionary methods are not restricted to the Khepera robot, although this is the platform used by Lund and Miglino (1998). Robot evolution has played an important role in the development of other robots that have been discussed in earlier chapters. Jansen used genetic algorithms to discover his holy numbers that were introduced in Chapter 5 (Jansen, 2007), and later evolved his Strandbeest by racing different walking robots on the beach, disassembling the losers, and using their parts to make (possibly mutated) copies of the winners. Evolutionary techniques were also central to Braitenberg’s proposal for the development of his more advanced vehicles (Braitenberg, 1984).

9.11 REACTIVE ROBOTS AND ROTATIONAL ERRORS

9.11.1 Reactive Reorientation

Lund and Miglino (1998) evolved controllers for Khepera robots to perform the reorientation task in a rectangular arena without feature cues. A fitness function for this task is easily created; a fitter robot will find itself closer to the goal than will a less fit robot. The robots were started from eight different locations in the arena. A satisfactory controller for the robot was achieved after 30 generations of evolution, navigating the machine (from any of the starting locations) to the goal on 41% of trials. This controller also produced rotational error— it navigated the robot to the corner 180° from the goal on another 41% of the test trials. This data is very similar to results obtained from rats (e.g., Gallistel, 1990).

Lund and Miglino (1998) noted it was impossible for their Khepera robots to represent arena shape. “The geometrical properties of the environment can be assimilated in the sensory-motor schema of the robot behavior without any explicit representation. In general, our work, in contrast with traditional cognitive models, shows how environmental knowledge can be reached without any form of direct representation” (p. 198).

How does a completely reactive robot achieve this performance? When the robot is far enough from the arena walls that none of the sensors are detecting an obstacle (about 4 cm), the controller weights are such that it moves in a gentle curve to the left. As a result, when
the robot leaves from any of the eight starting locations, it never encounters a short wall! When a long wall is encountered, the robot turns left, and follows the wall until it stops in a corner. A robot that always turns left, and never encounters a short arena wall, must necessarily produce rotational errors!

9.11.2 Representative Reaction

One problem with Lund and Miglino’s (1998) robot is that it, like the thoughtless walker discussed in Chapter 5, depends critically upon its environment. If it is placed in an arena that is even slightly different in size, its emulation of rat reorientation behaviour diminishes (Nolfi, 2002). The robot would be more representative of animals if it were not limited to a particular arena.

Nolfi (2002) addressed this problem by evolving robots in arenas of varying sizes, and using a much wider variety of starting positions and orientations. Nolfi required 500 generations of robot evolution to achieve satisfactory performance in this more difficult task. However, at this point his robots produced rotational errors, and could do so in different-sized arenas.

How did Nolfi’s (2002) robot deliver such performance in the absence of a cognitive map? First, the robot tends to move forward, avoiding walls, eventually encountering a corner. In this situation, signals from the walls cause it to adjust itself until it is at a 45° angle from one wall, and then it will turn either clockwise or counterclockwise (depending upon whether the sensed wall is to the robot’s left or the right).

The final turn away from the corner means that the robot will be pointing in such a way that it will follow a long wall. This is because sensing a wall at 45° provides an indirect measurement of whether the wall is short or long! “If the robot finds a wall at about 45° on its left side and it previously left a corner, it means that the actual wall is one of the two longer walls. Conversely, if it encounters a wall at 45° on its right side, the actual wall is necessarily one of the two shorter walls. What is interesting is that the robot ‘measures’ the relative length of the walls through action (i.e., by exploiting sensory–motor coordination) and it does not need any internal state to do so” (p. 141). In other words, the sensory states of the robot permit it to indirectly measure the relative lengths of walls without directly comparing or representing length. It will use this sensed information to follow the long wall, which will necessarily lead the robot to either the goal corner or the corner that results in a rotational error, regardless of the actual dimensions of the rectangular arena.
9.12 REORIENTING LEGO ROBOTS

9.12.1 Motivating AntiSLAM

We have seen that there is tremendous interest in studying navigation, and that one paradigm used to conduct this study is the reorientation task. We have also seen that much of the general study of navigation is consistent with sense–think–act models of cognition (Healy, 1998; Milford, 2008); this is also true of the reorientation task (Gallistel, 1990; Miller & Shettleworth, 2007, 2008). However, more recent work shows that reorientation task regularities can be produced by reactive robots that are incapable of building and using spatial representations (Lund & Miglino, 1998; Nolfi, 2002).

One theme being explored in this chapter is the use of LEGO robots as scientific tools. To illustrate this possibility, we now describe a LEGO machine developed to navigate through an environment, with the hope that it too can provide insight into the reorientation task. One of the important problems faced by roboticists who develop autonomous, navigating robots is SLAM: simultaneous localization and mapping. The robot that we are about to describe is a very simple, reactive device that is not capable of creating or exploiting internal representations of the world. For this reason, we call it antiSLAM.

The antiSLAM robot began its development as a machine designed to follow walls, explore its environment, and avoid obstacles (see Section 9.31). However, we discovered that it could also serve as an alternative reactive robot for the reorientation task (Lund & Miglino, 1998; Nolfi, 2002). It differs from these robots in several respects. First, it uses a far simpler (and cheaper) architecture: it is a LEGO robot that uses far fewer sensors than the Khepera robots that have been described. Second, rather than using evolutionary techniques to develop controllers for this machine, we instead developed a subsumption architecture for navigation. Third, the most advanced version of antiSLAM uses both ultrasonic and light sensors that permit it to react to both local feature cues as well as the overall geometry of its environment.

