Neural Process Lett (2011) 34:277–291
DOI 10.1007/s11063-011-9198-5

Path Integration Mechanism with Coarse Coding of Neurons

DaeEun Kim · Jiwon Lee

Published online: 11 September 2011

Abstract Many animals can return home accurately after exploring for food using their own homing navigation algorithm. Path integration plays a critical role in homing navigation. It is believed that animals are able to recognize their relative location from the nest by accumulating both distance and direction experienced during their travel. We tested possible patterns of neuronal organization for a path integration mechanism. The neural networks consisted of a circular array of neurons, following population coding. We describe here a neural model of path integration involving a relatively small number of neurons and discuss how well the model operates for homing navigation. Robotic simulations suggest that a neural structure with only a few sensor neurons can successfully handle the path integration needed for homing navigation.

Keywords Path integration · Homing navigation · Coarse coding · Population coding · Circular array neurons

1 Introduction

Many animals can explore for food and return to their nest using their own navigation systems [1,2]. Among the various orientation systems encountered in the animal kingdom, path integration (or dead-reckoning) refers to navigation back to the starting point by integrating the movements performed. There are two main types of path integration: the *arthropod type*, in which turning angles are estimated with respect to a compass, and the *vertebrate type*, in which turning angles are estimated inertially. The present paper focuses on a model for the first type. Path integration is an orientation mechanism that is fed by two types of inputs: local movement direction (or change in direction) and local movement length. How this information is organized in neural networks has been an open question. We present a

D. Kim (✉) · J. Lee
Biological Cybernetics Lab, School of Electrical and Electronic Engineering,
Yonsei University, Shinchon, Seoul 120-749, South Korea
e-mail: daeeun@yonsei.ac.kr

neural model of path integration, where the first type of input (movement direction) is coarse-coded or population-coded, while the other (movement distance) is simply coded by the input activation intensity.

There are many animals and insects that use path integration for homing navigation. For example, the major axis of the fiddler crab, *Uca vomeris*, is always directed towards its nest [3,4]. This orientation means that the fiddler crabs are ready to return to their nest at any moment via a path calculated during travel. The crabs are known to draw some kind of path map and calculate the total distance between the current location and their nest by counting the total number of footsteps taken [4]. Rats are also believed to obtain navigational information from an integrating process. Their neural system guides spatial navigation based on path integration [5]. Another rodent species, blind moles, also uses path integration for home navigation by adopting geo-magnetic sensing [6]. Each of these animals has their own style of navigation behaviour and sensor system, but all share the common aspect of a path integration mechanism. Each integrates distance and relative direction during travel and then decides upon the primary direction by choosing the maximum integrated value. Desert ants, after exploring for food, choose the direction towards home from the food source and return home in almost a straight line, even if they travel a long distance. They measure the orientation of their movement relative to the sun azimuth. Desert ants are known to use path integration during homing, and demonstrate accurate homing from far away, even without landmarks near the nest or other environmental information [7].

Many navigation algorithms applied in robotics can be implemented with mathematical formulas that include trigonometric components and transcendental functions [1]. How animals handle those complex geometric functions with neural mechanisms is less understood, although they presumably use both egocentric and geocentric information for homing navigation [8]. Their path integration, that is, the egocentric information, is one of main resources for their decisions in navigation. Several neural models have been proposed so far to explain path integration. Wittmann and Schwegler [9] suggest that a sinusoidal array of neurons could handle path integration in an elegant way. The light compass information is encoded by peak activity in a ring structure of neurons where each cell represents a light compass direction. Another layer represents homing vectors and has incoming weights from the sensory layer and recurrent connection weights in the sinusoidal array itself to make short-term memories, thereby recording the homing vector. Specific neuron connection weights are required to keep the homing vector in memory. Alternately, Kubie and Fenton's work [10] uses head-direction accumulators for sensor activation, with each accumulator following a cosine tuning curve. The shortcut vectors connecting all pairs of visited waypoint locations are calculated with the head-direction accumulators, ultimately deriving the direct route from an arbitrary exploration point to the nest. In fact, head direction cells have been discovered in rodents [11,12]. A neural mechanism with head direction cells may be a plausible mechanism to explain the biological process of path integration. Similar to Wittmann and Schwegler's work, Kim and Hallam [13] describe two circular arrays of neurons (a variation of the sinusoidal array), where the first layer records the sensor readings for the current head direction relative to the light compass and the second layer represents the homing direction. A distribution of trigonometric weights allow for path integration. In this model, the sensor readings in the first layer choose the cell with maximum activation. The direction of the single cell indicates the current head direction and the cell records the motor actions. This model may have a resolution problem for head direction when the number of cells in the layer is relatively small.

We extended the approach of two circular arrays of neurons. The homing vector was represented with a set of neurons that could be narrow-tuning or broad-tuning. In particular, sensor activation as a head-direction accumulator may have coarse coding. We applied our neural structure to the homing navigation of a mobile robot in simulations, and the control structure with neural population coding achieved path integration, even for noisy sensor readings and motor actions. In addition, only a few sensor nodes in the sensor layer were needed to demonstrate homing navigation. A few output neurons, if they were broad-tuning, could also accurately determine homing direction. In our model, the head-direction cell uses the motor speed with an efference copy command to estimate the movement distance more accurately. We first introduce our method and neural network structure for path integration. Then robotic simulation experiments will be provided to support the approach.

2 Methods

In our simulation, the robot measures all steering angles and moving distances on an outward path and then integrates them to calculate a homing vector. We assumed a Khepera-style mobile robot with infrared sensors and ambient light sensors [14]. Such light sensors show their sensor readings based on light intensity, and the robot can detect the direction of the light. The light source plays a role of the reference compass, and the current head direction can be described with respect to the light source direction.

