A movement model to predict resource discovery by central place foragers : an application to bumblebees

Ana Morán

Université Fédérale
Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse 3 - Paul Sabatier

Présentée et soutenue par

Ana MORÁN

Le 26 mai 2023

Un modèle de déplacement et découverte de ressources par des butineurs: le cas des bourdons

---

Ecole doctorale : **SEVAB - Sciences Ecologiques, Vétérinaires, Agronomiques et Bioingenieries**

Spécialité : **Ecologie, biodiversité et évolution**

Unité de recherche :
CRCA - Centre de Recherches sur la Cognition Animale

Thèse dirigée par
Jacques GAUTRAIS

Jury

Mme Carmen BESSA GOMES, Rapporteure
M. Simon BENHAMOU, Rapporteur
M. Richard FOURNIER, Examinateur
M. Jacques GAUTRAIS, Directeur de thèse
M. Frederic BARTUMEUS, Président

Acknowledgements

First, I would like to thank Simon Benhamou, Carmen Bessa-Gomes, Frederic Bartumeus and Richard Fournier for accepting to review this manuscript and be part of my defense jury. Thanks Antoine Wystrach, Cristophe Andalo and FX for being part of my thesis comittee. You were the best comittee I could've hoped for, you human qualities really helped me when I needed it.

My adventure in the CRCA started like 7 years ago, way before I started my PhD. Thank you Mathieu Lihoreau for welcoming in your team several years ago for my first internship. This all started thanks to you. Thank you Jacques for accompanying me during these 4 years. I am so happy that I got to learn so much from you.

Gracias Alfonso. I literally could not have finished this thesis without you. Thank you not only for helping me so much and guiding me, but thank you most of all for being one of the best friends I could have ever wish for.

J'ai eu l'immense chance de partager mon bureau avec des gens incroyables. Merci Aurèle pour tous les moments qu'on a partagé quand c'était juste nous deux dans le bureau aux 4R3. Je vais jamais oublier nos conversations et nos petages de câble face à nos ordis. Merci Antoine, Anthony et Alex pour votre bonne humeur et pour ameliorer exponentiellement mes journées.

Un grand merci à toutes les autres personnes que j'ai rencontré au CRCA: Merci Stephane et Matthias pour toujours être là pour moi, un peu comme des grands frères. Merci Flo pour ton sens de l'humour et pour toujours être disponible pour aider tout le monde autour de toi. Merci Gérard pour toutes nos conversation pendant tes pauses cafés, pour ta bonne humeur et pour m'avoir écouter quand j'en avais besoin. Merci Greg pour tous nos goûters. Merci Louise, Louis, Mathilde, Emilie, Manon, Mathieu, Ines, Nour, Julie, Alid, Flora, Renaud, Léo, Alex et Jordan.

Gracias Tamara por estar siempre ahí para mí, por escucharme, por todas las conversaciones sobre gatos. Gracias por todo. Gracias Olga y Gabriel por siempre preocuparse por mí, sé que puedo contar con ustedes para lo que sea.

Gracias a mi familia que no es de sangre. Eduardo, Linda, París, gracias por estar siempre aunque sea a la distancia.

Gracias a toda mi familia, especialmente a mis papás. Lo siento por desaparecer un poco a lo largo de estos años de tesis pero aquí estamos ya, terminando. Esto fue un trabajo en equipo con ustedes.

Gracias Mamama y Mamainda, sin ustedes no estaría aquí.

Abstract

Understanding how pollinators move across space is key to understanding plant mating patterns. Bees are usually assumed to search for flowers randomly or using simple movement rules so that the probability of discovering a flower depends primarily on its distance to the nest. However, experimental work shows this is not always the case. Until now, no one has successfully enunciated a realistic model of bee movement that considers the fact that they are Central Place Foragers and thus they start and end all their movements in the same place: the nest.

To further our knowledge of the exploratory movement of central place foraging bees, I propose a model of central place foraging that produces realistic bee trajectories by accounting for the autocorrelation of the bee's angular speed, the attraction to the nest (homing), and Gaussian noise. The four parameters of this model have been tuned based on experimental trajectories collected on bumblebees (Bombus terrestris) in the field. The model not only has the potential to describe the movement patterns of bees but also those of other central place forager animals.

The proposed model paves the way to compute theoretical predictions about pollination in the field. Here, I explored the statistics of flower discovery, depending on flower patch sizes and densities. Simulations of bumblebee trajectories highlight two effects that were previously overlooked: a *masking effect* that reduces the detection of flowers close to another and a *scale effect* that modulates this first effect as a function of the distance between flowers. At the plant level, flowers distant from the nest were more often discovered by bees in low-density environments. At the colony level, foragers found the most flowers when they were small and at medium densities. These results suggest that pollination would be optimized in a range of intermediate flower densities: when the density is too low, few flowers are discovered (due to the central place forager effect); when the density is too high, flowers distant from the nest become masked by closer ones (due to the masking effect).

These results indicate that the processes of search and discovery of resources are

potentially more complex than usually assumed, and question the importance of resource distribution and abundance on bee foraging success and plant pollination.

Preface

When considering a spatial arrangement of resources, including their density, size, and location, along with the behavior of a Central Place Forager, various questions arise. For instance, we can inquire about the accessibility of different resources, which ones are visited first, or even determine if there exists an optimal resource density that maximizes discovery. These questions lie at the intersection of statistical physics, which deals with the theoretical study of hitting processes, and behavioral ecology. While some formal results are available in theoretical physics, they cannot be readily applied to practical ecological scenarios. These results are typically applicable only in borderline situations where theoretical derivations are possible, such as infinite time or when the target size approaches zero. On the other hand, significant progress has been made in the field of Behavioral Ecology while studying Optimal Foraging. However, the focus of this research primarily centers around exploring the strategies employed by animals, rather than considering the perspective of the resources themselves.

To address these questions, two primary components are required: a model that represents the navigation and exploration behavior of a Central Place Forager, and a systematic examination of the influence of resource characteristics. For the purpose of this study, I will focus on bumblebees (Bombus terrestris) as a case study, as I have access to experimental data that captures realistic trajectories on an appropriate scale.

The objectives of this research are as follows:

1. Develop a comprehensive behavioural model for Central Place Foragers.

2. Calibrate the model using experimental data, specifically data obtained from bumblebees in this case.

3. Investigate the dynamics of the discovery process by analyzing the interplay between the properties of the behavioral model and the characteristics of the resources.

Develop a comprehensive behavioral model for Central Place Foragers.

In many modeling studies, the time sampling of data is explicitly incorporated into the random walk model. The model's parameters are adjusted as if the animal were genuinely moving in a straight line between the series of data points. Consequently, depending on the sampling frequency, the model can vary between studies, even when examining the same animal. In my case, the time series I have available for acquisition are separated by a relatively large time step (3 seconds), which significantly exceeds the characteristic timescale of bumblebee turning. If we were to treat the data as true trajectories, it would introduce a bias when computing discoveries. To mitigate this issue, I will adopt a model that represents the animal's behavior itself, rather than directly modeling the data. This approach assumes that the navigational model underlies the data series and will be adjusted accordingly. This navigational model operates in continuous time, free from the constraints of discrete time steps, allowing it to account for any type of navigational behavior exhibited by the bumblebees.

Calibrate the model using experimental data

Given the frequency of data acquisition, the set of parameters can not be directly adjusted. I will present a method to calibrate the model at continuous-time from a time series.

Investigate the dynamics of the discovery process by analysing the interplay between the properties of the behavioural model and the characteristics of the resources.

I added to the model a behavioural specific component to the decision process when exploring resource (in particular the detection of a resource and the crop capacity of the individual).

I'm using this comprehensive behavioural model to explore the discovery process numerically, by varying the properties of the resource field (densities, size etc.).

The work presented in this thesis has been partially published in Morán, A., Lihoreau, M., Escudero, A.P. & Gautrais, J. (2023). Modeling bee movement shows how a perceptual masking effect can influence flower discovery. *PLOS Computational Biology*, 19, e1010558. Publisher: Public Library of Science (Appendix A).

Contents

Chapter 1

Introduction

Contents

All living organisms, from bacteria to plants and animals, move at some stage of their life and it has consequences at the individual level, the population or even the species. It's an important process that allows to look for food, mating, shelter, avoid predators, etc. It can be active like the movement of bacteria using flagella, animals walking, or plants growing, or passive like the pollen moving in the air or jellyfish floating in the water.

Pollinators are an example of how the movement of organisms can have a cascading effect on the environment. For instance, the movement of pollinators between plants not only allows for reproduction but can also influence the distribution and abundance of plant species, which in turn impacts other organisms that rely on these plants for food or habitat.

1.1 Pollinators

Bees and other insects play a crucial role in providing pollination services that are essential for both crops and wild plants. In fact, up to two-thirds of all crops and one-third of global food crops depend on insect pollination. Wild bees and other pollinators account for approximately 50% of this pollination, while managed bees, such as honey bees, contribute the other half (Klein *et al.*, 2007). Hence, the decline of both managed and wild pollinators across the globe is a significant concern. Significant scientific efforts are currently underway to comprehend the decline of pollinators (Potts *et al.*, 2010), with the aim of identifying effective strategies to halt and potentially reverse it (Brown & Paxton, 2009; Winfree, 2010). Human land-use is commonly considered as a contributor to pollination decline through the loss of habitat for pollinators or habitat fragmentation. Many agricultural management entities aim to counteract the negative effects of land-use by creating wild habitats within agricultural landscapes and introducing wild pollinators. Several models of foraging bees have been proposed to help this management strategies by providing information about the potential visiting rate to flowers (Becher *et al.*, 2016; MacQueen *et al.*, 2022). However, none of these models have considered search behavior. They either assume that insects already know the locations of all available feeding sites in their environment or that they discover them based on fixed probabilistic laws (e.g., the probability of finding a flower at a given location is proportional to $1/L^2$, where L represents the distance to that flower). However, a growing body of experimental evidence suggests that this is not the case. Bees, like many pollinators, are central place foragers, which means that every foraging trip commences and ends at the nest site. Consequently,

their range of action is limited.

Central place foraging refers to a strategy used by certain animal species to establish a central location, usually a nest or burrow, from which they venture out to search for food and return to deposit it. This strategy is often observed in social animals such as ants, bees, and certain bird species that gather food for their offspring or the colony. It is a complex behavior that involves a series of decisions by the foraging animals, including where to establish the central location, how to navigate to and from it, and where to search for food.

The aim of this thesis is to propose a model of bee movement that considers the fact that they are central place foragers and use this model to make predictions about the probability of discovering flowers in different scenarios.

1.2 Movement Ecology

Movement ecology is a field of study that focuses on understanding the movement of organisms, from individuals to populations and communities, and its implications for their survival, behavior, and ecological interactions. It is a multidisciplinary area that draws on concepts and tools from various fields, such as ecology, physics, mathematics, and computer science, among others (Nathan *et al.*, 2008; Joo *et al.*, 2022).

Studying movement ecology is crucial for several reasons. First, movement is a fundamental aspect of life for most organisms, as it enables them to find resources, escape from predators, explore new habitats, and mate. Understanding the drivers and patterns of movement is therefore essential for predicting the distribution, abundance, and persistence of species in a changing world. Second, movement ecology provides insights into the mechanisms that shape ecological interactions, such as competition, predation, and disease transmission, and their consequences for ecosystem functioning and services. For instance, the movement of pollinators is critical for maintaining plant diversity and productivity, while the dispersal of seeds and nutrients influences the structure and function of plant communities and ecosystems. Finally, movement ecology has numerous applications in conservation biology and management (Allen & Singh, 2016), as it can inform the design of effective protected areas, the management of invasive species and diseases, and the restoration of degraded habitats. It can also help predict the impacts of global change, such as climate change, land-use change, and urbanization, on the movement and distribution of species and the services they provide.

Studying movement ecology is essential for understanding the fundamental

processes that drive ecological interactions, predicting the responses of species and ecosystems to environmental change, and informing conservation and management strategies.

1.2.1 What do I mean by ”modeling”?

One of the aims of this thesis is to use a modeling approach to answer theoretical ecology questions. However, this approach can be highly criticized due to a misunderstanding of the role of models in ecology and the idea that they should be "replicas" of reality. Here, I will explain my modeling approach and how it should be understood in the rest of this manuscript.

Over the past few decades, the advancement in computational power and the increased accessibility of computer languages have led to a significant rise in the use of models in scientific research. Models have numerous applications, from explaining intricate data (like statistical models) to serving as a hypothesis (natural selection evolution). We often think of models as mathematical representations of systems relying on our understanding of the said system. Nonetheless, I first want to note that I like to consider a broader definition of "model". Experimentalists often employ the term "model organism" when referring to *Drosophila melanogaster* or *Mus musculus*. Without being mathematical equations or verbal constructs, these are still "models" (Zuk & Travisano, 2018). They are not used to describe what is happening "in real life" but rather setting the limits of what can happen and what cannot. "Any formal device that facilitates 'what if' reasoning can be called a model", as written by Seger & Adler (2002). My model will describe the movement behavior of a pollinator model, the bumblebee *Bombus terrestris*. I will create a mathematical model that accurately describes their behavior, which will be formalized through a set of equations in continuous time. These equations will serve as the foundation for implementing the model numerically and performing simulations using a discretized time step. Through these simulations, I will conduct ”what if” reasoning to determine the probability of resource discovery by the bumblebees. This approach will allow me to test different scenarios and their potential outcomes

Several models can be proposed to describe or predict certain situations and, in that process, models can evolve or be discarded. Similar to empirical knowledge, theoretical knowledge is also prone to errors and is not infallible. Some may consider that theory should not be considered a part of ecology until "validated". However, no aspect of knowledge (empirical or theoretical) can be fully validated.

Experimental results can also be contradicted by later experiments. Therefore, "to exclude fallible theories from ecology while admitting fallible experiments is absurd" (Caswell, 1988)

Models omit certain details because they are not designed to be exact replicas of reality. Very often in biology, the choice of a model is based on its adjustment to experimental data. Here, the criteria I will use are based on simplicity and parsimony. It is not assumed that by not including something on the Model (e.g. the vision) I'm stating that it's unimportant. The factors or parameters included in a model are not necessarily the only important ones, similar to factors manipulated in an experiment. Both models and experiments are part of broader contexts, and it is unrealistic to expect a single study to fully resolve an issue. "The failure to include a factor in a theoretical model is a legitimate criticism only if it can be argued that its inclusion would materially affect the answer to the question under consideration." (Caswell, 1988)

1.3 Movement models

Researchers in behavioral ecology and physics have extensively examined animal movement from a theoretical perspective (Codling *et al.*, 2008). In this section, I will discuss some of the most prevalent movement models found in the literature.

In 2D movement, steps controlled by two parameters are often used to approximate the movement. These parameters are the length traveled during a step, denoted as $l(t)$, and the direction of movement denoted as $\theta(t)$.

1.3.1 Simple Random Walks

The simplest case of random walks assumes that individuals have no memory or fatigue and move in a homogeneous environment. In this case, the distribution of distances traveled before a change in direction follows an exponential distribution. New directions are derived from the uniform law, and there is no correlation between successive directions. This kind of random walk is also known as Brownian motion at a macroscopic scale. This type of random walk is, for example, used to describe bacteria movements (e.g Peterson & Noble (1972)) (Fig 1.1).

However, in many cases, trajectories are more complicated, and there is a need to change the distributions of l or θ. One such example is Lévy Walks.

1.3.2 Lévy Walks

Lévy Walk [1] models have been developed particularly for optimal resource exploration. In this type of random walk, the distances before changing direction follow a heavy-tailed (or sub-exponential) distribution, resulting in a series of small steps occasionally interspersed with long ones (Fig 1.1). Although there is much debate (Benhamou, 2007; Pyke, 2015), Lévy Walks are claimed to be more efficient during resource exploration and seem to account for the search strategy of several animals such as wandering albatrosses (Viswanathan *et al.*, 1996), jackals (Atkinson *et al.*, 2002), reindeer (Mårell *et al.*, 2002), dinoflagellates (Bartumeus *et al.*, 2003), and spider monkeys (Ramos-Fernández *et al.*, 2004). However, it is also argued that Lévy Walk patterns can emerge when combining two Random Walk distributions: one for interpatch movement and another for intrapatch movement, and can thus arise from the way an animal interacts with its environment (Benhamou, 2007). Moreover, Lévy Walks seem ill-suited to model the movement of central place foragers since they produce trajectories that go too far from the starting point.

1.3.3 Correlated Random Walks

Another modification that can be done of a Simple Random Walk, is the modification of the distributions from where the new headings are drawn. In a Correlated Random Walk, the new headings represent a small variation of the previous one. The turning angle are drawn from a wrapped Normal Distribution or a Von Mises distribution. The persistence in direction depends on how much the distribution is concentrated around 0 (Fig 1.1).

This type of model is very often used for describing the movement of several animals like ants, fish, mammals, etc (Kareiva & Shigesada, 1983; Bergman *et al.*, 2000; Kadota *et al.*, 2011; Khuong *et al.*, 2013).

1.4 Why do we need a model for bee movement?

While different types of random walks have been used to describe several types of animal movements, no satisfying models have been proposed so far to describe bee's exploration patterns under the constraint that they are central place for-

[1] We can often find both terms "Lévy Walk" and "Lévy Flight" used interchangeably in the literature. However, in a Lévy Walk, the time taken to complete a given step depends on the length of that step. In a Lévy Flight, the walker jumps between successive locations instantaneously.

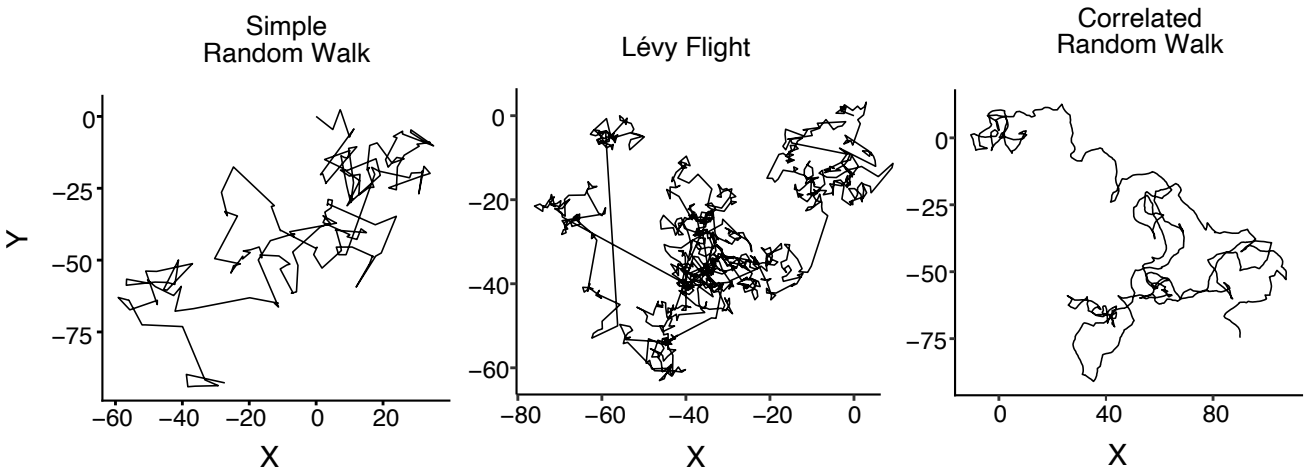

Figure 1.1: **Examples of different types of movement models**

agers. While many models of bee foraging behavior have been developed to predict population dynamics, pollination services, or route learning, none of these models incorporate the central place foraging component of their behavior (Reynolds *et al.*, 2013; Becher *et al.*, 2016; Dubois *et al.*, 2021). The omission of this critical aspect of bee behavior can lead to unrealistic predictions, highlighting the need for a more realistic model that complements existing foraging models. By developing a more comprehensive model that considers central place foraging, we can gain a better understanding of how bees explore and navigate in their environments, which has important implications for ecosystem health and the sustainability of agriculture.

Two main problems arise when we try to describe bee movement using typical random walks. First, when we observe an individual flying in real life, they do not make sudden changes in direction, making models that discretize direction inappropriate. Secondly, as Central Place Foragers, their movements cannot be described by diffusive models as their foraging trips always start and end at their nest.

Two different models have been recently proposed by Lenz *et al.* (2013) and Noetel *et al.* (2018).

Lenz *et al.* (2013) proposed a model for bee movement in their study, which is based on two differential equations. The first equation describes the speed of an individual bee, while the second equation describes the turning angle. They do not consider the fact the bees are Central Place Foragers and the model is diffusive in the long-time limit (Fig 1.2).

In Noetel *et al.* (2018), the authors propose a movement model of Central Place Foraging. Drawing inspiration from celestial mechanics, the authors propose

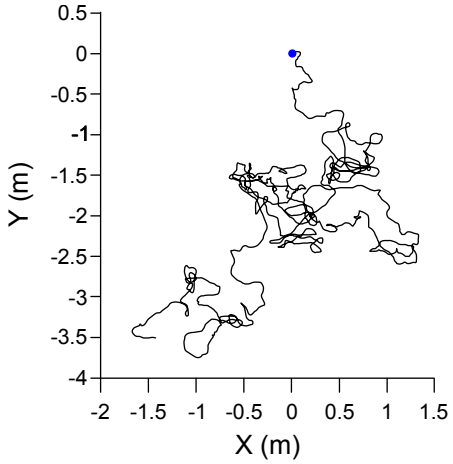

Figure 1.2: Example of a trajectory produced by the model of Lenz

a unique approach in which the movement of the particle resembles that of a planet in orbit around a central body under the influence of an attracting potential (analogous to gravity), with added noise (Fig 1.3)

In this manuscript, I present a novel model that offers a higher degree of flexibility in generating trajectory shapes than the previous model proposed by Noetel *et al.* (2018), because it is designed from a behavioral perspective rather than from a "mechanical force" perspective.

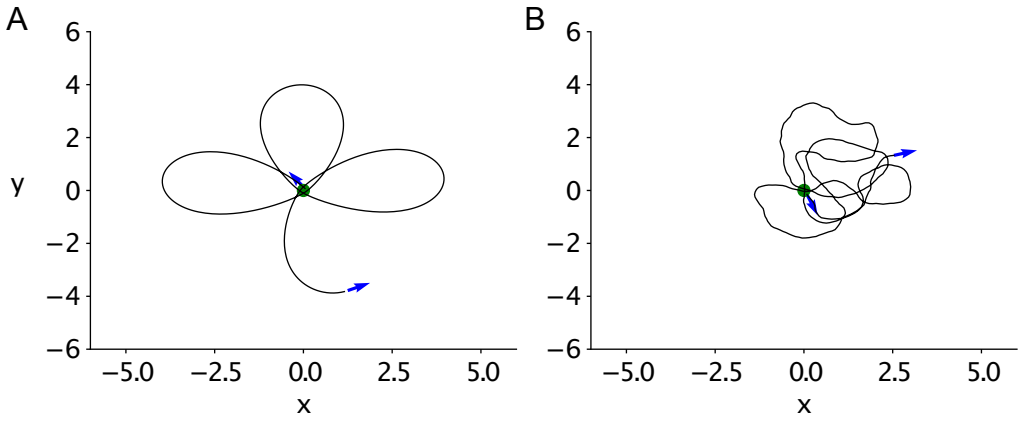

Figure 1.3: **Example of trajectories from Noetel *et al.* (2018)** A. Trajectory without noise. B. Trajectory with noise

1.4.1 The Persistent Turning Walker

I considered the Persistent Turning Walker (PTW) model which is an extension of the Correlated Random Walker (CRW). While in a CRW a trajectory is considered as a succession of straight moves separated by instantaneous jumps in the orientation domain, in PTW a trajectory is considered as a succession of moves with

constant curvatures, separated by instantaneous jumps in the curvature domain. Such a description allows for a continuously defined velocity with time-correlation of the curvature. A stochastic version has been used to model fish motion behavior (Gautrais *et al.*, 2009). The PTW model describes motion at constant speed, where the heading is driven by an Ornstein-Uhlenbeck process (see Chapter 1) acting on the turning speed.

1.5 Outline of the thesis

The objective of this thesis is to propose a new model that captures the exploration behavior of Central Place Foraging bees. In Chapter 2, I will describe the theoretical approach used to construct the model and outline its various characteristics. In Chapter 3, I will demonstrate how to link the model with experimental data by calibrating it to generate realistic trajectories. In Chapter 4, I will use the model to predict resource discovery in both simple and ecologically relevant scenarios. Finally, I will discuss the potential applications of the model beyond resource exploration and its broader implications for understanding the behavior of bees as well as their impact on the environment.

Chapter 2

The Model

Contents

The objective of this chapter is to introduce a model capable of replicating realistic bee trajectories. The inspiration for this model comes from the Persistent Turning Walker model, which was originally developed to describe the movement of fish in a tank. By adapting this model to account for the unique behaviors and characteristics of bees, I expect to create a simulation that accurately reflects their movements.

2.1 The Persistent Turning Walker

Numerous theoretical and experimental studies have focused on the displacement of fish, particularly on the dynamics of schools of fish. Many authors have tried to understand their collective movement through a theoretical approach by proposing interactions between individuals that results on their synchronization (Couzin *et al.*, 2005, 2002; Gautrais *et al.*, 2008; Viscido *et al.*, 2007). The interactions between neighbors are often summarized by a set of rules that modify the movement of each individual fish. In some studies, the null-model used to describe these movements is a Correlated Random Walk. Here, the new heading of an individual is a small variation of the previous one. The estimation of the interaction parameters between individuals strongly depends on the chosen movement model.

An important step before analyzing these interactions is therefore to check whether the correlated random walk as a null-model holds. Otherwise, it is necessary to develop a better-grounded model. In Gautrais *et al.* (2009), the authors performed experiments where they quantified the trajectories of fish to test if a correlated random walk could effectively describe their movements.

In their experiment, they filmed isolated individuals in a tank and tracked their position every 1/12 s during 2 minutes (Fig 2.1).

Then, they estimated the swimming speed and the angular speed of each individual. The angular speed W_i was estimated by fitting a circle to the three consecutive points (x_{i-1}, x_i and x_{i+1}) as shown in figure 2.2.

They found that the swimming speed was constant for each fish (but different among individuals) and that the angular speed had an auto-correlation time of several seconds. Since a Correlated Random Walk model only accounts for the autocorrelation of headings, and not angular speed, it is not an appropriate model to describe the movements of fish.

They then defined a model that described the changes of the angular speed of the fish through time to account for its auto-correlation.

Next, I'll present in detail how the model presented in Gautrais *et al.* (2009)

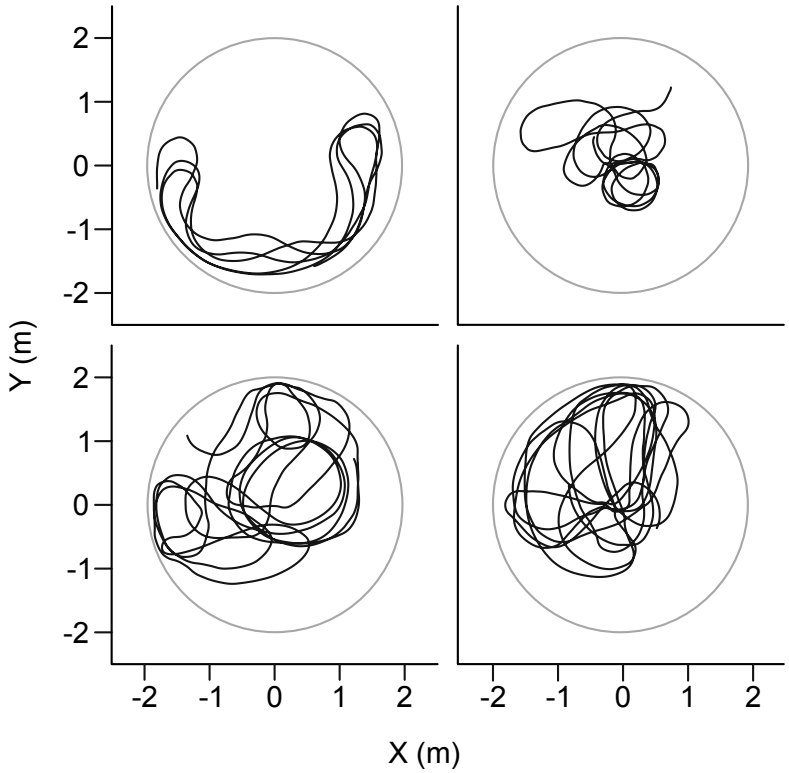

Figure 2.1: **Examples of fish trajectories obtained by Gautrais *et al.* (2009)** The individuals where isolated in a tank and tracked their position every 1/12 s during 2 minutes. The outer circle represents the tank.

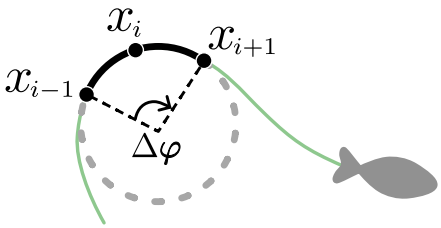

Figure 2.2: **Estimation of the angular speed.** The angular speed W is estimated from the experimental data using $W_i = \frac{\Delta\varphi}{2\Delta t}$. x_i represents the position of the fish at the time step i.

is built.

2.1.1 Building the model

2.1.1.1 State variables

The position $\vec{x}$ of the individual is defined by

$$\vec{x}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} \tag{2.1}$$

and its velocity $\vec{v}$ by

$$\vec{v}(t) = v(t) \begin{pmatrix} \cos\theta(t) \\ \sin\theta(t) \end{pmatrix} \tag{2.2}$$

where $v(t)$ is the speed of the individual and $\theta(t)$ is the heading. From now on, it is assumed that the speed $v(t)$ of the individual is constant and is noted v (Fig 2.3).

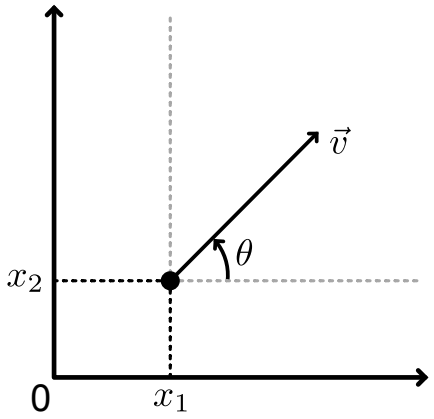

Figure 2.3: **State variables of the model.** x represents the position, $\vec{v}$ the velocity and θ the direction

The aim is now to model the changes of positions through the changes in angular speed. For this, the authors use a stochastic process called an *Ornstein-Uhlenbeck* (OU) process (Uhlenbeck & Ornstein, 1930).

In the next section, I will present the OU process and its properties.

2.1.2 The Ornstein-Uhlenbech process

In ecology, we often define a stochastic process as any process describing the evolution in time of a random phenomenon. It is a variable ω that changes over time in an "uncertain" way. They became a very common tool used for describing from the dynamics of a population to the movement of individuals.

2.1.2.1 Definition of the OU process

A variable $\omega(t)$ is said to follow an Ornstein-Uhlenbech stochastic process if it satisfies:

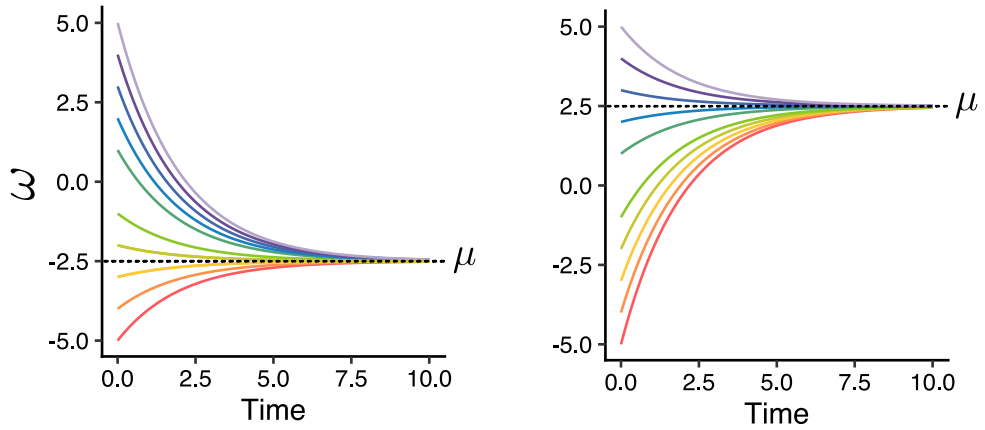

Figure 2.4: $\omega(t)$**.** Each color represents a different values of $\omega(0)$. Left: Long-expected value $\mu = -2.5$. Right: $\mu = 2.5$

$$\mathrm{d}\omega(t) = \gamma[\mu - \omega(t)]\mathrm{d}t + \sigma\mathrm{d}W(t) \tag{2.3}$$

where $\mathrm{d}\omega(t)$ [1] is the increment during $\mathrm{d}t$ [2]; μ is the long-term expected value of ω (it is assumed to be constant), $\gamma > 0$ is the speed of the reversion of $\omega(t)$ towards the expected value μ (it is also assumed constant and I also call it "autocorrelation parameter"), $\mathrm{d}W(t)$ is an increment during $\mathrm{d}t$ of a Wiener process, which is scaled by $\sigma > 0$.

2.1.2.2 Building some intuition: The deterministic part of the OU process

To solve equation 2.3, we can start by building some intuition by first analyzing the deterministic part of the OU process.

When we remove the stochastic component $\sigma\mathrm{d}W(t)$ of equation 2.3, $\omega(t)$ follows a deterministic process :

$$\mathrm{d}\omega(t) = \gamma[\mu - \omega(t)]\mathrm{d}t \tag{2.4}$$

$\omega(t)$ will drift to a equilibrium towards the long-term expected value μ. This is called a *mean reversion process*. This is shown in figure 2.4. γ determines the speed of this process.

To understand why this happens, I'll start by solving equation 2.4.

[1] Note that the variation of ω over time is usually noted $\frac{\mathrm{d}\omega}{\mathrm{d}t}$ but since the Wiener process is not differentiable, I'll keep the notation $\mathrm{d}\omega(t)$ for the rest of this manuscript.

[2] It is important to remember that in a continuous-time series, $\mathrm{d}t$ represents an infinitesimal increment of time.

Solution

$$\mathrm{d}\omega(t) = \gamma[\mu - \omega(t)]\mathrm{d}t$$

Separation of variables:

$$\frac{\mathrm{d}\omega(t)}{\omega(t) - \mu} = -\gamma \mathrm{d}t \tag{2.5}$$

Integration:

$$\int \frac{\mathrm{d}\omega(t)}{\omega(t) - \mu} = \int -\gamma \mathrm{d}t \tag{2.6}$$

$\mathrm{d}\omega(t)$ represents a variation of ω. Since μ is constant, we can state:

$$\mathrm{d}\omega(t) = \mathrm{d}(\omega(t) - \mu) \tag{2.7}$$

and plug it back into equation 2.6, we get:

$$\int \frac{\mathrm{d}(\omega(t) - \mu)}{\omega(t) - \mu} = \int -\gamma \mathrm{d}t \tag{2.8}$$

We define

$$Y = \omega(t) - \mu \tag{2.9}$$

So 2.6 becomes:

$$\int \frac{\mathrm{d}Y}{Y} = \int -\gamma \mathrm{d}t \tag{2.10}$$

Solving the two integrals, from 0 to t, we get:

$$\int_0^t \frac{\mathrm{d}Y}{Y} = \Big[\log(Y)\Big]_0^t = -\gamma.t - \gamma.0 \tag{2.11}$$

$$\log(Y(t)) - \log(Y(0)) = -\gamma t \tag{2.12}$$

so that:

$$Y(t) = Y(0)e^{-\gamma t} \tag{2.13}$$

and writing back to the original variable $\omega(t)$, we finally get:

$$\omega(t) - \mu = [\omega(0) - \mu]e^{-\gamma t} \tag{2.14}$$

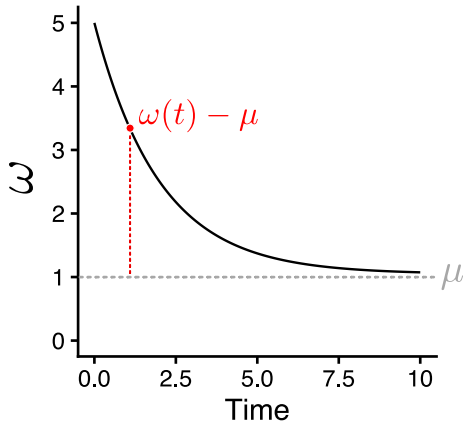

Figure 2.5: **Representation of equation 2.14.** $\omega(t) - \mu$ represents the distance between $\omega(t)$ and the expected value μ. This distance decreases exponentially with time and the speed of the decreasing depends on γ.

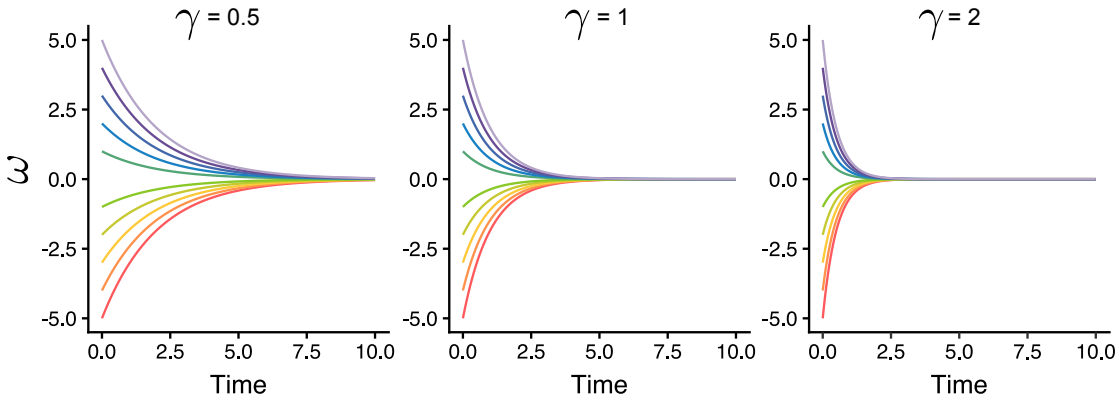

Figure 2.6: $\omega(t)$ **for different values of** γ. Each color represents a different value of $\omega(0)$. Each panel represents a different speed of reversion γ towards the expected value μ.

The left-hand side value of equation 2.14 (Fig 2.5) represents the distance between ω and the expected value μ. This difference decreases exponentially with time and the speed of relaxation to μ depends on γ (Fig 2.6).

Equation 2.14 can also be written as:

$$\omega(t) = \omega(0)e^{-\gamma t} + \mu(1 - e^{-\gamma t}) \tag{2.15}$$

where, in this form, we can read directly that the initial condition $\omega(0)$ is forgotten in exponential time, and that $\omega(t)$ tends to μ at infinite time, where $e^{-\gamma t} \to 0$. Equation 2.15 is the solution of equation 2.4.

2.1.2.3 What happens if we model the angular speed as an OU deterministic process?

Now we can define the angular speed as ω. We can model the changes of the angular speed ω of an individual using equation 2.15.

We start by fixing $\omega(0) \neq 0$ and $\mu = 0$. The angular speed will approach μ with time. This will produce trajectories that have a strong curvature at the beginning and will become more straight with time.

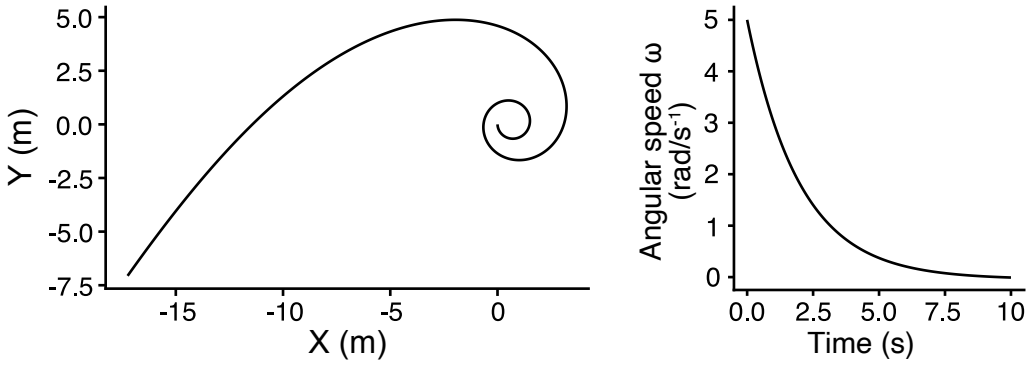

Figure 2.7: **Example of a trajectory simulated using equation 2.15 with $\mu = 0$ and $\gamma = 2$**

2.1.2.4 Adding noise

We can now continue analyzing the equation of the OU process in equation 2.3 by adding back the stochastic component $\sigma \mathrm{d}W(t)$ to the deterministic process in equation 2.4 :

$$\mathrm{d}\omega(t) = \gamma[\mu - \omega(t)]\mathrm{d}t + \sigma \mathrm{d}W(t) \tag{2.16}$$

It is not possible to solve this equation using the same method used to solve equation 2.4 due to the presence of the Wiener process $W(t)$.

To solve equation 2.3, we can apply another procedure called Itô's lemma and define the function $Y(t) = \omega(t)e^{\gamma t}$, so that :

$$\mathrm{d}Y(t) = \mathrm{d}[\omega(t)e^{\gamma t}] = \gamma\mu e^{\gamma t}\mathrm{d}t + \sigma e^{\gamma t}\mathrm{d}W(t) \tag{2.17}$$

We can then integrate equation 2.17 from 0 to t (with $t > 0$), yielding:

$$\omega(t) = \omega(0)e^{-\gamma t} + \mu[1 - e^{-\gamma t}] + \int_0^t [\sigma e^{-\gamma(t-t')}]\mathrm{d}W(t') \tag{2.18}$$

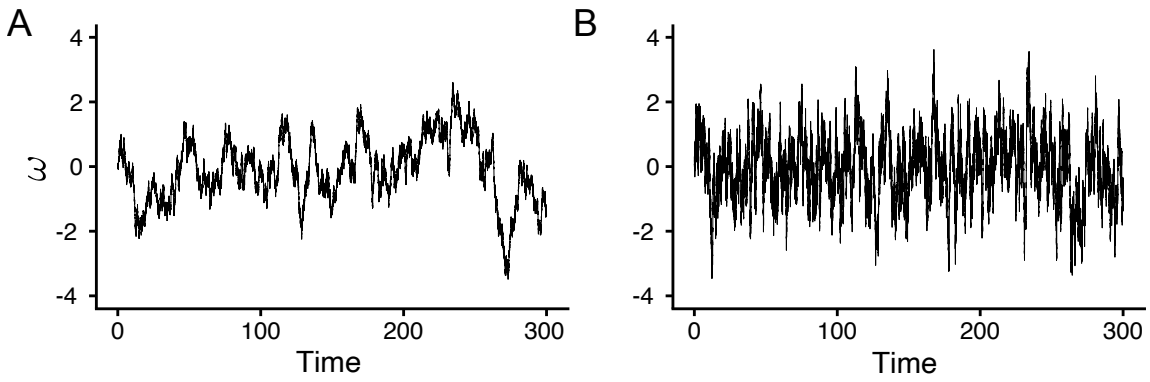

Figure 2.8: **Examples of two time series with the same variance ($\Omega = 1$) but different auto-correlation time.** (A) $\gamma = 0.1$, $\sigma = 0.44$. (B) $\gamma = 1$, $\sigma = 1.41$

From equation 2.18, it is possible to calculate the expected value $\mathbb{E}[\omega(t)]$ and the variance $\mathrm{Var}[\omega(t)]$ of the stochastic process at any time t (Uhlenbeck & Ornstein, 1930):

$$\mathbb{E}[\omega(t)] = \omega(0)e^{-\gamma t} + \mu(1 - e^{-\gamma t}) \tag{2.19}$$

$$\mathrm{Var}[\omega(t)] = \frac{\sigma^2}{2\gamma}(1 - e^{-2\gamma t}) \tag{2.20}$$

and we can derive the long-term expected value $\mathbb{E}^*$ and variance Var^* by taking the limit when $t \to \infty$:

$$\mathbb{E}^* = \mu \tag{2.21}$$

$$\mathrm{Var}^* = \frac{\sigma^2}{2\gamma} \equiv \Omega \tag{2.22}$$

The variance depends both on γ and σ. It is therefore possible to have two distributions of $\omega(t)$ with the same variance but with very different auto-correlation times γ (Fig 2.8)

2.2 The Central Place Forager Model

2.2.1 Enunciation of the model

The Persistent Turning Walker (PTW) model produces trajectories that are diffusive in the large time limit (Degond & Motsch, 2008; Cattiaux *et al.*, 2010; Weber *et al.*, 2011; Daltorio *et al.*, 2013; Nötel *et al.*, 2017; Du *et al.*, 2021). However, the

aim of this thesis is to produce a model of bees as Central Place Foragers. They explore their environment by performing loops around the nest. Their behavior is thus not purely diffusive.

To take this into account, I added an attraction component to equation 2.16 that will restrain the movement around the nest and will control the return of the bee.

