In-silico Behavior Discovery System: An Application of Planning in Ethology

Haibo Wang[1], Hanna Kurniawati[1], Surya Singh[1,2], and Mandyam Srinivasan[1,2]
{h.wang10, hannakur, spns, m.srinivasan}@uq.edu.au
[1] School of Information Technology and Electrical Engineering [2] Queensland Brain Institute
The University of Queensland, Brisbane, Australia

Abstract

It is now widely accepted that a variety of interaction strategies in animals achieve optimal or near optimal performance. The challenge is in determining the performance criteria being optimized. A difficulty in overcoming this challenge is the need for a large body of observational data to delineate hypotheses, which can be tedious and time consuming, if not impossible. To alleviate this difficulty, we propose a system — termed "*in-silico* behavior discovery" — that will enable ethologists to simultaneously compare and assess various hypotheses with much less observational data. Key to this system is the use of Partially Observable Markov Decision Processes (POMDPs) to generate an optimal strategy under a given hypothesis. POMDPs enable the system to take into account imperfect information about the animals' dynamics and their operating environment. Given multiple hypotheses and a set of preliminary observational data, our system will compute the optimal strategy under each hypothesis, generate a set of synthesized data for each optimal strategy, and then rank the hypotheses based on the similarity between the set of synthesized data generated under each hypothesis and the provided observational data. In particular, this paper considers the development of this approach for studying mid-air collision-avoidance strategies of honeybees. To perform a feasibility study, we test the system using 100 data sets of close encounters between two honeybees. Preliminary results are promising, indicating that the system independently identifies the same hypothesis (optical flow centering) as discovered by neurobiologists/ethologists.

Introduction

What are the underlying strategies that animals take when interacting with other animals? This is a fundamental question in *ethology*. Aside from human curiosity, the answer to such a question may hold the key to significant technological advances. For instance, understanding how birds avoid collisions may help develop more efficient collision avoidance techniques for Unmanned Aerial Vehicles (UAVs), while understanding how cheetahs hunt may help develop better conservation management programs.

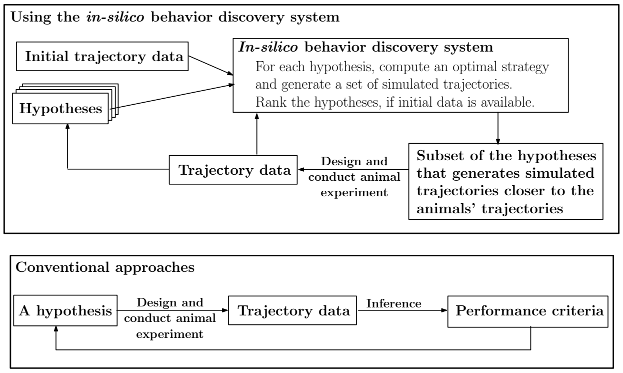


Figure 1: *In-silico* and conventional approaches as considered for studying honeybee collision avoidance.

Although it is now widely accepted that a variety of interaction strategies in animals have been shaped to achieve optimal or near optimal performance (Breed and Moore 2012; Davies and Krebs 2012), determining the exact performance criteria that are being optimized remains a challenge. Existing approaches require ethologists to infer the criteria being optimized from many observations on how the animals interact. These approaches present two main difficulties. First, the inference is hard to do, because even extremely different performance criteria may generate similar observed data under certain scenarios. Second, obtaining observational data are often difficult; some interactions rarely occur. For instance, to understand collision avoidance strategies in insects and birds, it is necessary to observe many near-collision encounters, but such events are rare because these animals are very adept at avoiding collisions. Ethologists can resort to experiments that deliberately cause close-encounter events, but such experiments are tedious, time-consuming, and may not faithfully capture the properties of the natural environment. Furthermore, such experiments may not be possible for animals that have become extinct, such as dinosaurs, in which case ethologists can only rely on limited historical data, such as fossil tracks.

This paper presents our preliminary work in developing a system — termed "*in-silico* behavior discovery" — to enable ethologists study animals' strategies by simultaneously comparing and assessing various performance criteria on the basis of limited observational data. The difference between an approach using our system and conventional approaches is illustrated in Figure 1.

Our system takes as many hypotheses as the user choose to posit, and data from preliminary experiments. Preliminary data are collision-avoidance encounter scenarios, and each encounter consists of a set of flight trajectories of all honeybees involved in one collision-avoidance scenario. Each hypothesis is a performance criterion that may govern the honeybees' collision-avoidance strategies. For each hypothesis, the system generates an optimal collision-avoidance strategy and generates simulated collision-avoidance trajectories based on that. Given the set of simulated collision-avoidance trajectories, the system will rank the hypotheses based on the similarity between the set of simulated trajectories and the preliminary trajectory data. By being able to generate simulated data under various hypotheses, the *in-silico* behavior discovery system enables ethologists to "extract" more information from the available data and better focus their subsequent data gathering effort, thereby reducing the size of exploratory data required to find the right performance criteria that explain the collision avoidance behavior of honeybees. This iterative process is illustrated in Figure 1

Obviously, a key question is how to generate the collision-avoidance strategy under a given performance criterion. In our system, we use the Partially Observable Markov Decision Processes (POMDPs) framework. One may quickly argue that it is highly unlikely an insect such as a bee runs a POMDP solver in its brain. This may be true, but the purpose of our system is *not* to mimic honeybee neurology. Rather, we use the widely accepted idea in biology — i.e., most interaction strategies in animals achieve optimal or near optimal performance — to develop a *tool that helps ethologists predict and visualize their hypotheses prior to conducting animal experiments to test the hypotheses*, thereby helping them to design more focused and fruitful animal experiments. In fact, POMDPs allow us to relax the need to model the exact flight dynamics and perception of the honeybees. No two animals are exactly alike, even though they are of the same species. This uniqueness causes variations in various parameters critical to generate the strategies. For instance, some honeybees have better vision than others, enabling them to sense impending collisions more accurately and hence avoid collisions more often, different honeybees have different wing beat frequencies causing varying manoeuvrability, etc. Our system frames these variations as stochastic uncertainties — commonly used modelling in analysing group behavior — and takes them into account when computing the optimal collision-avoidance strategy under a given performance criterion.

