Evolving a Neural Model of Insect Path Integration

Citation for published version:
Haferlach, T, Wessnitzer, J, Mangan, M & Webb, B 2007, 'Evolving a Neural Model of Insect Path Integration' Adaptive Behavior, vol. 15, no. 3, pp. 273-287. DOI: 10.1177/1059712307082080

Digital Object Identifier (DOI):
10.1177/1059712307082080

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Publisher's PDF, also known as Version of record

Published In:
Adaptive Behavior

Adaptive Behavior

Evolving a Neural Model of Insect Path Integration

Thomas Haferlach, Jan Wessnitzer, Michael Mangan, Barbara Webb
Institute of Perception, Action and Behaviour, University of Edinburgh, UK

Path integration is an important navigation strategy in many animal species. We use a genetic algorithm to evolve a novel neural model of path integration, based on input from cells that encode the heading of the agent in a manner comparable to the polarization-sensitive interneurons found in insects. The home vector is encoded as a population code across a circular array of cells that integrate this input. This code can be used to control return to the home position. We demonstrate the capabilities of the network under noisy conditions in simulation and on a robot.

Keywords path integration · direction cells · genetic algorithm · neural network · simulation · robot

1 Introduction

Path integration, a term coined by H. Mittelstaedt and Mittelstaedt (1973), is a navigational strategy that explains the ability of some animals to return home on a direct path after a long and tortuous foraging excursion. Even in environments completely lacking distinct landmarks, or in the dark, an animal can use sensing of distance and direction traveled to integrate its velocity vector over time and thus estimate where it is in relation to its starting position. Path integration is a strategy used by a wide range of animals, including rats (Benhamou, 1997; Kimchi, Etienne, & Terkel, 2004), dogs (Seguinot, Cattet, & Benhamou, 1998), humans (M. Mittelstaedt & Mittelstaedt, 2001), spiders (Moller & Goerner, 1994) and desert ants (Wehner, 2003). The importance of path integration to the field of robotics as a navigation strategy is best exemplified by NASA's *Mars Rover* of the *Pathfinder* mission (Matthies et al., 1995).

The homing behavior in the desert ant (*Cataglyphis fortis*) has been intensively studied and there is a rich corpus of experimental data available (e.g., Bisch-Knaden & Wehner, 2003; Collett, Collett, & Srinivasan, 2006; Wohlgemuth, Ronacher, & Wehner, 2001). To perform path integration, ants use the polarized light from the sun to derive their current compass heading (Wehner, 1998). *Sahabot*, developed at the University of Zurich by Lambrinos, Moeller, Labhart, Pfeifer, and Wehner (2000), uses polarization sensors modeled on those of the ant, but the underlying path integration system relies on the equations proposed by H. Mittelstaedt and Mittelstaedt (1973) rather than a neural implementation. Ronacher and Wehner (1995) showed that even after removing all allothetic (external) cues desert ants can accurately estimate distance traveled, and recent evidence suggests they actually "count" their steps to gauge distance traveled (Wittlinger, Wehner, & Wolf, 2006). Ants thus rely on allothetic and idiothetic (internal) inputs to determine heading and distances traveled, respectively.

Several neural network models for path integration in ants have been developed. Among the best known are the models by Hartmann and Wehner (1995) and Wittmann and Schwegler (1995). Both models use a hand-designed neural network to encode the homing

Correspondence to: Jan Wessnitzer, Institute of Perception, Action and Behaviour, University of Edinburgh, The King's Buildings
Mayfield Road, Edinburgh, EH9 3JZ, Scotland, UK.
E-mail: jwessnit@inf.ed.ac.uk
Tel.: +44 (0)131 651 7195; *Fax*: +44 (0)131 651 3435

(2007), Vol 15(3): 273–287.
DOI: 10.1177/1059712307082080

vector and translate it into the appropriate homing navigation. The first model uses a linear neural chain to encode distance and circular neural chains to encode the home vector angle and the current heading. The second uses a sinusoidal array (Touretzky, Redish, & Wan, 1993), which represents the distance and angle of the home vector as amplitude and phase of a sine wave spatially discretized across a set of neurons. Both models can do accurate path integration and also, with suitable parameter tuning, reproduce the systematic errors seen in the experimental data. There are also several neural models of path integration in rats, usually designed as input to place cell systems based on the hippocampus (e.g., Burgess, Donnett, & O'Keefe, 1998; Degris, Lacheze, Boucheny, & Arleo, 2004).

More recently, Vickerstaff and Di Paolo (2005) used an evolutionary approach and found a compact network that implements the bi-component model suggested by H. Mittelstaedt and Mittelstaedt (1973). However, this result depended on the input to the network taking the form of a sinusoidal compass sensor response function. When using more biologically plausible sensor response functions, Vickerstaff and Di Paolo report a failure in evolving successful homing behavior.

In this article we also take an evolutionary approach, but focus on finding a network for path integration that works with more realistic inputs from direction cells. In particular we consider how a population of neurons can encode (lower-dimensional) sensory information for path integration. This results in an elegant network solution, which combines distance and heading information using a population code in a small circular neural chain. These findings are reminiscent of recent neurobiological findings regarding the map-like representation of e-vector orientations (Heinze & Homberg, 2007) in the brain of an insect. We test the evolved model in noisy simulations and on a robot.

2 Methods

The agent used in the genetic algorithm (GA) is simply modeled as having a position (x,y) and heading θ on an unbounded two-dimensional plane. The agent has predefined sensors and effectors (described below) connected by a neural network for which the structure, weights and parameters are evolvable, being directly encoded on the agent's chromosome.

2.1 Sensors and Effectors

Some insects (for example, Hymenoptera – including the desert ant *Cataglyphis fortis*) have specialized dorsal arrays of photoreceptors at the rim of the compound eye adapted to sensing the polarized light patterns in the sky. This provides directional information, sometimes called a "sky-compass". Interneurons, termed *POL* neurons, that receive input from this specialized region of the eye, and thus exhibit activity changes relative to the heading of the animal, have been intensively studied in the cricket (Labhart & Meyer, 2002).

The sensors available to the agent approximate these interneurons by defining a set of "direction cells" with preferred directions evenly distributed around 360°. In the evolutionary runs described here, the number of cells was fixed at three. The activation of each cell is calculated by performing the dot product of the preferred direction of the cell (h_p) and the actual heading (h_a) of the agent:

$$\text{firing rate} = \begin{bmatrix}\cos(h_\mathrm{p})\\ \sin(h_\mathrm{p})\end{bmatrix} \cdot \begin{bmatrix}\cos(h_\mathrm{a})\\ \sin(h_\mathrm{a})\end{bmatrix} + [\xi \sim N(0,\sigma)] \tag{1}$$

where ξ is a Gaussian noise term with mean 0 and standard deviation σ. The ensemble of direction cells thus encodes the current heading of the agent as a population code as shown in Figure 1.

Each agent has two turn effectors, for turning left or right. The input defines how sharp the turn will be. Currently the agent is maximally allowed a 20° turn per iteration. The agent is also driven forward at a constant speed.