I assumed that bees can locate the direction of the nest at any time and defined a "Homing vector" $\vec{H}(t)$ that points towards the position of the nest (Collett *et al.*, 2013). I call $\varphi(t)$ the angle between the velocity $\vec{v}(t)$ and the homing vector $\vec{H}$ (Fig. 2.11). In order to go towards the nest, the bee has to adopt an angular speed that makes $\varphi(t)$ decrease so $\vec{v}$ gets closer to $\vec{H}$. I define this targeted angular speed ω^*.

$$\omega^*(t) \equiv \eta(t)\varphi(t) \tag{2.23}$$

where $\eta(t) \in \mathbb{R}^+$ represents the strength of the attraction to the nest. I integrate this attraction component to the model in equation 2.16 and the model becomes

$$\mathrm{d}\omega(t) = -\gamma[\omega(t) - \omega^*(t)] + \sigma \mathrm{d}W(t). \tag{2.24}$$

Now, in the absence of noise, the angular speed will tend to ω^* following:

$$\omega(t) = \omega(0)e^{-\gamma t} + \omega^*(t)(1 - e^{-\gamma t}) \tag{2.25}$$

The higher the strength of attraction $\eta(t)$, the fastest the heading of the bee will change to go towards the nest.

Here, I considered two different options to define the attraction component η. It could be simply defined as a constant $\eta = \eta^*$. In this case, in the absence of noise, the trajectories would resemble to satellites as shown in figure 2.9.

This does not resemble experimental trajectories that are presented in Chapter 2. This is why I chose a second option. To be able to model bees looping around the nest, I divided the movement into two phases: an exploration phase and a return phase. In Chapter 3, I will expose the arguments that led me to choose this second option.

For this, η becomes a function of time and it works like a switch going from $\eta = 0$ (no attraction) to $\eta = \eta^*$ (attraction). I assume that the bee changes its behavior instantaneously between the two phases with

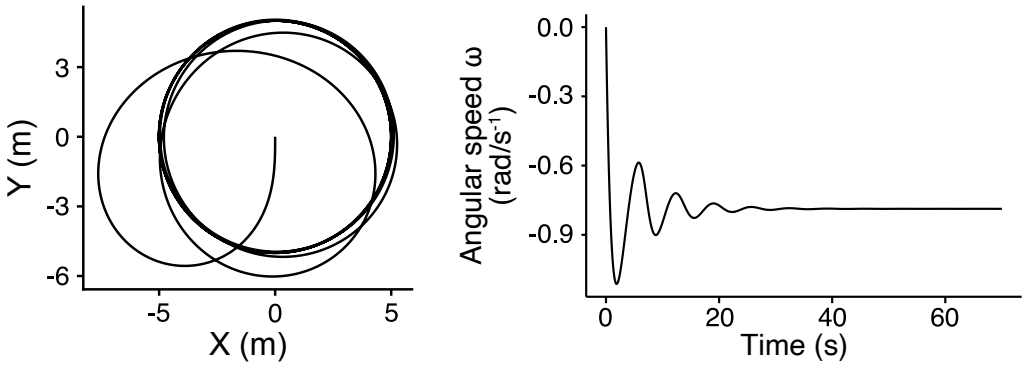

Figure 2.9: **Example of a trajectory simulated using equations 2.23 and 2.24 with** $\eta(t) = 0.5$**,** $\gamma = 1$ $\sigma = 0$

$$\eta(t) = \begin{cases} 0 & t < \tau \\ \eta^* & t \geq \tau \end{cases} \tag{2.26}$$

where τ is the time at which the switch happens.

For this delay τ, I assumed the simplest random distribution: a memoryless process which translates into an exponential distribution. I denote α the parameter associated with this distribution, i.e the mean time before the switch to the return phase. Once the bee is back at the nest, the movement switches back to the exploration phase. This model is able to produce loops centered around the nest (see figure 2.10).

If we now reintroduce the noise, the trajectories become less stereotypical. In the exploration phase, $\eta(t) = 0$ so $\omega^* = 0$, and in consequence the movement is diffusive. In the return phase, $\eta(t) = \eta^*$ so $\omega^*(t)$ continuously adapt to $\varphi(t)$, and the movement becomes advecto-diffusive (as illustrated in figure 2.11).

In summary, the trajectories produced by the model are conditioned by 4 components

- γ which governs the autocorrelation of the angular speed
- σ which governs the white noise
- The component attraction ruled by :
  - η which governs how quickly the individual has to go back to the nest
  - α which governs the average switching time between exploration and return phases

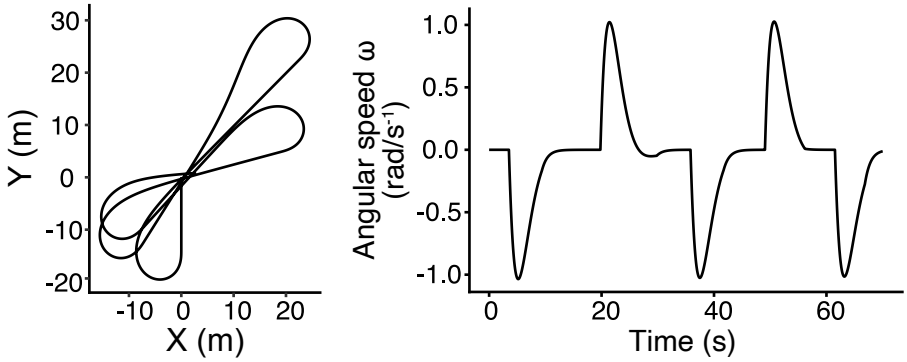

Figure 2.10: **Example of a trajectory simulated using equations 2.23, 2.24 and 2.26 with $\eta^*(t) = 0.5$, $\gamma = 1$ and $\sigma = 0$.** The bee departs from the nest in exploration mode and with $\omega(0) = 0$, so in absence of noise, it follows a ballistic trajectory away from the nest. After the switch, $\omega(t)$ adapts to $\varphi(t)$ so the bee draws a u-turn up to a heading pointing to the nest. Then the bees keep going straight to the nest, where a new loop can start in exploration mode. Five loops are represented (left) as well as the corresponding evolution of $\omega(t)$ (right).

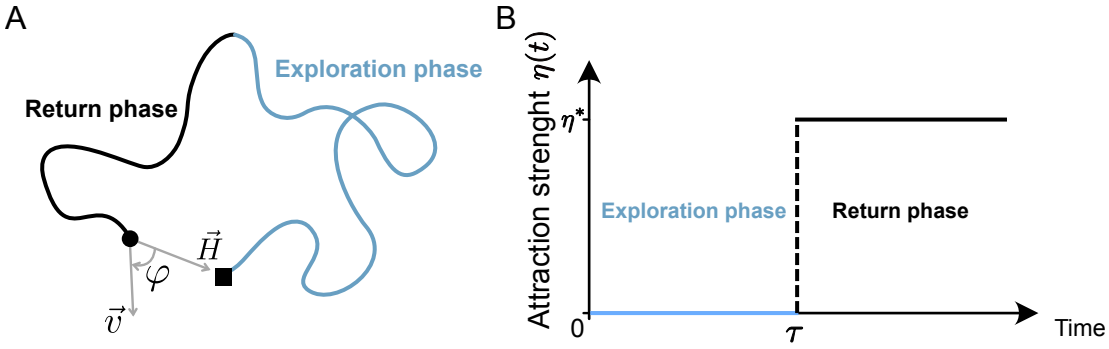

Figure 2.11: **Illustration of the model.** (A) Example of theoretical trajectory. Blue line: Trajectory during the exploration phase. Black line: Trajectory during the return phase. Black circle: bee. Black square: nest. $\vec{H}$ is the homing vector pointing towards the nest. $\vec{v}$ is the velocity of the bee. φ is the angle between $\vec{v}$ and $\vec{H}$. (B) Evolution of the return strength (η) over time. At time τ, η switches from 0 (no attraction) to $\eta*$.

Note that γ and σ both control the variance $\Omega = \frac{\sigma^2}{2\gamma}$ (Eq. 2.22). Since Ω is a parameter that's easier to visualize in a time series than σ, I will refer to γ, Ω, α and η^* as the parameters of the model.

2.2.2 Discretization

In order to be able to simulate trajectories, I discretized the model using an explicit forward discretization scheme over a time step Δt.

2.2.2.1 Discretization of the Ornstein-Uhlenbeck process

In a continuous-time series, dt represents an infinitesimal interval of time. To produce a sample of the stochastic process given by 2.18 between $(t - \Delta t)$ and t, it is necessary to replace the integral term by a sampling of the noise over Δt. Equation 2.18 then yields:

$$\omega(t) = \omega(t - \Delta t)e^{-\gamma \Delta t} + \omega^*(t - \Delta t)(1 - e^{-\gamma \Delta t}) + \sqrt{\frac{\sigma^2}{2\gamma}(1 - e^{-2\gamma \Delta t})}\varepsilon_t \qquad (2.27)$$

where ε_t is sampled from a Gaussian random variable of mean 0 and variance 1.

Equation 2.27 can also be written as:

$$\omega_{i\Delta t} = \omega_{(i-1)\Delta t}e^{-\gamma \Delta t} + \omega^*_{(i-1)\Delta t}(1 - e^{-\gamma \Delta t}) + \varepsilon_i \qquad (2.28)$$

where ε_i is now sampled from a Gaussian variable E_i of mean 0 and variance s^2 defined as:

$$s^2 \equiv \mathrm{Var}(\mathrm{E}_i\ ; \Delta t) = \frac{\sigma^2}{2\gamma}(1 - e^{-2\gamma \Delta t}) \qquad (2.29)$$

The approximation introduced in this scheme is only due to the fact that $\omega^*(t)$ is considered as constant between $(i - 1)\Delta t$ and $i\Delta t$ (in the diffusive case when $\omega^*(t) = 0$, the scheme is then exact).

In equation 2.28, $\omega_{(i-1)\Delta t}$ is a predictor of $\omega_{i\Delta t}$ and only this previous value is used to predict ω at $i\Delta t$. In statistics of discrete-time random processes, equation 2.28 represents an auto-regressive process of order 1 (AR(1)).

We can conclude that the Auto-regressive process of order 1 is the discrete-time approximation of the Ornstein-Uhlenbeck process (should $\omega^*(t)$ be constant over time, that would be the discrete-time sampling equivalent).

This AR(1) (eq.2.28) can be written as:

$$\omega_i = g\omega_{i-1} + b_{i-1} + \varepsilon_i \tag{2.30}$$

where

$$g = e^{-\gamma \Delta t} \tag{2.31}$$

and

$$b_{i-1} = \omega^*_{(i-1)\Delta t}(1 - e^{-\gamma \Delta t}) \tag{2.32}$$

In the case of the diffusive process ($b_{i-1} = 0, \forall i$), we can then expect from equation 2.31 to estimate γ from an estimate of g with:

$$\gamma = -\frac{\log(g)}{\Delta t} \tag{2.33}$$

and from equation 2.29, to estimate σ^2 from an estimate of s^2:

$$\sigma^2 = s^2 \frac{2\gamma}{1 - e^{-2\gamma \Delta t}} \tag{2.34}$$

2.2.2.2 Discretization of the trajectories

The code used for the discretization is available in Appendix B.

For each loop, I draw a time τ from an exponential distribution with parameter α. When $t = \tau$, I switch $\eta(t) = \eta^*$. At the end of a loop, I reset $\eta(t) = 0$.
To start, I set the parameters of the model :

- the speed: v
- the autocorrelation parameter: γ
- the variance of the Wiener process: Ω
- the parameter governing the attraction component: α and η^*

Then, I initialize the value of the variables at $t = 0$

- angular speed $\omega(0)$
- direction $\theta(0)$
- position $\vec{x}(0)$
- the force of the attraction to the nest $\eta(0)$

I calculate the position $\vec{x}(t+\Delta t)$ using an Euler scheme, considering $\vec{v}(t)$ constant over Δt.

$$\vec{x}(t+\Delta t) = \vec{x}(t) + \vec{v}(t)\Delta t \tag{2.35}$$

Then I calculate the direction $\theta(t+\Delta t)$ using (Fig. 2.12):

$$\theta(t+\Delta t) = \theta(t) + \omega(t)\Delta t \tag{2.36}$$

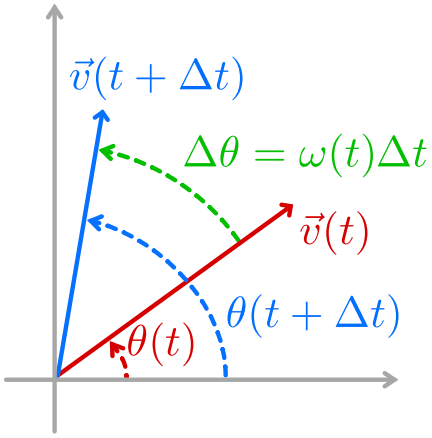

Figure 2.12: **Discretization of the direction θ as in equation 2.36**

The velocity $\vec{v}(t+\Delta t)$ is given by

$$\vec{v}(t+\Delta t) = v \cdot \begin{pmatrix} \cos\big(\theta(t+\Delta t)\big) \\ \sin\big(\theta(t+\Delta t)\big) \end{pmatrix} \tag{2.37}$$

Lastly, I calculate the angular speed $\omega(t+\Delta t)$.
First, I calculate the targeted angular speed ω^* with eq. 2.38, where φ is a signed angle between $\vec{x}$ and the homing vector $\vec{H}$

$$\omega^*(t) = \eta \cdot \varphi(t) \tag{2.38}$$

Then, I compute the discrete-time equivalent of γ and σ:

$$g = e^{-\gamma\Delta t} \tag{2.39}$$

$$s^2 = \sigma^2 \frac{1 - e^{-2\gamma\Delta t}}{2\gamma} \tag{2.40}$$

Finally, I calculate the angular speed $\omega(t + \Delta t)$ using the equation 2.41 with ε representing a realisation of a Gaussian random variable with mean 0 and variance s^2.

$$\omega(t + \Delta t) = g\omega(t) + (1 - g)\omega^*(t) + \varepsilon \tag{2.41}$$

2.3 Properties

2.3.1 Attraction to the nest limits the exploration range of bees

An unrealistic feature of existing diffusive models is their long-term behavior: If given enough time, the forager reaches extremely far distances with respect to the nest, never returning to it. To illustrate the impact of central place foraging on the simulation of bee exploration range, I compared this model with attraction to the nest to an alternative one in which the attraction is absent (i.e., making $\eta^* = 0$ in eq. 2.23). I simulated 1000 trajectories with each model for different amounts of time and studied how the distribution of bees around the nest changes over time. As expected, attraction retains bees tightly localized around the nest (Fig. 2.13A-B, blue). More interesting, it makes the distribution of bees stationary: In a model without attraction, bees constantly wander away from the nest, and their distribution depends on how much time we allow for the bee to explore, becoming wider as time goes by (Fig. 2.13B-C, orange). In contrast, the attraction component makes the forager return to the nest periodically, so the distribution remains stationary once the forager has had enough time to perform more than one loop on average (Fig. 2.13D, blue).

This demonstrates the importance of taking into account the attraction component when modeling bee movements (instead of considering a diffusive model) to be able to reproduce realistic trajectories. To highlight the effect of behavioral parameters upon the spatial extent covered by exploration, I present in Fig. 2.14 the marginal effect of the three main parameters (γ, $\eta*$ and Ω) on the range explored by the bee, as measured by the long-time Mean Square Displacement (MSD), i.e. the variance of the stationary distribution.

2.3.2 Effect of the parameters on the Mean Square Displacement

The effect of α , the mean duration in exploration mode, appears quite linear, which is not surprising since the time spent flying away from the nest increases with α During the return phase, η^* represents the intensity of the steering, since for smaller values of η^*, the relaxation to the preferred turning speed would be less effective. As η^* gets smaller and smaller, the steering vanishes, so that the bee would adopt a diffusive behavior, with no stationary distribution. On the other

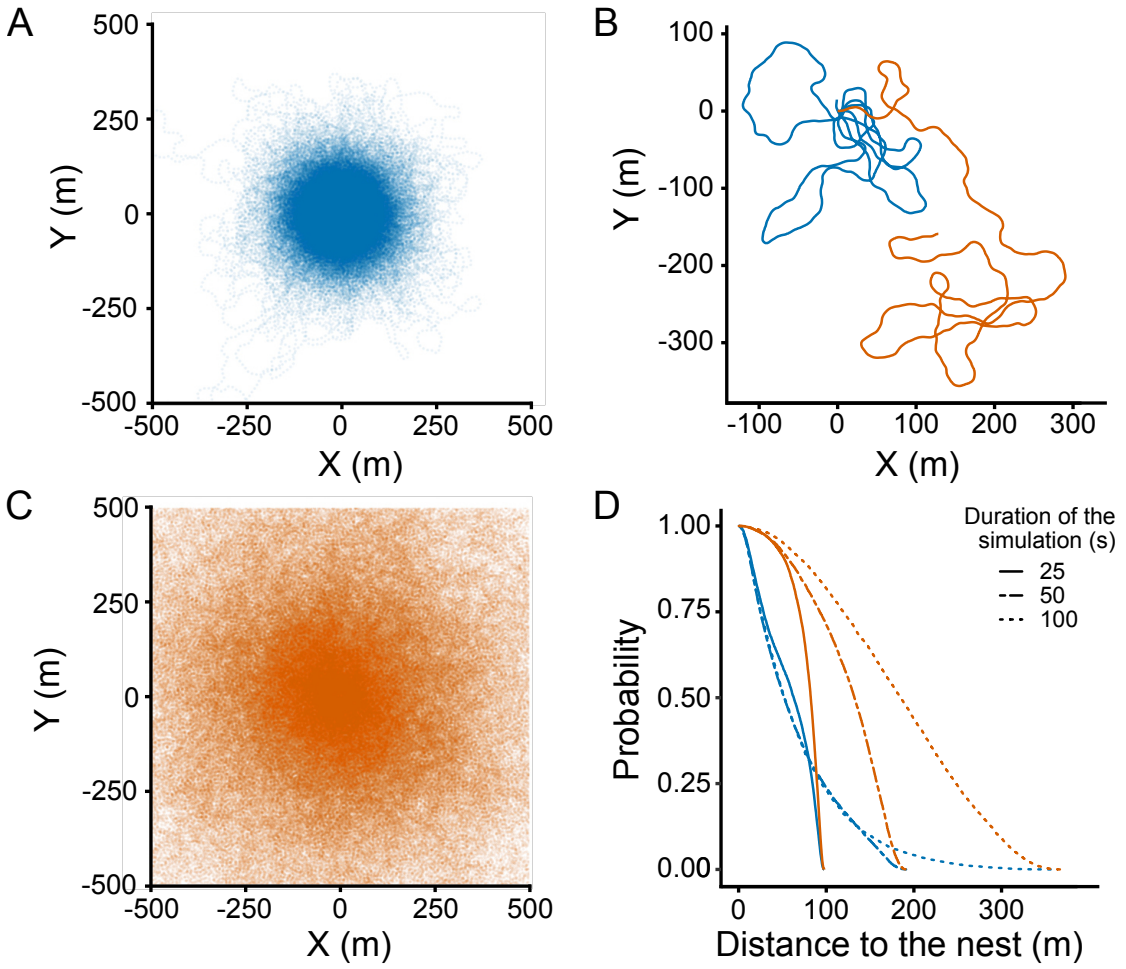

Figure 2.13: **Probability of presence of a bee around the nest.** (A) Overlay of 1000 trajectories with attraction to the nest ($\eta^* = 0.2$ s^{-1}) simulated during 900 s. (B) Example trajectories with and without attraction, simulated during 500 s. The nest is located at (0,0). Blue: model with attraction. Orange: model without attraction (C) Same as A but without attraction to the nest ($\eta = 0$) (D) Probability to find a bee below a given distance to the nest (i.e., inverse cumulative probability distribution for the distance to the nest) after different amounts of time.

hand, we observe a clear effect of saturation for large enough values of η^*, meaning that the effectiveness of the steering is limited by the relaxation time γ. Finally, Ω controls the level of noise the turning speed can undergo. For $\Omega = 0$, the turning speed has no noise at all, and the process becomes deterministic: the bee would fly from the nest ballistically in the exploration phase and go back ballistically to the nest after turning maneuver induced by the steering process (as in figure 2.10). In this case, given the initial condition of null turning speed, the trajectories would push bees the farthest from the nest during exploration (hence, maximal MSD), while larger values of Ω would drive bees to meander around the nest, leading to trajectories that remain closer to the nest.

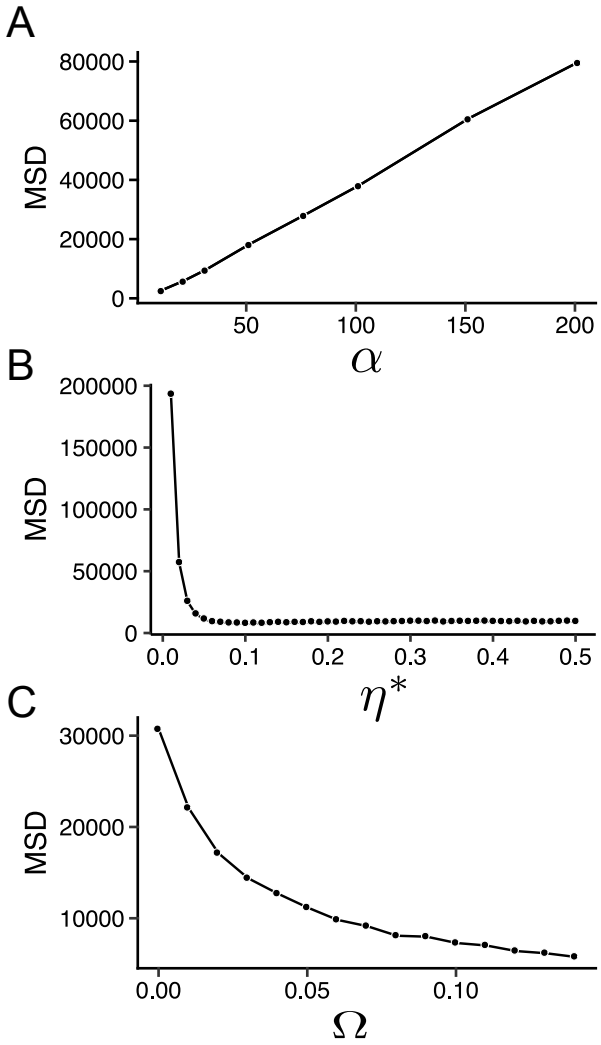

Figure 2.14: **Sensitivity of the Mean Square Displacement (MSD) to the behavioral parameters.** I estimated the MSD (m^2) by the Monte Carlo method, varying each parameter in turn, (A) α (s), (B) $\eta^*(s^{-1})$ or (C) $\Omega = \sigma^2/2\gamma$ ($rad^2.s^{-2}$), while leaving the others to their default values: $\gamma = 1.0\ s^{-1}$, $\sigma = 0.37\ rad.s^{-1/2}$, $\alpha = 30$ s and $\eta^* = 0.2\ s^{-1}$. Each MSD value was estimated using 10^5 simulated loops (error bars are within the symbol size). Note that for smaller values of η^*, the distribution tends to (non stationary) diffusion, so some loops are censored at one hour.

2.3.3 Effect of the parameters on the shape of the trajectories

This model is capable of reproducing a high variety of trajectories. I will present how each parameter impacts the shape of the trajectories.

2.3.3.1 The autocorrelation : γ

In the model in equation 2.24, γ represents the inverse of the auto-correlation time of the angular speed. This means that low values will translate into trajectories with a high auto-correlation of the angular speed. In this case, ω can wander away from 0 for large amounts of time. This will produce very circular trajectories as in figure 2.15.A. This kind of trajectory would be impossible to obtain using a Correlated Random Walk model in which there is no such memory of the changes in direction.

When $\gamma = 2$, ω tends to remain close to 0, leading to minimal changes in direction and resulting in a predominantly straight trajectory. Since there is not a lot of autocorrelation on the direction changes, this kind of trajectory could be produced by a Correlated Random Walk.

2.3.3.2 The Brownian noise

As the value of σ increases, the variance Ω of ω also increases. This relationship is evident in the time series depicted in the right panels of Figure 2.16. When σ is low, the angular speed of the trajectory remains close to 0, resulting in straight trajectories. However, when σ is high, there are frequent deviations from 0, causing the trajectories to become "curly" in shape (see Figure 2.16-C).

Another scenario to consider is when the value of variance Ω is held constant while we vary the values of σ (eq. 2.22). In this case, each variation in σ will also cause the auto-correlation time γ to change.

When we set σ to a low value, a high auto-correlation is necessary to achieve the fixed variance value so that jumps of the angular speed away from 0 last enough time for the variance to increase. For $\sigma = 0.1$ (Fig 2.17A), this traps the trajectory intro very localized spirals that last a long time. To fully visualize the variance in the figure 2.17A, right panel, we would need to wait for a very long time.

If we fix $\sigma = 5$ (Fig 2.17), we have the opposite case. Here, the autocorrelation is weaker and produces more straight trajectories. These types of trajectories are

similar to those produced by a Correlated Random Walk model.

For the intermediate case where $\sigma = 1$, we observe a pattern of localized spirals followed by similar periods where ω remains close to 0 and produces straight jumps. These trajectories can be compared to those generated by Lévy flight models, which also exhibit intermittent periods of localized behavior and sudden long-range jumps.

2.3.3.3 The attraction component: α and η^*

Two parameters control the attraction to the nest: α controlling the time spent on the "exploration phase" and η^*, which governs the "return phase". Specifically, higher values of η^* result in a more direct return path towards the nest, as indicated by the colored lines in figure 2.18 A-B-C.

When $\alpha = 0$, there is no exploration phase, causing the trajectory to continuously orbit around the central point without any departure from the nest. When there is no noise present and $\sigma = 0$, the resulting course will be circular (as in figure 2.9).

The time spent in the exploration phase increases with α, this means that the distance traveled away from the central point increases (see MSD in figure 2.14).

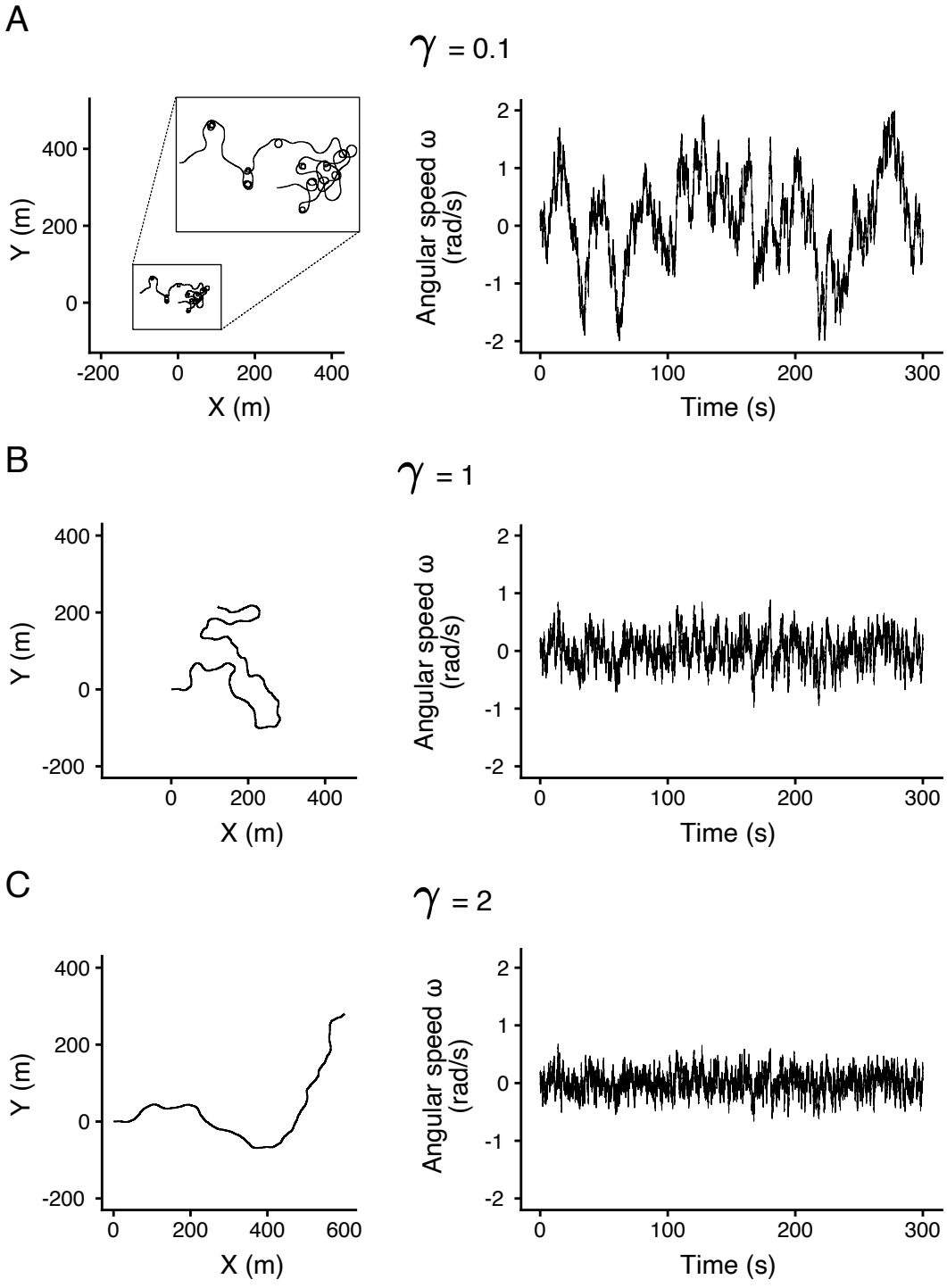

Figure 2.15: **Effect of the auto-correlation parameter γ on the trajectories**. The rest of the parameters are fixed. $\sigma = 0.4$, $\alpha = 0$, $\eta^* = 0$. Note that for trajectories, the spatial unit is the same across figures, for easier visual comparison. If needed, a zoom is given in inset.

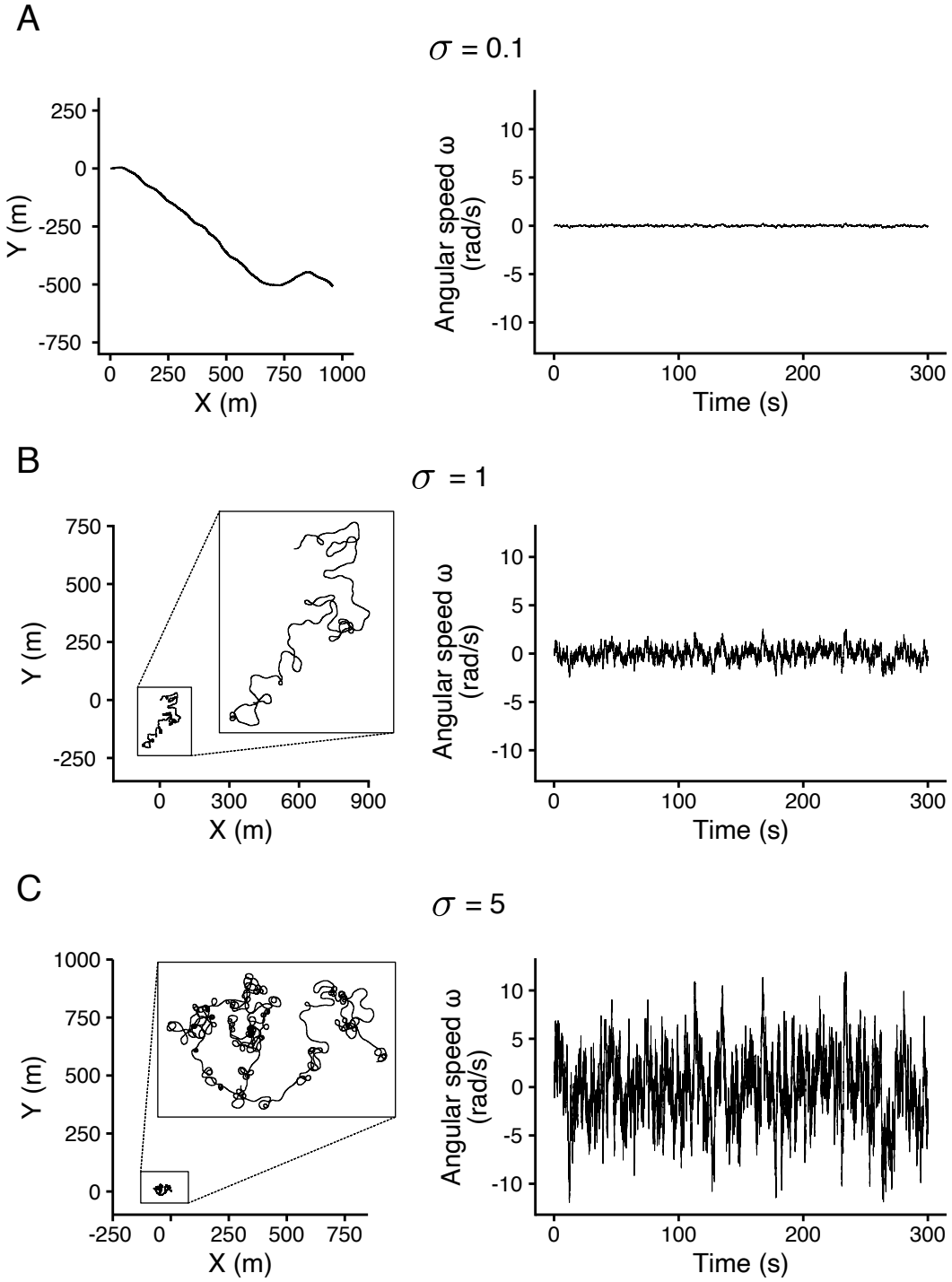

Figure 2.16: **Effect of σ on the trajectories.** The rest of the parameters are fixed $\gamma = 1$, $\alpha = 0$, $\eta^* = 0$. The variance Ω is different for each trajectory. Left: Example of trajectories. Right: Example of time series of the angular speed ω A. $\Omega = 0.005$, B. $\Omega = 0.5$, C. $\Omega = 12.5$

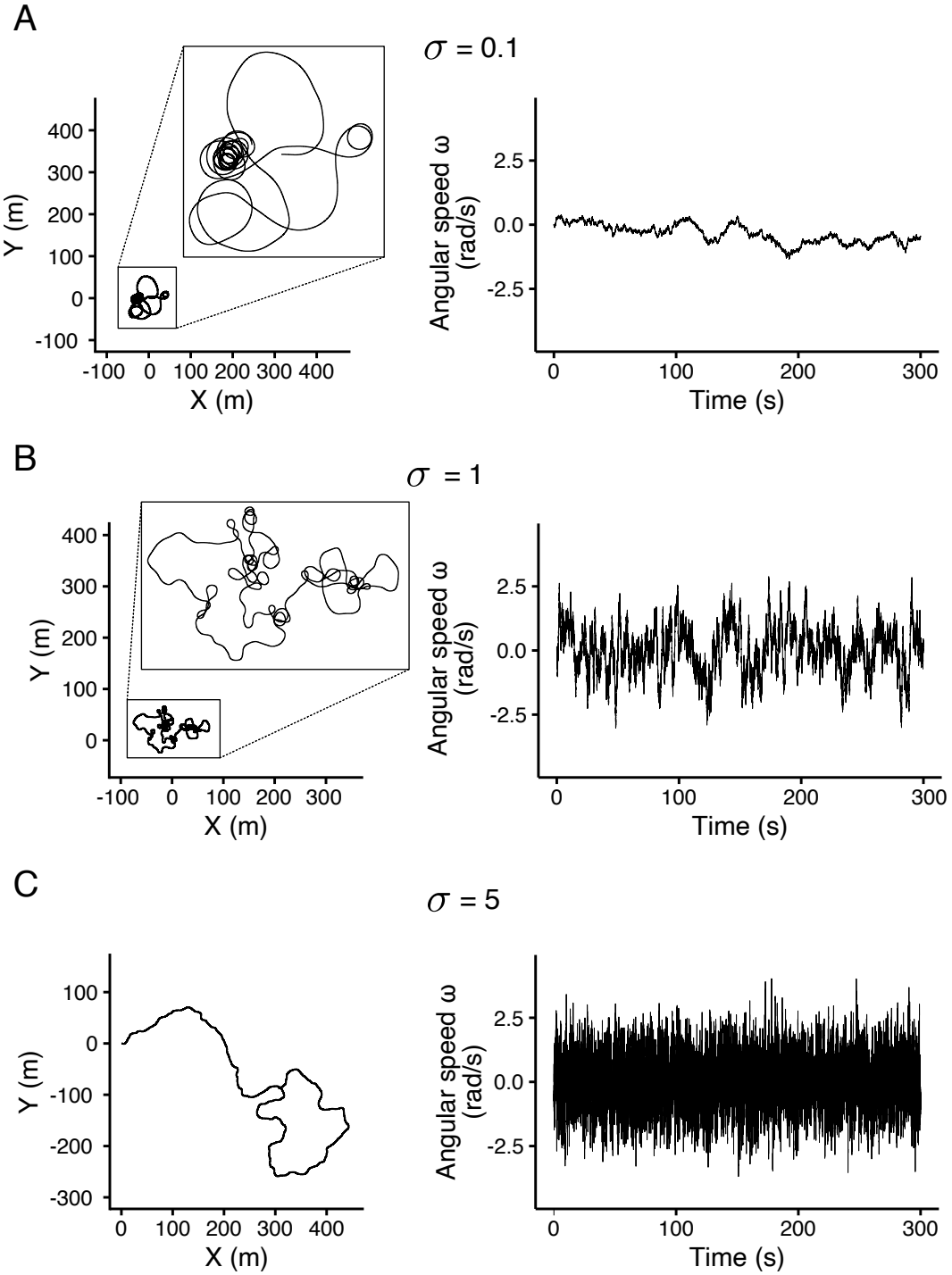

Figure 2.17: **Effect of σ with a fixed variance** $\Omega = 0.07$**.** Left: Example of trajectories. Right: Example of time series of the angular speed ω. $\alpha = 0$, $\eta^* = 0$. The value of γ is different for each trajectory following $\Omega = \sigma^2/2\gamma$. A. $\gamma = 0.005$ B. $\gamma = 0.5$ C. $\gamma = 12.5$

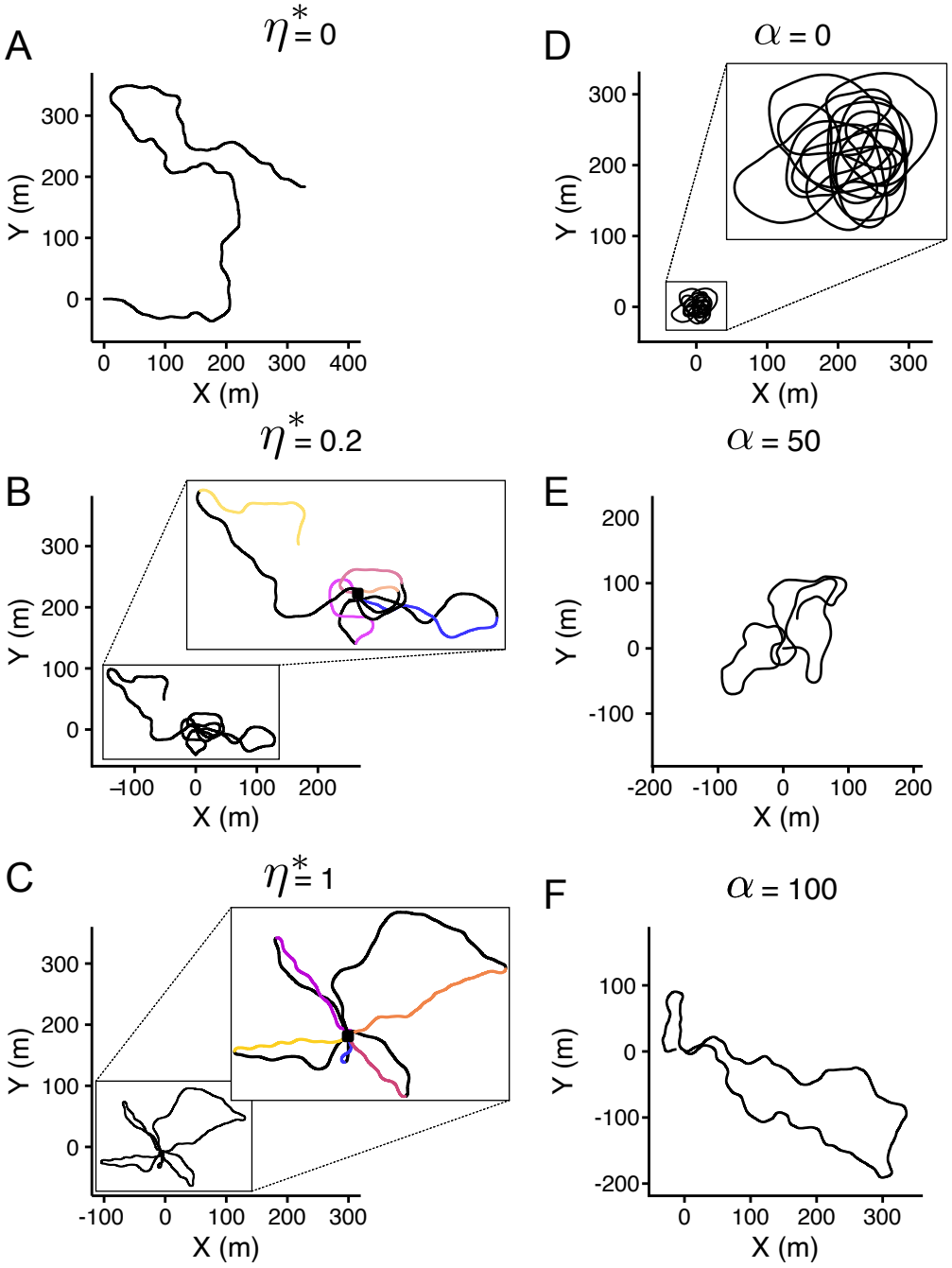

Figure 2.18: **The effect of η^* and α on the shape of the trajectories.** (A-B-C). The colored lines corresponds to the fragments of the loops where $\eta = \eta^*$ (aka the "return phase")

Chapter 3

Calibrating the Model

Contents

Chapter 2 has demonstrated that the Model can generate a wide range of trajectories. The purpose of this chapter is to discuss how to establish a connection between the Model and experimental data to calibrate and produce trajectories that closely resemble those of animal models.

Initially, I calibrated the Model analytically by excluding the attraction component, which is equivalent to the Persistent Turning Walker model presented in Gautrais *et al.* (2009) While the Persistent-Turning Walker (Chapter 2) was first intended to simulate fish movement, I tested if it was possible to calibrate it using data from a different species. Specifically, I sought to investigate whether this approach could enable the capture of movement dynamics that are not replicable using more conventional models (such as a Correlated Random Walk). This was done using high-frequency ant trajectory data. The data set captured the ant's position several times per second. Subsequently, I calibrated the Model using low-frequency bumblebee trajectory data, where the position of the bee was obtained every few seconds. Given the low-frequency data, it was not feasible to use the same methodology as the one employed for the ants.

3.1 Using the Persistent Turning Walker to describe ant movements

I had access to trajectory data from a dessert ant: *Cataglyphis velox*. The data was collected using a track-ball system in a virtual reality environment. The experiments were performed during the internship of Blandine Massot in the team of Antoine Wystrach at the CRCA in 2021.

3.1.1 Animal model

Cataglyphis velox is a Mediterranean ant living in very hot and dry habitats. Worker's body size range from 4.5 to 12 mm. They are thermophilic diurnal scavengers. Their physiology, body and behavior allows them to forage under extreme extremely high temperature conditions. They navigate through desert searching for arthropods corpses. Since they navigate without using pheromones and rely on visual cues, *Cataphyphis sp.* became a model species on insect navigation. This species (like other insects), is known for performing right/left oscillations while walking (Clement et al, in prep).

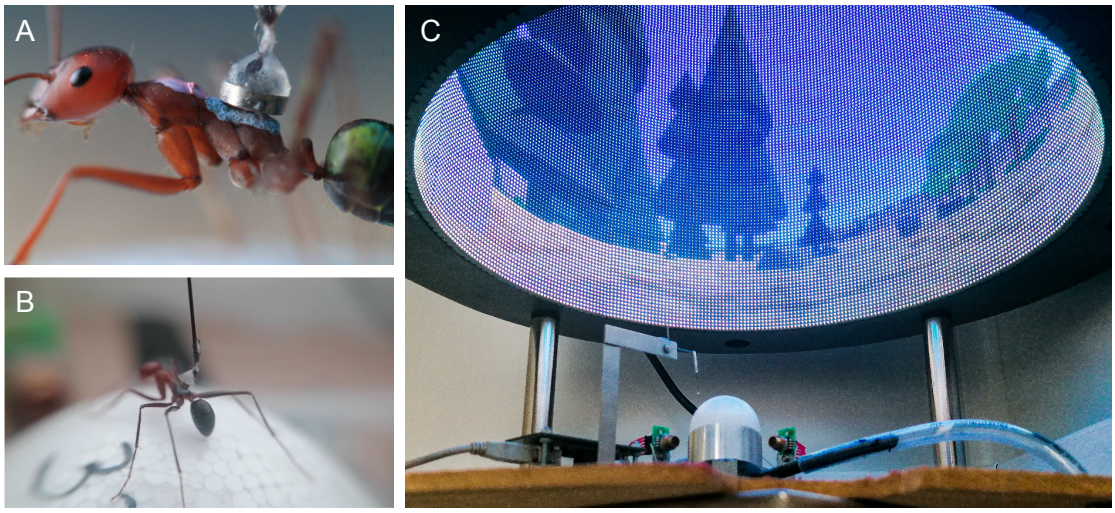

Figure 3.1: **Ant experimental setup**. (A-B). Photo of a Cataglyphis velox, with colored paint and metallic paint, magnetized to the track-ball attachment system. (C). Virtual-Reality set-up. The photos were taken by Blandine Massot.

