I2Bot: an open-source tool for multi-modal and embodied simulation of insect navigation

Xuelong Sun,[1,2] Michael Mangan,[3] Jigen Peng, [1,2]* Shigang Yue[1,4]*
[1]Machine Life and Intelligence Research Center, Guangzhou University
[2]School of Mathematics and Information Science, Guangzhou University, Guangzhou, China
[3]Sheffield Robotics, Department of Computer Science, University of Sheffield
[4]School of Computing and Mathematical Sciences, University of Leicester
*E-mail: jgpeng@gzhu.edu.cn; sy237@leicester.ac.uk

Achieving a comprehensive understanding of animal intelligence demands an integrative approach that acknowledges the interplay between an organism's brain, body, and environment. Insects like ants, despite their limited computational resources, demonstrate remarkable abilities in navigation. Existing computational models often fall short in faithfully replicating the morphology of real insects and their interactions with the environment, hindering validation and practical application in robotics. To address these gaps, we present I2Bot, a novel simulation tool based on the morphological characteristics of desert ants. This tool empowers robotic models with dynamic sensory capabilities, realistic modeling of insect morphology, physical dynamics, and sensory capacity. By integrating gait controllers and sensory-motor loops, our approach elucidates fundamental navigation principles. By open-sourcing I2Bot, we aim to accelerate the understanding of insect intelligence and foster advances in the development of autonomous robotic systems.

Introduction

Unraveling the underlying mechanisms that give rise to animal's intelligent behaviors represents a formidable challenge, one that demands an integrative approach that transcends conventional boundaries. Central to this endeavor is the recognition of the dynamic interplay between an organism's brain, body, and environment (*1, 2*) – a concept central to the biorobotics approach (*3–5*). By bridging the gap between biology and robotics, biorobotics offers a unique opportunity to gain insights into the mechanisms underlying intelligent behavior (*3, 6*) while simultaneously inspiring the development of more advanced robotic systems (*7–9*).

Insect navigation stands as a remarkable example of intelligent behavior, offering a fertile ground for exploration (*10*). Despite possessing compact neural architectures with limited computational resources, insects demonstrate an astonishing capacity to navigate complex environments (*11, 12*), integrating multisensory inputs (*13, 14*), learning from experience (*15*), and executing adaptive decision-making processes (*16–19*). This ability to solve intricate problems with constrained computational power has sparked profound curiosity and a desire to understand the principles that govern these behaviors.

Computational models played important roles in revealing the neural circuits underlying insect navigation (*20–22*), and some of these models have been verified using mobile robots (*21, 23, 24*). However, these robots could not fully recapitulate the real insects' body due to differences in morphology. Simulations offer significant advantages over real-world robots when studying complex systems, such as the ability to conduct controlled experiments flexibly, modify parameters easily, and rapidly iterate through multiple scenarios (*25*). Although neuromechanical simulation tools for *Drosophila* have recently been developed (*26–28*), there remains a lack of tailored, realistic simulation tools for the systematic and comprehensive validation of insect navigation models, particularly for desert ants, that account for physical constraints.

To address these limitations, by leveraging the user-friendly and open-sourced software - ***Webots*** (*29*), a widely used robot simulation tools in academia (*30–33*), we developed a simulation tool based on the morphological characteristics of desert ants, endowing our navigating agent with dynamic vision, olfactory, tactile, and mechanosensory capabilities (see Figure 1). Leveraging this framework, we have implemented simple Forward Kinematics (FK) and Inverse Kinematics (IK) gait controller, enabling the realization of anatomically constrained computation model for path integration (PI). Additionally, our simulation tool facilitates the integration of vision and olfactory senses, allowing us to explore the sensory-motor closed loop inherent in insect navigation. Through systematic testing, we have uncovered evidence of how body-environment interactions can simplify the design of control models (see Table 1). Compared with other related simulation tools, the proposed open-source tool offers advantages in multi-modal sensory capacity, flexibility in constructing 3D world, support for multiple programming languages, lower learning costs, and a user-friendly interface (see Table 2 and Discussion for more detailed discussion).

By open-sourcing our simulation tool, we aim to provide a valuable resource for the research community, accelerating our collective understanding of insect intelligence. Furthermore, we believe that our work holds significant implications for the development of insect-inspired autonomous robots, offering insights that could pave the way for more adaptive and efficient robotic systems (*7, 8*).

Results

The constructed robot and representative examples of simulation environments are depicted in Figure 1, illustrating the intricate interactions between brain, body, and environment. The neural models can be integrated into the *Controller* component of the Webots' Robot. The morphology and spatial configurations of individual organs and limbs are reconstructed based

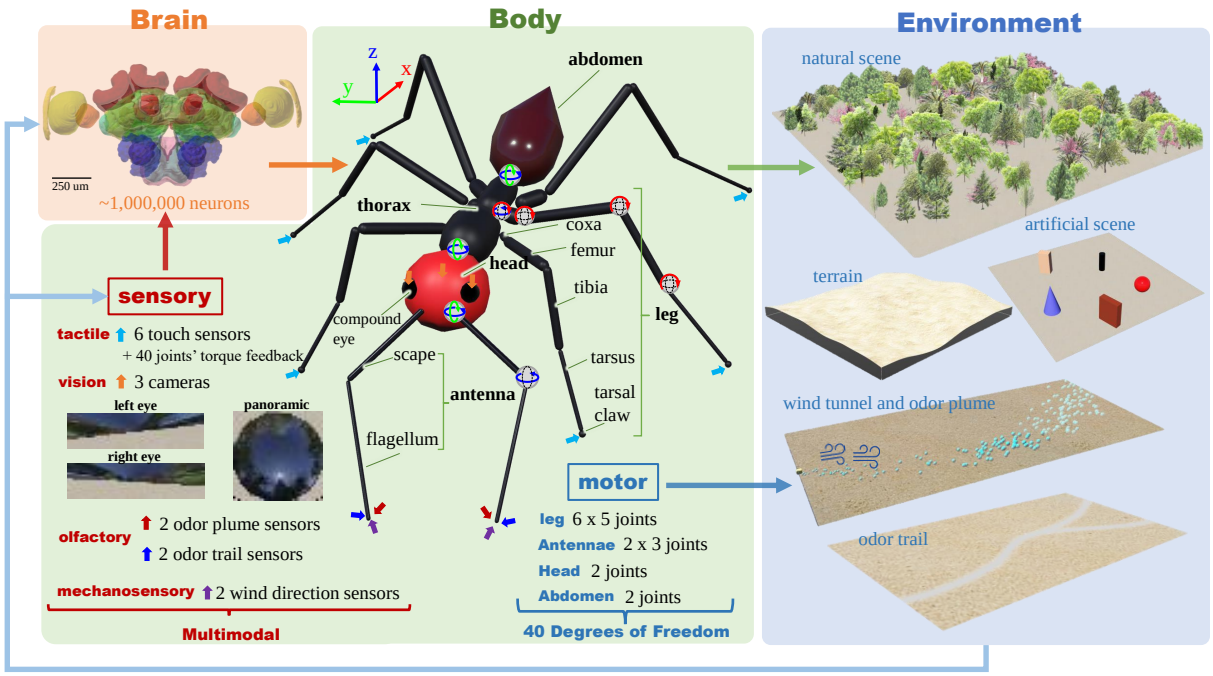

Figure 1: I2Bot summary. The implementation of the proposed tool reflects the interaction among brain, body, and environment, encompassing the sensory-motor loop. Color-coded arrows indicate the positions of corresponding sensors, while joints are identified by their rotation axes.

on measurements obtained from real desert ants (refer to Robot design for detailed information). The environments are assembled using Webots' built-in modules (refer to Environment construction for detailed information).