We tested the feasibility of our *in-silico* behavior discovery system using a data set comprising 100 close encounter scenarios between two honeybees. The results indicate that the system independently identifies the same hypothesis (optical flow centering) as discovered by neurobiologists/ethologists.

POMDP Background & Related Work

A POMDP model is defined by a tuple $\langle S, A, O, T, Z, R, \gamma, b_0 \rangle$, where S is a set of states, A is a set of actions, and O is a set of observations. At each time step, the POMDP agent is at a state $s \in S$, performs an action $a \in A$, and perceives an observation $o \in O$. A POMDP represents the uncertainty in the effect of performing an action as a conditional probability function, called the transition function, $T = f(s' \mid s, a)$, with $f(s' \mid s, a)$ representing the probability the agent moves from state s to s' after performing action a. Uncertainty in sensing is represented as a conditional probability function $Z = g(o \mid s', a)$, where $g(o \mid s', a)$ represents the probability the agent perceives observation $o \in O$ after performing action a and ending at state s'.

At each step, a POMDP agent receives a reward $R(s, a)$, if it takes action a from state s. The agent's goal is to choose a sequence of actions that will maximize its expected total reward, while the agent's initial belief is denoted as b_0. When the sequence of actions may have infinitely many steps, we specify a discount factor $\gamma \in (0, 1)$, so that the total reward is finite and the problem is well defined.

The solution of a POMDP problem is an *optimal policy* that maximizes the agent's expected total reward. A policy $\pi \colon B \to A$ assigns an action a to each belief $b \in B$, and induces a value function $V(b, \pi)$ which specifies the expected total reward of executing policy π from belief b. The value function is computed as

$$V(b, \pi) = E[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) | b, \pi] \tag{1}$$

To execute a policy π, a POMDP agent executes an action selection and a belief update repeatedly. Suppose the agent's current belief is b. Then, it selects the action referred to by $a = \pi(b)$, performs action a and receives an observation o according to the observation function Z. Afterwards, the agent updates b to a new belief b' given by

$$\begin{aligned} b'(s') &= \tau(b, a, o) \\ &= \eta Z(s', a, o) \int_{s \in S} T(s, a, s') ds \end{aligned} \tag{2}$$

where η is a normalization constant.

A more detailed review of the POMDP framework is available in (Kaelbling, Littman, and Cassandra 1998).

Although computing the optimal policy is computationally intractable (Papadimitriou and Tsitsiklis 1987), results over the past decade have shown that approximating optimality provides speed (Pineau, Gordon, and Thrun 2003; Smith and Simmons 2005; Kurniawati, Hsu, and Lee 2008; Silver and Veness 2010), a POMDP can start becoming practical for various real world problems (Bai et al. 2012; Horowitz and Burdick 2013; Koval, Pollard, and Srinivasa 2014; Williams and Young 2007).

To the best of our knowledge, the in-silico behavior discovery system is the first one that applies planning under uncertainty to help ethologists reduce the number of necessary observational data. This work significantly expands on (Wang et al. 2013). It uses POMDP to generate a near-optimal collision avoidance strategies of honeybees under a given hypothesis, and lets biologists observe the simulated trajectories, manually. In contrast, this work proposes a system that takes multiple hypotheses at once and provides a ranking of how likely the hypotheses generate the observational data.

The *In-silico* Behavior Discovery System

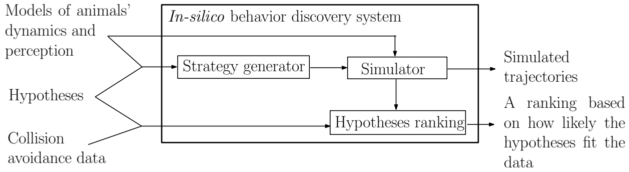

Models of animals' dynamics and perception
Hypotheses
Collision avoidance data
In-silico behavior discovery system
Strategy generator
Simulator
Hypotheses ranking
Simulated trajectories
A ranking based on how likely the hypotheses fit the data

Figure 2: The inputs, outputs, and main components of the proposed *in-silico* behavior discovery system.

An *in-silico* behavior discovery system is advanced for studying the collision avoidance strategies of honeybees Figure 2. This takes as input the animals' flight dynamics and perception models, a set of hypotheses on the performance criteria used by honeybees to avoid mid-air collision, and a set of collision avoidance trajectories of honeybees. These trajectories are usually small in number and act as preliminary observational data. The system computes the optimal collision avoidance strategy for each hypothesis. It outputs a set of simulated trajectories under each strategy, along with a ranking on which hypotheses are more likely to explain the preliminary data. The ranking is based on the similarity between the simulated trajectories of the hypotheses and the preliminary data. The system consists of three main modules:

- *Strategy Generator*, which computes the optimal strategy under each hypothesis.
- *Simulator*, which generates the simulated trajectories under each strategy that has been computed by the Strategy Generator module.
- *Hypothesis Ranking*, which identifies the hypotheses that are more likely to explain the observational data. For each hypothesis, the strategy generator and the simulator modules generate the simulated trajectories under the hypothesis. Once the sets of simulated trajectories have been generated for all hypotheses, the hypothesis ranking module will rank the hypotheses based on the similarity between the simulated trajectories and the observational data.

