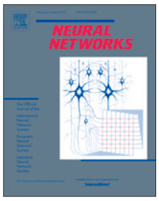
Neural Networks 165 (2023) 106–118

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



An insect-inspired model facilitating autonomous navigation by incorporating goal approaching and collision avoidance

Xuelong Sun [a,b,1], Qinbing Fu [a,b,1], Jigen Peng [a,b,*], Shigang Yue [c,d,*]

[a] School of Mathematics and Information Science, Guangzhou University, Guangzhou, 510006, China
[b] Machine Life and Intelligence Research Centre, Guangzhou University, Guangzhou, 510006, China
[c] Computational Intelligence Lab (CIL)/School of Computer Science, University of Lincoln, Lincoln, LN6 7TS, United Kingdom
[d] School of Computing and Mathematical Sciences, University of Leicester, Leicester, LE1 7RH, United Kingdom

ARTICLE INFO

Article history:
Received 23 May 2022
Received in revised form 17 March 2023
Accepted 17 May 2023
Available online 24 May 2023

Keywords:
Path integration
Collision detection-and-avoidance
Vector-based navigation
Insect navigation
Sensory–motor
Lobula giant movement detector (LGMD)

ABSTRACT

Being one of the most fundamental and crucial capacity of robots and animals, autonomous navigation that consists of *goal approaching* and *collision avoidance* enables completion of various tasks while traversing different environments. In light of the impressive navigational abilities of insects despite their tiny brains compared to mammals, the idea of seeking solutions from insects for the two key problems of navigation, i.e., *goal approaching* and *collision avoidance*, has fascinated researchers and engineers for many years. However, previous bio-inspired studies have focused on merely one of these two problems at one time. Insect-inspired navigation algorithms that synthetically incorporate both *goal approaching* and *collision avoidance*, and studies that investigate the interactions of these two mechanisms in the context of sensory–motor closed-loop autonomous navigation are lacking. To fill this gap, we propose an insect-inspired autonomous navigation algorithm to integrate the *goal approaching* mechanism as the global working memory inspired by the sweat bee's path integration (PI) mechanism, and the *collision avoidance* model as the local immediate cue built upon the locust's lobula giant movement detector (LGMD) model. The presented algorithm is utilized to drive agents to complete navigation task in a sensory–motor closed-loop manner within a bounded static or dynamic environment. Simulation results demonstrate that the synthetic algorithm is capable of guiding the agent to complete challenging navigation tasks in a robust and efficient way. This study takes the first tentative step to integrate the insect-like navigation mechanisms with different functionalities (i.e., global goal and local interrupt) into a coordinated control system that future research avenues could build upon.

1. Introduction

Navigation is a fundamental and crucial ability for not only animals in nature which need foraging, homing and finding mates, but also mobile robots which are required to approach task-specified goals. Being the most common problem of robot navigation, path planning involves the determination of a collision-free and low-cost path from the start point to the target point (Gul, Rahiman, & Nazli Alhady, 2019; Raja & Pugazhenthi, 2012). This reflects the two key problems in navigation: ***goal approaching*** and ***collision avoidance***. To solve these two problems, apart from engineering techniques, for instance, the Dijkstra algorithm (Hart, Nilsson, & Raphael, 1972; Soltani, Tawfik, Goulermas, & Fernando, 2002) and A-star (Duchoň et al., 2014) used in global navigation and the soft computing methods like neural networks (Singh & Thongam, 2019; Yang & Meng, 2000) and fuzzy logic (Faisal, Hedjar, Al Sulaiman, & Al-Mutib, 2013; Omrane, Masmoudi, & Masmoudi, 2016) (for reviews see: Algabri, Mathkour, Ramdane, & Alsulaiman, 2015) applied in local navigation, the nature has provided fantastic inspirations in light of the fact that many animals are highly capable navigators. Among the animals' kingdom, insects stands out by their impressive feats of navigation in spite of their tiny brain possessing less than 1 million neurons. Navigation in insect has been reported to cover most of the cognitive mechanisms like learning, memory, decision-making in an astonishingly efficient way (Menzel & Giurfa, 2001; Srinivasan, 2010). The distinct contradictory of strictly constrained computation resources versus the efficient and robust behaviors really motivates researchers to pursue insect-inspired solutions for navigation problem. Thus an insect-inspired method could result in a most efficient navigation model for robots (Dupeyroux,

[*] Corresponding authors at: Guangzhou University, No. 230, Outer-ring West Road Guangzhou Higher Education Mega Center, Guangzhou, 510006, China.
E-mail addresses: jgpeng@gzhu.edu.cn (J. Peng), sy237@leicester.ac.uk (S. Yue).

[1] Xuelong Sun and Qinbing Fu are joint first authors.

https://doi.org/10.1016/j.neunet.2023.05.033

Serres, & Viollet, 2019; Franceschini, 2014; Quinn, Nelson, Bachmann, Kingsley, Offi, & Ritzmann, 2001; Webb, 2020; Wystrach & Graham, 2012).

For ***goal approaching***, insects have exhibited a rich array of behaviors across different sensory domains (e.g. olfactory: Álvarez-Salvado et al., 2018; Jung, Hueston, & Bhandawat, 2015, vision: Collett & Collett, 2002, etc.), among which navigation is most typical and crucial (Buehlmann, Mangan, & Graham, 2020). Insects like ants and bees have navigation abilities rival those of mammals (Webb & Wystrach, 2016). For example, desert ants can travel hundreds of meters when foraging (food site as the goal) and return accurately to their nest (nest site as the goal) (Wehner, 2003). Recently published neurobiological data (Hulse et al., 2021; Sayre, Templin, Chavez, Kempenaers, & Heinze, 2021) and neural models (Stone et al., 2017; Sun, Yue, & Mangan, 2020) have greatly put forward the understandings of the neural basis and computation requirements underlying the robust and efficient navigation behaviors in insects (Heinze, 2017; Honkanen, Adden, da Silva Freitas, & Heinze, 2019), especially the path integration (PI) whereby foragers track the distance and direction to their nest by integrating the series of directions and distances traveled (Stone et al., 2017; Wehner, 2003). The core mechanism of PI reveals the bio-plausible computation basis of vector-based navigation (Le Moël, Stone, Lihoreau, Wystrach, & Webb, 2019; Lyu, Abbott, & Maimon, 2022). All these recent advances in insect navigation indeed provides a great opportunity beyond early studies seeking natural solutions towards robot navigation (Goldschmidt, Manoonpong, & Dasgupta, 2017; Lambrinos, Möller, Labhart, Pfeifer, & Wehner, 2000; Möller, 2000).

For ***collision avoidance***, the prerequisite is the reliable ability of collision detection (Mukhtar, Xia, & Tang, 2015). Recognizing approaching objects on a collision course, timely and accurately, is critically important for animals and also robots in either self-maneuver or flocking (Fu, Wang, Hu, & Yue, 2019). In this regard, insects like flies and locusts have demonstrated amazing capabilities of collision detection and avoidance in navigation (Muijres, Elzinga, Melis, & Dickinson, 2014), for example, a locusts swarm migrating for hundreds of miles free of collision (Kennedy, 1951). Compared to mammals and humans, insects possess much more simplified neural circuits for collision perception which involve only hundreds-to-thousands of neurons (Franceschini, 2014). The elaborate and compact neural systems of insects inspired engineers to build parsimonious collision detection visual systems for autonomous navigation of robots (Fu, Hu, Liu and Yue, 2018). More specifically, a group of lobula giant movement detectors (LGMD) located in the locust's preliminary visual circuitry has been found to act as the basis of collision perception (Gabbiani, Krapp, Koch, & Laurent, 2002; Rind et al., 2016, 2016). The LGMD has been studied and modeled extensively in robotic systems for guiding quick collision avoidance in navigation (Cizek & Faigl, 2019; Fu, Sun, Liu, Hu, & Yue, 2021; Salt, Howard, Indiveri, & Sandamirskaya, 2020). The computational simplicity and robustness make the LGMD models competitive with other vision-based collision detection methods (Fu, Hu, Peng, Rind, & Yue, 2020; Fu, Hu, Peng and Yue, 2018).

Although the aforementioned mechanisms that endow insects with the ability of solving ***goal approaching*** and ***collision avoidance*** have been modeled and even implemented onto robots, they were investigated and verified separately. That is to say, previous studies only take one mechanism into account at a time, i.e., approaching goals in a obstacles-free environment or detecting/avoiding collisions without planned destination. In this case, two questions herein are raised for insects inspired autonomous navigation: (1) *how these two mechanisms could be combined by a simple mechanism?* (2) *could these two mechanisms be integrated to build a valid autonomous navigation system?* To answer these questions, in this paper, we combine a PI model and an LGMD model into a navigation control system by taking their advantages as global working memory and local interactive cue respectively. The performance of the proposed navigation system is verified by the simulations wherein an agent implementing the proposed navigation algorithm is required to complete specific navigation tasks within a static or a dynamic environment in which the positions of the obstacles vary with respect to time.

The proposed navigation system reflects the minimal model that could facilitate truly autonomous navigation involving working memory gaining/retrieving and local immediate information processing. Our simulation results demonstrate the effectiveness of the navigation model. Our main contributions are threefold:

1. The first minimal model integrating insect ***goal approaching*** and ***collision avoidance*** mechanisms facilitates flexible vector-based navigation in static and dynamic simulated world.
2. The locust-inspired LGMD model greatly improves the performance of collision avoidance in the defined scenario, demonstrating its potential application for robot navigation in closed-loop manner.
3. The sweat bee inspired path integration model acting as a goal approaching mechanism still works well when interrupted by the collision avoidance, arguing that global working memory could cooperate with local cues with a reasonable motion control strategy.

Taken together, this paper presents the minimal insect-inspired autonomous navigation model that could drive the agent to complete navigation tasks in a sensory–motor closed-loop manner. Being as the basic building blocks of animal-like navigation system, this model exemplified the idea of learning from nature, learning the way how animals efficiently take great advantages of perceptible cues to make intelligent decisions.

2. Simulated environment and agent definition

All simulations are implemented by Python 3.9 and making use of its packages-*numpy*, *matplotlib*, *scipy*, *PIL* and *cv2*. All the source code is available via Github.

2.1. Simulated 3D world

The simulated environment consists of cubic obstacles and manually defined home (or start point) and food sites (they all could be goal positions under specific circumstances, for details see Section 2.3). The size of the cubic obstacles are determined by its side lengths which are sampled from normal distribution $N(0.05, 0.005)$. An example of the environment's settings used in this study is shown in (a) of Fig. 1, where there are 12 obstacles, 3 food sites positioned at three points of a equilateral triangle with side length of 400-steps in the x-y plane. The start point is positioned at [0, 0, 0] which is the bisection of the vertical line of the triangle shaped by the three food sites.