There are a limited number of sensory neuron cells that follow a cosine-(or Gaussian-) tuning curve for a specific head direction. A set of neurons represents a direction, and the voting mechanism is applied to the decision of the head direction. This is called population coding. The path integration mechanism has two layers: a sensory layer and homing vector layer. These two layers can estimate the homing vector with appropriate neuronal connections. The sensory layer represents the current head direction relative to the light compass, and the homing vector layer indicates the homing direction through population coding. We assumed that the activation of a head direction cell could match the activation of sensory cells for the light source. In the experiments, we simulated the sensor activations to follow a cosine tuning curve, reflecting the light direction. For our approach, the sensory layer includes a set of neurons to accumulate the neuron activation proportional to the motor speed for a specific moving direction. This neuron activation is then mapped into a set of circular neurons in the hypercolumn. This layered structure resembles the resource vector mapping from a set of sensory neurons to the resource direction [15]. The head direction cell described above is similar to the head-direction accumulator in Kubie and Fenton's neural model [10], and we propose that the firing rate of the cell is an integral of distance moved in a particular direction. Assuming the integral process is repeated over the instantaneous movement speed at regular time intervals, the cell encodes the traveled distance in a particular direction. Accordingly, the traveled distance can be better represented in our model.

For the k-th light sensor and its angular position θ_k, the first array of head-direction accumulator records the sensor activation z_k (the angular position of the k-th light sensor is not the same as that of the k-th head-directional cell for z_k). The head-directional cell in the circular array is found at the angle of $2\pi - \theta_k$ for the k-th light sensor readings. Here, the light source is a kind of reference compass and the mobile robot should know the current head direction angle relative to the light reference. For example, if a light sensor is positioned at 30° counterclockwise from the front (head direction) of the robot, the corresponding head-direction cell will have the maximum intensity when the robot heads towards −30° (330°) counterclockwise from the light source direction. The intensity depends on the angle between the light source direction and the normal direction of the sensor.

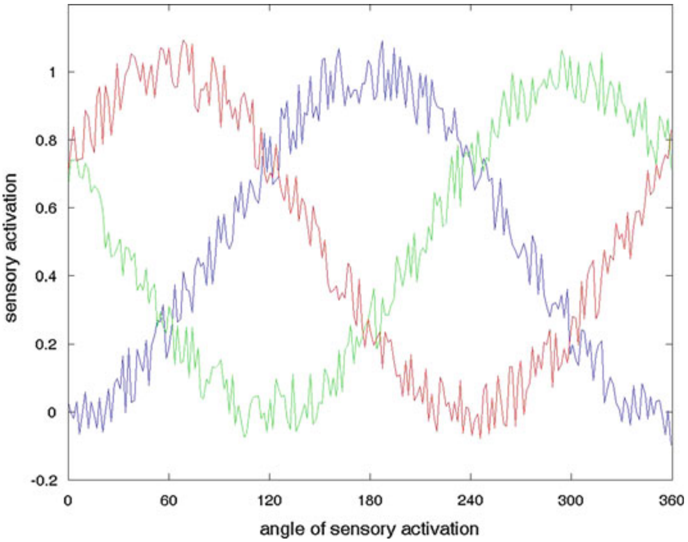

Fig. 1 Activation of a sensory cell (head-direction cell) in the first array (10% random noise was added); we assume three sensors positioned at angles 60°, 180°, and 300° on a robot platform, respectively

Each sensory cell (head-direction cell) in the first array has its own activation depending on the head direction relative to the light compass. Figure 1 shows an example of tuning curves for the three sensors positioned at angles 60°, 180°, and 300°, respectively. For example, if the light source is available at the direction of 180° relative to the head direction of a mobile robot, the activation values of the three sensors would be approximately 0.2, 1, and 0.2, respectively. In this manner, a set of these three values can determine the current head direction relative to the light direction. The activation of head direction cell is then multiplied with the current motor speed if the sensory activation is positive-valued, and the activation is accumulated to reflect the forward movement of the mobile robot. Faster motion produces higher activation of the sensory cell, indicating that the robot has traveled farther. That is, each sensory cell encodes the magnitude of distance that the robot travels in the preferred direction.

In fact, the two-layered networks for homing vectors can be decomposed into three layers [13], as displayed in Fig. 2a. The intermediate layer estimates the current location from the nest, which is represented as the Cartesian coordinate position (x, y).

$$\tau \dot{z}_k = -z_k + k_m m_{avg}$$
$$x = \sum_{k=1}^{N_1} z_k \cos(\theta_k), \qquad y = \sum_{k=1}^{N_1} z_k \sin(\theta_k) \tag{1}$$

where τ is the time constant for the sensory activation, k_m is the scaling factor, N_1 is the number of cells, z_k is the accumulated activation of the k-th cell (with the motor speed). Also, m_{avg} is the averaged movement speed of the robot body. It can be estimated as the average of the left and the right motor speed, $(m_L + m_R)/2$, where m_L, m_R are the left and right motor speeds, respectively and they can also be replaced by the motor speed commands. We encode the weight parameters with sinusoidal terms $cos(\theta_\kappa)$ and $sin(\theta_\kappa)$ and develop the home vector $(-x, -y)$ from the position (x, y). The home vector information is represented