3.1.2 Experimental setup

Nests from *Cataglyphis velox* were collected in Sevilla, Spain. In the laboratory, they were reared in a ventilated room, at 24-30C, 15-40% humidity and with a natural day/night light cycle. The nest were carved into a cellular concrete block and connected to two plastics bins serving as a hunting area where individuals can drink from a sugar solution.

During the experiments, the ants were captured while they were in the hunting area. They where then placed in an air-suspended polystyrene trackball using micro-magnets and metallic paint applied directly on the ant's thorax. The ball turns from under the ant's legs as it walks and allows it to trace the path taken. The track-ball was then placed into a Virtual-Reality cylinder composed of panels of LEDs. The movements of the track-ball are linked to he virtual environment so that when the ant moves in a direction, the virtual environment reacts. The ant was allowed to move freely in any direction, then, when it reaches a distance of 4m away from the start point, the ant is removed from the setup ant put back into the hunting area.

I had access to 50 trajectories of 50 different ants.

3.1.3 The trajectories

The raw data contains the position of the ant that is being tested with a frequency of ∼100 points per second. This high frequency sampling produces trajectories

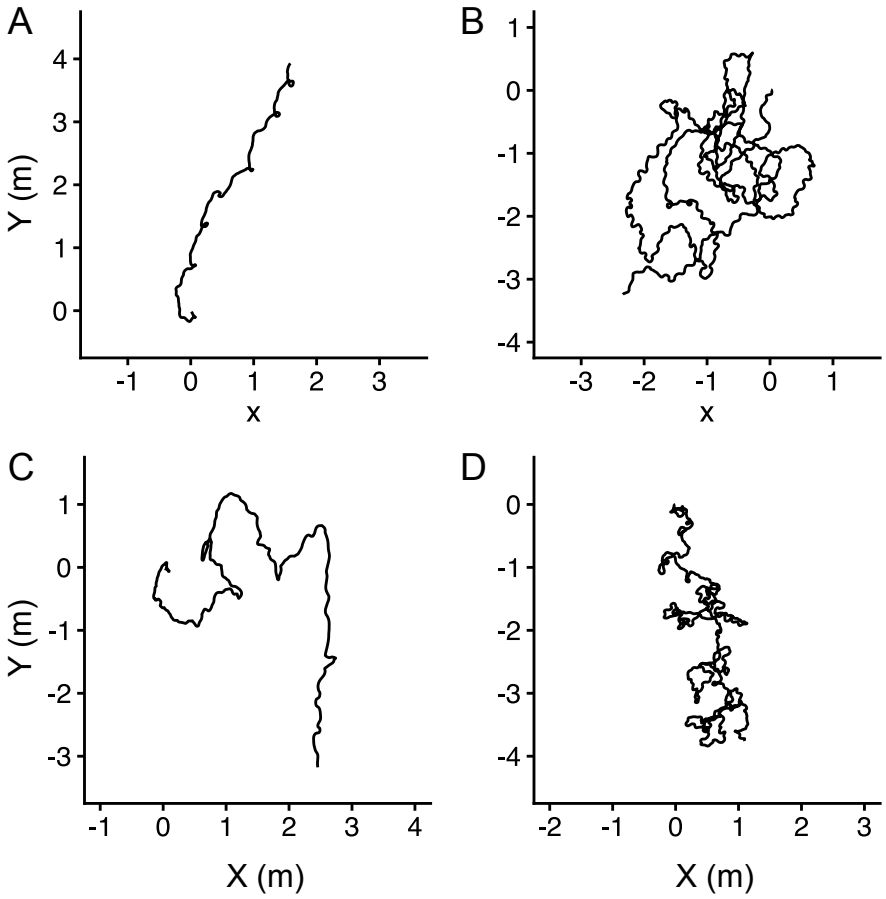

Figure 3.2: **Examples of ant's trajectories obtained thanks to a Virtual Reality setup.** All trajectories starts at [0,0] Scale is the same for the four plots. The ants were placed in a Virtual Reality setting described in 3.1.2

very susceptible to noise. I performed a first smoothing of the trajectories by only keeping the points every 0.04s. Then, I performed a second smoothing by calculating a moving average of the x and y coordinates with a window of 0.2s (Fig 3.3).

The oscillation movements are particularly visible in figure 3.2 B

3.1.4 Quantifying the angular speed

The model of the Persistent Turning Walker describes the changes in angular speed through time. It has 2 parameters : an autocorrelation of the angular speed γ and the variance of the Wiener process (White noise) σ. The aim of the calibration procedure is to estimate the values of those parameters from the observed angular speed of the experimental data. In order to do this calibration, it was necessary to quantify the trajectories of the ants. For this, I calculated the angular speed of

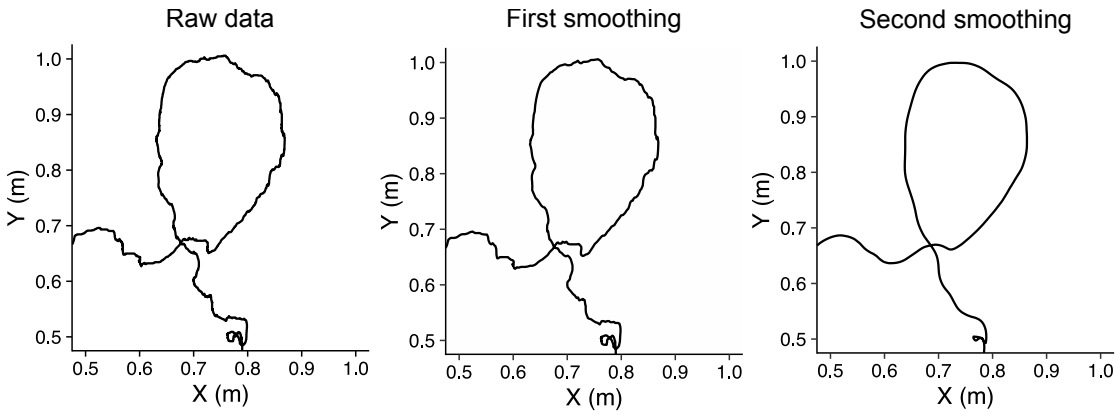

Figure 3.3: **Smoothing of the ant's trajectories.** The raw data contains a point every 0.01s. The first smoothing was done by only keeping the coordinates every 0.04s. The second smoothing was obtained by doing a moving average of the coordinates with a window of 0.2s.

the individuals as a function of time.

First, I calculated the angle $\Delta\theta_i$ between each point p_i of the trajectory

$$\Delta\theta_i = \theta_{i+1} - \theta_i \tag{3.1}$$

Then, the angular speed ω is defined by

$$\omega_i = \frac{\Delta\theta_i}{\Delta t}. \tag{3.2}$$

I obtained a time series of the angular speed of each of the 50 ant's trajectories (Fig 3.4).

Figure 3.5 shows the relationship between the angular speed at time t and the angular speed at time $t + \Delta t$. This correlation suggests that the angular speed is auto-correlated. This finding indicates that a Correlated Random Walk model may not be the best approach for describing these trajectories, as it does not consider the auto-correlation of the angular speed. To account for this correlation, I fitted the Persistent Turning Walker model, which is better suited for modeling auto-correlated angular speed.

3.1.5 Calibration

The model of the Persistent Turning Walker has 2 parameters, the autocorrelation time of the angular speed γ and the variance of the Wiener process σ

In section, chapter 2, I demonstrated that the discrete time equivalent of γ and σ are, respectively,

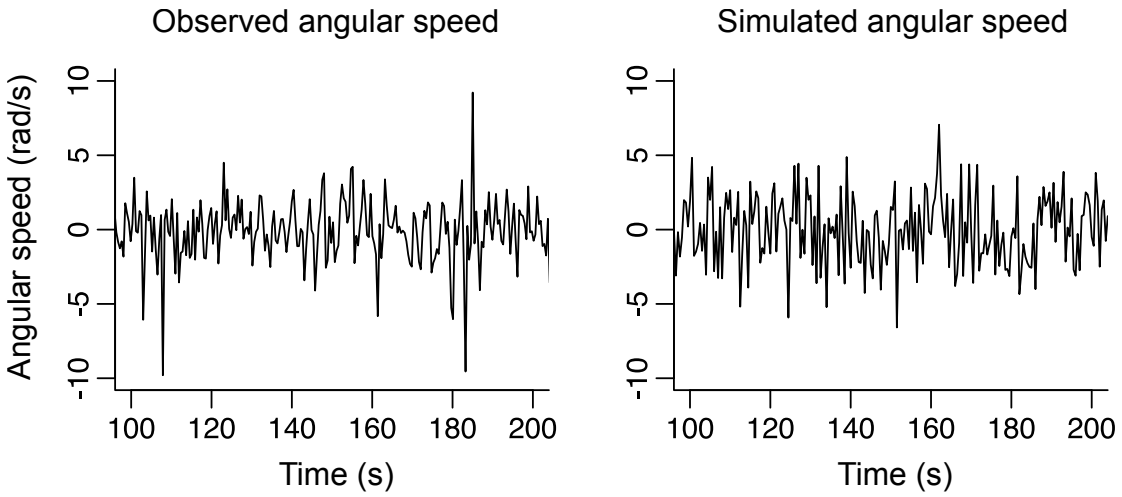

Figure 3.4: **Example of an observed and a simulated time series of the angular speed ω of an ant.** The parameters used for the simulated time series are : $\gamma = 11$, $\sigma = 9$

$$\gamma = -\frac{\log g}{\Delta t} \tag{3.3}$$

and

$$\sigma^2 = s^2 \frac{2\gamma}{1 - e^{-2\gamma\Delta t}} \tag{3.4}$$

The parameters a and g can be estimated from the time series of the angular speed. This can be done by using a regression model and applying a least square regression method using equations 3.5 and 3.6 (Gautrais *et al.*, 2009). Some R functions compute this estimation (`arima` function). The results are presented in figure 3.6.

$$\widehat{g} = \frac{\sum_i^{N-1} W_i W_{i+1}}{\sum_i^{N-1} W_i^2} \tag{3.5}$$

$$\widehat{s^2} = \frac{1}{N} \sum_i^{N-1} (W_{i+1} - \widehat{g} W_i)^2 \tag{3.6}$$

When examining the experimental data, we can observe different types of trajectories. Figure 3.2 illustrates a range of trajectories, including relatively straight paths in panels A and C, trajectories with pronounced oscillations in panel B, and highly coiled ones in panel D.

In figure 3.7 we can observe that the calibration procedure is able to capture most of the different characteristics of the experimental data, excluding the oscil-

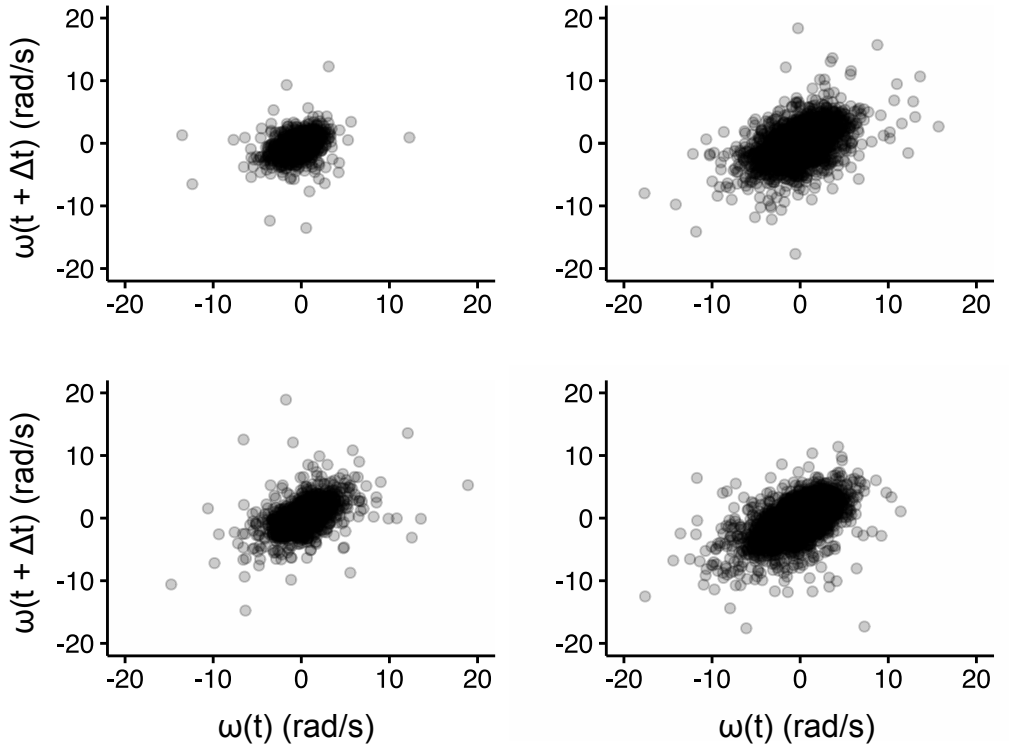

Figure 3.5: **Angular speed at time $t + \Delta t$ in function of the angular speed at time t for the four ants shown in fig 2.2.**

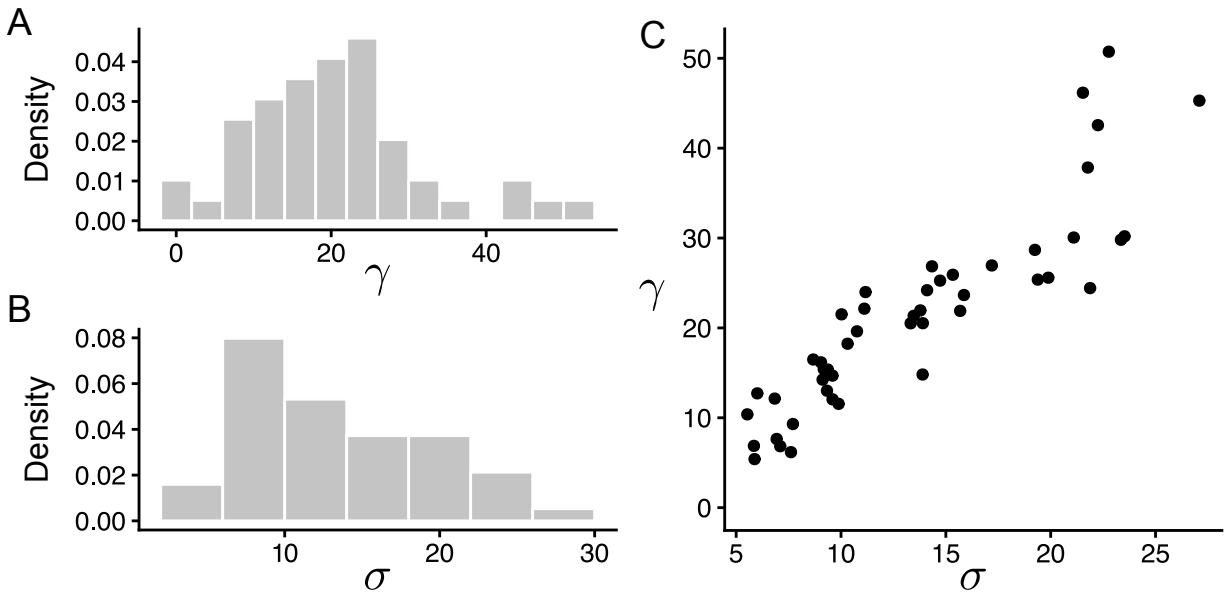

Figure 3.6: **Results of the calibration.** (A-B). Distribution of the estimation of the auto-correlation of the angular speed γ and the variance of the Wiener process σ for all trajectories of ants. (C). γ in function of σ

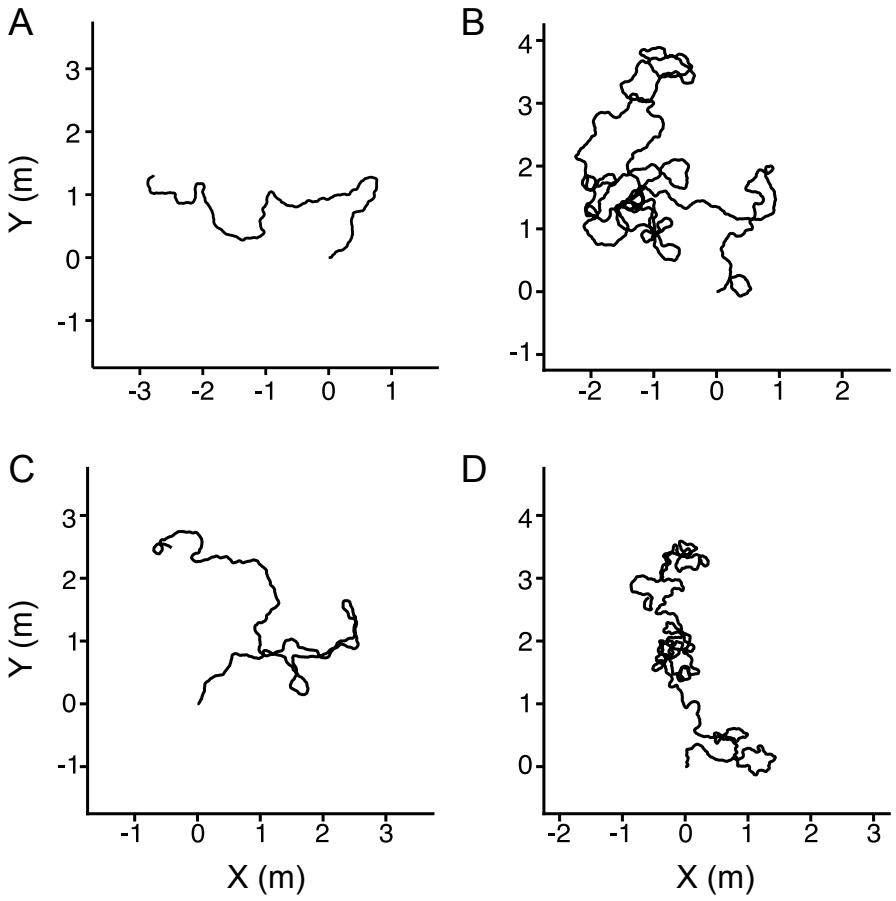

Figure 3.7: Example of simulated trajectories during 300s using pairs of calibrated parameters. All trajectories start at [0,0] A. $\gamma = 24$ $\sigma = 11.2$ B. $\gamma = 15.3$ $\sigma = 9.3$ C. $\gamma = 19.6$ $\sigma = 10.7$ D. $\gamma = 11.5$ $\sigma = 9.9$

latory behavior. It would be necessary to force a periodic value of ω^* to be able to reproduce the oscillations (Clement *et al.*, 2023). This would take into account a more long-term memory that is not taken into account in the Persistent Turning Walker model.

3.2 Calibration with Bumblebee data

I applied the Model outlined in Chapter 2 to simulate the movement of bumblebees. The goal of this thesis was to create a model that could predict the likelihood of bumblebees finding flowers in various environments. To achieve this, I first calibrated the model using experimental data on bumblebee trajectories. This ensured that the model generated realistic simulation results.

3.2.1 Animal model

Bumble bees are eusocial insects that start colonies in the spring with a queen bee who lays eggs and raises the first batch of non-reproductive worker bees. Later in the season, the colony produces males and new queens who mate and hibernate for the winter. Bumble bee nests can range from 20 to 400 workers, and the lifespan of a worker bee varies between species from 13 to 41 days. Bumble bees forage for pollen and nectar, with pollen providing protein for the young and nectar providing energy for adult bees. Bumblebees, like other social insects, are *Central Place Foragers*. This means that the exploration and harvesting of resources (food, building material, water...) is done from a fixed point: the nest. If a resource is rich enough, *Bombus terrestris* can forage as far as 1 to 2 km from the nest (Walther-Hellwig & Frankl, 2000). Unlike the honey bee (Apis mellifera), bumblebees do not dance to pass information about food resources to their conspecifics and recruit them. Individuals, therefore, discover flowers rather independently, although naïve individuals may be attracted by the presence of conspecifics on the flowers.

3.2.2 Experimental setup

The data was collected by Cristian Pasquaretta, Tamara Gómez Moracho, Thibault Dubois and Mathieu Lihoreau before the beginning of my thesis.

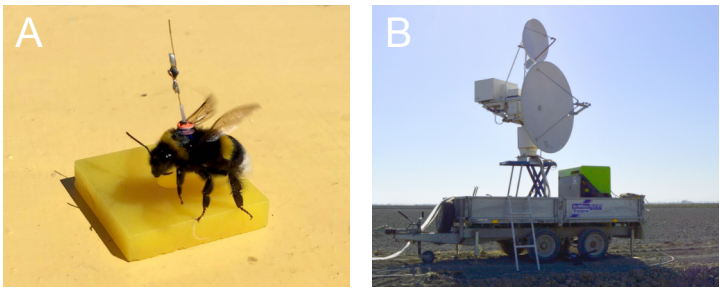

Figure 3.8: **Bumblebees' Experimental setup.** (A) Transponder used to track bumblebees (B) Harmonic radar. Photos provided by Tamara Gómez-Moracho (radar) and Joe Woodgate (bumblebee on feeder).

3.2.2.1 Study Site

The data was collected during the spring of 2018 on a rice farm land in Sevilla (Spain). The study site was located in a commercial area for rice production. The site was wide, flat and had uncultivated soil with no vegetation and few visual repairs.

3.2.2.2 Bumblebees and training

The individuals came from commercially sourced colonies of *Bombus terrestris* (Biobest NV, Westerlo, Belgium). They had never previously had access to the outside world until the training phase. Bees were marked using colour numbered identifier and transferred into wooden boxes (dimensions). Once in the field bees were trained to forage on three artificial flowers (i.e. blue platform with 40% sucrose solution) positioned 2 m in front of the nest box. Once an individual performed several foraging bouts (flights between the nest and the flowers), it was identified as a "regular forager". Then, the entrance of the nest was closed and each flower was randomly moved (a few dozen meters). The bumblbee was equipped with a transponder (16 mm vertical dipole) when leaving the nest and tracked with an harmonic radar until it returned to the nest. The radar was located 350 m away from the nest (Fig 3.8). This data allowed to obtain the 2D coordinates of the bee every 3.3 s. Each bee was allowed to be tested several times. The location of each flower was changed between each trip.

32 trajectories of 18 different bumblebees were obtained (Fig 3.9).

3.2.3 Calibration

The experimental data obtained using this setup cannot be calibrated using the same procedure as the one used for the ants in section 2.1. This is because the

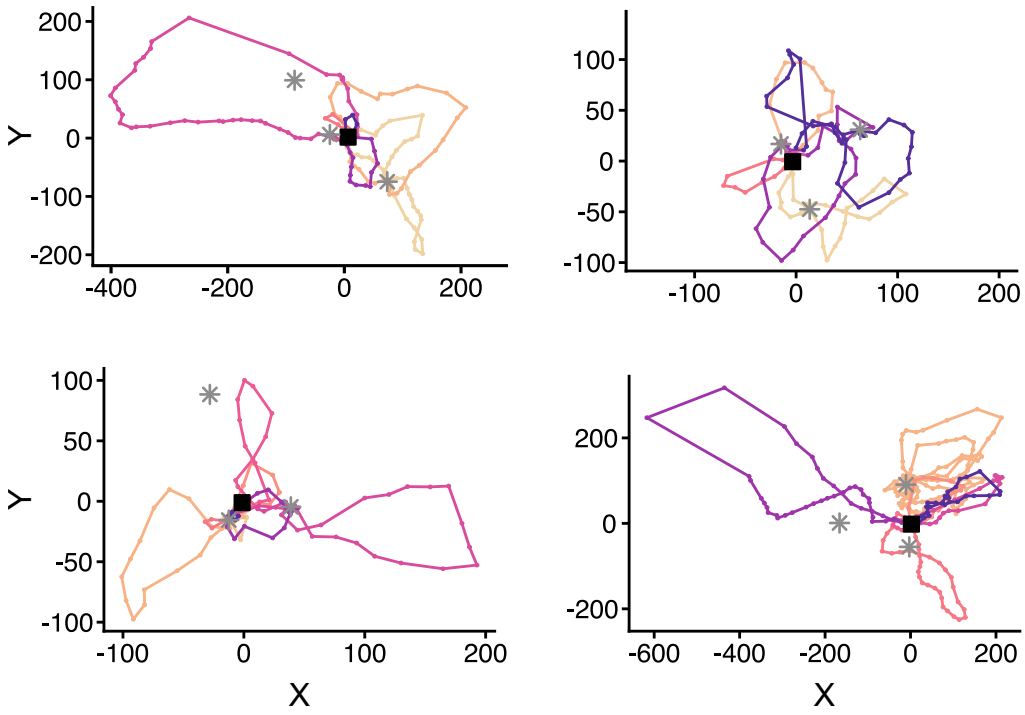

Figure 3.9: **Examples of an experimental trajectories.** Each dot represents the position of a bee recorded by a harmonic radar approximately every 3s. Different colors represent different flight loops around the nest. The sequential order of the loops is represented by the color gradient where the first loops have lightest colors (yellow to purple)

harmonic radar used to collect the data is only able to determine the bee's position every $\Delta t = 3.3$ s, at best. Considering that bumblebees fly at an average speed of around 4 m/s (Fig 3.10), an individual could travel dozens of meters between each sampling.

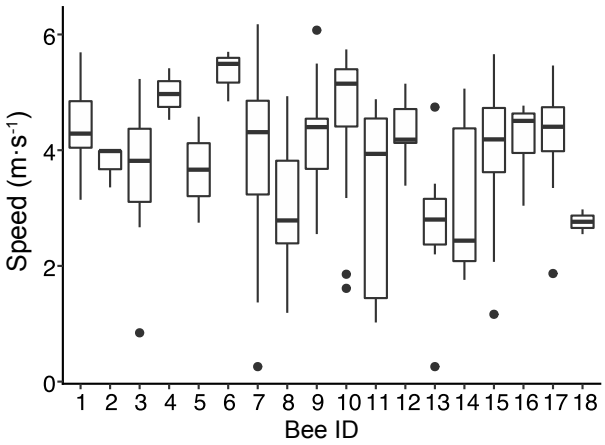

Figure 3.10: **Variability of speed across individuals in the experimental dataset.** Speed (m/s) for each bee. Boxplots, show the median (middle line), 25 and 75% quantiles (box), range of data within 1.5 interquartile deviations (whiskers), and outliers (dots).

Consequently, the trajectories obtained with this experimental setup have a very low frequency sampling, making it impossible to fit the model analytically as in the previous section. This is because the heading of an individual could vary significantly between each sampled point. Therefore, it was necessary to develop a different method to quantify the trajectories.

As Central Place Foragers (CPF), bumblebees perform loops around the nest when exploring and foraging. I divided the trajectories into "loops" and quantified them.

3.2.3.1 Dividing the tracks into loops

I defined a loop as a fragment of a trajectory that starts when the bee leaves that nest and finishes when it enters back. I considered that an individual was "in the nest" when it was able to detect it. The colony nest box used in the experiments was rectangular, with a diagonal of 37 cm. Bumblebees can detect an object when it forms an angle of 3 degrees on the retina of their compound eyes (Kapustjansky *et al.*, 2010). This means that the bees were able to see the nest at approximately 7m. However, in this case I set a higher threshold of 13 m to avoid including learning flights (i.e., flights during which the bee makes characteristic loops to

acquire visual memories of target locations such as the nest for navigation) into the set of exploratory data.

3.2.3.2 Loops statistics

I quantified the trajectories using four different observables (Fig 3.11):

- Loop length: Total length of the trajectory for a given loop
- Loop extension: Maximum distance between the bee and the nest for a given loop
- Number of intersections: Number of times the loop intersects with itself
- Number of re-departures, where a re-departure is defined as three consecutive positions such that the second position is closer to the nest than the first one, but the third is again further away than the second. These events indicate instances in which the bee seemed to be returning towards the nest and turned back

These observables display a large span of values which reflects a high diversity of loops.

3.2.3.3 Fitting

The Model is an extension of the Persistent Turnin Walker used in section 2.1. In addition to the autocorrelation parameter γ and the variance of the Wiener process σ, it has two more parameters controlling the attraction to the nest: η^* governing how quickly the individual has to go back to the nest and α controlling the mean time in exploration mode before switching to "the return to the nest" phase

In order to fit the parameters of the model, I explored systematically all relevant combinations within the relevant range for each parameter. To do this more efficiently, I transformed one of the four parameters of the model into a more tractable one: I substituted the variance of the noise introduced by the Wiener's process (σ) for the variance of the angular speed ω, which has a more direct impact on the experimental data. These two variables are related by

$$\mathrm{Var}(\omega) = \Omega = \frac{\sigma^2}{2\gamma} \tag{3.7}$$

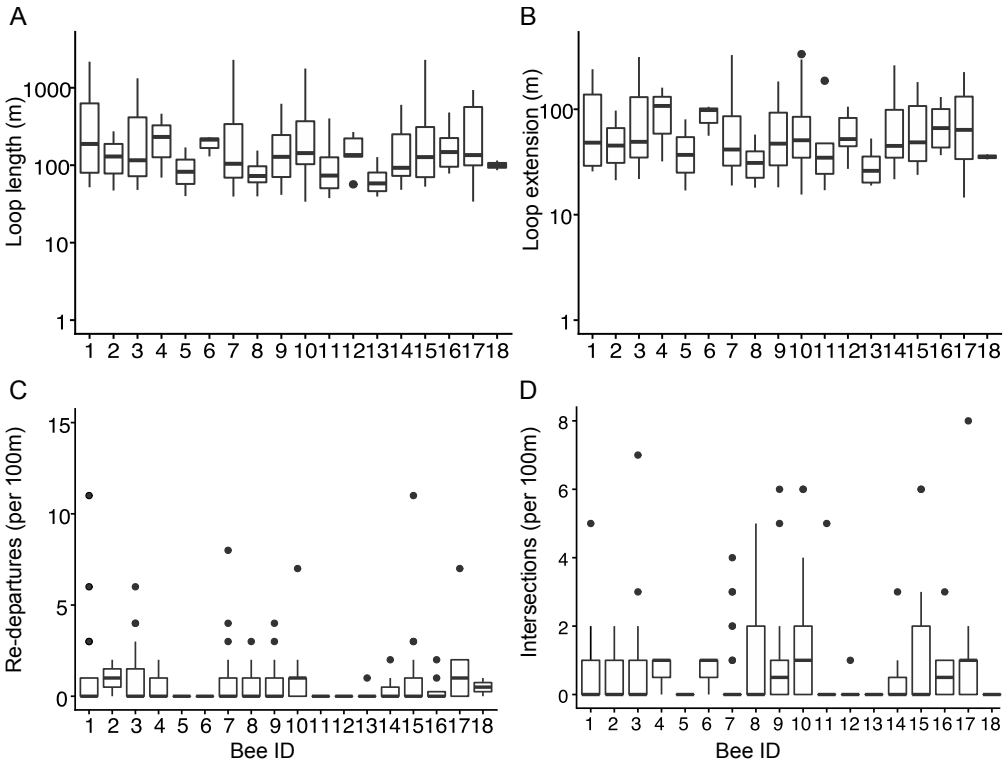

Figure 3.11: **Variability of each observable across individuals in the experimental dataset.** (A) Loop lengths (m) for each bee, as defined in Fig. 3 in the main text. Boxplots, show the median (middle line), 25 and 75% quantiles (box), range of data within 1.5 interquartile deviations (whiskers), and outliers (dots). (B) Same as A but for the loop extension (maximum distance between the nest and the individual). (C) Same as A, but for the number of re-departures per 100m traveled. A re-departure is defined as three consecutive positions such that the second position is closer to the nest than the first one, but the third is again further away than the second. (D) Same as A but for the intersections (number of times the loop intersects with itself)

I explored all 6160 different combinations resulting from the following values of each parameter:

- $\gamma \in (0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5)$
- $\Omega \in (0.01, 0.03, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.125, 0.15)$
- $\alpha \in (10, 20, 25, 30, 35, 40, 50)$
- $\eta \in (0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4)$

For each combination of parameters, I simulated 1000 loops and computed the distribution of each of the four observables. Then I computed the distance

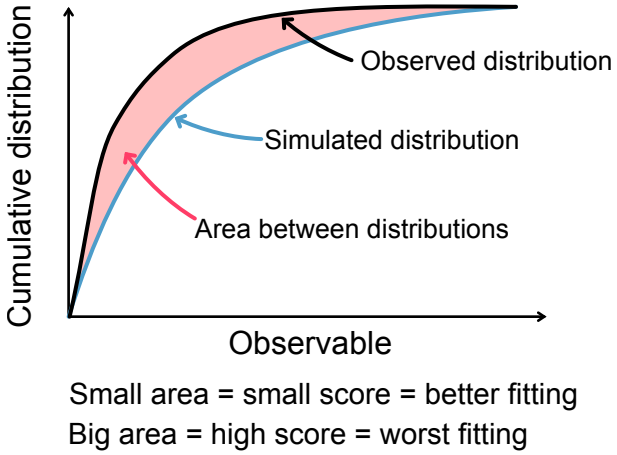

Figure 3.12: **How the scores are calculated as explained in section 3.2.3.3**

between the experimental distribution of each observable and the simulation results: for the two continuous observables (loop length and extension), this distance was computed as the area between the observed cumulative distribution function and the simulated one (Fig.3.12). For the two discrete observables (numbers of self-intersection and re-departures), the distance was computed as the sum of the absolute differences between all points of the two probability distributions. This yielded four distributions of distances over the 6160 combinations. Since the four observables are heterogeneous (two are continuous measures, two are discrete), I had to re-normalize the distances to ensure that each observable is given the same weight. I did this by translating the distances into scores according to their quantile in their corresponding cumulative distribution (e.g., a distance translated into 0.12 means that it is within the lowest 12%). Finally, I retained the combination that yielded the lower score averaged over the four observables.

3.2.3.4 The best fitting

The best parameters combination was found to be: $\gamma = 1.0\ \mathrm{s}^{-1}$, $\Omega = 0.07\ \mathrm{rad}^2$, $\alpha = 30\ \mathrm{s}$ and $\eta^* = 0.2\ \mathrm{s}^{-1}$. It corresponds to the marginal local minima for the four observables (Fig 3.13). Simulated trajectories closely resemble data trajectories (Fig 3.15), and the model is able to produce loops with an elongated shape, as well as a diversity of loop lengths. Note that this calibration procedure yields a switching time α from the "exploration phase" to the "return phase" of 30 seconds on average. This may be an underestimation since the radar can detect targets at a maximum distance of about 1km and bumblebees can fly beyond this limit depending on forage availability in the landscape (Riley *et al.*, 1996).

This unique set of parameters assumes that all bees are identical, while in reality, inter-individual differences exist, for example, due to differences in age, experience, learning, or size. However, each bee can display a large diversity of loop parameters, covering a similar range as the overall population. I, therefore, considered that separate fits for each individual were not justified. The fact that the model reproduces not only the mean but also the variability of the four observables supports this choice.

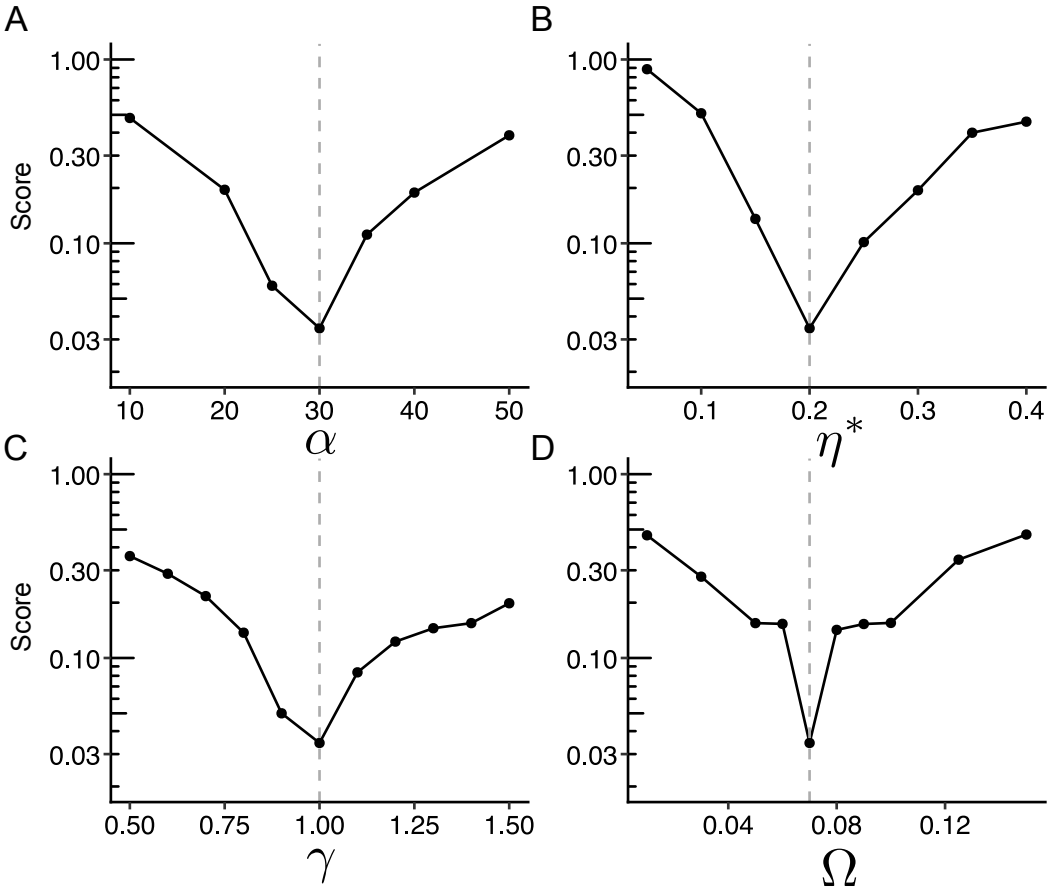

Figure 3.13: **Ratings for each parameter, when others are kept constant.** For each parameter, the rating is reported over the range of explored values, while the other parameters are kept the same ($\alpha = 30$, $\eta = 0.2$, $\gamma = 1$, $\Omega = 0.07$). For each parameter, the best rating (minimal value, dotted line) is obtained for the fitted value.

3.2.4 The effect of each parameter on loops statistics

The distributions found in the previous section allows us to better understand the constraints of each parameter over each one of the observables.

During the fitting procedure, the simulation results were compared to the

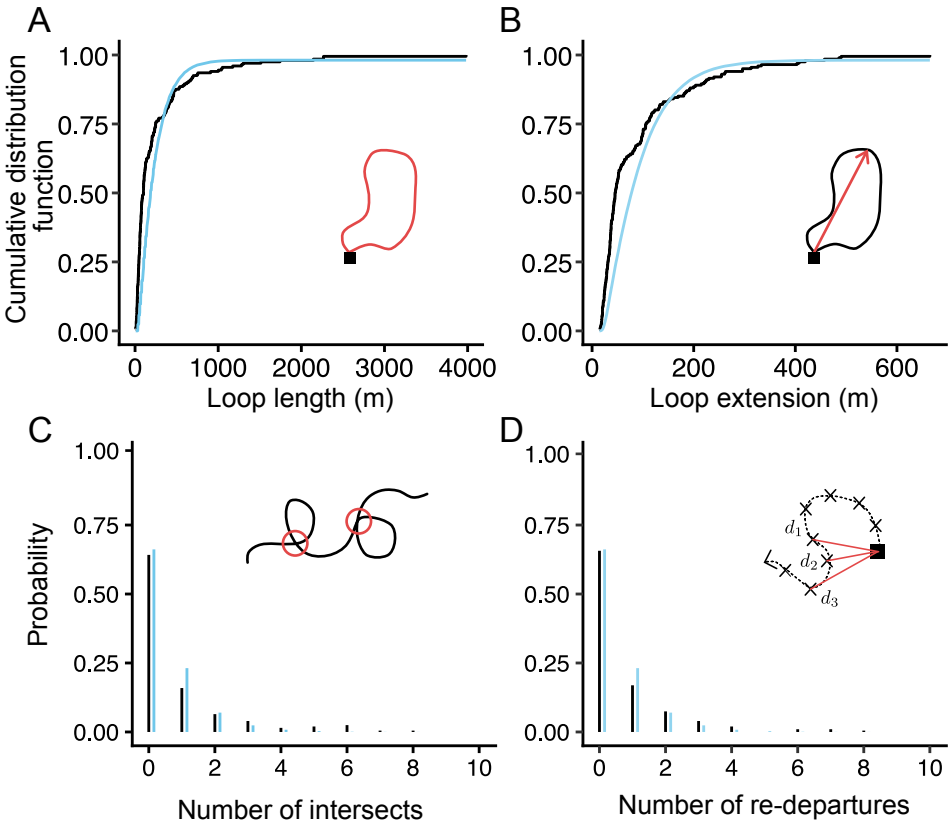

Figure 3.14: **Distributions of the four observables, for experimental and simulated data**. Black lines: experimental data. Blue lines: model predictions using the fitted values: $\gamma = 1.0$ s^{-1}, $\sigma = 0.37$ rad.s$^{-1/2}$, $\alpha = 30$ s and $\eta* = 0.2$ s^{-1}. Insets: Schematic of each observable. (A) Cumulative distribution function of loop lengths for our full dataset. (B) Same as A, but for the loops extension. (C) Probability distribution of the number of trajectories intersects per 100m traveled. (D) Same as C, but for the number of re-departures per 100 m traveled.

experimental distributions of each observable, resulting in four different scores. These scores were then averaged to obtain a "total" score, which was used to rank the different combinations of parameters. The top and worst 1% of the scores were selected, and their distribution is represented in figure 3.16.

The best parameters, represented by the dotted lines, were chosen based on the best average of the "total score." The "total score" distributions showed that some parameters had worse scores that were far away from the selected parameters. For instance, η^* had worse scores when the values were too low, while the inverse was the case for α. When η^* was too low, the model approached a diffusive process, and high values of α resulted in longer periods before the switch to the "return phase." These outcomes indicate that a diffusive process is not the best descriptor for the experimental data. Conversely, γ had a more uniform distribution, although

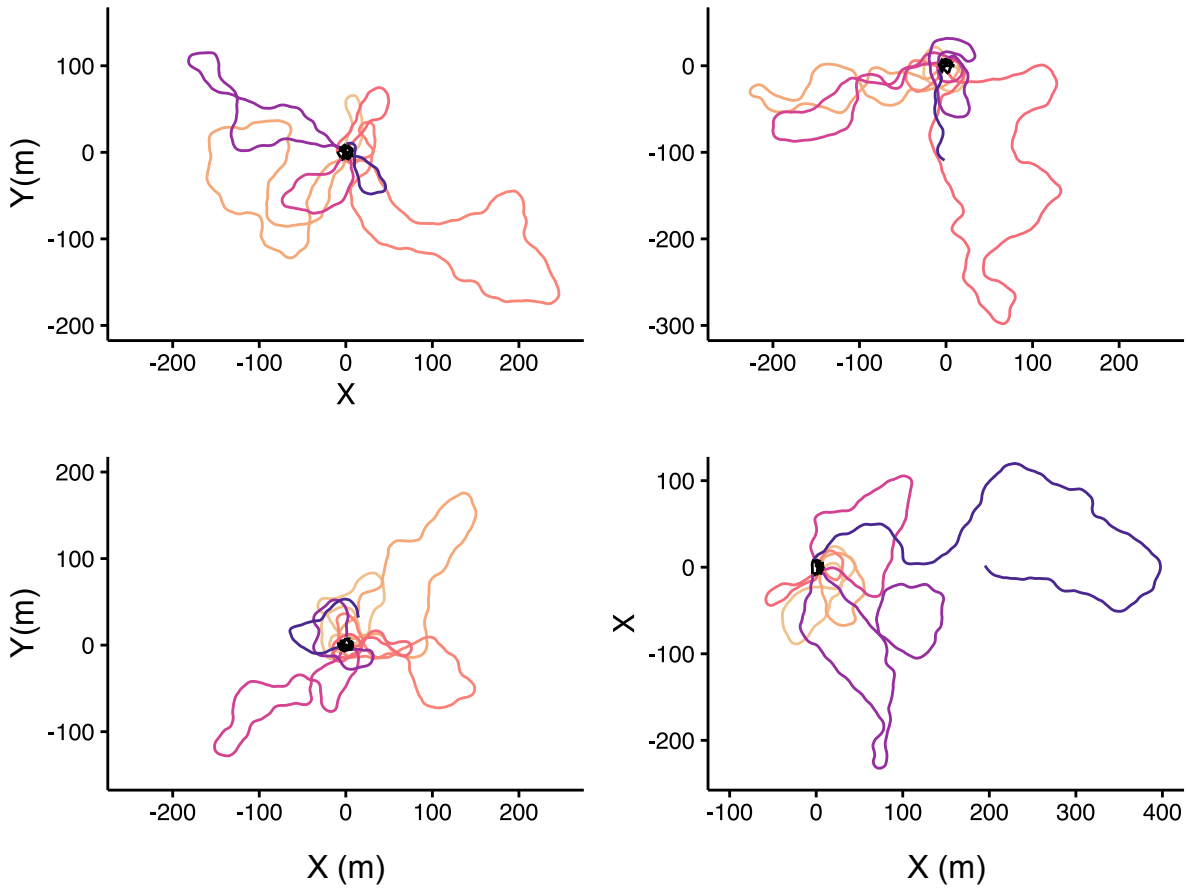

Figure 3.15: **Example of simulated trajectories.** Different colors represent different flight loops around the nest. The sequential order of the loops is represented by the color gradient where the first loops have lightest colors (yellow to purple). The parameters are: $\gamma = 1.0$ s^{-1}, $\sigma = 0.37$ rad.s$^{-1/2}$, $\alpha = 30$ s and $\eta* = 0.2$ s^{-1}.

smaller values appeared to be the worst.

