The Journal of Experimental Biology 208, 3349-3366
Published by The Company of Biologists 2005
doi:10.1242/jeb.01772

Evolving neural models of path integration

R. J. Vickerstaff* and E. A. Di Paolo

Centre for Computational Neuroscience and Robotics, School of Life Sciences, University of Sussex, Brighton, BN1 9QG, UK

*Author for correspondence (e-mail: robertvi@sussex.ac.uk)

Accepted 29 June 2005

Summary

We use a genetic algorithm to evolve neural models of path integration, with particular emphasis on reproducing the homing behaviour of *Cataglyphis fortis* ants. This is done within the context of a complete model system, including an explicit representation of the animal's movements within its environment. We show that it is possible to produce a neural network without imposing *a priori* any particular system for the internal representation of the animal's home vector. The best evolved network obtained is analysed in detail and is found to resemble the bicomponent model of Mittelstaedt. Because of the presence of leaky integration, the model can reproduce the systematic navigation errors found in desert ants. The model also naturally mimics the searching behaviour that ants perform once they have reached their estimate of the nest location. The results support possible roles for leaky integration and cosine-shaped compass response functions in path integration.

Key words: path integration, *Cataglyphis fortis*, genetic algorithm, leaky integration.

Introduction

Path integration (PI) is part of the navigational repertoire of a wide range of animals, including mole rats (Kimchi et al., 2004), hamsters (Etienne et al., 2004), humans (Mittelstaedt and Mittelstaedt, 2001), crabs (Layne et al., 2003a,b), ants (Wehner, 2003), bees (Dyer et al., 2002) and spiders (Moller and Görner, 1994; Ortega-Escobar, 2002). PI is the same method of navigation that was employed by sailors, under the name of dead reckoning, and functions without the need for landmarks, requiring only a compass and odometer to monitor the path the animal travels. This information is continuously integrated during a journey to give a running estimate of the current position relative to a fixed point, usually the animal's nest or refuge. This accumulated information is referred to as the home vector (HV).

The current paper addresses the possible neural mechanisms responsible for PI and deals in a new way with the question of how the HV may be stored and processed in the brain. We focus on the behaviour of *Cataglyphis fortis* (Forel) ants, whose PI behaviour has been extensively studied (Wehner, 2003) and modelled (e.g. Müller and Wehner, 1988, 1994). Neural models of *C. fortis* PI were presented by Hartmann and Wehner (1995) and Wittmann and Schwegler (1995). Both are based on different methods of encoding the HV chosen by the modeller, since no neurophysiological data are available to suggest the actual mechanism employed. We show that it is possible to produce neural PI models without imposing a particular HV representation in advance, by using a genetic algorithm (GA) to generate networks automatically. We also explore the advantages of producing a complete model (Chiel and Beer, 1997) that includes an explicit representation of the animal's behaviour in its environment.

In common with many other species (Maurer and Séguinot, 1995), systematic PI navigation errors are observed in *C. fortis* when it is forced to walk along L-shaped paths (Müller and Wehner, 1988). Ants leaving the L shape turn too much, and the degree of inaccuracy in the homing direction is a function of the shape and size of the L-shaped outward journey. Errors are not seen when the outward path is straight, unless it is very long (Sommer and Wehner, 2004). Assuming the errors are not an artefact of the experimental manipulation, there are several possible explanations, including at least: (1) the ant has an accurate PI system but deliberately deviates from the direct homeward path under certain conditions, (2) the system only approximates accurate PI, and the degree of error depends on factors such as the journey's size and shape and (3) the ant has a PI system that sometimes operates in an accurate fashion and sometimes does not. In (2) and (3), the properties of the ants' homing errors might be used to infer some information about the PI mechanism, although (3) may give misleading results since the mechanism is changeable. Explanation (1) cannot readily reveal anything about the underlying PI mechanism but can still produce homing errors that vary systematically with the journey shape.

One question is whether, as (1) and (3) above require, our current understanding of information processing in neurons suggests they are capable of performing accurate PI given the resources available in a *C. fortis* ant's brain. If they are not, then the errors show a basic limitation of the ant's brain, and (1) and (3) above are rendered unlikely. If we believe they are capable, we still have options (1), (2) and (3) available: (2) cannot be ruled out since selection may still only have produced an approximate system, provided it is sufficiently good.

Wittmann and Schwegler (1995) present a system that can perform accurate PI using only simple and relatively few model neurons, suggesting that the reason is not a limitation of neural processing. Kim and Hallam's neural PI model supports this conclusion (Kim and Hallam, 2000). The model employs a different mechanism but is also capable of accurate PI. The Wittmann–Schwegler model can reproduce the ants' homing errors if the PI process is assumed to start only after the ant has travelled a certain distance from the nest, making it a type (3) model dependent on when PI begins to operate.

Müller and Wehner (1988) suggest an algorithm that performs an approximation of PI such that, after L-shaped journeys, the ants' errors are reproduced, but under normal foraging conditions navigation is suitably accurate, making it a type (2) model. This represents an hypothesis that the foraging and PI systems have coevolved such that efficient navigation is achieved without the need for exact PI. The mathematical algorithm is argued to be more simple than an exact one. Hartmann and Wehner (1995) present a neural network PI model that shares the property of only producing observable errors after certain journey shapes. However, now that the system is built using neurons, the model is no longer any simpler than an exact PI system (Wittmann and Schwegler, 1995).

Mittelstaedt's bicomponent PI model is a mathematical description of accurate PI and, as such, does not generate systematic errors (Mittelstaedt, 1962, 1985). This feature has been criticised (Hartmann and Wehner, 1995), since the model cannot explain systematic errors or produce any deviations from accurate navigation. This model has not previously been implemented using model neurons. The results presented here show that a neural implementation is possible and that the ants' systematic navigation errors are then generated in a straightforward way using it. Our model has an integrator time constant parameter that can be tuned to accurate or erroneous PI, making it a type (3) model.

Modelling path integration

Mathematically, PI on a flat two-dimensional surface is easily expressed in terms of the operations required to update the HV (Benhamou and Séguinot, 1995; Maurer and Séguinot, 1995) and can be expressed in several alternative forms (see Fig. 1), including using a rectangular (Cartesian) or polar coordinate system and using an egocentric or a geocentric system. For example, Mittelstaedt's bicomponent model (Mittelstaedt, 1985) is in rectangular geocentric form:

$$\mathrm{d}x/\mathrm{d}t = s\cos\theta\ ,$$
$$\mathrm{d}y/\mathrm{d}t = s\sin\theta\ , \qquad (1)$$

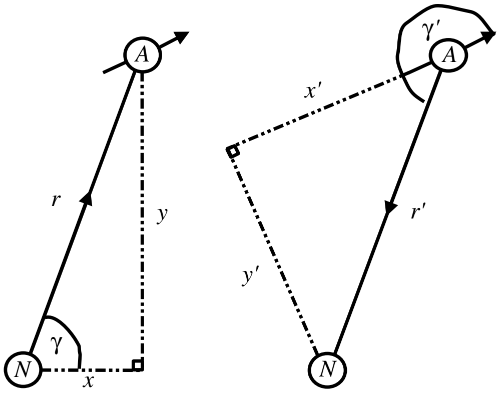

Fig. 1. The four main classes of possible home vector. Left, geocentric rectangular (x,y) and polar (r,γ); right, egocentric rectangular (x',y') and polar (r',γ'). N is the nest position, A is the animal's position, the short arrow indicating its orientation. We use the convention of measuring angles as positive anti-clockwise from the x-axis or from the animal's body axis.

where s is the animal's speed, θ is its compass heading, (x,y) is its geocentric rectangular HV and t is time. In polar geocentric form, we have:

$$\mathrm{d}r/\mathrm{d}t = s\cos(\theta-\gamma)\ ,$$
$$\mathrm{d}\gamma/\mathrm{d}t = (s/r)\sin(\theta-\gamma)\ , \qquad (2)$$

where r is the animal's distance and γ is its bearing from the origin. Under a geocentric system, the HV expresses the animal's position relative to a fixed reference point (e.g. its nest) that forms the origin of the coordinate system. An egocentric system expresses the reference point's coordinates relative to the animal's position and orientation, such that the HV changes even if the animal is rotating on the spot (see Fig. 1).

Homing

Many species use PI as a means of returning as rapidly as possible to their nest or refuge. In an open environment, this naturally corresponds to a straight path home from the current location. This can be formalised using an equation describing the animal's rate and direction of rotation during homing as a function of the HV. The form of this equation depends on the type of HV used. Mittelstaedt's bicomponent model (Mittelstaedt, 1962, 1985), a geocentric rectangular description, uses:

$$\mathrm{d}\theta/\mathrm{d}t = (x\sin\theta) - (y\cos\theta)\ . \qquad (3)$$

We can plot $x\sin\theta$ and $y\cos\theta$ against θ, holding (x,y) constant (see Fig. 2A). dθ/dt will be positive where $x\sin\theta>y\cos\theta$ and negative where $x\sin\theta<y\cos\theta$, telling us whether the animal will turn anticlockwise or clockwise, respectively, and shows that there are two equilibrium values of θ provided the HV is not (0,0). One equilibrium is stable and points towards home, the other is unstable and points directly away from home. Fig. 2A also shows that the animal will always turn to face home efficiently, i.e. through the smallest of the two

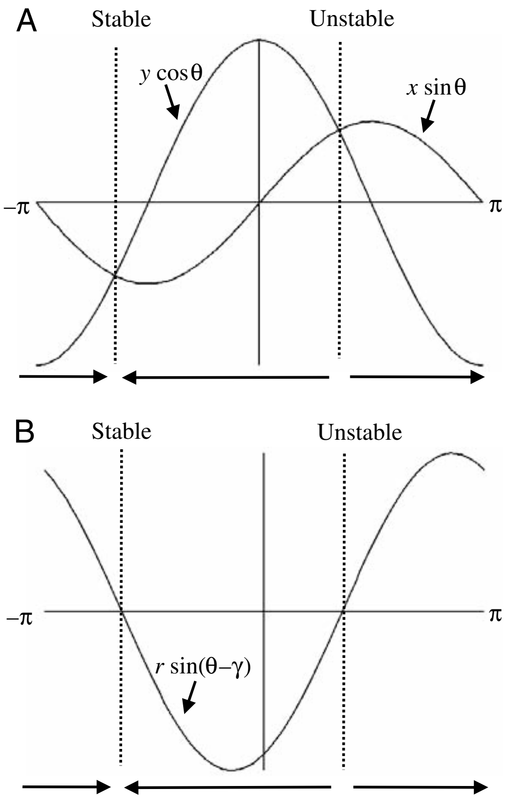

Fig. 2. Dynamics of homing using (A) Eqn 3 and (B) Eqn 4. The horizontal axis shows the animal's orientation, θ; the vertical axis shows the animal's rate of turn dθ/d*t*, where (*x*,*y*) is the geocentric rectangular home vector (HV) and (*r*,γ) is the geocentric polar HV (see Fig. 1). This example shows the case where (A) $x>0$, $y=2x$ and, in equivalence, (B) $\gamma=\tan^{-1}2=1.107$. Both schemes lead to one stable and one unstable equilibrium heading with the same respective values.

possible angles. The equivalent description using a geocentric polar HV (see Fig. 2B), which is very similar to the scheme used by Hartmann and Wehner (1995), is:

$$d\theta/dt = r\sin(\theta-\gamma)\ . \qquad (4)$$

Although these formal descriptions may be of some help classifying any proposed neural implementation of the same process, several caveats should be borne in mind. Firstly, the simplicity of a mathematical description does not guarantee simplicity when the same processes are carried out or approximated by neurons (Maurer and Séguinot, 1995). Secondly, even the classes of description may change when implementing a model neurally. Wittmann and Schwegler (1995) note how their geocentric polar HV, represented in their network using a sinusoidal array (see below), becomes equivalent to Mittelstaedt's geocentric Cartesian bicomponent model (Mittelstaedt, 1985) for a specific set of parameters.