The details of each module are described in the following sub-sections.

Strategy Generator and Simulator

The strategy generator module is essentially a POMDP planner that generates an optimal collision avoidance strategy under hypothesized performance criteria used by honeybees to avoid mid-air collisions in various head-on encounters. Since this paper focuses only on head-on encounters, the number of honeybees involved in each encounter is only two. In this work, we also assume that the bees do not communicate/negotiate when avoiding collision. This assumption is in-line with the prevailing view in the relatively open question of whether bees actually negotiate for avoiding collision. Furthermore, it simplifies our POMDP model in the sense that it suffices to model each bee independently as a single POMDP agent, rather than all bees at once as a multi-agent system.

The POMDP framework is used to model a honeybee "agent" that tries to avoid collisions with another honeybee, assuming the agent optimizes the hypothesized performance criteria. The flying dynamics and perception models become the transition and observation functions of the POMDP model, while each hypothesis is represented as a reward function of the POMDP model. POMDPs enable the system to take into account variations in the honeybees' flight dynamics, for instance due to their weight and wingspan, or variations in the honeybees' perceptive capacities, and captures the agent's uncertainty about the behavior of the other bee.

One may argue that even the best POMDP planner today will not achieve the optimal solution to our problem within reasonable time. This is true, but a near optimal solution is often sufficient. Aside from results in ethology that indicate animals often use near optimal strategies too (Breed and Moore 2012; Davies and Krebs 2012), our system can help focus subsequent animal experiments as long as the strategy is sufficient to correctly identify which hypotheses are more likely to be correct, based on the similarity of the simulated trajectories under the hypotheses and the trajectories from real data. In many cases, we can correctly identify such hypotheses without computing the optimal collision-avoidance strategies, as we will show in our Results section.

Another critique of using the POMDP framework is that POMDPs require Markov assumption that is unlikely to be true in bees' motion. However, POMDPs are Markovian in the belief space. Since beliefs are sufficient statistics of the entire history, a POMDP agent, and hence our simulated bees, selects the best actions by considering the entire history of actions and observations. POMDP does require the transition function to be Markovian. However, this can often be satisfied by suitable design of the state and action space. In this work, we assume bees are kinematic — a commonly used simplification in modelling complex motion — where the next position and velocity are determined by the current position, velocity, and acceleration. More details on this model are discussed in subsequent paragraphs.

Our POMDP model is an adaptation of the POMDP model (Bai et al. 2012) designed for the Traffic Alert and Collision Avoidance System (TCAS) — a collision avoidance system mandatory for all large commercial aircraft. One would argue that this model is not suitable because the flight dynamics and perception model of aircraft are totally different than those of honeybees. Indeed their dynamics and perception are different. However, the model in (Bai et al. 2012) is a highly abstracted flight dynamics and perception model of aircraft, such that if we apply the same level of abstraction to the flight dynamics (simplified to its kinematic model) and perception of honeybees (simplified to visibility sensors), we would get a similar model, albeit with different parameters. We adjust the parameters based on the literature and data on the flight dynamics and perception capabilities of honeybees.

We describe the POMDP model here together with the required parameter adjustment. Although our POMDP will only control one of the honeybees involved in the close-encounter scenarios, the position, heading, and velocity of the two honeybees determine the collision avoidance strategy. Therefore, the state space S consists of the joint flight state spaces of the two honeybees involved. A flight state of a honeybee is specified as (x, y, z, θ, u, v), where (x, y, z) is the 3D position of the bee, θ is the bee's heading angle with respect to the positive direction of X axis, u is the bee's horizontal speed, and v is the bee's vertical speed.

The action space A represents the control parameters of only one of the honeybees. It is a joint product of vertical acceleration a and turn rate ω. Since most practical POMDP solvers (Kurniawati, Hsu, and Lee 2008; Silver and Veness 2010) today only perform well when the action space is small, we use bang-bang controller, restricting the acceleration a to be $\{-a_m, 0, a_m\}$ and the turning rate ω to be $\{-\omega_m, 0, \omega_m\}$, where a_m and ω_m are the maximum vertical acceleration and the maximum turn rate, respectively. Although the control inputs are continuous, restricting their values to extreme cases is reasonable because under the danger of near mid-air collisions, a bee is likely to maximize its manoeuvring in order to escape to a safe position as fast as possible. And control theory has shown that maximum-minimum (bang-bang) control yields time-optimal solutions under many scenarios (Latombe 1991). We assume that the other bee —whose action is beyond the control of the POMDP agent— has the same possible control parameters as the POMDP agent. However, which control it uses at any given time is unknown and is modelled as a uniform distribution over the possible control parameters.

The transition function represents an extremely simplified flight dynamics. Each bee is treated as a point mass. And given a control (a, ω), the next flight state of an animal after a small time duration Δt is given by

$$\begin{array}{rclrcl} x_{t+1} & = & x_t + u_t \Delta t cos\theta, & \theta_{t+1} & = & \theta_t + \omega \Delta t, \\ y_{t+1} & = & y_t + u_t \Delta t sin\theta, & u_{t+1} & = & u_t, \\ z_{t+1} & = & z_t + v_t \Delta t, & v_{t+1} & = & v_t + a \Delta t. \end{array}$$

Although a honeybee's perception is heavily based on optical flow (Si, Srinivasan, and Zhang 2003; Srinivasan 2011), to study the collision avoidance behavior, we can abstract its perception to the level of where it thinks the other bee is, i.e., the perception after all sensing data has been processed into information about its environment. Therefore, we can model the bee's observation space in terms of a sensor that has a limited field of view and a limited range.