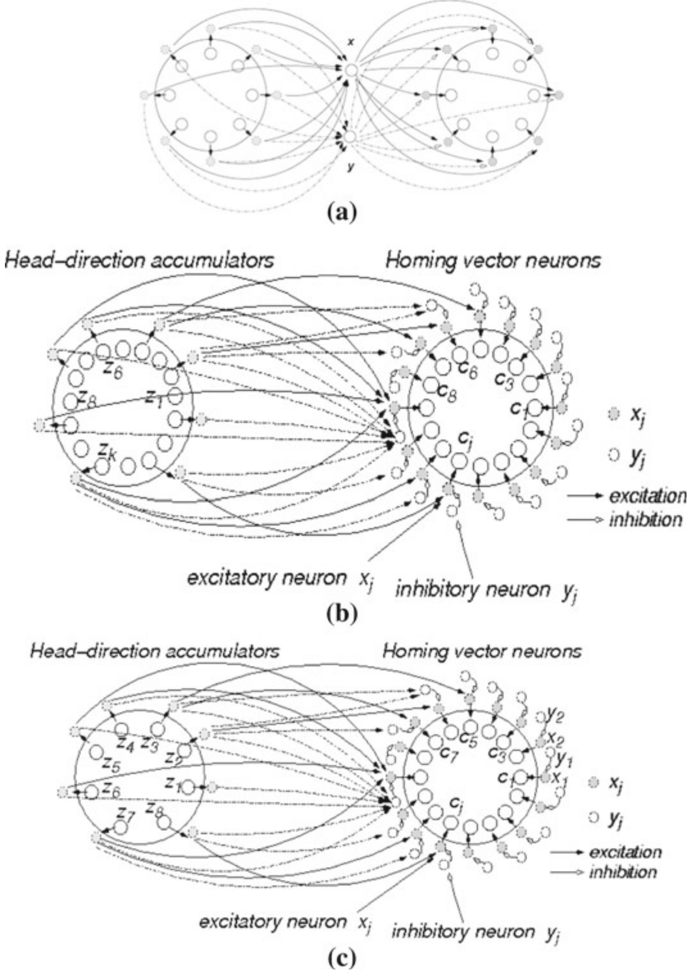

Fig. 2 Neural network structure from sensory layer to the home vector layer **a** Neural network architecture without direct mapping (reprinted from [13]) **b** Neural network architecture with population coding, when N_1 is as large as N_2 **c** when N_1 is smaller than N_2

with a population coding of neurons in the last layer. The layer includes uniformly distributed direction cells, and a direction cell fires maximally only when the estimated home direction is equal to the cell's preferred direction. We take the rotation of a point $(-x, -y)$ with angle θ to obtain a new (x', y'). If we obtain $y' = 0$ with a certain rotation angle θ, then the angle indicates the homing direction in the polar coordinate, and x' at the moment is the homing distance. These parameters give us a homing vector with a three-layer neural structure—Fig 2a. More detailed information is given in [13].

This three-layer network is equivalent to the two-layer structure described above. There are a number of cells for each layer, and the two layers take the form of circular array at regular intervals of the angles. Each head-direction accumulator in the first array plays a role of a sensor that is activated when excited by the external stimulus (in this case, the sunlight). The number of neurons in the sensor array determines the angular resolution. The homing vector is determined by mapping from the first layer to the second. The neuronal connection mapping determines the activation values based on population coding of sensory neurons, which is given as below:

$$\begin{aligned} x_j &= \cos\left(\alpha_j\right) \sum_{k=1}^{N_1} z_k \cos\left(\theta_k\right) + \sin\left(\alpha_i\right) \sum_{k=1}^{N_1} z_k \sin\left(\theta_k\right) \\ y_j &= -\sin\left(\alpha_j\right) \sum_{k=1}^{N_1} z_k \cos\left(\theta_k\right) + \cos\left(\alpha_i\right) \sum_{k=1}^{N_1} z_k \sin\left(\theta_k\right) \end{aligned} \tag{2}$$

where z_k is the neuron activation of the k-th cell in the first layer, θ_k is the direction of the k-th cell in the first layer, and α_j is the direction of the j-th cell in the second layer. We can obtain the activations, x_j, y_j for the j-th cell in the second layer.

Then we simplify the equation as follows:

$$\begin{aligned} x_j &= \sum_{k=1}^{N_1} \cos\left(\theta_k - \alpha_j\right) z_k = \sum_{k=1}^{N_1} \omega_{kj}^{x} z_k \\ y_j &= \sum_{k=1}^{N_1} \sin\left(\theta_k - \alpha_j\right) z_k = \sum_{k=1}^{N_1} \omega_{kj}^{y} z_k \end{aligned} \tag{3}$$

We allow a broad tuning (cosine or Gaussian) of the sensor readings, and the sensor readings are accumulated into $z_{k,}$ as time passes. The weight vectors include sinusoidal terms, as well as cosine and sine values.

The appropriate excitatory and inhibitory weights with sinusoidal terms determine the homing vector. This circular array of neurons projects the homing vector onto each of directional units in a polar form, while a circular array of inhibitory neurons corresponds to the orthogonal component of the projected vector. The cell with the maximum activation value in the second circular array of neurons is chosen by a winner-take-all method, and its activation value in the second layer is calculated [15] as follows:

$$c_j = \exp\left(-y_j^2/2\sigma^2\right) x_j \tag{4}$$

Here, the activation x_j is not directly transmitted into the neuron c_j and the exponential form of inhibition y_j plays a gate neuron [16] to control the activation of the neuron c_j. It has been conjectured that this kind of multiplication process can be implemented with a neural mechanism [17]. Gabbiani et al. [18] argue that there is evidence for the existence of a multiplicative operation in the neural system. Interestingly, the above neuron connection resembles the LGMD neuron (lobula giant movement detector neuron) in the locust visual system [18]. In this way, we can choose the cell with the maximum activation in the second layer.

$$j^* = \arg\max_j c_j = \arg\max_j \exp\left(-y_j^2/2\sigma^2\right) x_j \tag{5}$$

The direction of the neuron c_{j^*} with the maximum activation represents the homing direction and the amount of the neuron activation encodes the travel distance. The cell activation c_j in Eq. 4 will be the same as x_j value when $y_j = 0$. If y_j is not zero, then the inhibition term $exp(-y_j^2/2\sigma^2)$ decays the x_j value. From this property, the cell c_{j^*} with the maximum activation in Eq. 5 is the cell with the value y_j equal or close to zero. The direction of the cell c_{j^*} is the homing direction in the circular distribution of cells. The mobile robot compares the current direction and the homing direction, and then decides to turn left, right, or to go straightforward. A small level of activations at the cell c_{j*} implies that x_j and y_j are small, and the current position of the robot is quite close to the home location.

The neuron activation in the first layer forms a head-direction accumulator [10,13]. The activation is then transmitted into the second layer of neurons with synaptic weights. The incoming activation to the second layer with the neuronal weights calculates the contribution to the shortcut vector from a distribution of head-direction accumulators. The activation of each cell in the second layer shows the contribution level to the shortcut vector. Thus, the cell with the largest activation indicates the homing direction. Larger activation means more distance from the nest. As a result, the path integration mechanism can be represented as two layers of neural networks and the homing vector is computed in parallel with a set of neurons. A coarse coding of neurons can be applied to the circular arrays.

Here, we introduce three neural models for the path integration, depending on the number of cells for each layer of networks.