2.1.1. Static obstacles

The positions of the obstacles $P_{obs} = [x, y, 0]$ (the left-bottom point of the cubes) are set to be in the equally-divided points (e.g., bisection, trisection points, etc.) of the lines connecting home and food sites, and different food sites given the number of obstacle. For instance, in a world with 3 food sites and 12 obstacles, there will be 2 obstacles at the trisection points of the lines connecting each two food sites, and 2 obstacles at the trisection of the lines connecting the nest and each food site (see (b) of Fig. 1). Note that the low-left point of the obstacle cube is set at the above described points rather the center so it looks a bit upper-right offset in Fig. 1(b) and (c).

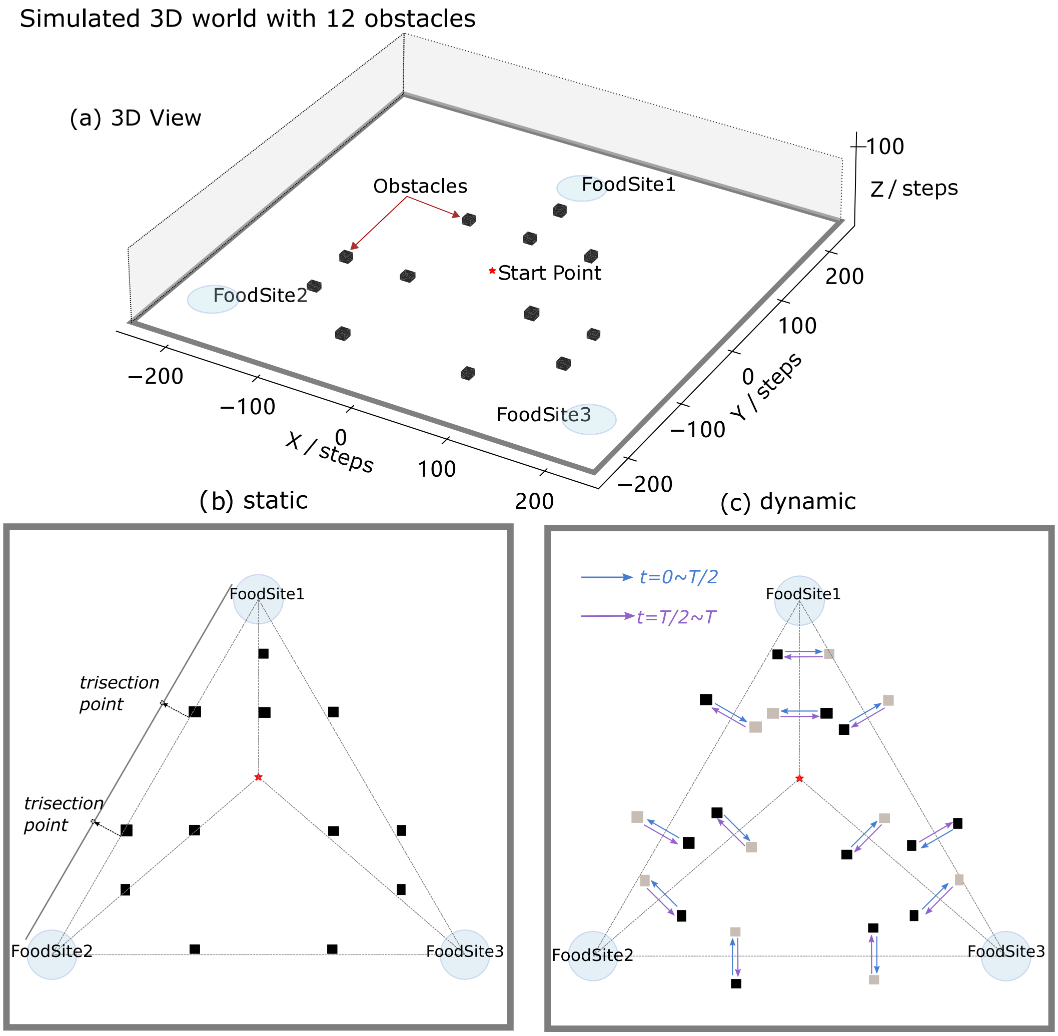


Fig. 1. An example of the simulated 3D environment (3 food sites and 12 obstacles) where the agents navigate in. (a) is the 3D view of the environment; (b) is the bird-eye view of static version of the example environment; (c) is the bird-eye view of the dynamic version of the environment where the all obstacles moves continuously. Direction of the obstacles' moving are indicated by the colored arrows showing corresponding time period.

2.1.2. *Dynamic obstacles*

In the setting of dynamic environment, the obstacles are moving back and forth and perpendicularly to the line in which they are positioned at a constant speed (see (c) of Fig. 1). The speed of the moving obstacles could be determined by setting the motion period and the distance moved, in this study, the period is 100 time-steps and the length is 50 steps (see Video S2 for more details). Note that the way we setting up the moving patterns of the obstacles raises the possibility of the agent's collision with the moving obstacles compared with randomly moving, see Video S3 in the supplementary material for an example of the agents' navigating in a dynamic simulated world with randomly moving obstacles.

2.2. *Agent definition and camera model*

The agent in this study is abstracted to be a directional point in the 3D space whose state is determined by $[x, y, z, \theta]$. A camera is embedded on the agent with the horizontal field of view 120° from $\theta - 60°$ to $\theta + 60°$ and vertical field of view 105° from $-15°$ to 90°. Examples of the camera data captured from the 3D simulated world depicted in Fig. 1 are shown in Fig. 2.

The settings of the environment and agent implement a closed-loop scenario where the agent's motion will effect its views in the environments and the views acted as the visual inputs in turn determines the motion control. Thus, this provides us with a simple but sufficient system for investigating the minimal requirements of a bio-inspired autonomous navigation model that leverages both *goal approaching* and *collision avoidance*.

2.3. *Navigation task definition*

As shown in Fig. 3, there are two phases of the navigation task that agents are required to complete: (1) In Phase 1, the agent should explore the unknown environment and after its finding the food, it should return to the start point. This process should be resumed until all the food sites are visited; (2) In Phase 2, the agent is required to start navigation from the same start point and after visiting all the food sites as remembered in Phase 1, go back to the start point. Note that there is no obstacle in Phase 1, while in Phase 2, the agent should not only approach goals determined by the vector memories but also avoid collisions with the obstacles. This setting is in consideration of the efficiency of running simulations: in Phase1, the agent is randomly wandering in the arena, environment with obstacles will greatly increase the time consumed to find all food sites but bring no significant effects to the accumulated vector memories (data is not shown). In addition, considering that there is no visual memories involved in the proposed navigation algorithm, changing the visual scene will not affect navigation performance.

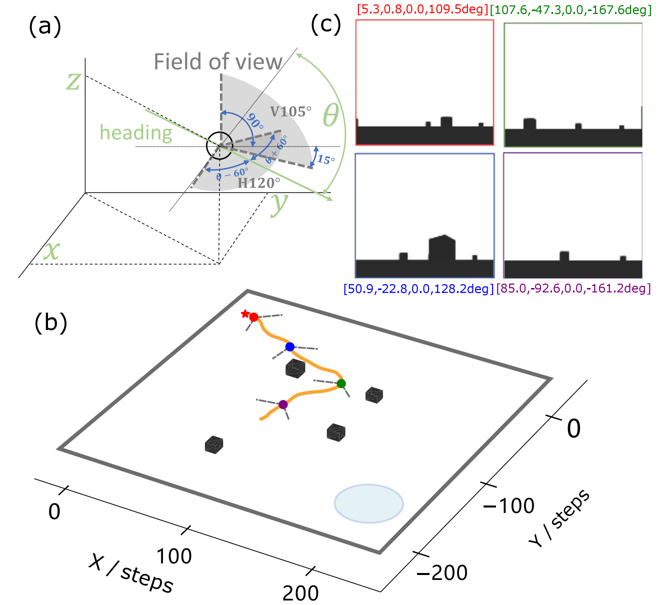


Fig. 2. The agent and camera model. (a) The agent with current state $[x, y, z, \theta]$ and the camera with 120° horizontal field of view and 105° vertical field of view. (b) The example route traveled by the agent in simulated 3D world (part of that in Fig. 1) and four sample points. (c) The view (camera data) captured at the sample points (color-coded) shown in (b) with $[x, y, z, \theta]$ labeled.

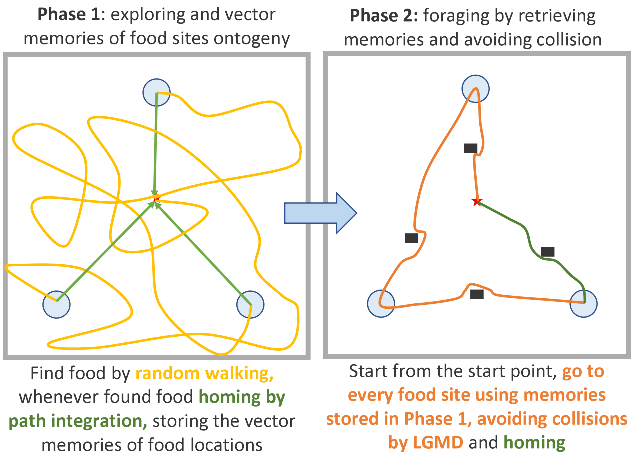

Phase 1: exploring and vector memories of food sites ontogeny
Phase 2: foraging by retrieving memories and avoiding collision
Find food by random walking, whenever found food homing by path integration, storing the vector memories of food locations
Start from the start point, go to every food site using memories stored in Phase 1, avoiding collisions by LGMD and homing

Fig. 3. The navigation task defined in this paper consists of two phases. Light-blue circles indicates the catchment areas of the food, black rectangles mark the shapes and locations of the obstacles.

3. Insect inspired navigation algorithm

Insects have fascinated researchers for hundreds of years by the mystery that how could such tiny brain solve complex tasks? Studying insects' neural network capable of solving navigation problems, i.e., goal approaching and collision avoidance, may bring insights into the efficient computation and usage of sensory information. The proposed model is developed from insects' neural networks aiming to complete navigation tasks in an insect-like, efficient way.

3.1. Bee path integration model

As the most fundamental navigation mechanism of insects, path integration (also known as *dead reckoning*) tracks the distance and direction to their start point by integrating the series of directions and distances traveled (for reviews see Collett, 2019; Heinze, Narendra, & Cheung, 2018). Stone et al. has recently published an anatomically constrained model unraveling a potential substrate for sweat bee's path integration (Stone et al., 2017) and has been extended for vector-based navigation (Le Moël et al., 2019). We adapted this model as the goal approaching algorithm. Specifically, as shown in Fig. 4, the home vector (pointing to the goal) stored as the neural activation of CPU4 neurons serving as the working memory to direct the agent's motion.