2.2 Neurons

The networks used two kinds of neurons. A standard sigmoid neuron was implemented:

$$o_i = \frac{1}{\left(1 + e^{-aI_i + b_s}\right)} \tag{2}$$

where the output o_i depends on the summed input I_i, a parameter a and bias b_s. The input to a cell is calculated by summing up the output of all connected cells multiplied by the connection weights:

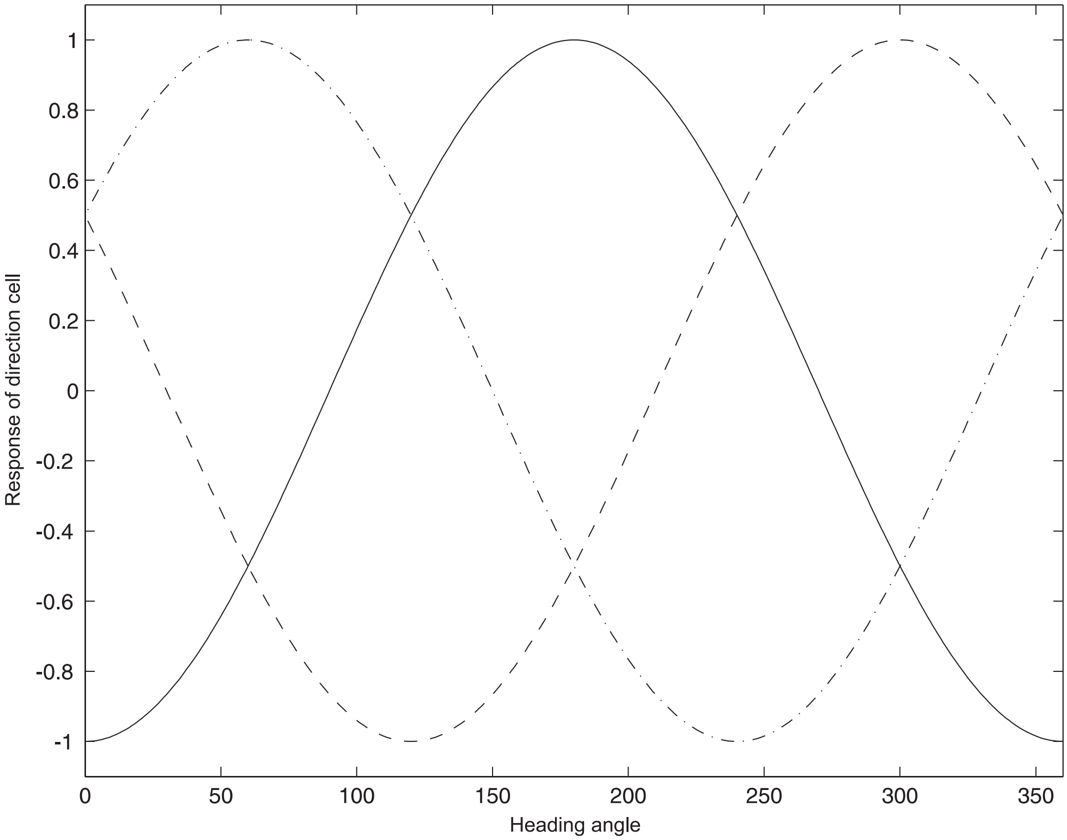


Figure 1 Response of three direction sensors with preferred directions of 60°, 180° and 300°.

$$I_i = \sum_{j=1}^{n} w_{ji} o_j \tag{3}$$

where I_i is the input to cell i, w_{ji} is the connection weight from cell j to cell i and o_j is the output of cell j. In addition to the sigmoid neuron, a CTRNN (continuous-time recurrent neural network) neuron was implemented. The cell potential c of the neuron is updated as follows:

$$\tau \frac{\mathrm{d}c}{\mathrm{d}t} = -c + I \tag{4}$$

The output of the neuron applies the sigmoid function to the cell potential:

$$o_i = \frac{1}{\left(1 + \mathrm{e}^{-c + b_c}\right)} \tag{5}$$

A biological equivalent of such a neuron that integrates its input over time was described in the rat (Egorov, Hamam, Fransen, Hasselmo, & Alonso, 2002).

2.3 Evolutionary Process

2.3.1 Genetic Encoding and Selection Method A genetically inspired neural encoding was adopted based on Fullmer and Miikkulainen (1992). The chromosome allows a variable number of neurons and connections encoded as a list of integers. Neurons and sensors are defined and encoded between start markers and end markers. A start marker indicates the start of a neuron or sensor and the end marker indicates the end of the encoding. A neuron's parameters and excitatory or inhibitory connections are encoded between these markers. Figure 2 shows an example of how a chromosome (the genotype) is converted to an agent's neural network (the phenotype). The topography, connection weights, sensors and effectors of the agent are all encoded on a fixed-size chromosome. The genome size was limited to 500 parameters, which gave an approximate upper bound of 50 neurons.

Localized *tournament selection* was used to select individuals. Each genome has a location in one-

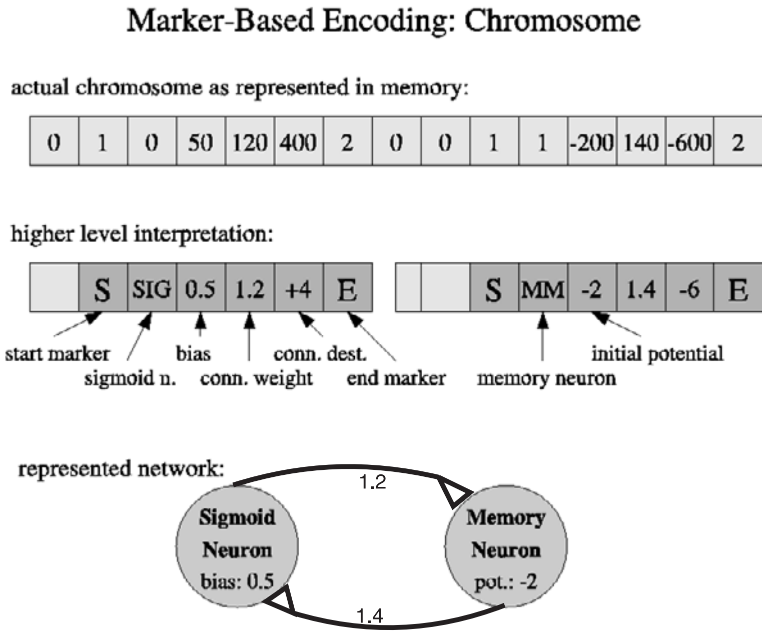


Figure 2 Marker-Based Genetic Encoding: a chromosome consisting of integers is interpreted as a neural network. A start marker is any number that modulo *k* results in a remainder of 1 and an end marker is any number that modulo *k* results in a remainder of 2. *k* is typically chosen somewhere between 5 and 15.

dimensional space, and is compared to a randomly chosen neighbor within a maximum distance k, usually around four places. The individual with the higher fitness is selected for mutation and crossover. Selection amongst local neighbors in the population ensures that a good solution will not take over the whole population but only a local part, and thus spread slowly through the population. Given the population size s_p a fit individual will therefore take about $s_p/(k/2)$ on average to take over the population, therefore a higher diversity is guaranteed and the population can contain multiple different solutions with high fitness in different neighborhoods of the array.

2.3.2 Fitness Evaluation

Outbound and homebound phases. In order to evaluate the agent's fitness, the agent must visit waypoints on an outbound path and then its ability to get back to the starting point is evaluated. In order to simulate an outbound journey, two random points are chosen and the agent is moved through these beacons. During this phase the sensors provide input to the path integration network, but the network's output to the effectors is ignored. During the homing phase, the agent is moved according to its *turn effectors* as activated by the network.

For every generation a set of random waypoints for the outward path is chosen. In order to compare fitness between generations, every few generations fitness is evaluated on a fixed set of routes that do not change between generations. This gives an insight into how the objective fitness of the agents is developing.

Basic fitness function. Fitness is measured by determining how well an agent manages to navigate back to the origin. An agent is evaluated until a maximum number of time steps m is reached. At each cycle the distance to the origin $dist_t$ is measured. The fitness is simply the inverse of the squared distances:

$$\text{fitness} = \frac{1}{\sqrt{\int_0^m dist_t^2 \,\mathrm{d}t}} \qquad (6)$$

Penalizing spiraling. Although Equation 6 evolves valid solutions, the agents often tend to move toward their goal on a spiraling path (cf. Vickerstaff &

Di Paolo, 2005). Although an agent using a direct path would globally score higher fitness values, the GA seemed to get stuck in local maxima that did not allow it to move on to non-spiraling solutions easily. A possible reason for its quick discovery of such solutions is that it is the simplest form of goal-based navigation with the available set of sensors and effectors. The agent will be able to move towards its goal using only one turning effector. A commonly evolved, simple strategy is to compare the homing vector with the current compass direction (simple approximation of dot-product) and, if they point in the same direction, switch off the turn effector. Otherwise, the turn effector is switched on.