2.1 Model I

We consider two layers of neural architecture. As shown in Fig. 2b, the number of cells in the first layer, and the second layer are N_1, N_2, respectively. For the model I, N_1, N_2 are large numbers. A high angular resolution of sensor array is expected, and the number N_2 determines the resolution of homing directions. We assume a high resolution of homing directions in this model with a narrow tuning curve of activation.

2.2 Model II

In Model II, we test a relatively small number of cells in the first array (sensor array), but with a high angular resolution of homing vector in the second array. The number of cells in the first layer, N_1, is small compared to that in the second layer—see Fig. 2c. Here, the cells in the sensor array have a broad tuning curve of activation. Neighbor neurons will show degraded sensor readings around a high peak of activation. More distant neighbors will have smaller readings, which follows a coarse coding of neurons. The integrated distance is represented with a set of sensory neurons as the head direction accumulators. We assume that the second layer has a number of neuron cells to encode a high angular resolution of home vector. Here, we will test if a small number of cells, that is, a coarse coding of neurons in the first array can be effective for homing navigation.

The second layer has many neurons to pinpoint the homing direction. Each cell has a very narrow tuning curve of activation in this model.

2.3 Model III

Unlike the model I or model II, we can use a broad tuning of activations in the first and second layer. That is, a small number of neuron cells can be applied for each layer. We will test only three neurons in both the first layer and the second layer ($N_1 = N_2 = 3$). The broad tuning can be achieved by controlling σ^2 in Eq. 4. As a result, a collection of three neuron activations in the second layer determine the homing direction and distance.

3 Experiments

We simulated robotic homing navigation depending purely on path integration. The robot first explored the arena randomly to search for food for a while, and then returned to the nest place

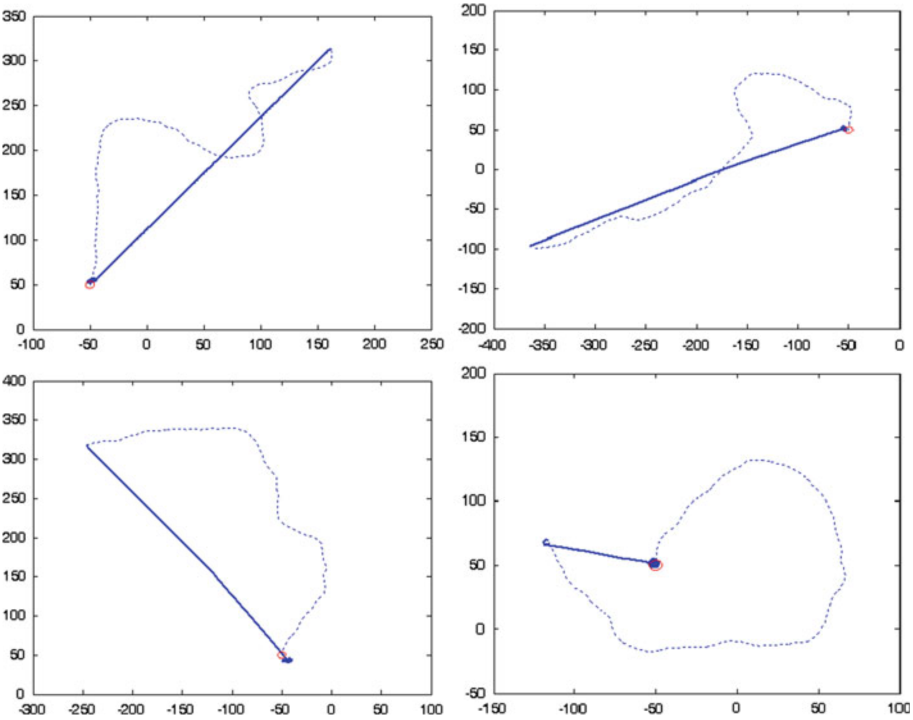

Fig. 3 Examples of path integration for homing navigation (*dotted* exploring path, *solid* returning path, *circle* nest); home is positioned at (−50 cm, 50 cm) and $N_1 = 3,\ N_2 = 100$

with a direct route. Here, 10% random noise was given to the sensor readings and 5% random noise to the motor speed command. Figure 3 shows examples of path integration for homing navigation without any use of other environmental features. The home nest was not always reached without error. The path integration mechanism updated the current direction with the light compass, and the directional error was corrected every time step. Thus, relatively small homing errors were observed with the suggested neural structure of path integration.

The mobile robot estimated the homing direction and homing distance for every movement, including both the exploration and return phases. Equation 5 finds the cell with the maximum activation, and that cell direction is the homing direction. The activation of the cell c_{j^*} is proportional to the homing distance. When the robot returned home and became closer to the home location, the homing distance as well as the corresponding cell activation became smaller. We set a threshold so that the robot would stop when the activation of the cell c_{j^*} was smaller than the threshold.

We varied the number of neuron cells, N_1, in the sensor array with a fixed number of cells in the second layer ($N_2 = 100$). With $N_1 = 2$, the robot had difficulty returning to the nest, because only two direction cells were available and at some moment, the robot had difficulty choosing the homing direction. The angular resolution in the sensor layer was too low. With $N_1 = 3$, the robot returned to the nest successfully without any problem. At least three directional cells were required to guide the robot. Here, it is notable that the robot could return home with only three cells in the sensor array. Neural models for path integration have assumed many neuronal cells in the sensor array. We found, however, that the performance with three cells was as good as with $N_1 = 100$ (Fig. 4b–c). When N_1 was larger than 2,