In this section, we will first introduce the sensory-motor system and the fundamental locomotion capabilities of the proposed ant robot. Following this, as summarized in Table 1, we will present several case studies demonstrating the utilization of the proposed tool to implement classical control scenarios in a sensory-motor loop fashion. These simulations provide compelling examples of how I2Bot can offer new insights into understanding insect intelligence through the dynamic interaction between brain, body, and environment.

Table 1: Summary of the presented case studies with I2Bot. Text in red highlights the new insights and contributions.

	Sensor	Motor	Environment
path integration	velocity + direction	6 legs	plain terrain
	Bio-plausible neural output and simple gait control facilitate homing.		
visual beacons	binocular/monocular vision	6 legs	plain terrain and 3D trees
	Embodied sensory input effects the performance of controller		
visual compass	panoramic vision	6 legs	plain terrain and 3D trees
	How internal representation of direction could control body orientation		
odor trail following	odor concentration (stable)	6 legs + 2 antenna	optic-simulated odor trail
	Embodied sensors simplify the design of the controller		
odor plume tracking	odor concentration (fluctuated) + wind direction	6 legs	virtual wind and odor puffs
	First implementation of the embodied plume tracking		

Sensors and motors

Insects rely on multimodal sensory information to perceive their surrounding environment and make decisions (*34*). Among these modalities, vision, olfaction, and tactile cues are considered crucial for insect navigation. A valuable tool should facilitate easy access to these sensory signals. To address this, the ant robot has been equipped with visual, olfactory, and tactile sensors. As depicted in Figure 1, two distinct vision sensors—binocular and panoramic—have been incorporated to serve different purposes in visual processing. Similarly, the proposed tool enables simulation of olfactory signals with various spatial-temporal patterns, such as stable odor trails for simulating trail following and dynamic odor plumes for investigating navigation in turbulent olfactory environments.

The locomotion capabilities of insects have long intrigued researchers and have served as inspiration for the design of six-legged robots (*4, 8*). To replicate the authentic locomotion of insects, we have defined five degrees of freedom (DoF) for each leg (i.e., front left-FL, middle left-ML, hind left-HL, front right-FR, middle right-MR, hind right-HR). The joints are designated based on the anatomy of body and leg segments. For instance, the joint ThCz refers to the joint connecting the *Thorax* and *Coxa*, rotating along the z-axis. To streamline body

movement, the head and abdomen each possess two degrees of freedom, allowing rotation along the y and z axes. Recent studies have emphasized the pivotal role of active olfactory sensing (*35, 36*) in shaping the navigational behaviors of insects. Therefore, each antenna of I2Bot is equipped with three degrees of freedom to enable agile movement of the antenna.

Basic locomotion

Locomotion is crucial for insect navigation, and we have implemented a simple gait control algorithm based on forward and inverse kinematics (FK and IK). As depicted in Figure 2 (a) and (b), inverse kinematics-based gait control exhibits greater stability but demands more computational resources. To assess the motion control's performance, we evaluated its ability to navigate uneven terrain and climb walls. Figure 2 (c) illustrates the comparison of walking on floors with varying degrees of unevenness. Here, we observed faster changes in the z-position of the body with negligible differences in joint torques, indicating the six-legged robot's inherent capacity to navigate uneven terrain embedded in its physical dynamics. In climbing vertical walls, insects utilize adhesion, we simulated this by dynamically applying force to the leg's ground contact points. Figure 2 (d) presents the climbing performance with varying levels of force applied to the ground support points. Interestingly, larger force doesn't necessarily translate to better wall-climbing performance. This may be due to the competition between the force required to lift the leg and the force needed to maintain contact with the wall Different gait types are demonstrated in Figure 2 (g,h), with the tripod gait achieving the fastest body movement speed, while the wave tripod is the slowest. It's worth noting that other gait controllers such as CPG and feedback models (*4, 37–39*) could also be integrated. Here, we present the simple FK-based gait controller to showcase the proposed tool's feasibility.

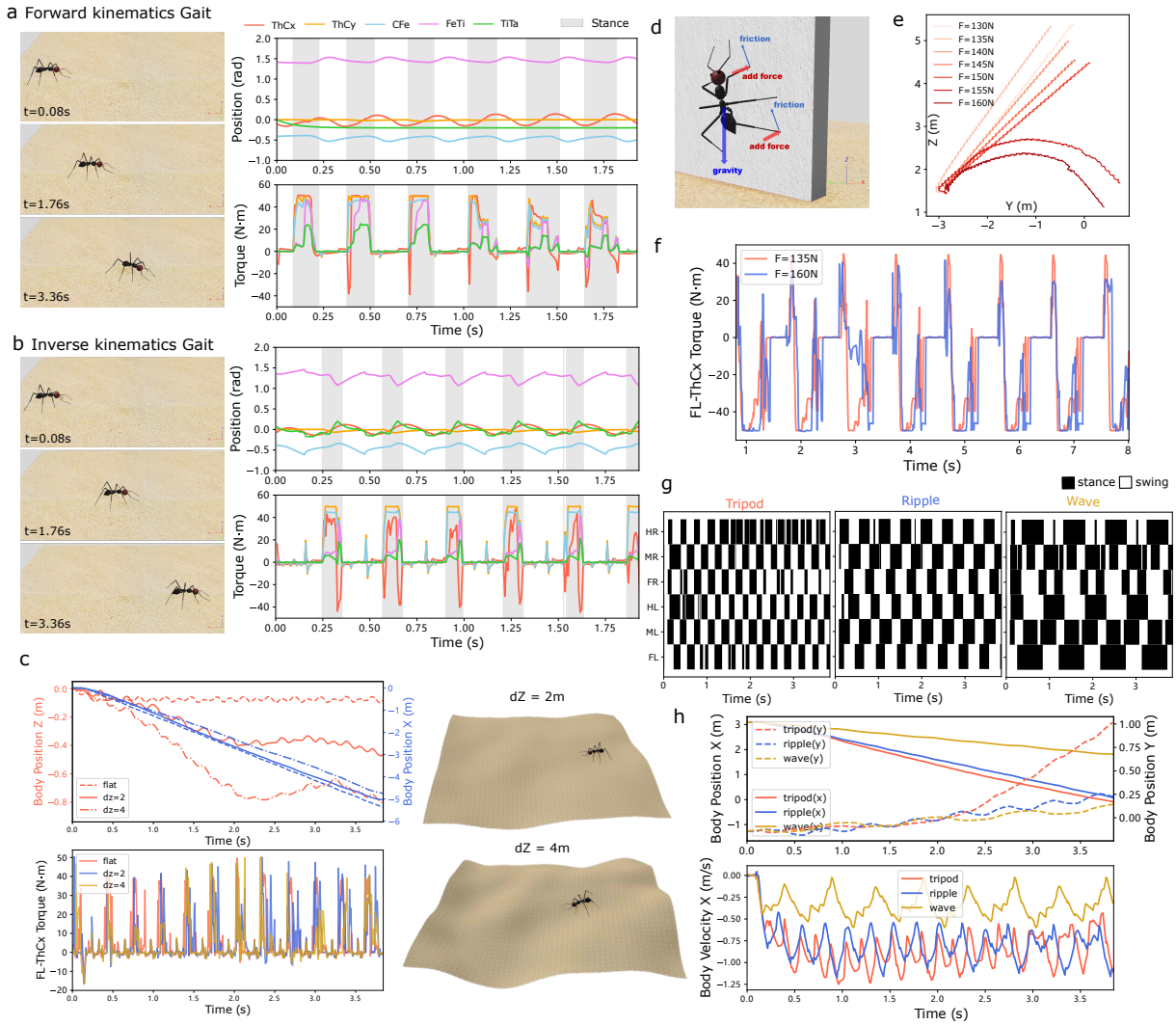

Figure 2: Locomotion and gait control. (a)-(b) Left: snapshots of walking ants at specific times; Right: joint position and torque of the middle right leg based on forward and inverse kinematics. (c) Walking on uneven terrain. (d) Climbing the wall using adhesion. (e) Forward kinematics-based implementation of different gait types: tripod, ripple, and wave.