Looking at each observable, the loop length was heavily influenced by η^*, which determined the strength of the attraction to the nest and indirectly affected the time spent coming back to the nest and the length of the loops. The noise parameter Ω chosen as the "total best" did not capture the best one when only looking at the length. The worst scores had a bimodal distribution, indicating that trajectories went too far away when there was not enough noise, and too sinuous when there was too much noise. The variance of the distributions of the worst scores for α was high, indicating that its effect on the loop length could be compensated for by adjusting other parameters.

For the loop extension, α and η^* were once again the most decisive parameters. The "worst 1%" had a bimodal distribution, indicating that too low or too high values resulted in the worst scores. A good pairing of $\alpha\eta^*$ was necessary for the best fit. The value of α determined when the switch to the return phase occurred, while the value of η^* determined if the trajectory went straight to the nest or continued to drift away for a bit. The variance Ω had a similar effect as for the loop length.

The redepartures variable was the worst-fitted, with the best value being far away from the "best 1%" for all four parameters. α did not have any effect, and the distributions looked the same for both cases.

The most determining parameter for the intersects was Ω, with a bimodal distribution of the worst scores. High values of Ω resulted in too many intersections, while too low values resulted in too few. The autocorrelation γ had a similar effect, with too low values resulting in trajectories with many small loops that intersected too often, and too high values resulting in trajectories that were too straight with very few intersects.

Overall, γ had a neutral effect across all observables.

3.2.5 Calibration when the attraction to the nest is kept constant

At the beginning of the thesis, a previous model where η is kept constant, so $\eta(t) = \eta^*$, was proposed. In this case, there is not a "switch" between the "exploration phase" and the "return phase". This model is presented in Chapter 2. At first glance, this model is not able to reproduce the "elongated" loops observed in experimental data. It rather produce "round" loops. I used the same calibration procedure to calibrate this model. The combinations of parameters with the best

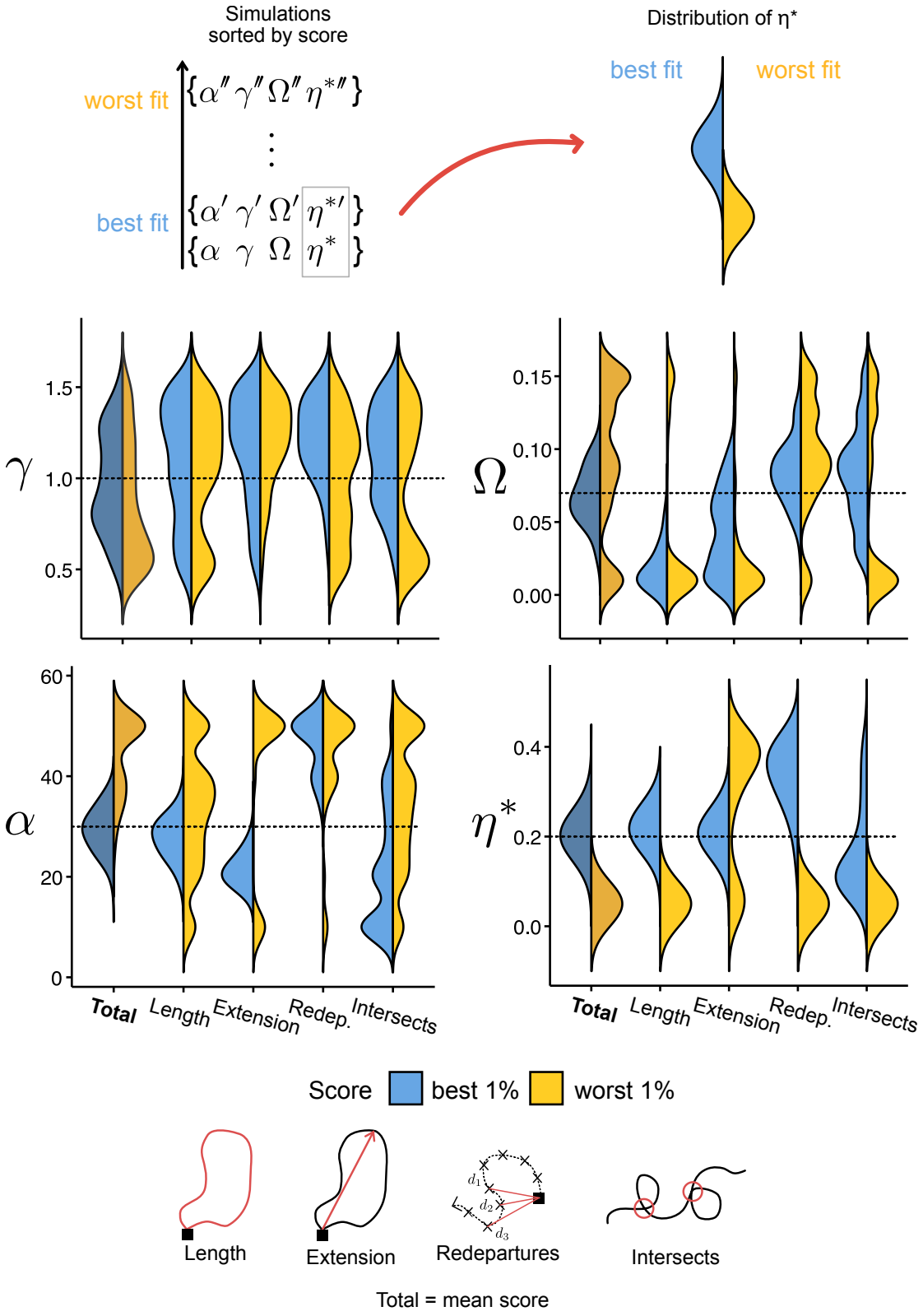

Figure 3.16: **Distribution of the best and the worst scores for each observable.** As commented in section 3.2.4

fist corresponds to $\gamma = 0.8$, $\eta^* = 0.1$, $\Omega = 0.07$. This versions of the model, yields an auto-correlation parameter and a force of attraction weaker than the final version of the model (respectively $\gamma = 1$ and $\eta^* = 0.2$) even if the variance Ω remains the same. With the weakest attraction η^* I am able to reproduce the loop length (Fig 3.18.A, yellow line). If the attraction were stronger, the individual would never be able to get away from the nest for a long enough amount of time to replicate the experimental data. The value of the variance Ω allows to reproduce the experimental distributions of the intersects and the re-departures (Fig 3.18.C-D, yellow line). However this parameters doesn't allow to replicate the loop extension. Here, loops are shorter (they don't exceed 150m) (Fig 3.18.B, yellow line) and more round (ref figure traj). The loops observed in the experimental data are elongated (Fig 3.9). For this reason, I decided to switch to our final version of the model where having a switch from an "exploration phase" to a "return phase" allows to have loops with an elongated form.

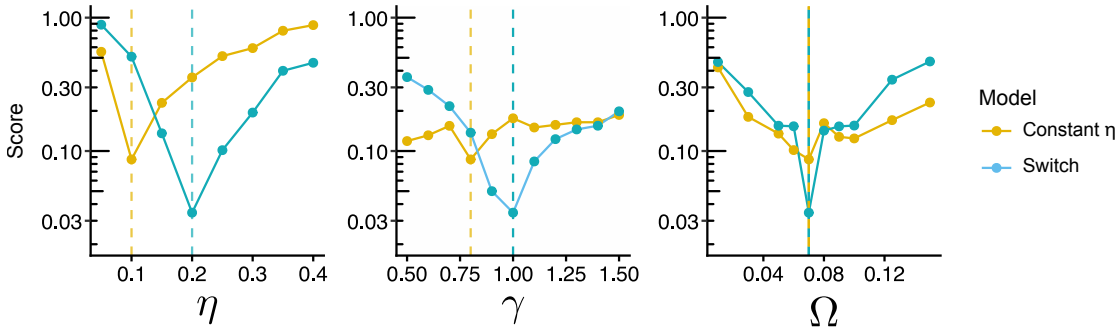

Figure 3.17: **Ratings for each parameter, when others are kept constant for both versions of the Model.** For each parameter, the rating is reported over the range of explored values, while the other parameters are kept the same. For the model with constant attraction η: $\eta = 0.1$, $\gamma = 0.8$, $\Omega = 0.07$. For the model with the switch: $\eta = 0.2$, $\gamma = 1$, $\Omega = 0.07$ .For each parameter, the best rating (minimal value, dotted line) is obtained for the fitted value.

3.3 Conclusion

In this chapter, I have demonstrated the possibility of calibrating the Model introduced in Chapter 2 using experimental data from two different insects and capturing various characteristics of their trajectories. To achieve this, I presented two distinct calibration methods, one for high-frequency data and the other for low-frequency data.

In the next chapter, I will utilize the parameters obtained through calibration

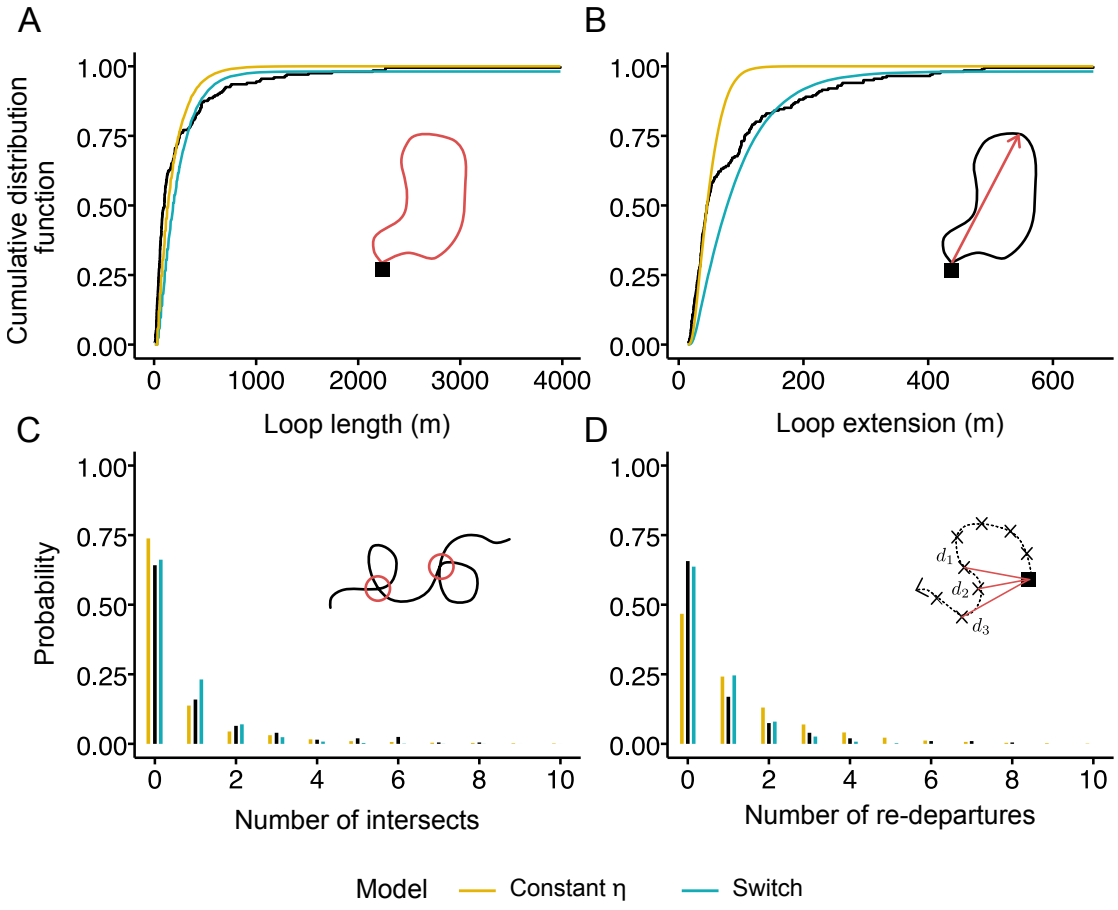

Figure 3.18: **Distributions of the four observables, for experimental and simulated data with both versions of the model**. Black lines: experimental data. Red lines: model predictions using the optimal set: $\gamma = 1.0\ \mathrm{s}^{-1}$, $\sigma = 0.37\ \mathrm{rad.s}^{-1/2}$, $\alpha = 30$ s and $\eta* = 0.2\ \mathrm{s}^{-1}$. Insets: Schematic of each observable. (A) Cumulative distribution function of loop lengths for our full dataset. (B) Same as A, but for the loops extension. (C) Probability distribution of the number of trajectories intersects per 100m traveled. (D) Same as C, but for the number of re-departures per 100 m traveled.

to describe the trajectories of bumblebees and estimate the probabilities of finding flowers under different scenarios.

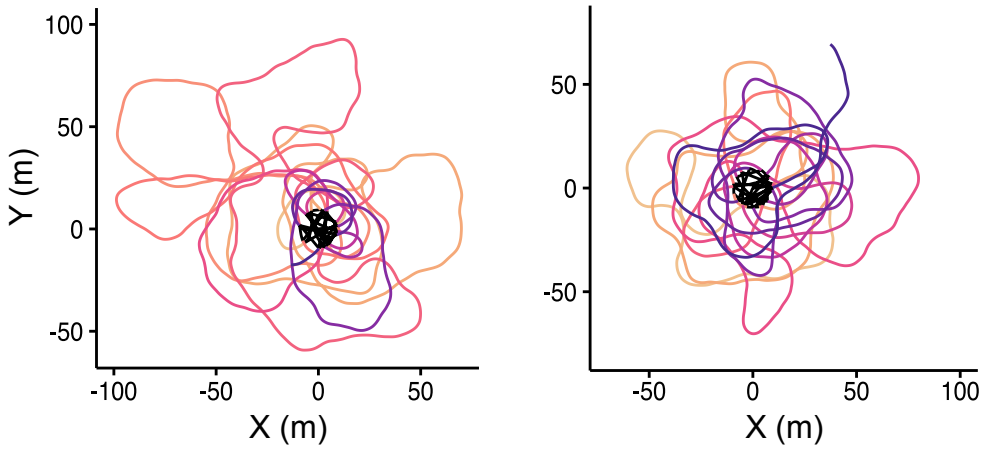

Figure 3.19: **Example of trajectories where η is kept constant.** The set of parameters of the simulation are the best ones according to the calibration procedure 3.17. The parameters are $\eta = 0.1$, $\gamma = 0.8$ and $\Omega = 0.07$

Chapter 4

Predicting flower discovery

Contents

Pollinators, such as bees, wasps, flies, butterflies, bats, and birds, mediate a key ecosystemic service on which most terrestrial plants and animals, including humans, rely. When foraging for nectar, animals transfer pollen between flowers, which facilitates plant reproduction. Understanding how pollinators move, find and choose flowers is thus a key challenge of pollination ecology (Mayer *et al.*, 2011). In particular, this may help predict and act on complex pollination processes in a context of a looming global pollination crisis, when food demand increases and populations of pollinators decline (Buchmann & Nabhan, 1996; Goulson *et al.*, 2015).

Foraging pollinators have long been assumed to move randomly (Lenz *et al.*, 2013; Reynolds *et al.*, 2007a,b, 2009) or use hard-wired movement rules such as visiting the nearest unvisited flower (Ohashi *et al.*, 2007), exploiting flower patches in straight line movements (Pyke & Cartar, 1992), navigating inflorescence from bottom to top flowers (Pyke, 1978), or using win-stay lose-leave strategies (Lihoreau *et al.*, 2016). Pollination models relying on these observations typically predict diffusive movements in every direction (Vallaeys *et al.*, 2017). However, recent behavioral research shows this is not true when animals forage across large spatial scales (Collett *et al.*, 2013). In particular, studies using radars to monitor the long-distance flight paths of bees foraging in the field demonstrate that foragers learn features of their environment to navigate across landscapes and to return to known feeding locations (Woodgate *et al.*, 2016; Brebner *et al.*, 2021). This enables them to develop shortcuts between feeding sites (Menzel *et al.*, 2012) and build efficient multi-location routes (traplines) minimizing overall travel distances (Lihoreau *et al.*, 2012b; Woodgate *et al.*, 2017). These routes are re-adjusted each time a feeding site is depleted and new ones are discovered (Lihoreau *et al.*, 2010).

How bees learn such foraging routes has been modeled using algorithms implementing spatial learning and memory (Lihoreau *et al.*, 2012a; Reynolds *et al.*, 2013; Dubois *et al.*, 2021). While this has greatly advanced our understanding of bee exploitative movements patterns, none of these models have looked at search behaviors, either assuming insects already know the locations of all available feeding sites in their environment or discover them according to fixed probabilistic laws (i.e. the probability to discover a flower at a given location is proportional to 1/L2 where L represents the distance to that flower (Lihoreau *et al.*, 2012a; Reynolds *et al.*, 2013; Dubois *et al.*, 2021)).

However, experimental data indicate that this is not the case. Firstly, bees, like many pollinators, are central place foragers so every foraging trip starts and ends at the nest site (Capaldi *et al.*, 2000). This implies that their range of action is

limited. Recordings of bee search flights show how individuals tend to make loops centered at the nest when exploring a new environment and looking for flowers (Woodgate *et al.*, 2016; Capaldi *et al.*, 2000). These looping movements are not compatible with the assumption that bees make diffusive random walks or Lévy flights (Edwards *et al.*, 2007; Benhamou, 2007). Secondly, the spatial structure of the foraging environment itself may also greatly influence flower discovery by bees. In particular, the probability of finding a flower heavily depends on the location of the flower visited just before, ultimately affecting the direction and geometry of the routes developed by individuals (Lihoreau *et al.*, 2012b; Woodgate *et al.*, 2017; Lihoreau *et al.*, 2010, 2012a; Ohashi *et al.*, 2008; Lihoreau *et al.*, 2011). Since bees are more attracted to larger flowers than to smaller ones (Stout, 2000), this suggests that small isolated flowers could be missed if they are located next to a larger patch. Such "masking effect " on the probability of visiting specific flowers depending on the presence of other flowers around could have significant consequences for bee foraging success, for instance by precluding the discovery of some highly rewarding flowers. This could also influence plant pollination if bees are spatially constrained to single flower patches and plant out-crossing is limited.

Here I explored potential effects by developing a model of bee search movement simulating the tendency of bumblebees to make loops around their nest. I used the Model with the parameters established in the previous chapter to examine the probability of bees discovering flowers in environments defined by resources of various sizes and abundances. We hypothesized that looping movements characteristic of bee exploratory flights combined with perceptual masking effects by which the probability of finding given flowers is affected by the presence of others would result in strikingly different predictions for flower discovery rates than the typical diffusive random walk movements.

4.1 Methods

4.1.1 Simulations

I used Monte Carlo simulations to explore different scenarios. I simulated 100000 trajectories in each different scenario. Every simulation starts at the nest.

In order to simulate a flower encounter I start by defining the size of the nest, the flowers, and their location.

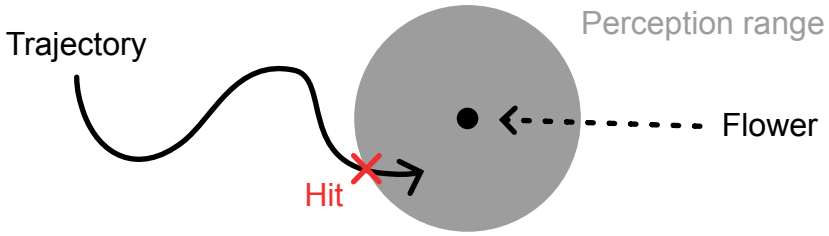

Figure 4.1: **Scheme of a flower discovery.** The perception range is corresponds to the distance at which the individual can see the flower. See section 4.1.2 for details

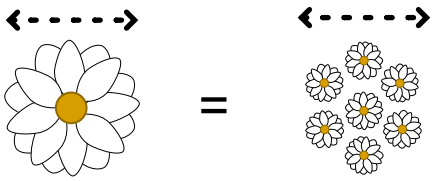

Figure 4.2: **Scheme of a "patch" of flowers.** There is no distinction between a single flower and a similar size patch of smaller flowers

4.1.2 Flower Detection

Bumblebees can detect an object when it forms an angle of 3° on the retina of their compound eyes (Kapustjansky *et al.*, 2010)). Therefore, for every model simulation, I set the flowers' size and calculated the distance at which the bees are able to detect them. I call this the "perception distance". I considered that a bee discovered a flower when it was located at a distance to the bee inferior to the perception distance. I did not take into account the olfactory perception since it could be less reliable because of other factors like wind direction and the flower's species. If taken into account, this would only impact the perception distance of the flowers and the results would not be qualitatively different.

In the simulations, there is no distinction between big individual flowers and patches of similar size composed of several flowers. A flower with a size of, for example, 1 meter may seem not realistic but it should be understood as a big patch formed by smaller flowers (Fig 4.2).

4.2 Simple scenarios

First, in a setup with two flower patches present in the neighborhood of the nest, I compared how statistics of discovery are affected by the attraction component of the model with regard to what a pure diffusive model would yield. Next, I present how the geometrical arrangement of the two patches affects the statistics of the first discovery. Then, I show how those statistics are modified when in presence

of a third patch.

4.2.1 Comparison to diffusive model

I compared the predictions by the Model using the set of parameters fitted in Chapter 3 by the same model, except that I removed the attraction-to-nest component. This corresponds to the Persistent Turning Walker (Gautrais *et al.*, 2009). In the latter case, the behavior becomes purely diffusive at large times.

I used a setup with a central nest, and two circular patches of flowers (denoted F1 and F2) which are of the same size f and placed at equidistant locations from the nest (at a distance l), and symmetrically around the nest exit.

I considered 3 different scenarios; illustrated in Fig 4.3:

1. **Scenario "distance scale"**: I increased the distance l between patches, of fixed diameter $f = 0.5$ m, scaling it according to $l = 10 + 10\ s$ with scale $s \in [2..9]$, hence from 30 m to 100 m. The value 1 is not included in the interval because in that case, the flower is too close to the nest and can be seen from it.

2. **Scenario "size scale"**: I increased the size of patches, located at fixed distance $l = 100$ m, scaling the diameter according to $f = 0.1 + 0.1\ s$ with scale $s \in [1..9]$.

3. **Scenario "distance and size scale"**: I scaled simultaneously distance and size.

I compared the statistics of either finding a patch or getting back to the nest within a maximal amount of time and of hitting a patch before hitting back to the nest. Overall, the two models yield widely different predictions.

We can first note that the Model ensures that the bee either finds a patch or hits back the nest within one hour in every case. On the contrary, pure diffusive search results in lots of censoring by the maximal amount of time, with a hit hardly more than one every two loops. It can even be as low as one every five loops for the smallest scales in patch size, which is the dominant factor that controls hit probability (Fig 4.4 A,C and E).

Regarding the probability of hitting the nest before a flower, we can see that this probability is always higher for the diffusive model (Fig 4.4 B,D and F). The distance effect is the most prominent, as it strongly affects the probability of hitting a flower before hitting back the nest (i.e. finding a flower within one

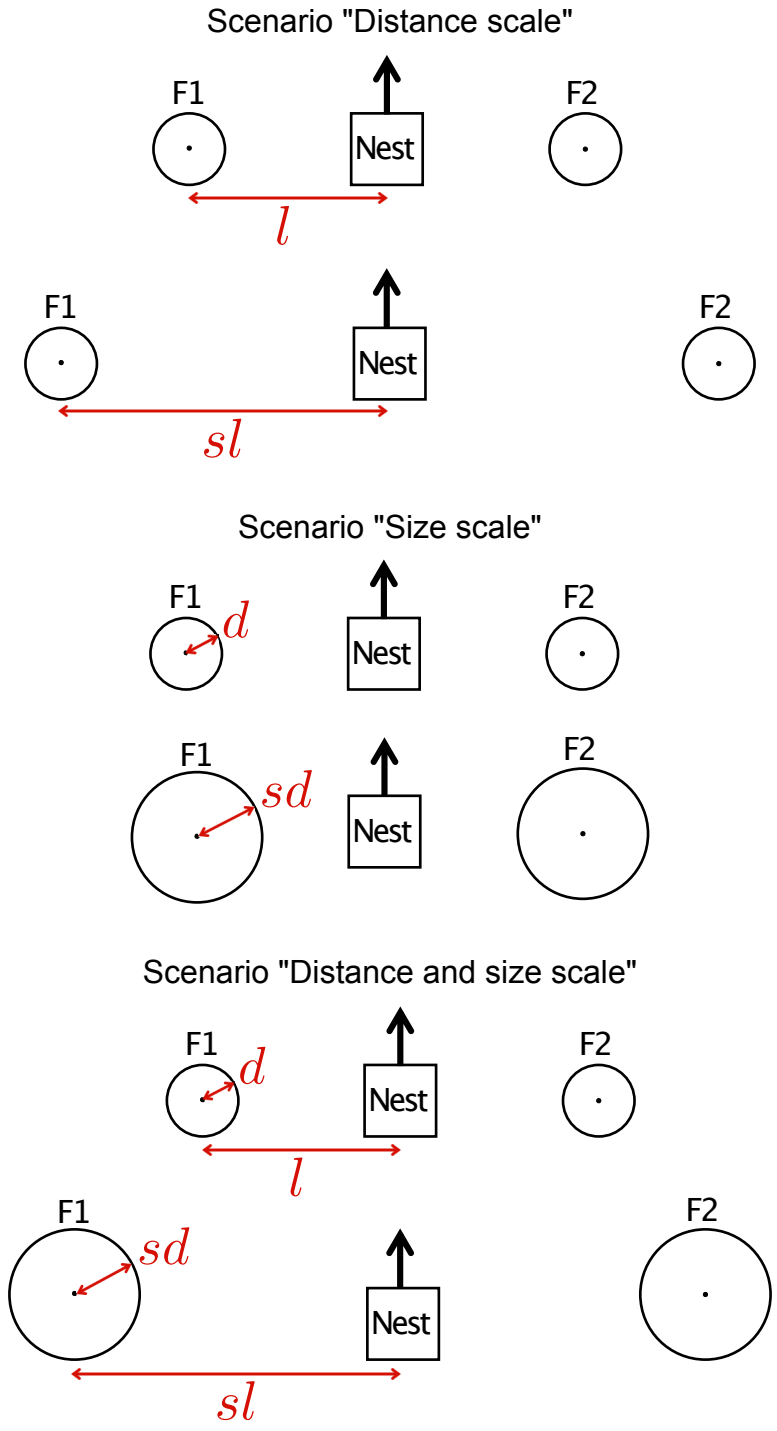

Figure 4.3: **Scheme of the 3 different scenarios tested.** Each flower is represented by a dot and its perception range is represented by a circle.

loop) whereas it has only limited impact on this probability when merely diffusing (Fig 4.4 B). Considering now flowers 100 m away from the nest (Fig 4.4 D), the probability of finding them within a loop is quite low for the Model, because this distance is not often reached within a loop (about only 25% loops have maximal extensions beyond 100 m), whereas the diffusive mode can easily reach it within the time limit. With the range of sizes that we have considered, the size itself makes a poor difference. When combined, the distance effect dominates (Fig 4.4 F).

Overall, the model allows that patches in the vicinity to be actually discovered while ensuring that bees regularly return to their nest.

4.2.2 Effect of scale upon the masking effect

I then tested the influence of a potential effect in which the probability of discovering a flower does not only depend on its distance to the nest but can also be influenced by the presence of other flowers around it. This dependence exists because a bee that finds a flower might not continue its trajectory, but might rather stop to collect nectar. Once the nectar collection is over, the bee may continue exploring, but after visiting a few flowers the bee returns to the nest to unload its crop. For example, in a scenario where there are just 2 flowers equidistant to the nest, both flowers should be visited equally. However, if another flower is added, it can capture visits that would otherwise visit one of the original flowers, reducing the probability that it's discovered (4.5).

To quantify the effect of geometrical configurations of flowers upon the first discovery statistics, I considered the pair of flowers of the same size and equidistant to the nest as above. I estimated the number n_1 of trajectories that hit F1 before F2 (resp. n_2), discarding loops that end in the nest before hitting any flower. We quantify the odd hitting F2 before F1:

$$O_{(1,2)} = \frac{n_2}{n_1} \tag{4.1}$$

For instance, a value $O_{(1,2)} = 2$ means that F2 is hit before F1 twice as many times as F1 is hit before F2.

Then, I added a third patch (F3) at a location equidistant between the nest and F1, and of the same size as F1 and F2. Therefore, a fraction of paths that would have led to F1 (resp. F2) are interrupted before reaching F1 (resp. F2)

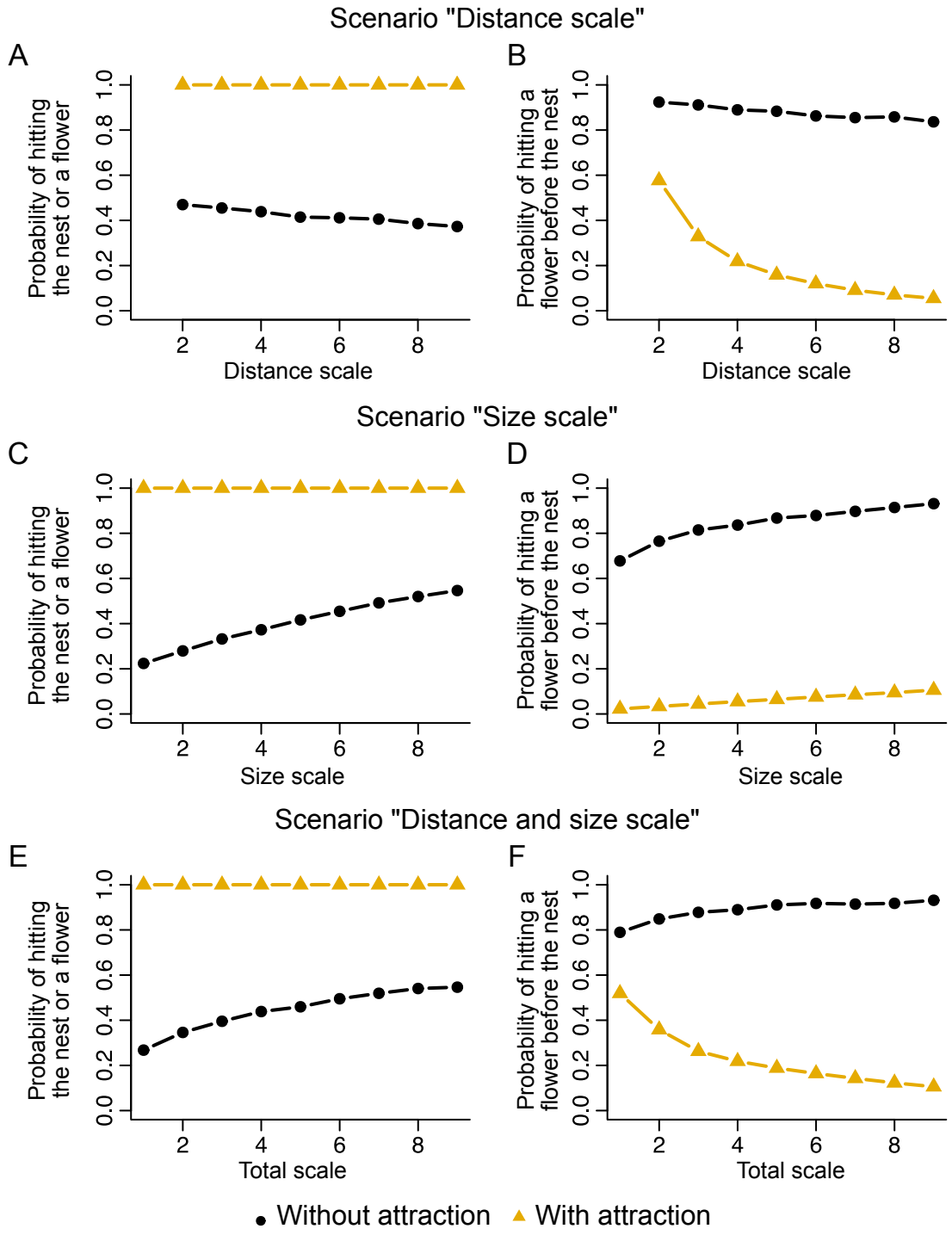

Figure 4.4: (A-C-E). Probability of hitting a flower or the nest within one hour. B-D-F. Probability of hitting a flower before hitting back the nest

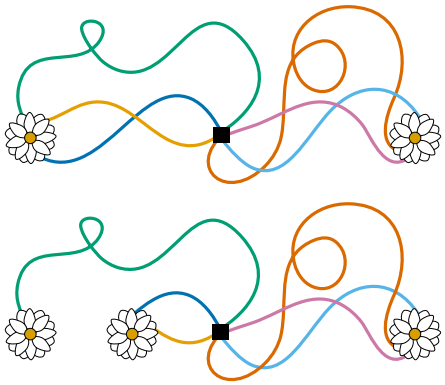

Figure 4.5: **Illustration of the masking effect.** The probability of discovering a flower depends on the presence of other flowers. In a scenario where there are just 2 flowers equidistant to the nest, both flowers should be visited equally (top). However, if another flower is added, it can capture visits that would otherwise visit one of the original flowers (bottom). Black square: nest.

by hitting F3 first. Since the geometry has become asymmetrical, the respective fractions of paths to F1 and F2 are expected to be different. I denote $O_{(1,2,3)}$ the corresponding odd, hitting F2 before F1, in presence of F3.

In order to quantify the "masking effect" (ME), we consider the odds ratio:

$$\mathrm{ME} = \frac{O_{(1,2)}}{O_{(1,2,3)}} \tag{4.2}$$

Finally, I considered all 3 scenarios in two conditions regarding the orientations for nest exit: either symmetrical around the patches, or oriented towards F1. This introduces a second source of geometry asymmetry.

For all scenarios with the symmetrical configuration, in the absence of F3, patches F1 and F2 are visited equally, as expected ($O_{(1,2)} = 1$). When introducing F3, the difference between the fraction of paths to F1 which hit F3 before reaching F1, and the fraction of paths to F2 which hit F3 before reaching F2 translates into $O_{(1,2,3)} > 1$, meaning that F2 is now more often discovered as first than F1.

For a given patches' size ($f = 0.5$ m) in the scenario "Distance scale", the effect is the strongest for the shortest scale, where F2 is first hit almost twice more often than F1. This effect decreases only slightly as the configuration is extended (Fig 4.6A). This effect translates into a masking effect which is slightly lower than for a diffusive model at a short scale, and which becomes larger for larger scales (Fig 4.6B).

When patches F1 and F2 are located 100 m away from the nest (and hence F3 50 m away) in the scenario "Size scale", the effect of increasing patches size is

the strongest for the larger size, where F2 is first hit more than twice more often than F1 (up to $O_{(1,2,3)} \simeq 2.16$ for $s = 9$). It is lower and lower as the patches' size decreases, becoming negligible for patches of diameter 0.2 m (Fig 4.6C).

When, in scenario "Distance and size scale", both patches' size and locations are extended simultaneously, the effect of patches' size upon the odds appears dominant over the distance effect (Fig 4.6E and F). While the masking effect predicted by the diffusive model becomes insensitive to the scale, it is still increasing with scale, from negligible for scale 1 (patches of diameter 0.2 m, F1, F2 at 20 m) to larger than 2 for scale 9 (patches of diameter 1 m, F1, F2 at 100 m) (Fig 4.6F).

I also considered an asymmetric configuration where the exit of the nest was set in the direction of F1. This asymmetry favors the number of first discoveries of F1 in relation to F2 even in the absence of F3. This means that $O_{(1,2)}$ and $O_{(1,2,3)}$ will be lower than 1. For instance, in Fig 4.7E at scale 1 without F3, $O_{(1,2)} = 0.5$ means that F1 is discovered first twice more often than F2, purely due to nest exit asymmetry. The lower $O_{(1,2)}$ and $O_{(1,2,3)}$, the stronger this effect of nest exit asymmetry. The predictions for the asymmetric effect are very different between the two models because in the Model, the limitation of exploring around the nest covers an area that is biased towards F1 more strongly than in the diffusive model (Fig 4.8).

In scenario "Distance scale", without F3, the effect of the nest's exit asymmetry becomes stronger for increasing scales (decreasing $O_{(1,2)}$) down to an asymptotic value below 0.2 (F2 discovered first one every five times) (Fig 4.7A). This pattern is opposite to the predictions from the diffusive model where the asymmetry effect decreases for larger scales (where $O_{1,2}$ increases from 0.3 to 0.6, not shown). Since nest exit is also oriented towards F3, the presence of F3 attenuates the asymmetry effect, especially at the smallest scale where it nearly compensates for it. Here, the masking effect is maximal over all explored scenarios (Fig 4.7B). This attenuation of the asymmetry effect tends to vanish for larger scales, and so does the masking effect. Compared to the diffusive model, the masking effect is lower at small scales but becomes higher for larger scales (F1 and F2 50 to 100m away from the nest).

With increased patches size located 100 m away from nest (Fig 4.7C), $O_{1,2} \simeq 0.2$ for every size: the size of the patch remains too small for having a compensating effect upon the nest exit asymmetry, and the distance effect dominates (same odds as the largest scale of distances). Note that, in the diffusive model, this odd decreases with scale, meaning that the increased size of F1 has an effect upon its discovery as first (not shown). In the presence of F3, there is a little effect with small patches (scale=1, $f = 0.2$m). But, while $O_{1,2}$ stay constant with scale,

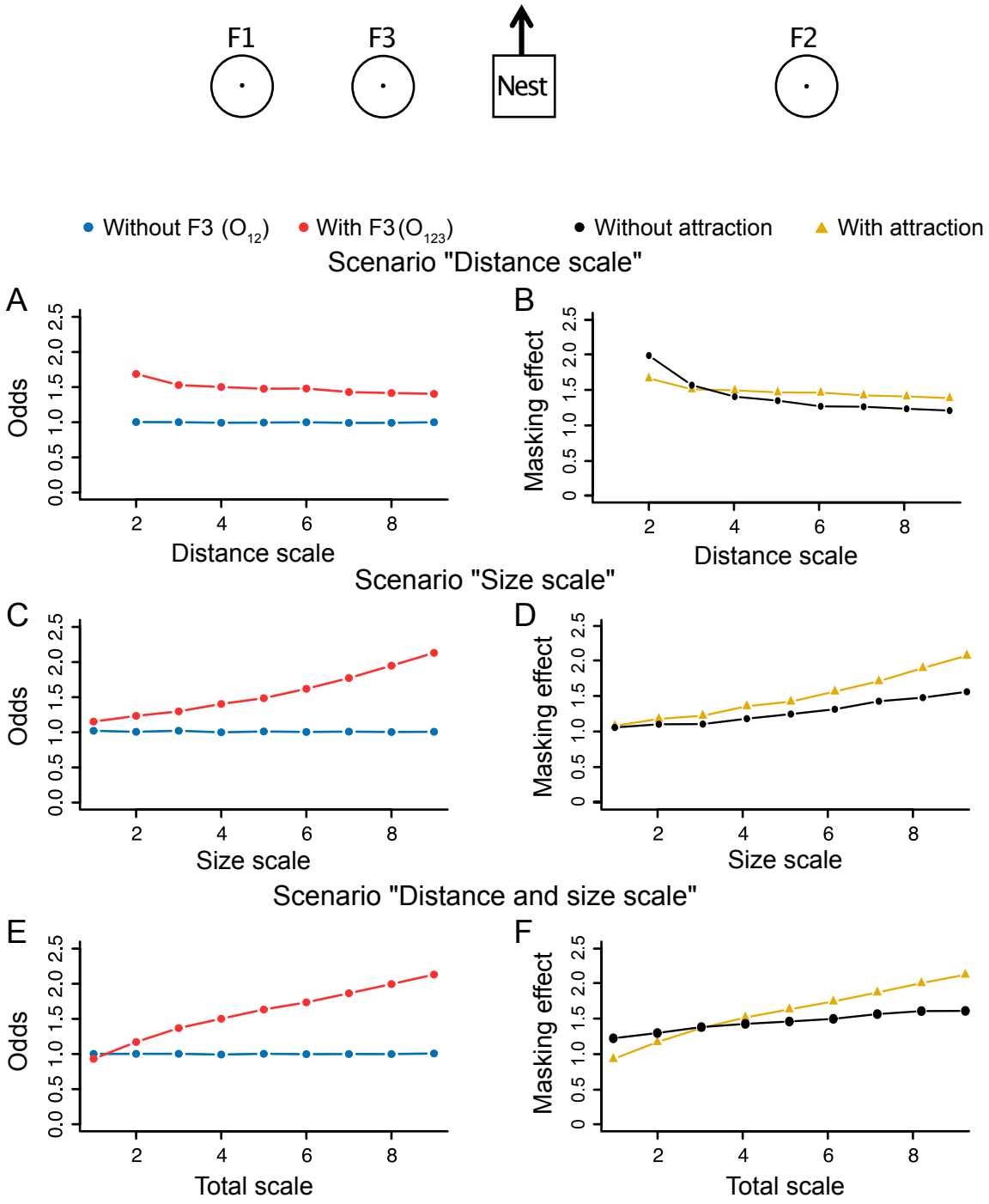

Figure 4.6: **Odds and masking effect in symmetrical scenarios.** (A) Odd of hitting the flower 1 (F1) before flower 2(F2), with and without flower 3 (FR3), in function of the distance scale. (B) Masking effect in the function of the distance scale for the Model with and without attraction. (C) Same as A but for the Size scale. (D) Same as B but for the Size scale. (E) Same as A but for the Distance and Size Scale. (F) Same as D but for the Distance and Size scale.

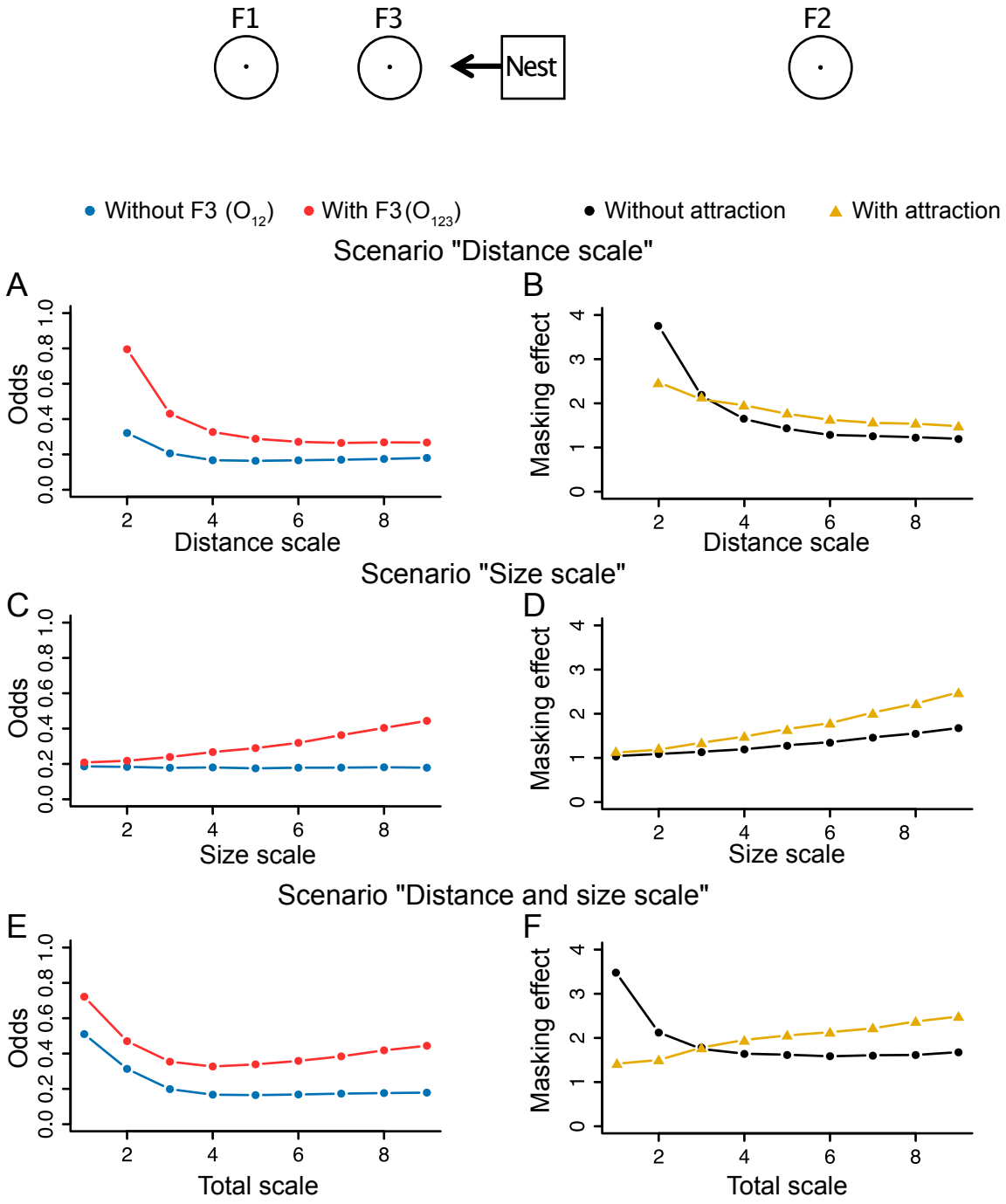

Figure 4.7: **Odds and masking effect in asymmetrical scenarios.**(A) Odd of hitting the flower 1 (F1) before flower 2(F2), with and without flower 3 (FR3), in function of the distance scale. (B) Masking effect in the function of the distance scale for the Model with and without attraction. (C) Same as A but for the Size scale. (D) Same as B but for the Size scale. (E) Same as A but for the Distance and Size Scale. (F) Same as D but for the Distance and Size scale.