9.12.2 Ultrasonic Sensors

The Khepera robots for the reorientation task dealt with rectangular arenas by using eight infrared sensors that would detect obstacles when the robot was fairly close to them. A single LEGO NXT brick does not permit this many sensors to be used at one time. Our alternative strategy was to start with a robot that had only two sensors, and to use sensors that had a longer range than the ones described in preceding pages (Lund & Miglino, 1998; Nolfi, 2002).
Our initial robot used two LEGO ultrasonic sensors. These sensors essentially act as sonar devices, sending out ultrasonic signals and listening for an echo (Astolfo et al., 2007). The timing of the echo is used as a measure of the distance between the sensor and the obstacle that reflected the signal; the maximum range of these sensors is around 100 inches (Boogaarts, 2007). As is the case with other sensors, the ultrasonic sensor can be set to return values in different modes—for instance, in inches or in centimetres. We decided to use the sensor in raw mode, where it returns a value of 255 when a reflecting obstacle is at the maximum range of the sensor, and decreases to a minimum of 0 as the sensor moves closer and closer to the obstacle.

One issue with ultrasonic sensors is that LEGO builders do not recommend using more than one at a time. “It cannot be used in an area in which another ultrasonic sensor is already at work because the signals sent out by the two sensors will interfere with each other and cause misreading” (Boogaarts, 2007, p. 39). However, because we needed our robot to be simultaneously sensitive to walls at different positions (e.g., when facing a corner of a rectangular arena), we needed to use more than one ultrasonic sensor. Thus, our first version of antiSLAM explored whether two ultrasonic sensors could be successfully used in Level 0 of a subsumption architecture for navigation.

**9.13 antiSLAM overview**

**9.13.1 Modifying Vehicle 2**

The antiSLAM robot (Figure 9-5) can be thought of as a descendant of a Braitenberg Vehicle 2 (whose construction was described in detail in Chapter 4). Indeed, the initial construction of antiSLAM is identical to the first several steps used to build Vehicle 2. As well, both robots include two light sensors that can be used to independently control the speed of two rear motors, steering the robot around an environment. However, antiSLAM differs from Vehicle 2 in two important ways. First, its light sensors point outward from the robot. Second, antiSLAM includes two additional ultrasonic sensors that also point outward. Compare Figure 9-5 to Figure 4-1 in Section 4.2 to see the similarities and differences between the two robots.

AntiSLAM, in its complete programmed form, uses its ultrasonic sensors to follow walls in a rectangular arena, slowing to a halt when these sensors detect a corner. It then initiates a turning routine to exit the corner and continue exploring. Its light sensors can be used to process local features—for instance, it can have a preference to approach an illuminated location. It uses these capabilities to reliably
find a target location that is associated with particular geometric and local features. When local features are removed, it navigates the arena using geometric cues only, and produces rotational errors. In short, it produces some of the key features of the reorientation task—however, it does so without creating a cognitive map, and even without representing a goal.

AntiSLAM’s subsumption architecture is also interesting. We will see that as specific layers are added to this architecture, antiSLAM transforms from a Vehicle 2 to a Vehicle 3 and upward through Braitenberg’s evolutionary progression of machines (Braitenberg, 1984).

9.14 FROM VEHICLE 2 ONWARD

9.14.1 Foraging for Parts

As Braitenberg explored different vehicles in his thought experiments, he did so by following an evolutionary path, where one machine could be described as being the previous device with an additional specialization or capability. For example, Vehicle 3 is a Vehicle 2 that includes more than one kind of sensor, with a myriad of connections (inhibitory and excitatory, ipsilateral and contralateral) between sensors and motors (Braitenberg, 1984). Vehicle 4 is a Vehicle 3 that incorporates non-linear relationships between sensor readings and motor behaviours.
When it is constructed and programmed, it will be apparent that antiSLAM is an example of a Braitenberg Vehicle 4. The parts that are required to construct an antiSLAM robot are provided in Figure 9-6. The pages that follow provide words and images that describe how to construct this machine. If the reader would prefer to use wordless, LEGO-style instructions, they are available as a pdf file from the website that supports this book (http://www.bcp.psych.ualberta.ca/~mike/BricksToBrains/).

**9.15 A SPINE FOR ANTI SLAM**

**9.15.1 Creating a Chassis**

As antiSLAM can be viewed as an evolutionary descendant of Vehicle 2, it should not be surprising to find that antiSLAM’s chassis — its internal spine — is identical to the chassis that was constructed for Vehicle 2 and described in Sections 4.4 and 4.5. The images below indicate how to construct the chassis.
9.16 STRUCTURE FROM MOTORS

9.16.1 Motors and Axles

The next step in building antiSLAM is to attach two motors to the chassis, and then to attach two axle assemblies to each of the motors, as shown in Figure 9-9. At this time additional pins are added to the robot; soon these pins will be used to permit the NXT brick to be attached to the chassis. As was the case for Vehicle 2, the physical structure of each motor is incorporated into the chassis design in order to reinforce the chassis.

9.17 SENSOR SUPPORTS AND FRONT WHEELS

9.17.1 Creating Sensor Supports

AntiSLAM requires that four different sensors be mounted near the front of the robot. They are supported by two double-bent liftarms that are pinned to the NXT brick—which is also added at this stage—as is illustrated in Figure 9-10. The additional axles and pins that are inserted into the double-bent liftarms will be used to support wheels, ultrasonic sensors, and light sensors on either side of the robot.
9.17.2 Front Wheels

While Vehicle 2 used a “front slider” to support the front weight of the robot, antiSLAM works best if two small wheels are used to keep the front stable as it scurries about the environment. The wheels are wedge belt wheels, with tires, that are mounted onto beams that in turn are attached to the double-bent liftarms, as shown in Figure 9-11.

9.18 SENSOR ARRAYS

9.18.1 Mounting Sensors

The sensor mounts that were added in Section 9.17 are used to support two ultrasonic sensors. It is these sensors that permit antiSLAM to respond to “geometric features” when reorienting itself. Each sensor is mounted on a LEGO hinge so that it can be pointed outward from the front of the robot, as shown in Figure 9-11. A light sensor, that also points outward, is also mounted just above and behind each ultrasonic sensor, as is also illustrated in Figure 9-11.
9.19 ANTI-SLAM’S REAR WHEELS AND CABLES

9.19.1 Rear Wheels

As was the case for Vehicle 2, 81.6 × 15 LEGO motorcycle wheels, with tires, are used as the rear wheels for antiSLAM, as shown in Figure 9-12. Each of these wheels is independently powered by its own motor, so the robot can be steered by manipulating robot speed. These are very large wheels, and will be able to move the robot along at a fairly high speed.