In order to penalize spiraling behavior, we add a penalty proportional to the amount of time the agent is pointing away from the origin at each time step. A direct route, with the agent always facing toward its goal, is therefore rewarded. Equation 7 takes this into account by subtracting the nest heading ω from the agent's heading θ and adding it as a penalty factor under the integral:

$$\text{fitness} = \frac{1}{\sqrt{\int_0^m dist_t^2 |\theta - \omega| \mathrm{d}t}} \tag{7}$$

Rewarding simple neural network structures. A typical evolved genome will contain many unnecessary neurons and connections that have little influence on the navigational behavior. In order to eliminate this problem and promote compactness of solutions, each neuron and connection comes at a small cost of fitness. This could be seen as an energy usage on a per cell basis. The more unnecessary cells a network is composed of, the more energy a network requires, consequently lowering the fitness. Equation 8 includes a per cell penalty, k_n is a neuron penalty constant, k_c is a connection penalty constant, c_n is the neuron count and c_c is the connection count.

$$\text{fitness} = \frac{1}{(1 + k_\mathrm{n})^{c_\mathrm{n}}(1 + k_\mathrm{c})^{c_c}\sqrt{\int_0^m dist_t^2 |\theta - \omega| \mathrm{d}t}} \tag{8}$$

A two-stage evolutionary process. The GA was first run using the fitness function that does not reward simple neural networks, that is, Equation 7. When the GA stabilized, the best solution was subsequently used as the seed for a new run of the GA, this time rewarding simple neural structures over complex ones. This pruned the unnecessary and redundant connections from the solution. It was found that this two-stage evolution was more effective in finding solutions than trying to evolve for both successful homing and simple networks simultaneously. Such a two-stage process is reminiscent of multi-objective evolutionary optimization processes (e.g., Brown & Smith, 2005), capable of evolving more effective behavior and of scaling up to harder tasks than is possible by direct evolution (Gomez & Miikkulainen, 1997; Whiteson, Kohl, Miikkulainen, & Stone, 2005).

Reducing the freedom of the network topology. Initial trials found no acceptable solutions, even when letting the GA run for a very long time. The GA would reach local maxima and show hardly any improvement over time after that. This could be the result of the genetic encoding not allowing much change to the network topology without a momentary decline of fitness.

In order to tackle the problem, the search space was reduced by predefining certain topological aspects of the neural network structures. The following constraints were set on the network topology:

- each direction cell had to be connected to at least one memory neuron;
- each direction cell had to be connected to at least one sigmoid neuron;
- sigmoid neurons had to be connected to turn effectors;
- a maximum of two turn effectors, one left and one right were allowed.

3 Results

3.1 The Evolved Network

A fit solution that was consistently evolved is depicted in Figure 3. This solution was obtained by a multistage evolutionary process, as described above, making it more uniform by removing spiraling behavior and any low weight connections or redundant neurons. A typical simulation run is shown in Figure 4.

It is reasonably easy to understand how the network works, by analyzing its activity in various situa-

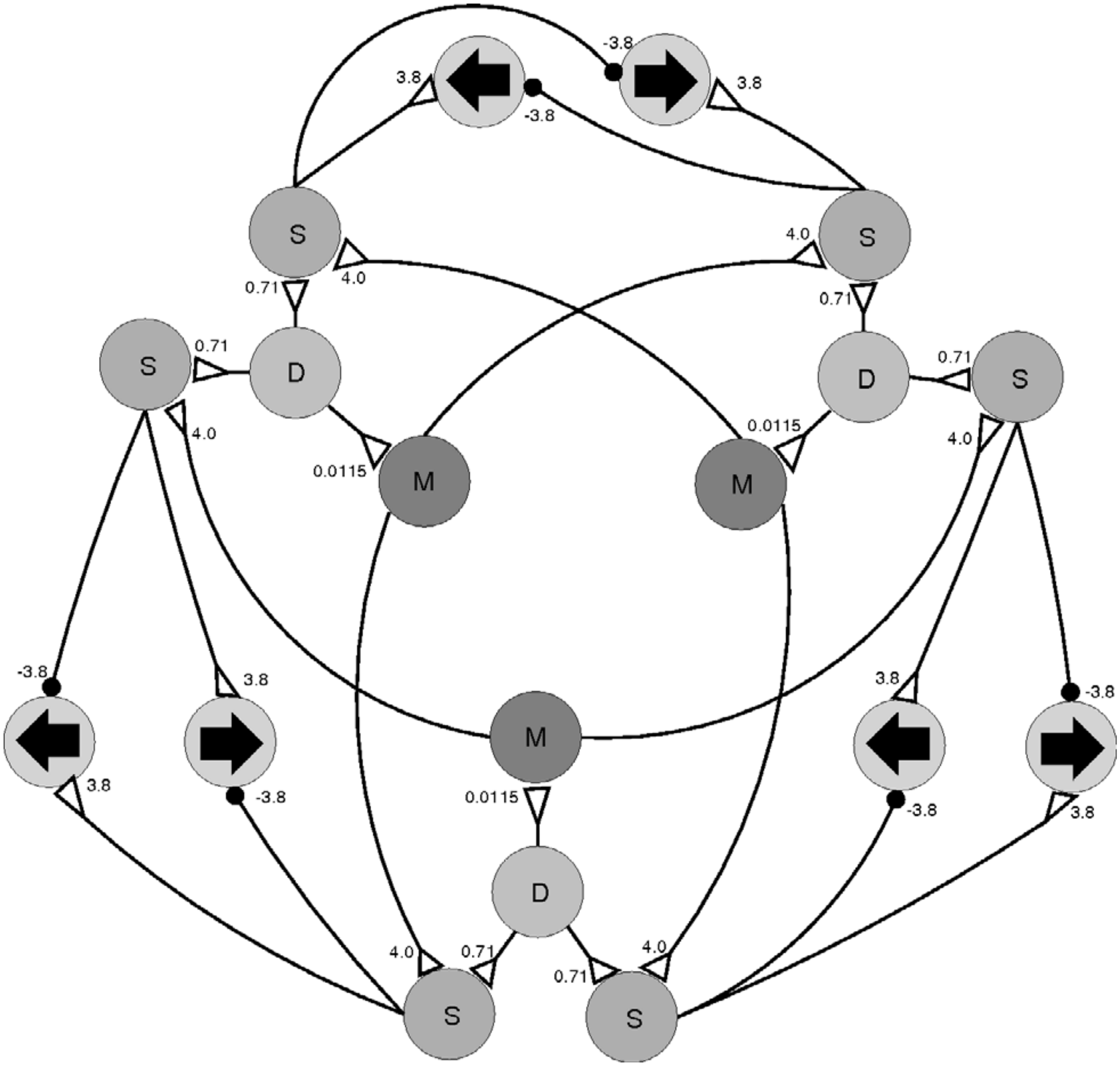


Figure 3 Evolved network using direction (D) cells. Each direction cell has one memory (M) CTRNN cell and two sigmoid (S) neurons associated with it. For illustration purposes, the turn motors (represented by the left and right arrows) are duplicated and the recurrent connections of the memory neurons onto themselves are omitted.

tions, as shown in Figure 5. Each direction cell is directly connected via an excitatory connection to a memory neuron. As a result, by integrating its input, the memory neuron records the distance traveled in the direction signaled by that direction cell. The population of memory neurons therefore encodes the homing vector in a very natural way, similar to the way the population of direction cells encodes the current heading direction. Each direction cell and each memory cell is also connected to two sigmoid neurons, which in turn connect to the turning effectors. These thus become active only if both the direction cell and at least one of the memory neurons are active. During homing, the memory neurons thus act like repelling forces, turning the agent away from the direction that it traveled on its outbound path.