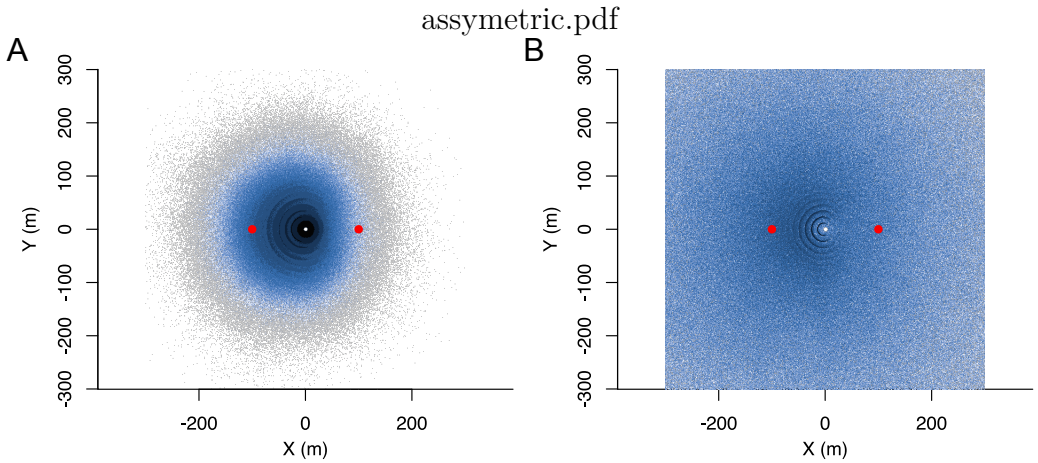

Figure 4.8: **Presence map of the asymmetrical configuration.** (A) Model with the attraction component (B) Model without the attraction component. The red dots symbolize the flowers

$O_{1,2,3}$ increases (to $O_{1,2,3} \simeq 0.5$), as in the symmetrical scenario. Due to the asymmetry of exploration, the masking effect increases faster than the one given by the diffusive model as F3 size increases (Fig 4.7D).

In contrast to the symmetrical configuration, the effect of the scale in the scenario "Distance and size scale" is not monotonic, as it compounds both distance and size effects (Fig 4.7E). At small scales (from F1 and F2 at 20m and f = 0.2m), the distance effect dominates, and both odds decrease with scale. For scales beyond scale 4 (F1 and F2 at 50m and $f = 0.5$), the odd $O_{1,2} \simeq 0.2$ like in the two previous scenarios, but $O_{1,2,3}$ increases because the size effect becomes dominant.

As far as the masking effect is concerned (Fig 4.7F), the non monotonic trend of $O_{1,2,3}$ still translates into a monotonic trend. The composition of decreasing trend due to distance scale and increasing trend due to scale size translates into an increasing trend, dominated by size effect, yet with a slower slope than for size alone. In this scenario, the mixed effects of the patches' size and distance upon the masking effect display the largest contrast to predictions from the diffusive model, which is dominated by the distance effect.

4.2.3 Conclusion

In both symmetrical and asymmetrical nest exit orientations, the Model's predictions are primarily influenced by the scale effect, whereas the diffusive model's predictions are primarily influenced by the distance effect. Consequently, the masking effect in the Model is similar for both orientations, with parallel trends

and a slight shift. In contrast, the diffusive model is highly sensitive to orientation and produces opposing trends.

This analysis in very simple and theoretical scenarios shows the importance of taking into account the attraction component (in comparison to a diffusive model) of The Model when looking at the probability of discovering flowers. They produce predictions that could be easily tested experimentally with the use of artificial flowers.

4.3 Predictions at ecologically relevant scales

In this section, I will analyze the probabilities of finding flowers in more realistic scenarios than the previous sections, where several flowers are randomly distributed around the nest.

4.3.1 The probability for a flower to be discovered

For a flower to be pollinated, it must first be located. I will begin by examining the relationship between flower density and the likelihood of discovery, which is expected to have a significant correlation with pollination probability.

I estimated these probabilities in function of the distance between the flowers and the nest by creating environments with a uniform distribution of flowers. The environment around the nest was divided into bins with a width of 10 meters as in figure 4.9. I simulated trajectories starting at the nest and counted how many encounters there were in each bin. Each probability P was calculated following :

$$P = \frac{\frac{N_v}{T}}{N_f} \tag{4.3}$$

where N_v is the total number of visits in the bin, T is the total number of simulated trajectories and N_f is the number of flowers in the bin.

I estimated the probabilities of flowers to be discovered (and thus potentially pollinated) by bees in a simulated field characterized by a random and uniform distribution of flowers, an average density of 1.3 10^{-4} flowers/m^2, and a diameter of 70 cm.

Using these conditions, I simulated 1000 foraging trips, each of them lasting 900 seconds, and for each flower, I computed the probability to be found in a given trip (i.e., the proportion of simulations in which the trajectory overlaps with the flower's area of attraction). This probability rapidly falls exponentially with the

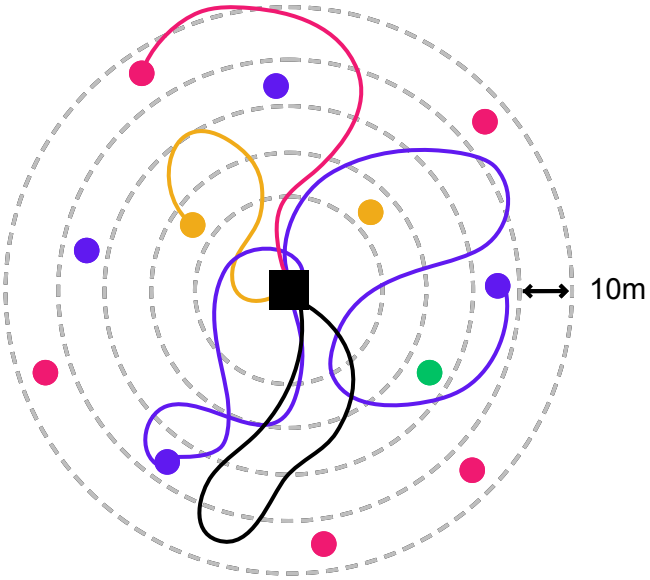

Figure 4.9: Example of how the probability of discovering a flower in the function of their distance to the nest is calculated following equation 4.3 The color of the trajectory represents the color of the flower that was discovered by it. Trajectories are just represented until the "hit" of the flower. For the purple flowers: $N_v = 3$, $T = 6$, $N_f = 4$ and $P = \frac{\frac{3}{6}}{4} = 0.125$

distance between the flower and the nest (Fig. 4.10A, red line).

For the sake of simplicity, I assumed that each bee returns to the nest after discovering a single flower. The first qualitative consequence of the masking effect is to reduce the probability that flowers distant from the nest are discovered (Fig. 4.10A, blue). The second consequence is that it can introduce a dependence on flower density on the discovery rate. In the absence of masking, only two factors determine the probability that a flower is discovered: its size (which determines the distance from which it can be perceived) and its distance to the nest. In contrast, when the masking effect is taken into account, the number of discovery visits also depends on the overall density of flowers in the environment, falling more sharply with distance when this density is higher (Fig. 4.10C).

This dependence on flower density implies that the area around the nest where flowers have a high probability of being discovered depends on flower density. To estimate the size of this area, I set a threshold at a probability of 10^{-2} per trip (black dotted line in Fig. 4.10C) and computed the "discovery radius" as the distance at which flowers' probability of being discovered remains above this threshold. At low flower densities, the discovery radius reaches 270 meters and is limited by the bees' exploration range (i.e., their tendency to return to the nest after a certain time, even if no flowers have been found). Due to the masking

effect, the discovery radius decreases as flower density increases (Fig. 4.10D).

4.3.2 Flower discovery by a group of bees

I then explored the influence of the masking effect on the total number of flowers discovered by a population of bees (i.e. a colony).

To study this effect, I computed the total number of flowers discovered by a bee colony as a function of density and flower size. I considered a field with flowers of a given size uniformly and randomly distributed with a given flower density, simulated 100 exploration trips and counted the number of flowers that were discovered at least once. When I performed this simulation neglecting the masking effect (i.e., assuming that a bee discovers all the flowers that intersect with its trajectory, not being affected by previous discoveries), I found that the number of flowers discovered increased with flower density and flower size, as these factors make flowers more plentiful and easier to find (Fig. 4.11, dashed lines). However, the masking effect reverses this trend (Fig. 4.11, solid lines): For low densities, the masking effect is weak and the number of discovered flowers increases with density, but at high flower densities, bees become "trapped" around the nest by the flowers immediately surrounding it, which accumulate most of the first visits. Therefore, there is an optimum density that results in the highest number of different flowers discovered. Since the masking effect is stronger for larger flowers, the effect of size is also reversed, and the number of flowers discovered as first in a foraging loop decreases as flower size increases (Fig. 4.11, solid lines).

4.3.3 Relaxing assumptions

These results are robust to the assumptions of the model. Firstly, they do not depend on the precise number of flowers discovered by each bee in each exploration flight. In many natural conditions, bees may need to discover and visit several flowers to fill their nectar crop to capacity before deciding to return to the nest. Taking these multiple discoveries into account (1 to 8 discoveries) leaves results qualitatively unchanged (Fig 4.11B and Fig 4.12). These results are also robust to nectar depletion. Once a flower is visited by a bee, its nectar load may be partially depleted. The next bee visiting the flower may therefore be less inclined to terminate its exploration flight and return to the nest after visiting this flower, so the number of flowers discovered by each bee before returning to the nest may depend on the previous exploration flights performed by other bees in the colony. To account for flower depletion, I ran a simulation in which bees will ignore any

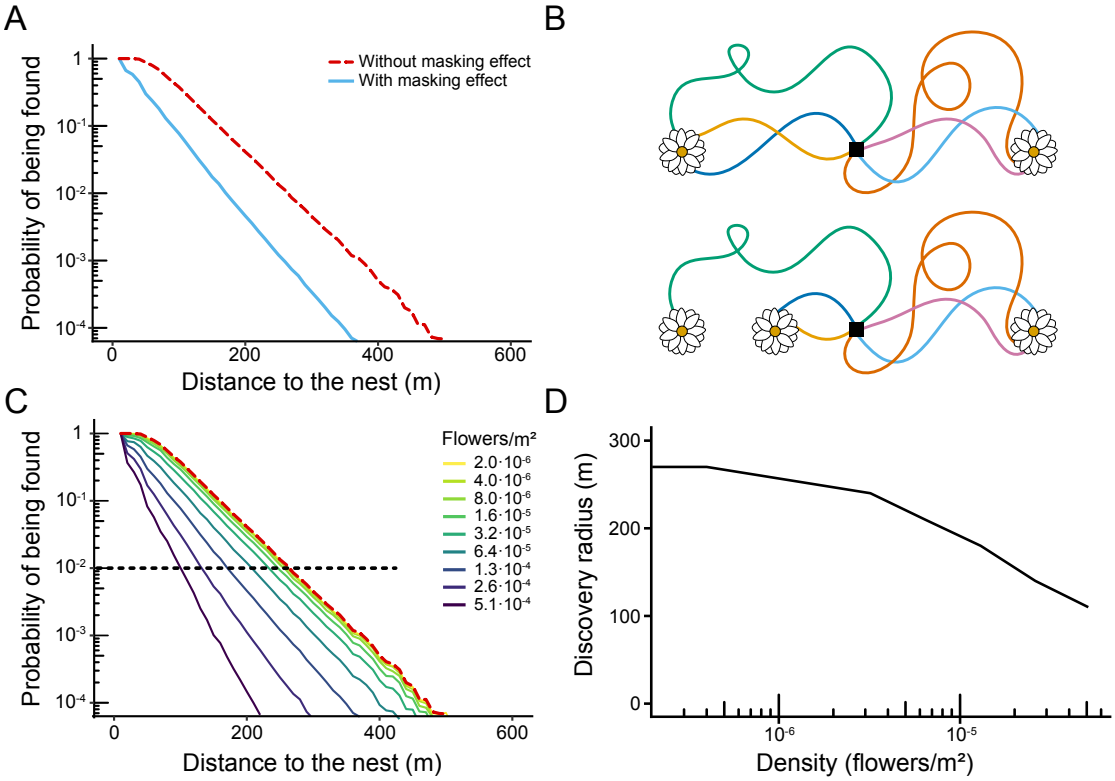

Figure 4.10: Probability that flowers are discovered. (A) Probability that a flower is found as a function of its distance to the nest. We simulated exploration trips in a field of uniformly distributed flowers with density 1.3 10-4 flowers/m2 and flower size 70 cm. For each flower, we computed the probability that it was found in each exploration trip, and we show this probability as a function of the distance between the flower and the nest. Results were computed over 6000 simulated trips of 900s in 10000 environments for each density (negligible bar errors not reported). Red line: Probability calculated without taking into account the masking effect. Blue line: Probability calculated taking into account the masking effect (i.e., only counting the first flower that was discovered in each trip). (B) Illustration of the masking effect. The probability of discovering a flower depends on the presence of other flowers. In a scenario where there are just 2 flowers equidistant to the nest, both flowers should be visited equally (top). However, if another flower is added, it can capture visits that would otherwise visit one of the original flowers (bottom). Black square: nest. (C) Same as (A), but for different flower densities. Red dotted line: Probability calculated without taking into account the masking effect. This probability is independent of the density of flowers. Solid lines: Probability calculated taking the masking effect into account. Black dotted line: threshold probability at which we consider an area that has a high probability of being pollinated. (D) Radius of the area around the nest that has a high probability of being discovered (i.e., where the probability that flowers are discovered is above 10-2) as a function of flower density.

flower that has already been visited in a previous exploration flight. In this case, the maximum at intermediate densities is lost, with higher flower density always leading to more discovered flowers (Fig **??**E). However, for large patch sizes, we know that flower depletion is only partial, and visiting each flower has a cost in terms of time and energy (bees must land on each flower, even if it is depleted). To account for this cost, I limited the total number of depleted flowers a bee will visit before returning to the nest, and in this condition, we again observe a maximum of discovered flowers at intermediate densities (Figure 4.13 A-D).

4.3.4 Conclusion

Simulations showed the importance of taking into account the "masking effect" when estimating the probabilities of discovering flowers. This effect can have a broader influence on bee foraging success and plant pollination since it can impact site fidelity by bees and their tendency to develop traplines to regularly revisit known feeding locations.

According to the simulations, colonies showed a greater tendency to discover more flowers when they were small and in areas with medium flower densities. This finding indicates that there may be an optimal flower size and density that maximizes collective foraging efficiency.

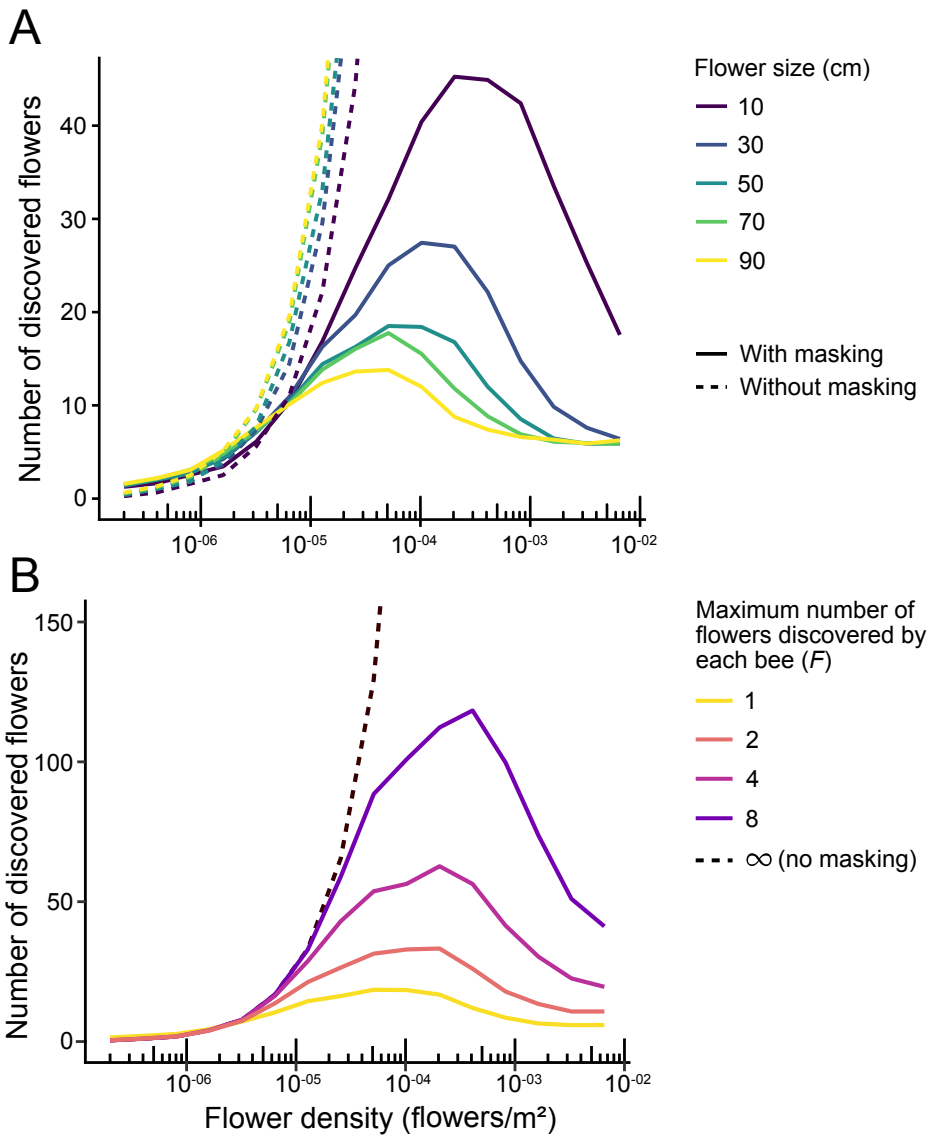

Figure 4.11: Number of different flowers discovered by a group of bees as a function of flower density. (A) Number of different flowers discovered in 100 exploration trips of 900 s, in an environment with randomly distributed flowers. Results are averaged over 80 simulations, keeping the environment fixed for every simulation. Solid lines: Number calculated taking into account the masking effect (i.e., only counting the first flower that was discovered in each trip). Dotted lines: Results without taking into account the masking effect. (B) Same as A but for a given flower size (50 cm), and assuming that each bee will return to the nest only after having discovered a number of flowers (F) (note that box A corresponds to F=1 for the simulations with masking effect, and F=∞ for the simulations without masking effect). Line colors represent the maximum number of flowers discovered by each bee (F).

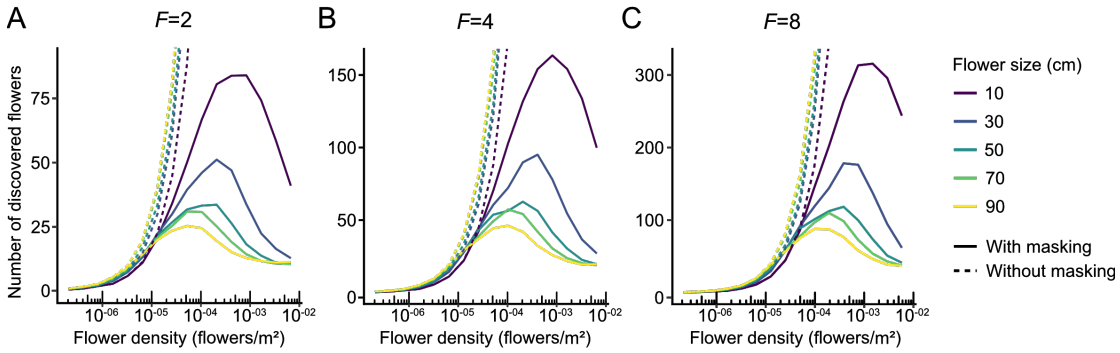

Figure 4.12: Number of different flowers discovered by a group of bees as a function of flower density, when bees discover more than one flower per trip. (A) Number of different flowers discovered in 100 exploration trips of 900 s, in an environment with randomly distributed flowers. Results are averaged over 80 simulations, keeping the environment fixed for every simulation. Solid lines: Number calculated taking into account the masking effect (i.e., only counting the first flowerF=2 flowers that wereas discovered on each trip). Dotted lines: Probability calculated without taking into account the masking effect. (B) Same as (A), but for F=4. (C) Same as (A), but for F=8. Note the difference of scales for the ordinates.

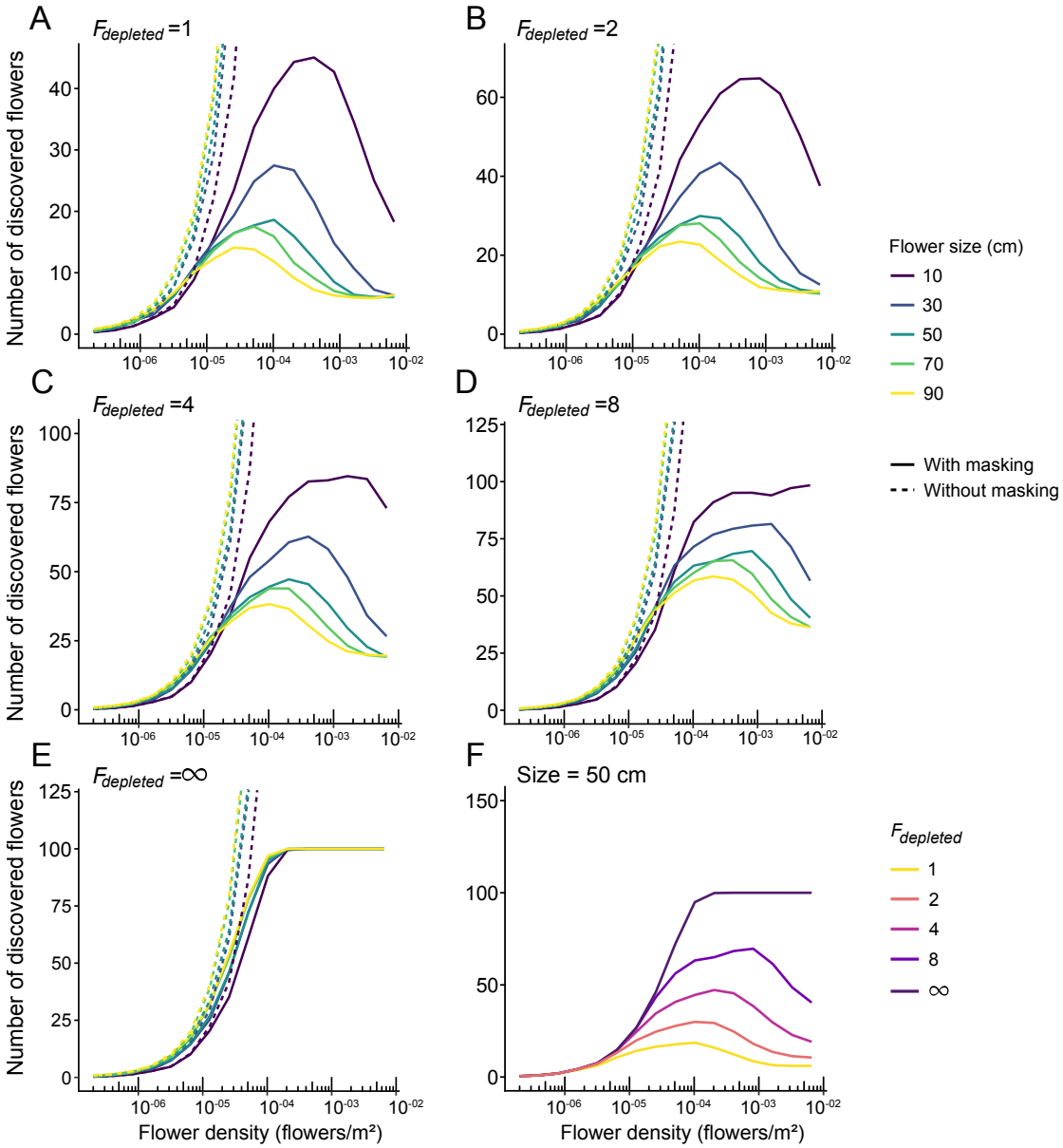

Figure 4.13: Number of flowers discovered by a group of bees as a function of flower density, when accounting for flower depletion. Here we assumed that an individual will continue exploring after visiting an already-explored flower, and will return to the nest only when encountering a fresh flower or after a fixed number of visits to already-visited flowers (Fdepleted). (A) Number of different flowers discovered in 100 exploration trips of 900 s, in an environment with randomly distributed flowers, and with Fdepleted.=1 (note that this value of Fdepleted makes the simulation identical to that in the main text). Results are averaged over 80 simulations, keeping the environment fixed for every simulation. Solid lines: With masking effect. Dotted lines: Without masking effect (the bee does not react to previous flower encounters, and all discovered flowers were counted). Colors correspond to different flower sizes. (B) Same as A but with Fdepleted =2. (C) Same as A but with Fdepleted=4. (D) Same as A but with Fdepleted=8. (E) Same as A but with Fdepleted=∞. (F) Same as A but for a given flower size (50 cm), with colors representing the value of Fdepleted

Chapter 5

Perspectives

Contents

5.1 Foraging exploration

How pollinators search for flowers is of fundamental importance for behavioral research, pollination, and conservation but remains poorly quantified. Here I developed a realistic model of bee search movements based on their observed tendency to make exploratory loops that start and end at their nest location. I used a Persistent Turning Walker model which has inspired some developments in other animals, especially fish (Gupta & Jayannavar, 2022; Kumar *et al.*, 2020; Evans *et al.*, 2020) as well as in robotics (Masó-Puigdellosas *et al.*, 2019) where it has been proven to display better coverage properties than classical random walks (Gupta *et al.*, 2021). This Model, calibrated with real bee behavioral data, produces two-dimensional trajectories with progressive changes of direction driven by the continuous evolution of the angular velocity $\omega(t)$. Using this approach, I documented a neglected yet potentially fundamentally important effect for flower discovery by bees: a perceptual masking effect that influences the probability of bees finding flowers based on not only their size and spatial location but also the presence and characteristics of other flowers around them.

Previous models assume that bees explore their environment randomly using Lévy flights or other diffusive processes (Vallaeys *et al.*, 2017; Reynolds *et al.*, 2013; Dubois *et al.*, 2021). In a diffusive model, individuals can wander away from the nest indefinitely if given enough time. In contrast to these models, our model replicates looping trajectories observed in real bees (Woodgate *et al.*, 2016; Osborne *et al.*, 2013), which confines the presence of individuals around a nest. As a consequence of the periodic returns of bees to the nest, their distribution becomes independent of the time given to explore. This result has the important consequence that, under the assumptions of The Model, longer simulation duration results in a more thorough exploitation of the foraging area around the nest, but not in a larger area.

By explicitly simulating individual trajectories in complex environments, The Model revealed how the presence of a flower may decrease the probability of discovering another, a phenomenon that I call the "perceptual masking effect". Although in this study I strictly focused on flower discovery during exploration flights, results suggest that the masking effect can have broader influences on exploitation patterns, bee foraging success, and plant pollination. Indeed, at the individual level, flower discovery can impact site fidelity by bees and their tendency to develop traplines to regularly revisit known feeding locations (Lihoreau *et al.*, 2012b). Given the perceptual masking effect, flower patches may not be discovered with

the same probability or in the same order, which may impact their likelihood to be exploited and the ability of bees to develop efficient traplines minimizing overall travel distances (Reynolds *et al.*, 2013). The masking effect may also influence the global foraging success of the colony, which depends on the number of flowers discovered collectively by all the bees of a colony because a flower discovered and exploited by a bee will be at least partially depleted, giving marginal benefit to later visitors. For this reason, what counts is not the total number of visits that bees perform, but rather the total number of different flowers discovered by the colony. In the simulations, colonies tended to find more flowers when they were small and at medium densities. This suggests that there is an optimal flower size and density at which collective foraging efficiency is optimized (although the effect of size on foraging efficiency will be compounded with the greater reward provided by bigger flowers on average).

Similar extrapolations of our results on first flower discovery can be made regarding pollination. At the plant level, we found that flowers distant from the nest were more often visited in low-density environments. Since bees disseminate pollen (and thus mediate plant reproduction) when visiting flowers, this may generate lower probabilities of pollination at high flower densities, by which the area that is pollinated around the nest decreases as the density of flowers increases. If this prediction is verified in future studies, this would mean that the overall distribution of flower patches directly impacts their pollination and should be taken into account when designing strategies for crop production and assisted pollination.

This search model is a scaffold for future quantitative characterization of the movement of bees, or any central place forager, across time and landscapes. Although I limited our study to flower discovery probability, and therefore only provided predictions for first flower discovery, the model could be used to investigate the full foraging trips of bees, and how they change through time as bees acquire experience with their environment and develop spatial memories (Collett *et al.*, 2013). It would be particularly interesting to integrate this exploration model into existing learning exploitation models proposed to replicate route formation by bees (Lihoreau *et al.*, 2012a; Reynolds *et al.*, 2013; Dubois *et al.*, 2021) and to study the dynamics of resource exploitation by populations of bees (Becher *et al.*, 2016). Once a flower is discovered, its location can be learned, and new exploration may start from it, ultimately allowing for the establishment of traplines. This would be modeled via a modification of the attraction component, which can be modified to point toward previously-discovered flowers instead of the nest. Importantly, model predictions (flower discovery probability, visitation order, flight

trajectories) can be experimentally tested and the model calibrated for specific study species. This will facilitate the improvement and validation of potential applications. As discussed above, robust predictive models of bee movements including both exploration and exploitation would be particularly useful for improving precision pollination (to maximize crop pollination), pollinator conservation (to ensure population growth and maintenance), but also in ecotoxicology (to avoid exposure of bees to agrochemicals) and legislation (to avoid unwanted gene flow between plants). Beyond pollinators, our minimal persistent turning walker model could be calibrated to apply to a wide range of species, providing a data-based quantification and predictions for further exploration of the broader interactions between central place foraging animals and their environment.

5.2 Beyond exploration: Foraging optimization

A century ago, pioneers in animal cognition, such as Charles H. Turner and Karl von Frisch (Von Frisch, 1965; Dona & Chittka, 2020), ignited a renewed interest in studying the cognitive abilities of bees. Subsequent research has demonstrated that bees possess a range of cognitive abilities, including olfactory, visual, and tactile learning (Schubert *et al.*, 2002), path integration (Collett & Collett, 2000), social learning (Leadbeater & Chittka, 2007), concept learning (Avarguès-Weber & Giurfa, 2013), numerosity (Chittka & Geiger, 1995; Dacke & Srinivasan, 2008), and metacognition (Perry & Barron, 2013). Due to their ease of maintenance and manipulation, abundance, adaptability to new foraging tasks, and constant foraging behavior, honey bees and bumblebees are popular choices for studying insect foraging behavior

Bees exhibit a foraging behavior that involves visiting a sequence of flower patches spread out in their surroundings and collecting nectar from a certain number of flowers during each visit. Studies have revealed that bees tend to follow specific rules when visiting a patch of flowers. They move between flowers by selecting the nearest unvisited flower and often maintain their direction for several subsequent movements (Ohashi *et al.*, 2007; Saleh & Chittka, 2007; Pyke, 1978) . Bees also decide when to leave a patch and move on to another one. These foraging behaviors, which are guided by innate rules, represent simple cognitive strategies for maximizing foraging efficiency.

One interesting behavior observed during the study of bee foraging behavior is the establishment of traplines (Ohashi *et al.*, 2007; Lihoreau *et al.*, 2012b). Traplines are stable routes that bees establish between the same patches of flowers,

revisiting them in a consistent order during foraging expeditions. Recent research has shown that bees establish traplines by minimizing the total distance traveled, selecting the shortest path from their nest to all the flowers and back to their nest (Lihoreau *et al.*, 2012a,b). The process of finding the shortest path between a set of points is analogous to the "Traveling Salesman Problem," a math problem that requires testing all options to determine the optimal solution.

Only a few models have been developed to explore how foraging strategies can arise (Reynolds *et al.*, 2013; Dubois *et al.*, 2021). Their models suggest that bees compare multi-leg routes to find the shortest path. In summary, in their models, a foraging *bout* for a bee comprises a sequence of movements and flower visits starting from the moment it leaves the nest until it returns to it. In both models, bees jump from flower to flower with a probability based on a matrix of transition probabilities. The learning across successive bouts is made by adjusting the transition probabilities.

In Reynolds *et al.* (2013), this adjustment is made at the end of each bout by considering the total distance traveled. By contrast, in Dubois *et al.* (2021), the probability to move between each flower is adjusted "on line" during the bout and it doesn't depend on the distance itself but rather on the fact that a transition from one flower to another result in the bee receiving positive or negative reinforcements after finding a rewarding flower or an unrewarding one, respectively (note that each flower can only be visited successfully once per bout). Figure 5.1 shows a flowchart of their model.

Two metrics are used by Dubois *et al.* (2021) to compare the output of their model with experimental data collected by Lihoreau *et al.* (2012b) In this study, the authors gathered the visitation sequence of bumblebees in a pentagon array of artificial flowers. The closest distance between flowers was 50m. (Fig 5.2).

The two metrics are the quality of the route (QL) and a similarity index (SI) between flower visitations sequences. The quality is calculated with

$$QL = \frac{\frac{F^2}{d}}{QL_{opt}} \tag{5.1}$$

where F is the number of rewarding flowers visited during a foraging bout and d is the net length of all transition movement traveled during the foraging bout. This quality is standardized between 0 and 1 by the quality of the optimal route QL_{opt} (this is, the shortest possible route to visit all 5 flowers)

The similarity index between two consecutive foraging bouts a and b is calculated with

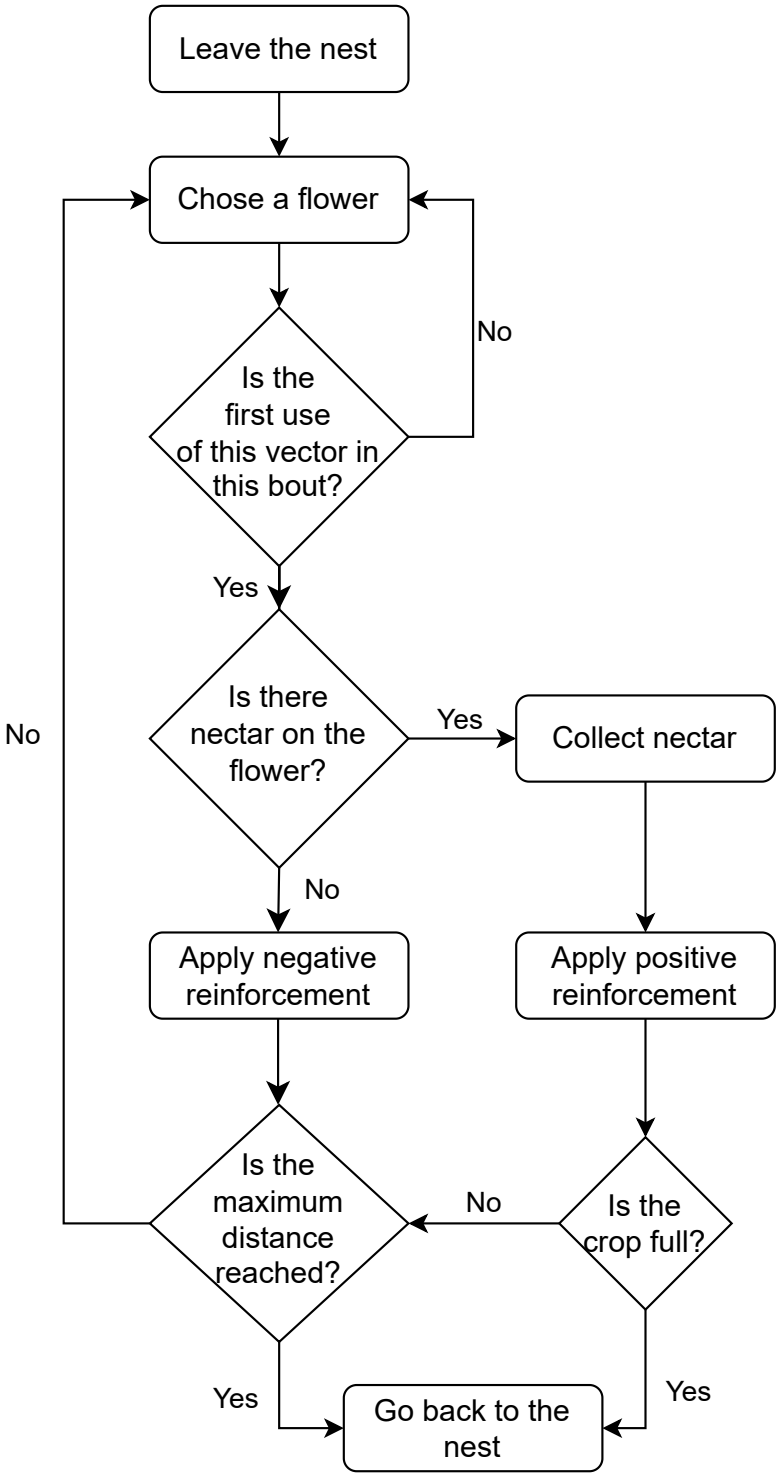

Figure 5.1: **Flowchart summarizing the model presented in Dubois *et al.* (2021)** Rectangles represent actions performed by a bee. Diamonds indicate conditional statements.

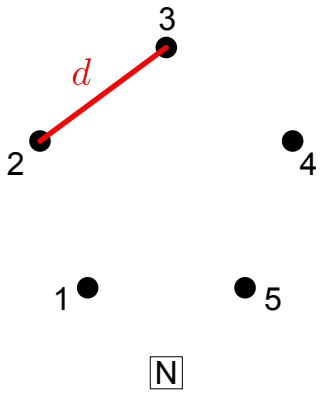

Figure 5.2: **Spatial configuration of artificial flowers** In Lihoreau *et al.* (2012b) (big pentagone) $d = 50$. In Lihoreau *et al.* (2011) (small pentagone), $d = 4$.

$$SI_{ab} = \frac{s_{ab}}{2l_{ab}} \tag{5.2}$$

where s_ab represents the number of flowers in transitions found in both sequences and l_{ab} the length of the longest flower visitation sequence between i and j multiplied by 2 to make sure that $SI_{ab} = 1$occurs only when two consecutive sequences sharing the same transitions also have the same length. (see Dubois *et al.* (2021) for more details)

The predictions made by their models closely resemble the experimental data of (Lihoreau *et al.*, 2012b) (Big pentagon, Fig 5.3). However, it doesn't fit the experimental data of similar studies where the distance between flowers was smaller. This is the case for data collected by (Lihoreau *et al.*, 2011) (Small pentagon, Fig 5.4). In their study, Lihoreau *et al.* (2011) collected visitations sequences in a similar pentagon flower array where the closest distance between flowers was 4m.

As I showed in Chapter 4, the probability of discovering flowers depends on the scale of the environment and the relative position of the flowers.

In Dubois *et al.* (2021), the authors use an initial distance-based probability matrix that was chosen to be inversely proportional to the square distance between each element.