9.19.2 Connecting Cables

The final step in building antiSLAM is to use cables to connect the various sensors and motors to the NXT brick’s ports. The light sensor on the robot’s right is connected to Input Port 4, and the ultrasonic sensor on the robot’s right is connected to Input Port 3. The light sensor on the robot’s left is connected to Input Port 1, and the ultrasonic sensor on the robot’s left is connected to Input Port 2. The robot’s right motor is connected to Output Port B, while the robot’s left motor is connected to Output Port C.

When Vehicle 2 was cabled in Chapter 4, it was noted that there was a choice between an ipsilateral and contralateral relationship between sensors and motors (Section 4.14). Furthermore, these relationships were put in place by changing the ports that the motors were plugged into. Because antiSLAM uses a larger number of sensors, the cables will not be manipulated after they are connected to the ports described in the previous paragraph. It is still the case that relationships between sensor types can be ipsilateral or contralateral, but the nature of the sensor-to-motor mapping will be handled by software in this robot.
9.20 AntiSLAM Level 0: drive

9.20.1 Subsumption Architecture

With Section 9.17, the construction of antiSLAM is complete. However, we need to program this robot in order to bring it to life. As was the case with the LEGO Tortoise (Chapter 7) and the LEGO Lemming (Chapter 8), we have adopted a subsumption architecture to do so.

The code below provides Level 0 of this subsumption architecture; this level provides the basic capability “drive.” That is, Level 0 converts a signal from an ultrasonic sensor into a motor speed, and drives the motor forward at that speed. The right motor is controlled by the task DriveRight(), while the left motor is controlled by DriveLeft().

The Level 0 code that is provided does not explicitly state how motor speeds are calculated. This is actually accomplished by a lower-level task that is affected by several higher levels in the architecture; this “Level -1” task will be described momentarily. For the time being, recognize that as an obstacle gets closer to an ultrasonic sensor, the value output by the sensor decreases. This behaviour is mapped into motor speed, so that as a sensor gets closer and closer to a wall in a rectangular arena, the motor that the sensor controls will slow, eventually coming to a virtual halt.

The relationship between ultrasonic sensors and motors in antiSLAM is contralateral. Thus, Level 0 essentially causes antiSLAM to become a version of Vehicle 2 that uses ultrasonic sensors instead of light sensors. The robot will steer away from walls—if started in a corridor, it will move quickly down the hallway, keeping as far away from the walls on both sides as best it can. However, eventually the corridor will end. As the robot approaches the end, it will begin to slow, and it will also begin to turn. Eventually the relationship between its ultrasonic sensors and its motors will cause it to turn into a corner of the corridor, where it will come to a halt.

/*=====Level 0: Drive==================================
Feed the distance from each ultrasonic sensor to the motor.
The robot is wired contralaterally, and thus avoids all walls equally. As a result, when it reaches a corner, it slows down and ends up stopping, getting corner detection for free.
*/

task DriveRight(){
    while(true){
        OnFwd(RightMotor, RightSpeed);
    }
}

290 FROM BRICKS TO BRAINS
task DriveLeft(){
    while(true){
        OnFwd(LeftMotor, LeftSpeed);
    }
}

9.21 LEVEL 1: ESCAPE

9.21.1 Importance of Escaping

Our earlier discussion of reactive robots that were capable of performing the reorientation task in a similar fashion to animals (Nolfi, 2002; Nolfi & Floreano, 2000) noted that rotational errors were produced by the methods used by the robot to leave a corner. Level 1, whose code is provided below, is a simpler version of this approach—it causes the robot to spin out of a corner, but the actual spin is accomplished without using sensors to control precise movements.

Level 1 begins by taking advantage of the rotational sensors that are built into NXT motors. The task Retreat() examines the output of these sensors, and determines if the wheels are rotating less than a threshold amount. If so, the task assumes that an obstacle is impeding the robot from all frontward directions, and so the motors are controlled in such a way that the robot spins in place. When the spin is completed, the robot will be pointing approximately $45^\circ$ away from the obstacle. Just prior to spinning, the routine causes the robot to emit a sound. This sound can be used by the experimenter as an objective measure of when this routine is called, which will be important when the behaviour of the robot is measured.

The robot’s spin is accomplished by manipulating variables used in the still-to-be-described Level-1 to alter motor speed. When Sensitivity is set to 0, motor speed will not be affected by other sensors, such as the ultrasonic detectors. When Reverse is set to 35, this will cause the motors to move in opposite directions, producing the spin, which is simply conducted for a set period of time using the Wait function.

Note that Level 1 fits into the subsumption architecture by sending signals that directly modifies the behaviour of one lower level (Level -1), which in turn affects the behaviour of another lower level (Level 0).

Note too that Level 1 converts antiSLAM from a Vehicle 2 into a Braitenberg Vehicle 3 (Braitenberg, 1984), because now motor speeds are affected by two different types of sensors (ultrasonic sensors and rotation sensors), which can have different, and even opposing, sensor-to-motor relationships.
/*-----Level 1: Escape -------------------------------
If the motors move less than a stated threshold over a delay period, the robot's sensors are temporarily overridden as it spins around. It ends up pointing approx. 45 degrees from the corner when normal operation resumes. */

```c
int Threshold, Delay;

void Spin(){
    ResetRotationCount(LeftMotor); ResetRotationCount(RightMotor);
    Sensitivity = 0; Reverse = 35; //Disable sensors, enable spin term
    Wait(4000); //Time to spin in milliseconds
    Reverse = 0; Sensitivity = 1; //Return to default settings
    ResetRotationCount(LeftMotor); ResetRotationCount(RightMotor);
}

task Retreat(){
    long RotCount; //Tracks motor rotation.
    while(true){
        RotCount = MotorRotationCount(LeftMotor) + MotorRotationCount(RightMotor);
        Wait(Delay);
        if(((MotorRotationCount(LeftMotor)+MotorRotationCount(RightMotor) - RotCount)
            < Threshold) && Reverse==0){//If motors slow down while not spinning
            PlayTone(440, 500); //Beep to indicate spinning and data point.
            Spin();
            Wait(500);
        }
    }
}
```