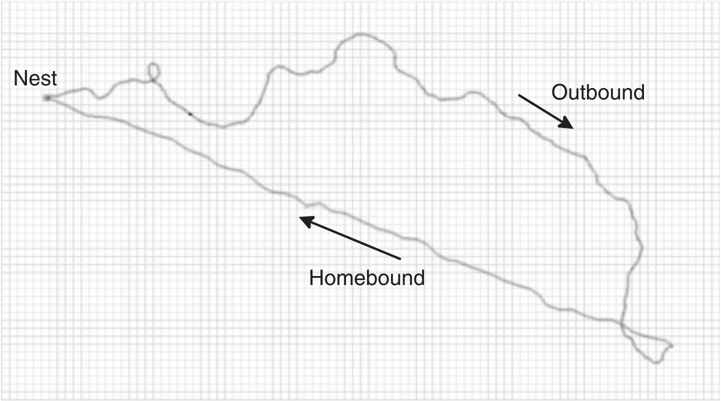


Figure 4 Example simulation run.

Table 1 Evolved neuron parameters.

Parameter	Evolved value
a	0.667
b_s	4.372
b_c	1.164
τ	135,518

Table 2 Evolved connection weights.

Weights	Excitatory value	Inhibitory value
$w_{\text{S-left}}$	3.974	3.974
$w_{\text{S-right}}$	3.976	3.976
$w_{\text{POL-S}}$	0.719	–
$w_{\text{POL-M}}$	0.012	–
$w_{\text{M-S}}$	3.962	–
$w_{\text{M-M}}$	10^{-4}	–

The network is somewhat similar to Hartmann and Wehner's neural model, in which the heading to the nest is encoded using a circular neural structure. The difference is that our solution encodes both the distance and the heading to the nest continuously in this circular structure, whereas Hartmann and Wehner encode these separately and store these discretely.

The number of direction cells was fixed at three during initial evolutionary runs. The GA was not able to evolve control networks with significantly larger numbers of direction cells in reasonable time. To help the discovery and evolution of networks with larger numbers of direction cells, the structure of the neural network for three direction cells was used to make a template structure for a network capable of using N direction cells. This was not overly complex, since the evolved structure with three direction cells was uniform and the redundant and low-impact connections had already been pruned. The topography of the neural network was therefore fixed and the genetic algorithm could be used to evolve connection weights and neuron biases; a much more constrained problem. The genome in these trials was not in marker-based encoding, but was simply an array of connection weights and bias values. Another obvious optimization consisted of exploiting the network structure's inherent symmetry by defining the connection weights once for all symmetrical structures of the network. A network with six direction cells can be seen in Figure 6.

3.2 Performance Evaluations

Figure 4 shows that the simulated agent could reliably home after traveling on a long and complex outbound path. Comparisons between agents with different numbers of direction cells were made under noise and no-noise conditions. As shown in Figure 7, performance improved for an increased number of cells. Without noise, the performance leveled off with five to seven cells; with noise there was further improvement as the number increased.

We tested a three-direction-cell network on a Koala[1] robot to validate that the evolved network pro-

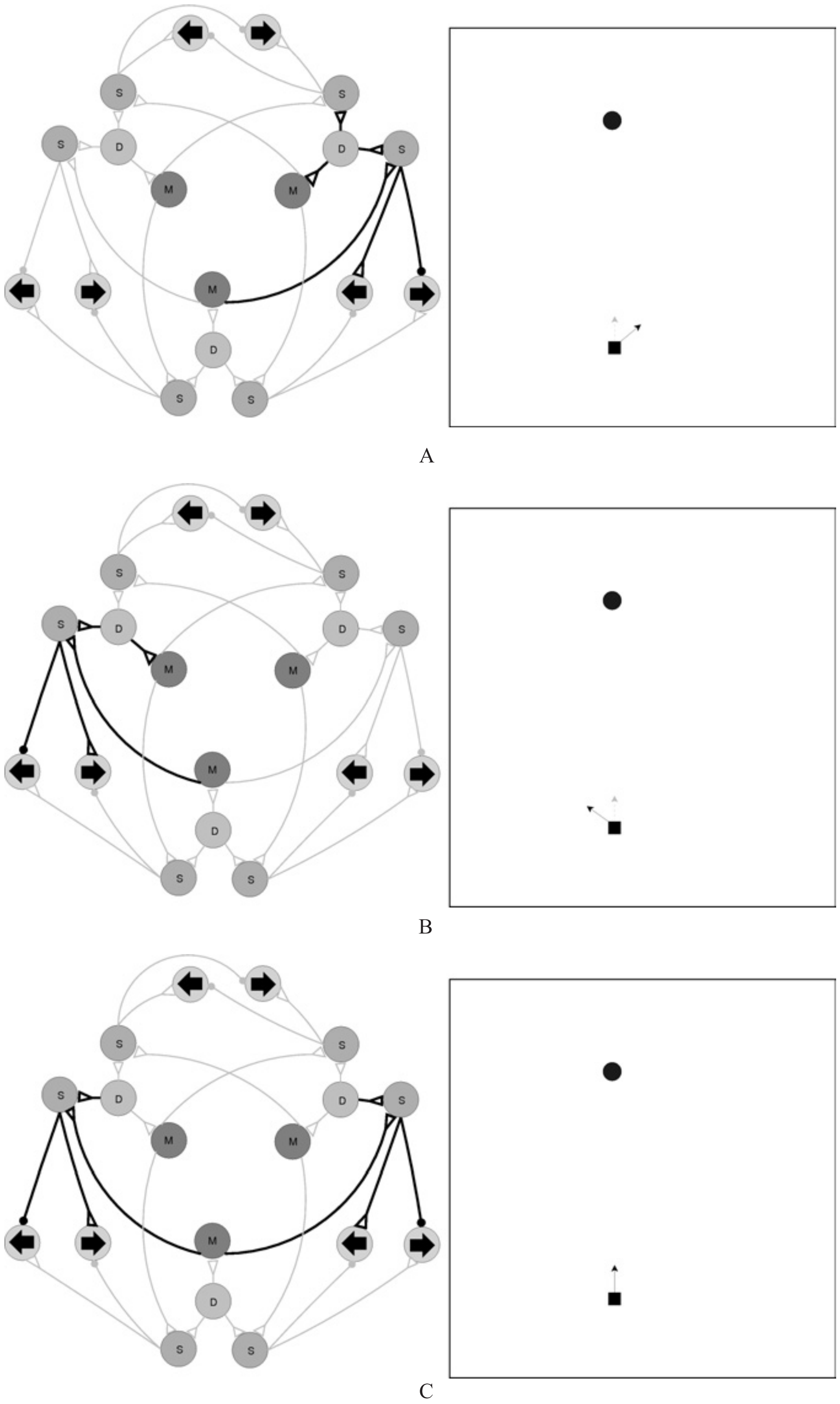


Figure 5 Examples of network activations for different situations. Bold lines indicate strong activations. (A) The agent is heading north-east and the top-right direction cell is strongly active. The sigmoid neuron connected to the left turn motor is the only neuron to receive strong inputs from both a memory neuron and the direction cell, thus causing the agent to turn left towards the nest. (B) The agent is heading north-west. This time, the top-left direction cell as shown in the left part of the figure is active and, in conjunction with the active memory neuron, it causes the agent to turn right. (C) When the agent is pointing towards the nest, the activations of both turn effectors cancel each other out, and the agent moves straight ahead.