The initial matrix is obtained using the coordinates of all flowers and the nest to compute the distance between each pair of entities. Then, the probability P to go from a flower i to a flower j is determined by :

$$P(i \to j) = \frac{\frac{1}{d_{ij}^n}}{\sum_j \frac{1}{d_{ij}^n}} \tag{5.3}$$

where d_{ij} is the distance between the flower i and the flower j and n an expo-

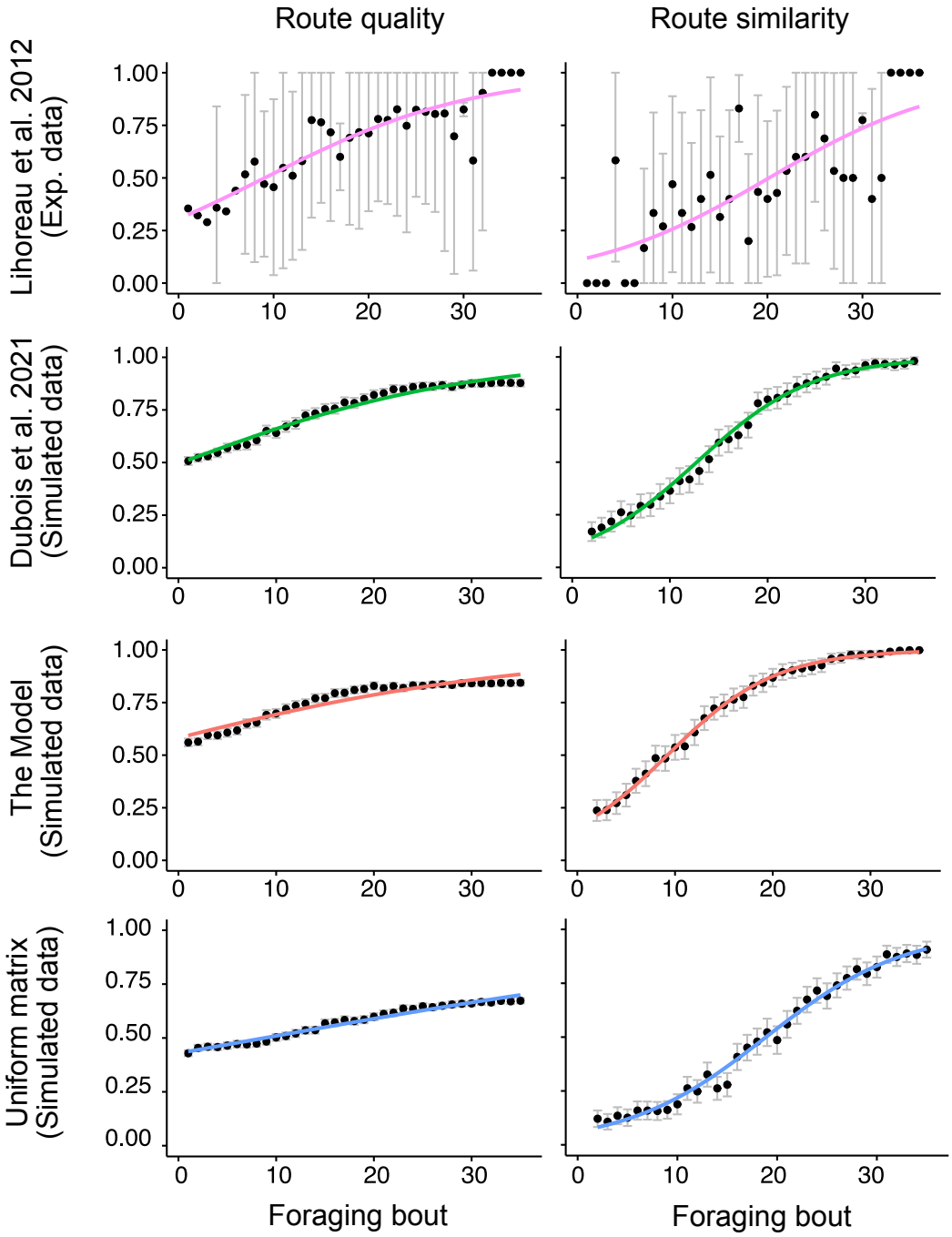

Figure 5.3: **Comparisons of experimental and simulated route qualities.** Comparisons of route qualities and route similarities between simulations and experimental data

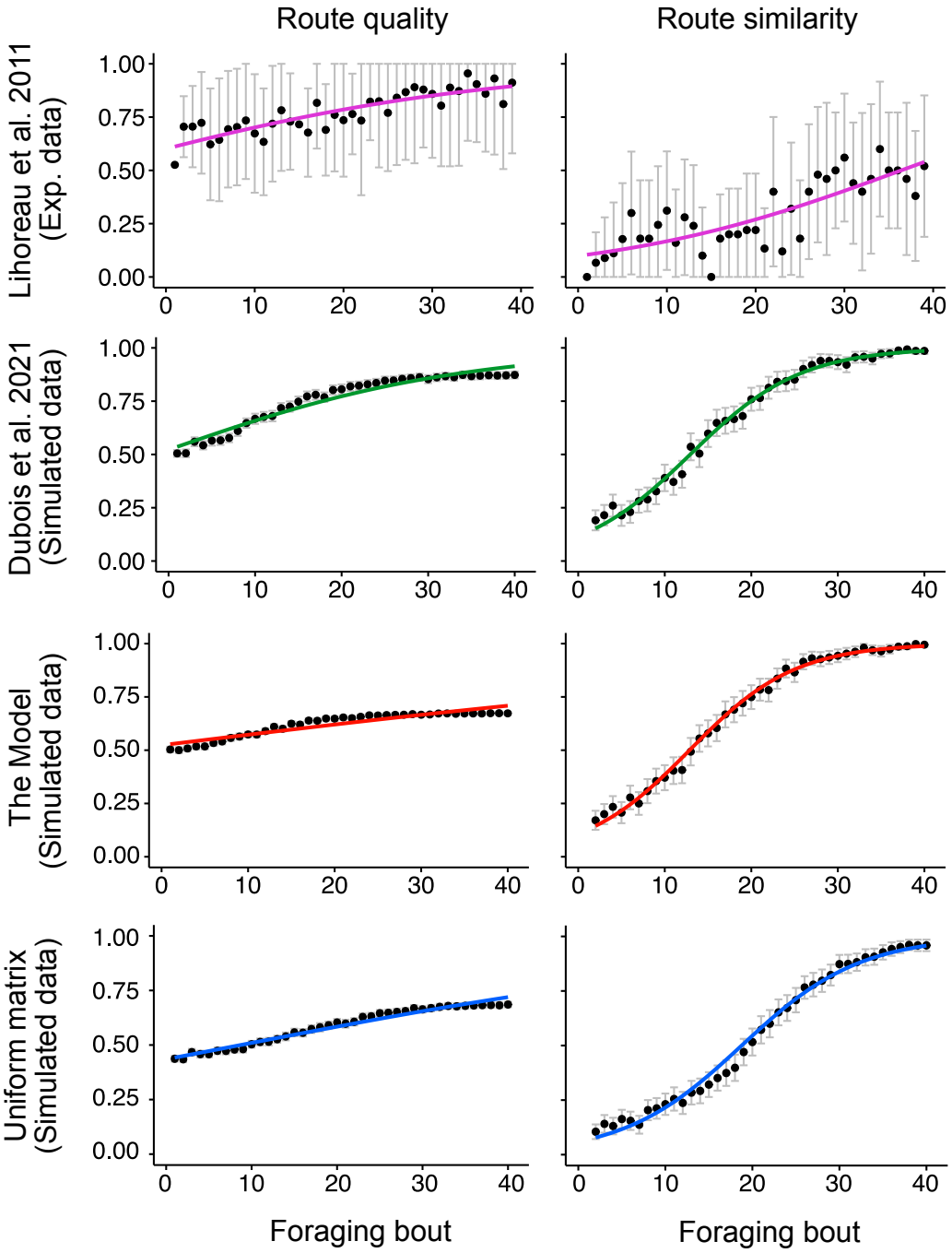

Figure 5.4: **Comparisons of experimental and simulated route qualities.** Comparisons of route qualities and route similarities between simulations and experimental data

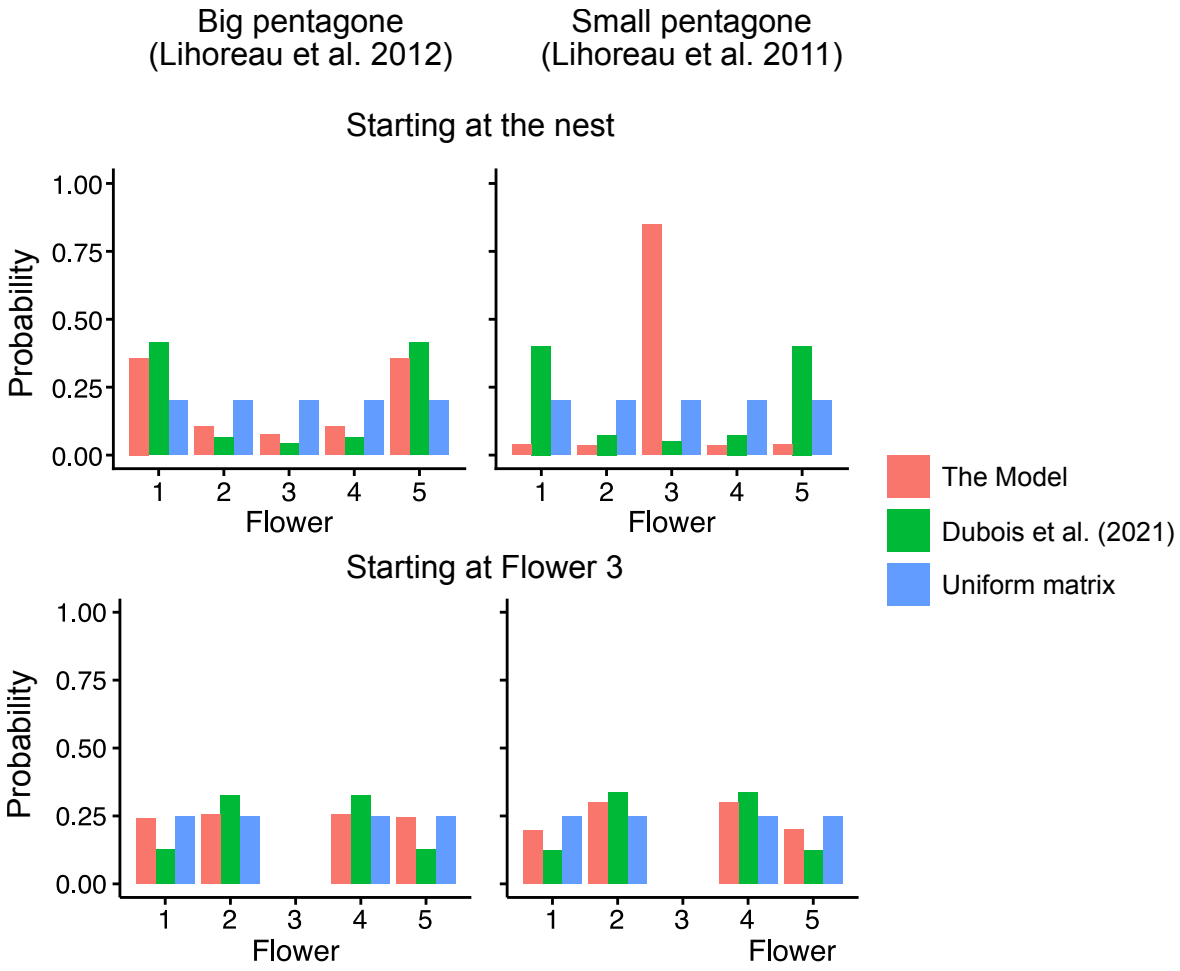

Figure 5.5: Probability of visiting each flower when starting at the nest for the different initial matrix.

nent arbitrarily chose. In their study $n = 2$. This design was chosen because it approximated closely the probabilities to find the flowers using a simple random walk (Dubois *et al.*, 2021).

To enlighten the importance of designing the initial matrix (when the bee is still in exploratory mode) on the learning curve, I tested two additional cases: a uniform matrix, and the matrix predicted by the Model (with the parameters calibrated as in Chapter 3).

To develop the latter one, I simulated 100000 trajectories from the nest and counted how many flowers were discovered on each path. From there, I determined the probability of encountering each flower when beginning at the nest. I then replicated this procedure for trajectories originating at each individual flower.

First, we will take a look at how the initial matrix differs between Dubois *et al.* (2021) and the Model (Fig. 5.5). At a big scale, in the big pentagon, the initial matrix calculated from The Model is close to the one from Dubois *et al.*

(2021) when starting from the nest, but the central place foraging component of the model has a visible effect on these statistics when starting from a flower (e.g. Flower 3) such that they become independent of the distance. At this scale, the masking effect is not very visible because there are few and small flowers. For the matrix calculated from The Model in the small pentagon, we can clearly see the effect of the initial direction. It's a ballistic regime rather than a diffusive one.

As far as the similarity is concerned, the learning algorithm of Dubois *et al.* (2021) converges always to a high similarity, independently of the initial matrix. This corresponds to the dynamics experimentally observed in the big pentagon (Fig 5.3) but not for the smaller one (Fig 5.4).

As far as the progression of route quality is concerned, the Model matrix yields the same results as Dubois *et al.* (2021) for the large pentagon (Fig 5.3), but yields lower quality in the small one (Fig 5.4). The ballistic bias towards the farthest flower is the worst initial condition for the learning of an optimized foraging route, and indeed, this bias will stabilize and will lead to a sub-optimal solution.

For comparison, I tested the same learning algorithm with the uniform initial matrix (the probability to go to each flower is the same). At the large scale, the routes converge towards lower qualities than for Dubois *et al.* (2021) and the Model. At the small scale, this uniform matrix produces routes of similarly low quality as the Model-matrix predictions.

To sum up, in this preliminary attempt to combine the Model in the learning context, we observe that the initial matrix with a bias toward the closest flowers (1 and 5) allows to produce visitation sequence with high quality. Without this initial bias, the final quality is lower (Fig 5.3). To further the analysis, the comparison between the three cases should be pursued at the detailed level of how routes evolve with time.

5.3 Open questions

I enunciated an exploration model calibrated with one set of experimental data. The Model could also be confronted with more experimental data, especially on statistics of first flower discovery.

For now, the Model is only able to replicate the exploration part of the foraging behavior. However, it should be possible to add more behavioral components to take into account learning, motivation, or the effect of the natural landscapes on the navigation (Menzel *et al.*, 2019).

I propose a behavioral exploration model that is governed by a function of the

target angular speed ω^*, which operates in an "all or nothing" fashion. Specifically, the model switches from 0 when homing is not activated to η^* when it is activated, with the homing vector $\vec{H}$ dictating the direction of movement. To integrate this movement model into a learning framework, such as the one presented in Dubois *et al.* (2021), a temporal sequence of targeted angular speeds is required. One possible hypothesis is that the learning of routes involves the acquisition of a succession of visual snapshots. Accordingly, we can associate a "direction-controlling" vector with each snapshot and update the vector that governs ω^* whenever a familiar snapshot is encountered. This presents an opportunity to establish links with underlying neural structures that have been previously modeled for snapshot learning (Ardin *et al.*, 2016; Müller *et al.*, 2018).

Cognitive components at larger time scales can be regulated by the parameter α, which determines the duration before the "return phase" in each loop. In an exploration context, we can imagine that α starts with a low value that encourages explorations around the nest. If the search is unsuccessful, α can increase with time. Reynolds *et al.* (2007b) describe such a behavior and involve a "Lévy flight" type of movement in a context of optimizing discoveries. Since their bees actually do loops, "Lévy flight" is not really appropriate, but the concept of "Lévy loops" could emerge to denote such behaviors. On the other hand, if the search is successful and energy is obtained, α could adopt higher values to reflect the availability of energy. As such, α may serve as an indicator of both cognitive (i.e., exploring farther due to lack of findings) and physiological (i.e., energy status) components.

The parameter $\eta(t)$ representing the force of the attraction could also be generalized to produce different types of trajectories. Here, I decided to produce elongated loops by switching the values of $\eta(t)$ from 0 to η^* but it is possible to have a constant value that produces circular trajectories. It is also possible to reflect a "motivation component" in the sense that high values produce straight movements toward the target. This could mean that we can have low values during the exploration phases and higher ones during the exploitation of resources.

Chapter 6

Bibliography

Allen, A.M. & Singh, N.J. (2016). Linking Movement Ecology with Wildlife Management and Conservation. *Frontiers in Ecology and Evolution*, 3.

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, e1004683. Publisher: Public Library of Science.

Atkinson, R.P.D., Rhodes, C.J., Macdonald, D.W. & Anderson, R.M. (2002). Scale-free dynamics in the movement patterns of jackals. *Oikos*, 98, 134–140. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1034/j.1600-0706.2002.980114.x.

Avarguès-Weber, A. & Giurfa, M. (2013). Conceptual learning by miniature brains. *Proceedings of the Royal Society B: Biological Sciences*, 280, 20131907. Publisher: Royal Society.

Bartumeus, F., Peters, F., Pueyo, S., Marrasé, C. & Catalan, J. (2003). Helical Lévy walks: Adjusting searching statistics to resource availability in microzooplankton. *Proceedings of the National Academy of Sciences*, 100, 12771–12775.

Becher, M.A., Grimm, V., Knapp, J., Horn, J., Twiston-Davies, G. & Osborne, J.L. (2016). BEESCOUT: A model of bee scouting behaviour and a software tool for characterizing nectar/pollen landscapes for BEEHAVE. *Ecological Modelling*, 340, 126–133.

Benhamou, S. (2007). How Many Animals Really Do the Lévy Walk? *Ecology*, 88, 1962–1969. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1890/06-1769.1.

Bergman, C.M., Schaefer, J.A. & Luttich, S.N. (2000). Caribou movement as a correlated random walk. *Oecologia*, 123, 364–374.

Brebner, J.S., Makinson, J.C., Bates, O.K., Rossi, N., Lim, K.S., Dubois, T., Gómez-Moracho, T., Lihoreau, M., Chittka, L. & Woodgate, J.L. (2021). Bumble bees strategically use ground level linear features in navigation. *Animal Behaviour*, 179, 147–160.

Brown, M.J. & Paxton, R.J. (2009). The conservation of bees: a global perspective. *Apidologie*, 40, 410–416.

Buchmann, S.L. & Nabhan, G.P. (1996). The Pollination Crisis. *The Sciences*, 36, 22–27. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.2326-1951.1996.tb03254.x.

Capaldi, E.A., Smith, A.D., Osborne, J.L., Fahrbach, S.E., Farris, S.M., Reynolds, D.R., Edwards, A.S., Martin, A., Robinson, G.E., Poppy, G.M. & Riley, J.R. (2000). Ontogeny of orientation flight in the honeybee revealed by harmonic radar. *Nature*, 403, 537–540. Number: 6769 Publisher: Nature Publishing Group.

Caswell, H. (1988). Theory and models in ecology: A different perspective. *Ecological Modelling*, 43, 33–44.

Cattiaux, P., Chafai, D. & Motsch, S. (2010). Asymptotic analysis and diffusion limit of the Persistent Turning Walker Model. *Asymptotic Analysis*, 67, 17–31. Publisher: IOS Press.

Chittka, L. & Geiger, K. (1995). Can honey bees count landmarks? *Animal Behaviour*, 49, 159–164.

Clement, L., Schwarz, S. & Wystrach, A. (2023). An intrinsic oscillator underlies visual navigation in ants. *Current Biology*, 33, 411–422.e5.

Codling, E.A., Plank, M.J. & Benhamou, S. (2008). Random walk models in biology. *Journal of The Royal Society Interface*, 5, 813–834. Publisher: Royal Society.

Collett, M., Chittka, L. & Collett, T.S. (2013). Spatial Memory in Insect Navigation. *Current Biology*, 23, R789–R800.

Collett, M. & Collett, T.S. (2000). How do insects use path integration for their navigation? *Biological Cybernetics*, 83, 245–259.

Couzin, I.D., Krause, J., Franks, N.R. & Levin, S.A. (2005). Effective leadership and decision-making in animal groups on the move. *Nature*, 433, 513–516. Number: 7025 Publisher: Nature Publishing Group.

Couzin, I.D., Krause, J., James, R., Ruxton, G.D. & Franks, N.R. (2002). Collective Memory and Spatial Sorting in Animal Groups. *Journal of Theoretical Biology*, 218, 1–11.

Dacke, M. & Srinivasan, M.V. (2008). Evidence for counting in insects. *Animal Cognition*, 11, 683–689.

Daltorio, K.A., Tietz, B.R., Bender, J.A., Webster, V.A., Szczecinski, N.S., Branicky, M.S., Ritzmann, R.E. & Quinn, R.D. (2013). A model of exploration and goal-searching in the cockroach, *Blaberus discoidalis*. *Adaptive Behavior*, 21, 404–420.

Degond, P. & Motsch, S. (2008). Large Scale Dynamics of the Persistent Turning Walker Model of Fish Behavior. *Journal of Statistical Physics*, 131, 989–1021.

Dona, H.S.G. & Chittka, L. (2020). Charles H. Turner, pioneer in animal cognition. *Science*, 370, 530–531. Publisher: American Association for the Advancement of Science.

Du, G., Kumari, S., Ye, F. & Podgornik, R. (2021). Model of metameric locomotion in smooth active directional filaments with curvature fluctuations. *Europhysics Letters*, 136, 58003. Publisher: IOP Publishing.

Dubois, T., Pasquaretta, C., Barron, A.B., Gautrais, J. & Lihoreau, M. (2021). A model of resource partitioning between foraging bees based on learning. *PLOS Computational Biology*, 17, e1009260. Publisher: Public Library of Science.

Edwards, A.M., Phillips, R.A., Watkins, N.W., Freeman, M.P., Murphy, E.J., Afanasyev, V., Buldyrev, S.V., da Luz, M.G.E., Raposo, E.P., Stanley, H.E. & Viswanathan, G.M. (2007). Revisiting Lévy flight search patterns of wandering albatrosses, bumblebees and deer. *Nature*, 449, 1044–1048. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 7165 Primary_atype: Research Publisher: Nature Publishing Group.

Evans, M.R., Majumdar, S.N. & Schehr, G. (2020). Stochastic resetting and applications. *Journal of Physics A: Mathematical and Theoretical*, 53. _eprint: 1910.07993.

Gautrais, J., Jost, C., Soria, M., Campo, A., Motsch, S., Fournier, R., Blanco, S. & Theraulaz, G. (2009). Analyzing fish movement as a persistent turning walker. *Journal of Mathematical Biology*, 58, 429–445.

Gautrais, J., Jost, C. & Theraulaz, G. (2008). Key Behavioural Factors in a Self-Organised Fish School Model. *Annales Zoologici Fennici*, 45, 415–428. Publisher: Finnish Zoological and Botanical Publishing Board.

Goulson, D., Nicholls, E., Botías, C. & Rotheray, E.L. (2015). Bee declines driven by combined stress from parasites, pesticides, and lack of flowers. *Science*, 347, 1255957.

Gupta, D., Pal, A. & Kundu, A. (2021). Resetting with stochastic return through linear confining potential. *Journal of Statistical Mechanics: Theory and Experiment*, 2021, 043202. _eprint: 2012.12878.

Gupta, S. & Jayannavar, A.M. (2022). Stochastic Resetting: A (Very) Brief Review. *Frontiers in Physics*, 10, 1–16. _eprint: 2106.07693.

Joo, R., Picardi, S., Boone, M.E., Clay, T.A., Patrick, S.C., Romero-Romero, V.S. & Basille, M. (2022). Recent trends in movement ecology of animals and human mobility. *Movement Ecology*, 10, 26.

Kadota, M., Torisawa, S., Takagi, T. & Komeyama, K. (2011). Analysis of juvenile tuna movements as correlated random walk. *Fisheries Science*, 77, 993–998.

Kapustjansky, A., Chittka, L. & Spaethe, J. (2010). Bees use three-dimensional information to improve target detection. *Naturwissenschaften*, 97, 229–233.

Kareiva, P.M. & Shigesada, N. (1983). Analyzing insect movement as a correlated random walk. *Oecologia*, 56, 234–238.

Khuong, A., Lecheval, V., Fournier, R., Blanco, S., Weitz, S., Bezian, J.J. & Gautrais, J. (2013). How Do Ants Make Sense of Gravity? A Boltzmann Walker Analysis of Lasius niger Trajectories on Various Inclines. *PLoS ONE*, 8, e76531.

Klein, A.M., Vaissière, B.E., Cane, J.H., Steffan-Dewenter, I., Cunningham, S.A., Kremen, C. & Tscharntke, T. (2007). Importance of pollinators in changing landscapes for world crops. *Proceedings of the Royal Society B: Biological Sciences*, 274, 303–313.

Kumar, V., Sadekar, O. & Basu, U. (2020). Active Brownian motion in two dimensions under stochastic resetting. *Physical Review E*, 102, 1–19. Publisher: American Physical Society _eprint: 2008.03294.

Leadbeater, E. & Chittka, L. (2007). Social Learning in Insects — From Miniature Brains to Consensus Building. *Current Biology*, 17, R703–R713.

Lenz, F., Chechkin, A.V. & Klages, R. (2013). Constructing a Stochastic Model of Bumblebee Flights from Experimental Data. *PLoS ONE*, 8, e59036.

Lihoreau, M., Chittka, L., Le Comber, S.C. & Raine, N.E. (2012a). Bees do not use nearest-neighbour rules for optimization of multi-location routes. *Biology Letters*, 8, 13–16.

Lihoreau, M., Chittka, L. & Raine, N. (2010). Travel Optimization by Foraging Bumblebees through Readjustments of Traplines after Discovery of New Feeding Locations. *The American Naturalist*, 176, 744–757.

Lihoreau, M., Chittka, L. & Raine, N.E. (2011). Trade-off between travel distance and prioritization of high-reward sites in traplining bumblebees: Distance reward trade-off in bees. *Functional Ecology*, 25, 1284–1292.

Lihoreau, M., Chittka, L. & Raine, N.E. (2016). Monitoring Flower Visitation Networks and Interactions between Pairs of Bumble Bees in a Large Outdoor Flight Cage. *PLOS ONE*, 11, e0150844.

Lihoreau, M., Raine, N.E., Reynolds, A.M., Stelzer, R.J., Lim, K.S., Smith, A.D., Osborne, J.L. & Chittka, L. (2012b). Radar Tracking and Motion-Sensitive Cameras on Flowers Reveal the Development of Pollinator Multi-Destination Routes over Large Spatial Scales. *PLoS Biology*, 10, e1001392.

MacQueen, S.A., Braun, W.J. & Tyson, R.C. (2022). Memory-guided foraging and landscape design interact to determine ecosystem services. *Journal of Theoretical Biology*, 534, 110958.

Masó-Puigdellosas, A., Campos, D. & Méndez, V. (2019). Transport properties and first-arrival statistics of random motion with stochastic reset times. *Physical Review E*, 99, 1–9. _eprint: 1810.12798.

Mayer, C., Adler, L., Armbruster, W.S., Dafni, A., Eardley, C., Huang, S.Q., Kevan, P.G., Ollerton, J., Packer, L., Ssymank, A., Stout, J.C. & Potts, S.G. (2011). Pollination ecology in the 21st Century: Key questions for future research. *Journal of Pollination Ecology*, pp. 8–23.

Menzel, R., Lehmann, K., Manz, G., Fuchs, J., Koblofsky, M. & Greggers, U. (2012). Vector integration and novel shortcutting in honeybee navigation. *Apidologie*, 43, 229–243.

Menzel, R., Tison, L., Fischer-Nakai, J., Cheeseman, J., Balbuena, M.S., Chen, X., Landgraf, T., Petrasch, J., Polster, J. & Greggers, U. (2019). Guidance of Navigating Honeybees by Learned Elongated Ground Structures. *Frontiers in Behavioral Neuroscience*, 12.

Morán, A., Lihoreau, M., Escudero, A.P. & Gautrais, J. (2023). Modeling bee movement shows how a perceptual masking effect can influence flower discovery. *PLOS Computational Biology*, 19, e1010558. Publisher: Public Library of Science.

Mårell, A., Ball, J.P. & Hofgaard, A. (2002). Foraging and movement paths of female reindeer: insights from fractal analysis, correlated random walks, and Lévy flights. *Canadian Journal of Zoology*, 80, 854–865. Publisher: NRC Research Press.

Müller, J., Nawrot, M., Menzel, R. & Landgraf, T. (2018). A neural network model for familiarity and context learning during honeybee foraging flights. *Biological Cybernetics*, 112, 113–126.

Nathan, R., Getz, W.M., Revilla, E., Holyoak, M., Kadmon, R., Saltz, D. & Smouse, P.E. (2008). A movement ecology paradigm for unifying organismal movement research. *Proceedings of the National Academy of Sciences*, 105, 19052–19059. Publisher: Proceedings of the National Academy of Sciences.

Noetel, J., Freitas, V.L.S., Macau, E.E.N. & Schimansky-Geier, L. (2018). Search and return model for stochastic path integrators. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28, 106302.

Nötel, J., Sokolov, I.M. & Schimansky-Geier, L. (2017). Diffusion of active particles with stochastic torques modeled as α -stable noise. *Journal of Physics A: Mathematical and Theoretical*, 50, 034003.

Ohashi, K., Leslie, A. & Thomson, J.D. (2008). Trapline foraging by bumble bees: V. Effects of experience and priority on competitive performance. *Behavioral Ecology*, 19, 936–948.

Ohashi, K., Thomson, J.D. & D'Souza, D. (2007). Trapline foraging by bumble bees: IV. Optimization of route geometry in the absence of competition. *Behavioral Ecology*, 18, 1–11.

Osborne, J.L., Smith, A., Clark, S.J., Reynolds, D.R., Barron, M.C., Lim, K.S. & Reynolds, A.M. (2013). The Ontogeny of Bumblebee Flight Trajectories: From Naïve Explorers to Experienced Foragers. *PLoS ONE*, 8, e78681.

Perry, C.J. & Barron, A.B. (2013). Honey bees selectively avoid difficult choices. *Proceedings of the National Academy of Sciences*, 110, 19155–19159. Publisher: Proceedings of the National Academy of Sciences.

Peterson, S.C. & Noble, P.B. (1972). A Two-Dimensional Random-Walk Analysis of Human Granulocyte Movement. *Biophysical Journal*, 12, 1048–1055.

Potts, S.G., Biesmeijer, J.C., Kremen, C., Neumann, P., Schweiger, O. & Kunin, W.E. (2010). Global pollinator declines: trends, impacts and drivers. *Trends in Ecology & Evolution*, 25, 345–353.

Pyke, G.H. (1978). Optimal foraging: Movement patterns of bumblebees between inflorescences. *Theoretical Population Biology*, 13, 72–98.

Pyke, G.H. (2015). Understanding movements of organisms: it's time to abandon the Lévy foraging hypothesis. *Methods in Ecology and Evolution*, 6, 1–16.

Pyke, G.H. & Cartar, R.V. (1992). The Flight Directionality of Bumblebees: Do They Remember Where They Came from? *Oikos*, 65, 321–327. Publisher: [Nordic Society Oikos, Wiley].

Ramos-Fernández, G., Mateos, J.L., Miramontes, O., Cocho, G., Larralde, H. & Ayala-Orozco, B. (2004). Lévy walk patterns in the foraging movements of spider monkeys (Ateles geoffroyi). *Behavioral Ecology and Sociobiology*, 55, 223–230.

Reynolds, A.M., Lihoreau, M. & Chittka, L. (2013). A Simple Iterative Model Accurately Captures Complex Trapline Formation by Bumblebees Across Spatial Scales and Flower Arrangements. *PLoS Computational Biology*, 9, e1002938.

Reynolds, A.M., Smith, A.D., Menzel, R., Greggers, U., Reynolds, D.R. & Riley, J.R. (2007a). Displaced honey bees perform optimal scale-free search flights. *Ecology*, 88, 1955–1961.

Reynolds, A.M., Smith, A.D., Reynolds, D.R., Carreck, N.L. & Osborne, J.L. (2007b). Honeybees perform optimal scale-free searching flights when attempting to locate a food source. *Journal of Experimental Biology*, 210, 3763–3770.

Reynolds, A.M., Sword, G.A., Simpson, S.J. & Reynolds, D.R. (2009). Predator Percolation, Insect Outbreaks, and Phase Polyphenism. *Current Biology*, 19, 20–24.

Riley, J.R., Smith, A.D., Reynolds, D.R., Edwards, A.S., Osborne, J.L., Williams, I.H., Carreck, N.L. & Poppy, G.M. (1996). Tracking bees with harmonic radar. *Nature*, 379, 29–30. Number: 6560 Publisher: Nature Publishing Group.

Saleh, N. & Chittka, L. (2007). Traplining in bumblebees (Bombus impatiens): a foraging strategy's ontogeny and the importance of spatial reference memory in short-range foraging. *Oecologia*, 151, 719–730.

Schubert, M., Lachnit, H., Francucci, S. & Giurfa, M. (2002). Nonelemental visual learning in honeybees. *Animal Behaviour*, 64, 175–184.

Seger, J. & Adler, F. (2002). Mathematical models. In: *Encyclopedia of Evolution*. Oxford University Press, pp. 684–688.

Stout, J.C. (2000). Does size matter? Bumblebee behaviour and the pollination of *Cytisus scoparius* L. (Fabaceae). *Apidologie*, 31, 129–139.

Uhlenbeck, G.E. & Ornstein, L.S. (1930). On the Theory of the Brownian Motion. *Physical Review*, 36, 823–841.

Vallaeys, V., Tyson, R.C., Lane, W.D., Deleersnijder, E. & Hanert, E. (2017). A Lévy-flight diffusion model to predict transgenic pollen dispersal. *Journal of The Royal Society Interface*, 14, 20160889. Publisher: Royal Society.

Viscido, S.V., Parrish, J.K. & Grünbaum, D. (2007). Factors influencing the structure and maintenance of fish schools. *Ecological Modelling*, 206, 153–165. Publisher: Elsevier.

Viswanathan, G.M., Afanasyev, V., Buldyrev, S.V., Murphy, E.J., Prince, P.A. & Stanley, H.E. (1996). Lévy flight search patterns of wandering albatrosses. *Nature*, 381, 413–415.

Von Frisch, K. (1965). Die Tänze der Bienen. In: *Tanzsprache und Orientierung der Bienen* (ed. Von Frisch, K.). Springer, Berlin, Heidelberg, pp. 3–330.

Walther-Hellwig, K. & Frankl, R. (2000). Foraging habitats and foraging distances of bumblebees, Bombus spp. (Hym., Apidae), in an agricultural landscape. *Journal of Applied Entomology*, 124, 299–306. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1046/j.1439-0418.2000.00484.x.

Weber, C., Radtke, P.K., Schimansky-Geier, L. & Hänggi, P. (2011). Active motion assisted by correlated stochastic torques. Publisher: arXiv Version Number: 2.

Winfree, R. (2010). The conservation and restoration of wild bees. *Annals of the New York Academy of Sciences*, 1195, 169–197. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1749-6632.2010.05449.x.

Woodgate, J.L., Makinson, J.C., Lim, K.S., Reynolds, A.M. & Chittka, L. (2016). Life-Long Radar Tracking of Bumblebees. *PLOS ONE*, 11, e0160333.

Woodgate, J.L., Makinson, J.C., Lim, K.S., Reynolds, A.M. & Chittka, L. (2017). Continuous Radar Tracking Illustrates the Development of Multi-destination Routes of Bumblebees. *Scientific Reports*, 7, 17323.

Zuk, M. & Travisano, M. (2018). Models on the Runway: How Do We Make Replicas of the World? *American Naturalist*, 192, 1–9. Place: Chicago Publisher: Univ Chicago Press WOS:000435128300005.

Appendix A

Modeling bee movement shows how a perceptual masking effect can influence flower discovery

This appendix is published in **PLOS Computational Biology**.
10.1371/journal.pcbi.1010558

PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Modeling bee movement shows how a perceptual masking effect can influence flower discovery

Ana Morán*, Mathieu Lihoreau, Alfonso Pérez-Escudero*☯, Jacques Gautrais*☯

Centre de Recherches sur la Cognition Animale (CRCA), Centre de Biologie Intégrative (CBI), Université de Toulouse, CNRS, UPS, 118 route de Narbonne, Toulouse, France

☯ These authors contributed equally to this work.
* ana.moran-hernandez@univ-tlse3.fr (AM); alfonso.perez-escudero@univ-tlse3.fr (APE); jacques.gautrais@univ-tlse3.fr (JG)

OPEN ACCESS

Citation: Morán A, Lihoreau M, Pérez-Escudero A, Gautrais J (2023) Modeling bee movement shows how a perceptual masking effect can influence flower discovery. PLoS Comput Biol 19(3): e1010558. https://doi.org/10.1371/journal.pcbi.1010558

Editor: Ricardo Martinez-Garcia, Center for Advanced Systems Understanding (CASUS), GERMANY

Received: September 9, 2022

Accepted: March 1, 2023

Published: March 24, 2023

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pcbi.1010558

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Abstract

Understanding how pollinators move across space is key to understanding plant mating patterns. Bees are typically assumed to search for flowers randomly or using simple movement rules, so that the probability of discovering a flower should primarily depend on its distance to the nest. However, experimental work shows this is not always the case. Here, we explored the influence of flower size and density on their probability of being discovered by bees by developing a movement model of central place foraging bees, based on experimental data collected on bumblebees. Our model produces realistic bee trajectories by taking into account the autocorrelation of the bee's angular speed, the attraction to the nest (homing), and a gaussian noise. Simulations revealed a « masking effect » that reduces the detection of flowers close to another, with potential far reaching consequences on plant-pollinator interactions. At the plant level, flowers distant to the nest were more often discovered by bees in low density environments. At the bee colony level, foragers found more flowers when they were small and at medium densities. Our results indicate that the processes of search and discovery of resources are potentially more complex than usually assumed, and question the importance of resource distribution and abundance on bee foraging success and plant pollination.

Author summary

Understanding how pollinators move in space is key to understand plant reproduction and its consequences on terrestrial ecosystems. Current models assume simple movement rules that predict flowers are more likely to be visited—and hence pollinated—the closer they are to the pollinators' nest. Here we developed an explicit movement model that incorporates realistic features of bumblebee behaviour, and calibrated it with experimental data collected in naturalistic conditions. Our model shows that the probability to visit a flower does not only depend on its position, but also on the position of other flowers around that may mask it from the forager. This perceptual masking effect means that

Funding: AM was supported by a PhD Fellowship from the French Government. ML was supported by grants of the Agence Nationale de la Recherche (3DNaviBee ANR-19-CE37-0024), and the European Commission (FEDER ECONECT MP0021763, ERC Cog BEE-MOVE GA101002644). APE acknowledges funding from a CNRS Momentum grant (https://www.cnrs.fr/) and a Fyssen Foundation Research grant (https://www.fondationfyssen.fr/en/). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

pollination efficiency depends on the density and spatial arrangement of flowers around the pollinators' nest, often in counter-intuitive ways. Taking these effects into account may be key for improving practical actions in precision pollination and pollinator conservation.

Introduction

Pollinators, such as bees, wasps, flies, butterflies, but also bats and birds, mediate a key ecosystemic service on which most terrestrial plants and animals, including us humans, rely on. When foraging for nectar, animals transfer pollen between flowers, which mediates plant reproduction. Understanding how pollinators move, find and choose flowers is thus a key challenge of pollination ecology [1]. In particular, this may help predict and act on complex pollination processes in a context of a looming crisis, when food demand increases and populations of pollinators decline [2,3].

Foraging pollinators have long been assumed to move randomly [4–7] or use hard wired movement rules such as visiting the nearest unvisited flower [8], exploiting flower patches in straight line movements [9], navigating inflorescences from bottom to top flowers [10], or using win-stay lose-leave strategies [11]. Accordingly pollination models relying on these observations typically predict diffusive movements in every direction [12]. However, recent behavioural research shows this is not true when animals forage across large spatial scales [13]. In particular, studies using radars to monitor the long distance flight paths of bees foraging in the field demonstrate that foragers learn features of their environment to navigate across landscapes and to return to known feeding locations [14,15]. This enables them to develop shortcuts between feeding sites [16] and use efficient multi-destination routes (traplines) minimizing overall travel distances [17,18]. These routes are re-adjusted each time a feeding site is depleted and new ones are discovered [19].

How bees learn such foraging routes has been modelled using algorithms implementing spatial learning and memory [20–22]. While this has greatly advanced our understanding of bee exploitative movements patterns, none of these models have looked at search behaviour, either assuming insects already know the locations of all available feeding sites in their environment or discover them according to fixed probabilistic laws (e.g. the probability to discover a flower at a given location is proportional to $1/L^2$ where L represents the distance to that flower [20–22]).

However, an increasing number of experimental data indicates that this is not the case. Firstly, bees, like many pollinators, are central place foragers so that every foraging trip starts and ends at the nest site [23]. This implies that their range of action is limited. Recordings of bee search flights show how individuals tend to make loops centered at the nest when exploring a new environment and look for flowers [14,23]. These looping movements are not compatible with the assumption that bees make diffusive random walks or Lévy flights [24,25]. Secondly, the spatial structure of the foraging environment itself may also greatly influence flower discovery by bees. In particular, the probability of finding a flower heavily depends on the location of the flower visited just before, ultimately influencing the direction and geometry of the routes developed by individuals [17–20,26,27]. Since bees are more attracted by large flowers than by smaller ones [28], this suggests that small isolated flowers could be missed if they are located next to a larger patch. Such « masking effect » on the probability to visit specific flowers depending on the presence of other flowers around could have important consequences for bee foraging success, for instance by precluding the discovery of some highly

rewarding flowers. This could also influence plant pollination, if bees are spatially constrained to single flower patches and plant outcrossing is limited.

Here we explored these potential effects by developing a model of bee search movement simulating the tendency of bumblebees to make loops around their nest. We used our model to examine the probability for bees to discover flowers in environments defined by resources of various sizes and abundances. We hypothesized that looping movements characteristic of bee exploratory flights combined with perceptual masking effects by which the probability of finding given flowers is affected by the presence of others, would result in strikingly different predictions for flower discovery rates than the typical diffusive random walk movements.

Model background

Our model was designed to describe exploratory behaviour of bees, render realistic trajectories with regard to their motion and account for their "central place foraging" constraints to remain in the vicinity of their nest, with regular returns to it. We used bumblebees as model system to calibrate our bee movement model since their flights have been best described in the field [14,17–20,26–29] and we had access to experimental data (see below). We then needed to compute the probability of finding a given flower, a problem that pertains to the field of First Passage Time statistics and splitting probabilities (i.e. whether a target is hit sooner than another one) realized by a random walker with range-limited trajectories [30]. Two main approaches exist for rendering exploratory trajectories of animals (or any active matter): the Active Brownian Particle [31] and the Persistent Random Walker (or run-and-tumble [32]).

Active Brownian Motion

Active Brownian Motion and its derivatives describe trajectories by a stochastic process governing the particle's velocity [33]. These models can incorporate a central place foraging component through two main mechanisms: harmonic potentials and stochastic resetting.

Harmonic potentials trap the particle in the vicinity of the trap center, yielding a Non Equilibrium Stationary State (NESS) in the large-time limit [34–40] with particles orbiting around the center [33]. Stochastic resetting forces the particle back to the origin periodically [41], either instantaneously (1D [42], 2D [43], following a ballistic trajectory [44] or via an intermittent potential [38,45,46]).

These models can provide in some cases analytical solutions for the distribution of animals around the center, and in some cases even for the time needed to discover a target (mean first passage time, MFPT) [47–50]. For instance, the resetting rate can be tuned in order to minimize MFPT ([42,51–53], see [54] for generalization to any dimension and motion models, and [55] for generalizations to any resetting model).

However, these results are only valid under two strong assumptions: long times so that particles are in the diffusive regime, and low target density. Both of these assumptions are broken in realistic datasets of bumblebee exploration. Furthermore, none of these models provide trajectories that resemble those measured for bumblebees during exploration flights (Fig 1C).

Persistent Turning Walker

We, therefore, considered the Persistent Turning Walker (PTW) model which is an extension of the Persistent Random Walker (PRW [56], see [57–59] for a previous use in modeling ants behaviour from a cognitive perspective).

While in PRW a trajectory is considered as a succession of straight moves separated by instantaneous jumps in the orientation domain, in PTW a trajectory is considered as a

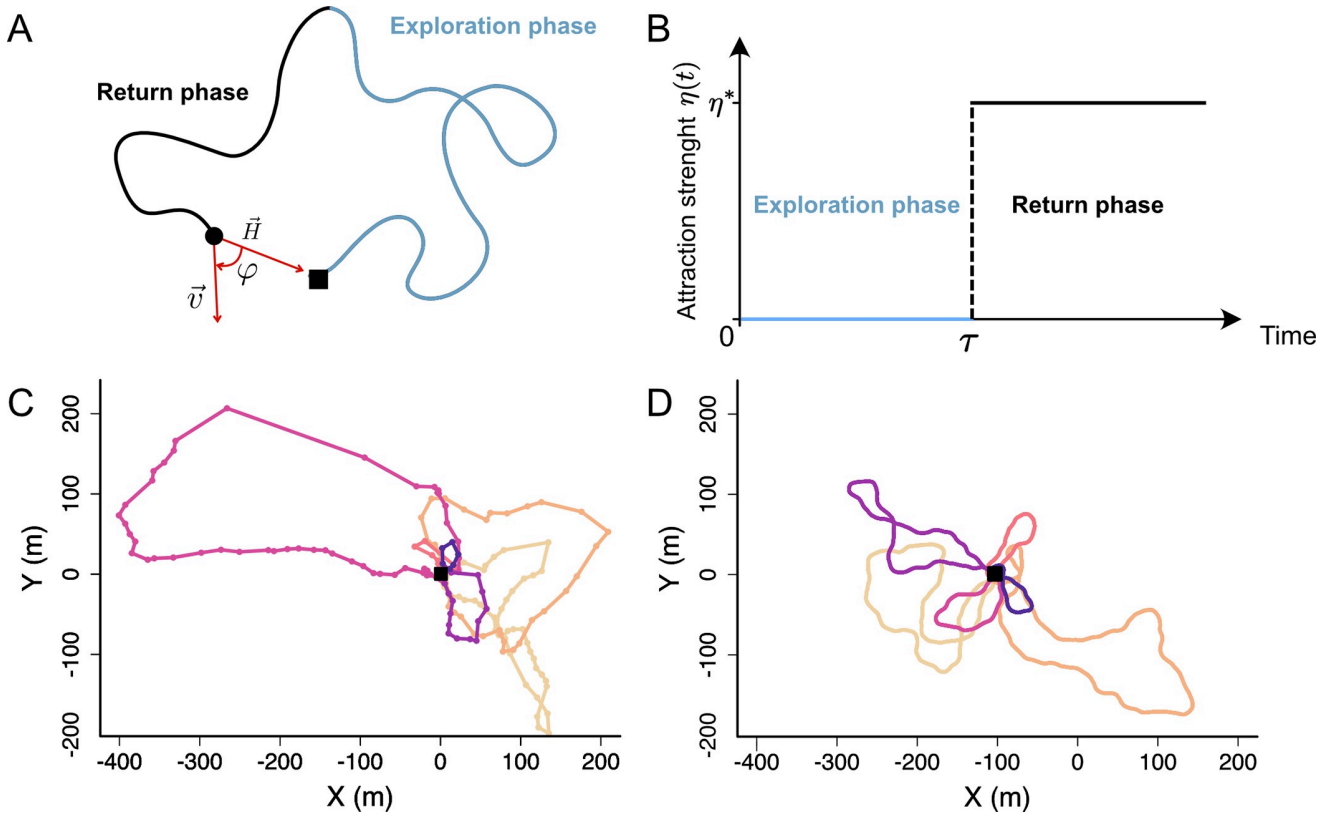

Fig 1. Illustration of the model. (A) Example of theoretical trajectory. Blue line: Trajectory during the exploration phase. Black line: Trajectory during the return phase. Black circle: bee. Black square: nest. H is the homing vector pointing towards the nest. $\vec{v}$ is the velocity of the bee. φ is the angle between $\vec{v}$ and $\vec{H}$. **(B)** Evolution of the return strength (η) over time. At time = τ, η switches from 0 (no attraction) to η^*. **(C)** Example of an experimental trajectory [71]. Each dot represents the position of a bee recorded by a harmonic radar approximately every 3s. Different colors represent different flight loops around the nest. The sequential order of the loops is represented by the color gradient where the first loops have lightest colors (yellow to purple). **(D)** Same as C, but for a simulated trajectory with parameters $\gamma = 1.0\ s^{-1}$, $\sigma = 0.37$ rad/ $s^{1/2}$, $p_{return} = 1/30\ s^{-1}$ and $\eta^* = 0.2\ s^{-1}$.

https://doi.org/10.1371/journal.pcbi.1010558.g001

succession of moves with constant curvatures, separated by instantaneous jumps in the curvature domain.

Such a description allows for a continuously defined velocity with time-correlation of the curvature. A stochastic version has been used to model fish motion behaviour [60,61].