Path integration: incorporating locomotion with bio-constrained neural model

Path integration (PI) stands as a cornerstone of the insect navigation toolkit (*40*). Foragers adeptly track the distance and direction to their nest by integrating the series of directions and distances traveled into a *home vector* (*11, 41*). Guided by this vector, pointing towards the start

position (typically the nest), desert ants can return accurately to their nest even after traversing hundreds of meters (Figure 3a). Recent neuroethological investigations have unveiled the central complex (CX) within the insect brain as pivotal for the computational processes involved in path integration. The CX not only hosts a ring attractor network encoding the animal's heading direction (*42, 43*) but also receives optic flow information as a velocity encoding (*21, 44*).

To demonstrate the ease of incorporating biologically constrained neural models with locomotion controllers using I2Bot, we implemented the popular insect path integration model proposed by (*21*). We utilized its output as tuning factors for the previously described forward kinematics gait controller. Specifically, the neural activation of the left and right PFL/CPU1 neurons from the insect steering circuit modifies the *hip swing* (speed) and *rotation* (direction) of the gait (Figure 3b,c). This marks the first instance of testing this popular neural model in a hexapod robot with physical constraints and ant-like morphology. Notably, the simulated ant can perform curved walking, unlike a recent biorobotics study (*24*) where the robot could only rotate on the spot. Several simulations were conducted, and we found that this bio-constrained model performed effectively (see Figure 3d,e), suggesting a robust and efficient solution for robot path integration. Figure 3f&g illustrate the dynamic neural activation of the encoded *home vector* and the agent's velocity during foraging and homing. This exemplifies how neural computation and encoding, coupled with a simple locomotion controller, can generate robust navigation behaviors.

Vision-guided maneuvers

Vision serves as one of the primary means for animals to perceive their environment, and insects rely on visual cues for navigation in various ways (*45, 46*). In this section, we present scenarios involving monocular, binocular, and widely used panoramic vision to demonstrate vision-motor control in the simulated ant robot.

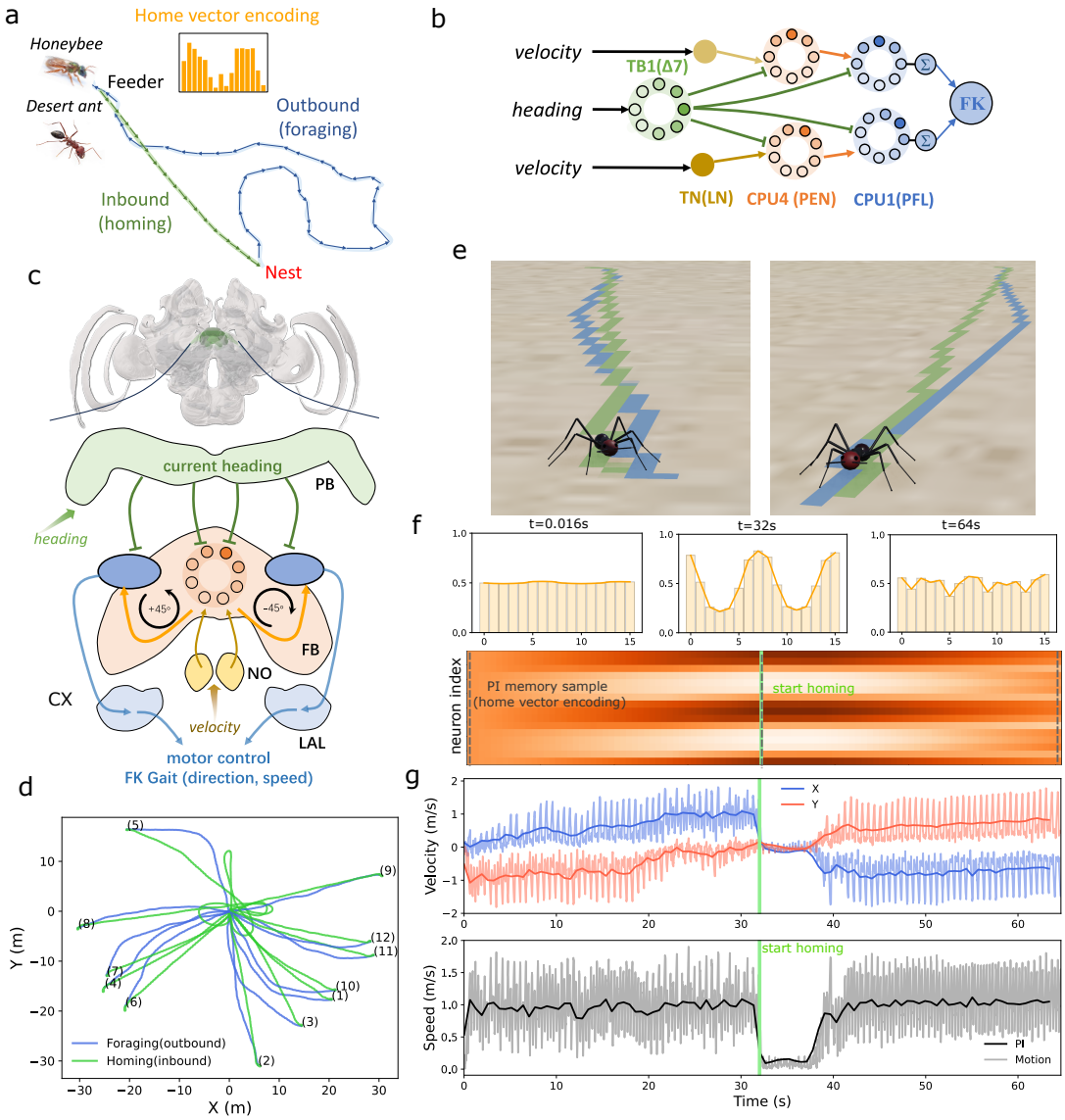

Figure 3: Implementation of path integration model. (a) Schematic diagram of central-foraging insects' path integration. (b) The neural model of path integration adapted from (*21*). The output of the model is linked to the forward kinematics gait controller. (c) Insect brain regions within the central complex related to PI computation. (d) Trajectories of simulated ants conducting foraging trips and homing using the proposed PI model. (e) Snapshots of two typical experiments. Trajectories are drawn using the *Pen* function within Webots. (f) Neural activation of PI memory neurons (CPU4 or FPN neurons) during foraging and homing, with sampled points marked by gray dotted lines. Corresponding activation profiles are placed above. (g) Velocity in the x- and y-axis (top) and speed (bottom) during navigation. Dark solid lines indicate the inputs fed into the PI model, while lighter curves show instant velocity measurements of the walking robot.