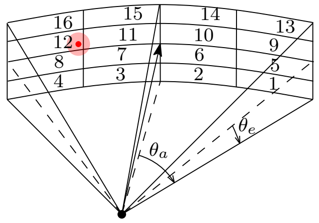


Figure 3: The observation model. The black dot is the position of the agent; the solid arrow is the agent's flying direction. The red dot is the position of the incoming honeybee. Due to bearing and elevation errors, our agent may perceive the other bee to be at any position within the shaded area.

The observation space O is a discretization of the sensor's field of view. The discretization is done on the elevation and azimuth angles such that it results in 16 equally spaced bins along the elevation and azimuth angles. Figure 3 illustrates this discretization. The observation space O is then these bins plus the observation NO-DETECTION, resulting in 17 observations in total.

As long as the incoming animal comes into the agent's sensor range (denoted as D_R) and into the visible space, it appears in a certain observation grid, with some uncertainty. The observation function models the uncertainty in bearing and elevation, as well as false positives and false negatives.

The parameters for the observation model are:

- Range limit, parameterized as D_R.
- Azimuth limit, parameterized as θ_a.
- Elevation limit, parameterized as θ_e.
- Bearing error standard deviation, parameterized as σ_b.
- Elevation error standard deviation, parameterized as σ_e.
- False positive probability, parameterized as p_{fp}.
- False negative probability, parameterized as p_{fn}.

The reward function will be different for different hypotheses of the performance criteria used by the honeybees in avoiding collision.

A POMDP simulator is used, in the sense that it takes the POMDP model and policy as inputs, and then generates the collision avoidance trajectories of the bee under various head-on encounter scenarios. The scenarios we use in the simulator are similar to the encounter scenarios in the real trajectories. Recall that a collision-avoidance trajectory is a set of flight trajectories of all the honeybees involved in the encounter scenario. In this work, only two honeybees are involved in each scenario, as we focus on head-on encounters. Our simulator uses similar encounter scenarios as the real data, in the sense that we only simulate the collision avoidance strategies of one of the two honeybees, while the other bee follows the flight trajectory of the real data. Therefore, each set of collision-avoidance trajectories generated by our simulator will have a one-to-one mapping with the set of real collision-avoidance trajectories that has been given to the system. For statistical significance, in general, our system generates multiple sets of simulated trajectories.

Hypotheses Ranking

The key in this module is the metric used to identify the similarity between a set of simulated collision-avoidance trajectories and a set of real collision-avoidance trajectories. Recall that each set of simulated collision avoidance trajectories has a one-to-one mapping with the set of real collision-avoidance trajectories. Let us denote this mapping by g. Suppose $\mathscr{A}$ is a set of simulated collision-avoidance trajectories and $\mathscr{B}$ is the set of real collision-avoidance trajectories given as input to the system. We define the similarity $sim(\mathscr{A}, \mathscr{B})$ between $\mathscr{A}$ and $\mathscr{B}$ as a 3-tuple $\langle \overline{F}, \overline{M}, C \rangle$, where:

- The notation $\overline{F}$ is the average distance between the flight path of the simulated trajectories and that of the real trajectories. Suppose L is the number of trajectories in $\mathscr{A}$. Then, $\overline{F(\mathscr{A}, \mathscr{B})} = \frac{1}{L} \sum_{i=1}^{L} F(A_i, B_i)$ where

$F(A_i, B_i)$ denotes the **Fréchet Distance** between the curve traversed by the simulated bee in $A_i \in \mathscr{A}$ and the curve traversed by the corresponding bee in $B_i = g(A_i) \in \mathscr{B}$. In our system, each curve traversed by a bee is represented as a polygonal curve because, the trajectory generated by our simulator assumes discrete time (a property inherited from the POMDP framework).

The Fréchet Distance computes the distance between two curves, taking into account their course. A commonly used intuition to explain Fréchet Distance is based on an analogy of a person walking his dog. The person walks on one curve and the dog on the other curve. The Fréchet Distance is then the shortest leash that allows the dog and its owner to walk along their respective curves, from one end to the other, without backtracking (Chambers et al. 2008; 2010). Formally,

$$F(A_i, B_i) = \min_{\substack{\alpha[0,1] \to [0,N] \\ \beta[0,1] \to [0,M]}} \left(\max_{t \in [0,1]} dist\Big(A_i(\alpha(t)), B_i(\beta(t)) \Big) \right)$$

where $dist$ is the underlying distance metric in the honeybees' flight space. In our case, it is the Euclidean distance in $\mathbb{R}^3$. N and M are the number of segments in the polygonal curves A_i and B_i respectively. The function α is continuous with $\alpha(0) = 0$ and $\alpha(1) = N$ while β is continuous with $\beta(0) = 0$ and $\beta(1) = M$. These two functions are possible parameterizations of A_i and B_i.

- The notation $\overline{M}$ denotes the average absolute difference in **Minimum Encounter Distance** (*MED*). MED of a collision-avoidance trajectory computes the smallest Euclidean distance between the two honeybees, e.g., for a collision-avoidance trajectory A_i, $MED(A_i) = \min_{t=1}^{T} dist(A_i(t), A_i'(t))$ where T is the smallest last timestamp among the trajectories of the two honeybees, $A_i(t)$ and $A_i'(t)$ are the trajectories of bee-1 and bee-2 in A_i at time t respectively, and $dist$ is the Euclidean distance between the two positions. MED measures how close two honeybees can be during one encounter. Small $\overline{M}$ is a necessary condition for a simulated trajectory to resemble the real trajectory, in the sense that if the simulated trajectories of the incoming and the outgoing bees are similar to the observed trajectories, then the minimum encounter distance between the simulated incoming and outgoing bees should be similar to that of the observed trajectories.
- The notation C denotes the absolute difference in the **Collision Rate**. The *Collision rate* is defined as the percentage of the collision that occur. Small C is a necessary condition for a simulated trajectory to resemble the real trajectory, in the sense that if the simulated and the real bees have similar capabilities in avoiding collisions, then assuming the trajectories and environments are similar, the collision rate of the simulated and real bees should be similar.