Note that in the following formulas, C_{name}^{index} denotes the membrane potential (output) of the neurons with its name and index while I_{name}^{index} denote the current (input). A Sigmoid activate function is applied to calculate C from I, i.e., $C = sigmoid(I)$.

3.1.1. Basic model

The core of the path integration model proposed in Stone et al. (2017) is that CPU4 neurons receives both the direction and velocity signal from upstream neurons TB1 and TN respectively. For TB1 neurons, its activation is updated as:

$$I_{TB1}^{j}(t) = (1 - c)(1.0 - C_{TL}^{j}(t)) + c\sum_{i=1}^{8} W_{TB1}^{ij} C_{TB1}^{j}(t-1) \tag{1}$$

where $c = 1/3$ is a balance factor to modify the strength of the inhibition and the excitation to make the network stable. TL neurons ($C_{TL} = \cos(\theta_{TL} - \theta_h)$) receive the heading input θ (sensory information assumed accessible) with its preference $\theta_{TL} \in \{0, \pi/4, \pi/2, 3\pi/4, \pi, 5\pi/4, 3\pi/2, 7\pi/4\}$. The 8 TB1 neurons per se organized as a ring attractor (Kim, Rouault, Druckmann, & Jayaraman, 2017; Turner-Evans et al., 2020) with mutual connections:

$$W_{TB1}^{ij} = \frac{\cos(\theta_{TB1}^{i} - \theta_{TB1}^{j}) - 1}{2} \tag{2}$$

where θ_{TB1} is the preference direction of TB1. Thus, the population coding $C_{TB1}^{t,j}, j = 0, 1, \ldots, 7$ represents the heading of the agent at time t.

For the velocity sensitive TN neurons,

$$\begin{cases} I_{TN2_L} = [\sin(\theta_h + \theta_{TN2}), \cos(\theta_h + \theta_{TN2})]\boldsymbol{v} \\ I_{TN2_R} = [\sin(\theta_h - \theta_{TN2}), \cos(\theta_h - \theta_{TN2})]\boldsymbol{v} \end{cases} \tag{3}$$

where $\boldsymbol{v}$ is the velocity (see Eq. (21)) of the agent and θ_{TN2} is the preference angle of the TN2 neurons. In this study $\theta_{TN2} = \pi/4$. The activation function applied to TN2 neurons (TN1 neurons are not modeled here) is the rectified linear function given by:

$$C_{TN2} = \max(0, 2I_{TN2}) \tag{4}$$

Then the firing rate of the CPU4 neurons are updated by:

$$I_{CPU4}(t) = I_{CPU4}(t-1) + r(C_{TN2}(t) - C_{TB1}(t) - k) \tag{5}$$

where the rate of the memory accumulation $r = 0.005$; the memory loss $k = 0.1$; the initial memory charge of CPU4 neurons $I_{CPU4}(0) = 0.5$. Therefore, the 16 CPU4 neurons represented the current coding of the home vector pointing to the goal direction. The initial value of CPU4 neurons is set to be 0.5, i.e., $I_{CPU4}(t) = 0.5$. The steering neurons, i.e., CPU1 neurons ($C_{CPU1}^{i}, i = 1, 2, 3 \ldots 16$) receive excitatory input from the CPU4 neurons and inhibitory input from the current heading (TB1 neurons) to generate the turning signal:

$$C_{CPU1}^{i}(t) = \begin{cases} I_{CPU4}^{i-1}(t) - I_{TB1}^{i}(t) & i = 1, 2, \ldots, 7 \\ I_{CPU4}^{i+1}(t) - I_{TB1}^{i}(t) & i = 8, 2, \ldots, 16 \end{cases} \tag{6}$$

Note that the one-column left/right shifting of the connections from left/right CPU4 to left/right CPU1 (left neurons with index of $i = 1, 2, ..7$ and right neurons with index of $i = 8, 9, ..15$)

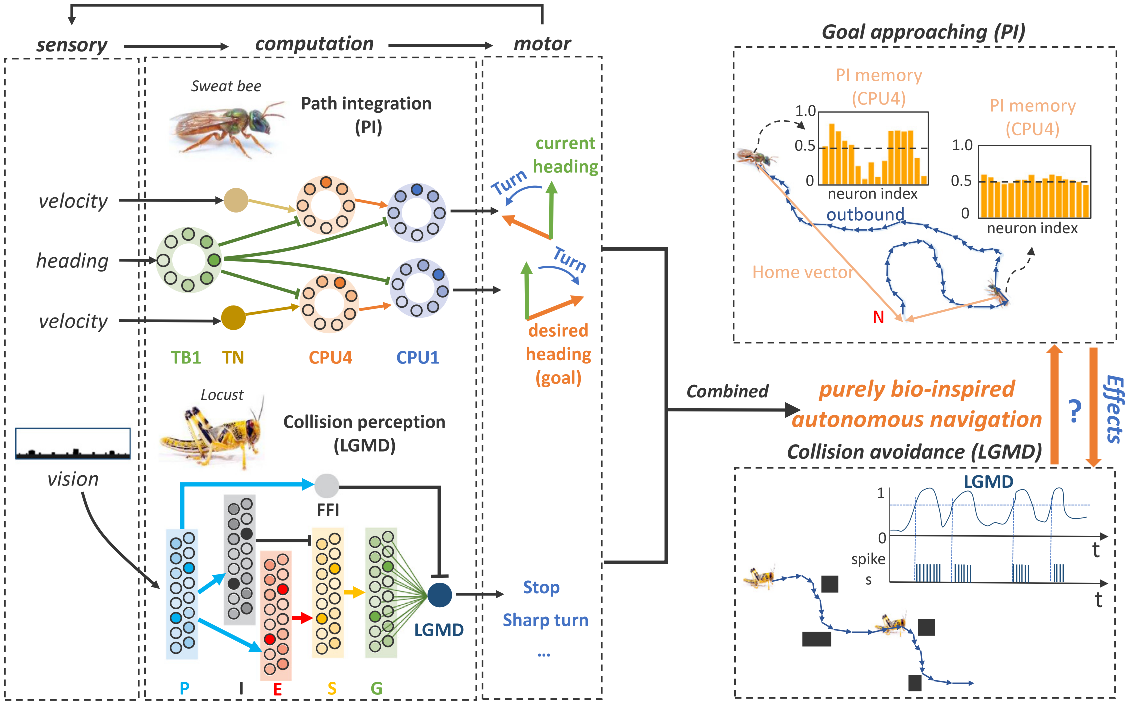


Fig. 4. The path integration model of sweat bee and collision detection model of locust underlying the goal approaching and collision avoidance algorithm of the proposed autonomous navigation. Left panel shows the neural network of PI and LGMD and the corresponding sensory–motor closed-loop. Right panel gives examples of goal approaching and collision avoidance and the neural activation as the model outputs at specific time steps.

realizing the measurement of the difference between the population coding of the vector represented in CPU4 neurons and CPU1 neurons. Therefore, the difference between left and right CPU1 neurons determines the ideal turning:

$$M_{PI}(t) = k_{PI}(\sum_{i=0}^{7} I_{CPU1}(t) - \sum_{i=8}^{16} I_{CPU1}(t)) \tag{7}$$

where $k_{PI} = 0.25$ is a factor to adjust the strength of the motor output.

3.1.2. Vector-based navigation

As the aforementioned path integration model will drive the agent back to the start point, i.e., reducing the CPU4 memory via the steering circuits composed of left and right CPU1 neurons. Thus, this could be the basis for vector-based navigation by taking the CPU4 memory (see Eq. (5)) as the working memory encoding the goal-vector. The idea in Le Moël et al. (2019) makes foraging, multi-location, shortcut navigation become realistic by applying inhibitory synapse on the axon between CPU4 and CPU1. To simplify this implementation, here we directly change the CPU4 memory given the state of the agent to realize vector-memory based navigation. Specifically, as the CPU4 neurons in PI encodes the vector of start point, we inverse the CPU4 to be the working memory $WM(t)$ for foraging:

$$WM(t) = \begin{cases} I_{CPU4}(t) & homing \\ 1.0 - I_{CPU4}(t) & foraging \end{cases} \tag{8}$$

and then Eq. (6) should be re-written as:

$$C^i_{CPU1}(t) = \begin{cases} WM^{i-1}(t) - I^i_{TB1}(t) & i = 1, 2, \ldots, 7 \\ WM^{i+1}(t) - I^i_{TB1}(t) & i = 8, 2, \ldots, 16 \end{cases} \tag{9}$$

Similarly, once the working memory is set to be the coding of the goal-vector, the PI mechanism will control the agent to go to that goal. For example, if the agent has stored two vector memories of two locations denoted as VM_A and VM_B, after setting the working memory as $VM_B - VM_A$, a short cut path from A to B could be spontaneously generated via the steering circuit (Eq. (9)).

3.2. Locust collision detection model

The looming sensitive neurons named lobula giant motion detector (LGMD) has been intensively modeled (for review see Fu et al. (2019)), this study adapted the model proposed in Yue and Rind (2006) as the collision avoidance functionality of our autonomous navigation (see left-bottom part of Fig. 4).

3.2.1. LGMD model with feature enhancement

The LGMD-based neural network for collision perception consists of four populations of neurons-the photoreceptor (P), excitatory (E), inhibitory (I) and summing (S) and two signal cells (LGMD and the feed-forward inhibition (FFI) as shown in the left-bottom of Fig. 4). Specifically, the photoreceptor (P layer of LGMD model) defined as:

$$P^{i,j}(t) = L^{i,j}(t) - L^{i,j}(t-1) + \sum_{f}^{n_f} p_f P^{i,j}(t-f) \tag{10}$$

where $L^{i,j}(t)$ is the brightness of the pixel located in the ith row and jth column of the input image at time t. n_f defines the visual persistence's maximum number of time steps and $p_f = (1+e^{uf})^{-1} \in (0,1)$ is the persistence coefficient. Therefore, the P neurons encode the temporal luminance change of the views perceived by the agent.

In the next layer, excitatory neurons E directly receives P neurons' activation, i.e., $E^{i,j}(t) = P^{i,j}(t)$ while inhibitory neurons' activation is determined by the one-frame delayed P as:

$$I^{i,j}(t) = \sum_r \sum_c P^{i+r,j+c}(t-1) w_I^{r,c} \quad (if \quad r = c, \quad then \quad c \neq 0) \tag{11}$$

where w_I is the local inhibition weight. The condition *if* $r = c, c \neq 0$ constrains the spread out of the inhibition to the neighboring neurons in the next layer.