9.22 LEVEL 2: FOLLOWING WALLS

9.22.1 Biasing Lower-level Behaviour

As was previously described, the Level 0 behaviour makes all walls equally aversive — antiSLAM attempts equally far away from walls on the left and walls on the right when its driving behaviour is primarily controlled by Level 0. Level 2 introduces a bias in the robot’s behaviour that causes it to have a wall-following preference. For instance, one can set this bias so that the robot keeps closer to the wall on its right as it moves through the arena. A change in the bias — a change in the robot’s “handedness” — can result in the robot keeping closer to the wall on its left.

It is the combination of robot “handedness” with its corner-escaping behaviour that should produce rotational errors in the reorientation task. That is, given its position when spinning out of a corner, and its preference to follow a wall on one side, the robot should move from that corner to the corner that is diagonally opposite in a reorientation task arena.
Note that Level 2 operates by manipulating variables (LeftBias, RightBias) that are used by the to-be-described Level -1 to determine motor speed. This is because one can give the robot a preference to be nearer to a wall on one side by providing a tendency or bias to turn in that direction, which can be accomplished by “tweaking” motor speeds. That is, by making the motor on the robot’s left turn faster than the motor on the robot’s right, the robot will have a tendency to turn to the right. When the robot gets too close to a wall on its right because of this turning, it will be straightened out because of the ultrasonic signals that detect and avoid the wall (Level 0). In short, the turning bias defined here in Level 2, combined with the obstacle avoidance achieved in Level 0, will interact to make the robot keep a wall closer on one side than the other as it moves.

```c
/*=====Level 2: Follow ================================
Introduce a bias to the robotís motors, causing it to îpreferî one motor (a sort of îhandednessî). The difference in motor power induces a turn, pushing the robot closer to one wall. Level 0 will straighten it out after it gets close enough, resulting in a robot that follows walls on one side. */
int Nearest(bool hand){ /*This returns the value of the sensor nearest the wall.
   Note: înearestî is defined by the robotís îpreferredî side*/
    if (hand) return SensorUS(RightEar);
    else return SensorUS(LeftEar);
}
bool preferred; //Determines which side the robot prefers. True = right turns.
int bias; //The strength of the bias term. See the main task.
task Seek(){
    //Set the bias to the appropriate side.
    while(true){
        if (preferred) {RightBias = bias; LeftBias = 0;}
        else {LeftBias = bias; RightBias = 0;}
    }
}
```

**9.23 LEVEL 3: USING LIGHT AS A LOCAL FEATURE**

9.23.1 Local Feature Sensitivity

Nolfi’s robots (Nolfi, 2002; Nolfi & Floreano, 2000) used an array of sensors to detect proximity to walls, but did not use any sensors to process local features, which is a key ingredient of the reorientation task. One of our goals for antiSLAM was to use its light sensors to detect a particular local feature (wall “colour”). This ability is provided by Level 3, whose
code is given below. Our expectation is that when Level 3 is operating, if a goal corner is marked by local features (i.e., if it was illuminated by lights, and therefore was brighter than other locations in the arena), then this information can be used by the robot to prevent rotational error from occurring.

How is such feature sensitivity to be accomplished? The two light sensors mounted on antiSLAM can be used to influence motors in a fashion similar to that used in Vehicle 2 (Chapter 4). In antiSLAM the light sensors are connected in such a way that they influence the motor on the contralateral side of the robot. Thus, when light is detected, the robot has a tendency to turn toward it and approach it, accelerating as the light gets brighter. The only difference between antiSLAM and Vehicle 2 is that in antiSLAM motor speeds are affected by ultrasonic sensors as well. As a result, rather than affect motor speed directly, Level 3 computes values that are passed on to Level -1, which combines readings from all sensors to determine motor speed. Level -1 is described on the next page.

Note that in the code below the sensor values are modified by the term Vision. This provides a weight to light sensor information (relative to a different weight applied to ultrasonic sensor information). In the experiments reported later in this chapter, the light sensors were given a 60% weight, and the ultrasonic sensors were given a 40% weight.

One important characteristic of antiSLAM is that its embodiment affects what Level 3 detects when the robot gets very close to a brightly lit corner. Because the light sensors are spread apart, and because they are angled outward, the robot has a “blind spot” to nearby light directly in front of the machine. So, when it moves into a brightly lit corner, when the light enters the bright spot the light sensors detect little light, and the robot slows down.

```c
/*=====Level 3: Feature ===============================
Enables and reads the light sensors (eyes) as a percentage based on Vision (a sensitivity term), such that more light = more speed. Since the connection is contralateral, this results in the robot turning toward sources of light. However, level -1 weighs this visual sense with the earlier ultrasonic sense, allowing both terms to influence the robotís final behavior. */

int Vision; //The strength of the light sensors in percent.

task See(){
    //Sets the strength of the robotís visual response to a scaled percentage.
    while(true){
        LVis = Sensor(LeftEye)*Vision/100;
        RVis = Sensor(RightEye)*Vision/100;
    }
}
```

FROM BRICKS TO BRAINS
9.24 LEVEL -1: DETERMINING MOTOR SPEEDS

9.24.1 Finally, Level -1

The behaviour of antiSLAM is completely determined by the speeds of its two motors. The speeds of these motors depend on the signals being sent by ultrasonic, rotation, and light sensors. The speeds are directly affected by combinations of these signals, as well as by other factors, such as the robot’s bias to follow walls on one side or the other.