The mathematical description of homing is dependent on the form of the HV; similarly, the neural homing mechanisms will be dependent on the neural HV representation. A parsimonious neural PI model should therefore give equal weight to homing as to updating the HV. A feature of *C. fortis* PI is that the HV continues to be updated during the homeward journey, allowing the animal to detour around obstacles (Schmidt et al., 1992). Thus, HV updating and homing together constitute a feedback process. A description of the behaviour of the animal is not properly defined until both systems are in place. The present paper is concerned with producing a complete description of this type. The task consists of an arena devoid of landmarks (in keeping with the salt pan habitat of *C. fortis*), such that PI is the only available strategy for homing. The agent's (the model animal's) locomotion is restricted to the usual foraging behaviour of the ant, i.e. walking only forwards, not sideways or backwards, and employing the same class of sensory cues, namely an allothetic compass (Wehner, 1994) and idiothetic odometer.

Existing neural models

We now briefly introduce two existing neural models of *C. fortis* PI behaviour.

The Hartmann–Wehner model

This model (Hartmann and Wehner, 1995) uses neural chains to store and manipulate the HV. A neural chain is a linear or circular chain of neurons whose pattern of firing represents a linear or circular variable, respectively, in this case primarily the distance and angular components of a geocentric polar HV. Each chain neuron has two supporting neurons to allow the activity pattern to be stabilised or modified. The distance chain (called the *r*-chain) works like a thermometer. Each chain neuron can be either fully activated or fully inactivated. Activity spreads from one end of the chain to the other as the value increases, and recedes back again as it decreases. Only the position of the boundary of the active region encodes information. The angular chain (called the ν-chain) works similarly, except that the chain is circular and contains a group of adjacent active neurons. The number of active neurons is fixed, but the activity can shift clockwise or anticlockwise around the circle to represent changes to the stored value. The model has five more circular chains. One, the λ-chain, represents the ant's current compass heading. The final four chains, called C^+, C^-, S^+ and S^-, are needed to calculate the value of $\lambda-\nu$ and apply approximations of the trigonometric functions cosine and sine to them. Chains S^+ and S^- provide a homing signal. For further details, see Hartmann and Wehner (1995).

The model reproduces the systematic navigation errors displayed by *C. fortis* (Müller and Wehner, 1988) by nature of the approximate PI mechanism it instantiates. Nine free parameters of the model, controlling the relative phases and widths of the activity regions on chains ν and λ, and thus the approximation of cos and sin functions, are used to fit the model to the data. It is also possible to make the model produce more accurate PI behaviour by choosing different values for these parameters (Wittmann and Schwegler, 1995).

Since the model separates the polar HV into distance and angular components, a potential discontinuity in ν (the angular HV component) arises at the HV zero point: if the animal were to pass over its estimate of the home location, the angular component of the HV would have to jump by 180°. This might cause problems if the network were employed to model the ant's systematic search behaviour, since PI must continue during searching, which involves regularly returning to the HV zero point (Wehner and Srinivasan, 1981).

Chapman (1998) reports implementing and testing the model on a mobile robot, using approximately 900 simulated neurons. Flaws in the *r*-chain and homing mechanisms are reported. After modifying the model to correct these flaws, the robot was able to navigate successfully. This result shows that the basic mechanisms employed by the model are valid and can accomplish PI outside of a simulation.

The Wittmann–Schwegler model

This model (Wittmann and Schwegler, 1995) uses a sinusoidal array to represent its geocentric polar HV as a phasor. The distance and angular HV components [called (r,ϕ)] specify the amplitude and phase, respectively, of a sinusoidal wave, which is represented spatially along a circular neuronal array. Using N array neurons, the activity of neuron i is $k_0[r\cos(\phi+2\pi i/N)]+b_0$, where k_0 and b_0 are positive constants. The animal's compass heading is assumed to be available as a symmetrical single-peaked activity pattern on another circular array of neurons. Connections from the compass array to the HV array, modulated by the animal's current speed, convert the compass information into the same phasor representation as the HV. Simple element-wise addition is then enough to update the HV correctly. Recurrent connections within the HV array act as a memory and stabilise the shape of the sine wave. Homing is achieved by comparison of the current compass and HV arrays. For full details, see Wittmann and Schwegler (1995).

The ease of performing vector addition with a sinusoidal array is stated to justify its usage in PI. Touretzky et al. (1993) implement a sinusoidal array using a spiking neuron model. The array is a redundant encoding where each unit of the array is made up of 100 spiking neurons, where a high precision can be attained even if each neuron has relatively few distinguishable firing rates and is subject to noise. However Touretzky et al. state that the sinusoidal array has no advantage over a Cartesian encoding unless its ability to perform vector rotation is used, which the Wittmann–Schwegler model does not employ.

Like the Hartmann–Wehner model, Wittmann and Schwegler fit their model to the data from *C. fortis* (Müller and Wehner, 1988). The fit is visually as good as the former model's. Since the network performs exact, error-free PI, an extra parameter and mechanism were introduced specifically to produce errors. Whilst fitting to data using a single parameter is arguably stronger evidence for a model than fitting with nine parameters, the model does not generate navigation errors as a result of its fundamental PI mechanism, as the Hartmann–Wehner model does.

Evolving neural models

Both the Hartmann–Wehner and Wittmann–Schwegler models are built around neural representations of the HV that were chosen on the basis of robustness. The HV representation cannot be based on experimental findings, since no relevant data exist. The big differences between the models, and the relatively large number of neurons used, suggest that many such PI networks of similar competence could be constructed using various mechanisms.

We approach the problem of the neuronal mechanisms of PI here without making any *a priori* assumptions about the way the HV is encoded but rather use an automated network generation technique, the GA, to explore a set of possible networks that enable an agent to home using PI after an excursion. This procedure permits a less-biased exploration of possible neural mechanisms and also has other advantageous features for system design. We must fully and explicitly specify the sensory and locomotory capabilities of the agent, the information-processing capabilities of individual neurons and synapses, the sources of noise present and the behaviours we wish the agent to display before we can use the GA. The simulation code must explicitly generate the behaviour of each candidate PI model and agent in order to evaluate it. This reduces the chance of producing a flawed or fragile model. The GA is also able to prune the evolving networks to favour simplicity of neural structures. Finally, we can readily have the GA produce models under variations of the assumptions, such as the class of network model or the nature of the animal's internal compass representation.

The GA is not intended to closely model evolution by natural selection but it does capture one feature of PI systems, namely the potential chicken-and-egg problem of HV maintenance and HV-mediated homing. Unless both of these capabilities arise at the same time there is no selective advantage for either in isolation, at least in terms of PI navigation. A human-designed system need not suffer from this problem since both systems are constructed simultaneously and in their full complexity. The GA must develop and improve both capabilities at the same time for the system to operate, resulting potentially in a better mutual adjustment between them.

Evolved networks are often too complicated to understand using purely analytical methods. Consequently, we may not be able to concisely state the behavioural properties of an evolved PI system and may have to take a more investigative approach, similar to that used in real-world biological systems (Peck, 2004; Di Paolo et al., 2000).

Neural mechanisms

The goldfish oculomotor neural integrator (Major et al., 2004) is a mechanism that shares some properties with the HV in PI and is better neurophysiologically characterised. These neural integrators perform temporal integration of incoming velocity signals to track the position of the fish's eyes. It is not yet known whether this is carried out at the single neuron level (Shen, 1989; Loewenstein and Sompolinsky, 2003) or using carefully tuned positive feedback within a recurrent neural circuit (Seung, 1996). Whatever the neural basis, functionally the integrator can display leaky, stable or unstable behaviours (Major et al., 2004). This suggests employing neural models capable of readily evolving similar behaviour (at least as a subset of their behaviour) in our PI investigations. Recent experimental findings (Egorov et al., 2002) show that single entorhinal cortex neurons are capable of performing temporal integration of their inputs, expressed as persistent graded changes in firing rate, at least over the course of tens of seconds. Here, we use a standard leaky integrator firing rate model, the continuous time recurrent neural network (CTRNN; Beer, 1990; Dayan and Abbott, 2001; Koch, 1999). Unlike traditional connectionist models, each neuron works in continuous time and responds relatively faster or slower depending on its time constant parameter.

Since CTRNN networks cannot easily perform multiplication, we also use an augmented version, the modified CTRNN (ModCTRNN), introduced here for the first time, which allows multiplication *via* changes in synaptic strengths. Multiplication may be an important mechanism for PI. Both the Hartmann–Wehner and Wittmann–Schwegler models used multiplicative or gating synapses to adapt PI to variations in the animal's speed. In his bicomponent model of PI, Mittelstaedt (1962, 1985) used multiplication to generate homing behaviour. Multiplication has been conjectured (Koch, 1999; Salinas and Abbott, 1996) and observed (Hatsopoulos et al., 1995; Gabbiani et al., 1999; Anzai et al., 1999; Andersen et al., 1985) to be carried out by many neural mechanisms. We investigate whether the ModCTRNN model results in a significant decrease in overall network complexity or an increase in performance over the CTRNN model.

Materials and methods

Genetic algorithm

A GA (Goldberg, 1989) with a population of 30 genotypes was used, each encoding a neural network. Each genotype was evaluated in 10 independent trials per generation. Each trial, the agent, controlled by the encoded network, was required to perform the PI tas, and was assigned a fitness value. The genotype's overall fitness was the mean fitness over the 10 trials. The fittest five genotypes were retained unmodified in the population each generation. Each was also copied five times to produce 25 new genotypes, which were mutated and used to replace the 25 least fit genotypes. As well as mutating the parameter values of the network, the GA could also change the number of neurons and synapses, referred to as links here, present in the network (see Appendix 1 for more details). There is no special significance in this selection scheme and we expect a number of similar GA schemes to work just as well.

The path integration task

For each trial, the agent started at the nest with a random orientation and was presented with a series of between one and three visual beacons that it was required to visit (defined as approaching to a distance of 0.01 or less) by phototaxis. Each beacon was immediately removed when the agent reached it, and the next one activated. Beacons were placed by selecting a random distance from the nest in the range [0.5,1.0] and random angle from the nest in the range [0,2π] radians, except the last beacon's angle, which was selected using a stratified random scheme that divided [0,2π] into 10 equal-sized blocks, one for each trial, thus ensuring a more even coverage of final HVs per evaluation. After the last beacon was removed, the agent's orientation was randomised (to prevent it homing by simply turning through π radians for a one-beacon trial) and it was required to return to the nest (again defined as approaching within 0.01 distance units) using only its compass sensors, speed sensor and internal state, i.e. the nest could not be directly detected by the agent. During a given generation, all 30 agents were presented with the same 10 sets of beacon locations and given the same initial orientations at the nest and were subjected to identical noise. This made fitness comparisons less subject to noise.

Each experiment was initially performed with the agent's speed fixed at maximum, so that a solution could evolve without requiring the speed sensor. If the GA produced a satisfactory solution, the experiment was repeated with the agent's speed no longer restricted to maximum. The agent was now also held captive (stationary) for a randomly selected time drawn from [0,0.5] upon reaching the last beacon before being allowed to attempt homing. If this task was solved, the experiment was continued with an increase in the level of noise on the motor outputs (see below), so forcing the agent to take full account of the speed sensor in order to perform accurate PI.