Then the summing neurons S summed the E and I neurons:

$$S^{i,j}(t) = E^{i,j}(t) - W_I I^{i,j}(t) \tag{12}$$

where W_I is the global inhibition weight. To enhance the excitation passed from S layer to the LGMD cell, G layer is added as:

$$G^{i,j}(t) = S^{i,j}(t) Ce(t)^{i,j} \omega(t)^{-1} \tag{13}$$

where $\omega(t)$ is the scale factor and calculated by:

$$\omega(t) = \Delta c + max(abs[Ce(t)]) C_{\omega}^{-1} \tag{14}$$

and $Ce(t)$ is the passing coefficient and can be computed by convolution:

$$Ce_t = S(t) * w_e, \quad w_e = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{15}$$

The activation of G layer neurons $G(t)$ in Eq. (13) is gated by the threshold T_g to get $\tilde{G}(t)$. And then finally the LGMD neurons summing up all the G layer neurons and then be activated by the sigmoid function:

$$LGMD(t) = (1 + e^{-\sum_i \sum_j abs(\tilde{G}^{i,j}(t)) N_{cell}^{-1}})^{-1} \tag{16}$$

where N_{cell} is the total number of the G neurons, i.e., the size of the input image. The feed-forward inhibition FFI is calculated by:

$$FFI(t) = \sum_{i=1}^{n_r} \sum_{j=1}^{n_c} abs(P^{i,j}(t-1) n_{cell}^{-1}) \tag{17}$$

where n_r and n_c is the number of rows and columns of the input image.

3.2.2. LGMD-based collision perception

As described in the previous section, once we computed $LGMD(t)$ and $FFI(t)$, we could use this two signal to detect if there is a collision at time t by applying the following spiking mechanism:

$$C^{LGMD}(t) = \begin{cases} 1 & \text{if } N_s \geq n_{sp} \wedge FFI(t) < T_{FFI} \\ 0 & otherwise \end{cases} \tag{18}$$

where N_s is the number of spikes of LGMD which is computed by $\sum_{t-t_{sp}}^{t} LGMD^{spike}(t)$; T_{FFI} is the threshold of FFI, and if the LGMD spikes more than n_{sp} times during t_{sp} time-steps then the final output from LGMD neurons is TRUE spiking (note that $n_{sp} \leq t_{sp}$). Specifically, $LGMD^{spike}(t)$ is the computed by:

$$LGMD^{spike}(t) = \begin{cases} 1 & \text{if } LGMD(t) \geq T_{LGMD} \\ 0 & otherwise \end{cases} \tag{19}$$

where T_{LGMD} is the spiking threshold of LGMD neuron. Note that here collision means LGMD detects the potential risk of collision rather than the actual collision. The critical parameters of the LGMD used in this study are listed in Table 1.

Table 1
The parameters of the LGMD model.

Parameters.	Static simulated world	Dynamic simulated world
$n_{cell} = n_r \times n_c$	$40\,000 = 200 \times 200$	$40\,000 = 200 \times 200$
w_I (11)	$\begin{bmatrix} 0.125 & 0.25 & 0.125 \\ 0.25 & 0 & 0.25 \\ 0.125 & 0.25 & 0.125 \end{bmatrix}$	$\begin{bmatrix} 0.125 & 0.25 & 0.125 \\ 0.25 & 0 & 0.25 \\ 0.125 & 0.25 & 0.125 \end{bmatrix}$
W_I (12)	0.35	0.35
C_{ω} (14)	0.5	0.5
T_g	15	15
T_{FFI}	0.35	0.35
T_{LGMD}	0.98	0.96
n_{sp}	7	8
t_{sp}	7	8

3.3. Navigation algorithm

The state of the navigating agent at time t is denoted as $S(t) = [P_{agent}(t), V_{agent}(t), \theta(t)]$ and $P(t) = [x(t), y(t), z(t)]$, $V(t) = [v_x(t), v_y(t), v_z(t)]$. In this study, the agent is constrained to move in the xy-plane, so let $z = 0.01$ and $v_z = 0$. Therefore, the navigation problem becomes determining the $V_{agent}(t)$, and then the position of the agent could be updated by:

$$P_{agent}(t+1) = P_{agent}(t) + V_{agent}(t) \tag{20}$$

and velocity of the moving agent is formulated as:

$$\begin{aligned} V_{agent}(t+1) = \; & V_{agent}(t) \\ & + acc(t)[\sin\theta(t), \cos\theta(t), 0] \\ & - drag(t) V_{agent}(t) \end{aligned} \tag{21}$$

where $acc(t)$ is the acceleration and $drag(t)$ defines the drag factor. $\theta(t)$ is heading of the agent and updated by:

$$\theta(t+1) = \theta(t) + M(t) \tag{22}$$

where $M(t)$ is the turning angle (motion command). Thus the agent's state could updated continuously if $acc(t)$, $drag(t)$ and $M(t)$ is determined at every time t.

The flowchart of the proposed navigation algorithm is shown in Fig. 5, where how the state of the agent (F_{state}) is switched is illustrated as the finite state machine. Note that for F_{state} in Phase2, different vector working memories are set given the current navigating goals in the PI-homing state. Specifically:

1. Start from the start point, go to the nearest food site: set the working memory in Eq. (8) be the vector memory (inversed as that in Eq. (8)) having the lowest summed-up activation, i.e.:

$$\begin{aligned} & WM(t) = 1.0 - VM_k \\ & \arg\min_k \sum_{i=0}^{15} VM_k^i \end{aligned} \tag{23}$$

2. Got the food site, go to the next nearest food site: set the working memory be the subtraction of the vector memory of current food site and the rest food site that have the lowest summed-up activation, i.e., :

$$\begin{aligned} & WM(t) = 1.0 - VM_k \\ & \arg\min_k \sum_{i=0}^{15} VM_k^i - VM_c^i \end{aligned} \tag{24}$$

   where VM_c is the vector memory of the current food site.
3. Got the last food site, go back to the start point: set the working memory be the activation of CPU4 neurons, i.e., the normal PI driven homing as that in Eq. (8).

This again reflects the flexibility of vector-based navigation and how could this be encoded as the unified population coding of the neurons in the insect brain, which could be adapted and applied in efficient autonomous navigation of robots.

Note that the motion control strategy for collision avoidance varies depending on the type of environment. In a static simulated environment, when the LGMD network detects a collision, the agent executes a sharp turn in a randomly chosen direction to avoid the obstacle. However, in a dynamic environment, the agent stops for a brief period of time (i.e., two time-steps) before resuming navigation. We also experimented with using a sharp

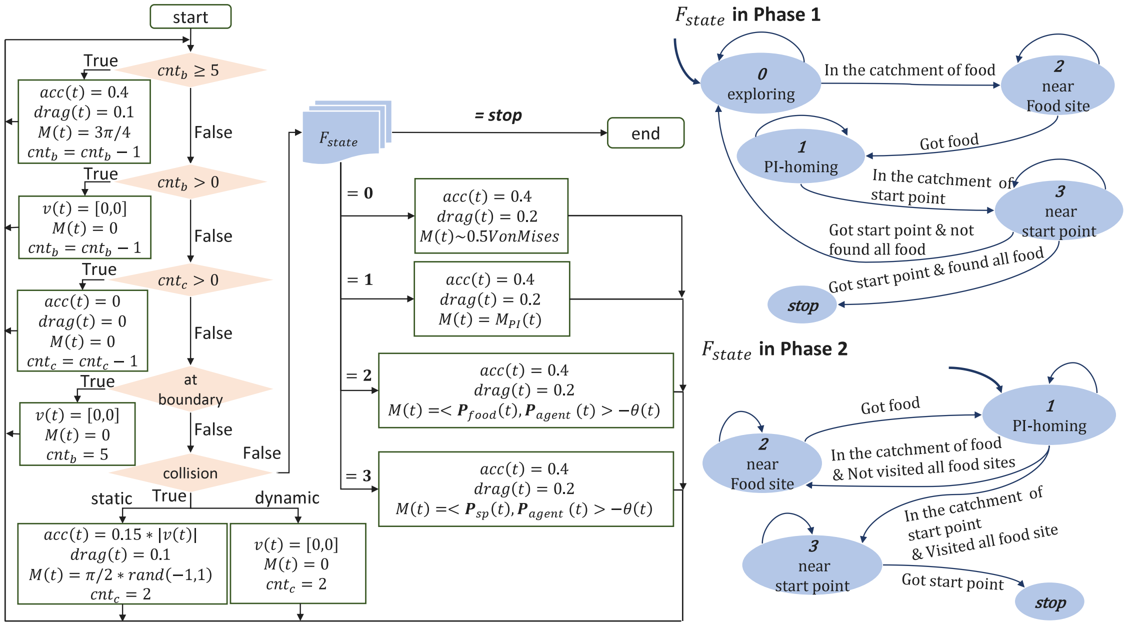


Fig. 5. The control strategy for autonomous based on the proposed PI for goal approaching and LGMD for collision avoidance. Left part is the flow chart and the right part depicts the state machine of the F_{state} in the flow chart for Phase1 and Phase2 of the defined navigation task (for the details of navigation task definition see Fig. 3 and Navigation task definition). cnt_c and cnt_b are the counter variable of the collision and boundary respectively and their initial values are 0. They are used to control the motion of the agent when at the boundary of the arena and when LGMD-based collision is positive. Note that motion variables (i.e., $acc(t)$, $drag(t)$, $M(t)$, $v(t)$) will keep unchanged when there is no specific assignment command. $\langle P_{food}, P_{agent}(t)\rangle$ denotes the angle between the vector of food sites and agent's position and $\langle P_{sp}, P_{agent}(t)\rangle$ is the angle between the start point and the agent's position.

turn strategy in a dynamic environment, but its performance was suboptimal. Therefore, to maintain the focus on the main objective of this study, we selected the control strategy with the best performance under the current simulation conditions.

It is important to note that LGMD detects potential risk of collision which is prior to the actual collision, namely, LGMD predicts collision when its response peaks, this is before the time when real collisions occur. This prediction will cause a large $\theta(t)$ in (22) when in static simulated world or $v_{agent}(t+1) = 0$ in (21) when in dynamic simulated world.

4. Simulation results

To verify the proposed autonomous navigation algorithm, we undertake simulations wherein the agent aims to complete the navigation task defined in Navigation task definition.

In our simulations, we do not impose any physical constraints. Instead, we determine collision numerically by comparing the distance between the agent and the center of obstacles with the size of the obstacle in the corresponding dimension. Although collision with obstacles has no effect on the agent's motion, the obstacles are visible to the agent. As a result, the vision-based LGMD algorithm can detect impending collisions and detour the agent's motion route via the control rule.

4.1. Metrics

Several metrics are defined to evaluate the navigation performance in static and dynamic simulated environments within a certain period of time T (T could be the value of time-out T_{out} which is set to be 20 000 steps or the time when navigation task is completed). First, the length of the route traveled (L_R) during navigation which is formulated as:

$$L_R = \sum_{t=0}^{T-1} |P_{agent}(t+1) - P_{agent}(t)| \tag{25}$$

Second, the time T (in unit of time-steps) indicates how long the agent takes to complete the task. Thus, we have two metrics to assess the performance of completing the navigation task, the length of the trajectory (L_R) in static simulated world wherein the agent makes sharp-turns when collision is perceived which makes the path more tortuous, and the completing time (T) in dynamic simulated world wherein the agent will stop when collision is detected which takes longer time to finish the task. Therefore, shorter path and less time costed indicates better performance and higher efficiency.