For statistical analysis, in general, our system generates multiple sets of simulated collision-avoidance trajectories for each hypothesis. The goodness of the hypothesis in explaining the input data is then defined as the average 3-tuple metric over all sets of simulated trajectories generated by the system. Suppose the system generates K sets of simulated trajectories, e.g., $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_K$ for hypothesis H_1. Then the goodness of H_1 in explaining the input data is a 3-tuple where the first element is $\frac{1}{K} \sum_{i=1}^{K} \overline{F(\mathscr{A}_i, \mathscr{B})}$, the second element is $\frac{1}{K} \sum_{i=1}^{K} \overline{M(\mathscr{A}_i, \mathscr{B})}$, and the third element is $\frac{1}{K} \sum_{i=1}^{K} C(\mathscr{A}_i, \mathscr{B})$ where $\mathscr{B}$ is the set of real collision-avoidance trajectories that the system received as inputs. The order in the tuple acts as prioritization. The system assigns a higher rank to the hypothesis whose goodness value has the smaller first element. If the goodnesses of two hypotheses have a similar first element, i.e., they are the same with more than 95% confident based on student t-test hypothesis testing, then the second element becomes the determining factor, and so on. Now, this ranking system may not be totally ordered, containing conflict on the the ordering. When such a conflict is found, we apply the Kemeny-Young voting method (Kemeny 1959; Young 1988; Levin and Nalebuff 1995) to enforce a total ordering of the resulting ranking.

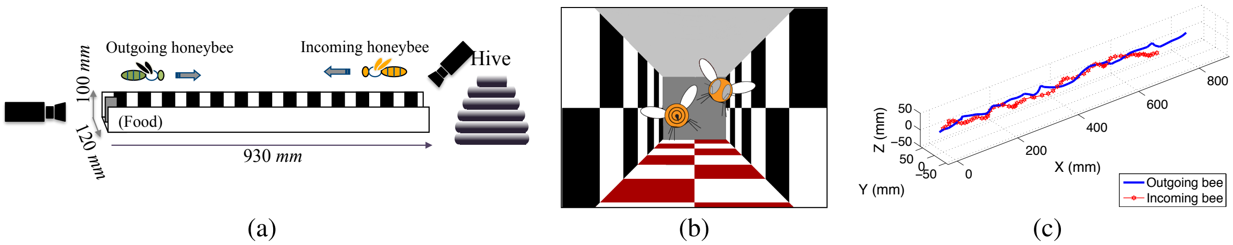
Note that although in this paper, there are only two honeybees involved in each collision-avoidance trajectory, it is straightforward to extend the aforementioned similarity metric and ranking strategy to handle encounter scenarios where many more honeybees are involved.

System Verification

To verify the system, we will use the system to rank several hypotheses in which the performance criterion closest to the correct one is known.

Collision-Avoidance Trajectories of Real Honeybees

To verify the applicability of our system, we use 100 sets of collision-avoidance trajectories as preliminary data.The data are gathered from experiments conducted at the Neuroscience of Vision and Aerial Robotics Laboratory in the Queensland Brain Institute. These data are the results of experimental recording of 100 head-on encounters of two honeybees flying along a 3-dimensional tunnel. Figure 4 illustrates the experimental setup to gather the data. The tunnel dimensions are $930mm \times 120mm \times 100mm$. The roof of the tunnel is transparent. The left, right, and bottom wall of the tunnel are covered with checkerboard patterns, where each square is of size $2.2cm \times 2.2cm$. The left and right patterns are colored black/white, while the bottom pattern is red/white, to aid the detection of the honeybees, which are generally dark in color. These patterns aid the honeybees' navigation through the tunnel. The tunnel is placed with its entrance near a beehive, and a sugar water feeder is placed inside the tunnel at its far end. To record a collision-avoidance trajectory, a bee is first released from the hive to the tunnel. This bee will fly towards the feeder, collect the food, and then fly back to the hive. When the bee starts to fly back to the hive, another bee is released from the hive to the tunnel and flies towards the feeder. We denote the bee flying towards the feeder as the incoming bee and the bee flying towards the hive as the outgoing bee.



(a) (b) (c)

Figure 4: Illustration of experimental design and setup for gathering collision-avoidance trajectories of real honeybees. These trajectories are used as an input (Initial Trajectory Data) to our system. (a) The tunnel. (b) The inner part of the tunnel. (c) A collision-avoidance trajectory gathered from this experiment.

The trajectories of the honeybees are recorded using two cameras —one positioned above the tunnel, looking down, and another camera positioned at the far end of the tunnel, looking axially into the tunnel. The stereo cameras capture the bees' flight at 25 frames per second. Based on the positioning and the resolution of the two stereo cameras, the estimated precision of the reconstructed 3D trajectories is approximately $2mm \times 2mm \times 2mm$.

Figure 4(c) shows the coordinate frame and one example of a collision-avoidance trajectories reconstructed in 3D. The possible coordinate values are $-30 \leq X \leq 900$, $-60 \leq Y \leq 60$, and $-50 \leq Z \leq 50$. Each collision-avoidance trajectory consists of the trajectories of the two honeybees, represented as a sequence of positions of the two honeybees. Each element of the sequence follows the following format $(x_1, y_1, z_1, x_2, y_2, z_2)$, where (x_1, y_1, z_1) is the position of the outgoing honeybees and (x_2, y_2, z_2) is the position of the incoming bee.