Monocular and binocular visual beacons

Research has indicated that insects instinctively use visual landmarks as beacons (*47, 48*). To showcase the feasibility of using visual images as input with our proposed tool, we implemented a simple visual beacon algorithm (see Figure 4a,b) based on the biologically plausible steering circuit used in neural models of the insect brain (*21, 22, 44*). To compare the effects of different types of visual inputs on motor output, we employed monocular (i.e., solely left or right image) and binocular images as inputs to the visual beacon model. The results are illustrated in Figure 4d, where the simulated ant with binocular image inputs outperforms others When ranked by time to complete the task. Eye occlusion resulted in ipsilateral motor bias, where agents with the left eye occluded (receiving only right eye information) approached the landmark with left-biased trajectories (red curves in Figure 4c). This illustrates how embodied sensory input influences navigation performance guided by the same neural controller, providing partial insights into the advantages of binocular vision.

Panoramic visual compass

In addition to binocular vision, insects are believed to use panoramic views when navigating (*49–51*). Computational models also leverage panoramic images to extract useful information for solving visual navigation tasks, particularly in the frequency domain (*22, 52, 53*). Here, we demonstrate how processing panoramic views in the frequency domain can aid the agent in sensing direction, known as the visual compass (*51, 54*). The methods used to process the panoramic view here are the same as those in (*22, 53*), which could also serve as the foundation for insect visual route following behavior (see Figure 5a). The initially extracted phase encoding in the frequency domain is stored as the desired heading direction. When the simulated 3D world is manually rotated, the received panoramic image and the phase encoding rotate accordingly. The difference between the stored phase and the current phase is output to

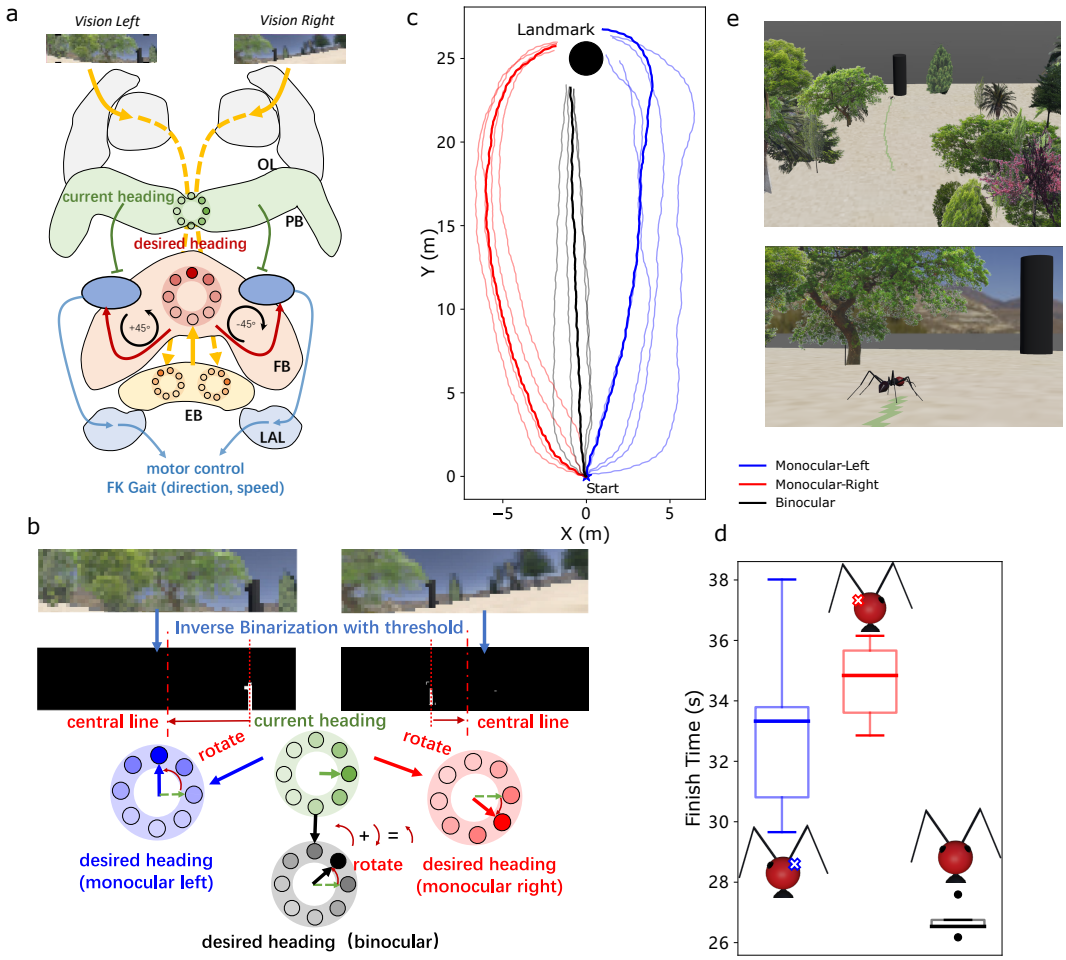

Figure 4: Visual beacons with monocular and binocular vision. (a) Schematic diagram of the visual beacons model overlapped with the insect brain regions. (b) Illustration of the generating desired heading from the visual inputs with monocular and binocular visual input. (c) Trajectories of the visual beacon using left, right, and binocular vision. (d) Time taken by the robot to reach the position near the landmark. (e) Snapshots of the simulated ant robot and environment during visual beacon navigation.

the steering circuit, enabling the agent to turn and continuously maintain its initial heading. To test the performance of the visual compass-based head tracking, we rotated the 3D world in two modes: incremental rotation and jump rotation (sudden and fast). As shown in Figure 5c and supplementary Video 8-9, the simulated ant efficiently tracked its stored initial heading in both