Calculating fitness

The fitness assigned for each trial is calculated by one of four equations depending on how many phases of the task were completed by the agent. The overall fitness is always between zero and one, and the fitness for reaching each new phase is equivalent to achieving optimal fitness in all previous phases. Firstly, if the agent failed to reach the first beacon within twice the time it would have taken to visit all of the trial's beacons by a direct course at maximum speed the trial was ended and the fitness was $0.25/(1+\epsilon_B)$, where ϵ_B is the time integral of the agent's Euclidean distance from the first beacon over the trial (a measure of the average distance from the beacon). Secondly, if the agent reached the first beacon but failed to reach all subsequent beacons within the above time limit, the trial was ended and the fitness was $0.25+0.25\times \mathit{visited}/\mathit{total}$. These two criteria simply facilitate the evolution of phototaxis. Thirdly, if the agent visited all beacons but failed to reach the nest within three times the minimum time required from the location of the last beacon at full speed, the trial ended and the fitness was $0.5+0.25/(1+\epsilon_N)$, where ϵ_N is the integral of the Euclidean distance of the agent from the nest starting from the moment it arrived at the last beacon until the end of the trial. If the agent returned to the nest, its fitness was $0.75+0.25/(1+t_N)$, where t_N is the time taken to complete the entire trial; thus, the fittest possible agent would visit all beacons and then return to the nest using straight, direct paths for all legs of the journey and travel at full speed. Owing to sensor noise, cumulative navigation errors would arise, meaning that an ideal agent would also search efficiently for the nest after reaching the HV zero point until it found the nest or until the trial timed out.

The agent

The agent was modelled as having a position (x_A,y_A) on an unbounded two-dimensional plane, with orientation θ_A radians measured positive anticlockwise from the *x*-axis (or 'east'; all angles are defined positive anticlockwise from the *x*-axis or sometimes from the agent's body axis). The origin (0,0) of the plane was defined as the nest location. One distance unit was defined as the maximum distance the agent was ever required to travel from the nest. Maximum speed was also unity; the agent therefore took at least one time unit to reach maximum distance from the nest. The agent had two beacon sensors (for phototaxis), two (sometimes more) compass sensors, one speed sensor, one food sensor, two rotation motors and one forward motor. The agent's motion was restricted to rotation and forward translation. No backward or sideways movement was allowed, in keeping with the usual behaviour of foraging ants. The agent was assumed to have low inertia; motor output therefore specified forward speed and rotation directly, rather than supplying forces. Speed was controlled by the forward motor neuron firing rate (firing rate is defined in the next section), *F*, and rotation by the two opposing left and right rotation motor neuron firing rates, R_L and R_R:

$$d\theta_A/dt = 150\ (R_L - R_R)\ , \qquad (5)$$

giving a maximum rate of turn of 150 radians per time unit, and maximum speed of 1 distance unit per time unit. This scheme was chosen, rather than simply simulating a two-wheeled robot, so that the agent's speed could be easily fixed at maximum without preventing it from steering.

The agent's body, consisting of sensors, motors, neurons and links, was constrained to bilateral symmetry. Aside from single-copy components such as the speed sensor and forward motor neuron, all components were created as symmetrical pairs, including all other neurons, sensors and links.

Compass sensor neurons had an activation function, f, which responded maximally to a particular agent orientation, $\theta_A=a$, and declined in a symmetrical way either side of this value. Each pair of compass sensor neurons had complementary values such that if one responded maximally at $\theta_A=a$, the other responded maximal to $\theta_A=-a$. This could be considered as the response of light sensors to a distant light in the east. All activation functions define the stimulus–response properties of the sensory neuron in terms of its firing rate (see next section). The function f was varied between experiments and was mostly based on the cosine function (see Fig. 3). As can be seen, some functions have negative as well as positive firing rates. This unbiological feature can be removed by replacing each compass sensor with two sensors, each giving a complementary half-wave rectified output (see Discussion). Negative values were used to simplify the analysis of the evolved networks. The agent was either given one pair of compass sensors with maximum responses set to $\theta_A=\pm\pi/4$ or had an evolvable number of sensor pairs with evolvable maximum response parameters. The pair of beacon sensor neurons was defined as having activation $[\cos(\theta_B-\pi/2)/2]+0.5$ for the left sensor and $[\cos(\theta_B+\pi/2)/2]+0.5$ for the right, where θ_B is the angle to the current beacon from the agent's body axis (the activation is therefore independent of distance to the beacon). Additionally a 'food' sensor neuron was used: its activation was 0 before reaching the final beacon (where the agent might be imagined to have collected an item of food to motivate its return to the nest) and 1 thereafter (this sensor was ignored by most of the evolved networks). The speed sensor neuron gave a proprioceptive measure of the agent's current speed, including motor noise, linearly mapped to the range [0,1]. Noise was applied to all initial neuron states (v_t=0), sensors and motors by adding a uniformly distributed random offset in the range $[-\eta,\eta]$. The offsets for sensors and motors where held constant across multiple numerical integration steps (see Appendix 1); changes in value were triggered using a Poisson process rate *r* (the average number of events per unit time) to calculate the time of the next value change: $t_{\mathrm{change}}=[-\ln(1-x)/r]+t$, where *x* is a uniformly distributed random value drawn from the interval [0,1], and *t* is the current time. This scheme makes noise effectively independent of the numerical integration step size used. The values of η and *r* could be set for each sensor/motor type; η=0.01 and *r*=20.0 were used initially for all sensors and motors; for some experiments, this was increased part way through for the forward motor neuron to η=0.7, *r*=2.0 and for the rotation motor neurons to η=0.1, *r*=20.0 in order to impose large long-lasting variations in the agent's velocity. Overall, the presence of noise in the simulation is intended to promote the evolution of robust PI solutions and to cause small cumulative navigation errors, as occur in natural PI.

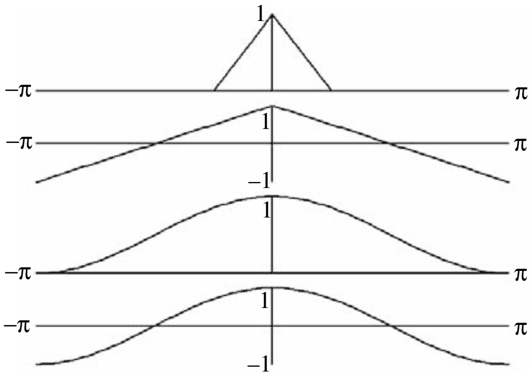

Fig. 3. The four compass activation functions used. From the bottom: cosine, positive cosine, piecewise linear approximation of cosine, head direction cell. Plot of sensor output (*y*-axis) against θ_A–*a* (*x*-axis), where θ_A is the agent's current orientation and *a* is the sensor's preferred direction.

Neural networks

The control network used was either a CTRNN (with fixed weights) or a ModCTRNN (with modifiable weights, allowing multiplication). The overall network architecture was always bilaterally symmetrical but was not otherwise constrained. It was not prevented from forming recurrent connections or forced to be fully connected. The GA fully determined which components were linked to which other components. This had the advantage of allowing a form of stochastic evolutionary pruning to remove redundant components where the GA was left running with its addition operator disabled but deletion operator enabled (see Appendix 1 for details).

The state equation used here for a CTRNN neuron is:

$$\tau_i\,(\mathrm{d}v_i/\mathrm{d}t) = -v_i + \Sigma_j\, w_j z_j\,, \qquad (6)$$

where i indexes all neurons, j indexes all links inputting to neuron i (which may be an empty set), τ_i is a time constant, v_i is the neuron state (analogous to a membrane potential), w_j is the link weight and z_j is the activation of the sensor or neuron attached to link j. For a neuron, $z_j=1/\{1+\exp[-(v_j+b_j)]\}$, where b_j is a bias parameter. For a sensor, z_j is the current activation. The initial value of the membrane potential, $v_{i,t=0}$, was also encoded for each neuron. z represents a dimensionless firing rate, i.e. a firing rate divided by its maximum possible rate. Since the model does not generate explicit spikes, it could alternatively be considered to represent the dimensionless membrane potential of a non-spiking neuron. Synaptic weights, w, in the CTRNN model are fixed, having no dynamics or plasticity of any kind.

The ModCTRNN uses the same state equation for its neurons but also applies an additional analogous leaky integration equation to its weights, changing them from fixed parameters into potential variables. This is achieved by allowing links to point to other links and input a signal that can modify the target link's weight:

$$\alpha_1\,(\mathrm{d}w_i/\mathrm{d}t) = -w_i + \beta_i + \Sigma_j\, w_j z_j\,, \qquad (7)$$

where i indexes all network weights, j indexes all links inputting to weight i (which may be an empty set), α is a time constant, β is a bias term, w_j is a link weight and z_j is the firing rate of the neuron or sensor attached to link j. All weights w_i are initialised to β_i, therefore any weights that receive no inputs remain constant at this value throughout a trial. A link acting to modify the weight of another link can itself be the target of modification, and so on, allowing an arbitrary degree of higher-order weight changes to take place, limited only by the number of links available to the GA. See Appendix 1 for the parameter ranges allowed. The ModCTRNN equation is capable of producing effects similar to synaptic depression or facilitation but should not be considered as a detailed realistic model of synaptic plasticity.

Experiments

Experiment 1A

The agent was given one pair of compass sensors giving a cosine-shaped response (see Fig. 3). The CTRNN model was used. The agent always moved at its maximum speed. Once the agent had evolved to solve the task reliably, the GA was run in pruning mode until the size of the genotypes stabilised, indicating that most or all network redundancies had been removed.

Experiment 1B

As experiment 1A except that the agent's speed was not fixed at maximum and the agent was held stationary for a random interval at the final beacon.

Experiment 2A

As experiment 1A except that the ModCTRNN model was used.

Experiment 2B

As experiment 2A except that the agent's speed was not constant and the agent was held captive for a time at the last beacon. Once the agent had evolved a good fitness, a further perturbation was introduced by setting η=0.7, r=2.0 for the forward motor neuron (forcing large long-lasting changes to the agent's speed *via* the noise offset mechanism) and η=0.1 for the rotation motor neurons. Most of the Results section is dedicated to an in-depth analysis of the network evolved in this experiment.

Experiment 3A

As experiment 2A except that compass sensor responses were of the 'linear cosine' type (see Fig. 3) and the number and maximum response direction of the compasses were allowed to evolve (up to a maximum of 10 compass pairs).

Experiment 3B

As experiment 3A except that the agent's speed was variable, as in experiment 1B.

Experiment 4

As experiment 3A except that the compass sensor response was the 'head direction cell' type (see Fig. 3), intended to model the response properties of head direction cells in rats (Taube, 1997).

Experiment 5

As experiment 3A except that the compass sensor response was the 'positive cosine' type (see Fig. 3).

Three replicates were performed for each experiment. Results are described for the most successful of the three runs.

Results

See Table 1 for a summary of the results.

Experiment 1A: CTRNN constant-speed PI

The fittest agent after approximately 67 000 generations returned to the nest within the time limit in 989 of 1000 test trials. The bilaterally symmetrical network (see Fig. 4)

Table 1. *Summary of results*

Expt	Network model*	Speed†	Compass response‡	Returns (%)§	Size¶
1A	C	Const.	Cos	98.9	6,24
1B	C	Var.	Cos	0.0	–
2A	M	Const.	Cos	90.4	0,8
2B	M	Var.	Cos	99.2	0,12
3A	M	Const.	Lin. cos	39.9	0,6
3B	M	Var.	Lin. cos	8.2	0,8
4	M	Const.	HD cell	5.6	0,12
5	M	Const.	Pos. cos	0.0	–

Results shown are for the best network obtained from three replicate experiments.

*Network model used: C, CTRNN; M, ModCTRNN.

†Speed during trial: Const., constant; Var., variable.

‡Compass response function (see Fig. 3): Cos, cosine; Lin. cos, linear approximation of cosine; HD cell, head direction cells; Pos. cos, positive cosine.

§Percentage of successful returns to nest out of 1000 test trials.