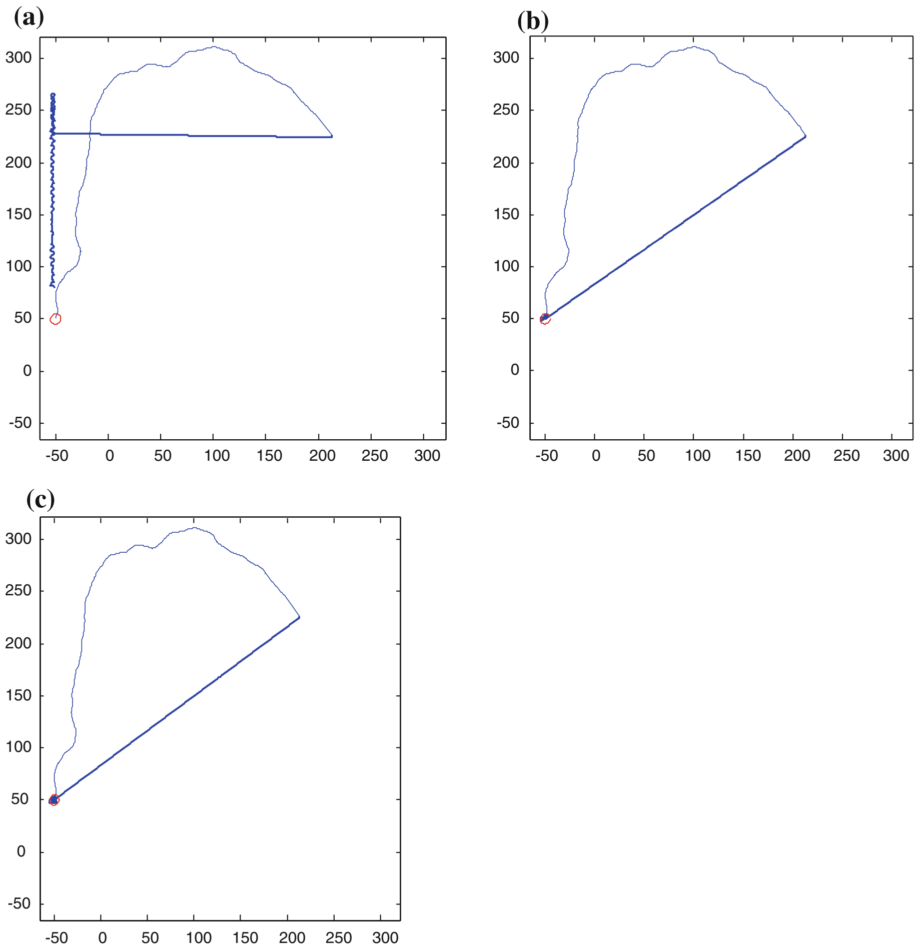

Fig. 4 Robotic movement for path integration depending on the number of cells in the sensor array (home: $(-50\,\mathrm{cm}, 50\,\mathrm{cm})$, *dotted* exploring path, *solid* returning path, $N_2 = 100$) **a** $N_1 = 2$ **b** $N_1 = 3$ **c** $N_1 = 100$

the trajectories were almost similar and the robot could accurately return to the nest by using path integration.

Figure 5 shows how sensor activation progressed. Here, we assumed that only three sensory neuron cells were available in the first layer. The exploration movement was done for 1000 time steps. We saw random movements in the exploration path in Fig. 5a and the corresponding sensor activations of the three neurons in Fig. 5c. For the returning path, each of the sensor activations was stabilized except for high-frequency noise (Fig. 5d). The accumulated motor activation for a particular direction in the first layer is displayed in Fig. 5b. The three activations met together at time step 3000, indicating that the robot had reached the nest position. The sensory array needed only a small number of neurons to effectively carry out path integration. With at least three neurons, linearly independent vectors could form to describe any vector in the path integration space. Interestingly, the suggested network efficiently handled noisy signals in the sensor activation through head direction accumulation and sinusoidal mapping into the second layer. Also for any arbitrary combination of paths,

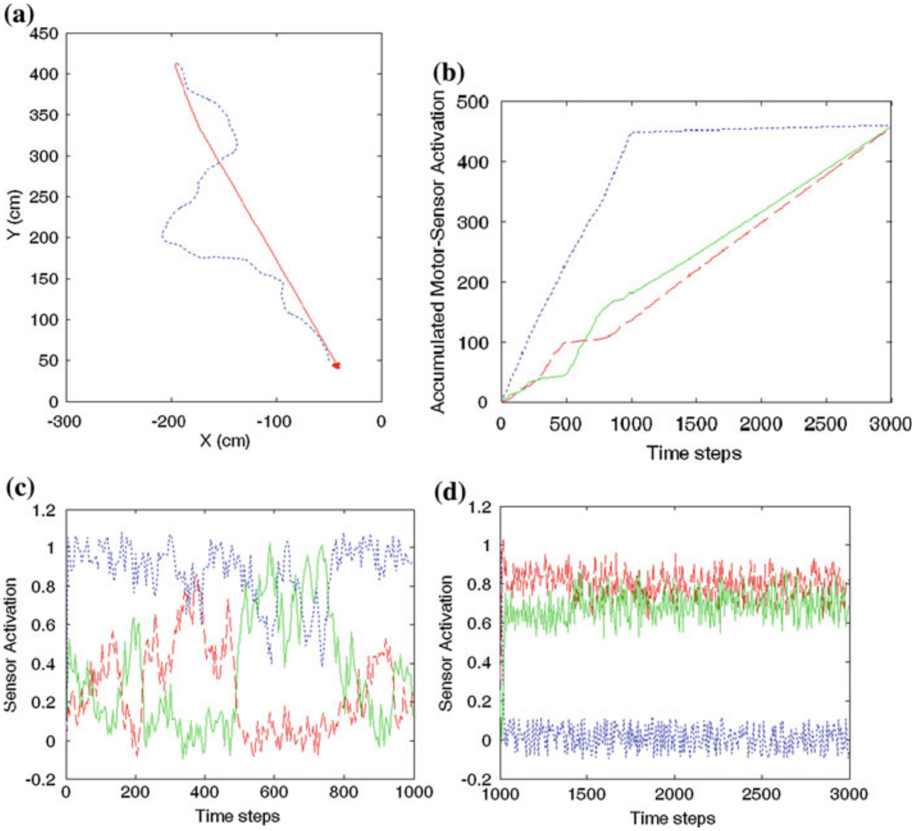

Fig. 5 Robotic movement and sensor activation **a** robotic movement (*dotted* exploration path up to 1000 time steps, *solid* returning path from 1001 to 4000 time steps) **b** accumulated sensor activation **c** sensor activation during exploration **d** sensor activation during returning movement (each cell has its own preferred angle, *dotted* 120°, *dashed* 240°, *solid* 360°)

even with varying motor speeds, the neural structure successfully demonstrates the homing navigation.