The actual calculation of motor speed requires that these various factors be combined, and this is accomplished by Level -1, whose code is provided below. All of the variables that are used in the calculations given below have values that depend either directly on sensor readings or on signals that are sent down to this level by higher levels in the subsumption architecture, as has been noted in the preceding pages of this chapter. Note that the ultrasonic sensors are weighted by the Hearing variable in this code. This variable provides a weight to the ultrasonic signal; a similar weight was applied to the light sensor signal in the code that was described for Level 3.

```c
/* ====== Level -1: Integration ============= */

Each of the terms (Sensitivity, Reverse, LeftBias, RightBias) is part of a later level’s connection to the motors. See the main task to see their defaults.
On its own, this task does nothing. However, it will function at every level without modification. */

int Sensitivity, Reverse, LeftSpeed, RightSpeed, LeftBias, RightBias, LVis, RVis;
int Hearing;
task Drive(){
    while(true){
        //Hearing/255i converts from responsive raw ultrasonic to % motor speed.
        RightSpeed = ((SensorUS(RightEar)*(Hearing-LeftBias)/255)+RVis)*Sensitivity+Reverse;
        LeftSpeed = ((SensorUS(LeftEar)*(Hearing-RightBias)/255)+LVis)*Sensitivity-Reverse;
    }
}

9.25 THE MAIN TASK

9.25.1 Putting It Together

In order to get all of the levels in the subsumption architecture working, the main task is used to initialize some important variables and to call the various tasks that define each level. Note that by modifying the
main task—for instance, by commenting out a call to a task—one can selectively remove layers from the subsumption architecture in order to investigate changes in behaviour. Note too that the variables that are used to weight the light sensor and ultrasonic sensor signals (Vision and Hearing) are set in the main task. Note that the complete program for this robot is available from the website that supports this book (http://www.bcp.psych.ualberta.ca/~mike/BricksToBrains/).

// Anti-SLAM v8
// Convention: Left and Right refer to the SENSOR/MOTOR, NOT THE WIRING.
// See the wiring diagrams in Chapter 9.
define LeftMotor OUT_B
define RightMotor OUT_C
define LeftEar S2
define RightEar S3
define LeftEye S1
define RightEye S4

//=== Main Task =======================================================
task main(){
// Set up ultrasonic sensors and speed calculation weights.
SetSensorLowspeed(LeftEar);
SetSensorLowspeed(RightEar);
SetSensorMode(LeftEar, SENSOR_MODE_RAW);
SetSensorMode(RightEar, SENSOR_MODE_RAW);
Sensitivity = 1; // Level 0 connection: Ultrasonic sensitivity. Default 1.
Reverse = 0; // Level 1 connection: Lets robot spin and escape. Default 0.
LeftBias = 0; // Level 2 connection: Causes robot to prefer left turns. Def. 0.
RightBias = 0; // Level 2 connection: As above, but prefers right turns. Def. 0.
LVis = 0; // Impact of the left eye on movement. Zero at this level.
RVis = 0; // Impact of the right eye on movement. Zero at this level.
Hearing = 100; // Strength of ultrasonic sense. (Overridden at level 3.)
start Drive; // Starts mapping the motor speeds to the collective input.
// Level 0.
start DriveRight; // Turn the right motor on.
start DriveLeft; // Turn the left motor on.
// Level 1. (Delete below this line for a level 0 robot.)
Threshold = 360; // Combined motor movement to be considered on. Default 360.
Delay = 5000; // How long the robot needs to have been stopped. Default 5000ms.
start Retreat; // Allow the robot to escape corners.
// Level 2. (Delete below this line for a level 1 robot.)
pREFERRED = true; // True for left-handed (right-following), false otherwise.
bias = 40; // Fixed value for handedness bias. Default 40.
start Seek; // Follow the wall on your preferred side.
9.26 PRIMITIVE BEHAVIOURS

9.26.1 Levels -1 + 0

With the robot constructed and programmed, we are now in a position to observe its behaviour. As was the practice in Chapter 7, let us consider how antiSLAM’s behaviour changes as more and more layers of its subsumption architecture are added. Examples of the behaviour described in the following pages are provided in Video 9-1; this video is provided by the website for this book (http://www.bcp.psych.ualberta.ca/~mike/BricksToBrains/).

The most primitive behaviour of antiSLAM is produced when only Level -1 (which computes motor speed) and Level 0 (which uses the two ultrasonic sensors to influence motor speed) of the subsumption architecture are operational. When placed in a long hallway, antiSLAM turns itself so that it is pointing down the length of the hallway, accelerating as it points away from nearby walls. It then quickly propels itself down the hallway, keeping to its centre. During this journey, the walls of the hallway may not be perfectly uniform, because of doorways or small alcoves. When these variations are encountered, they influence the ultrasonic sensors, and the robot veers slightly toward the open space.

If the hallway is not completely clear, more interesting behaviour is observed. The legs of chairs, tables, and people all serve as obstacles in the hallway, and are detected by antiSLAM’s ultrasonic sensors. The robot nimbly steers around these obstacles, and resumes its dash down the middle of the hallway when the obstacles are behind it.

The hallway, however, is not infinitely long. Eventually antiSLAM encounters a set of doors that block its way and that cannot be avoided. As the robot nears the end of the hallway, it begins to decelerate. It slows and turns, and eventually is trapped in a corner from which it cannot escape. It approaches the corner, its wheels barely turning, and its journey has ended.

// Level 3. (Delete below this line for a level 2 robot.)
// Set up eyes.
SetSensorType(LeftEye, SENSOR_TYPE_LIGHT_INACTIVE);
SetSensorMode(LeftEye, SENSOR_MODE_PERCENT);
SetSensorType(RightEye, SENSOR_TYPE_LIGHT_INACTIVE);
SetSensorMode(RightEye, SENSOR_MODE_PERCENT);
Hearing = 40; // % of ultrasonic sense that feeds to the motors. Default 40.
Vision = 60; // % of light sense that feeds to the motors. Default 60.
start See;
Thus, the most primitive antiSLAM is very capable of moving through its world, turning away from obstacles whenever possible, and keeping as great a distance between it and walls as possible. Eventually, though, it finds itself in a corner. Note that it does all of this without having a cognitive map, and without any explicit knowledge about hallways, obstacles, or corners. All of this behaviour is in essence the product of a cousin of Vehicle 2 whose two motors slow down when obstacles are near, and speed up when the path is clear. That is, this behaviour is the result of some basic rules, chance, and the structure of the environment. It is not the result of spatial representations.

9.26.2 Levels -1 + 0 + 1