¶Number of interneurons and links.

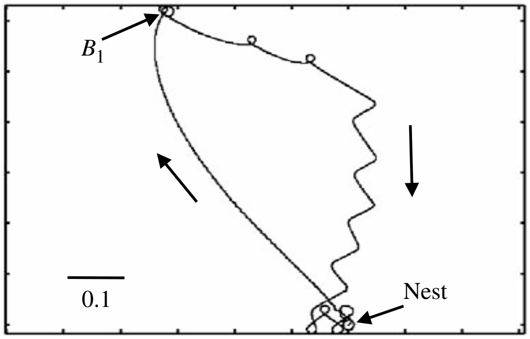

Fig. 5. Experiment 1A CTRNN agent performing the constant speed path integration task. B_1 indicates the single beacon. Beacon and nest are drawn as circles of radius 0.01.

contained 12×2 links (of which two pairs share the same source and target and so are not visible in the figure) and 3×2 interneurons. The shape of the return journey was highly dependent on the bearing to the nest from the last beacon and was generally not straight or direct (see Fig. 5 for one example) but usually consisted of two or more phases characterised by different patterns of oscillation in four of the neurons (shown in white in Fig. 4; Fig. 6 shows the neural dynamics), resulting in various looping and zigzagging behaviours. The non-oscillatory neurons act as integrators of the compass sensor inputs. Their output and the current compass sensor output feeds into the oscillatory group. A plot of fitness against the angle to the beacon from the nest for 360 single beacon trials (data not shown) showed a clear sinusoidal-shaped relationship, caused by the agent taking longer to return to the nest from some regions than others. This was clearly a suboptimal solution that did not home as *C. fortis* does in a straight line. This network was therefore not analysed in any further detail.

Experiment 1B: CTRNN variable-speed PI

This experiment failed to produce any successful agents after running three independent populations for 35 000 generations. Seeding the GA with the fittest genotype from experiment 1A also failed for three populations after 65 000 generations.

Experiment 2A: ModCTRNN constant-speed PI

Experiment 2A produced good results and, although evolved independently, produced a very similar network to the 2B experiment. The 2A results are not presented in any further detail here, since they do not add anything to the 2B results.

The 2A experiment was also repeated using the same settings except the maximum time constant values (τ and α)

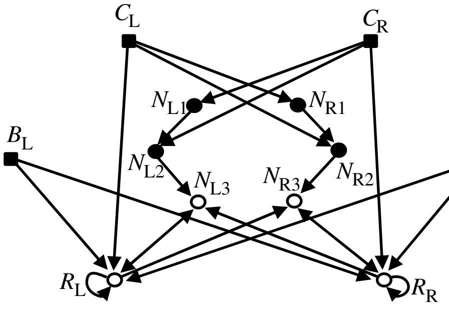

Fig. 4. Experiment 1A CTRNN network solving the constant speed path integration task. $B_{L/R}$, left/right beacon sensor; $C_{L/R}$, left/right compass sensor; $R_{L/R}$, left/right rotation motor neuron; $N_{L/R}$, interneurons. Arrows are directional weighted links; double-headed arrows indicate two links running in opposite directions. Open circles are neurons involved in the oscillations responsible for generating the various modes of behaviour seen in this agent during its return journey (see Fig. 5).

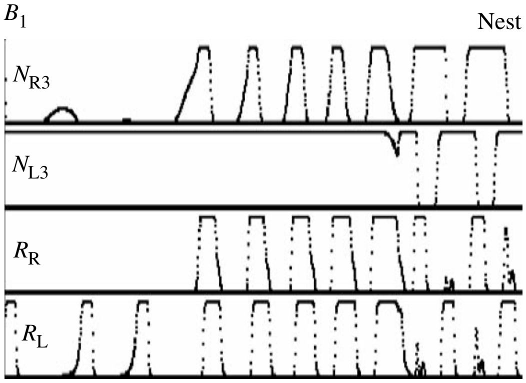

Fig. 6. Experiment 1A CTRNN network dynamics during the return phase of the journey (see Fig. 5). Plots are of neuron firing rates; the *y*-axis runs from 0 (bottom) to 1 for each. B_1 and nest indicate the time of arrival at the beacon and the nest, respectively (the nest is reached at the very end of the trial). $N_{L3/R3}$ and $R_{L/R}$ are as in Fig. 4.

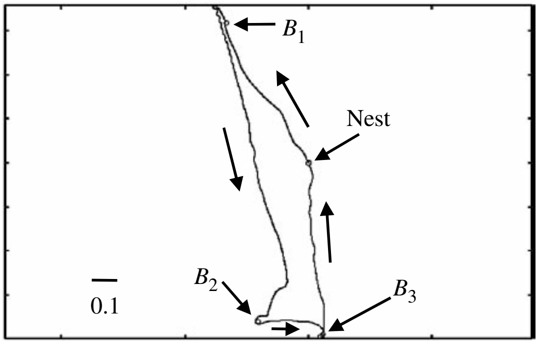

Fig. 7. Experiment 2B agent (see Fig. 8) performing the variable-speed path integration task. $B_{1,2,3}$ indicate the order the beacons were presented in. Beacons and nest are drawn as circles of radius 0.01.

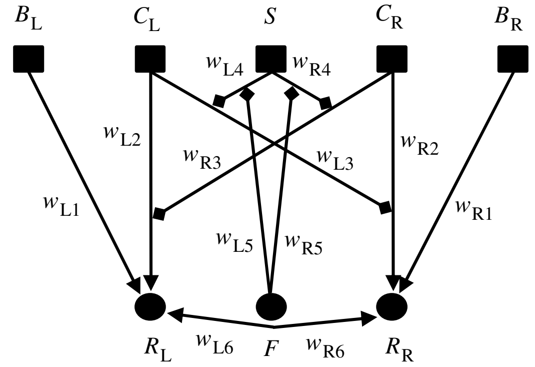

Fig. 8. Experiment 2B ModCTRNN network solving the variable-speed path integration task. $B_{L/R}$, left/right beacon sensor; $C_{L/R}$, left/right compass sensor; $R_{L/R}$, left/right rotation motor neuron; S, speed sensor; F, forward motor neuron. Arrows are directional weighted links; double-headed arrows are two links going in opposite directions between the same end points. Lines ending in small squares are links that modify other links. $w_{L1/R1}...w_{L6/R6}$ are weighted links (synapses). See Table 2 for parameter values.

allowed were 0.01, thus preventing neurons and weights from acting individually as integrators, but the GA failed to find any solutions.

Experiment 2B: ModCTRNN variable-speed PI

Since this experiment provided the most interesting and complete set of results, a detailed and extensive analysis is presented in this section. The results for the remaining experiments follow after. The fittest agent after about 35 000 generations returned to the nest within the time limit in 992 of 1000 test trials. The network contained six link pairs and no interneurons. The return journeys were approximately straight (see Fig. 7) considering the level of motor noise.

To test for artefacts resulting from numerical integration, 500 trials were performed using an integration step size of 0.001 (as during evolution) and 500 more with 0.0001; the agent reached the nest in 488 and 489 trials, respectively, a non-significant difference (χ^2=0, P>0.1; chi-squared test).

To test whether the ModCTRNN network was significantly better able to evolve solutions to this task than the CTRNN model, five more replicates of experiments 1B and five of 2B were performed for 60 000 generations each, evolving the ability to return home from a single randomly placed beacon with full motor noise applied. None of the 1B runs produced an agent that returned to the nest in any of 1000 test trials, but for all 2B runs (some of which were stopped early when it became clear PI had evolved) the agent returned home in greater than 50% of 1000 test trials. Taking a random course from the beacon would result in between 1.6 and 3.2 returns to the nest per 1000, therefore zero is worse and 500 much better than this random strategy. Five failures for the CTRNN and five successes for the ModCTRNN is a statistically significant difference (P<0.01; Fisher's exact test).

Following Mittelstaedt's bicomponent model (Mittelstaedt, 1985), this experiment assumed a sinusoidal compass sensor response function. The two compass sensors (C_L and C_R; later abbreviated as $C_{L/R}$) gave responses of $\cos(\theta_A+\pi/4)$ and $\cos(\theta_A-\pi/4)$, respectively, where θ_A is the agent's heading anticlockwise from the x axis. Taking $\lambda=\theta_A+(\pi/4)$, we obtain $C_L=\cos\lambda$ and $C_R=\sin\lambda$, exactly as Mittelstaedt used. It is therefore sufficient to integrate these values, multiplied by the agent's speed, to obtain an HV in geocentric rectangular form (see Eqn 1), and this is exactly what the network does (see Fig. 8; Table 2). Links $w_{L3/R3}$ integrate $C_{L/R}$, respectively, and store the values contralaterally in the values of weights $w_{R2/L2}$, respectively. Since weights $w_{L3/R3}$ are themselves governed by the speed sensor, *via* links $w_{L4/R4}$, their value is the agent's speed multiplied by the value of $w_{L4/R4}$ and they act to multiply the compass values by the current speed before they are integrated to the HV.

Weight values $w_{R2/L2}$ therefore constitute the HV, but they have a sufficient degree of leakiness over the agent's journey that significant homing errors would occur if nothing acted to compensate. Their time constants are 8.44, much less than the maximum allowed value during evolution of 1000. Compensation is achieved by normalising the values of $w_{L4/R4}$

Table 2. *Parameter values of the evolved network from experiment 2B (see Fig. 8)*

Component	Parameter values
$w_{L1/R1}$	w=12.0720
$w_{L2/R2}$	α=8.4355, β=0.0001
$w_{L3/R3}$	α=0.0123, β=2.0477
$w_{L4/R4}$	α=5.1753, β=–98.7613
$w_{L5/R5}$	w=65.9304
$w_{L6/R6}$	w=–3.5159
F	τ=0.0489, v_0=38.2195, b=42.8689
$R_{L/R}$	τ=0.0106, v_0=–3.6629, b=0.2994

Abbreviations: F, forward motor neuron; $R_{L/R}$, left/right rotation motor neuron; $w_{L1/R1}...w_{L6/R6}$, weighted links (synapses); α, weight time constant; β, weight bias; τ, neuron time constant; b, neuron bias; v_0, initial neuron potential (when t=0).

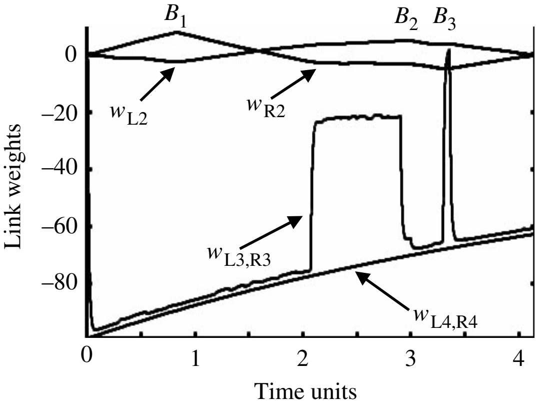

Fig. 9. Experiment 2B network performing the variable-speed path integration task. Plot shows link weights (y-axis; see Fig. 8) over the whole trial (x-axis; see Fig. 7). $B_{1,2,3}$ show the approximate time of arrival at each beacon (the nest is reached at the very end of the trial). From ~2 time units into the trial until 3 time units, a large change in the values of $w_{L3/R3}$ reflects noise applied to the forward motor output, as detected by the speed sensor. The second abrupt change in these values reflects the period of enforced captivity of the agent at the last beacon before homing begins.

to approximately match the amount of leakage that $w_{R2/L2}$ have undergone since the start of the journey. This is a simple, fixed-rate exponential decay function, visible in Fig. 9 in the values of $w_{L4/R4}$ and $w_{L3/R3}$, and is caused by the fixed output of the forward motor neuron *via* fixed weights $w_{L5/R5}$ acting to reduce the magnitude of weights $w_{L4/R4}$.