Figure 6 shows the path integration error according to the number of cells in the first layer. We evaluated the error by dividing the sum of area between the desired path and the actual path by the number of time steps used for returning home. The calculated error corresponds to the average deviation in the return path. The number of time steps in the return path was fixed to 3000 steps. The optimal return path (zero path integration error) follows a straight line from the food source to the nest. Here, we tested varying number of cells in the sensor array, N_1, from 3 to 100 with $N_2 = 100$. For each number of cells, 100 exploration paths were tested and the average performance was calculated (Fig. 6). Performance with $N_1 = 10$ was significantly better than with $N_1 = 3$, but did not improve further with more than 10 cells. From the experiments, we conclude that more sensory cells tend to reduce the path integration error. In other words, better reading of the head direction with more sensory cells can improve the performance of path integration.

When we changed the number of cells, N_2, in the second layer with narrow tuning, it influenced the performance ($\sigma = 10$ in Eq. 4). With a small number of neurons (N_2),

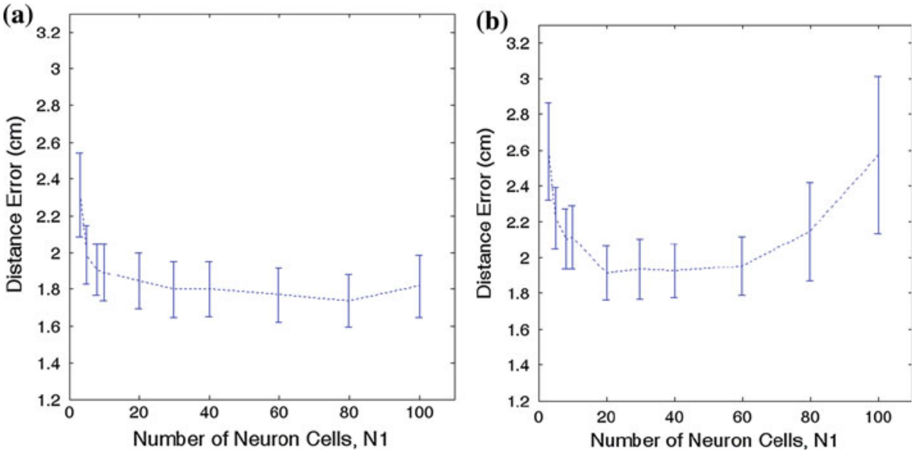

Fig. 6 Distance error between the nest position and the robot's final position after homing was dependent on the number of neurons in the first layer, $3 \leq N_1 \leq 100$ (average performance with 95% confidence intervals by the t-distribution is displayed) **a** $N_2 = 100$, **b** $N_2 = 50$

the performance degraded, and the distance error increased because the angular resolution became worse. Once the second layer has more than five cells, the robot showed good performance for returning home (Fig. 7c–d). Even with five neurons in the second array, the neural network guided the robot to the nest within the boundary of acceptable error. Figure 7b shows neuron activation in the second array with $N_2 = 30$. The specific neuron with the highest activation directly pinpoints the homing direction, and its activation level can determine the distance to the nest. The experimental results are affected by the fact that the homing vector cell in the second array has a narrow tuning curve of activation with large inhibitory exponentials in Eq. 4.

If there are a large number of neurons in the second array, more choices are available for homing direction. Alternately, we can have a small number of neurons but with a broad tuning curve for each cell (Model III). For example, the homing direction can be determined with only three neurons ($\sigma = 100$ in Eq. 4) in the second array. In this case, further processing is needed to decode a set of activations into a homing direction. A third layer can transform the broad tuning activation into the fine resolution of homing direction with the similar mapping neural network shown above (using Eqs. 3–4). Or population coding [19,20] can achieve accurate estimation of the homing direction with the equation given below:

$$ce^{i\phi} = \sum_{j=1}^{m} c_j e^{i\phi j} \tag{6}$$

where c_j is the firing rate for the j-th cell, φ_j is the angular position for the j-th cell and φ is the homing direction that the agent chooses. Possibly the maximum likelihood estimation method could be used to decode the population coding [21].

Figure 8 shows the robotic movement when a coarse coding of neurons is applied in both the first and second layer (model III with only three neurons for each layer). If three neurons in the second layer have narrow-tuning activations, the robot can choose a limited number of homing directions, but the robot can return home successfully with a sequence of possible choices. If the broad-tuning activation can be used, it finds more direct route to reach home. As an alternative, a mapping network from the second layer to the two motor neurons