Third, to evaluate the performance of the LGMD-based collision detection, we use the number of collision happened ($C_{collision}$) during navigation which is determined by:

$$C_{collision} = \sum_{t=0}^{T} F_{collision}(t) \tag{26}$$

where $F_{collision}(t)$ indicates that if the agent collided with the obstacles at time t, as formulated:

$$F_{collision}(t) = \begin{cases} 1 & if\ P_{agent}(t) \in \bigcup_{i=0}^{N_{obs}} \Omega_i \\ 0 & otherwise \end{cases} \tag{27}$$

where Ω_i is the ith of N_{obs} obstacle's occupation of the space. Therefore, one may notice that here collision means the real collision happens in our simulated world instead of the potential risk of collision predicted by the LGMD model.

Note that to calculate $C_{collision}$, the agent in all simulations is set to be virtually collided with the obstacles not physically, which means that the agent has not successfully avoid the collision but the simulation will continue without physical damages. This may not be true in the real world, but is convenient to quantitatively evaluate the effectiveness of the obstacle-avoid algorithm in a complete navigation trial.

We then apply these metrics to test the performance of the proposed autonomous navigation algorithm for driving the agent

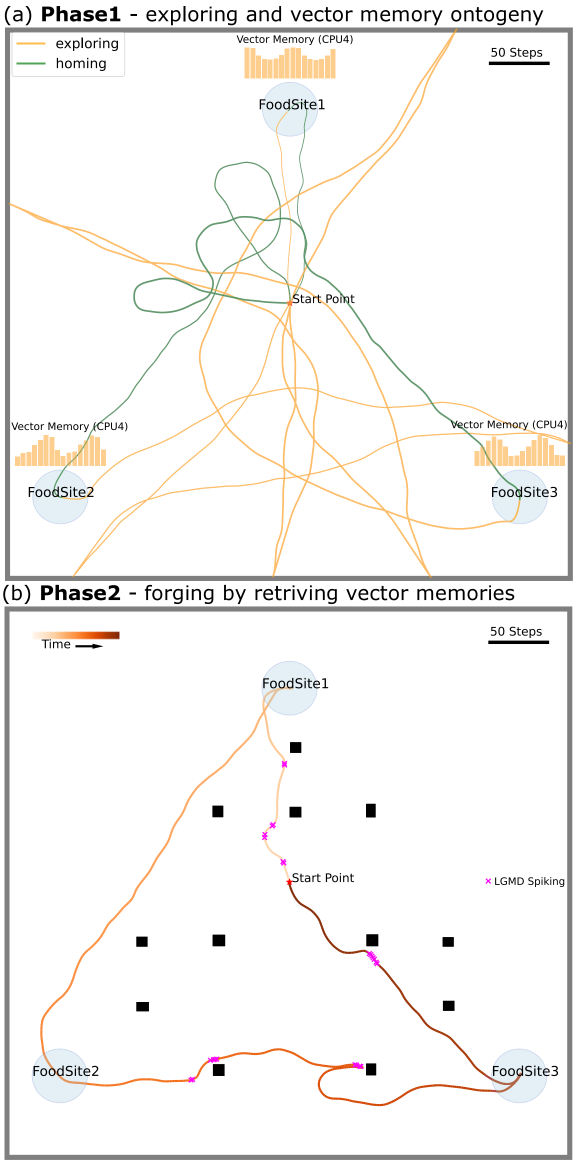


Fig. 6. An example performance of the agent's completing navigation task in a static simulated environment with $N_{obs} = 12$ obstacles. (a) The trajectories of the agent in phase1 and the stored vector memories at each food site is depicted by the inserted bar plot. (b) The trajectories of the agent in phase2 and the positions at which the LGMD-based collision perception are positive (Eq. (18)) are indicated by magenta cross-markers.

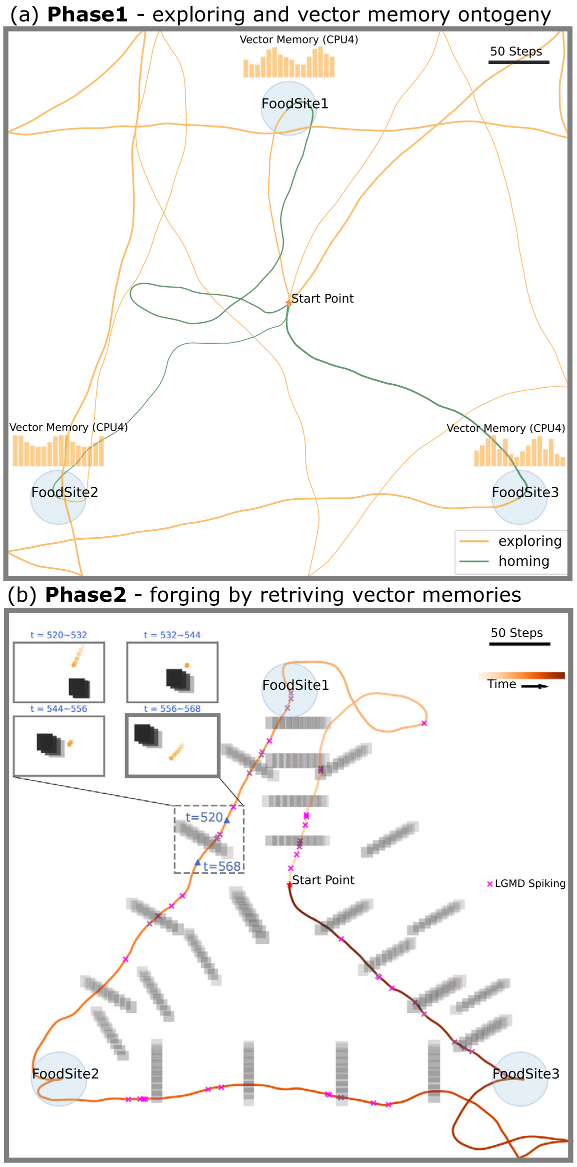


Fig. 7. An example performance of the agent's completing navigation task in a dynamic environment with $N_{obs} = 24$ obstacles. (a) The trajectories of the agent in phase1 and the stored vector memories at each food site is depicted by the inserted bar plot. (b) The trajectories of the agent in phase2 and the positions at which the LGMD-based collision perception are positive (Eq. (18)) are indicated by magenta cross-markers. The inserted plot draws the obstacle and agent's moving route from $t = 520$ to $t = 268$ (orange dots marks the agent and black rectangles indicates the obstacles. Opacity indicates time-step where darker objects means nearer current time-step) which demonstrates that the agent is capable of applying LGMD-triggered stop to avoid colliding with the moving obstacle.

to complete the navigation task in Phase2 with main aspects of goal-approaching and collision-avoidance.

4.2. Performance of completing the defined navigation task

Firstly, we show that the proposed model can drive the agent to complete the navigation task (see Navigation task definition) free of collision in both static and dynamic simulated environments. Figs. 6 and 7 (also Video S1 and Video S2 in the supplementary material) demonstrates the satisfactory performance of the agent's completing the defined navigation tasks in static and dynamic simulated environments, respectively.

In Fig. 6, the simulation was undertaken in a world with 12 obstacles. After exploring the environment and got the vector memories of the food sites, the agent could successfully complete the task of Phase 2 with a relatively straight route. Note that LGMD-based collision detection works well as it detected all the potential collisions based on which the motion control driven the agent to timely avoid the obstacles. This demonstrates that in the static simulated environment, PI-based *goal approaching* and LGMD-based *collision avoidance* could cooperate to facilitate the autonomous navigation.

Similarly, in Fig. 7, the simulation was undertaken in a world with 6 moving obstacles. After Phase1's exploring and vector-memory developing, the defined navigation task has been completed by the agent without colliding with the moving obstacles. As shown in Fig. 5, the agent will stop for 2 time-steps to let the obstacle passed by and then continue its journey. The inserted

Table 2
The overall performance of navigation.

Type	N_{obs}	$avg(L_R)$/step	$avg(T)$/time-step	$med(C_{collision})$
Static	6	1755.1	1164.0	0
	12	1877.5	1254.2	0
	18	2077.0	1412.6	2
	24	2039.4	1402.8	2
	30	2492.1	1746.4	3
	36	2700.3	1906.0	3
	42	2673.9	1902.5	6
Dynamic	6	1836.8	1264.7	0
	12	1791.8	1341.1	2
	18	1758.8	1420.3	2
	24	1762.7	1516.3	2
	30	2059.6	1890.7	5
	36	1684.2	1694.5	5
	42	1657.4	1806.7	8

figure shows that the agent successfully avoid the collision based on the output from the LGMD model. This demonstrates that in the dynamic environment, PI-based *goal approaching* and LGMD-based *collision avoidance* could also cooperate to facilitate the autonomous navigation.

The formation of the navigational routes in our simulation resembles that of Bertrand, Lindemann, and Egelhaaf (2015), which deduces that combining of collision avoidance and goal approaching may form the common routes in real insects' navigation. Despite that this is still an open question, both Bertrand et al. (2015) and this study demonstrate the validity of integrating insect-like goal approaching and collision avoidance and its potential in developing efficient autonomous navigation system. Moreover, in spite of the analogous idea of integrating goal approaching and collision avoidance, we present a complete autonomous navigation system involving insect-like goal approaching and collision avoidance while Bertrand et al. (2015) artificially provides an abstracted goal-direction. Therefore, the proposed model makes the first attempt to link the detailed neural circuity with the navigation behaviors, leading to a parsimonious navigation algorithm that could benefit solving mobile robots' navigation tasks.

To systematically evaluate the performance of the proposed navigation algorithm, we run 15 trials for different environments (static or dynamic with different number of obstacles). Table 2 shows the four metrics defined in the previous subsection for all the simulations, where $avg(X)$ indicates the averaged value of X and $med(X)$ is the median of X which represents the results of $N_{trial} = 15$ simulations. These results demonstrate that the proposed model is sufficient to generate autonomous navigation maneuver in a sensory–motor closed-loop manner. Given the data in Table 2, agents could complete the Phase2 navigation by following a rather straight trajectory within a limited time period (even with 42 obstacles in the environment, the agent could complete the task within 2000 time-steps). Completing the defined navigation tasks becomes more and more changeling as the number of the obstacles increase, resulting in more number of collision happens and longer time-steps taken to arrive the destinations.