Once the last beacon is reached, the beacon sensors become inactive, phototaxis ceases and the HV is free to control the agent's behaviour. The HV causes the agent to turn towards home using an instantiation of, once again, Mittelstaedt's bicomponent model (Mittelstaedt, 1962; Eqn 3), where the two speed-weighted compass sensor integrals (i.e. the HV) are multiplied by the current value of the contralateral compass sensor. These values are fed into the two opposing rotation motors, $R_{L/R}$, by links $w_{L2/R2}$. The agent's rotation is the difference between these two motor outputs, completing the instantiation of Mittelstaedt's homing model.

The initial phototactic stage of the agent's journey is achieved by ipsilateral links $w_{L1/R1}$, which cause positive phototaxis (Braitenburg, 1984). These links feed into $R_{L/R}$ with higher weight values than $w_{L2/R2}$ and so suppress homing behaviour until the last beacon disappears. Thus, homing begins immediately and automatically as soon as the last beacon disappears, without any further trigger (the network ignores the food sensor). When the home vector becomes very large, phototactic suppression of homing may be incomplete and noise can cause positive phototaxis to be temporarily lost before the beacon is reached. The agent adopts a negative phototactic or sometimes homeward trajectory (see Fig. 10) before returning to positive phototaxis. This shows an interaction between the agent's two main modes of behaviour, which, we speculate, could be selected for to allow the agent to ignore beacons that were too far from its nest.

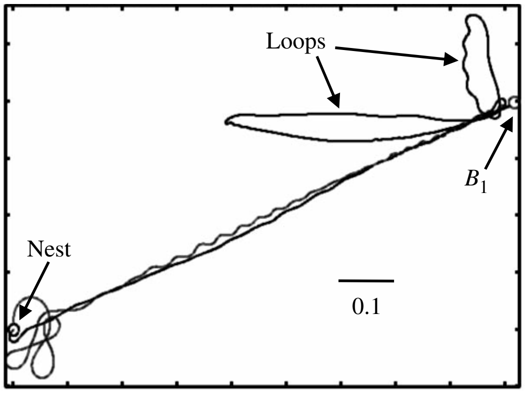

Fig. 10. Experiment 2B agent showing transient negative phototactic behaviour (loops) near the limit of the 'foraging' range selected for during evolution. Here, the agent twice turns away from the beacon (marked B_1) before finally reaching it and homing normally.

Simplified analytic model

A simplified analytic model (SAM) of the evolved network was used to compare its behaviour with that of *C. fortis* (see Appendix 2 for full details). All internal stateful dynamics of the network were removed except for the leaky integration of the HV. The leakage normalisation mechanism was removed (provisionally justified by assuming it approximates the behaviour of an unnormalised but less leaky network). The SAM also assumes that the agent travels in straight lines to visit all the beacons in turn before turning immediately to adopt the homing direction determined by the homing mechanism and the HV state. The SAM then produces a straight run to the point where the HV is zero. As the integrator time constants are increased, the system converges on an accurate PI system. For smaller values of the time constant, the HV no longer defines a fixed point on a fixed geocentric coordinate system. We can say rather that the agent has reached its estimate of home when the HV has reached zero (the 'HV zero point') and that the location of this point is a function of the agent's outward journey shape and its speed during both exploration and homing. Information stored in the HV for longer has undergone more decay, giving rise to systematic errors returning from straight and L-shaped journeys that closely resemble the systematic errors seen in *C. fortis* ants once the SAM's integrator time constant, α, has been fitted to the data.

L-*shaped routes*

Müller and Wehner (1988) conducted an extensive series of experiments testing the homing behaviour of *C. fortis* after L-shaped outward excursions. They showed that the error in the heading of the ant's homeward run (i.e. its angular deviation from a direct homeward path) varied systematically with features of the outward journey, specifically the length of the

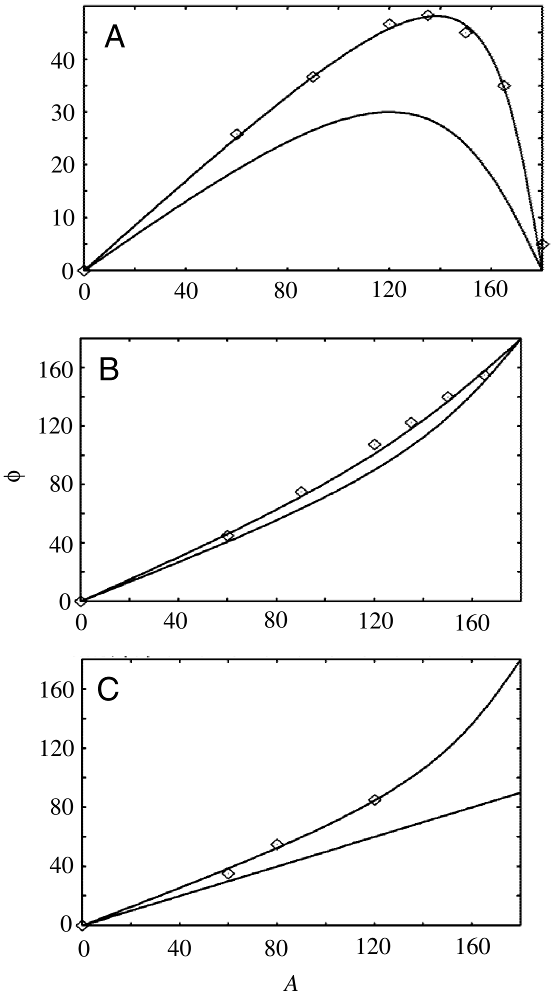

Fig. 11. The simplified analytic model (SAM) fitted to data (Müller and Wehner, 1988) taken directly from figs 11a–c in Hartmann and Wehner (1995). The ants walked along a channel from the nest, turned through an angle *A*, walked along a second channel before being released on a test field. The lengths of the channels were (A) 10 m and 5 m, (B) 5 m and 10 m and (C) 10 m and 10 m, respectively. φ shows the ants' homing direction (diamonds). In each graph, the top line shows the SAM using a time constant of α=18.38; the lower line shows the correct homing angle.

first straight section, the angle turned moving from the first to the second section and the length of the second straight section. The data are reproduced in fig. 11a–c from Hartmann and Wehner (1995) and were used to fit the SAM's α time constant parameter to the behaviour of *C. fortis* (see Fig. 11). α was fitted to the first of the three experiments, and the remaining experiments were used to check the model's generalisation. The fit is visually as good as that achieved by both the Hartmann–Wehner and Wittmann–Schwegler models. It should be borne in mind that the former was designed from the outset to mimic the data it was fitted to and that the latter was modified from its original form also specifically to fit the data. The SAM, however, was derived from a network evolved under selection for optimal PI and had all but one free parameter removed before fitting it to the *C. fortis* data. This makes leaky integration a strong candidate for a mechanistic explanation of the observed errors.

Long straight routes

Sommer and Wehner (2004) showed that, on straight journeys, *C. fortis* tends to underestimate the length of the return leg as an increasing function of the outward journey length. They fitted several models to this 'error function' and concluded that a leaky integrator function was one of the best fits (the other was a logarithmic function). The leaky integrator equation fitted to the data was:

$$y = \alpha\,(1 - e^{-\beta x})\,, \tag{8}$$

where x is the length of the outward journey to the feeder, y is the centre of the ant's search pattern at the end of the return journey and α and β are free parameters. This is similar to the model presented in Mittelstaedt and Glasauer (1991). We now present a slightly different model (which can be derived from the SAM model simplified for straight journeys), which fits the data equally well. We assume the ant's speed is constant throughout the journey and that it does not stop or search at the feeder for any significant length of time. The simplest leaky integrator model has only one parameter that influences the error function, namely its time constant (see Appendix 3), giving an error function of:

$$y = \alpha\,[1 - e^{-(x/\alpha)}]\,. \tag{9}$$

Eqn 9 corresponds to the integrator state upon reaching the feeder or to the assumption that the ant measures the outward journey using a leaky integrator but turns the leak off for the return journey. It is known, however, that in the species being modelled, PI continues during the return journey, due to the behaviour of *C. fortis* after forced detours (Schmidt et al., 1992). In keeping with this result, we assume that integrator leak is constant throughout both legs of the journey, giving us a single parameter error function of (see Appendix 3):

$$y = \alpha\,\ln\,[2 - e^{-(x/\alpha)}]\,. \tag{10}$$

This model can be obtained by reducing the SAM to the case of a straight journey at constant speed and is similar to the neural model evolved by GA in Vickerstaff (2003). Once leakage is assumed to be constant, we still have at least two options for a PI system: continuous and discontinuous (Collett and Collett, 2000). The continuous model leads to Eqn 10. In the discontinuous model, the integrator state at the feeder is stored, then zeroed. The compass response function is then inverted, and the ant navigates towards the point where its integrator state is the same as the stored value. This leads to a prediction of $y=x$ (no systematic error) regardless of the value of the integrator time constant. The shape of the Sommer–Wehner data clearly favours the continuous model over the discontinuous model under the leaky integrator model presented here. Fig. 12 shows Eqns 9 and 10 fitted to the data, plotted alongside Eqn 8. The three models are visually indistinguishable. Once again, the model presented here

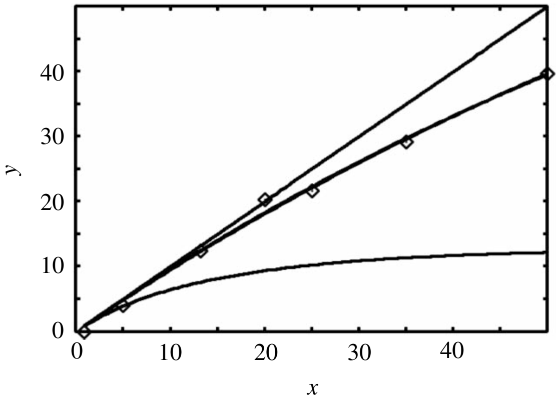

Fig. 12. The top line represents $y=x$, i.e. the correct homing distance. The middle line represents three leaky integrator models fitted to data (diamonds) taken directly from fig. 2 of Sommer and Wehner (2004). The models are Eqn 8 with $\alpha=98.27$, $\beta=0.0103$ (taken from their paper), Eqn 9 with $\alpha=103.2$ and Eqn 10 with $\alpha=193.6$ (fitted using least squared error). The models are indistinguishable. Also shown as the bottom line is Eqn 10 with $\alpha=18.38$, the value obtained fitting the simplified analytic model (SAM) to the L-shaped journey experiments (see text), clearly a poor fit.

contains fewer parameters but fits equally well compared with existing models that were explicitly designed [in the case of Sommer and Wehner's model by a process of searching for the best model from a set of four candidates (Sommer and Wehner, 2004)] to fit the data. The α time constants arrived at for the L-shaped (18.38) and straight journeys (193.6) are clearly very different (see Fig. 12); consequently, if we wish to use the same model to explain both data sets, we must propose that the effective integrator time constant can vary according to some unknown factor(s). This is in keeping with a type (3) model (see Introduction) but also reduces the model's parsimony unless there is a natural reason for such a variation to occur.

Search patterns

One notable feature of the homing equation (Eqn 3) is the behaviour produced after reaching the HV zero point. If we assume that the animal continues to move forward at a fixed speed at all times, it will pass directly over its estimate of home, thereby switching from the stable to the unstable equilibrium of the equation since it will now be facing directly away from the zero point. Any slight perturbation will turn it back round onto a new stable homeward trajectory, causing it to again cross over the home position and begin another outward loop, and so on. The result is that it would continue to loop around its current HV zero point. The exact behaviour produced would depend on the amount of noise present and the animal's rate of turn but would consist roughly of a series of equal-sized loops taking the animal away from and back towards the HV zero point, very roughly as is seen in *Cataglyphis* ants (Wehner and Srinivasan, 1981). Thus, the homing equation itself appears enough to generate a crude search behaviour when combined with a fully specified model of the animal's motion. Kim and Hallam (2000) have suggested that the homing mechanism of PI might be enough to also explain searching behaviour.