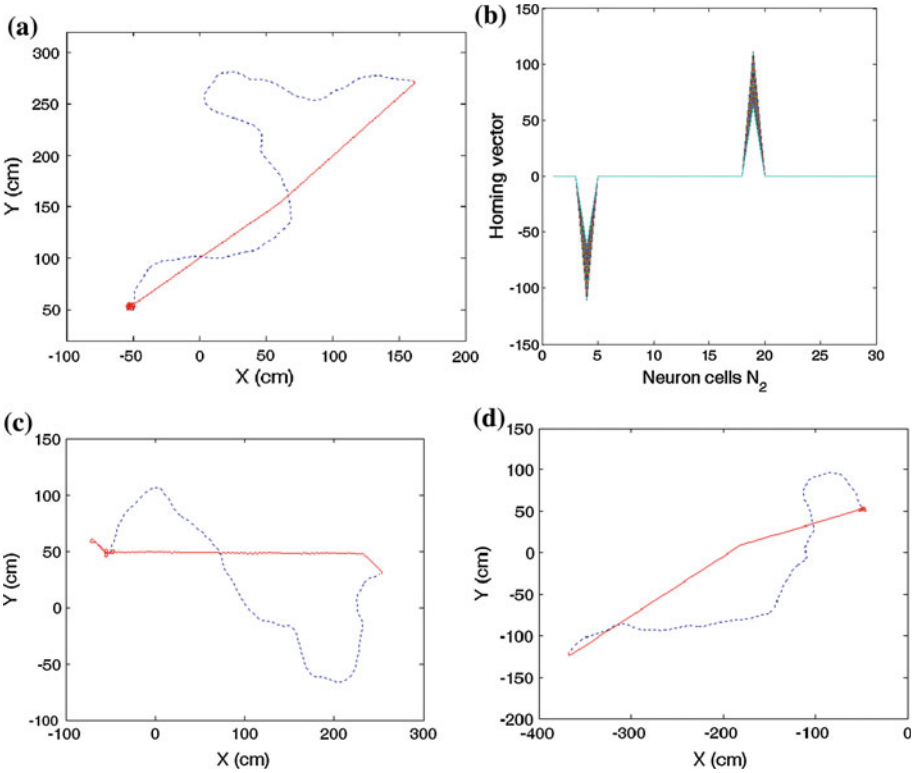

Fig. 7 Path integration with $N_1 = 3$ **a** robotic movement with $N_2 = 30$ (*dotted* exploration path, *solid* returning path) **b** neuron activation in the second array (19-th cell activation indicates homing vector) **c** robotic movement with $N_2 = 5$ **d** robotic movement with $N_2 = 10$

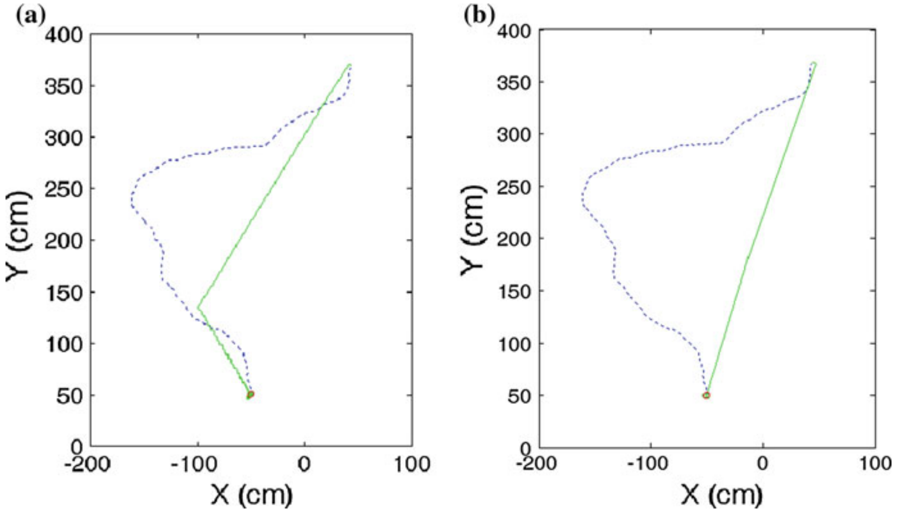

Fig. 8 Path integration with a coarse coding of neurons, $N_1 = N_2 = 3$ **a** robotic movement with narrow-tuning activation in the second layer **b** robotic movement with broad-tuning activation in the second layer (*dotted* exploration path, *solid* returning path, *circle* home)

(left and right motor wheel actions) could be considered. A coarse coding of neurons with continuous-valued activations includes sufficient information for homing navigation.

The sinusoidal array of neurons has been suggested to explain a directional cue with a population of neurons [22]. The array neurons have positive firing rates. Here, some homing vector neurons in the second circular array may have negative activations—see Eq. 4. For convenience, we used all the activations to estimate the homing direction and demonstrate the mathematical validity. The second circular array has symmetrical activations, positive and negative, according to the symmetrical neural structure. We can change the model, however, to restrict the neuron activations to only positive firing rates. We can add offset to each neuron activation to prevent negative firing rates such that the minimum value of the activation is nonzero. More easily, the neuronal connections from the first layer to the second array can be defined only for $|\theta_k - \alpha_j| \leq \pi/2$ in Eq. 4. Then every neuron cell will have non-negative activation. Those methods also do not influence the path integration performance.

4 Discussion

The core of a path integration algorithm is the accumulation of position and direction experienced during travel. Homing navigation systems based on path integration can easily be applied to mobile robots. The mechanism requires relatively little information about the robot's travel, position, and surrounding environment, and only uses the egocentric information of the efference copy command of motor actions. It is simpler, therefore, than methods based on complex mechanical and mathematical calculations that may allow more prompt homing navigation by a mobile robot. The mechanism is also useful when landmarks are unavailable or in an unfamiliar terrain far away from the nest.

Homing navigation may involve both egocentric and geocentric information. Here, we only tested the path integration mechanism. The combination of geocentric information [8] with the path integration process may improve homing navigation behavior. The addition of more information may facilitate robust decision making in the context of noisy sensor readings. Our neural model of path integration might be a plausible system for a biological neural model of animals, although it is simplified. Direct mapping from the accumulated sensor readings to the homing vector is a simple structure for encoding navigation information.

Based on our simulations, a minimum of three cells for head direction is sufficient for homing navigation. This finding implies that we should only need three light sensors for robotic experiments on path integration. More cells should work robustly for noisy environments. Kim and Hallam's approach to path integration [13] selects a single sensor cell for the current head direction information and corresponds to a narrow tuning of sensor activations. In contrast, we allow a broad tuning curve of sensor activations, which reduces the number of sensor nodes without influencing path integration performance. If the activation tuning curve was more narrow, as in some real-world settings, we would need more sensors for compass direction. Similar analysis can be applied to the homing direction cells in the second array. Three direction cells are sufficient even in the second array. A broad-tuning curve of the cell activations can give us more accurate homing direction.

Our network model is similar to Kubie and Fenton's work [10] in terms of the head-direction accumulator. We also assumed that the accumulator follows a cosine tuning curve. In our neural network model, however, the head-direction accumulator has neuron activation proportional to the motor speed. More speedy movement towards the head direction will have a larger activation in the corresponding neuron. Kubie and Fenton's work did not address motor-speed action. Wittmann and Schwegler [9] have also described circular neuron networks where each neuron follows a cosine tuning activation, suggesting that the cosine tuning activation is multiplied by the motor speed to calculate the path integration. The efference copy signal is transmitted into all of the neurons and a short-term memory with feedback connections is built for the homing vector. In contrast, the motor speed in our network contributes to a coarse coding of head-direction cells in the current direction. Accordingly, our suggested network structure becomes simple and has a feed-forward neuron structure. If the robot moves in a specific direction, the motor speed multiplied by the elapsed time provides the distance the robot has traveled in that direction. Considering all the directional movement, the neuron model can calculate the distance from the home location even for varying motor speeds. As a result, the robot can estimate the homing distance as well as homing direction. Every movement the robot calculates the direction and distance from the nest. When the distance becomes small, then robot can guess that it is near the home location. In addition, unlike Kubie and Fenton's network, there is no need to assume that head-direction accumulators follow a cosine tuning curve. With a sequence of different motor speeds, the accumulation activations become asymmetrical. The second layer of neurons automatically calculates the homing vector from those activations through the interaction of excitation and inhibition mechanisms.