The most primitive version of antiSLAM is capable of finding corners, but when it succeeds at this task, it is trapped. This problem is solved by adding Level 1 to the mix. Recall that this level uses the rotation sensors in the two motors to detect when antiSLAM has slowed to the point that it has essentially stopped. In this situation, Level 1 manipulates the motor speed equation of Level -1 in such a way that the robot reverses itself and turns around. The robot emits a brief tone to inform its observers that Level 1 has detected the circumstances that cause it to change the robot’s behaviour. After the evasive manoeuver has been performed, the robot reverts to its normal exploratory behaviour.

When this more advanced robot is run, its behaviour is very similar to the robot described in Section 9.27.1. It too moves down the middle of hallways, avoids obstacles, and stops at discovered corners. However, once a corner has been found, after remaining stationary for a bit the robot beeps, turns away from the corner, and points at an angle down the hallway. Then it accelerates, and continues its journey in the opposite direction. Again, this behaviour is produced without the need of explicit spatial representations or memories.

9.27 BIAS AND REORIENTATION
9.27.1 Levels -1 + 0 + 1 + 2
The next stage in the evolution of antiSLAM is to add Level 2, which adds a bias that causes the robot to be closer to the wall on its right than it is to the wall on its left. This is accomplished by having the robot always turn slightly to its right; when it gets too near the wall, the activity of Level 0 causes it to straighten out.

When this version of the robot is observed, its behaviour is very similar to its ancestor described in Section 9.27.2, with one exception: the robot no longer keeps to the middle of the hallway as it explores.
Instead, it has a marked tendency to be closer to the wall on its right than on its left. Again, it avoids obstacles, and finds (and then escapes) corners. However, after watching the robot perform its exploration for a while, it becomes evident that of the four corners of the hallway that it could discover, it has a strong tendency to only find two, and these two corners are geometrically equivalent. Is this simple robot—that only reacts to obstacles and has a turning bias, and does not have a cognitive map or spatial representation—capable of producing a key finding in the reorientation task, that of rotational error?

### 9.27.2 Rotational Error and AntiSLAM

In order to answer this question, we conducted an experiment that was similar to the one used to examine the spatial reorientation of a more sophisticated reactive robot (Nolfi, 2002; Nolfi & Floreano, 2000). Our robot was placed in a small, empty testing room that provided rectangular arena that was 2.4 metres long and 1.65 metres wide. Location 4 in Figure 9-15 was considered to be the goal corner. Adopting Nolfi’s methodology, antiSLAM was placed in one of eight different starting locations. The eight locations are illustrated in Figure 9-13. In this figure, each location is represented by an arrow. The base of each arrow indicates where the back of antiSLAM was positioned, and the arrowhead indicates the direction in which antiSLAM was pointed. The robot was started, and then permitted to explore the arena for 5 minutes. Each time that the robot initiated its Level 1 “escape corner” routine, the location of the robot in the arena was recorded. The entire experiment involved recording the robot’s behaviour after it was started once at each of the eight starting locations.
The results of the experiment are provided in Figure 9-14, where the number provided in each corner indicates the percentage of times that the “escape corner” routine was executed there. Note that while the robot would occasionally discover corners 1 or 3, most of the time it would stop at either the goal location (corner 4) or its geometric equivalent (corner 2). Furthermore, it visited these two locations roughly equally. As well, it would typically follow a path that took it directly between corner 4 and corner 2, and back again, once it had discovered one of these two corners, as indicated by the arrows in the figure. This pattern of results demonstrates that this version of antiSLAM produces rotational error!

**9.28 BEYOND ROTATIONAL ERROR**

9.28.1 Nolfi and Beyond

It could be argued that antiSLAM is limited by its bias to the right—while this enables rotational errors between corners 4 and 2 (Figure 9-16), it prevents the robot from generating similar behaviour if corner 1 or corner 3 were the goal. However Level 2 can be easily used to convert antiSLAM’s bias to the right into a bias to the left. One could imagine a higher level in its architecture (Level 4) that allowed the robot to explore all of an arena by occasionally changing its bias from right to left, and later from left to right. This could be associated with a learning routine: if the robot was reinforced when it stopped (e.g., by receiving a Bluetooth signal), it would be more likely to preserve its current “handedness.”
9.28.2 Feature Sensitivity

AntiSLAM can be extended beyond Nolfi’s robots by adding sensitivity to local features. In antiSLAM, this is accomplished by Level 3. Now, in addition to being sensitive to obstacles, antiSLAM will be attracted to lights. We can define corner 4 as the goal location by illuminating it with lamps.

We activated Level 3 in the robot, and defined Level -1 so that light sensitivity was given a weight of 60 and ultrasonic sensitivity was given a weight of 40. We hung two small lights over corner 4 and used them to project this local feature to this location. We then repeated the experimental methodology that was described in Section 9.28. The results are shown in Figure 9-15.

The addition of light sensitivity, and the lighting of corner 4, has changed the robot’s behaviour markedly. Now, from any location, it quickly sees the light and stops in corner 4. When this corner is escaped, the robot moves away. However, now it rarely goes to corner 2, or to any other corner for that matter. Instead, it usually slowly veers and heads toward location X midway between the long wall between corners 3 and 2, as shown in Figure 9-15. Here it stops — in spite of this location not being a corner! — executes its escape routine, and heads back to corner 4. This is very similar to the behaviour of ants in the re-orientation task, who — when featural information informs them that they are not going to the goal location — execute a U-turn and head back to a corner that is marked by the correct local feature (Wystrach & Beugnon, 2009).
The data reported in Figure 9-15 is atypical because it includes locations other than the four corners that are typically examined in the reorientation task. If we followed this convention, we would only be reporting the robot’s visits to the corners, which account for 65.4% of the data in the figure. Focusing only on these trials, antiSLAM visited the goal corner 82.3% of the time (i.e., 53.8/65.4), visited corner 3 on 5.8% of these trials, visited corner 2 on 11.9% of this subset of trials, and never visited corner 1. When reported in this way, the data reveals a strong effect of the local feature; reported as in Figure 9-17, though, the data makes the same case and provides some more interesting sense on how the path taken by the robot was affected.