In addition, as the visual contrast in our simulation is 100%, we further test the robustness of the proposed navigation model in the dynamic simulated world with 30 obstacles at different visual contrasts (i.e., the varied brightness differences between the background and the obstacles whereby the contrast is calculated by $(L_{max} - L_{min})/255$). The results of number of collision ($C_{collision}$) are illustrated in supplementary Figure S1, which show insignificant differences among the performance in dynamic simulated world with varied visual contrast. However, in the world with relatively low contrast, LGMD becomes less sensitive. Making the firing threshold of LGMD cell (T_{LGMD} in (19)) to be adaptive can improve the performance (Yue & Rind, 2006).

4.3. Significant decrements of the number of collision using LGMD-based collision avoidance

Whether the agent driven by the proposed model could benefit from the LGMD-based collision avoidance determines the validation of integrating local fast interactive cues with global working memories. To answer this question, here we compare the number of collision ($C_{collision}$, Eq. (27)) of the agents with/without LGMD-based collision avoidance mechanism when completing the Phase2 navigation task in static and dynamic simulated environments with varying density of obstacles. The violin plots Figs. 8 and 9 display the results of $N_{trial} = 15$ simulations for each static and dynamic simulated world with different number of obstacles respectively.

In Fig. 8, it is apparent that agents with LGMD-based collision avoidance mechanism (light-red violins) outperforms that with no LGMD embedded (light-blue violins), especially in the environment with relatively lower-density obstacles (decrease the medians of $C_{collision}$ by 100% in the world with 6 and 12 obstacles). The LGMD-based collision avoidance not only greatly reduces the median number of collision happens when guided by PI-driven goal approaching but also makes the navigation more robust which is implied by the decline of the variance (σ) of the data labeled alongside the violins of each simulation group in Fig. 8.

Similarly, in Fig. 9, significant decrease (with maximum 100% and minimum 16.67%) of the number of collision and improvement of overall robustness of navigation performance during navigation in dynamic environment can be seen though not as much as that in static simulated environment. Given that moving obstacles are more unpredictable and the agent just stop to wait which may not be the perfect control strategy. However, for the seek of proof-of-concept validation, these results support idea that LGMD-based collision avoidance could be applied in developing an autonomous navigation system though with imperfect control algorithm.

We also conduct the same experiments in the dynamic simulated world with randomly moving obstacles which demonstrate similar results (see supplementary Figure S2). Taken together, simulation results suggested that the LGMD-based collision avoidance could be the cornerstone of closed-loop autonomous navigation in the aspect of stopping the agent from colliding with stationary or moving obstacles.

4.4. Relatively slight impact of collision avoidance on goal approaching

On the other hand, not until we have assessed the impact of the LGMD-based *collision avoidance* on PI-based *goal approaching* behavior, could the conclusion that an insect-inspired model is capable of facilitating the closed-loop autonomous navigation be drawn. To this end, firstly as shown in Fig. 10, the navigating trajectories of the LGMD agent is relatively more torturous and dispersed from the straight pathways but the optimal route is also maintained (i.e., *start point* → *food site 1* → *food site 2* → *food site3* → *start point*). That is to say, incorporating the LGMD-based collision avoidance will not impact the overall performance of the goal approaching guided by the working memories from the PI mechanism though the trajectory straightness is slightly declined. This result is in line with the statement in Stone et al. (2017) that PI is capable of accounting for deviations around objects (obstacles) thanks to PI's dynamically updating the home vector. Here we give a concrete validation by building a more complete navigation mechanism with a bio-inspired collision avoidance model integrated.

To further assess the collision avoidance's impact on the performance of goal approaching, we compare the route length (R_L

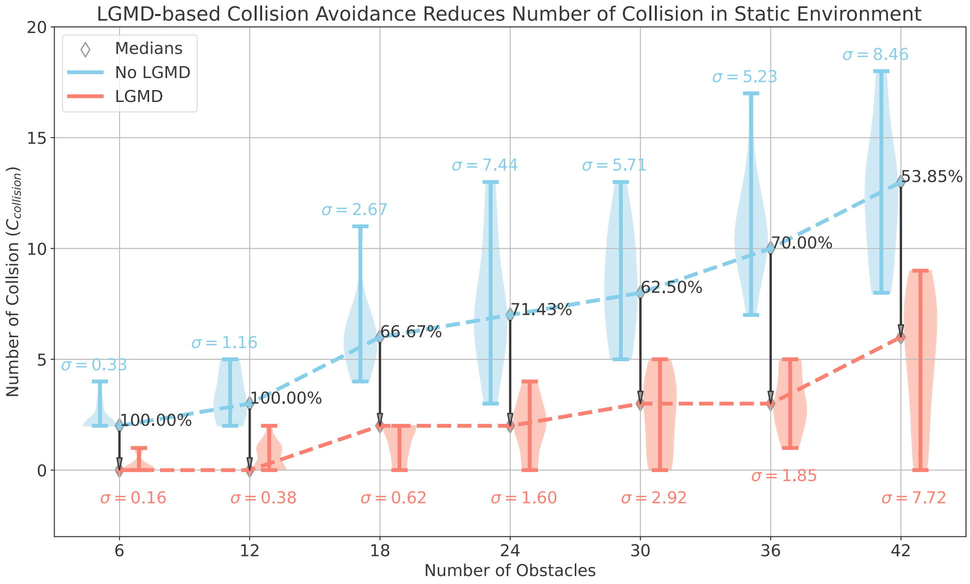


Fig. 8. The violin plot showing the comparison between the number of collision ($C_{collision}$, see Eq. (27)) happened when agent is navigating in static simulated world with different number of obstacles. Downward arrows indicate how much the medians of number of collision has been reduced by the LGMD-based collision avoidance.

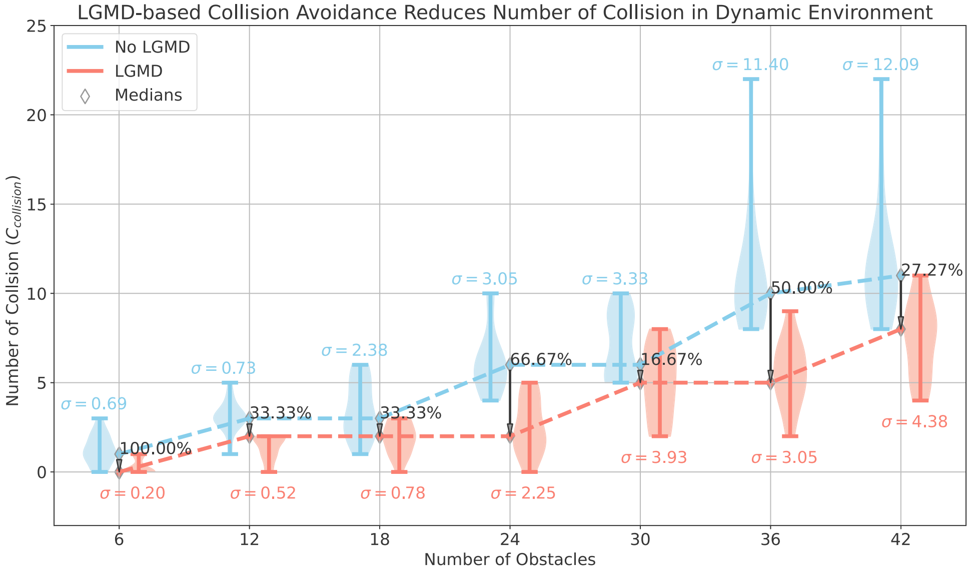


Fig. 9. The violin plot showing the comparison between the number of collision ($C_{collision}$, see Eq. (27)) happened when agent is navigating in dynamic simulated world with different number of obstacles. Downward arrows indicate how much the medians of number of collision has been reduced by the LGMD-based collision avoidance.

in Eq. (25)) and finish time (T) of completing the phase2 task in both static and dynamic environment as shown in Fig. 11. Given the different control strategies when collision is detected by the LGMD network in static and dynamic environment, the route length in static simulated environment and finish time in dynamic environment of the LGMD agents is relatively longer than that of agents without collision avoidance capability. Note that when navigating in the world with high density of obstacles, the performance of goal approaching becomes unstable because of the high frequency of collision. But the overall performance is not significantly affected by the collision avoidance, which suggests that the PI-based goal approaching could cooperate well with the LGMD-based collision avoidance, leading to a minimal, effective model for autonomous navigation.

4.5. Summarize- a robust and efficient navigation algorithm

By summarizing the results from previous subsections: 4.2 shows the effectiveness of the proposed autonomous navigation model; 4.3 discusses how the insect-inspired collision avoidance model dramatically reduces the times of collisions by 17%–100%; while 4.4 emphasizes that though with local LGMD-based collision avoidance incorporated, the efficiency of global goal approaching can still be maintained (decreased only by about 0%–14%). The results demonstrate well the proposed insect-inspired algorithm could facilitate robust and efficient autonomous navigation in static and dynamic simulated environments.

The proposed navigation model in this study exemplified the *goal approaching* and *collision avoidance* by path integration and

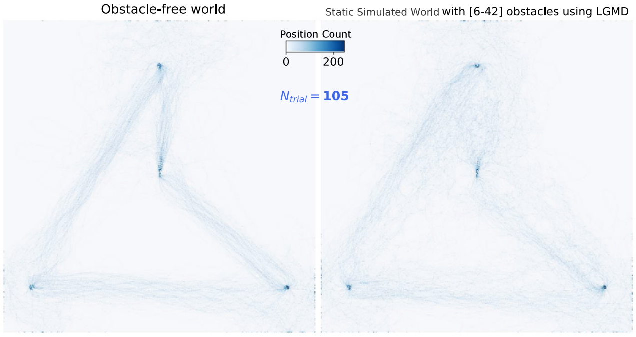


Fig. 10. The heatmap plot that depicts the accumulated times of spatial locations' (meshed into grids with accuracy of 1 step) being occupied by the navigating agents. The left is the data of obstacle-free world while the right part is that of static simulated world with varying number of obstacles (run 15 trials for each world with 6–42 obstacles).

LGMD respectively. Nevertheless, other bio-inspired models could also be applied given the specific navigational tasks, such as the optic-flow based collision avoidance method in Bertrand et al. (2015). We also tested the performance of another version of LGMD (Yue & Rind, 2005) and compare it with that of the applied LGMD (Yue & Rind, 2006) in the proposed navigation model (see details in supplementary Figure S3). These results show similar performance of *collision avoidance* and its impact on *goal approaching*, which further supports the proposed proof-of-concept that the insect-inspired model combined with even simplest control mechanism could facilitate autonomous navigation.

5. Discussion

From the simulation results, one can see that when there are few obstacles, the agent could successfully avoid all the obstacles ($C_{collision} = 0$), but when the density of the obstacles is higher, some collisions still happened. This leads us to the future improvement of the motion control algorithm when a collision is perceived, but is beyond the scope of this study.