Hypotheses

To verify our system, we use six hypotheses as the input to our system. These hypotheses are selected in a way that we know exactly which hypotheses are closer to the correct performance criteria. Each hypothesis is represented as a reward function in the POMDP problem. It is essentially a summation of the component cost and reward. We will discuss the detailed values of all component costs and reward in the Simulation Setup section. The hypotheses, summarized in Table 1, are:

- H_{Basic} is the basic collision avoidance hypothesis. In this hypothesis, the reward function is the summation of collision cost and movement cost.
- $H_{BasicDest}$ is the hypothesis that the honeybees do not forget their goal of reaching the feeder or the hive, even though they have to avoid mid-air collision with another bee. In this hypothesis, we provide a high reward when the bee reaches its destination. The reward function is then the summation of the collision cost, the movement cost, and the reward for reaching the goal. This behavior is evident from the 100 collision-avoidance trajectories that were recorded. All trajectories indicate that the honeybees fly toward both ends of the tunnel, instead of wandering around within the tunnel or turning back before reaching their goals.
- H_{LR} is the hypothesis that the honeybees tend to perform horizontal centering. This is related to the optical flow matching nature of honeybee visual flight control (Si, Srinivasan, and Zhang 2003; Srinivasan et al. 1996). By optical flow we mean the observed visual gradient in time due to the relative motion (of the honeybee) and the objects in the scene. It has been shown that a honeybee navigates by matching the optical flow of the left and right eyes, which suggests that a honeybee has a mechanism and tendency to perform horizontal centering, but not one for vertical centering. This behavior is also visible in our 100 sets of honeybees' collision-avoidance trajectories. If we project all data points from the trajectories onto XY-plane, we find that the mean of all Y values is 1.13, with a standard deviation 12.49. A plot of this projection is shown in Figure 5.

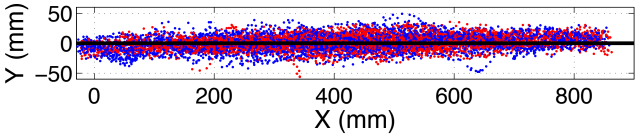


Figure 5: Data points of the 100 encounters are projected to XY-plane. Red points are projected data points from incoming honeybees, while blue points are projected data points from outgoing honeybees.

- H_{UD} is the hypothesis that the honeybees tend to perform vertical centering. This is actually an incorrect hypothesis we set to verify that the system can delineate bad hypotheses. The honeybees are actually biased to fly in the upper half of the tunnel because they are attracted to light. The transparent roof and solid bottom means more light is coming from the top. This behavior is evident from the 100 recorded collision-avoidance trajectories of honeybees. If we project all data points from the trajectories onto the XZ-plane, we find that 80% of the data points lies in the upper side of the tunnel. In fact, the Z values of all the data points have a mean of 12.89, a median of 15.67 and a standard deviation of 16.76, which again confirms the biased distribution toward the ceiling of the tunnel. A plot of this projection is shown in Figure 6.

Table 1: Hypotheses with the Corresponding Component Cost/Reward Functions

Penalties or Rewards	Hypotheses					
	H_{Basic}	$H_{BasicDest}$	H_{LRUD}	H_{LR}	H_{UD}	H_{LRDest}
Collision Cost	✓	✓	✓	✓	✓	✓
Movement Cost	✓	✓	✓	✓	✓	✓
LR-Penalty	—	—	✓	✓	—	✓
UD-Penalty	—	—	✓	—	✓	—
Destination Reward	—	✓	—	—	—	✓

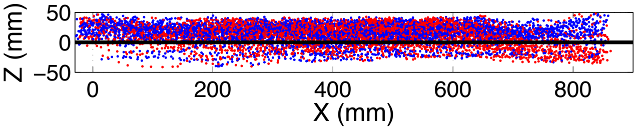


Figure 6: Data points of the 100 encounters, projected on to *XZ*-plane. Red points are projected data points from incoming honeybees, while blue points are projected data points from outgoing honeybees.

- H_{LRUD} is the combination of the previous two hypotheses: H_{LR} and H_{UD}. The reward function is the summation of collision cost, movement cost, the penalty cost for moving near the left and right walls, and the penalty cost for moving near the top and bottom walls.
- H_{LRDest} is the combination of H_{LR} and $H_{BasicDest}$, i.e., the reward function is the summation of collision cost, movement cost, the penalty cost for moving near the left and right walls, and the reward for reaching the goal. This hypothesis is the closest to the correct performance criteria, based on the existing literature.

Simulation Setup

We use POMCP (Silver and Veness 2010) to generate near optimal solutions to the POMDP problem that represents each of the hypotheses. POMCP is an online POMDP solver, which means it will plan for the best action to perform at each step, execute that action, and then re-plan. The on-line computation of POMCP helps to alleviate the problem with long planning horizon problem of this application — since bees can see relatively far, the collision-avoidance manoeuvring may happen far before the close encounter scenario actually happens. In our experiments, POMCP was run with 8,192 particles.

For each hypothesis, we generate 36 sets of collision-avoidance trajectories. Each of these sets of trajectories consists of 100 collision-avoidance trajectories, resulting in a total of 3,600 simulated collision-avoidance trajectory for each hypothesis. Each trajectory corresponds to exactly one of the encounter scenarios in the initial trajectory data gathered from the experiments with real honeybees. In each of the simulated collision-avoidance trajectories, our system generates the outgoing honeybee's trajectory based on the POMDP policy and sets the incoming bee to move following the incoming bee in the corresponding real collision-avoidance trajectory. All experiments are carried out on a Linux platform with a 3.6GHz Intel Xeon E5-1620 and 16GB RAM.