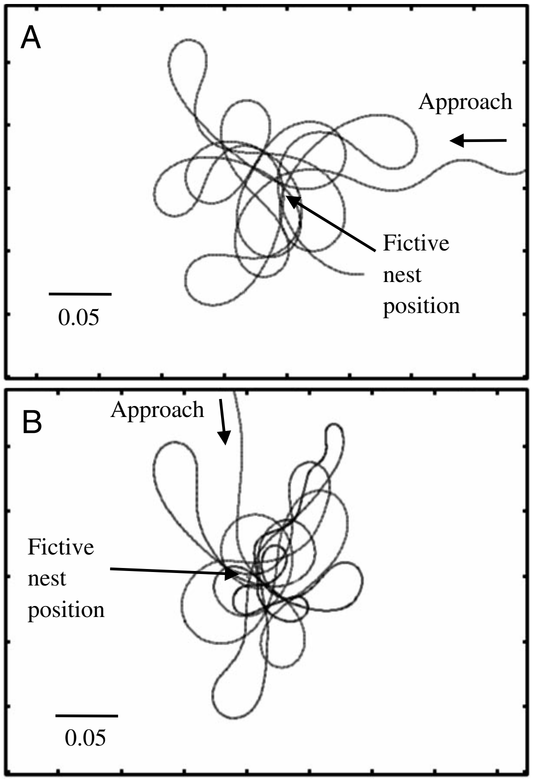

Fig. 13. The experiment 2B agent's behaviour if the nest is removed (A) with all sources of noise removed and (B) with 1% sensor noise, 10% rotation motor noise and 70% forward motor noise (as during evolution). A searching behaviour that does not rely on the presence of noise is clearly visible and arises from the coupled dynamics of the agent's network and motion.

The trajectory of the 2B experiment agent is more complex than the expected looping behaviour and shows little dependence on the amount of noise present (Fig. 13). Thus, the network appears to have evolved to produce a more noise robust, and possibly more efficient, search behaviour. Since the fourth fitness criteria rewards shorter total journey times, we have been implicitly selecting for efficient search as well as direct, rapid initial homing. Since the agent reaches home in ~99% of trials, the third fitness criterion, penalising higher average distances from the nest during homing if the agent did not reach home before the end of the trial, will not have been significant during the final stages of the network's evolution, allowing broad search patterns to evolve if necessary.

To compare the experiment 2B agent's search with that seen in *C. fortis*, the average search density of 50 trials returning from a beacon 0.75 distance units east was plotted (Fig. 14), showing an approximately symmetrical bell shape. *C. fortis* (Wehner, 1992) and other *Cataglyphis* species (Wehner and Srinivasan, 1981) show a similar search profile. Additionally, the ant performs a broader search pattern the longer the preceding outward journey was (Wehner, 1992) and therefore

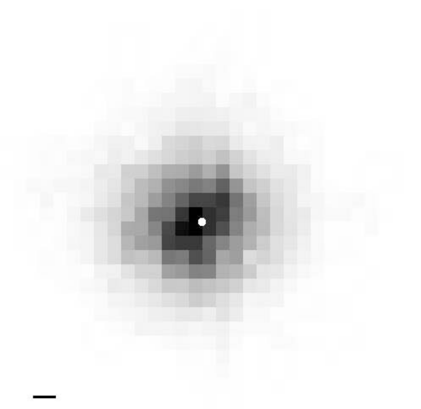

Fig. 14. Experiment 2B search density of the agent when the nest has been removed, averaged over the final 35% of 50 independent trials. The agent is returning from a single beacon placed at a distance of 0.75 to the right in all trials. The white dot shows the fictive nest position; the black bar shows the size of the nest (diameter 0.02).

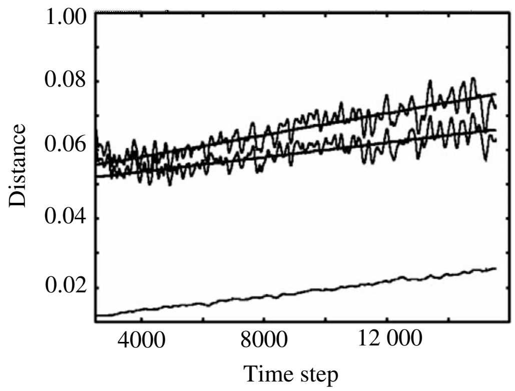

Fig. 15. Distances averaged over 200 trials where the agent has returned from a beacon 0.75 distance units away to the fictive nest position and is performing the search behaviour. Top line: average distance of agent from the nest; linear regression fitted is d=1.5820×10^{-6}t+0.0519. Middle line: average distance of agent from home vector (HV) zero point (see text); model fitted is d=1.0598×10^{-6}t+0.0496. Bottom line: average distance of HV zero point from nest; linear regression model (not plotted) is d=1.0484×10^{-6}t+0.0091. The middle line, which increases during the search behaviour, suggests the agent is 'deliberately' searching further from the nest the longer it has been searching.

the greater its uncertainty about the home location. Also, during a single search, the ant gradually increases the width of the pattern the longer it has been searching (Wehner and Srinivasan, 1981). The 2B network effectively has a built-in timer in the exponential decay of weights w_{L4} and w_{R4}, which influences the magnitude of its HV and which appears to be the only mechanism that might influence the search pattern width over time. In fact, the timer could roughly reproduce both features of the ants' search, provided only the search width is inversely related to the magnitude of the decaying weights, since lengthier journeys take longer to complete on average. The ants' search pattern stays centred on the same location during a search (Wehner and Srinivasan, 1981) but that of the agent may begin to drift around due to cumulative HV errors, which must therefore also be accounted for when characterising its search behaviour.

A plot was made of the average distance of the agent from the fictive nest position during the return and search phases of 200 journeys where the agent was returning from a single beacon 0.75 distance units away in a random direction, (Fig. 15). The nest had been removed and each trial the agent was allowed to search for three times as long as during evolution. The agent's HV error was also estimated during the 200 runs (1) by taking a running average of the agent's position over a time window long enough to contain multiple search loops and (2) by using the SAM to derive the approximate position where the agent's HV (weights w_{L2} and w_{R2}) would be zero were it to move to that location in a straight line (see Appendix 2). Since these were in good agreement (data not shown), only the second estimate was used. On the plausible assumption that the search efficiency cannot be increased by deliberately introducing HV errors, only the agent's distance from its estimate of the nest location can be considered as evidence of an evolved search behaviour similar to the ants. This distance is obtained by subtracting the HV error vector from the agent's coordinates. The average distance from the nest, the HV error and the agent's distance from the HV zero point all increase during the search period, suggesting that both random and systematic increases in search width are at play (see Fig. 15).

Plots of the agent's initial search pattern width after returning from beacons at different distances from the nest did not show any trend towards an increase in relation to journey length. This may be due to the relatively small range of journey lengths used (0.5 to 1.0). The agent cannot readily be tested at greater distances without being re-tuned by evolution, since its homing becomes inaccurate beyond the foraging range it has experienced during its evolution.

Adding long delays

C. fortis can retain some memory of its HV for long periods (up to about four days) if forcibly held still (Ziegler and Wehner, 1997). To test the agent's ability to perform the same task with its leaky integrators we further evolved the network obtained in experiment 2B whilst subjecting the agent to a gradually increasing holding time at the last beacon. Finally, the agent was held for up to 50 time units (i.e. 50 times the longest possible direct return journey). The fittest agent returned to the nest within the time limit in 931 of 1000 test trials.

The network had evolved to retain the same structure but had increased the time constant on the HV storing weights $w_{L2/R2}$, from 8.44 to 113.8, and had also increased the magnitude of weights $w_{L4/R4}$ from approximately 100 to 153 (this was achieved by adding a new pair of links with the same start and end points, since the maximum allowed strength of any single link was set to ±100). This modification is understandable since increasing the time constant of the integrators leads to less leakage (and so approximates perfect integration more closely) but smaller HV values. Smaller values in turn reduce the agent's maximum rate of turn during homing, which influences the straightness of its path under noisy conditions, and the shape of its search pattern. Increasing the magnitude of $w_{L4/R4}$ compensates by scaling the HV values back up.

Varying the compass sensor

All the remaining experiments consist of applying variations to the shape of the compass sensor response function. This was felt to be an important assumption to study since the ant's representation is not known (but see Labhart, 2000), and the evolved solution may clearly be heavily influenced by the choice made.

Experiment 3A: ModCTRNN constant speed with linear cosine

The fittest agent used two pairs of compass sensors with maximum responses set to approximately ±π/4 and ±3π/4, but, since responses of sensors separated by π radians are equal in magnitude but opposite in sign, this was equivalent to a single pair of sensors at ±π/4. Taking this into account, the network had the same structure as that evolved in experiment 2B except it lacked link pairs 4, 5 and 6 (see Fig. 8). The agent returned to the nest in 399 of 1000 random independent trials. The SAM was modified to model this network by changing the compass response function to the linear cosine used here. The systematic errors seen after L-shaped journeys are clearly a much poorer fit to the data from *C. fortis* than those produced by a cosine-shaped compass response (see Fig. 16) and are also dependent on the angle to the compass cue relative to the L shape (data not shown).

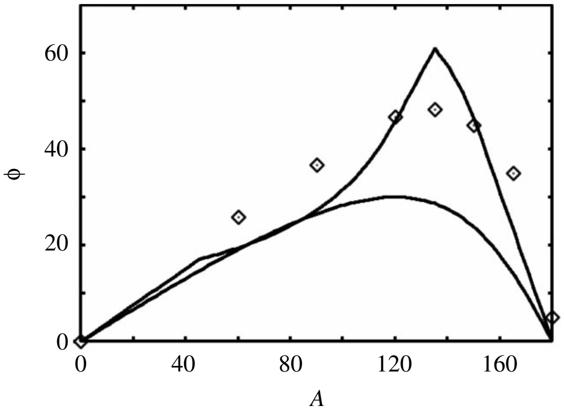

Fig. 16. The simplified analytic model (SAM) modified to use a piecewise linear approximation of cosine (see Fig. 3) as the compass response function (upper line) fitted using least sum of squared error to data (diamonds; Müller and Wehner, 1988) taken from fig. 11a in Hartmann and Wehner (1995). The lower line shows the correct homing angle. The value of the time constant obtained is α=38.00. The fit is noticeably poorer than that obtained with the cosine-shaped compass sensor response function (see Fig. 11A). The axes are labelled as in Fig. 11.

Experiment 3B: ModCTRNN variable speed with linear cosine

The fittest agent returned to the nest in only 82 of 1000 test trials. The agent used one compass sensor pair with maximum response angles of approximately ±π/4. The network had the same architecture as experiment 3A, i.e. it did not take into account information from the speed sensor, even though speed was not constant in this experiment. This accounts for its much poorer performance.

Experiment 4: ModCTRNN constant speed with head direction cells

The fittest agent returned to the nest in only 56 of 1000 test trials. The agent used three pairs of compass sensors with maximum response angles set to approximately ±π/4, ±7π/12 and ±3π/4. The network structure was very different from those evolved in the other experiments but, since the performance was so poor, it cannot be considered a viable PI model and will not be presented here.

Experiment 5: ModCTRNN constant speed with positive cosine

This experiment failed to produce any successful agents.