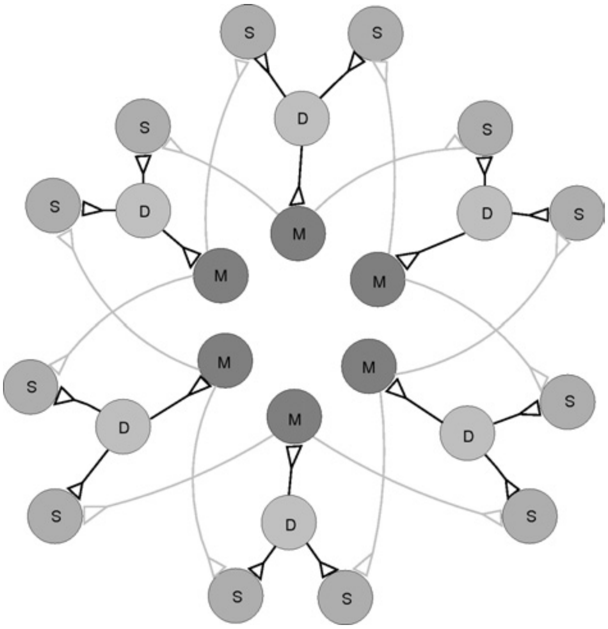


Figure 6 A network with six direction cells. Motors and the recurrent of memory neurons were not drawn. Each sigmoid neuron is connected to either a left or a right turn motor.

vides a plausible real world solution. The Koala is equipped with 16 infra-red (IR) proximity sensors, which are utilized to detect obstacles, and a 22 MHz Motorola 68331 processor with 1 Mb of RAM (as shown in Figure 8). Robot motion is controlled by two DC motors configured in a differential drive set-up. The current heading of the robot is determined by monitoring the changes in wheel encoder readings produced during robot motion. The derived heading is then used to activate the corresponding direction cells appropriately.

The initial experimental procedure produced the outbound journey by defining two beacons through which the robot must pass. After the final beacon was met the robot performed homing using motor commands under the control of the evolved PI network. Figure 9 plots the robot path, overlaid with the defined beacon route for various beacon configurations. Throughout these and all subsequent robot tests the Koala's initial heading was aligned with the positive *X*-axis. Figure 9 shows that the robot passed within close proximity of the beacons before correctly navi-

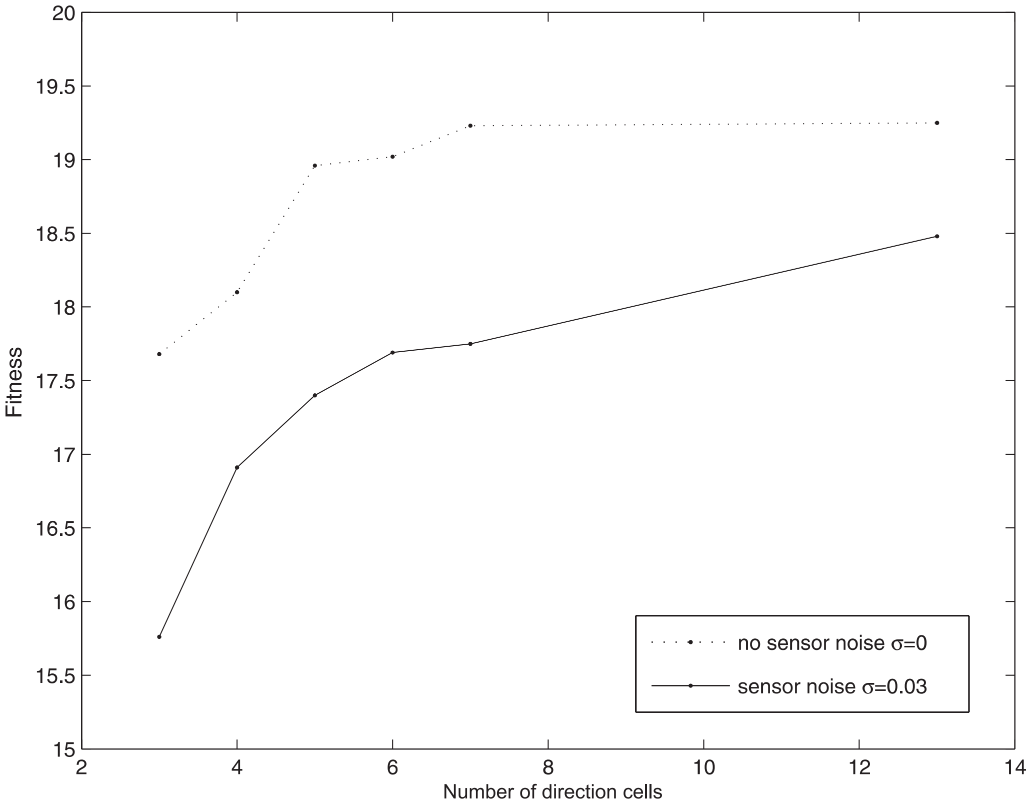


Figure 7 Performance comparison for different numbers of direction cells with noise turned on and off. Standard deviation of performance under noise was near 0.3 for all configurations.

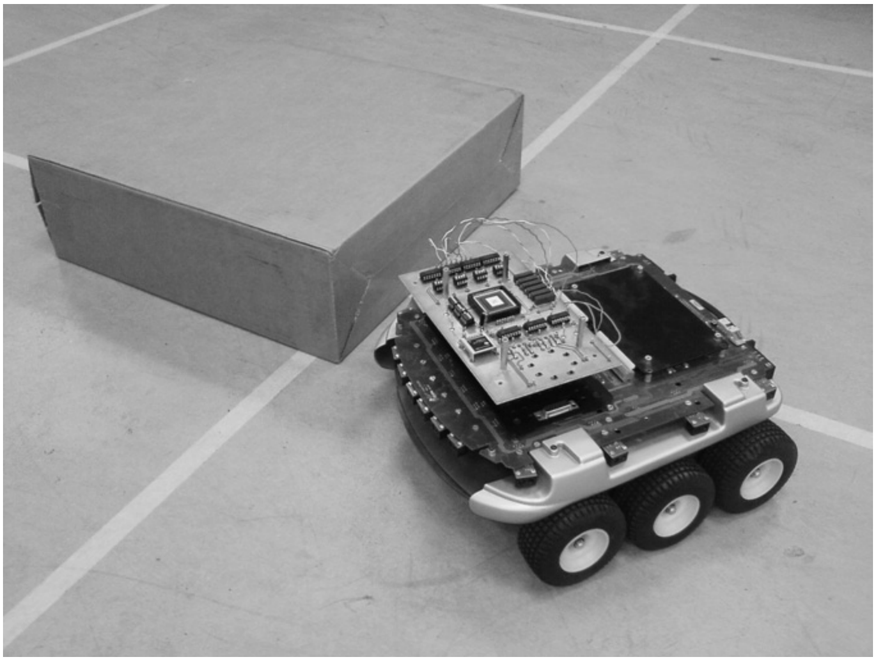


Figure 8 The Koala robot used in the experiments.