It has been suggested that information about paths might provide information about strategies for solving the task (Cheng, 2008a); of course, antiSLAM never changes its strategy—changes in behaviour reflect changes in the environment in which it is situated, changes to which it reacts.

### 9.29 Moving the Local Feature

#### 9.29.1 Moving the Light

It was previously noted that researchers were interested in the reorientation task because they could reposition local cues before an agent was reintroduced to the arena, placing local features and geometric features in conflict. How does this type of manipulation affect antiSLAM when all of its software levels are running?

To answer this question, we conducted the same experiment that was described in Section 9.29, but in this version of the experiment corner 3 was illuminated. Now the local feature was present in a corner that was not preferred by the robot because of its bias to keep walls on its right. The results shown in Figure 9-16 indicate that there was a clear conflict between the local feature and antiSLAM’s geometric preferences.

The top portion of Figure 9-16 indicates the percentage of trials that antiSLAM executed its “escape” routine at various locations in the arena. Note that in addition to the four corners, the midpoints of all four walls now become locations of interest. If we restrict our attention to the corner locations, we note that the results indicate that antiSLAM’s behaviour is guided by both local and geometric features. It is most likely to stop at corner 3, which is illuminated as the goal location, but is not preferred by antiSLAM’s ultrasonic mechanisms (recall Figure 9-16). It still has a moderately strong preference to visit corner 4, which is nearly equal to its preference to visit the geometrically equivalent corner 2. Corner 1, which lacks the preferred geometry and the local feature, is
rarely visited. Such results are typical of studies of humans and animals in this task (Cheng & Newcombe, 2005).

The lower part of Figure 9-16 illustrates an example journey of anti-SLAM in this condition. The arrows indicate the path taken by the robot; the letters at the base of each arrow indicate their temporal order. In this example, the robot starts at location W, and moves to location X before being attracted to the light at corner 3. Note that the complicated journey depicted in this figure is the result of the competing influences of the walls, the turning bias, the brightly illuminated corner, and light reflecting from this corner to other walls. It is not the result of planning, or strategies, or the use of a cognitive map.
9.30 ALL LEVELS WITH NO LOCAL FEATURE
9.30.1 Turning Lights Off

In Section 9.28, we observed rotational error in antiSLAM when only ultrasonic signals were driving the motors. How does this robot behave when all of its levels are running, but when there are no lights turned on to serve as local features? We answered this question by observing the fully functional antiSLAM in a final version of the experiment where the conditions were identical to those explored with the less advanced robot in Section 9.28. The results are presented in Figure 9-17.

If we restrict ourselves to examining visitations to corner locations, which are almost always the locations of interest to reorientation task researchers, we see clear evidence of rotational error once again. AntiSLAM has a strong preference to visit corner 4, an equally strong preference to visit corner 2, and rarely executes its “escape corner” routine in either of the other corners.

However, antiSLAM executes its “escape corner” routine at other locations too; specifically locations X and Y in Figure 9-17. The arrows in Figure 9-17 illustrate a complicated course that antiSLAM takes to produce rotational error results. For instance, it frequently starts at corner 4, and heads to location X. It then turns from location X and stops at location Y. From location Y it will turn and stop at corner 2. A journey in the opposite direction, with stops at Y and then X, is frequently undertaken to move antiSLAM from corner 2 to corner 4. This journey is
not always taken. For instance, sometimes it will circle between corner 4 and location X, or between corner 2 and location Y.

Had we adopted the more typical practice of only reporting visitations to one of the four corner locations, our results would be very similar to those reported earlier in Figure 9-14. Clearly, though, the path that was taken by the robot in that figure is markedly different from the path that is illustrated in Figure 9-17. Unfortunately, except for rare instances (Wystrach & Beugnon, 2009), researchers do not report the paths taken by their biological subjects in reorientation arenas, so we cannot evaluate either Figure 9-14 or Figure 9-17 in terms of their fit to animal behaviour.

Why does the full-fledged robot produce this more complicated journey, compared to the more typical journey between corners that produced the data in Figure 9-14? The answer is that this robot is still functioning with the ultrasonic sensors weighted at 40, and the light sensors weighted at 60. There is a small amount of ambient light in the room, but it does not differentiate corners. However, the lower weighting of the ultrasonic signal means that its object-detecting abilities are diminished. As a result, these weaker abilities can be overwhelmed by its bias to turn toward the right. Much of the “clover leaf” trajectory that it takes between corners 4 and 2 is dictated by this bias overriding the other senses of the robot. In other words, the difference in paths between Figures 9-17 and 9-14 does not reflect a difference in strategy, but instead reflects different kinds of interactions between sense–act mechanisms.

9.31 REORIENTING REORIENTATION

9.31.1 Building a Better Mouse

One of my undergraduate students, Mike M., created a LEGO “mouse” as a robotics project. The robot had two rotation sensors mounted in front, to which were attached long “whiskers.” When the robot bumped into an object, both whiskers were pushed backward. This caused the robot to back away from obstacles. However, when only one of the rotation sensors was active, this was usually because the robot was near a wall. In this case, the robot followed the wall by keeping its whisker in contact with it.

AntiSLAM did not begin as an agent for the reorientation task. Instead, it was our attempt to create an NXT version of Mike’s mouse, one that followed walls using ultrasonic sensors. However, at the same time we were also working on another project involving artificial neural networks and the reorientation task (Dawson et al., 2008). It was only
when we watched antiSLAM that we saw a connection between the two projects, and that antiSLAM was capable of generating rotational error. This led to our subsequent discovery of Nolfi’s robots (Nolfi, 2002; Nolfi & Floreano, 2000), and the addition of Level 3’s sensitivity to local features as we “tweaked” antiSLAM’s later design with the reorientation task in mind.