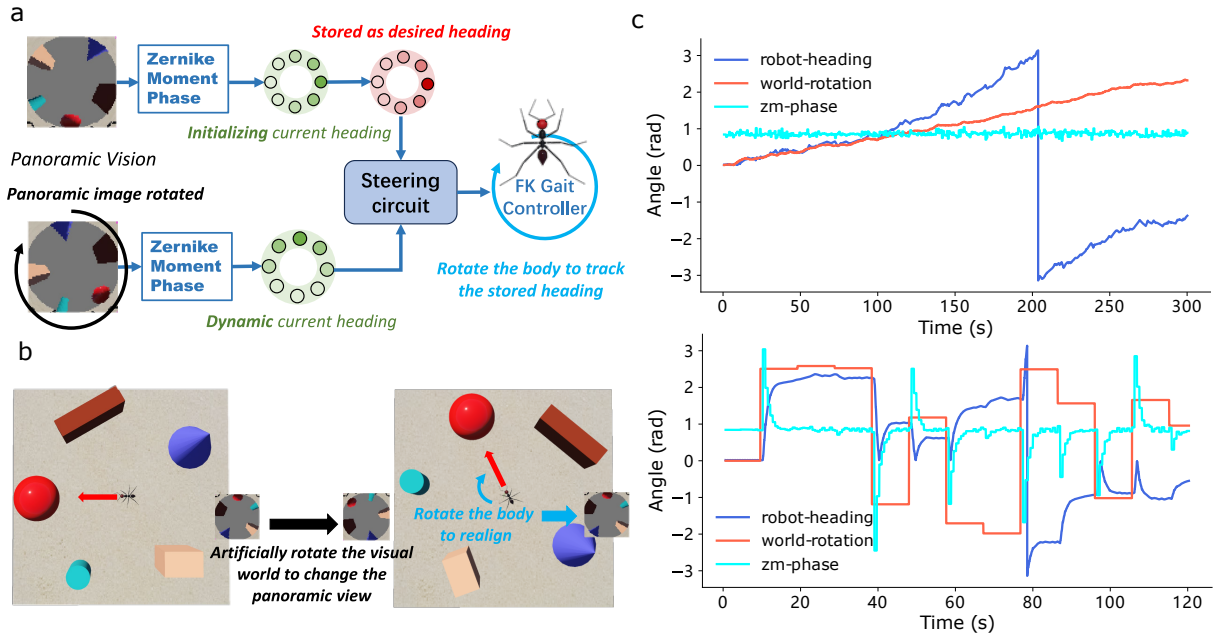

Figure 5: Visual compass using panoramic view. (a) Schematic diagram of the visual compass model. (b) Screenshot of the rotating simulated world from Webots. The rotation mode of the visual scene has two types: incremental (top of (c)) and jump (bottom of (c)). (c) The heading direction of the robot, the visual scene, and the phase of the Zernike moment are plotted when tracking the stored direction by the Zernike moment based visual compass.

rotation modes, consistently orienting towards the red ball in the 3D visual world.

Olfactory-guided maneuvers

Olfaction represents another crucial sensory domain guiding a diverse array of navigational behaviors (*55–58*). Different olfactory landscapes lead to varied navigation behaviors (*59*). To showcase how the proposed platform facilitates olfactory navigation simulation under environments with different spatial-temporal features, we implemented odor trail following behavior under a spatially stable distribution (*60*) and odor plume tracking behaviors in a turbulent olfactory environment (*57*).

Odor trail following using active sensing

Odor trail following (*36*) observed in carpenter ants (*Camponotus pennsylvanicus*) was realized to demonstrate how embodiment benefits the design of control strategies. We compared the trail following performance of agents with and without moving their antennae during odor tracking by measuring the time taken to navigate from the same start point to the end of the pheromone trail (finish time as shown in Figure 6d). Intriguingly, we found that agents with moving antennae significantly outperformed those with fixed antennae (Figure 6c,d) when tracking spatially wider trails, although their performance on trails with normal width was similar. This suggests that dynamic antennae movement enhances the adaptability of control strategies to handle a wide range of environmental conditions, as demonstrated in (*36*). This finding underscores how embodiment facilitates the design of effective control systems (*2*).

In addition, I2Bot simplifies body modification. For example, supplementary Video12 demonstrates a preliminary test where ant robots with varied antenna lengths, but identical control algorithms, exhibited different performances in odor trail following. This together with the results presented in the visual beacon simulations (Figure 4) further highlights the impact of body morphology and physical dynamics on the development of control algorithms and neural circuitry. It underscores that intelligence develops through the intricate interaction between body, brain, and environment (*1*).

Odor plume tracking

To investigate sensory-motor control in turbulent odor environments, we developed a simplified control strategy inspired by behaviors observed in walking flies (*57*). While this behavior has been extensively modeled in previous studies (*61, 62*), we adapted a conceptual model to validate the proposed tool's capacity to implement dynamic olfactory navigation behaviors. As illustrated in Figure 6e, once the odor is sensed (i.e., the perceived odor concentration exceeds

Table 2: Comparison between different simulation tools

	I2Bot	Hector	NeuroMechFly	NeuroMechFly2	CompoundRay
physics	✓ (ODE)	✓ (ODE)	✓ (PyBullet)	✓ (MuJoCo)	✘
vision	✓ (Binocular/Panoramic)	✘	✘	✓̃ (Binocular)	✓̃ (Binocular)
olfactory	✓	✘	✘	✓̃	✘
tactile	✓	✓̃	✓̃	✓	✘
Program Language	✓ (Python/Java/C/C++/Matlab)	✓̃ (Python)	✓̃ (Python)	✓̃ (Python)	✓̃ (Python)
User Interface	✓	✘	✘	✘	✘

the threshold), the agent orients upwind (aligning with the wind direction detected via the virtual mechanosensory signal). In the absence of odor, the agent performs random rotations in place, simulating scanning/searching maneuvers. Despite discontinuous olfactory and mechanosensory cues in this turbulent environment, agents successfully navigate to the odor source using this simple strategy (see Figure 6g and Supplementary Video13).As the first attempt to implement dynamic odor plume tracking in a simulated robot with realistic ant morphology and physical constraints, this case study may provide useful foundational work for future investigations into embodied olfactory-motor control.

Discussion

We presented I2Bot, an open-source tool for studying the mechanisms of multimodal and embodied insect navigation. Case studies demonstrate the the usability and flexibility of the proposed platform, which has the potential to accelerate research in related fields. Compared to other simulation tools (see Table 2), Hector (*63*) and NeuroMechFly (*26*) primarily focus on locomotion, while CompoundRay (*64*) is designed for insect vision reconstruction. The most similar tool is the recently developed NeuroMechFly2 (*27*), which also enables sensory-motor closed-loop simulation. However, I2Bot offers more precise and flexible olfactory stimulus