Many researchers have pointed out that real animals show no perfect path integration for homing navigation [9,23,24]. Two-leg experiments show both distance and angular errors [9,23]. Müller and Wehner found that the polar-form approximation of path integration can fit angular errors in two-leg experiments with desert ants. Wittmann and Schwegler suggested that the ants start the integration process after travelling some amount of distance, which fits well to the experimental data [9,15]. Seguinot et al. [19] also argue that moving animals may overestimate the motion parallax of the starting point and generate a path integration error. In this paper, we suggest a new hypothesis to explain the distance and angular errors found in the path integration of animals [9,24]. We assume that the head-direction accumulator decays the activation as time passes, which is accomplished in this neural structure by decaying the activation level of the sensory cell in the first layer. For two-leg experiments, the first leg movement is older than the second. More decay on the estimated distance in the direction of the first leg movement can be expected, which may explain the angular and distance errors found in real animal experiments [9,24]. The systematic validation of our hypotheses with experimental data is needed, and is left for future work.

5 Conclusion

We suggest here a two-layer neural model with coarse coding for a path integration mechanism, and robotic simulation with this neural structure showed successful homing navigation. The first and second layers encode the head-direction accumulators and homing vectors, respectively. Each layer is represented with a population of neurons. Of note, only a few cells in the sensor array are needed at minimum to demonstrate homing navigation when a broad tuning curve of sensor activation is allowed as the head-direction accumulator calculated with motor speed. More sensory cells can reduce path integration error for noisy sensor readings. In addition, a few neurons in the second layer of the homing vector can successfully guide the robot to the nest within an error boundary under the advantage of broad tuning activations. Simulation experiments suggest that the path integration mechanism might be realized with a relatively small number of neuronal cells if they follow a broad tuning curve. Our neural model with coarse coding provides a simple sensory-motor loop structure for homing navigation that may be useful in real robotic experiments.

Acknowledgments This work was supported by the Human Resources Development of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government Ministry of Knowledge Economy (No.201040 10100590).

References

1. Mittelstaedt H (1983) The role of multimodal convergence in homing by path integration. Fortschr Zool 28:197–212
2. Wehner R, Wehner S (1990) Insect navigation: use of maps or Ariadne's thread?. Ethol Ecol Evol 2:27–48
3. Zeil J, Hemmi J (2006) The visual ecology of fiddler crabs. J Comp Physiol A 192:1–25
4. Layne JE, Barnes WJ, Duncan LM (2003) 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
5. Dudchenko PA, Bruce C (2005) Navigation without landmarks: can rats use a sense of direction to return to a home site?. Connect Sci 17(1–2):107–125
6. Kimchi T, Etienne AS, Terkel J (2004) A subterranean mammal uses the magnetic compass for path integration. Proc Natl Acad Sci USA 101:1105–1109
7. Wehner R (2003) Desert ant navigation: how miniature brains solve complex tasks. J Comp Physiol A 189:579–583
8. Wehner R, Michel B, Antonsen P (1996) Visual navigation in insects: coupling of egocentric and geocentric information. J Exp Biol 199:129–140
9. Wittmann T, Schwegler H (1995) Path integration—a network model. Biol Cybern 73:569–575
10. Kubie JL, Fenton AA (2009) Heading-vector navigation based on head-direction cells and path integration. Hippocampus 19:456–479
11. Ranck JB Jr (1985) Head direction cells in the deep cell layer of dorsal presubiculum in freely moving rats. In: Buzsaki G, Vanderwolf CH (eds) Electrical activity of the archicortex. Hungarian Academy of Sciences, Budapest, pp 217–220
12. Taube JS, Muller RU, Ranck JB Jr (1990) Head-direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis. J Neurosci 10:420–435
13. Kim D, Hallam JCT (2000) Neural network approach to path integration for homing navigation. From Animals to Animats 6:228-235 MIT Press, 2000
14. Loffler A, Klahold J, Ruckert U (2001) The mini-robot Khepera as a foraging animate: synthesis and analysis of behavior. Auton Minirobots Res Edutainment AMiRE 97:93–130
15. Kim D (2006) Neural network mechanism for the orientation behaviour of sand scorpions towards prey. IEEE Trans Neural Netw 17(4):1070–1076
16. Haykin S (1999) Neural networks, a comprehensive foundation. 2. Prentice Hall Inc, New York
17. Koch C (1999) Biophysics of Computation. Oxford University Press, New York
18. Gabbiani F, Krapp HG, Koch C, Laurent G (2002) Multiplicative computation in a visual neuron sensitive to looming. Nature 420(6913):320–324
19. Georgopoulos AP, Kalaska JF, Caminiti R, Massey JT (1982) On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. J Neurosci 2(11):1527–1537
20. Lewis JE (1999) Sensory processing and the network mechanisms for reading neuronal population codes. J Comp Physiol A 185:373–378
21. Seung HS, Sompolinsky H (1993) Simple models for reading neuronal population codes. Proc Natl Acad Sci USA 90:10749–10753
22. Redish AD, Touretzky DS, Wan HS (1994) The sinusoidal array: a theory of representation for spatial vectors. In: Eeckman FH (ed) Computation in neurons and neural systems. Springer, Berlin, pp 269–274
23. Seguinot V, Cattet J, Benhamou S (1995) Path integration in dogs. Anim Behav 55:787–797
24. Müller M, Wehner R (1988) Path integration in desert ants. *Cataglyphis fortis*, PNAS 85(14):5287–5290