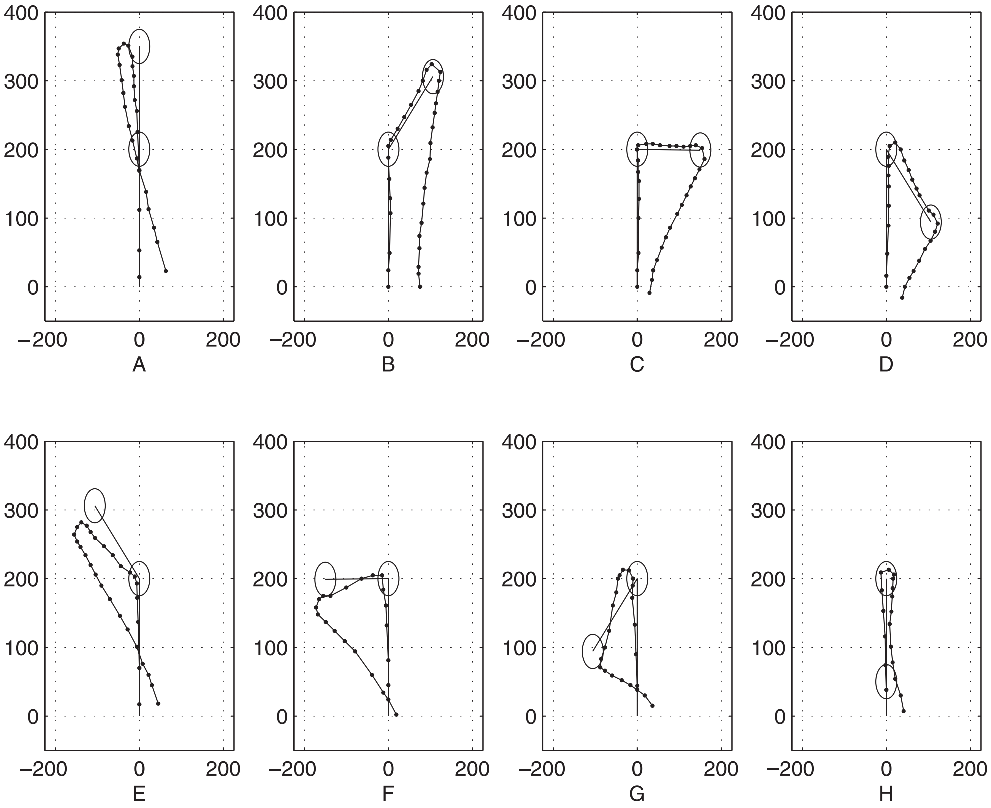


Figure 9 Robot path integration experiments. The home (starting) position of the robot is at (0,0) for all experiments. The positions of the waypoints are highlighted with circles. The mean error over these runs was 46.625 cm with a standard deviation of 18 cm. (A) 69 cm; (B) 75 cm; (C) 31 cm; (D) 43 cm; (E) 51 cm; (F) 21 cm; (G) 41 cm; (H) 42 cm.

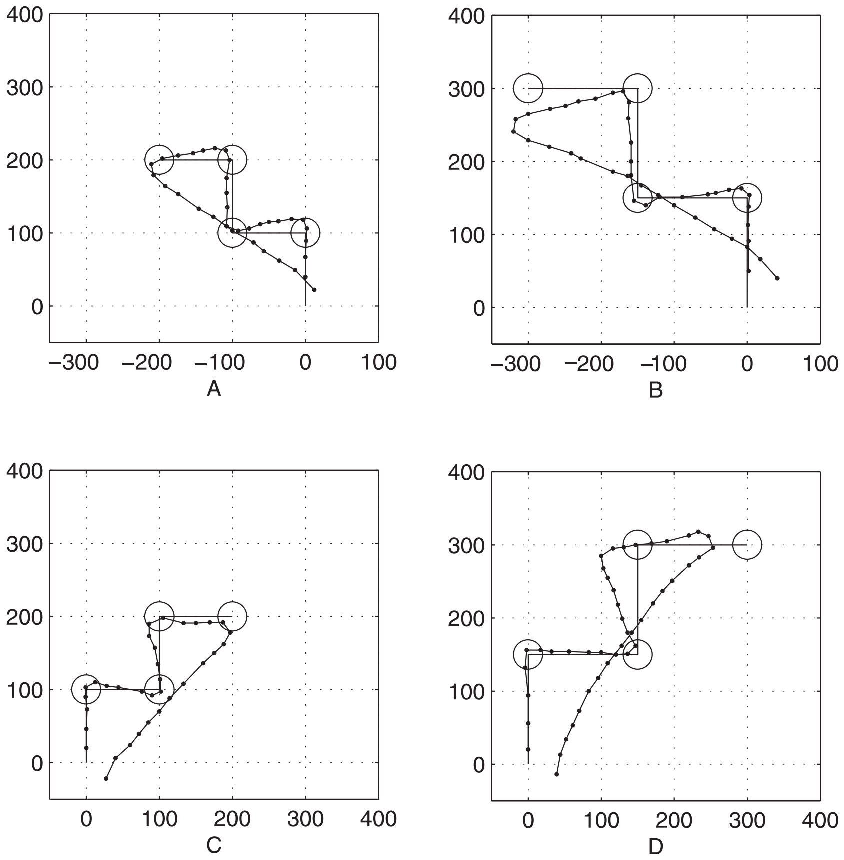


Figure 10 Robot path integration over a zig-zag path of different lengths: (A,C) 4 m and (B,D) 6 m. The errors were (A) 27 cm; (B) 60 cm; (C) 35 cm; and (D) 42 cm.

gating to an endpoint close to the home position for all test configurations. The results also highlight the inherent error encountered when using the wheel encoders to calculate the heading of the robot. Due to an imbalance in the Koala wheel encoders, the robot overshot left turns (cf. Figure 9e–g), whereas it undershot right turns (cf. Figure 9b–d). The calculated error when the wheel encoders were calibrated was 3° per 360°. As the general robot motion consists of a series of small yaw oscillations and forward steps the accumulated error becomes significant.

More complex outbound paths were studied by increasing the number of beacons. Foraging paths more akin to those found in nature were simulated by a zig-zag outbound path. Figure 10 shows that the robot correctly located the home position when subjected to zig-zag patterns of various lengths and directions. Additional paths were defined to test the global vector derivation ability of the network. Figure 11 shows an outbound path measuring 7 m in length, with the final beacon positioned only 1 m from the home position. The robot is again shown to reliably locate the home position. The sequences of three left and three right turns reduced accumulative error caused by the wheel encoders, which resulted in an error of position of 14 cm.

A simple obstacle-avoidance capability was added to the robot using the Koala IR sensors. The algorithm forced the robot to turn away from an obstacle overriding the motor commands issued by the path integration network. The network, however, still received input from the heading and distance sensors throughout the avoidance maneuver, and therefore updated the global vector appropriately. Thus when the robot encountered an obstacle, it first navigated around the obstacle before returning to its homing behavior. The ability of the robot to correctly home following an encounter with an obstacle on the homeward path was tested as shown in Figure 12. The robot skirted the obstacle