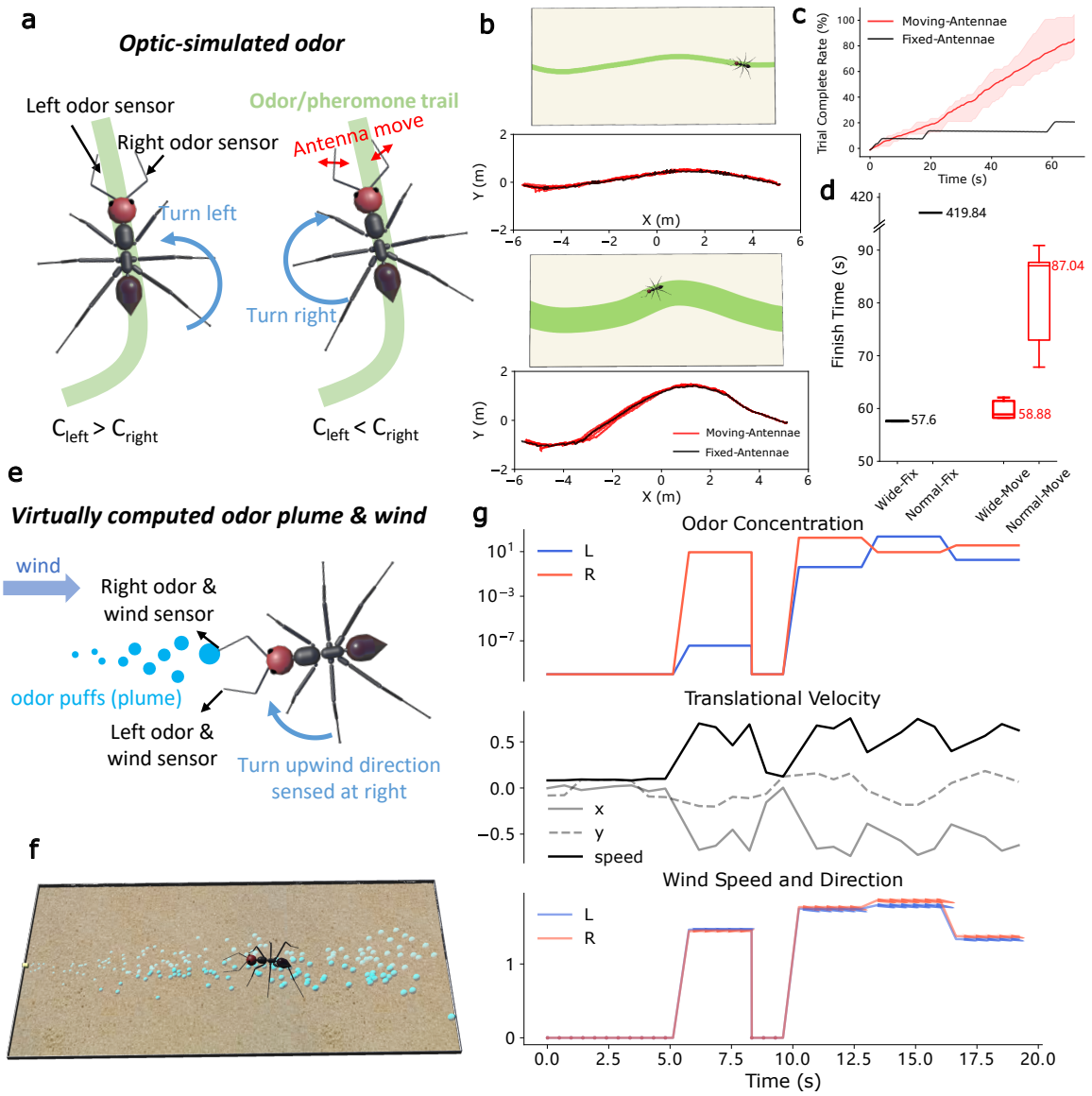

Figure 6: Olfactory-motor closed-loop control. (a)-(d) depict odor trail following, while panels (e)-(g) illustrate odor plume tracking. (a) Schematic diagram illustrating the control strategy of the trail following behavior. (b) Bird's-eye screenshots of the experiments and the corresponding trajectories of the agent with different widths of the odor trail. (c) Comparison of the completion rate between fixed-antenna and moving-antenna configurations. (d) Box plot of the finish time (n=5). (e) Model of plume tracking. (f) Screenshot of the experiments in the Webots 3D scene. (g) Perceived odor concentration (top), wind (bottom), and the moving velocity of the ant robot (middle) during one experiment.

generation and a more user-friendly interface, making it easier to learn and implement algorithms. Additionally, I2Bot provides more flexible visual and olfactory world construction and is simpler to extend with other tools. For example, CompoundRay (*64*) can be integrated with the camera model to generate more realistic insect visual inputs. Regarding the physics engine, both the Open Dynamic Engine (ODE) and MuJoCo are popular in the robotics field, each with its own pros and cons for physics simulation (for detailed comparison, see (*65, 66*)).

I2Bot can also be used as a (deep) reinforcement learning (RL/DRL) agent (*67, 68*), similar to NeuroMechFly2 (*27*). The sensory and body kinematic states of the robot can construct the *State Space*, while the motor commands sent to all available joints form the *Action Space*. The core of the RL system lies in the agent's dynamic interaction with the environment. The I2Bot platform offers a high variety of interaction forms that interest researchers from biology and robotics, facilitating research into the underlying mechanisms of intelligent behaviors and the training and testing of RL-based algorithms.

I2Bot is also suitable for simulating multi-agent systems, making it ideal for swarm intelligence and robotics studies (*69, 70*). Similar to previous studies using Webots for swarm robotics simulation (*71–73*), I2Bot can easily be applied to investigate similar scenarios by adding multiple ant robots (see supplementary Video12 for an example of multiple agents simulation). This approach can lead to embodied swarm intelligence, as the body morphology and physical dynamics are integrated into this platform.

Several features could be integrated into I2Bot in the future. Currently, I2Bot supports simulating only one type of odor at a time, but it could easily support multiple odors simultaneously by adding more identical plume simulators. Additionally, other useful tools like CompoundRay (*64*) can be efficiently embedded into I2Bot, thanks to the interfaces provided by Webots and the excellent properties of its *Controller*. It is also possible to use the recently developed FARMS framework (*74*) to build the model of I2Bot since they are compatible. These

flexibilities will allow researchers to continuously enhance the platform's capabilities and apply it to a broader range of studies in insect navigation and robotics.

Materials and Methods

I2Bot is build by the popular open-source robot simulator Webots (*29*). The controllers of the demos presented in this paper was programmed by Python 3.9 with packages such as *numpy*, *cv2*, *ikpy*, etc. Note that aside from Python, Webots supports multiple programming languages like Java, C/C++ and Matlab scripts. All the source codes, model of the robot and environments are open-sourced via Github `https://github.com/XuelongSun/I2Bot` under the MIT license.

Robot design

The morphology design of the ant is based on Wehner's measurement of the desert ant *Cataglyphis fortis* (*75*). We use computer vision method to correspond the length in picture and transfer it to real size using scale in Wehner's picture of the desert ant. To simplify the 3D constructing of the robot's body, we use Webots embedded geometries *sphere*, *cone* and *capsule* to model the body parts as that in (*32*).

As shown in Supplementary FigS1, the length of the hind leg is larger, making it distinctive. Note that the size of the ant is scaled 100 times larger for the ease of the physics simulation within Webots.

Environment construction

For visual environments, all visible objects, such as basic geometric shapes and trees used in the presented cases, are available in Webots. Vision is constructed using the camera model embedded in Webots. Additionally, simulated visual environments from previous studies on

Table 3: The calculation of ThCx joint range R for yaw-control of forward kinematics gait

	left legs (FL, ML and HL)	right legs (FR, MR and HR)
$\theta \geq 0$	$S_l d(2\theta - 1)$	$S_l d$
$\theta < 0$	$S_l d(2\theta + 1)$	$-S_l d$

insect navigation can be easily imported into the Webots 3D environment via the *Mesh* shape or *Cadshape* node. See Supplementary FigS2 for examples of visual scenes used in (*20, 22, 53, 76*). This feature allows researchers to directly obtain visual stimuli from the desired world without needing to configure a camera model, facilitating the comparison of different visual navigation models that share the same visual input.

For olfactory environments, there are two types of simulation: 1) for the odor trail, it is optically simulated as the specific texture of the floor. This texture can be drawn manually thus allows great flexibility. 2) for the odor plume, it is simulated by the filament-based model (*77*) and its Python implementation called *pompy*. This model mimics both short- and long-timescale features of odor plume evolving in a turbulent flow and has been widely used in olfactory related studies involving insect navigation (*55, 78*) and robot odor source localization (*79*).

Models

Forward kinematics

To simply the gait control, we keep the ThCx and TiTa joint to be constant and denote the angles of the left three joints- ThCz, CTr and FTi to be α, β and γ respectively. These values are determined by the hip swing (i.e., the step length S_l), lift swing (i.e., the step height S_h) and the moving direction (-1 for backwards and 1 for forwards). To realize yaw control of the robot body, we introduced rotation configuration denoted as θ. Table 3 lists the formulas to calculate the base range of the ThCx joint R which is used to compute the angle at a certain time step k during the gait circle with S_n steps:

$$\begin{cases} \alpha(k) = & -R + \alpha_0 + \frac{2R}{S_n}k \\ \beta(k) = & \beta_0 + \frac{S_h}{S_n}k \\ \gamma(k) = & \gamma_0 + \frac{S_h}{S_n}k \end{cases} , k = 0, 1, ..., S_n \tag{1}$$

where $\alpha_0, \beta_0, \gamma_0$ is the initial angle of ThCx, CTr and FTi joint respectively.