The PTW model describes motion at constant speed, where the heading is driven by an Ornstein-Uhlenbeck process acting on the turning speed. In free-range condition, it yields a diffusion process on large time scales [62]. It has been used recently to model an active directional filament in 2D free-range conditions [63], and some large-time properties (NESS and MFPT) have been derived when in presence of a steering potential acting upon the heading [64].

Since it yields trajectories that are similar to our data, we used the PTW model as a basis for our own model.

To confine trajectories around the nest, we added an intermittent steering potential acting on the turning speed and that activates when bees decide to return to the nest. Thanks to this steering potential, our model describes the full exploration trips of bees (starting and ending at the nest location), which facilitates comparisons between model simulations and experimental data.

Results

Description of the model

For the sake of simplicity, here we modelled bee movements in 2D, neglecting altitude. We assumed that bees fly at constant speed v and with varying angular speed, $\omega(t)$ (signed turning rate of the heading, measured in radians per second) which is governed by

$$d\omega(t) = -\gamma[\omega(t) - \omega^*(t)]dt + \sigma dW(t), \quad (1)$$

where γ is an auto-correlation coefficient and $\sigma dW(t)$ introduces a gaussian noise, governed by a Wiener process [65]. The two terms of Eq 1 have opposing effects: The first term pushes the angular speed towards a target angular speed, $\omega^*(t)$, with a strength controlled by the auto-correlation coefficient γ. The second term introduces noise in the angular speed making bees change direction. Therefore, high values of γ and low values of σ lead to smoother and more predictable trajectories. Setting $\omega^*(t) = 0$ leads to a trajectory with no preferred direction, whose angular speed changes smoothly around zero. This is the simplest condition, resembling a diffusive process in which the animal moves aimlessly and gets further and further from its initial position as time goes by [62,63,66–69].

We modelled central place foraging by adding an attraction component to the model in order to make bees return to the nest after a certain amount of time. To implement the return to the nest (homing) we assumed that bees can locate the direction of their nest at any time using path integration (i.e. navigational mechanism by which insects continuously keep track of their current position relative to their nest position [70]), and define a homing vector, $\vec{H}(t)$ that points towards the nest [13]. Then, we assumed that the bee tries to target the angular speed that will align its trajectory with the homing vector, so we modeled the target angular speed as

$$\omega^*(t) = \eta(t)\varphi(t), \quad (2)$$

Where $\varphi(t)$ is the angle between the bee's velocity $\vec{v}(t)$ and the homing vector $\vec{H}(t)$ (Fig 1A), and $\eta(t)$ is the attraction strength that controls a switch between the exploration and return phases: During the initial exploration phase we make $\eta(t) = 0$, so that bees explore randomly and distance themselves from the nest, while during the return phase we make $\eta(t) = \eta^* > 0$, so that the bee has a continuous tendency to turn towards the nest. We assumed that bees switch instantaneously between the exploration and return phases, so

$$\begin{cases} \eta(t) = 0 \; if \; t < \tau \\ \eta(t) = \eta^* \; if \; t \geq \tau \end{cases} \quad (3)$$

where τ is the time at which the switch happens (Fig 1B). This switch may happen at any time, with a constant probability per unit of time, p_{return}. This means that the switching times are exponentially distributed, with an average time of $1/p_{return}$.

The model therefore has four main parameters: The auto-correlation (γ) and the randomness (σ) control the characteristics of the flight, while the probability per unit of time to return (p_{return}) and the strength of the attraction component (η^*) control the duration of each exploration trip. Here we have described the continuous version of the model, but to implement it numerically we discretized it in finite time steps (see Methods).

Calibration with experimental data

In principle, our model can describe search movements of any central place forager. Here we explored its properties focusing on a model species for which we had access to high-quality

experimental data: the buff-tailed bumblebee *Bombus terrestris*. We used the dataset of Pasquaretta et al. [71] in which the authors used a harmonic radar to track 2D trajectories during exploration flights of bees in the field. Bees carrying a transponder were released from a colony nest box located in the middle of a large and flat open field, and performed exploration flights without any spatial limitation. The radar recorded the location of the bees every 3.3s over a distance of ca. 1km (Fig 1C). In these experiments the bees were tested until they found artificial flowers randomly scattered in the field. We used 32 tracks from 18 bees.

To quantify the experimental trajectories, we first divided tracks into flight "loops", each loop being a segment of trajectory that starts and ends in the nest (Fig 1C). This extraction yielded 207 loops. We then computed four observables for each loop (Fig 2):

- Loop length: Total length of the trajectory for a given loop (Fig 2A).
- Loop extension: Maximum distance between the bee and the nest for a given loop (Fig 2B).
- Number of intersections: Number of times the loop intersects with itself (Fig 2C).
- Number of re-departures, where a re-departure is defined as three consecutive positions such that the second position is closer to the nest than the first one, but the third is again further away than the second. These events indicate instances in which the bee seemed to be returning towards the nest and turned back (Fig 2D).

We extracted these four parameters from each loop and found substantial variability in all of them (Fig 2, black lines). We then used this information to find the optimal model parameters, aiming to describe not only the average value of each observable, but also their distributions. To do so, we performed simulations covering exhaustively all relevant combinations of

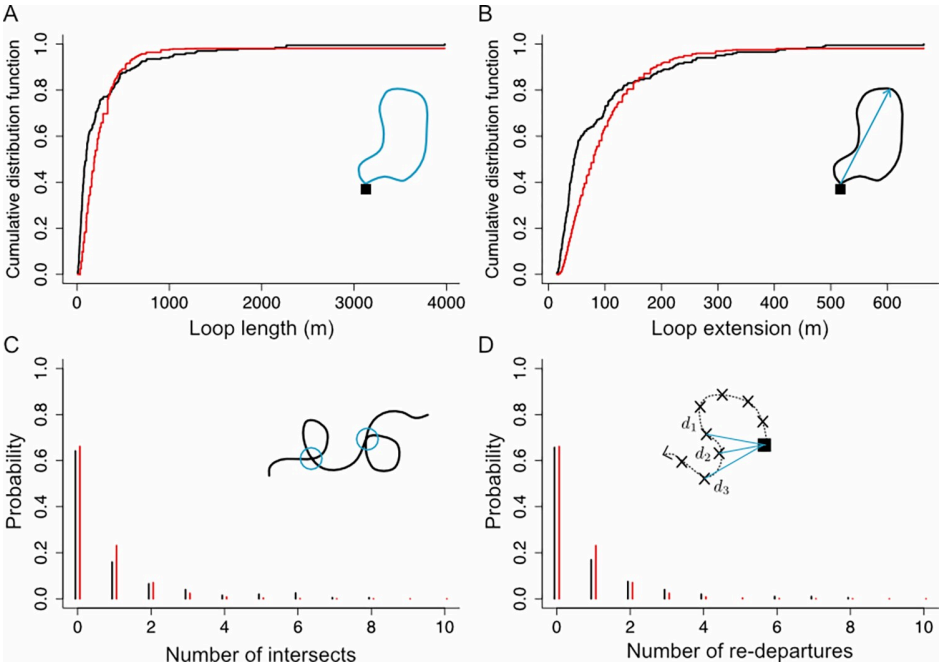

Fig 2. Distributions of the four observables, for experimental and simulated data. Black lines: experimental data. Red lines: model predictions using the optimal $\gamma = 1.0\ s^{-1}$, $\sigma = 0.37\ \mathrm{rad/s}^{1/2}$, $p_{\mathrm{return}} = 1/30\ s^{-1}$ and $\eta^* = 0.2\ s^{-1}$. Insets: Schematic of each observable. **(A)** Cumulative distribution function of loop lengths for our full dataset. **(B)** Same as A, but for the loops extension. **(C)** Probability distribution of the number of trajectories intersects per 100m traveled. **(D)** Same as C, but for the number of re-departures per 100 m traveled.

https://doi.org/10.1371/journal.pcbi.1010558.g002

our four parameters. For each combination of parameters, we simulated 1000 loops, extracted the distributions for the four observables, and chose the parameter combination that best approximated the experimental distributions for the four observables (see Methods). This procedure resulted in the optimal parameters $\gamma = 1.0\ s^{-1}$, $\sigma = 0.37$ rad/s$^{1/2}$, $p_{\text{return}} = 1/30\ s^{-1}$ and $\eta^* = 0.2\ s^{-1}$, which give a good approximation to the experimental distributions of observables (Fig 2, red lines), and trajectories that qualitatively resemble the experimental ones (Fig 1D).

Note that our calibration procedure yields a switching time from the "exploration phase" to the "return phase" of 30 seconds on average. This may be an underestimation since the radar can detect targets at a maximum distance of about 1km [72] and bumblebees can fly beyond this limit depending on forage availability in the landscape [73]. However, to the best of our knowledge, these are so far the best data available.

Model predictions

Attraction to the nest limits the exploration range of bees. An unrealistic feature of existing diffusive models is their long-term behaviour: If given enough time, the forager reaches extremely far distances with respect to the nest, never returning to it. To illustrate the impact of central place foraging on the simulation of bee exploration range, we compared our model with attraction to the nest to an alternative one in which the attraction is absent (i.e., making $\eta^* = 0$ in Eq 3). We simulated 1000 trajectories with each model for different amounts of time, and studied how the distribution of bees around the nest changes over time. As expected, attraction retains bees tightly localized around the nest (~250m) (Fig 3A and 3B, blue). More interesting, it makes the distribution of bees stationary: In a model without attraction, bees constantly wander away from the nest, and their distribution depends on how much time we allow for the bee to explore, becoming wider as time goes by (Fig 3B and 3C, orange). In contrast, the attraction component makes the forager return to the nest periodically, so the distribution remains stationary once the forager has had enough time to perform more than one loop on average (Fig 3D, blue).

This demonstrates the importance of taking into account the attraction component when modelling bee movements (instead of considering a diffusive model) to be able to reproduce realistic trajectories.

To highlight the effect of behavioural parameters upon the spatial extent covered by exploration, we present in S1 Fig the marginal effect of the three main parameters (α, η^* and σ) on the range explored by the bee, as measured by the long-time Mean Square Displacement (MSD), i.e. the variance of the stationary distribution (also known as the Non-Equilibrium Stationary State, or NESS, of the motion process in Statistical Physics).

Distant flowers are more often discovered in low-density environments. We estimated the probabilities of flowers to be discovered (and thus potentially pollinated) by bees in a simulated field characterized by a random and uniform distribution of flowers, an average density of $1.3\ 10^{-4}$ flowers/m^2 and a diameter of 70cm (for the sake of simplicity here a "flower" is equivalent to a feeding location, which may be a single flower or a plant containing several ones). We assumed that a flower was discovered by a bee whenever its distance to the bee's trajectory was below a threshold, given by the bee's visual perception range (see Methods). We focused on vision rather than olfaction because it is the main sense that bees use to accurately navigate the last meters towards a particular flower, while olfaction is used at a broader spatial scale [74]. Using these conditions, we simulated 1000 foraging trips, each of them lasting 900 seconds, and for each flower we computed the probability to be found in a given trip (i.e., the proportion of simulations in which the trajectory overlaps with the flower's area of attraction). This probability falls exponentially with the distance between the flower and the nest (Fig 4A, red line).

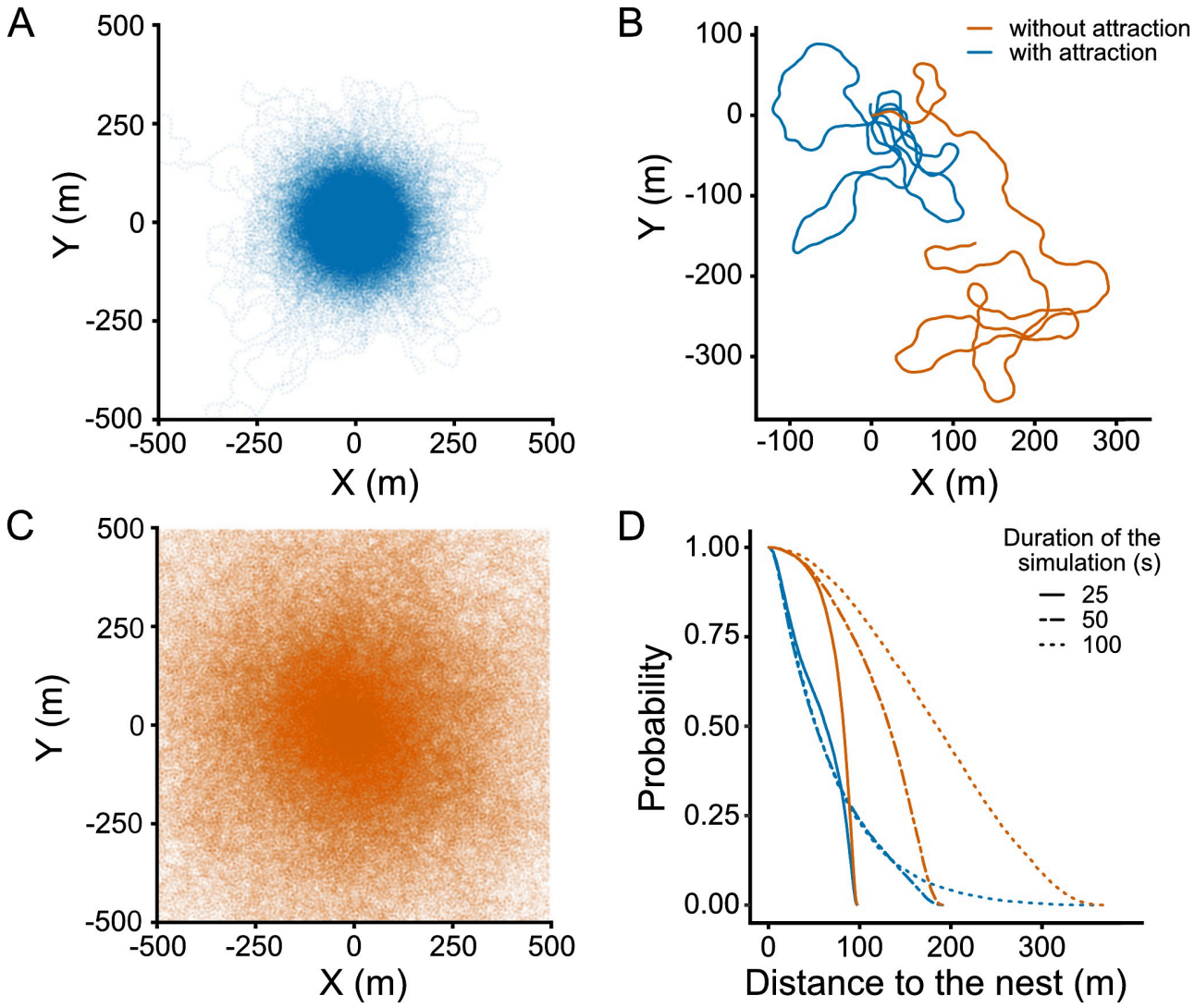

Fig 3. Probability of presence of a bee around the nest. (A) Overlay of 1000 trajectories with attraction to the nest ($\eta^* = 0.2\ s^{-1}$) simulated during 900 s. **(B)** Example trajectories with and without attraction, simulated during 500 s. The nest is located at (0,0). Blue: model with attraction. Orange: model without attraction. **(C)** Same as A but without attraction to the nest ($\eta^* = 0$). **(D)** Probability to find a bee below a given distance to the nest (i.e., inverse cumulative probability distribution for the distance to the nest) after different amounts of time. Blue: model with attraction (stationary distribution of bees). Orange: model without attraction (non-stationary distribution of bees).

https://doi.org/10.1371/journal.pcbi.1010558.g003

We then tested the influence of a potential perceptual "masking effect" in which the probability of discovering a flower does not only depend on its distance to the nest, but can also be influenced by the presence of other flowers around it (Fig 4B). This dependence exists because a bee that finds a flower might not continue its trajectory, but might rather stop to collect nectar. Once nectar collection is over, the bee may continue exploring, but after visiting a few flowers the bee returns to the nest to unload its crop. For example, in a scenario where there are just two flowers equidistant to the nest, both flowers should be visited equally. However, if another flower is added, it can capture visits that would otherwise visit one of the original flowers, reducing the probability that it is discovered (Fig 4B). For the sake of simplicity, we assumed that each bee returns to the nest after discovering a single flower. The first qualitative

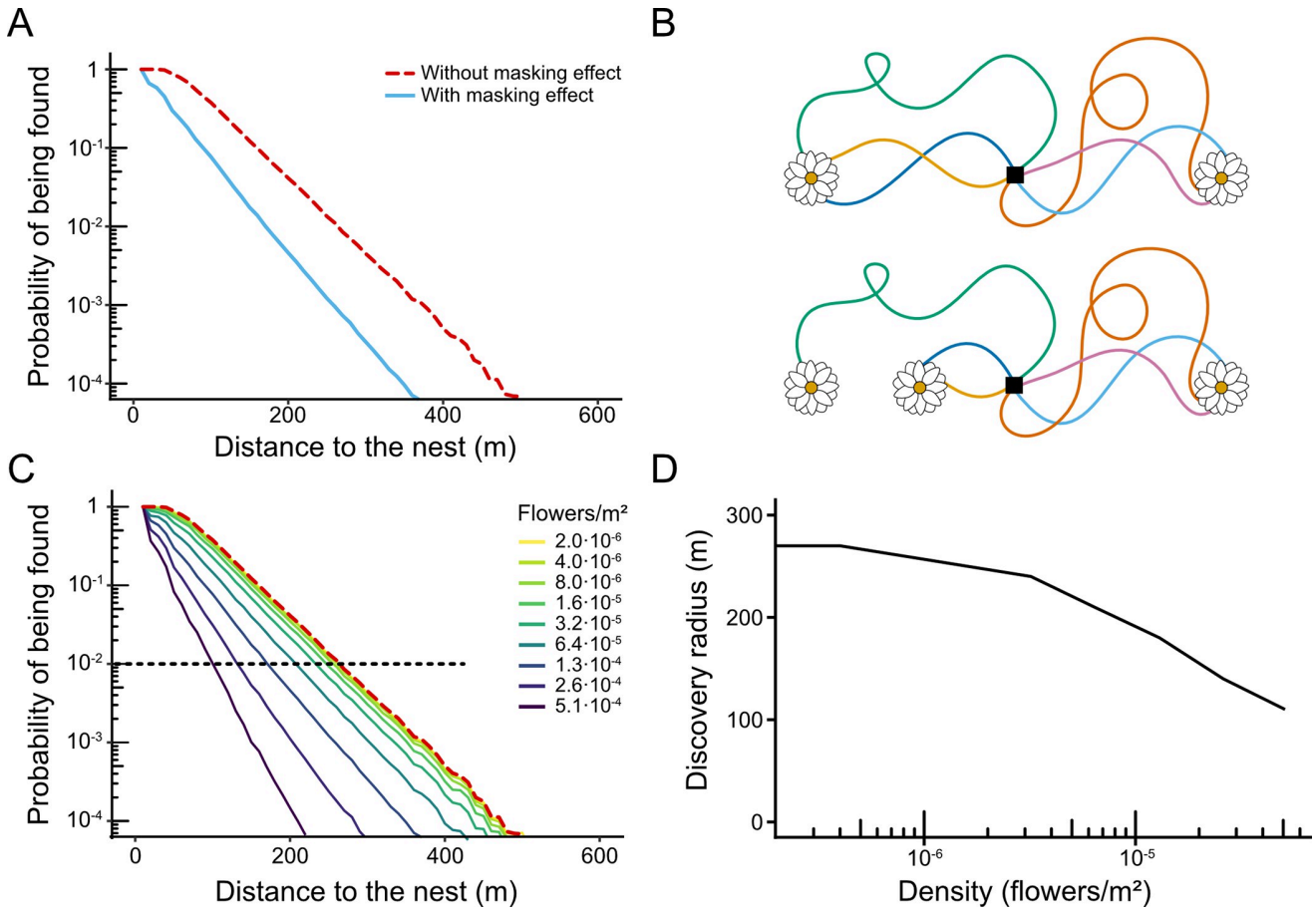

Fig 4. Probability that flowers are discovered. (A) Probability that a flower is found as a function of its distance to the nest. We simulated exploration trips in a field of uniformly distributed flowers with density 1.3 10^{-4} flowers/m^2 and flower size 70 cm. For each flower, we computed the probability that it was found in each exploration trip, and we show this probability as a function of the distance between the flower and the nest. Results were computed over 6000 simulated trips of 900s in 10000 environments for each density (negligible bar errors not reported). Red line: Probability calculated without taking into account the masking effect. Blue line: Probability calculated taking into account the masking effect (i.e., only counting the first flower that was discovered in each trip). **(B) Illustration of the masking effect.** The probability of discovering a flower depends on the presence of other flowers. In a scenario where there are just 2 flowers equidistant to the nest, both flowers should be visited equally (top). However, if another flower is added, it can capture visits that would otherwise visit one of the original flowers (bottom). Black square: nest. **(C)** Same as (A), but for different flower densities. Red dotted line: Probability calculated without taking into account the masking effect. This probability is independent of the density of flowers. Solid lines: Probability calculated taking the masking effect into account. Black dotted line: threshold probability at which we consider an area that has a high probability of being pollinated. **(D)** Radius of the area around the nest that has a high probability of being discovered (i.e., where the probability that flowers are discovered is above 10^{-2}) as a function of flower density.

https://doi.org/10.1371/journal.pcbi.1010558.g004

consequence of the masking effect is to reduce the probability that flowers distant to the nest are discovered (Fig 4A, blue). The second consequence is that it can introduce a dependence of flower density on discovery rate. In the absence of masking, only two factors determine the probability that a flower is discovered: its size (which determines the distance from which it can be perceived) and its distance to the nest. In contrast, when the masking effect is taken into account, the number of discoveries visits also depend on the overall density of flowers in the environment, falling more sharply with distance when this density is higher (Fig 4C).

This dependence with flower density implies that the area around the nest where flowers have a high probability of being discovered depends on flower density. To estimate the size of this area, we set a threshold at a probability of 10^{-2} per trip (black dotted line in Fig 4C), and computed the "discovery radius" as the distance at which flowers' probability of being

discovered remains above this threshold. At low flower densities, the discovery radius reaches 270 meters, and is limited by the bees' exploration range (i.e., their tendency to return to the nest after a certain time, even if no flowers have been found; compare this radius with the distribution in Fig 3C). Due to the masking effect, the discovery radius decreases as flower density increases (Fig 4D).

Populations of bees find more flowers at intermediate densities. We then explored the influence of the masking effect on the total number of flowers discovered by a population of bees (i.e. a bumblebee colony).

To study this effect, we computed the total number of flowers discovered by a bee colony as a function of density and flower size. We considered a field with flowers of a given size uniformly and randomly distributed with a given flower density, simulated 100 exploration trips, and counted the number of flowers that were discovered at least once. When we performed this simulation neglecting the masking effect (i.e., assuming that a bee discovers all the flowers that intersect with its trajectory, not being affected by previous discoveries), we found that the number of flowers discovered increased with flower density and flower size, as these factors make flowers more plentiful and easier to find (Fig 5, dashed lines). However, the masking effect reverses this trend (Fig 5, solid lines): For low densities, the masking effect is weak and the number of discovered flowers increases with density, but at high flower densities, bees become "trapped" around the nest by the flowers immediately surrounding it, which accumulate most of the visits. Therefore, there is an optimum density that results in the highest number of different flowers discovered. Since the masking effect is stronger for larger flowers, the effect of size also reversed, with the number of discovered flowers decreasing as flower size increases (Fig 5, solid lines).

These results are robust to the assumptions of the model. Firstly, they do not depend on the precise number of flowers discovered by each bee in each exploration flight. In many natural conditions, bees may need to discover and visit several flowers to fill their nectar crop to capacity before deciding to return to the nest. Taking these multiple discoveries into account (1 to 8 discoveries) leaves our results qualitatively unchanged (Figs 5B and S2).

Our results are also robust to nectar depletion. Once a flower is visited by a bee, its nectar load may be partially depleted. The next bee visiting the flower may therefore be less inclined to terminate its exploration flight and return to the nest after visiting this flower, so the number of flowers discovered by each bee before returning to the nest may depend on the previous exploration flights performed by other bees in the colony. To account for flower depletion, we ran a simulation in which bees will ignore any flower that has already visited in a previous exploration flight. In this case, the maximum at intermediate densities is lost, with higher flower density always leading to more discovered flowers (S3E Fig). However, for large patch sizes, we know that flower depletion is only partial and visiting each flower has a cost in terms of time and energy (bees must land on each flower, even if it is depleted). To account for this cost, we limited the total number of depleted flowers a bee will visit before returning to the nest, and in this condition we again observe a maximum of discovered flowers at intermediate densities (S3A–S3D Fig).

Discussion

How pollinators search for flowers is of fundamental importance for behavioural research, pollination and conservation but remains poorly quantified. Here we developed a realistic model of bee search movements based on their observed tendency to make exploratory loops that start and end at their nest location. We used a Persistent Turning Walker model which has inspired some developments in other animals, especially fish [41–43] as well as in robotics [44]

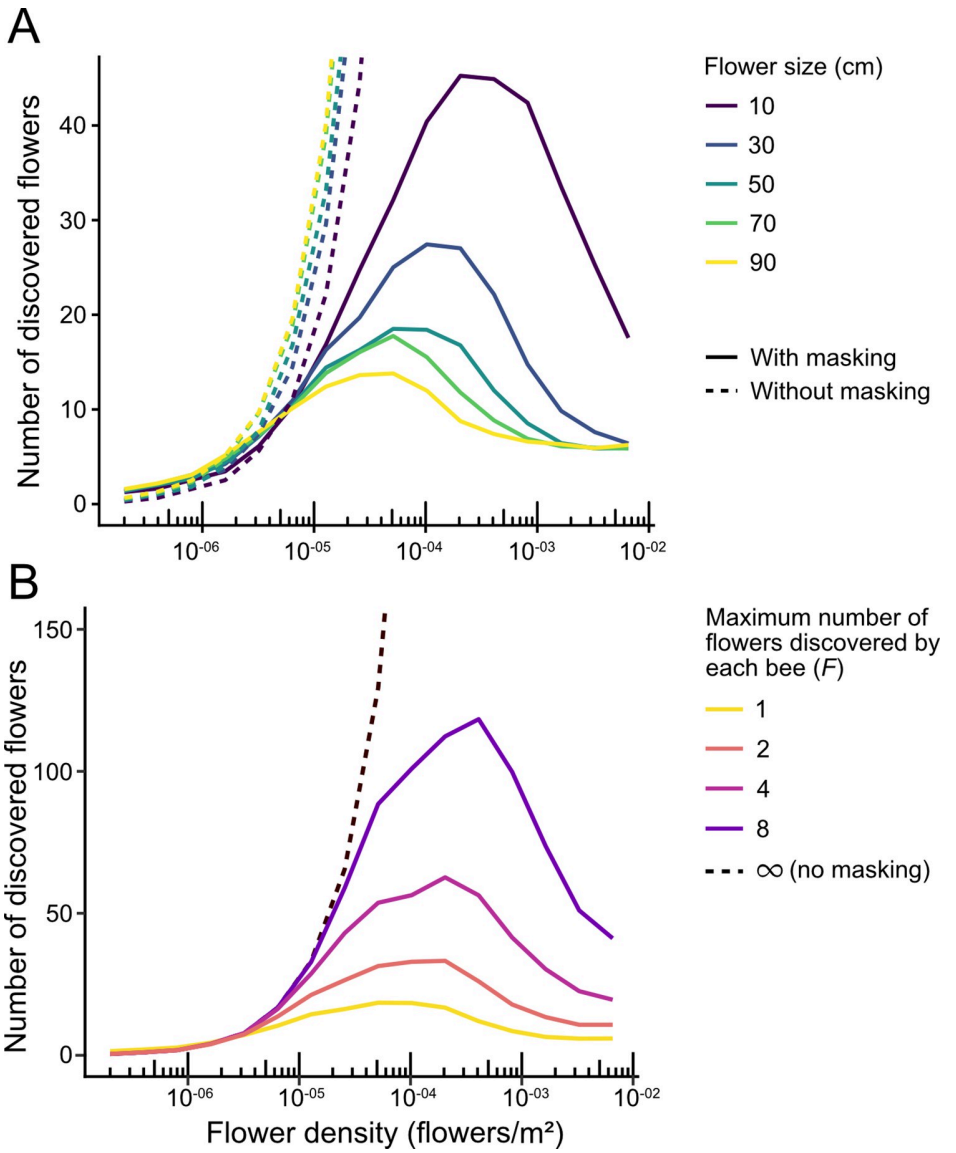

Fig 5. Number of different flowers discovered by a group of bees as a function of flower density. (A) Number of different flowers discovered in 100 exploration trips of 900 s, in an environment with randomly distributed flowers. Results are averaged over 80 simulations, keeping the environment stable for every simulation. Solid lines: Number calculated taking into account the masking effect (i.e., only counting the first flower that was discovered in each trip). Dotted lines: Results without taking into account the masking effect. **(B)** Same as A but for a given flower size (50 cm), and assuming that each bee will return to the nest only after having discovered a number of flowers (F) (note that box A corresponds to $F = 1$ for the simulations with masking effect, and $F = \infty$ for the simulations without masking effect). Line colors represent the maximum number of flowers discovered by each bee (F).

https://doi.org/10.1371/journal.pcbi.1010558.g005

where it has been proven to display better coverage properties than classical random walks [45]. Our model, calibrated with real bee behavioural data (i.e. bumblebee radar tracks), produces two-dimensional trajectories with progressive changes of direction driven by the continuous evolution of the angular velocity $\omega(t)$. Using this approach, we documented a neglected yet potentially fundamentally important effect for flower discovery by bees: a perceptual

masking effect that influences the probability of bees to find flowers not only based on their size and spatial location, but also on the presence and characteristics of other flowers around them.

Previous models assume that bees explore their environment randomly using Lévy flights or other diffusive processes [12,21,22]. In a diffusive model, individuals are able to wander away from the nest indefinitely if given enough time. In contrast to these models, our model replicates looping trajectories observed in real bees [14,29], which confines the presence of individuals around a nest (Fig 3). As a consequence of the periodic returns of bees to the nest, their distribution becomes independent of the time given to explore. This result has the important consequence that, under the assumptions of our model, longer simulation durations result in a more thorough exploitation of the foraging area around the nest, but not in a larger area.

By explicitly simulating individual trajectories in complex environments, our model revealed how the presence of a flower may decrease the probability of discovering another, a phenomenon that we call "perceptual masking effect". Although in this study we strictly focused on flower discovery during exploration flights, our results suggest that the masking effect can have broader influences on exploitation patterns, bee foraging success and plant pollination. Indeed, at the level of individual bees, flower discovery can impact site fidelity by foragers and their tendency to develop traplines to regularly revisit known feeding locations [17]. Given the perceptual masking effect, flower patches may not be discovered with the same probability or in the same order, which may impact their likelihood to be exploited and the ability for bees to develop efficient routes minimizing overall travel distances between different flowers [21]. The masking effect may also influence the global foraging success of the colony, which depends on the number of flowers discovered collectively by all the bees of a colony, because a flower discovered and exploited by a bee will be at least partially depleted, giving marginal benefit to later visitors. For this reason, what counts is not the total number of visits that bees perform, but rather the total number of different flowers discovered by the colony. In our simulations, colonies tended to find more flowers when they were small and at medium densities (Fig 5). This suggests that there is an optimal flower size and density at which collective foraging efficiency is optimized (although the effect of size on foraging efficiency will be compounded with the greater reward provided by bigger flowers on average).

Similar extrapolations of our results on first flower discovery can be made regarding pollination. At the plant level, we found that flowers distant to the nest were more often visited in low density environments (Fig 4). Since bees disseminate pollen (and thus mediate plant sexual reproduction) when visiting flowers, this may generate lower probabilities of pollination at high flower densities, by which the area that is pollinated around the nest decreases as the density of flower increases. If this prediction is verified in future studies, this would mean that the overall distribution of flower patches directly impacts their pollination and should be taken into account when designing strategies for crop production and assisted pollination.

Our search model is a scaffold for future quantitative characterization of the movement of bees, or any central place forager, across time and landscapes. Although we limited our study to flower discovery probability, and therefore only provided predictions for first flower discovery, the model could be used to investigate the full foraging trips of bees, and how they change through time as bees acquire experience with their environment and develop spatial memories [13]. It would be particularly interesting to integrate this exploration model into existing learning exploitation models proposed to replicate route formation by bees [20–22] and to study dynamics of resource exploitation by populations of bees [22,75]. Once a flower is discovered, its location can be learned, and new exploration may start from it, ultimately allowing for the establishment of traplines. This would be modelled via a modification of the attraction component, which can be modified to point towards previously-discovered flowers instead of the

nest. Importantly, such model predictions (flower discovery probability, visitation order, flight trajectories) can be experimentally tested and the model calibrated for specific study species. This will greatly facilitate improvement and validation for potential applications. As discussed above, robust predictive models of bee movements including both exploration and exploitation would be particularly useful for improving precision pollination (to maximize crop pollination), pollinator conservation (to ensure correct population growth and maintenance), but also in ecotoxicology (to avoid exposure of bees to harmful agrochemicals) and legislation (to avoid unwanted gene flow between plants). Beyond pollinators, our minimal persistent turning walker model could be calibrated to apply to a wide range of species, providing a data-based quantification and predictions for further exploration of the broader interactions between central place foraging animals and their environment.

Methods

The codes used to perform all the simulations, data analyses and figures are available in the S1 Data.

Modeling nest and flower detection

Bumblebees can detect an object when it forms an angle of 3˚ on the retina of their compound eyes [76]. Therefore, for every model simulation, we set the flowers' size and calculated the distance at which the bees are able to detect them. We call this the "perception distance". We considered that a bee discovered a flower when it was located at a distance to the bee inferior to the perception distance. We did not take into account the olfactory perception since it could be less reliable because of other factors like wind direction and the flower's species. If taken into account, this would only impact the perception distance of the flowers and the results would not be qualitatively different.

Analysis of experimental data

We used the dataset of Pasquaretta et al. [71] in which the authors tracked exploratory flight trajectories of bumblebees in the field with a harmonic radar. Bees carrying a transponder were released from a colony nest box located in the middle of a large and flat open field, and performed exploration flights without any spatial limitation. The radar recorded the location of the bees every 3.3s over a distance of ca. 800 m, and with an accuracy of approximately 2 m [14]. The bees were tested until they found one of three 20-cm artificial flowers randomly scattered in the field. The position of these flowers was changed whenever one of them was found to prevent the bees from learning their location, but their presence may still affect the bees' trajectories. We first attempted to control for this factor by removing all trajectories where bees passed near an artificial flower, but this introduced a significant bias towards short trajectories, because bees are less likely to find a flower when they stay near the nest. Therefore, we used the full dataset, and in order to remove the effect of the bees hovering around and exploiting the artificial flowers, we summarized all the points detected in an area within 6 m of an artificial flower as a single point at the location of the flower. This threshold of 6 m was derived from a 4 m perception distance corresponding to 20 cm flowers, plus 2 m to account for the experimental noise. All trajectories are given in S2-section V.

Dividing trajectories into loops

In order to quantify the trajectories, we divided the tracks into "loops". We defined a loop as a fragment of a trajectory that starts when the bee leaves that nest and finishes when it enters

back. The colony nest box used in the experiments was rectangular, with a diagonal of 37 cm, meaning that the bees were able to see it at approximately 7m. However, in this case we set a higher threshold of 13 m to avoid including learning flights (i.e., flights during which the bee makes characteristic loops to acquire visual memories of target locations such as the nest for navigation [70]) into the set of exploratory data. While our model does not produce learning flights, for consistency we also used the 13-m radius around the nest in our simulations.

Model simulations

All simulations start at the nest (which is located at position 0,0), with a random initial heading, and with zero angular velocity. To simulate the trajectories, we discretized the model using a time step $\Delta t = 0.01$. Therefore, at every step we calculated position $\vec{x}(t + \Delta t)$ with:

$$\vec{x}(t + \Delta t) = \vec{x}(t) + \vec{v}(t)\Delta t. \tag{M1}$$

Then, the direction $\theta(t + \Delta t)$ was calculated using

$$\theta(t + \Delta t) = \theta(t) + \omega(t)\Delta t. \tag{M2}$$

The velocity $v(t + \Delta t)$ was calculated with

$$\vec{v}(t + \Delta t) = v\begin{pmatrix} cos(\theta(t + \Delta t)) \\ sin(\theta(t + \Delta t)) \end{pmatrix}, \tag{M3}$$

where v is the speed, which is a constant in our model.

Lastly, we calculated the angular speed $\omega(t + \Delta t)$. For this, we used the Green function for Ornstein-Uhlenbeck processes over Δt (see [60] for details), obtaining

$$\omega(t + \Delta t) = \omega(t)e^{-\gamma\Delta t} + \omega^*(1 - e^{-\gamma\Delta t}) + \varepsilon, \tag{M4}$$

where ω^* is the target angular speed (governed by Eqs 2 and 3), and ε is a random number governed by a Gaussian distribution with mean 0 and variance

$$s^2 = \sigma^2 \frac{1 + e^{-2\gamma\Delta t}}{2\gamma}. \tag{M5}$$

Flower discovery models

Our basic model with masking effects assumes that a bee returns to the nest after discovering one flower and will ignore any flowers encountered on its way back. We implemented this by simply stopping our simulation when the bee discovers the first flower (if no flowers are discovered, the simulation continues until the central-place-foraging component makes the bee return to the nest).

Our multiple-discovery model (Figs 5B and S3) assumes that a bee will continue its original trajectory after each discovery (i.e. for simplicity we do not include the landing on the discovered flowers), and will return to the nest after discovering F flowers (so we stop the simulation when the bee discovers the F-th flower).

Our model with flower depletion (S4 Fig) takes into account the order in which bees make their exploration trips. If a flower has been discovered by a bee in a previous trip, it's marked as depleted. We stop the simulation when a bee discovers a non-depleted flower, or after discovering a maximum number F_{depleted} of depleted flowers.

Parameter fitting

In order to fit the parameters of the model, we explored systematically all relevant combinations within the relevant range for each parameter. To do this more efficiently, we transformed one of the four parameters of the model into a more tractable one: We substituted the variance of the noise introduced by the Wiener's process (σ) for the variance of the angular speed ω, which has a more direct impact on the experimental data. These two variables are related by [60]:

$$Var(\omega) = \Omega = \frac{\sigma^2}{2\gamma}.$$

We also defined

$$\alpha = \frac{1}{p_{return}}.$$

After these transformations, our model is defined by the four parameters (γ, Ω, α, η^*). To find the optimal values of these parameters, we exhaustively explored the 4-dimensional parameter space in the relevant range of each parameter: We explored all 6160 different combinations resulting from the following values of each parameter:

- $\gamma \in$ (0.5,0.6,0.7,0.8,0.9,1.0,1.1,1.2,1.3,1.4,1.5)
- $\Omega \in$ (0.01,0.03,0.05,0.06,0.07,0.08,0.09,0.1,0.125,0.15)
- $\alpha \in$ (10,20,25,30,35,40,50)
- $\eta^* \in$ (0.05,0.1,0.15,0.2,0.25,0.3,0.35,0.4)

For each combination of parameters, we simulated 10^3 loops and computed the distribution of each of the four observables defined in Fig 2. Then we computed the distance between the experimental distribution of each observable and the simulation results: for the two continuous observables (loop length and extension), this distance was computed as the area between the observed cumulative distribution function and the simulated one. For the two discrete observables (numbers of self-intersection and re-departures), the distance was computed as the sum of the absolute differences between all points of the two probability distributions. This yielded four distributions of distances over the 6160 combinations. Since the four observables are heterogeneous (two are continuous measures, two are discrete), we had to re-normalize the distances to ensure that each observable is given the same weight. We did this by translating the distances into their quantile in their corresponding cumulative distribution (e.g., a distance translated into 0.12 means that it is within the lowest 12%). Finally, we retained the combination that yielded the lower quantile averaged over the four observables (see S1 Text, section 2).

The best parameters combination was found to be: $\gamma = 1.0\ s^{-1}$, $\Omega = 0.07\ \mathrm{rad}^2/\mathrm{s}^{-2}$, $\alpha = 30\ s$ and $\eta^* = 0.2\ s^{-1}$. It corresponds to the marginal local minima for the four observables (see S1 Text, section 2). Simulated trajectories closely resemble data trajectories (Fig 1B), and the model is able to produce loops with an elongated shape, as well as a diversity of loop lengths.

This unique set of parameters assumes that all bees are identical, while in reality inter-individual differences exist (S2 Fig), for example due to differences in age, experience, learning or size [77,78]. However, each bee can display a large diversity of loop parameters, covering a similar range as the overall population (S1 Fig). We therefore considered that separate fits for each individual were not justified. The fact that our model reproduces not only the mean but also the variability of the four observables we defined (Fig 3) supports this choice.

Supporting information

S1 Fig. Sensitivty of the Mean Square Displacement (MSD) to the behavioural parameters. We numerically computed the MSD (m^2), varying each parameter in turn, leaving the others unchanged (keeping their values fitted from the dataset). The MSD was estimated using 10^5 simulated loops for each point, using the default parameters $\gamma = 1.0\ s^{-1}$, $\sigma = 0.37$ rad/ $s^{1/2}$, $p_{\text{re-turn}} = 1/30\ s^{-1}$ and $\eta^* = 0.2\ s^{-1}$ and varying (A) α (s), (B) η^* (s^{-1}) *or* (C) $\Omega = \frac{\sigma^2}{2\gamma}$ $(\text{rad}^2/\text{s}^{-2})$ while leaving the other parameters unchanged. Note that for smaller values of η^*, the distribution tends to (non stationary) diffusion, so, some loops are censored at one hour. The effect of $\alpha = \frac{1}{p_{turn}}$, the mean duration in exploration mode, appears quite linear, which is not surprising but would call for an analytical demonstration. During the return phase, η^* represents the intensity of the steering (the potential stiffness), since for smaller values of η^*, the relaxation to the preferred turning speed would be less effective. As η^* gets smaller and smaller values, the steering vanishes, so that the bee would adopts a diffusive behavior, with no Non Equilibrium Stationary State (NESS). On the other hand, we observe a clear effect of saturation for large enough values of η^*, meaning that the effectiveness of the steering is limited by the relaxation time gamma. Finally, Ω controls the level of noise the turning speed can undergo. For $\Omega = 0$, the turning speed has no noise at all, and the process becomes deterministic: bee would fly from the nest ballistically in the exploration phase and go back ballistically to the nest after turning maneuver induced by the steering process. In this case, given the initial condition of null turning speed, the trajectories would push bees the farthest from the nest during exploration (hence, maximal MSD), while larger values of Ω would drive bees to meander around the nest, leading to trajectories that remain closer to the nest.
(TIFF)

S2 Fig. Variability of each observable across individuals in the experimental dataset. (A) Loop lengths (m) for each bee, as defined in Fig 3 in the main text. Boxplots, show the median (middle line), 25 and 75% quantiles (box), range of data within 1.5 interquartile deviations (whiskers), and outliers (dots). **(B)** Same as A but for the loop extension (maximum distance between the nest and the individual). **(C)** Same as A, but for the number of re-departures per 100m traveled. A re-departure is defined as three consecutive positions such that the second position is closer to the nest than the first one, but the third is again further away than the second. **(D)** Same as A but for the intersections (number of times the loop intersects with itself).
(TIFF)

S3 Fig. Number of different flowers discovered by a group of bees as a function of flower density, when bees discover more than one flower per trip. (A) Number of different flowers discovered in 100 exploration trips of 900 s, in an environment with randomly distributed flowers. Results are averaged over 80 simulations, keeping the environment fixed for every simulation. Solid lines: Number calculated taking into account the masking effect (i.e., only counting the first flowerF = 2 flowers that wereas discovered on each trip). Dotted lines: Probability calculated without taking into account the masking effect. **(B)** Same as (A), but for $F = 4$. **(C)** Same as (A), but for $F = 8$. Note the difference of scales for the ordinates.
(TIFF)

S4 Fig. Number of flowers discovered by a group of bees as a function of flower density, when accounting for flower depletion. Here we assumed that an individual will continue exploring after visiting an already-explored flower, and will return to the nest only when encountering a fresh flower or after a fixed number of visits to already-visited flowers ($F_{\text{de-pleted}}$). **(A)** Number of different flowers discovered in 100 exploration trips of 900 s, in an

environment with randomly distributed flowers, and with F_{depleted}. = 1 (note that this value of F_{depleted} makes the simulation identical to that in the main text). Results are averaged over 80 simulations, keeping the environment fixed for every simulation. Solid lines: With masking effect. Dotted lines: Without masking effect (the bee does not react to previous flower encounters, and all discovered flowers were counted). Colors correspond to different flower sizes. **(B)** Same as A but with $F_{\text{depleted}} = 2$. **(C)** Same as A but with $F_{\text{depleted}} = 4$. **(D)** Same as A but with $F_{depleted} = 8$. **(E)** Same as A but with $F_{\text{depleted}} = \infty$. **(F)** Same as A but for a given flower size (50 cm), with colors representing the value of F_{depleted}.
(TIFF)

S1 Text. Raw results and figures.
(PDF)

S1 Data. Data and code sources for analysis and simulations.
(ZIP)

Acknowledgments

We thank Thibault Dubois, Tamara Gómez Moracho, Cristian Pasquaretta, Joe Woodgate, James Makinson, Joanna Brebner and Lars Chittka for sharing their data of bumblebee flight tracks using radar.