To further test the performance of the proposed model, experiments involving real robots in the real physical world is needed. One should notice that in this study, the agents are modeled as an orientated-point with a visual system which is not realistic for the real robot application. Therefore, implementing the proposed navigation model into some real robots with specific physical attributes and constrains is necessary prior for real-world application. For instance, the wheeled micro-robot embedded with LGMD-based collision detection named *Colias* (Hu, Arvin, Xiong, & Yue, 2016) and the legged robot inspired by ants called Antbot (Dupeyroux et al., 2019), etc. In addition, verifying the navigational performance of the robots in an ad-hoc physical platform (Liu, Sun, Hu, Fu, & Yue, 2021) with flexible modulation of sensory cues or in the field (Dupeyroux et al., 2019) could be a potential researching topic (also see Fig. 12). Once the specific real robot is determined, the temporal and spatial units which are abstracted as 'steps' in our simulation could also be defined accordingly. Real robotics verification with realistic physical constraints may further involve embodiment intelligence (Bermudez i Badia, Pyk, & Verschure, 2007; Franceschini, Pichon, & Blanes, 1992; Webb, 2020), which could advance the bio-inspired approaches in the context of autonomous navigation.

Another interesting extension of this work is to add visual memories into the working memory in light of the fact that there are massive kinds of vision-based *goal approaching* behaviors in insects (Collett, Chittka, & Collett, 2013; Ofstad, Zuker, & Reiser, 2011; Wystrach, Beugnon, & Cheng, 2012). Some of them have already been modeled (Ardin, Peng, Mangan, Lagogiannis, & Webb, 2016; Goulard, Buehlmann, Niven, Graham, & Webb, 2021; Sun et al., 2020; Zeil, 2012). Note that when vision is involved as one of the sources of working memory, obstacles per se becomes part of the visual scene, making the visual cue based *collision avoidance* not only interacts with the global working memory at the motion-level but also the sensory-level. And the collision avoidance caused additional motion patterns (like the abrupt turn

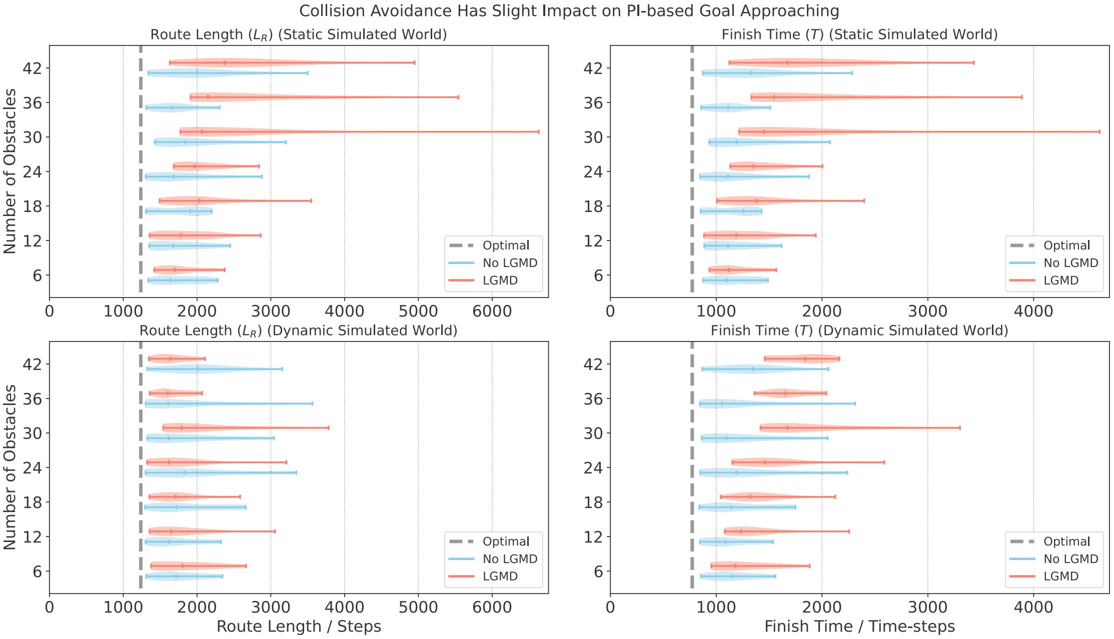


Fig. 11. The violin plots of the route length R_L (left column) and finish time T (right column) in static (top row) and dynamic environment (bottom row) of agent's with (light-red violins)/without (light-blue violins) LGMD-based collision avoidance completing the phase2 navigation task. The *optimal* lines (vertical dashed gray line) shows the shortest length of the route linking the three food sites in the left panel and the shortest time to travel across these food sites given the constant velocity.

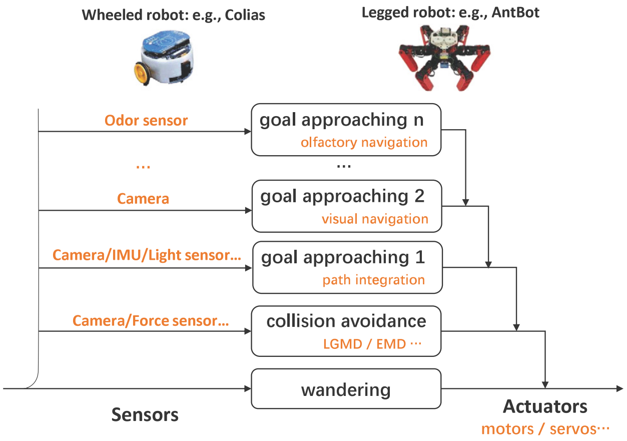


Fig. 12. The schematic diagram of integrating different goal-approaching mechanisms via subsumption architecture (Brooks, 1986). The example robots Colias (Hu et al., 2016) and Antbot (Dupeyroux et al., 2019) are both bio-inspired robots.

in the proposed algorithm within static simulated environment) will also impact the visual inputs of navigational learning in light of the sensory–motor closed loop.

From the biological perspective, the way how these two subsystems, i.e., the goal approaching that acts as the global memory and the collision avoidance that serves as the local immediate cue interacts with each other is still an open question which is worth investigating in the feature. Could they assess each other's information? If yes, how this will affect their own computation? In this study, we combine these two components through a simple control rule but we do not imply that this is the actual way real insect apply to navigate. We intended to keep everything simple to investigate the least effort to use insect-inspired models for robot's autonomous navigation. That is why we can call our model minimalist. Nevertheless, exploring different interaction ways between subsystems and then verifying the performance using the same paradigm in this study is also appealing and reflects the concept of bio-robotics (Webb, 2001): robotics study may provide insights to biology.

6. Conclusion

An effective insect-inspired navigation algorithm has been proposed by incorporating models of *global* goal approaching and *local* collision avoidance. The performance of this integrated model is verified by letting the simulated agent complete navigational tasks in static and dynamic simulated environments respectively. The proved fact that these two subsystems (i.e., the insect-inspired goal approaching and collision avoidance models) cooperate well under a unified control strategy makes the proposed model a building block for more complicated and robust navigation systems by applying the subsumption architecture (Brooks, 1986) to real robots (also see Fig. 12). This architecture per se is also biologically inspired that reflects one of the design principle of biological and technological system: the complexity is driven by robustness rather than optimum (Doyle & Csete, 2011). Drawing all these natural principles leads to a insect bio-inspired system that embedded with more biological understandings (Del Ser et al., 2019; Ng, Elgar, & Stuart-Fox, 2021; Ni, Wu, Fan, & Yang, 2016), thus could help us better reap the benefits of natural solutions featuring high adaptability and efficiency, at least in the context of autonomous navigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This research has received funding from the National Natural Science Foundation of China under the Grant No 62206066 and No 12031003, the China Postdoctoral Science Foundation under the Grant No 2022M710864, the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Grant Agreement No. 778062 ULTRACEPT, and the Social Science Fund of the Ministry of Education of China under the Grant No 22YJCZH032.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.neunet.2023.05.033.

References

Algabri, M., Mathkour, H., Ramdane, H., & Alsulaiman, M. (2015). Comparative study of soft computing techniques for mobile robot navigation in an unknown environment. *Computers in Human Behavior*, *50*, 42–56.

Álvarez-Salvado, E., Licata, A. M., Connor, E. G., McHugh, M. K., King, B. M., Stavropoulos, N., et al. (2018). Elementary sensory-motor transformations underlying olfactory navigation in walking fruit-flies. *Elife*, *7*, Article e37815.

Ardin, P., Peng, F., Mangan, M., Lagogiannis, K., & Webb, B. (2016). Using an insect mushroom body circuit to encode route memory in complex natural environments. *PLoS Computational Biology*, *12*(2), Article e1004683.

Bermudez i Badia, S., Pyk, P., & Verschure, P. F. (2007). A fly-locust based neuronal control system applied to an unmanned aerial vehicle: the invertebrate neuronal principles for course stabilization, altitude control and collision avoidance. *International Journal of Robotics Research*, *26*(7), 759–772.

Bertrand, O. J., Lindemann, J. P., & Egelhaaf, M. (2015). A bio-inspired collision avoidance model based on spatial information derived from motion detectors leads to common routes. *PLoS Computational Biology*, *11*(11), Article e1004339.

Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, *2*(1), 14–23.

Buehlmann, C., Mangan, M., & Graham, P. (2020). Multimodal interactions in insect navigation. *Animal Cognition*, *23*(6), 1129–1141.

Cizek, P., & Faigl, J. (2019). Self-supervised learning of the biologically-inspired obstacle avoidance of hexapod walking robot. *Bioinspiration & Biomimetics*, *14*(4), Article 046002.

Collett, T. S. (2019). Path integration: how details of the honeybee waggle dance and the foraging strategies of desert ants might help in understanding its mechanisms. *Journal of Experimental Biology*, *222*(11), jeb205187.

Collett, M., Chittka, L., & Collett, T. S. (2013). Spatial memory in insect navigation. *Current Biology*, *23*(17), R789–R800.

Collett, T. S., & Collett, M. (2002). Memory use in insect visual navigation. *Nature Reviews Neuroscience*, *3*(7), 542–552.

Del Ser, J., Osaba, E., Molina, D., Yang, X.-S., Salcedo-Sanz, S., Camacho, D., et al. (2019). Bio-inspired computation: Where we stand and what's next. *Swarm and Evolutionary Computation*, *48*, 220–250.

Doyle, J. C., & Csete, M. (2011). Architecture, constraints, and behavior. *Proceedings of the National Academy of Sciences*, *108*(Supplement 3), 15624–15630.

Duchoň, F., Babinec, A., Kajan, M., Beňo, P., Florek, M., Fico, T., et al. (2014). Path planning with modified a star algorithm for a mobile robot. *Procedia Engineering*, *96*, 59–69.