Inverse kinematics

Unlike the forward kinematics wherein the joint angles are computed directly given the gait parameters, when applying inverse kinematics, we first calculate the positions at a certain time step k of the leg tip in swing and stance state, and then utilize the inverse kinematics (with python package-*ikpy*) to compute the joint angles.

Path integration

The path integration model is adapted from (*21*), the output of the summed CPU1 (PFN) neurons are fed into the FK-based gait controller as following:

$$\theta(t) = k_{PI}[\sum_{i=0}^{7} C^i_{CPU1}(t) - \sum_{i=8}^{16} C^i_{CPU1}(t)] \tag{2}$$

where $C^i_{CPU1}(t)$ is the membrane potential of the ith CPU1 neuron at time t. Similar with that in (*21*), the difference between the summed activation of left and right CPU1 neurons modify the rotation configuration of the gait control thus guide guide the robot's turning decision. To tune the walking speed, step length S_l is computed by:

$$S_l = S^0_l + S^f_l \times (1 - \min[|\theta(t)|, 1]) \tag{3}$$

where the basis step length $S^0_l = 5°$ and the scale factor $S^f_l = 20°$ are set to be constant. Thus, the step length is confined in the range of $[5°, 25°]$.

Vision

Vision is provided through the embedded camera sensor in Webots. For panoramic view simulating, the projection mode of the camera is set to *spherical* while set to be *planar* in monocular and binocular settings. The parameters of the camera such as resolution, focal length can be customized in Webots easily through the scene tree.

- visual beacons: The resolution of the left and right eye camera is 74×19 with the horizontal and vertical field of view to be $2rad$ and $0.5rad$ respectively. The motor command is calculated by the 'copy-and-shift' mechanism proposed in (*22, 61*). The value of the *shift* is computed by extracting the averaged horizontal position of the dark landmark in the retina (i.e., image pixel domain). Then the motor value is fed into the steering circuit as that in PI model.

- visual compass: The resolution of the artificial panoramic vision is 72×72. The direction of the view is extracted by the phrase information of the Zernike moment (*22, 53*) coefficient (with order $n = 7$ and repetition $m = 1$). This direction is compared with the initial direction and then the difference determines turning parameter $\theta(t)$ of the FK gait control.

Olfactory

- odor trial following: As described above, the odor trail is simulated by the texture of the floor, thus the concentration is detected by the *DistanceSensor* (field type is set to be "infra-red") whose value is modified by a reflection factor depending on the color, roughness and occlusion properties of the object. The detected odor concentration of the odor (returned value of the *DistanceSensor*) from the left and right antenna are denoted as O_l and O_r, the difference between the left and right sensed odor concentration is denoted

Table 4: Odor trail following. The threshold $T_d = 5$ and $T_o = 900$ are set empirically according to the lookup table of the *DistanceSensor* used. $K_d = 400$ is a constant scale factor.

sensory state	odor sensed		no odor
	$O_d < T_d$	$O_d \leq T_d$	$O_l < T_o$ & $O_r < T_o$
θ (degree)	0	O_d/k_d	-1 or 1 (random)
S_l (degree)	8	4	3

as $O_d = O_l - O_r$. Table 4 shows how the sensed odor concentration determines the motor control (i.e., determines the hip swing and rotation of the FK gait control).

- odor plume tracking: In this simulation, the agent not only detects the odor concentration (O_l and O_r) but also the wind direction ($\mathbf{W_l} = [w_l^x, w_l^y]$ and $\mathbf{W_r} = [w_r^x, w_r^y]$). These values are computed virtually based on the filament-based model. When there is no odor sensed, the agent conducts a random searching around the spot (set θ to be -1 or 1 randomly and S_l to be a small value like $5°$). While the odor is sensed (i.e., the concentration value exceeds $T_o^p = 10^{-6}$), the agent compares its left and right odor concentration and turns to the up-wind direction sensed with higher odor concentration. Specially, the desired turning orientation is computed by $m = \tan^{-1}(w_y/w_x)$. Then the step length and rotation are set to be:

$$\begin{cases} \theta = & \text{sign}(m - h) \\ S_l = & \min([5 + |\text{d}\%5|, 15]) \end{cases} \tag{4}$$

where h is the current heading of the ant robot. Note that the agent will conduct 1 gait loop using the parameters defined in the above equation and then going forward ($S_l = 15, \theta = 0$) for 2 gait loop.

References

1. H. J. Chiel, R. D. Beer, *Trends in Neurosciences* **20**, 553 (1997).

2. A. Wystrach, *Biochemical and Biophysical Research Communications* **564**, 70 (2021).

3. B. Webb, *Behavioral and Brain Sciences* **24**, 1033 (2001).

4. A. J. Ijspeert, *Science* **346**, 196 (2014).

5. N. Franceschini, *Proceedings of the IEEE* **102**, 751 (2014).

6. P.-Y. Oudeyer, *IEEE Transactions on Autonomous Mental Development* **2**, 2 (2010).

7. B. Webb, *Science* **368**, 244 (2020).

8. P. Manoonpong, *et al.*, *Sensors* **21**, 7609 (2021).

9. A. Matsiko, *Science Robotics* **8**, eadi2720 (2023).

10. R. Wehner, *Journal of Comparative Physiology A* **205**, 641 (2019).

11. R. Wehner, *Journal of Comparative Physiology A: Sensory, Neural, and Behavioral Physiology* **189**, 579 (2003).

12. S. Heinze, *Current opinion in insect science* **24**, 58 (2017).

13. A. Wystrach, M. Mangan, B. Webb, *Proceedings of the Royal Society B: Biological Sciences* **282**, 20151484 (2015).

14. R. Wehner, T. Hoinville, H. Cruse, K. Cheng, *Journal of Comparative Physiology A* **202**, 459 (2016).

15. S. Heinze, *Current Biology* **30**, R984 (2020).

16. P. Bregy, S. Sommer, R. Wehner, *Journal of Experimental Biology* **211**, 1868 (2008).

17. C. Bühlmann, K. Cheng, R. Wehner, *Journal of Experimental Biology* **214**, 2845 (2011).

18. E. L. Legge, A. Wystrach, M. L. Spetch, K. Cheng, *Journal of Experimental Biology* **217**, 4159 (2014).

19. M. Collett, *Current Biology* **22**, 927 (2012).

20. P. Ardin, F. Peng, M. Mangan, K. Lagogiannis, B. Webb, *PLOS Computational Biology* **12**, e1004683 (2016).

21. T. Stone, *et al.*, *Current Biology* **27**, 3069 (2017).

22. X. Sun, S. Yue, M. Mangan, *Elife* **9**, e54026 (2020).

23. D. Lambrinos, R. Möller, T. Labhart, R. Pfeifer, R. Wehner, *Robotics and Autonomous systems* **30**, 39 (2000).

24. J. Dupeyroux, J. R. Serres, S. Viollet, *Science Robotics* **4**, eaau0307 (2019).

25. G. W. Brodland, *Seminars in cell & developmental biology* (Elsevier, 2015), vol. 47, pp. 62–73.

26. V. Lobato-Rios, *et al.*, *Nature Methods* **19**, 620 (2022).

27. S. Wang-Chen, *et al.*, *bioRxiv* pp. 2023–09 (2023).

28. R. Vaxenburg, *et al.*, *bioRxiv* pp. 2024–03 (2024).

29. O. Michel, *International Journal of Advanced Robotic Systems* **1**, 5 (2004).

30. Y. Xin, X. Rong, Y. Li, B. Li, H. Chai, *IEEE Access* **7**, 133265 (2019).

31. J. Knüsel, A. Crespi, J.-M. Cabelguen, A. J. Ijspeert, D. Ryczko, *Frontiers in neurorobotics* **14**, 604426 (2020).