Discussion

The GA successfully evolved a bicomponent model of PI using very few neurons and links (synapses), provided the neurons were given elaborate properties (the ModCTRNN model). The architecture is probably the simplest implementation of PI and was therefore easier for the GA to discover and is modular in that the two components of the HV do not have to interact in order to be correctly updated. The imposition of bilateral symmetry and compass input and motor output based on pairs of neurons probably also encouraged the production of a bicomponent system. So far, only cosine-shaped compass response functions have evolved to give successful solutions, probably because this allows the sensor output to be fed almost directly into the HV. Wittmann and Schwegler (1995) show that any symmetrical unimodal compass response function can be converted into a sinusoidal shape, but this requires extra processing between the sensors and the HV accumulator, which our experiments using non-cosine compasses failed to evolve, although this could be due to the small number of replicates performed. The compass representation we chose has negative as well as positive values, which clearly cannot be interpreted directly as a firing rate. However, each compass sensor could be replaced with two sensors, each having only positive outputs, one representing the positive half of the cosine shape, the other the negative part. This solution closely resembles the stimulus–response behaviour of four sensory interneurons of the cricket cercal sensory system (Miller et al., 1991), which show half-wave rectified sine waves. The evolved network is then easily modified by the addition of further links and would retain the same behaviour. The network then requires approximately 30 links in place of the original 12.

An objection previously directed at the bicomponent model is that it can only generate perfect, error-free PI. We have shown that, if implemented using leaky integrators, the model naturally generates errors that can quantitatively match those seen in *C. fortis*, once the time constant has been set appropriately (but can also still produce accurate navigation using larger time constant values). This arises from a model evolved only under selection for accurate, error-free navigation that was not originally intended to mimic errors.

The model also automatically generates search behaviour at the nest that is similar in some respects to the behaviour of the ant. This was unexpected and not the result of any explicit feature of the artificial selection scheme. It seems likely that the Wittmann–Schwegler and Hartmann–Wehner models could produce a similar behaviour. Indeed, Hartmann and Wehner (1995) note that their homing mechanism has the same two equilibrium states (one pointing home, the other directly away from home) as the SAM but never show any simulations of their agent's behaviour near to the nest. Instead, they assume that systematic search requires a separate control system that the HV simply triggers when needed. Here, we have shown that such an extra mechanism may be entirely unnecessary. The agent's search pattern produces a bell-shaped density profile similar to *Cataglyphis*'. The search also appears to get slightly wider over time, as does the ant's, although this is only detectable after averaging many searches and cannot yet be assigned an adaptive role for the agent with any confidence.

Clearly, our model is a simple one, particularly in the reduced SAM form, and should not be expected to reproduce all aspects of the ant's behaviour, indeed it does not. The SAM's single time constant can be set to reproduce the features of L-shaped journeys or long straight journeys, but not both at once. The SAM contains only one time constant, whereas in real systems, such as the goldfish oculomotor neural integrator (Major et al., 2004), where the integrator must respond rapidly to sacchades but decay only slowly between them, significant dynamics can occur on multiple time scales. The full dynamical model is needed to produce search behaviour.

There appear to be many possible future applications of the methodologies applied here for path integration and ant navigation in general [indeed this is not the first evolved model of insect navigation (Dale and Collett, 2001)]. In forthcoming work, the present PI model will be subjected to a more-detailed analysis. Follow-up experiments could also be aimed at reproducing different features of desert ant behaviour, such as the open-jaw learning experiments (Collett et al., 1999; Wehner et al., 2002) or the interactions between PI and landmark recognition (Collett et al., 2003). A further interesting possibility would be an investigation of the evolution of the dorsal rim area (DRA) of the ant's compound eye, which is able to detect the e-vector patterns of polarised skylight. The system is well enough understood (Labhart, 1999, 2000; Lambrinos, 2003) to attempt evolving the number and orientation of the DRA polarised light detectors within a simulation of the e-vector patterns present over the ant's desert habitat.

Finally, we would like to note the benefits and versatility of the modelling approach used in this paper, namely a whole agent simulation where a neural model explicitly controls an agent moving within a simple simulated world, in this case constructed with the aid of a GA. This allowed a direct comparison to be made between models evolved under differing conditions and enabled us to forego the usually unavoidable *a priori* assumptions concerning internal HV representations. Perhaps contrary to expectation (and partly due to the GA's pruning ability), the best evolved network is structurally simple and can be readily analysed. A whole-agent approach naturally leads us to focus on the agent's behaviour rather than the network alone and has allowed us to uncover unexpectedly rich behaviour with interesting parallels to the real behaviour of the species being modelled.

Appendix 1: details of the genetic algorithm

The agent's sensors, control network and motors were encoded using a variable-length genotype. This allowed genes to be added and deleted during mutation within defined frequency ranges. Each genotype had a probability of 0.1 of having its length changed per generation. This was chosen to be a single gene addition or deletion with the ratio 3:5, except when the genetic algorithm (GA) was in evolutionary pruning mode, where the ratio was 0:1. Within each gene are the parameters specifying the encoded component(s), each of which was mutated with probability 0.05 per generation by adding a random Gaussian value mean 0 standard deviation 0.02 multiplied by the maximum range of the parameter (detailed below).

Out-of-bounds values were corrected by bouncing back from the boundary or wrapping round to the other boundary for circular parameters. Recombination was not used (i.e. reproduction was asexual).

Gene parameters were mutated by the GA within the following ranges: $w,\beta\epsilon[-100,100]$, $\alpha,\tau\epsilon[-2,3]$ (but mapped using 10^x to [0.01,1000] before being used), $b,v_0\epsilon[-50,50]$ where w is link (or synapse) weight, β is weight bias, α is weight time constant, τ is neuron time constant, b is neuron bias and v_0 is initial neuron potential (see Eqns 6, 7). Large parameter ranges were selected since previous experience (Vickerstaff, 2003) had lead us to expect the evolution of integrative neurons with large time constants and large input weights, since this is the simplest way for a CTRNN to approximate temporal integration. Weights, biases and initial cell potential values for newly created genes (where a new gene was added to a genotype or a new population was created from scratch) were set to zero rather than being selected at random from the entire range, such that later creep mutations could gradually increase or decrease the values. Large initial values would have probably resulted in networks where most neurons were saturated either fully on or off. A more general way to avoid this problem is given by Mathayomchan and Beer (2002).

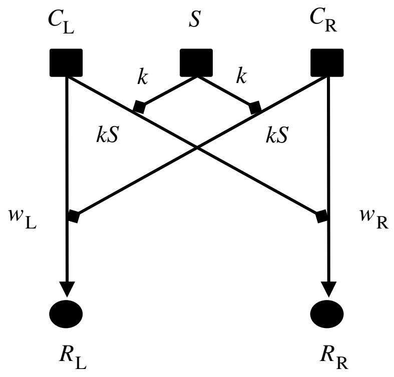

Fig. 17. Network used to derive the simplified analytic model (SAM), adapted from the network evolved in experiment 2B. See Appendix 2 for explanation.

Euler first-order method numerical integration was used for all numerical integrate with a step size of 0.001; evolved solutions were checked for artefacts by running with a step size of 0.0001.

Appendix 2: simplified analytic model

This model is based on the network evolved in experiment 2B (see Fig. 8), but the same basic mechanisms were also evolved independently in experiments 2A, 3A and 3B. The network is simplified as shown in Fig. 17 by the following assumptions: (1) the agent performs perfect positive phototaxis (i.e. moves in straight lines to visit all beacons) by a means that does not interact with the path integration mechanism (including homing); (2) the only network internal state is that held in the values of weights $w_{L2/R2}$ (referred to now as simply $w_{L/R}$, i.e. the values of the weights connecting the left and right compass sensors, respectively, to their ipsilateral rotation motor neurons); all other weights and the firing rates of the two rotation motor neurons are assumed to have sufficiently small time constants that they reach their equilibrium values immediately; (3) the agent's rate of turn is sufficiently fast that it reaches equilibrium immediately and (4) there is negligible sensor and motor noise. We are thus ignoring the dynamics of turning, which allow for the generation of the search pattern at the nest location, and the leakage normalisation mechanism (which we assume here has only the effect of increasing the effective integrator time constants).

Having removed links $w_{L5/R5}$, weights $w_{L4/R4}$ are now constant in value and are a parameter of the simplified model, referred to below as k (see Fig. 17). Assuming that weights $w_{L3/R3}$ have small time constants, they will always be at equilibrium value, namely kS by Eqn 7, where S is the agent's speed. We can now specify the behaviour of weights $w_{L/R}$, which store the HV, by integrating Eqn 7 under the assumption that the agent's velocity (speed and heading) is constant:

$$w_L(t_2) = k\,S\,C_R + [w_L(t_1) - k\,S\,C_R]e^{[-(t2-t1)]/\alpha}\,, \quad (11)$$

$$w_R(t_2) = k\,S\,C_L + [w_R(t_1) - k\,S\,C_L]e^{[-(t2-t1)]/\alpha}\,, \quad (12)$$

where $C_{L/R}$ are the compass sensor outputs, α is the time constant of the integrators (another free parameter of the model), t_1 is the time that we start running the model, t_2 is the time we wish to know the HV state, and $w_{L/R}(t_1)$ are the initial values of the HV. We can therefore break any journey down into a series of sections of uniform velocity and use Eqns 11 and 12 iteratively to determine the HV state at the end of an arbitrary journey [the HV should be initialised to (0,0)].

Once the agent has reached the end of the outward journey, we wish to determine where the homing mechanism will take it. We do this by assuming that it will immediately turn to adopt the stable equilibrium orientation as determined by Eqn 5 (see Fig. 2 for the case where compass responses are cosine shaped), such that R_L=R_R where R_L and R_R are the firing rates of the left and right rotation motor neurones, respectively (see Fig. 17). Assuming the rotation motor neurons have fast responses, this is satisfied when their inputs are the same, namely w_LC_L=w_RC_R, regardless of the precise compass response function used.

We also wish to know where the agent will be when its HV reaches zero, i.e. when (w_L,w_R)=(0,0), since this constitutes its estimate of the nest location. We can ask what compass output values C_L, C_R (i.e. which heading) are needed to bring both (w_L,w_R) to zero at the same time. Taking $w_L(t_1)$,$w_R(t_1)$ to be the HV state at the end of the outward journey, we set Eqns 11 and 12 equal to zero. Substituting for $\exp[-(t_2-t_1)/\alpha]$ in Eqn 11 using Eqn 12 and simplifying, we find that w_LC_L=w_RC_R, as before, indicating that the homeward journey will always be straight. Eqn 11 gives:

$$t_2 = -\alpha\ \ln\{(k\,S\,C_R)/[k\,S\,C_R - w_L(t_1)]\} + t_1, \quad (13)$$

where t_2 is the time the agent will reach its estimate of home. Note that there are two equilibrium values of the heading using cosine response functions (see Fig. 2), but we can use Eqn 13 to determine which is the stable homeward one using the condition $t_2 \geqslant t_1$.

Using the case of experiment 2B, where C_L=$\cos(\theta_A+\pi/4)$ and C_R=$\cos(\theta_A-\pi/4)$ (where θ_A is the agent's heading), taking λ=$\theta_A+\pi/4$, we get C_L=$\cos\lambda$, C_R=$\sin\lambda$, $d\lambda/dt$=$d\theta_A/dt$. To determine which direction the agent will home in, we use $w_L\cos\lambda$=$w_R\sin\lambda$, therefore:

$$\theta_A = \tan^{-1}(w_L/w_R) - \pi/4\ . \quad (14)$$

To determine the distance the agent will travel, we find the duration of the homeward journey using Eqn 13 and multiply by the agent's (constant) speed. If the value is negative, Eqn 14 is giving us the anti-homing value of θ_A and we correct this by adding π to θ_A. We can use this method to determine the homing errors predicted by the SAM under variations of integrator leakiness (parameter α) and compass sensor response function ($C_{L/R}$).