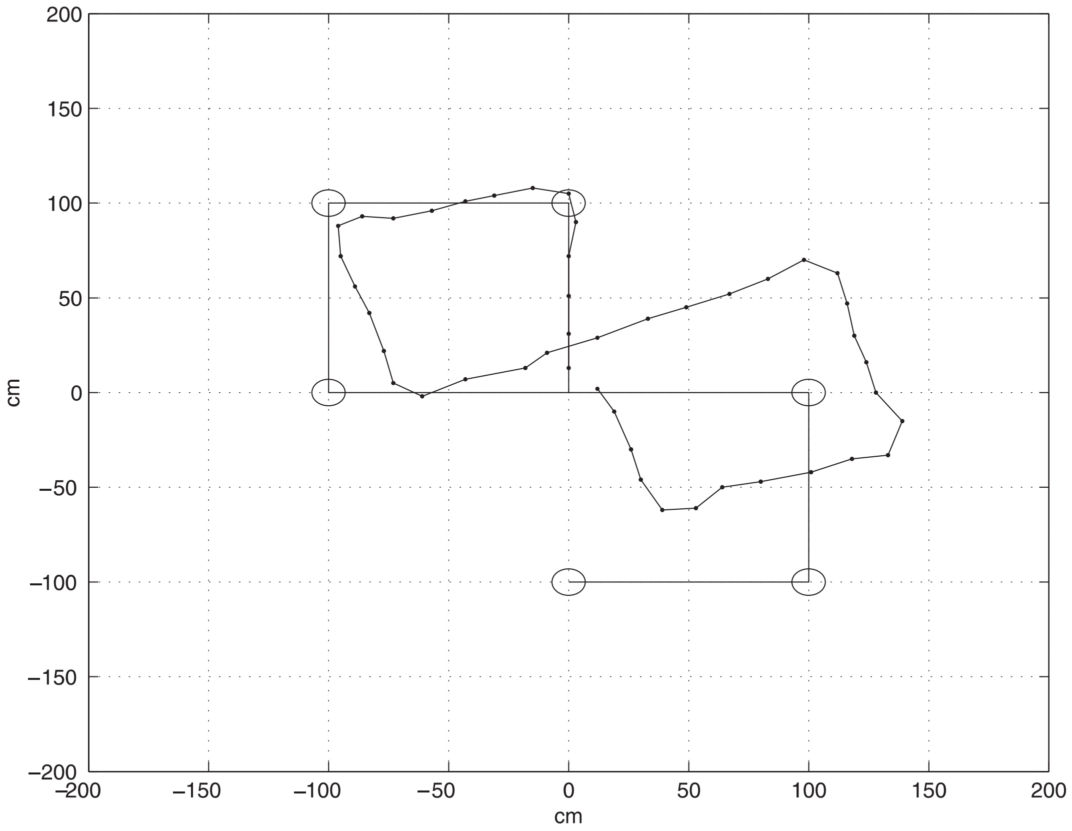


Figure 11 Convoluted path. The error for this run was 14 cm.

before following the altered global vector to the home position. The second obstacle avoidance test placed two staggered obstacles in the homeward path of the robot. Figure 12 shows the robot to correctly avoid both obstacles before following the corrected global vector to the home location. These results show that the network did update the homing vector as it was turning towards the home position after avoiding obstacles. The system accumulates error due to multiple heading adjustments when avoiding obstacles and is at present not highly accurate for large or multiple detours but could be improved with more accurate compass sensors.

4 Discussion and Conclusion

This article has presented a novel neural model of path integration employing biologically plausible direction cells. Through an evolutionary process, we found a compact and elegant network structure that maintains a population encoding of the home vector over a set of memory neurons that integrate the input from the direction cells. This solution works effectively for real-world noise as demonstrated in the robot tests. Although the version described here assumed a constant agent speed, we note that variable speed can be easily handled by weighting the update of the memory cell potential by a term that is directly proportional to the agent's speed.

Different sensory inputs affect the evolution of solutions for path integration. For example, we also tried using the cosine representation used by Vickerstaff and Di Paolo and evolved a similar bi-component network, but both evolution and control seemed unrealistic (Haferlach, 2006). The evolved solutions used memory neurons that would store separate components of the homing vector and thus, the agent did not evolve homing navigation for both components simultaneously. Similarly, homing control was found to function differently when the agent homed from different quadrants. Different parts of the network were

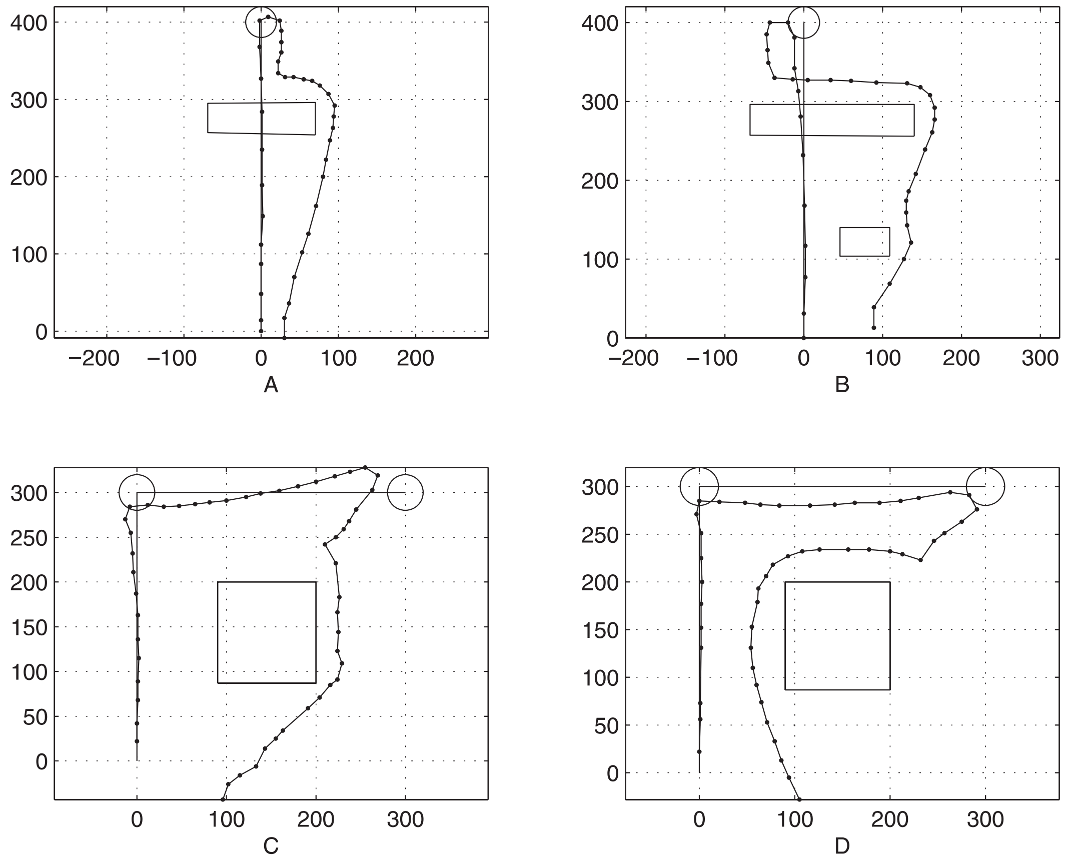


Figure 12 Robot path integration and obstacle avoidance experiments. The obstacles were only placed into the arena after the agent reached the last waypoint and started its homeward journey. The errors were (A) 32 cm; (B) 92 cm; (C) 105 cm; and (D) 108 cm.

active due to the different inputs from the sine/cosine compass sensors when, for example, the agent needed to turn right. The use of direction cells produced more naturalistic evolution and behavior.

The direction cell modeled in this article differs in some ways from natural direction cells in insects, such as the optic lobe *POL* neurons of the field cricket (Labhart & Meyer, 2002). Most importantly, the spiking response of *POL* neurons to e-vector orientation exhibits 180° aliasing. Thus, additional sensory (possibly proprioceptive) signals may be necessary to disambiguate the response and any evolved solutions may therefore differ from the solution presented in this article.

Thus, evolving solutions using realistic polarization sensors (that exhibit 180° aliasing) and testing these with actual polarization sensors on a robot are planned next steps in developing neural models for path integration. It should, however, be noted that the direction cells used in this article have a close resemblance to the response of head direction cells to a single preferred direction in rats (Taube & Bassett, 2003). The involvement of the hippocampus in spatial navigation is probably a common function in all mammals. Our network might provide a plausible link between direction cells and the more abstract spatial information represented in place cells.

The overall biological plausibility of the evolved network is an open question. Investigations into the polarization vision pathways of insects have suggested that the central complex, a neuroarchitecturally distinct neuropil in the insect brain, may be involved in functions of compass orientation and path integration (see Homberg, 2004, for a review). The central complex has a regular neuroarchitecture, which has been said to bear some similarities to models of path integration (cf. Mueller, Homberg, & Kuehn, 1997). It hosts a map-like representation of e-vector orientations (Heinze & Homberg, 2007) and has connections with motor centers in the thoracic ganglia. In some insects it receives, besides polarization-sensitive input, connections from circadian clock neurons, possibly involved in time compensation by adjusting compass bearings in relation to solar azimuth with time of day (Homberg, 2004). Given these developments in insect neuroscience, a future goal is to map more directly our models for path integration to the neuroarchitecture of the central complex.