AntiSLAM’s development illustrates one of the key advantages of the synthetic methodology: getting surprising results “for free.” AntiSLAM began as a set of simple capabilities that produced movement, wall following, and obstacle avoidance. It was never intended to produce rotational error, or provide insight into theories of spatial cognition—but we were fortunate enough to discover that it could accomplish these things.

9.31.2 Different Views of Reorientation

Another lesson from the history of antiSLAM’s development impacts theoretical considerations about the reorientation task itself. Of course, both antiSLAM and Nolfi’s robots indicate that some of the known regularities that govern this task can be produced by devices that simply react to their environment and do not employ spatial representations. Whether the same can be said of biological agents that perform this task is an open question (Cheng, 2008a). The reactive robots do raise important questions: clearly, benchmark results like rotational error do not necessarily imply the use of internal spatial representations. If biological agents are more than reactive, then additional results will be required to support this representational position.

Furthermore, theories that are specifically designed to explain the reorientation task are generally sense–think–act in nature, tacitly assuming that an agent has the primary goal of finding a previously reinforced location. So, researchers view the task in terms of possible goals—usually the four corner locations in a rectangular arena—and the features available at each location.

In contrast, because antiSLAM was designed to explore, it was not burdened by typical assumptions about the reorientation task. No goals were represented; no features present at corners were remembered. Reorienting behaviour emerged out of the interactions between simple sense–act reflexes and the environment. It is intriguing to consider how antiSLAM might behave in other traditional tasks used to study spatial behaviour, such as radial arm mazes.

Finally, the traditional view of the reorientation task usually places restrictions on what is relevant data. “The paths taken by animals have
not appeared in print. Nolfi’s work suggests that such paths can be relevant to interpreting what strategy an animal is using” (Cheng, 2008a, p. 155). AntiSLAM also shows that such non-traditional data is critical. Of particular interest were the paths that it took in the reorientation arena, and the fact that some locations—like midpoints between corners—were important determinants of its behaviour. Importantly, these trajectories say nothing about “strategy.” Instead, they show that complex routes emerge in the reorientation arena when simple sense–act couplings react to the information provided, in the absence of plans, strategies, or spatial representations.

9.32 HARD FUN AND HARD SCIENCE

9.32.1 Hard Fun

Synthetic psychologists often emphasize the simple nature of their machines by using toy-like descriptions (see Section 9.2.1). For instance, Braitenberg (1984, p. 20) notes that “it is pleasurable and easy to create little machines that do certain tricks,” making his synthetic psychology sound like fun.

There are, though, different kinds of fun. Consider the following anecdote (Negroponte, 1995). In 1989, elementary school children demonstrated their work with the MIT Media Lab’s LEGO and Logo projects to the media. “A zealous anchorwoman from one of the national TV networks, camera lights ablazing, cornered one child and asked him if this was not just all fun and games. … After her third repetition of the question and after considerable heat from the lights, this sweaty-faced, exasperated child plaintively looked into the camera and said, ‘Yes, this is fun, but it’s hard fun’” (p. 196).

“Hard fun” is an idea that has emerged from the study of how to use technology to enhance education (Papert, 1980, 1993). It is the idea that learners do not mind engaging in (and learning from) activities that are challenging provided that these activities are also fun, in the sense that they connect with learners’ interests (Picard et al., 2004).

Learning about embodied cognitive science by building LEGO robots is another example of hard fun. The fun part, of course, is engaging in bricolage in order to produce a working, behaving, and lifelike machine. What is it, though, about this kind of fun that makes it hard?

Building these robots is hard in the sense that it requires releasing traditional sense–think–act approaches to cognition and behaviour. Traditional robot development is the top-down task of “making the robot do what I want” (Petre & Price, 2004). This is accomplished by thinking of goal behaviours, by hypothesizing a set of internal mechanisms to
produce these behaviours, and by inserting this analysis into the robot’s program. However, in order to make a LEGO robot “do what I want,” this analytic approach must be abandoned, because it won’t succeed given the simplicity and memory limitations of these machines. Instead, one must permit the environment to make its important contribution to the performance of the machine, by working bottom-up to build simple sense–act mechanisms into the machine, and by letting emergent behaviours produced in early stages of development guide later modifications.

Overcoming this natural analytic tendency is a key issue in embodied cognitive science. For instance, many of the chapters in a recent handbook of situated cognition (Robbins & Aydede, 2009) reject some of the more interesting ideas in embodied cognitive science, such as the extended mind, and attempt to morph the notions of embodiment and situatedness into very traditional, and representational, cognitive theories (Dawson, 2009). One reason for this is a strongly entrenched goal of explaining behaviour by appealing to internal mechanisms, such as neural processing (MacIver, 2008).

The hard fun of using LEGO robots to explore embodied cognitive science is an attempt to deal with this issue at two different levels. First, the successful development of these machines requires students to focus on a simple robot’s immediate connection to its environment. Second, the hands-on experience of working with the robots and their environment provides a particular kind of scaffolding to support thinking about embodied and situated cognition. Simple environments scaffold the robots’ abilities; the robots themselves scaffold our understanding of embodied cognitive science.

9.32.2 Hard Science

This is not to say that the LEGO robots are merely pedagogical tools. Hopefully, the discussion of antiSLAM in the current chapter has shown that even a very simple LEGO robot can be used to contribute new insights into current issues in cognitive science. We hope that the reader will be inspired by this book to develop new machines that will continue this tradition.