Now, we need to set the parameters for the POMDP problems. To this end, we derive the parameters based on the experimental setup used to generate the initial trajectory data (described in the previous subsection) and from the statistical analysis of the data.

For the control parameters, we take the median over the velocity and acceleration of honeybees in our data and set $u = 300mm/s$, $a_m = 562.5mm/s^2$, and $\omega_m = 375deg/s$.

For the observation model, since honeybees can see far and the length of the tunnel is less than one meter, we set the range limit to D_R to be infinite, to model the fact that the range limit of the bee's vision will not hinder its ability to see the other bee. The viewing angle of the honeybees remain limited. We set the azimuth limit θ_a to be 60 degrees and the elevation limit θ_e to be 60 degrees. The bearing error standard deviation σ_b and the elevation error standard elevation σ_e are both set to be 1 degree. We assume that the false positive probability p_{fp} and the false negative probability p_{fn} are both 0.01.

Following the definition used by ethologists, a state $s = (x_1, y_1, z_1, \theta_1, u_1, v_1, x_2, y_2, z_2, \theta_2, u_2, v_2) \in S$ is a collision state whenever the centre-to-centre distance between two parallel body axes is smaller than the wing span of the bee. Based on the ethologists' observations on average wingspan of a honeybee, we set this centre-to-centre distance to be 12mm. And we define a state to be in collision when the two honeybees are within a cross-section distance (in YZ-plane) of 12mm and an axial distance (in X-direction) of 5mm, i.e., $\sqrt{(y_1 - y_2)^2 + (z_1 - z_2)^2} \leq 12$ and $\|x_1 - x_2\| \leq 5$.

As for the reward functions, we assign the following component costs and rewards as follows:

- Collision cost: $-10,000$.
- Movement cost: -10.
- LR-Penalty:

$$R_{LR}(s) = \begin{cases} 0 & \text{if } |y_1| \leq 12, \\ -20 \times \frac{|y_1| - 12}{60 - 12} & \text{otherwise.} \end{cases}$$

- UD-Penalty:

$$R_{UD}(s) = \begin{cases} 0 & \text{if } |z_1| \leq 12, \\ -20 \times \frac{|z_1| - 12}{50 - 12} & \text{otherwise.} \end{cases}$$

- Destination reward: $+10,000$.

The numbers are set based on the ethologists intuition on how important a particular criteria is.

Table 2: Hypotheses with corresponding rankings, where 1 indicates the most promising hypothesis. The observational bee data has a *collision rate* of 0.030 and an averaged *MED* of 30.61. Each metric value is the absolute difference of the corresponding metric values between the hypothesis and Bee. The value is in the format of mean and 95% confidence interval. The units for $\overline{F}$ and $\overline{M}$ are mm.

Hypotheses	Goodness of the hypotheses			
	$Average(\overline{F})$	$Average(\overline{M})$	$Average(C)$	Ranking
H_{Basic}	147.19 ± 0.833	33.02 ± 0.250	0.023 ± 0.0023	6
H_{UD}	149.19 ± 1.142	26.83 ± 0.266	0.021 ± 0.0036	5
H_{LRUD}	126.59 ± 1.057	12.11 ± 0.279	0.001 ± 0.0065	4
H_{LR}	121.22 ± 0.784	13.29 ± 0.260	0.006 ± 0.0042	3
$H_{BasicDest}$	115.47 ± 0.594	15.79 ± 0.257	0.002 ± 0.0049	2
H_{LRDest}	115.27 ± 0.678	11.49 ± 0.254	0.007 ± 0.0056	1

One may argue that when ethologists have little understanding on the underlying animal behavior, setting the above values are impossible. Indeed setting the correct value is impossible. However, note that different cost and reward values can construct different hypothesis. And, one of the benefits of the system is exactly that the ethologists can simultaneously assess various hypotheses. Therefore, when ethologists have little understanding, they can construct many hypotheses with different cost and reward values, and then use our behavior discovery system to identify hypotheses that are more likely to explain the input data.

Results

Table 2 shows each component of the goodness of each hypotheses along with their 95% confidence intervals. It also shows the ranking of the hypotheses, where 1 means best.

The results indicate that the *in-silico* behavior discovery system can identify the best hypothesis, i.e., the hypothesis that represents the performance criteria closest to that of a honeybee avoiding mid-air collision, which is maintaining its position to be at the center horizontally and reaching its destination (H_{LRDest}).

The results show that the ranking does indicate the known behavior of the honeybees. For instance, H_{LR} is ranked higher than H_{LRUD} and H_{LRUD} is ranked higher than H_{UD}, which means that the system can identify that horizontal centering is a criteria the honeybees try to achieve, but vertical centering is not, which conform to the widely known results as discussed in the Hypotheses subsection.

Furthermore, $H_{BasicDest}$ is ranked higher than H_{Basic}, which indicates that the system does identify that honeybees tend to remain focus on reaching its destination even in head-on encounter scenarios, which conforms to the widely known results as discussed in the Hypotheses subsection.