32. P. Ramdya, *et al.*, *Nature communications* **8**, 14494 (2017).

33. Y. Chen, J. E. Grezmak, N. M. Graf, K. A. Daltorio, *Bioinspiration & Biomimetics* **17**, 046001 (2022).

34. C. Buehlmann, M. Mangan, P. Graham, *Animal cognition* **23**, 1129 (2020).

35. N. Claverie, P. Buvat, J. Casas, *Integrative and Comparative Biology* **63**, 315 (2023).

36. R. W. Draft, M. R. McGill, V. Kapoor, V. N. Murthy, *Journal of Experimental Biology* p. jeb.185124 (2018).

37. E. Arena, P. Arena, L. Patané, *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)* (IEEE, 2012), pp. 1341–1346.

38. M. Schilling, T. Hoinville, J. Schmitz, H. Cruse, *Biological cybernetics* **107**, 397 (2013).

39. M. Schilling, H. Cruse, *PLoS computational biology* **16**, e1007804 (2020).

40. R. WEHNER, *Navigation* **55**, 101 (2008).

41. S. Heinze, A. Narendra, A. Cheung, *Current Biology* **28**, R1043 (2018).

42. S. S. Kim, H. Rouault, S. Druckmann, V. Jayaraman, *Science* **356**, 849 (2017).

43. B. K. Hulse, V. Jayaraman, *Annual review of neuroscience* **43**, 31 (2020).

44. A. Honkanen, A. Adden, J. Da Silva Freitas, S. Heinze, *Journal of Experimental Biology* **222**, jeb188854 (2019).

45. M. Collett, L. Chittka, T. S. Collett, *Current Biology* **23**, R789 (2013).

46. C. A. Freas, M. L. Spetch, *Animal Cognition* **26**, 319 (2023).

47. T. S. Collett, J. Baron, *Nature* **368**, 137 (1994).

48. C. Buehlmann, *et al.*, *Current Biology* **30**, 3438 (2020).

49. J. Zeil, M. I. Hofmann, J. S. Chahl, *JOSA A* **20**, 450 (2003).

50. A. Wystrach, G. Beugnon, K. Cheng, *Frontiers in Zoology* **8**, 1 (2011).

51. T. S. Collett, *Current Biology* **18**, R1058 (2008).

52. W. Stürzl, R. Möller, *International Work-Conference on the Interplay Between Natural and Artificial Computation* (Springer, 2007), pp. 61–70.

53. T. Stone, M. Mangan, A. Wystrach, B. Webb, *Interface focus* **8**, 20180010 (2018).

54. R. Wehner, K. Cheng, H. Cruse, *The new visual neurosciences* pp. 1153–1163 (2014).

55. R. T. Cardé, M. A. Willis, *Journal of Chemical Ecology* **34**, 854 (2008).

56. K. Steck, B. S. Hansson, M. Knaden, *Frontiers in Zoology* **6**, 1 (2009).

57. E. Álvarez-Salvado, *et al.*, *eLife* **7**, e37815 (2018).

58. T. J. Steele, A. J. Lanz, K. I. Nagel, *Journal of Comparative Physiology A* **209**, 467 (2023).

59. G. Reddy, V. N. Murthy, M. Vergassola, *Annual Review of Condensed Matter Physics* **13**, 191 (2022).

60. M. Traner, B. Raman, *bioRxiv* pp. 2023–02 (2023).

61. X. Sun, S. Yue, M. Mangan, *eLife* **10**, e73077 (2021).

62. A. M. Matheson, *et al.*, *Nature communications* **13**, 4613 (2022).

63. V. Dürr, *et al.*, *Frontiers in neurorobotics* **13**, 88 (2019).

64. B. Millward, S. Maddock, M. Mangan, *Elife* **11**, e73893 (2022).

65. T. Erez, Y. Tassa, E. Todorov, *2015 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, Seattle, WA, USA, 2015), pp. 4397–4404.

66. M. Körber, J. Lange, S. Rediske, S. Steinmann, R. Glück, Comparing Popular Simulation Environments in the Scope of Robotics and Reinforcement Learning (2021).

67. M. A. Wiering, M. Van Otterlo, *Adaptation, learning, and optimization* **12**, 729 (2012).

68. Y. Li, *arXiv* (2017).

69. M. Dorigo, *et al.*, *Scholarpedia* **2**, 1462 (2007).

70. M. Brambilla, E. Ferrante, M. Birattari, M. Dorigo, *Swarm Intelligence* **7**, 1 (2013).

71. B. Khaldi, F. Cherif, *International Journal of Computer Applications* **126**, 31 (2015).

72. A. L. Newton, C. L. Nehaniv, K. Dautenhahn, *European Conference on Artificial Life* (Springer, 2003), pp. 829–838.

73. M. Raoufi, A. E. Turgut, F. Arvin, *Towards Autonomous Robotic Systems: 20th Annual Conference, TAROS 2019, London, UK, July 3–5, 2019, Proceedings, Part I 20* (Springer, 2019), pp. 263–274.

74. J. Arreguit, S. T. Ramalingasetty, A. Ijspeert, *bioRxiv* (2023).

75. R. Wehner, *Senckenbergiana biol* **64**, 89 (1983).

76. B. Baddeley, P. Graham, P. Husbands, A. Philippides, *PLoS computational biology* **8**, e1002336 (2012).

77. J. A. Farrell, J. Murlis, X. Long, W. Li, R. T. Cardé, *Environmental fluid mechanics* **2**, 143 (2002).

78. S. H. Singh, F. Van Breugel, R. P. N. Rao, B. W. Brunton, *Nature Machine Intelligence* **5**, 58 (2023).

79. G. Kowadlo, R. A. Russell, *The International Journal of Robotics Research* **27**, 869 (2008).

80. M. Mangan, *thesis, University of Edinburgh* (2011).

Acknowledgments

This research has received funding from the National Natural Science Foundation of China under the Grant No 62206066 and No 12031003.

Supplementary Materials

Figures:

- FigS1: The design of I2Bot's morphology is based on real desert ant's measurement.
- FigS2: Examples of importing previous used visual 3D world (*20, 22, 53, 76, 80*) into I2Bot environment.

Videos:

- Video 1 gait control using forward kinematics.
- Video 2 gait control on uneven terrain.
- Video 3 wall climbing using adhesion.
- Video 4 path integration.
- Video 5-7 visual beacons using monocular and binocular.

- Video 8-9 visual compass (jump and incremental mode).
- Video 10-11 odor trail following (normal and wide trail).
- Video 12 Odor trail following with varied antenna length.
- Video 13 odor plume tracking.