Dupeyroux, J., Serres, J. R., & Viollet, S. (2019). AntBot: A six-legged walking robot able to home like desert ants in outdoor environments. *Science Robotics*, *4*(27), eaau0307.

Faisal, M., Hedjar, R., Al Sulaiman, M., & Al-Mutib, K. (2013). Fuzzy logic navigation and obstacle avoidance by a mobile robot in an unknown dynamic environment. *International Journal of Advanced Robotic Systems*, *10*(1), 37.

Franceschini, N. (2014). Small brains, smart machines: From fly vision to robot vision and back again. *Proceedings of the IEEE*, *102*, 751–781.

Franceschini, N., Pichon, J.-M., & Blanes, C. (1992). From insect vision to robot vision. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, *337*(1281), 283–294.

Fu, Q., Hu, C., Liu, P., & Yue, S. (2018). Towards computational models of insect motion detectors for robot vision. In *Towards autonomous robotic systems conference* (pp. 465–467).

Fu, Q., Hu, C., Peng, J., Rind, F. C., & Yue, S. (2020). A robust collision perception visual neural network with specific selectivity to darker objects. *IEEE Transactions on Cybernetics*, *5*(12), 5074–5088. http://dx.doi.org/10.1109/TCYB.2019.2946090.

Fu, Q., Hu, C., Peng, J., & Yue, S. (2018). Shaping the collision selectivity in a looming sensitive neuron model with parallel ON and OFF pathways and spike frequency adaptation. *Neural Networks*, *106*, 127–143. http://dx.doi.org/10.1016/j.neunet.2018.04.001.

Fu, Q., Sun, X., Liu, T., Hu, C., & Yue, S. (2021). Robustness of bio-inspired visual systems for collision prediction in critical robot traffic. *Frontiers in Robotics and AI*, *8*, Article 529872. http://dx.doi.org/10.3389/frobt.2021.529872.

Fu, Q., Wang, H., Hu, C., & Yue, S. (2019). Towards computational models and applications of insect visual systems for motion perception: A review. *Artificial Life*, *25*(3), 263–311.

Gabbiani, F., Krapp, H. G., Koch, C., & Laurent, G. (2002). Multiplicative computation in a visual neuron sensitive to looming. *Nature*, *420*(6913), 320–324.

Goldschmidt, D., Manoonpong, P., & Dasgupta, S. (2017). A neurocomputational model of goal-directed navigation in insect-inspired artificial agents. *Frontiers in Neurorobotics*, *11*, 20.

Goulard, R., Buehlmann, C., Niven, J. E., Graham, P., & Webb, B. (2021). A unified mechanism for innate and learned visual landmark guidance in the insect central complex. *PLoS Computational Biology*, *17*(9), Article e1009383.

Gul, F., Rahiman, W., & Nazli Alhady, S. S. (2019). A comprehensive study for robot navigation techniques. *Cogent Engineering*, *6*(1), Article 1632046.

Hart, P. E., Nilsson, N. J., & Raphael, B. (1972). Correction to" a formal basis for the heuristic determination of minimum cost paths". *ACM SIGART Bulletin*, (37), 28–29.

Heinze, S. (2017). Unraveling the neural basis of insect navigation. *Current Opinion in Insect Science*, *24*, 58–67.

Heinze, S., Narendra, A., & Cheung, A. (2018). Principles of insect path integration. *Current Biology*, *28*(17), R1043–R1058.

Honkanen, A., Adden, A., da Silva Freitas, J., & Heinze, S. (2019). The insect central complex and the neural basis of navigational strategies. *Journal of Experimental Biology*, *222*(Suppl_1), jeb188854.

Hu, C., Arvin, F., Xiong, C., & Yue, S. (2016). Bio-inspired embedded vision system for autonomous micro-robots: The LGMD case. *IEEE Transactions on Cognitive and Developmental Systems*, *9*(3), 241–254.

Hulse, B. K., Haberkern, H., Franconville, R., Turner-Evans, D. B., Takemura, S.-y., Wolff, T., et al. (2021). A connectome of the Drosophila central complex reveals network motifs suitable for flexible navigation and context-dependent action selection. *ELife*, *10*, Article e66039.

Jung, S.-H., Hueston, C., & Bhandawat, V. (2015). Odor-identity dependent motor programs underlie behavioral responses to odors. *Elife*, *4*, Article e11092.

Kennedy, J. S. (1951). The migration of the desert locust (Schistocerca gregaria Forsk.). I. The behaviour of swarms. II. A theory of long-range migrations. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, *235*, 163–290.

Kim, S. S., Rouault, H., Druckmann, S., & Jayaraman, V. (2017). Ring attractor dynamics in the Drosophila central brain. *Science*, *356*(6340), 849–853.

Lambrinos, D., Möller, R., Labhart, T., Pfeifer, R., & Wehner, R. (2000). A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, *30*(1–2), 39–64.

Le Moël, F., Stone, T., Lihoreau, M., Wystrach, A., & Webb, B. (2019). The central complex as a potential substrate for vector based navigation. *Frontiers in Psychology*, *10*, 690.

Liu, T., Sun, X., Hu, C., Fu, Q., & Yue, S. (2021). A versatile vision-pheromone-communication platform for swarm robotics. In *2021 IEEE international conference on robotics and automation* ICRA, (pp. 7261–7266). IEEE.

Lyu, C., Abbott, L., & Maimon, G. (2022). Building an allocentric travelling direction signal via vector computation. *Nature*, *601*(7891), 92–97.

Menzel, R., & Giurfa, M. (2001). Cognitive architecture of a mini-brain: the honeybee. *Trends in Cognitive Sciences*, *5*(2), 62–71.

Möller, R. (2000). Insect visual homing strategies in a robot with analog processing. *Biological Cybernetics*, *83*(3), 231–243.

Muijres, F. T., Elzinga, M. J., Melis, J. M., & Dickinson, M. H. (2014). Flies evade looming targets by executing rapid visually directed banked turns. *Science*, *344*(6180), 172–177.

Mukhtar, A., Xia, L., & Tang, T. B. (2015). Vehicle detection techniques for collision avoidance systems: A review. *IEEE Transactions on Intelligent Transportation Systems*, *16*(5), 2318–2338. http://dx.doi.org/10.1109/TITS.2015.2409109.

Ng, L., Elgar, M. A., & Stuart-Fox, D. (2021). From bioinspired to bioinformed: benefits of greater engagement from biologists. *Frontiers in Ecology and Evolution*, 903.

Ni, J., Wu, L., Fan, X., & Yang, S. X. (2016). Bioinspired intelligent algorithm and its applications for mobile robot control: a survey. *Computational Intelligence and Neuroscience*, *2016*.

Ofstad, T. A., Zuker, C. S., & Reiser, M. B. (2011). Visual place learning in Drosophila melanogaster. *Nature*, *474*(7350), 204–207.

Omrane, H., Masmoudi, M. S., & Masmoudi, M. (2016). Fuzzy logic based control for autonomous mobile robot navigation. *Computational Intelligence and Neuroscience*, *2016*.

Quinn, R. D., Nelson, G. M., Bachmann, R. J., Kingsley, D. A., Offi, J., & Ritzmann, R. E. (2001). Insect designs for improved robot mobility. In *Climbing and walking robots: From biology to industrial applications* (p. 59). Wiley.

Raja, P., & Pugazhenthi, S. (2012). Optimal path planning of mobile robots: A review. *International Journal of Physical Sciences*, *7*(9), 1314–1320.

Rind, F. C., Wernitznig, S., Polt, P., Zankel, A., Gutl, D., Sztarker, J., et al. (2016). Two identified looming detectors in the locust: Ubiquitous lateral connections among their inputs contribute to selective responses to looming objects. *Scientific Reports*, *6*, 35525.

Salt, L., Howard, D., Indiveri, G., & Sandamirskaya, Y. (2020). Parameter optimization and learning in a spiking neural network for UAV obstacle avoidance targeting neuromorphic processors. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(9), 3305–3318.

Sayre, M. E., Templin, R., Chavez, J., Kempenaers, J., & Heinze, S. (2021). A projectome of the bumblebee central complex. *Elife*, *10*, Article e68911.

Singh, N. H., & Thongam, K. (2019). Neural network-based approaches for mobile robot navigation in static and moving obstacles environments. *Intelligent Service Robotics*, *12*(1), 55–67.

Soltani, A. R., Tawfik, H., Goulermas, J. Y., & Fernando, T. (2002). Path planning in construction sites: performance evaluation of the Dijkstra, A*, and GA search algorithms. *Advanced Engineering Informatics*, *16*(4), 291–303.

Srinivasan, M. V. (2010). Honey bees as a model for vision, perception, and cognition. *Annual Review of Entomology*, *55*, 267–284.

Stone, T., Webb, B., Adden, A., Weddig, N. B., Honkanen, A., Templin, R., et al. (2017). An anatomically constrained model for path integration in the bee brain. *Current Biology*, *27*(20), 3069–3085.

Sun, X., Yue, S., & Mangan, M. (2020). A decentralised neural model explaining optimal integration of navigational strategies in insects. *Elife*, *9*, Article e54026.

Turner-Evans, D. B., Jensen, K. T., Ali, S., Paterson, T., Sheridan, A., Ray, R. P., et al. (2020). The neuroanatomical ultrastructure and function of a biological ring attractor. *Neuron*, *108*(1), 145–163.

Webb, B. (2001). Can robots make good models of biological behaviour? *Behavioral and Brain Sciences*, *24*(6), 1033–1050.

Webb, B. (2020). Robots with insect brains. *Science*, *368*(6488), 244–245.

Webb, B., & Wystrach, A. (2016). Neural mechanisms of insect navigation. *Current Opinion in Insect Science*, *15*, 27–39.

Wehner, R. (2003). Desert ant navigation: how miniature brains solve complex tasks. *Journal of Comparative Physiology A*, *189*(8), 579–588.

Wystrach, A., Beugnon, G., & Cheng, K. (2012). Ants might use different view-matching strategies on and off the route. *Journal of Experimental Biology*, *215*(1), 44–55.

Wystrach, A., & Graham, P. (2012). What can we learn from studies of insect navigation? *Animal Behaviour*, *84*(1), 13–20.

Yang, S. X., & Meng, M. (2000). An efficient neural network approach to dynamic robot motion planning. *Neural Networks*, *13*(2), 143–148.

Yue, S., & Rind, F. C. (2005). A collision detection system for a mobile robot inspired by the locust visual system. In *Proceedings of the 2005 IEEE international conference on robotics and automation* (pp. 3832–3837). IEEE.

Yue, S., & Rind, F. C. (2006). Collision detection in complex dynamic scenes using an LGMD-based visual neural network with feature enhancement. *IEEE Transactions on Neural Networks*, *17*(3), 705–716.

Zeil, J. (2012). Visual homing: an insect perspective. *Current Opinion in Neurobiology*, *22*(2), 285–293.