Summary and Future Work

This paper presents an application of planning under uncertainty to help ethologists study the underlying performance criteria that animals try to optimize in an interaction. The main difficulty faced by ethologists is the need to gather a large body of observational data to delineate hypotheses, which can be tedious and time consuming, if not impossible. This paper introduces a system — termed "*in-silico* behavior discovery" — that enables ethologists to simultaneously compare and assess various hypotheses with much less observational data. Key to this system is the use of POMDPs to generate optimal strategies for various postulated hypotheses. Preliminary results indicate that, given various hypothesized performance criteria used by honeybees, our system can correctly identify and rank criteria according to how well their predictions fit the observed data. These results indicate that the system is feasible and may help ethologists in designing subsequent experiments or analysis that are much more focused, such that with a much smaller data set, they can reveal the underlying strategies in various animals' interaction. Such understanding may be beneficial to inspire the development of various technological advances.

Nature, additionally, involves a multi-objective optimization. Another strength of this approach is to help tease out the mixing of these objectives. For example, honeybee flight is regulated not only by optical flow, but also by overall illumination (i.e., phototaxis). As seen between the Central Tendency and Left/Right Central Tendency hypothesis test, the method can help clarify the weighting of the mixing (between optical flow and phototaxis). Another advantage of this approach is that it reduces the amount of experiments where one has to hold other secondary conditions (e.g., temperature, food sources, etc.) stationary, thus saving time and further aiding discovery of the underlying behaviours.

Many avenues for future work are possible. First, we want to control the behavior of both honeybees. This extension enables us to understand the possible "negotiation" or rule of thumb for manoeuvring that honeybees may use. Second, we would like to understand the capability of our system to model encounter scenarios that involve more than two honeybees. Third, we would like to refine the transition and perception model of the POMDP problem in our system, so that it reflects the unique characteristics of honeybees, such as the waggle dances it may perform to communicate with another honeybees. Finally, we are also interested in extending this system for other types of animal interaction, such as the hunting behavior of cheetahs and dinosaurs.

Acknowledgements

This work was funded by the ARC Centre of Excellence in Vision Science (Grant CEO561903), an ARC DORA grant

(DP140100914) and a Queensland Smart State Premiers Fellowship. We thank Julia Groening and Nikolai Liebsch for assistance with the collection of the experimental honeybee data.

References

Bai, H.; Hsu, D.; Kochenderfer, M.; and Lee, W. 2012. Unmanned aircraft collision avoidance using continuous-state POMDPs. *Robotics: Science and Systems VII* 1.

Breed, M. D., and Moore, J. 2012. *Animal Behavior*. Academic Press, Burlington, San Diego, London.

Chambers, E. W.; Colin de Verdière, É.; Erickson, J.; Lazard, S.; Lazarus, F.; and Thite, S. 2008. Walking your dog in the woods in polynomial time. In *Proceedings of the twenty-fourth annual symposium on Computational geometry*, 101–109. ACM.

Chambers, E. W.; Colin de Verdière, É.; Erickson, J.; Lazard, S.; Lazarus, F.; and Thite, S. 2010. Homotopic fréchet distance between curves or, walking your dog in the woods in polynomial time. *Computational Geometry* 43(3):295–311.

Davies, N. B., and Krebs, J. R. 2012. *An Introduction to Behavioural Ecology*. Wiley-Blackwell, Chichester.

Horowitz, M., and Burdick, J. 2013. Interactive Non-Prehensile Manipulation for Grasping Via POMDPs. In *ICRA*.

Kaelbling, L.; Littman, M.; and Cassandra, A. 1998. Planning and acting in partially observable stochastic domains. *AI* 101:99–134.

Kemeny, J. G. 1959. Mathematics without numbers. *Daedalus* 88(4):577–591.

Koval, M.; Pollard, N.; and Srinivasa, S. 2014. Pre- and post-contact policy decomposition for planar contact manipulation under uncertainty. In *Proc. Robotics: Science and Systems*.

Kurniawati, H.; Hsu, D.; and Lee, W. 2008. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Proc. Robotics: Science and Systems*.

Latombe, J. 1991. *Robot Motion Planning*. Kulwer Academic.

Levin, J., and Nalebuff, B. 1995. An introduction to vote-counting schemes. *The Journal of Economic Perspectives* 3–26.

Papadimitriou, C., and Tsitsiklis, J. 1987. The Complexity of Markov Decision Processes. *Math. of Operation Research* 12(3):441–450.

Pineau, J.; Gordon, G.; and Thrun, S. 2003. Point-based value iteration: An anytime algorithm for POMDPs. In *IJCAI*, 1025–1032.

Si, A.; Srinivasan, M. V.; and Zhang, S. 2003. Honeybee navigation: properties of the visually drivenodometer'. *Journal of Experimental Biology* 206(8):1265–1273.

Silver, D., and Veness, J. 2010. Monte-Carlo planning in large POMDPs. In *NIPS*.

Smith, T., and Simmons, R. 2005. Point-based POMDP algorithms: Improved analysis and implementation. In *UAI*.

Srinivasan, M.; Zhang, S.; Lehrer, M.; and Collett, T. 1996. Honeybee navigation en route to the goal: visual flight control and odometry. *Journal of Experimental Biology* 199(1):237–244.

Srinivasan, M. V. 2011. Honeybees as a model for the study of visually guided flight, navigation, and biologically inspired robotics. *Physiological Reviews* 91(2):389–411.

Wang, H.; Kurniawati, H.; Singh, S. P. N.; and Srinivasan, M. V. 2013. Animal Locomotion In-Silico: A POMDP-Based Tool to Study Mid-Air Collision Avoidance Strategies in Flying Animals. In *Proc. Australasian Conference on Robotics and Automation*.

Williams, J. D., and Young, S. 2007. Scaling POMDPs for spoken dialog management. *IEEE Trans. on Audio, Speech, and Language Processing*.

Young, H. P. 1988. Condorcet's theory of voting. *American Political Science Review* 82(04):1231–1244.