Author Contributions

Conceptualization: Ana Morán, Mathieu Lihoreau, Jacques Gautrais.

Data curation: Ana Morán, Jacques Gautrais.

Formal analysis: Ana Morán, Jacques Gautrais.

Funding acquisition: Ana Morán, Mathieu Lihoreau, Jacques Gautrais.

Methodology: Ana Morán, Alfonso Pérez-Escudero.

Project administration: Alfonso Pérez-Escudero.

Software: Ana Morán, Jacques Gautrais.

Supervision: Mathieu Lihoreau, Alfonso Pérez-Escudero, Jacques Gautrais.

Writing – original draft: Ana Morán, Alfonso Pérez-Escudero.

Writing – review & editing: Ana Morán, Mathieu Lihoreau, Alfonso Pérez-Escudero, Jacques Gautrais.

References

1. Mayer C, Adler L, Armbruster WS, Dafni A, Eardley C, Huang SQ, et al. Pollination ecology in the 21st Century: Key questions for future research. J Pollinat Ecol. 2011 Mar 19;8–23.
2. Buchmann SL, Nabhan GP. The Pollination Crisis. The Sciences. 1996; 36(4):22–7.
3. Goulson D, Nicholls E, Botías C, Rotheray EL. Bee declines driven by combined stress from parasites, pesticides, and lack of flowers. Science. 2015 Mar 27; 347(6229):1255957. https://doi.org/10.1126/science.1255957 PMID: 25721506
4. Lenz F, Chechkin AV, Klages R. Constructing a Stochastic Model of Bumblebee Flights from Experimental Data. Aegerter CM, editor. PLoS ONE. 2013 Mar 8; 8(3):e59036. https://doi.org/10.1371/journal.pone.0059036 PMID: 23520551
5. Reynolds AM, Smith AD, Menzel R, Greggers U, Reynolds DR, Riley JR. Displaced honey bees perform optimal scale-free search flights. Ecology. 2007 Aug; 88(8):1955–61. https://doi.org/10.1890/06-1916.1 PMID: 17824426

6. Reynolds AM, Smith AD, Reynolds DR, Carreck NL, Osborne JL. Honeybees perform optimal scale-free searching flights when attempting to locate a food source. J Exp Biol. 2007 Nov 1; 210(21):3763–70. https://doi.org/10.1242/jeb.009563 PMID: 17951417

7. Reynolds AM. Lévy flight patterns are predicted to be an emergent property of a bumblebees' foraging strategy. Behav Ecol Sociobiol. 2009 Nov; 64(1):19–23.

8. Ohashi K, Thomson JD, D'Souza D. Trapline foraging by bumble bees: IV. Optimization of route geometry in the absence of competition. Behav Ecol. 2007 Jan; 18(1):1–11.

9. Pyke GH, Cartar RV. The Flight Directionality of Bumblebees: Do They Remember Where They Came from? Oikos. 1992; 65(2):321–7.

10. Pyke GH. Optimal foraging in bumblebees and coevolution with their plants. Oecologia. 1978 Jan 1; 36 (3):281–93. https://doi.org/10.1007/BF00348054 PMID: 28309915

11. Lihoreau M, Chittka L, Raine NE. Monitoring Flower Visitation Networks and Interactions between Pairs of Bumble Bees in a Large Outdoor Flight Cage. Ollerton J, editor. PLOS ONE. 2016 Mar 16; 11(3): e0150844. https://doi.org/10.1371/journal.pone.0150844 PMID: 26982030

12. Vallaeys V, Tyson RC, Lane WD, Deleersnijder E, Hanert E. A Lévy-flight diffusion model to predict transgenic pollen dispersal. J R Soc Interface. 2017 Jan 31; 14(126):20160889.

13. Collett M, Chittka L, Collett TS. Spatial Memory in Insect Navigation. Curr Biol. 2013 Sep 9; 23(17): R789–800. https://doi.org/10.1016/j.cub.2013.07.020 PMID: 24028962

14. Woodgate JL, Makinson JC, Lim KS, Reynolds AM, Chittka L. Life-Long Radar Tracking of Bumblebees. Pratt SC, editor. PLOS ONE. 2016 Aug 4; 11(8):e0160333. https://doi.org/10.1371/journal.pone.0160333 PMID: 27490662

15. Brebner JS, Makinson JC, Bates OK, Rossi N, Lim KS, Dubois T, et al. Bumble bees strategically use ground level linear features in navigation. Anim Behav. 2021 Sep 1; 179:147–60.

16. Menzel R, Lehmann K, Manz G, Fuchs J, Koblofsky M, Greggers U. Vector integration and novel shortcutting in honeybee navigation. Apidologie. 2012 May 1; 43(3):229–43.

17. Lihoreau M, Raine NE, Reynolds AM, Stelzer RJ, Lim KS, Smith AD, et al. Radar Tracking and Motion-Sensitive Cameras on Flowers Reveal the Development of Pollinator Multi-Destination Routes over Large Spatial Scales. Collett T, editor. PLoS Biol. 2012 Sep 20; 10(9):e1001392. https://doi.org/10.1371/journal.pbio.1001392 PMID: 23049479

18. Woodgate JL, Makinson JC, Lim KS, Reynolds AM, Chittka L. Continuous Radar Tracking Illustrates the Development of Multi-destination Routes of Bumblebees. Sci Rep. 2017 Dec; 7(1):17323. https://doi.org/10.1038/s41598-017-17553-1 PMID: 29230062

19. Lihoreau M, Chittka L, Raine NE. Travel Optimization by Foraging Bumblebees through Readjustments of Traplines after Discovery of New Feeding Locations. Am Nat. 2010 Dec; 176(6):744–57. https://doi.org/10.1086/657042 PMID: 20973670

20. Lihoreau M, Chittka L, Le Comber SC, Raine NE. Bees do not use nearest-neighbour rules for optimization of multi-location routes. Biol Lett. 2012 Feb 23; 8(1):13–6. https://doi.org/10.1098/rsbl.2011.0661 PMID: 21849311

21. Reynolds AM, Lihoreau M, Chittka L. A Simple Iterative Model Accurately Captures Complex Trapline Formation by Bumblebees Across Spatial Scales and Flower Arrangements. Ayers J, editor. PLoS Comput Biol. 2013 Mar 7; 9(3):e1002938. https://doi.org/10.1371/journal.pcbi.1002938 PMID: 23505353

22. Dubois T, Pasquaretta C, Barron AB, Gautrais J, Lihoreau M. A model of resource partitioning between foraging bees based on learning. PLOS Comput Biol. 2021 Jul 28; 17(7):e1009260. https://doi.org/10.1371/journal.pcbi.1009260 PMID: 34319987

23. Capaldi EA, Smith AD, Osborne JL, Fahrbach SE, Farris SM, Reynolds DR, et al. Ontogeny of orientation flight in the honeybee revealed by harmonic radar. Nature. 2000 Feb; 403(6769):537–40. https://doi.org/10.1038/35000564 PMID: 10676960

24. Edwards AM, Phillips RA, Watkins NW, Freeman MP, Murphy EJ, Afanasyev V, et al. Revisiting Lévy flight search patterns of wandering albatrosses, bumblebees and deer. Nature. 2007 Oct; 449 (7165):1044–8.

25. Benhamou S. How Many Animals Really Do the Lévy Walk? Ecology. 2007; 88(8):1962–9.

26. Ohashi K, Leslie A, Thomson JD. Trapline foraging by bumble bees: V. Effects of experience and priority on competitive performance. Behav Ecol. 2008; 19(5):936–48.

27. Lihoreau M, Chittka L, Raine NE. Trade-off between travel distance and prioritization of high-reward sites in traplining bumblebees: Distance reward trade-off in bees. Funct Ecol. 2011 Dec; 25(6):1284–92.

28. Stout JC. Does size matter? Bumblebee behaviour and the pollination of Cytisus scoparius L. (Fabaceae). Apidologie. 2000 Jan; 31(1):129–39.

29. Osborne JL, Smith A, Clark SJ, Reynolds DR, Barron MC, Lim KS, et al. The Ontogeny of Bumblebee Flight Trajectories: From Naïve Explorers to Experienced Foragers. Smagghe G, editor. PLoS ONE. 2013 Nov 12; 8(11):e78681.

30. Bressloff PC. Target competition for resources under multiple search-and-capture events with stochastic resetting: Target competition and resetting. Proc R Soc Math Phys Eng Sci. 2020; 476(2242).

31. Basu U, Majumdar SN, Rosso A, Schehr G. Active Brownian motion in two dimensions. Phys Rev E. 2018 Dec; 98(6):062121.

32. Singh P, Kundu A, Majumdar SN, Schawe H. Mean area of the convex hull of a run and tumble particle in two dimensions. J Phys Math Theor. 2022 Jun; 55(22):225001.

33. Caprini L, Sprenger AR, Löwen H, Wittmann R. The parental active model: A unifying stochastic description of self-propulsion. J Chem Phys. 2022 Feb; 156(7):071102. https://doi.org/10.1063/5.0084213 PMID: 35183083

34. Pototsky A, Stark H. Active Brownian particles in two-dimensional traps. Europhys Lett. 2012 Jun; 98 (5):50004.

35. Dauchot O, Démery V. Dynamics of a Self-Propelled Particle in a Harmonic Trap. Phys Rev Lett. 2019; 122(6):1–5.

36. Basu U, Majumdar SN, Rosso A, Schehr G. Long-time position distribution of an active Brownian particle in two dimensions. Phys Rev E. 2019; 100(6):62116. https://doi.org/10.1103/PhysRevE.100.062116 PMID: 31962395

37. Malakar K, Das A, Kundu A, Kumar KV, Dhar A. Steady state of an active Brownian particle in a two-dimensional harmonic trap. Phys Rev E. 2020; 101(2):1–10.

38. Santra I, Basu U, Sabhapandit S. Direction reversing active Brownian particle in a harmonic potential. Soft Matter. 2021; 17(44):10108–19. https://doi.org/10.1039/d1sm01118a PMID: 34726222

39. Caprini L, Marini Bettolo Marconi U. Inertial self-propelled particles. J Chem Phys. 2021 Jan; 154 (2):024902. https://doi.org/10.1063/5.0030940 PMID: 33445896

40. Chaudhuri D, Dhar A. Active Brownian particle in harmonic trap: exact computation of moments, and re-entrant transition. J Stat Mech Theory Exp. 2021 Jan; 2021(1):013207.

41. Gupta S, Jayannavar AM. Stochastic Resetting: A (Very) Brief Review. Front Phys. 2022; 10(April):1–16.

42. Evans MR, Majumdar SN. Diffusion with Stochastic Resetting. Phys Rev Lett. 2011 Apr; 106 (16):160601. https://doi.org/10.1103/PhysRevLett.106.160601 PMID: 21599344

43. Kumar V, Sadekar O, Basu U. Active Brownian motion in two dimensions under stochastic resetting. Phys Rev E. 2020; 102(5):1–19. https://doi.org/10.1103/PhysRevE.102.052129 PMID: 33327209

44. Masó-Puigdellosas A, Campos D, Méndez V. Transport properties of random walks under stochastic noninstantaneous resetting. Phys Rev E. 2019; 100(4):1–8. https://doi.org/10.1103/PhysRevE.100.042104 PMID: 31770871

45. Gupta D, Plata CA, Kundu A, Pal A. Stochastic resetting with stochastic returns using external trap. J Phys Math Theor. 2021 Jan; 54(2):025003.

46. Gupta D, Pal A, Kundu A. Resetting with stochastic return through linear confining potential. J Stat Mech Theory Exp. 2021 Apr; 2021(4):043202.

47. Lindsay AE, Spoonmore RT, Tzou JC. Hybrid asymptotic-numerical approach for estimating first-passage-time densities of the two-dimensional narrow capture problem. Phys Rev E. 2016 Oct; 94 (4):042418. https://doi.org/10.1103/PhysRevE.94.042418 PMID: 27841558

48. Schumm RD, Bressloff PC. Search processes with stochastic resetting and partially absorbing targets. J Phys Math Theor. 2021 Oct; 54(40):404004.

49. Bressloff PC. Asymptotic analysis of extended two-dimensional narrow capture problems. Proc R Soc Math Phys Eng Sci. 2021 Feb; 477(2246):20200771.

50. Bressloff PC. Asymptotic Analysis of Target Fluxes in the Three-Dimensional Narrow Capture Problem. Multiscale Model Simul. 2021 Jan; 19(2):612–32.

51. Chechkin A, Sokolov IM. Random Search with Resetting: A Unified Renewal Approach. Phys Rev Lett. 2018; 121(5):50601. https://doi.org/10.1103/PhysRevLett.121.050601 PMID: 30118268

52. Mercado-Vásquez G, Boyer D, Majumdar SN, Schehr G. Intermittent resetting potentials. J Stat Mech Theory Exp. 2020 Nov; 2020(11):113203.

53. Pal A, Kuśmierz Lukasz, Reuveni S. Search with home returns provides advantage under high uncertainty. Phys Rev Res. 2020; 2(4):1–15.

54. Evans MR, Majumdar SN, Schehr G. Stochastic resetting and applications. J Phys Math Theor. 2020; 53(19).

55. Masó-Puigdellosas A, Campos D, Méndez V. Transport properties and first-arrival statistics of random motion with stochastic reset times. Phys Rev E. 2019; 99(1):1–9. https://doi.org/10.1103/PhysRevE.99.012141 PMID: 30780220

56. Weiss GH. Some applications of persistent random walks and the telegrapher's equation. Phys Stat Mech Its Appl. 2002 Aug; 311(3–4):381–410.

57. Khuong A, Lecheval V, Fournier R, Blanco S, Weitz S, Bezian JJ, et al. How Do Ants Make Sense of Gravity? A Boltzmann Walker Analysis of Lasius niger Trajectories on Various Inclines. Pratt SC, editor. PLoS ONE. 2013 Oct 29; 8(10):e76531. https://doi.org/10.1371/journal.pone.0076531 PMID: 24204636

58. Weitz S, Blanco S, Fournier R, Gautrais J, Jost C, Theraulaz G. Modeling Collective Animal Behavior with a Cognitive Perspective: A Methodological Framework. de Polavieja GG, editor. PLoS ONE. 2012 Jun; 7(6):e38588. https://doi.org/10.1371/journal.pone.0038588 PMID: 22761685

59. Weitz S, Blanco S, Fournier R, Gautrais J, Jost C, Theraulaz G. Residence times and boundary-following behavior in animals. Phys Rev E. 2014 May; 89(5):052715. https://doi.org/10.1103/PhysRevE.89.052715 PMID: 25353837

60. Gautrais J, Jost C, Soria M, Campo A, Motsch S, Fournier R, et al. Analyzing fish movement as a persistent turning walker. J Math Biol. 2009 Mar; 58(3):429–45. https://doi.org/10.1007/s00285-008-0198-7 PMID: 18587541

61. Gautrais J, Ginelli F, Fournier R, Blanco S, Soria M, Chaté H, et al. Deciphering Interactions in Moving Animal Groups. Levin SA, editor. PLoS Comput Biol. 2012 Sep; 8(9):e1002678. https://doi.org/10.1371/journal.pcbi.1002678 PMID: 23028277

62. Degond P, Motsch S. Large Scale Dynamics of the Persistent Turning Walker Model of Fish Behavior. J Stat Phys. 2008 Jun; 131(6):989–1021.

63. Du G, Kumari S, Ye F, Podgornik R. Model of metameric locomotion in smooth active directional filaments with curvature fluctuations. Europhys Lett. 2021 Dec; 136(5):58003.

64. Du G, Ye F, Podgornik R. Trapping instability of an active particle in steering potential fields. Phys Rev Res. 2022; 4(4):1–6.

65. Szabados T. An elementary introduction to the Wiener process and stochastic integrals [Internet]. arXiv; 2010 [cited 2022 Sep 2]. Available from: http://arxiv.org/abs/1008.1510

66. Cattiaux P, Chafai D, Motsch S. Asymptotic analysis and diffusion limit of the Persistent Turning Walker Model. Asymptot Anal. 2010 Mar; 67(1–2):17–31.

67. Weber C, Radtke PK, Schimansky-Geier L, Hänggi P. Active motion assisted by correlated stochastic torques. 2011 [cited 2022 Sep 9]; Available from: https://arxiv.org/abs/1105.3363

68. Daltorio KA, Tietz BR, Bender JA, Webster VA, Szczecinski NS, Branicky MS, et al. A model of exploration and goal-searching in the cockroach, Blaberus discoidalis. Adapt Behav. 2013 Oct; 21(5):404–20.

69. Nötel J, Sokolov IM, Schimansky-Geier L. Diffusion of active particles with stochastic torques modeled as α-stable noise. J Phys Math Theor. 2017 Jan 20; 50(3):034003.

70. Heinze S, Narendra A, Cheung A. Principles of Insect Path Integration. Curr Biol. 2018 Sep 10; 28(17):R1043–58. https://doi.org/10.1016/j.cub.2018.04.058 PMID: 30205054

71. Pasquaretta C, Dubois T, Gomez-Moracho T, Delepoulle VP, Le Loc'h G, Heeb P, et al. Analysis of temporal patterns in animal movement networks. Methods Ecol Evol. 2021; 12(1):101–13.

72. Riley JR, Smith AD, Reynolds DR, Edwards AS, Osborne JL, Williams IH, et al. Tracking bees with harmonic radar. Nature. 1996 Jan; 379(6560):29–30.

73. Osborne JL, Martin AP, Carreck NL, Swain JL, Knight ME, Goulson D, et al. Bumblebee flight distances in relation to the forage landscape. J Anim Ecol. 2008; 77(2):406–15. https://doi.org/10.1111/j.1365-2656.2007.01333.x PMID: 17986207

74. Von Frisch K. Die Tänze der Bienen. In: Von Frisch K, editor. Tanzsprache und Orientierung der Bienen [Internet]. Berlin, Heidelberg: Springer; 1965 [cited 2022 Aug 30]. p. 3–330. Available from: https://doi.org/10.1007/978-3-642-94916-6_2

75. Becher MA, Grimm V, Knapp J, Horn J, Twiston-Davies G, Osborne JL. BEESCOUT: A model of bee scouting behaviour and a software tool for characterizing nectar/pollen landscapes for BEEHAVE. Ecol Model. 2016 Nov 24; 340:126–33.

76. Kapustjansky A, Chittka L, Spaethe J. Bees use three-dimensional information to improve target detection. Naturwissenschaften. 2010 Feb; 97(2):229–33. https://doi.org/10.1007/s00114-009-0627-5 PMID: 19960178

77. Klein S, Pasquaretta C, Barron AB, Devaud JM, Lihoreau M. Inter-individual variability in the foraging behaviour of traplining bumblebees. Sci Rep. 2017 Jul 4; 7(1):4561. https://doi.org/10.1038/s41598-017-04919-8 PMID: 28676725

78. Chittka L, Dyer AG, Bock F, Dornhaus A. Bees trade off foraging speed for accuracy. Nature. 2003 Jul; 424(6947):388–388.

Appendix B

Simulation code

UnNid.h

```c
#ifndef Un_Nid
#define Un_Nid

#include "Geometrie.h"

/* Definition types */

typedef struct{
    /*/ Description du nid/*/
    UneCoordonnee position;
    double orientationSortie;

    double taille;
    double distancePerception;

} UnNid;

/* Methodes */

void UnNid_setPosition(UnNid *ceNid, UneCoordonnee cettePosition);
UneCoordonnee UnNid_getPosition(UnNid *ceNid);

void UnNid_setOrientationSortie(UnNid *ceNid, double cetteDirection);
double UnNid_getOrientationSortie(UnNid *ceNid);

void UnNid_setTaille(UnNid *ceNid, double cetteTaille, double
    cetteAcuiteVisuelle);
double UnNid_getTaille(UnNid *ceNid);

double UnNid_getDistancePerception(UnNid *ceNid);

void UnNid_setLeNid(UnNid *ceNid, UneCoordonnee  cettePosition, double
    cetteTaille, double cetteDirection, double cetteAcuiteVisuelle);


#endif
```

UnNid.c

```c
/* implementations des fonctions */
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <gsl/gsl_randist.h>

#include "UnNid.h"
#define anglePerception (3.0*M_PI/180.0)


//La position
```

```c
void UnNid_setPosition(UnNid *ceNid, UneCoordonnee cettePosition)
{
    ceNid->position = cettePosition;
}

UneCoordonnee UnNid_getPosition(UnNid *ceNid)
{
    return ceNid->position;
}

//L'orientation
void UnNid_setOrientationSortie(UnNid *ceNid, double cetteDirection)
{
    ceNid->orientationSortie = cetteDirection;
}

double UnNid_getOrientationSortie(UnNid *ceNid)
{
    return ceNid->orientationSortie;
}

//Taille
void UnNid_setTaille(UnNid *ceNid, double cetteTaille, double
   cetteAcuiteVisuelle)
{
    ceNid->taille = cetteTaille;
    ceNid->distancePerception = (ceNid->taille/2)/tan(cetteAcuiteVisuelle
       /2);
}

double UnNid_getTaille(UnNid *ceNid)
{
     return (ceNid->taille);
}

//Distance de perception
double UnNid_getDistancePerception(UnNid *ceNid)
{
    return (ceNid -> distancePerception);
}

//Initialisation Fleur
void UnNid_setLeNid(UnNid *ceNid, UneCoordonnee  cettePosition, double
   cetteTaille, double cetteDirection, double cetteAcuiteVisuelle)
{
    UnNid_setPosition(ceNid,cettePosition);
    UnNid_setOrientationSortie(ceNid, cetteDirection);
    UnNid_setTaille(ceNid,cetteTaille,cetteAcuiteVisuelle);
}
```

UneFleur.h

```c
#ifndef Une_Fleur
#define Une_Fleur

#include "Geometrie.h"

typedef struct{
    /*/ Description de la fleur/*/
    UneCoordonnee position;
     double taille;
     double distancePerception;

} UneFleur;

/* Methodes */

void UneFleur_setPosition(UneFleur *cetteFleur, UneCoordonnee
   cettePosition);
UneCoordonnee UneFleur_getPosition(UneFleur *cetteFleur);

void UneFleur_setTaille(UneFleur *cetteFleur, double cetteTaille);
double UneFleur_getTaille(UneFleur *cetteFleur);

void UneFleur_setDistancePerception(UneFleur *cetteFleur, double
   visualAcuity);
double UneFleur_getDistancePerception(UneFleur *cetteFleur);

void UneFleur_setLaFleur(UneFleur *cetteFleur, UneCoordonnee
   cettePosition, double cetteTaille, double visualAcuity);

#endif
```

UneFleur.c

```c
/* implementations des fonctions */
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <gsl/gsl_randist.h>

#include "UneFleur.h"

// Position
void UneFleur_setPosition(UneFleur *cetteFleur, UneCoordonnee
   cettePosition)
{
     cetteFleur->position = cettePosition;
}

UneCoordonnee UneFleur_getPosition(UneFleur *cetteFleur)
{
     return (cetteFleur->position);
}
```

```
19
20 //Taille
21 void UneFleur_setTaille(UneFleur *cetteFleur, double cetteTaille)
22 {
23      cetteFleur->taille = cetteTaille;
24 }
25 double UneFleur_getTaille(UneFleur *cetteFleur)
26 {
27      return (cetteFleur->taille);
28 }
29
30 //Distance de perception
31 void UneFleur_setDistancePerception(UneFleur *cetteFleur, double
       visualAcuity)
32 {
33     cetteFleur->distancePerception = (cetteFleur->taille/2)/tan(
          visualAcuity/2);
34 }
35
36 double UneFleur_getDistancePerception(UneFleur *cetteFleur)
37 {
38     return (cetteFleur -> distancePerception);
39 }
40
41 //Initialisation Fleur
42 void UneFleur_setLaFleur(UneFleur *cetteFleur, UneCoordonnee
       cettePosition, double cetteTaille, double visualAcuity)
43 {
44     UneFleur_setPosition(cetteFleur,cettePosition);
45     UneFleur_setTaille(cetteFleur,cetteTaille);
46     UneFleur_setDistancePerception(cetteFleur, visualAcuity);
47 }
```

UnBourdon.h

```
1  #ifndef Un_Bourdon
2  #define Un_Bourdon
3
4  #include "Geometrie.h"
5  // #include "UnBourdonInit.h"
6
7
8  /* Definition types */
9  typedef struct struct_bb_init
10 {
11     // sims params
12     double  speed;
13     double  deltaT;
14     double  visualAcuity;
15
16     // state variables
17     double  X, Y;
```

```
18      double  theta;
19      double  omega;
20
21      // behav parms
22      double  varianceOmega;
23      double  alpha;
24      double  gamma;
25      double  eta;
26
27  } UnBourdonInit ;
28
29
30  typedef struct{
31
32      // sims params
33       double speed;
34       double deltaT;
35       double acuiteVisuelle;
36
37      // state variables
38       UneCoordonnee position;
39       double theta;
40       double omega;
41
42      // behav parms
43       double varianceOmega;
44       double alpha;
45       double  gamma;
46       double eta;
47
48       /* accessory variable */
49       double sepsilon ;
50       double expGammaDt;
51       double phi;
52       double wStar;
53
54  } UnBourdon;
55
56
57  /* Methodes */
58
59  void UnBourdon_Init(UnBourdon *ceBourdon, UnBourdonInit cetInit);
60  void UnBourdon_setBehav(UnBourdon *ceBourdon, double ceAlpha, double
        ceOmega, double ceGamma);
61  void UnBourdon_Reset(UnBourdon *ceBourdon, UneCoordonnee cettePosition,
        double ceTheta, double ceOmega);
62  void UnBourdon_setEta(UnBourdon *ceBourdon, double ceEta);
63
64
65  void UnBourdon_NextPosition(UnBourdon *leBourdon, UneCoordonnee *
        cetEndroitCible, gsl_rng *rng);
66
67
68  // access
```

```
69 UneCoordonnee UnBourdon_getPosition(UnBourdon *ceBourdon);
70 void UnBourdon_setPosition(UnBourdon *ceBourdon, UneCoordonnee
      cetttePosition);
71
72 double UnBourdon_getOmega(UnBourdon *ceBourdon);
73 double UnBourdon_getEta(UnBourdon *ceBourdon);
74 double UnBourdon_getTheta(UnBourdon *ceBourdon);
75 void UnBourdon_setTheta(UnBourdon *ceBourdon,double ceTheta);
76
77 void UnBourdon_echo(UnBourdon *ceBourdon);
78
79 #endif
```

UnBourdon.c

```
1  #include <stdio.h>
2  #include <math.h>
3  #include <stdlib.h>
4  #include <gsl/gsl_randist.h>
5
6  #include "UnBourdon.h"
7
8
9  ////////////////////////////////
10 /* Fonctions d'initialisation */
11 ////////////////////////////////
12
13 void UnBourdon_updateAccessoryVariables(UnBourdon *ceBourdon)
14 {
15     ceBourdon->sepsilon = sqrt( ceBourdon->varianceOmega * (1-exp(-2*
          ceBourdon->gamma * ceBourdon->deltaT)) );
16     ceBourdon->expGammaDt = exp(-ceBourdon->gamma*ceBourdon->deltaT);
17 }
18
19 void UnBourdon_setDeltaT(UnBourdon *ceBourdon, double ceDeltaT)
20 {
21     ceBourdon->deltaT = ceDeltaT;
22     UnBourdon_updateAccessoryVariables(ceBourdon);
23 }
24
25 void UnBourdon_Init(UnBourdon *ceBourdon, UnBourdonInit cetInit)
26 {
27     // sims params
28     ceBourdon->speed = cetInit.speed;
29     ceBourdon->acuiteVisuelle = cetInit.visualAcuity;
30
31         // state variables
32     UneCoordonnee posInit = {cetInit.X,cetInit.Y};
33     UnBourdon_setPosition(ceBourdon, posInit);
34     ceBourdon->theta = cetInit.theta;
35     ceBourdon->omega = cetInit.omega;
36
```

```
37      // behav parms
38      UnBourdon_setEta(ceBourdon, cetInit.eta);
39      UnBourdon_setBehav(ceBourdon,cetInit.alpha,cetInit.varianceOmega,
            cetInit.gamma);
40
41      // accessory variable
42      UnBourdon_setDeltaT(ceBourdon,cetInit.deltaT);
43      ceBourdon->phi = 0;
44      ceBourdon->wStar = 0;
45  }
46
47  void UnBourdon_setBehav(UnBourdon *ceBourdon, double ceAlpha, double
        ceOmega, double ceGamma)
48  {
49      ceBourdon->alpha  = ceAlpha;
50      ceBourdon->varianceOmega = ceOmega;
51      ceBourdon->gamma = ceGamma;
52      UnBourdon_updateAccessoryVariables(ceBourdon);
53  }
54
55  void UnBourdon_Reset(UnBourdon *ceBourdon, UneCoordonnee cettePosition,
        double ceTheta, double ceOmega)
56  {
57      // state variables
58      ceBourdon->position = cettePosition;
59      ceBourdon->theta    = ceTheta;
60      ceBourdon->omega    = ceOmega;
61      // accessory variables
62      ceBourdon->phi      = 0;
63      ceBourdon->wStar    = 0;
64  }
65
66  void UnBourdon_setEta(UnBourdon *ceBourdon, double ceEta)
67  {
68      ceBourdon->eta = ceEta;
69  }
70
71  ////////////////////////////////////////////////
72  /* Fonctions pour calculer les variables d'etat */
73  ////////////////////////////////////////////////
74
75  void CalculPhi(UnBourdon *leBourdon, UneCoordonnee *targetPos)
76  {
77      double cosPhi ;
78      double sinPhi;
79      int signPhi;
80      double phi;
81
82      double costheta = cos(leBourdon->theta);
83      double sintheta = sin(leBourdon->theta);
84      double targetHeading = atan2(targetPos->y-leBourdon->position.y,
            targetPos->x-leBourdon->position.x);
85      double cosHeading = cos(targetHeading);
86      double sinHeading = sin(targetHeading);
```

```
87
88      cosPhi = costheta * cosHeading + sintheta * sinHeading;
89      if (cosPhi < -1.0) cosPhi = -1.0;
90      if (cosPhi > 1.0) cosPhi = 1.0;
91
92      sinPhi = costheta * sinHeading - sintheta * cosHeading;
93      signPhi=copysign(1, sinPhi);
94
95      leBourdon->phi = signPhi * acos(cosPhi);
96  }
97
98  void CalculOmega(UnBourdon *leBourdon,UneCoordonnee *cetEndroit, gsl_rng *
        rng)
99  {
100     CalculPhi(leBourdon, cetEndroit);
101     leBourdon->wStar = leBourdon->phi * leBourdon->eta;
102     leBourdon->omega = leBourdon->omega * leBourdon->expGammaDt +
            leBourdon->wStar * (1-leBourdon->expGammaDt) + gsl_ran_gaussian(rng
            , leBourdon->sepsilon);
103 }
104
105 ////////////////////////////
106 /* Fonctions pour avancer */
107 ////////////////////////////
108
109 void UnBourdon_NextPosition(UnBourdon *leBourdon, UneCoordonnee *
        cetEndroitCible, gsl_rng *rng)
110 {
111     leBourdon->position.x = leBourdon->position.x + leBourdon->speed * cos
            (leBourdon->theta) * leBourdon->deltaT;
112     leBourdon->position.y = leBourdon->position.y + leBourdon->speed * sin
            (leBourdon->theta) * leBourdon->deltaT;
113     CalculOmega(leBourdon, cetEndroitCible, rng);
114     leBourdon->theta = leBourdon->theta + leBourdon->omega * leBourdon->
            deltaT ;
115 }
116
117 ////////////////////////////////////////
118 /* Fonctions pour retourner des infos */
119 ////////////////////////////////////////
120
121 UneCoordonnee UnBourdon_getPosition(UnBourdon *ceBourdon)
122 {
123     return(ceBourdon->position);
124 }
125
126 void UnBourdon_setPosition(UnBourdon *ceBourdon, UneCoordonnee
        cetttePosition)
127 {
128     ceBourdon->position.x = cetttePosition.x;
129     ceBourdon->position.y = cetttePosition.y;
130 }
131
132 double UnBourdon_getOmega(UnBourdon *ceBourdon)
```

```
133 {
134     return(ceBourdon->omega);
135 }
136 
137 double UnBourdon_getEta(UnBourdon *ceBourdon)
138 {
139     return(ceBourdon->eta);
140 }
141 
142 double UnBourdon_getTheta(UnBourdon *ceBourdon)
143 {
144     return(ceBourdon->theta);
145 }
146 
147 void UnBourdon_setTheta(UnBourdon *ceBourdon,double ceTheta)
148 {
149     ceBourdon->theta = ceTheta;
150 }
151 
152 //////////
153 /* Echo */
154 //////////
155 void UnBourdon_echo(UnBourdon *ceBourdon)
156 {
157     fprintf(stderr,"\n Bourdon :\n\n");
158     fprintf(stderr," Speed : %g \n", ceBourdon->speed);
159     fprintf(stderr," Variance de omega : %g \n\n", ceBourdon->
            varianceOmega);
160     fprintf(stderr," Alpha : %g \n\n", ceBourdon->alpha);
161     fprintf(stderr," Position : x: %g  y: %g \n", ceBourdon->position.x,
            ceBourdon->position.y);
162     fprintf(stderr," Eta : %g \n", ceBourdon->eta);
163     fprintf(stderr," Theta : %g \n", ceBourdon->theta);
164     fprintf(stderr," Omega : %g \n", ceBourdon->omega);
165     fprintf(stderr,"------------------\n");
166 
167 }
```

UnEnvironnement.h

```
1  #ifndef Un_Environnement
2  #define Un_Environnement
3  
4  #include "UnNid.h"
5  #include "UneFleur.h"
6  #include "UnBourdon.h"
7  #define MaxNbDeFleurs 4096
8  
9  /* Definition types */
10 
11 typedef struct{
12      /*/ Le nid /*/
```

```
13      UnNid leNid;
14
15     /*/ Les fleurs /*/
16      int nombreFleurs;
17      UneFleur laFleur[MaxNbDeFleurs];
18
19 } UnEnvironnement;
20
21
22 /* Methodes */
23 void UnEnv_setPositionNid(UnEnvironnement *cetEnv, UneCoordonnee
       cettePosition);
24 UneCoordonnee UnEnv_getPositionNid(UnEnvironnement *cetEnv);
25
26 void UnEnv_setOrientationNid(UnEnvironnement *cetEnv, double
       cetteDirection);
27 double UnEnv_getOrientationNid(UnEnvironnement *cetEnv);
28
29 void UnEnv_setNid(UnEnvironnement *cetEnv, double cetteTaille,
       UneCoordonnee cettePosition, double cetteDirection, double
       cetteAcuiteVisuelle);
30
31 void UnEnv_ajouterLaFleur(UnEnvironnement *cetEnv, UneCoordonnee
       cettePosition, double cetteTaille, double visualAcuity);
32 int  UnEnv_getNombreDeFleurs(UnEnvironnement *cetEnv);
33
34 void UnEnv_setDefault(UnEnvironnement *cetEnv);
35
36 void UnEnv_echo(UnEnvironnement *cetEnv);
37
38 double UnEnv_distanceFleur(UneFleur laFleur, UnBourdon *ceBourdon);
39 int UnEnv_collisionFleur(UnEnvironnement *cetEnv, UnBourdon *ceBourdon);
       // -1 si pas de collision, n si collision avec fleur n
40
41 double UnEnv_distanceNid(UnNid leNid, UnBourdon *ceBourdon);
42 int UnEnv_collisionNid(UnEnvironnement *cetEnv, UnBourdon *ceBourdon); //
       0 si pas de collision, 1 si collision avec nid. collision = le bourdon
       voit le nid
43 int UnEnv_rentrerDansNid(UnEnvironnement *cetEnv, UnBourdon *ceBourdon);
       // 0 si le bourdon n'est pas dans le nid, 1 si il est dedans
44
45 #endif
```

UnEnvironnement.c

```
1 /* implementations des fonctions */
2 #include <stdio.h>
3 #include <math.h>
4 #include <stdlib.h>
5 #include <gsl/gsl_randist.h>
6
7 #include "UnEnvironnement.h"
```

```
8  #include "Geometrie.h"
9  #include "UnBourdon.h"
10
11 /* Ajouter un nid */
12 //position
13 void UnEnv_setPositionNid(UnEnvironnement *cetEnv, UneCoordonnee
       cettePosition)
14 {
15     UnNid_setPosition(&(cetEnv->leNid), cettePosition);
16 }
17
18 UneCoordonnee UnEnv_getPositionNid(UnEnvironnement *cetEnv)
19 {
20     return UnNid_getPosition(&(cetEnv->leNid));
21 }
22
23 //direction
24 void UnEnv_setOrientationNid(UnEnvironnement *cetEnv, double
       cetteDirection)
25 {
26     UnNid_setOrientationSortie(&(cetEnv->leNid), cetteDirection);
27 }
28
29 double UnEnv_getOrientationNid(UnEnvironnement *cetEnv)
30 {
31     return UnNid_getOrientationSortie(&(cetEnv->leNid));
32 }
33
34 //taille
35 void UnEnv_setNid(UnEnvironnement *cetEnv, double cetteTaille,
       UneCoordonnee cettePosition, double cetteDirection, double
       cetteAcuiteVisuelle)
36 {
37     UnNid_setLeNid(&(cetEnv->leNid), cettePosition, cetteTaille,
           cetteDirection, cetteAcuiteVisuelle);
38 }
39
40 /* Ajouter des fleurs */
41 void UnEnv_ajouterLaFleur(UnEnvironnement *cetEnv, UneCoordonnee
       cettePosition, double cetteTaille, double visualAcuity)
42 {
43     if (cetEnv->nombreFleurs == MaxNbDeFleurs)
44     {
45         fprintf(stderr,"\n\nERROR : Nb de Fleurs Trop Grand : changer dans
               UnEnvironnement.h\n\n");
46         exit(-1);
47     }
48     UneFleur_setLaFleur(&(cetEnv->laFleur[cetEnv->nombreFleurs]),
           cettePosition, cetteTaille, visualAcuity);
49     cetEnv->nombreFleurs++;
50 }
51
52 int  UnEnv_getNombreDeFleurs(UnEnvironnement *cetEnv)
53 {
```

```
54      return cetEnv->nombreFleurs;
55  }
56
57  /*Environnement par defaut */
58  void UnEnv_setDefault(UnEnvironnement *cetEnv)
59  {
60      UnEnv_setPositionNid(cetEnv, LeCentreDuRepere);
61      cetEnv -> nombreFleurs = 0;
62  }
63
64  /* Pour connaitre ce qu'il y a dans l'environnement */
65  void UnEnv_echo(UnEnvironnement *cetEnv)
66  {
67
68      fprintf(stderr,"\n Environnement :\n\n");
69      fprintf(stderr," Nid : x: %g  y: %g\n", cetEnv->leNid.position.x,
            cetEnv->leNid.position.y);
70      fprintf(stderr," Nombre de fleurs : %d\n", cetEnv->nombreFleurs);
71      fprintf(stderr," Position et taille des fleurs : \n");
72      for (int f=0; f<cetEnv->nombreFleurs; f++)
73      {
74          fprintf(stderr,"\t %d: x: %g y: %g taille: %g\n", f, cetEnv->
                laFleur[f].position.x, cetEnv->laFleur[f].position.y,cetEnv->
                laFleur[f].taille);
75      }
76      fprintf(stderr,"------------------\n");
77  }
78
79  /* Interactions Environnement - Bourdon */
80  double UnEnv_distanceFleur(UneFleur laFleur, UnBourdon *ceBourdon)
81  {
82      double distanceFleur;
83
84      distanceFleur = sqrt(pow(laFleur.position.x - ceBourdon->position.x,2)
            + pow(laFleur.position.y - ceBourdon->position.y,2));
85
86      return (distanceFleur);
87  }
88
89  double UnEnv_distanceFleur2(UneFleur laFleur, UnBourdon *ceBourdon)
90  {
91      double dx, dy;
92
93      dx = laFleur.position.x - ceBourdon->position.x;
94      dy = laFleur.position.y - ceBourdon->position.y;
95      dx *= dx ; dy *= dy ;
96      return (dx+dy);
97  }
98
99  int UnEnv_collisionFleur(UnEnvironnement *cetEnv, UnBourdon *ceBourdon)
100 {
101     int i;
102     for(i=0;i<cetEnv->nombreFleurs;i++)
103     {
```

```
104       if (UnEnv_distanceFleur2(cetEnv->laFleur[i], ceBourdon) < cetEnv->
              laFleur[i].distancePerception*cetEnv->laFleur[i].
              distancePerception)
105          return i;
106      }
107      return -1;
108  }
109
110  double UnEnv_distanceNid(UnNid leNid, UnBourdon *ceBourdon)
111  {
112      double distanceNid;
113
114      distanceNid = sqrt(pow(leNid.position.x - ceBourdon->position.x,2) +
             pow(leNid.position.y - ceBourdon->position.y,2));
115
116      return (distanceNid);
117  }
118
119  double UnEnv_distanceNid2(UnNid leNid, UnBourdon *ceBourdon)
120  {
121      double dx,dy;
122      dx = leNid.position.x - ceBourdon->position.x;
123      dy = leNid.position.y - ceBourdon->position.y;
124      dx *= dx ; dy *= dy;
125      return (dx+dy);
126  }
127
128  int UnEnv_collisionNid(UnEnvironnement *cetEnv, UnBourdon *ceBourdon)
129  {
130       if (UnEnv_distanceNid2(cetEnv->leNid, ceBourdon) < cetEnv->leNid.
              distancePerception*cetEnv->leNid.distancePerception)
131       {
132          return 1;
133       } else {
134          return 0;
135       }
136  }
137
138  int UnEnv_rentrerDansNid(UnEnvironnement *cetEnv, UnBourdon *ceBourdon)
139  {
140      if (UnEnv_distanceNid(cetEnv->leNid, ceBourdon) < cetEnv->leNid.taille
             )
141       {
142          return 1;
143       } else {
144          return 0;
145       }
146  }
```

main.c

```
1  #include <stdio.h>
```

```
2  #include <math.h>
3  #include <stdlib.h>
4  #include <gsl/gsl_randist.h>
5
6  #include "UnBourdon.h"
7  #include "UnEnvironnement.h"
8
9
10 gsl_rng *rng;
11
12 /* les objets */
13 UnEnvironnement lEnviron;
14 UnBourdon leBourdon;
15 unsigned int maxTimeStep;
16 unsigned int modulo3sec;
17
18 #include "params.h"
19
20 /* pour l'output */
21 unsigned int returnToNest, outOfTime;
22
23
24 void init()
25 {
26     fprintf(stderr, " *********  Init *********  \n");
27
28     /////// Initialisation du rng
29     gsl_rng_env_setup();
30   rng = gsl_rng_alloc(gsl_rng_knuthran2); // Creer le random number
       generator
31
32   UnBourdon_Init(&leBourdon, leBourdonInit);
33
34   UnEnv_setDefault(&lEnviron);
35   UnEnv_setNid(&lEnviron, NestSize , Coordonnee(0,0) , NestOrientation,
       leBourdon.acuiteVisuelle);
36
37   maxTimeStep = (unsigned int)ceil(TmaxLoop/leBourdon.deltaT) + 1;
38   modulo3sec = (unsigned int)floor(3/leBourdon.deltaT);
39
40   // output in tsv
41   printf("X\tY\n");
42 }
43
44 void run()
45 {
46   fprintf(stderr, " *********  RUN! *********  \n");
47
48   UneCoordonnee laPositionCible = UnEnv_getPositionNid(&lEnviron);
49
50   unsigned int timeStep,timeToGoNest,timeout;
51   UneCoordonnee currentPos;
52
53   for (int laBoucle = 0 ; laBoucle<sampleLoops ; laBoucle++ )
```

```
54    {
55
56      UnBourdon_Init(&leBourdon, leBourdonInit);
57      UnBourdon_setTheta(&leBourdon,gsl_ran_flat(rng, 0.0, M_PI));
58      timeToGoNest = (unsigned int) floor(gsl_ran_exponential(rng, leBourdon
           .alpha)/leBourdon.deltaT);
59
60
61      timeStep=0;timeout=0;
62      while (UnEnv_collisionNid(&lEnviron, &leBourdon) == 0)
63      {
64        if ( timeStep % modulo3sec == 0 )
65        {
66          currentPos = UnBourdon_getPosition(&leBourdon);
67          printf("%f\t%f\n",currentPos.x,currentPos.y);
68        }
69        if (timeStep>maxTimeStep-1)
70        {
71          timeout=1;
72          break;
73        }
74        if (timeStep==timeToGoNest+1) UnBourdon_setEta(&leBourdon,SwitchEta)
             ;
75        UnBourdon_NextPosition(&leBourdon, &laPositionCible, rng);
76
77        timeStep++;
78      } // while
79    } // for laBoucle
80  }
81
82  void end()
83  {
84  }
85
86
87  int main(int argc, char *argv[])
88  {
89    init();
90    run();
91    end();
92  }
```