Appendix 3: one-dimensional model

If we have an animal walking at a constant speed of 1 along a straight path for a distance of x, how far will it walk on the return journey, y, if we assume it uses a leaky integrator, w, for measuring distance? We will assume that the integrator input is k during the outward leg and $-k$ during the return leg. Since speed is unity distance from the start, x equals time from the start t on the outward leg, and distance from feeder y equals time from feeder t on the return leg.

Case 1: continuous PI

Integrator dynamics during the outward leg are $\alpha \mathrm{d}w/\mathrm{d}x=(-w)+k$ and during the return leg are $\alpha \mathrm{d}w/\mathrm{d}y=(-w)-k$ where α is the integrator decay rate. The animal stops when w has reached zero again. This gives a predicted homing distance of $y=\alpha\ln[2-\exp(-x/\alpha)]$ (and is independent of k).

Case 2: discontinuous PI

Integrator dynamics during both legs are $\alpha\ \mathrm{d}w/\mathrm{d}t=(-w)+k$. At the feeder, the value of w is stored and w set to zero. The animal stops when w has reached the stored value again. This leads to a predicted homing distance of $y=x$ regardless of the value of α.

This work was financially supported by BBSRC grant number 02/A1/S/08410. We wish to thank Thomas Collett and Paul Graham for their many helpful suggestions and comments during the preparation of this manuscript, and also Kyran Dale and Matthew Quinn for their advice regarding genetic algorithms. We thank both anonymous reviewers for the additions and changes they suggested.

References

Andersen, R. A., Essick, G. K. and Siegel, R. M. (1985). Encoding of spatial location by posterior parietal neurons. *Science* **230**, 456-458.

Anzai, A., Ohzawa, I. and Freeman, R. D. (1999). Neural mechanisms for processing binocular information i. Simple cells. *J. Neurophysiol.* **82**, 891-908.

Beer, R. D. (1990). *Intelligence as Adaptive Behavior: An Experiment in Computational Neuroethology*. Boston: Academic Press.

Benhamou, S. and Séguinot, V. (1995). How to find one's way in the labyrinth of path integration models. *J. Theor. Biol.* **174**, 463-466.

Braitenburg, V. (1984). *Vehicles, Experiments in Synthetic Psychology.* Cambridge, MA: MIT Press.

Chapman, T. (1998). Human models of navigation in ants: testing the Hartmann Wehner model. *Göttingen Neurobiology Report 1998; Proceedings of the 26th Göttingen Neurobiology Conference 1998*, vol. I (ed. N. Elsner and R. Wehner), p. 28. Stuttgart: Georg Thieme Verlag.

Chiel, H. J. and Beer, R. D. (1997). The brain has a body: adaptive behaviour emerges from interactions of nervous system, body and environment. *Trends Neurosci.* **20**, 553-557.

Collett, M. and Collett, T. S. (2000). How do insects use path integration for their navigation? *Biol. Cybern.* **83**, 245-259.

Collett, M., Collett, T. S., Chameron, S. and Wehner, R. (2003). Do familiar landmarks reset the global path integration system of desert ants? *J. Exp. Biol.* **206**, 877-882.

Dale, K. and Collett, T. S. (2001). Using artificial evolution and selection to model insect navigation. *Curr. Biol.* **11**, 1305-1316.

Dayan, P. and Abbott, L. F. (2001). *Theoretical Neuroscience*. Cambridge, MA: MIT Press.

Di Paolo, E. A., Noble, J. and Bullock, S. (2000). Simulation models as opaque thought experiments. In *Artificial Life VII: Proceedings of the Seventh International Conference on Artificial Life* (ed. M. A. Bedau, J. S. McCaskell, N. H. Packard and S. Rasmussen), pp. 497-506. Cambridge, MA: MIT Press.

Dyer, F. C., Gill, M. and Sharbowski, J. (2002). Motivation and vector navigation in honey bees. *Naturwissenschaften* **89**, 262-264.

Egorov, A. V., Hamam, B. N., Fransen, E., Hasselmo, M. E. and Alonso, A. A. (2002). Graded persistent activity in entorhinal cortex neurons. *Nature* **420**, 173-178.

Etienne, A. S., Maurer, R., Boulens, V., Levy, A. and Rowe, T. (2004). Resetting the path integrator: a basic condition for route-based navigation. *J. Exp. Biol.* **207**, 1491-1508.

Gabbiani, F., Krapp, H. G. and Laurent, G. (1999). Computation of object approach by a wide-field motion-sensitive neuron. *J. Neurosci.* **19**, 1122-1141.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley Longman Publishing

Hartmann, G. and Wehner, R. (1995). The ant's path integration system: a neural architecture. *Biol. Cybern.* **73**, 483-497.

Hatsopoulos, N., Gabbiani, F. and Laurent, G. (1995). Elementary computation of object approach by a wide-field visual neuron. *Science* **270**, 1000-1003.

Kim, D. and Hallam, J. C. T. (2000). Neural network approach to path integration for homing navigation. In *From Animals to Animals 6: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior* (ed. J.-A. Meyer, A. Berthoz, D. Floreano, H. L. Roitblat and S. W. Wilson), pp. 228-235. Cambridge, MA: MIT Press.

Kimchi, T., Etienne, A. S. and Terkel, J. (2004). A subterranean mammal uses the magnetic compass for path integration. *Proc. Natl. Acad. Sci. USA* **101**, 1105-1109.

Koch, C. (1999). *Biophysics of Computation.* New York: Oxford University Press.

Labhart, T. (1999). How polarization-sensitive interneurons of crickets see the polarization pattern of the sky: a field study with an optoelectronic model neurone. *J. Exp. Biol.* **202**, 757-770.

Labhart, T. (2000). Polarization-sensitive interneurons in the optic lobe of the desert ant *Cataglyphis bicolor. Naturwissenschaften* **87**, 133-136.

Lambrinos, D. (2003). Navigation in desert ants: the robotic solution. *Robotica* **21**, 407-426.

Layne, J. E., Barnes, W. J. P. and Duncan, L. M. J. (2003a). Mechanisms of homing in the fiddler crab *Uca rapax* 1. Spatial and temporal characteristics of a system of small-scale navigation. *J. Exp. Biol.* **206**, 4413-4423.

Layne, J. E., Barnes, W. J. P. and Duncan, L. M. J. (2003b). Mechanisms of homing in the fiddler crab *Uca rapax* 2. Information sources and frame of reference for path integration. *J. Exp. Biol.* **206**, 4425-4442.

Loewenstein, Y. and Sompolinsky, H. (2003). Temporal integration by calcium dynamics in a model neuron. *Nat. Neurosci.* **6**, 961-967.

Major, G., Baker, R., Aksay, E., Mensh, B., Seung, H. S. and Tank, D. W. (2004). Plasticity and tuning by visual feedback of the stability of a neural integrator. *Proc. Natl Acad. Sci. USA* **101**, 7739-7744.

Mathayomchan, B. and Beer, R. D. (2002). Center-crossing recurrent neural networks for the evolution of rhythmic behavior. *Neural Comput.* **14**, 2043-2051.

Maurer, R. and Séguinot, V. (1995). What is modelling for? A critical review of models of path integration. *J. Theor. Biol.* **175**, 457-475.

Miller, J. P., Jacobs, G. A. and Theunissen, F. E. (1991). Representation of sensory information in the cricket cercal sensory system. I. Response properties of the primary interneurons. *J. Neurophysiol.* **66**, 1680-1689.

Mittelstaedt, H. (1962). Control systems of orientation in insects. *Annu. Rev. Entomol.* **7**, 177-198.

Mittelstaedt, H. (1985). Analytical cybernetics of spider navigation. In *Neurobiology of Arachnids* (ed. F. Barth), pp. 298-318. Berlin: Springer.

Mittelstaedt, M. L. and Glasauer, S. (1991). Idiothetic navigation in gerbils and humans. *Zool. Jb. Physiol.* **95**, 427-435.

Mittelstaedt, M. L. and Mittelstaedt, H. (2001). Idiothetic navigation in humans: estimation of path length. *Exp. Brain Res.* **139**, 318-332.

Moller, P. and Görner, P. (1994). Homing by path integration in the spider *Agelena labyrinthica* Clerck. *J. Comp. Physiol. A* **174**, 221-229.

Müller, M. and Wehner, R. (1988). Path integration in desert ants, *Cataglyphis fortis. Proc. Natl. Acad. Sci. USA* **85**, 5287-5290.

Müller, M. and Wehner, R. (1994). The hidden spiral: systematic search and path integration in desert ants, *Cataglyphis fortis. J. Comp. Physiol. A* **175**, 525-530.

Ortega-Escobar, J. (2002). Evidence that the wolf-spider *Lycosa tarentula* (Araneae, Lycosidae) needs visual input for path integration. *J. Arachnol.* **30**, 481-486.

Peck, S. L. (2004). Simulation as experiment: a philosophical reassessment for biological modeling. *Trends Ecol. Evol.* **19**, 530-534.

Salinas, E. and Abbott, L. F. (1996). A model of multiplicative neural responses in parietal cortex. *Proc. Natl. Acad. Sci. USA* **93**, 11956-11961.

Schmidt, I., Collett, T. S., Dillier, F.-X. and Wehner, R. (1992). How desert ants cope with enforced detours on their way home. *J. Comp. Physiol. A* **171**, 285-288.

Seung, H. S. (1996). How the brain keeps the eyes still. *Proc. Natl. Acad. Sci. USA* **93**, 13339-13344.

Shen, L. (1989). Neural integration by short term potentiation. *Biol. Cybern.* **61**, 319-325.

Sommer, S. and Wehner, R. (2004). The ant's estimate of distance travelled: experiments with desert ants, *Cataglyphis fortis. J. Comp. Physiol. A* **190**, 1-6.

Taube, J. S. (1997). Head direction cells and the neurophysiological basis for a sense of direction. *Prog. Neurobiol.* **55**, 225-256.

Touretzky, D. S., Redish, A. D. and Wan, H. S. (1993). Neural representation of space using sinusoidal arrays. *Neural Comput.* **5**, 869-884.

Vickerstaff, R. (2003). First steps in evolving path integration in simulation. In *Advances in Artificial Life, 7th European Conference, ECAL 2003, Dortmund, Germany, September 2003, Proceedings* (ed. W. Banzhaf, T. Christaller, P. Dittrich, J. T. Kim and J. Ziegler), pp. 209-216. Berlin: Springer-Verlag.

Wehner, R. (1992). Arthropods. In *Animal homing* (ed. F. Papi), pp. 45-144. London: Chapman and Hall.

Wehner, R. (1994). The polarization-vision project: championing organismic biology. In *Neural Basis of Behavioural Adaptations* (ed. K. Schildberger and N. Elsner), pp. 103-143. Stuttgart: Gustav Fischer.

Wehner, R. (2003). Desert ant navigation: how miniature brains solve complex tasks. *J. Comp. Physiol. A* **189**, 579-588.

Wehner, R. and Srinivasan, M. V. (1981). Searching behaviour of desert ants, genus *Cataglyphis* (Formicidae, Hymenoptera). *J. Comp. Physiol. A* **142**, 315-338.

Wehner, R., Gallizzi, K., Frei, C. and Vesely, M. (2002). Calibration processes in desert ant navigation: vector courses and systematic search. *J. Comp. Physiol. A* **188**, 683-693.

Wittmann, T. and Schwegler, H. (1995). Path integration – a neural model. *Biol. Cybern.* **73**, 569-575.

Ziegler, P. E. and Wehner, R. (1997). Time courses of memory decay in vector-based and landmark-based systems of navigation in desert ants, *Cataglyphis fortis*. *J. Comp. Physiol. A* **181**, 13-20.