Acknowledgments

This work was supported by the European Commission under project FP6-2003-IST2-004690 SPARK.

Note

[1] K-team. http://www.k-team.com

References

Benhamou, S. (1997). Path integration by swimming rats. *Animal Behaviour*, *54*, 321–327.

Bisch-Knaden, S., & Wehner, R. (2003). Local vectors in desert ants: context-dependent landmark learning during outbound and homebound runs. *Journal of Comparative Physiology*, *189*, 181–187.

Brown, M., & Smith, R. (2005). Directed multi-objective optimisation. *International Journal of Computers, Systems and Signals*, *6*, 3–17.

Burgess, N., Donnett, J., & O'Keefe, J. (1998). Using a mobile robot to test a model of the rat hippocampus. *Connection Science*, *10*, 291–300.

Collett, M., Collett, T., & Srinivasan, M. (2006). Insect navigation: measuring travel distance across ground and through air. *Current Biology*, *16*, 887–890.

Degris, T., Lacheze, L., Boucheny, C., & Arleo, A. (2004). A spiking neuron model of head-direction cells for robot orientation. In S. Schaal, A. Lispeert, A. Billard, S. Vijayakumar, J. Hallam, & J.-A. Meyer (Eds.), *Proceedings of the Eighth International Conference on the Simulation of Adaptive Behavior* (pp. 255–263). Cambridge, MA: MIT Press.

Egorov, A., Hamam, B., Fransen, E., Hasselmo, M., & Alonso, A. (2002). Graded persistent activity in entorhinal cortex neurons. *Nature*, *14*, 133–134.

Fullmer, B., & Miikkulainen, R. (1992). Using marker-based genetic encoding of neural networks to evolve finite-state behaviour. In F. Varela & P. Bourgine (Eds.), *Toward a Practice of Autonomous Systems. Proceedings of the First European Conference on Articial Life* (pp. 255–262). Cambridge, MA: MIT Press.

Gomez, F., & Miikkulainen, R.(1997). Incremental evolution of complex general behavior. *Adaptive Behavior*, *5*, 317–342.

Haferlach, T. (2006). *Evolving neural networks for path integration* (Tech. Rep./BSc thesis). Edinburgh, Scotland: School of Informatics, University of Edinburgh.

Hartmann, G., & Wehner, R. (1995). The ant's path integration system: a neural architecture. *Biological Cybernetics*, *73*, 483–497.

Heinze, S., & Homberg, U. (2007). Maplike representation of celestial e-vector orientations in the brain of an insect. *Science*, *315*, 995–997.

Homberg, U. (2004). In the search of the sky compass in the insect brain. *Naturwissenschaften*, *91*, 199–208.

Kimchi, T., Etienne, A., & Terkel, J. (2004). A subterranean mammal uses the magnetic compass for path integration. *Proceedings of the National Academy of Sciences USA*, *101*, 1105–1109.

Labhart, T., & Meyer, E. (2002). Neural mechanisms in insect navigation: polarization compass and odometer. *Current Opinion in Neurobiology*, *12*, 707–714.

Lambrinos, D., Moeller, R., Labhart, T., Pfeifer, R., & Wehner, R. (2000). A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, *30*, 39–64.

Matthies, L., Gat, E., Harrison, R., Wilcox, B., Volpe, R., & Litvin, T. (1995). Mars Microrover navigation: performance evaluation and enhancement. *Autonomous Robots*, *2*, 291–311.

Mittelstaedt, H., & Mittelstaedt, M.-L. (1973). Mechanismen der Orientierung ohne richtende Außenreize. *Fortschritte der Zoologie*, *21*, 46–58.

Mittelstaedt, M., & Mittelstaedt, H. (2001). Idiothetic navigation in humans: estimation of path length. *Experimental Brain Research*, *139*, 318–332.

Moller, P., & Goerner, P. (1994). Homing by path integration in the spider *Agalena labyrinthica* Clerck. *Journal of Comparative Physiology A*, *174*, 221–229.

Mueller, M., Homberg, U., & Kuehn, A. (1997). Neuroarchitecture of the lower division of the central body in the brain of the locust (*Schistocerca gregaria*). *Cell Tissue Research*, *288*, 159–176.

Ronacher, B., & Wehner, R. (1995). Desert ants, *Cataglyphis fortis*, use self-induced optic flow to measure distances travelled. *Journal of Comparative Physiology A*, *177*, 21–27.

Seguinot, V., Cattet, J., & Benhamou, S. (1998). Path integration in dogs. *Animal Behaviour*, *55*, 787–797.

Taube, J., & Bassett, J. (2003). Persistent neural activity in head direction cells. *Cerebral Cortex*, *13*, 1162–1172.

Touretzky, D., Redish, A., & Wan, H. (1993). Neural representation of space using sinusoidal arrays. *Neural Computation*, *5*, 869–884.

Vickerstaff, R., & Di Paolo, E. (2005). Evolving neural models of path integration. *Journal of Experimental Biology*, *208*, 3349–3366.

Wehner, R. (1998). The ants celestial compass system: spectral and polarization channels. In M. Lehrer (Ed.), *Orientation and communication in arthropods* (pp. 145–285). Basel: Birkhauser.

Wehner, R. (2003). Desert ant navigation: how miniature brains solve complex tasks. *Journal of Comparative Physiology A*, *189*, 579–588.

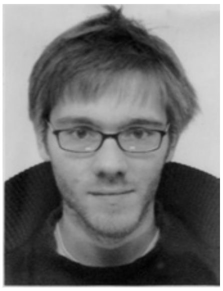
Whiteson, S., Kohl, N., Miikkulainen, R., & Stone, P. (2005). Evolving soccer keepaway players through task decomposition. *Machine Learning*, *59*, 5–30.

Wittlinger, M., Wehner, R., & Wolf, H. (2006). The ant odometer: stepping on stilts and stumps. *Science*, *312*, 1965–1967.

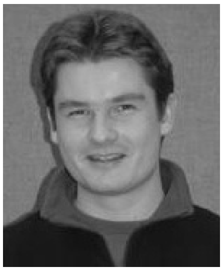
Wittmann, T., & Schwegler, H. (1995). Path integration – a network model. *Biological Cybernetics*, *73*, 569–575.

Wohlgemuth, S., Ronacher, B., & Wehner, R. (2001). Ant odometry in the third dimension. *Nature*, *411*, 795–798.

About the Authors



Thomas Haferlach was born in Germany and earned a BSc in computer science and artificial intelligence at the University of Edinburgh in 2006. His primary research interests lie in the field of biologically inspired computation with a focus on evolutionary algorithms and neural networks. During the summer he extended his BSc thesis, working at the Institute of Perception, Action and Behaviour at the University of Edinburgh. He is planning to begin a PhD in early 2008. *E-mail:* t.haferlach@sms.ed.ac.uk



Jan Wessnitzer is currently a research fellow at the University of Edinburgh investigating insect-inspired control architectures for robotics. He received a MSc in artificial intelligence from the University of Edinburgh in 2000 and a PhD in robotics from the University of the West of England in 2004.

Michael Mangan graduated with an MEng in avionics from the University of Glasgow in 2004. He also received an MSc in neuro-informatics from the University of Edinburgh in 2006. He is currently reading for his PhD at the Institute of Perception, Action and Behaviour at the University of Edinburgh. His research focuses on robotic modeling to aid the understanding of insect navigational strategies. *E-mail:* m.mangan@sms.ed.ac.uk

Barbara Webb received a BSc in psychology from the University of Sydney in 1987 and a PhD in artificial intelligence from the University of Edinburgh in 1993. She has held lectureships at the University of Nottingham and the University of Stirling, and is currently a Reader in Informatics at the University of Edinburgh. Her main research interest is robot modeling of insect sensorimotor systems. *E-mail:* bwebb@inf.ed.ac.uk