Animal Locomotion *In Silico*: A POMDP-Based Tool to Study Mid-Air Collision Avoidance Strategies in Flying Animals

Haibo Wang[1], Hanna Kurniawati[1], Surya Singh[1], and Mandyam Srinivasan[1,2]
[1]School of Information Technology and Electrical Engineering
[2]Queensland Brain Institute
The University of Queensland, Australia
{uqhwan15, hannakur, spns, m.srinivasan}@uq.edu.au

Abstract

In this paper, we bring modelling techniques from robotics to enable biologists perform an *in silico* study of mid-air collision avoidance strategies of flying animals. This *in silico* system is distinct from flying animal dynamics and trajectory simulation, as the focus is on the strategy behind the observed motions, rather than the specific motions in space. Our *in silico* system consists of a model and a simulator. To handle limited data and various variations in the flight dynamics and sensing parameters of the animals, we employ a Partially Observable Markov Decision Process (POMDP) framework —a general and principled approach for making decisions under uncertainty. Here, the solution to the POMDP problem is an optimal motion strategy to avoid mid-air collision with another animal. The system simulates the motion strategies in various head-on encounter scenarios. Preliminary results on comparing the simulated behaviours with 100 encounters from real honeybees are promising; the collision rate differs by less than 1%, while the difference in the minimum distance between two bees in 100 head-on encounters is on average around 12mm, which is roughly equivalent to the average wing span of the honeybees used to generate the data.

1 Introduction

Many robotic systems, from RoboBees [12] to BigDog [16] to RoboTuna [21], have benefited from a better understanding of animal motion. These theories and concepts of animal locomotion are developed based on observations over a large amount of data on the animals' motion. However, gathering such data is not always easy, especially when the manoeuvre under study seldom occurs, such as the mid-air collision avoidance of insects or the chase of a cheetah.

In this paper, we present our preliminary work in bringing modelling techniques from robotics to enable biologists perform an *in silico* study of underlying motion strategies. This is distinct from animal dynamics and trajectory simulation as the focus is on the decision making strategy behind the observed motions, rather than the specific motions in space. In this work, we model the animal under study as a decision making agent, and generate the best motion strategy assuming the animal is a rational agent that tries to maximize a certain objective function, such as avoiding collision with least effort or preying its lunch as fast as possible. We then use the motion strategy to generate simulated motions for the animal. Biologists can observe these simulated motions as if they are the motions of the animal being studied. This method of observation enables biologists to study various examples of motions that seldom occur. The model, the generated motion strategy, and the simulator make up the *in silico* system for studying motion strategies of certain animal behaviour.

A principled approach for decision making in the presence of limited and uncertain data and varying parameters, is the Partially Observable Markov Decision Process (POMDP) framework. This framework is well-suited for our purpose. Aside from the limited data, no two animals are exactly alike even though they are of the same species. This uniqueness causes variations in various parameters critical to generating rational motion strategies. For instance, some honeybees have better vision than others, enabling them to perceive possible collisions more accurately and hence avoid collisions more often, different honeybees have different wing beat frequencies causing varying levels of manoeuvrability, etc. These variations, while complex, are not random; indeed, animal morphology provides additional information on these uncertainties and their mean effects. As has been shown in various robotics domains [6, 7], POMDP provides a robust way to incorporate and reason about these uncertainties.

This paper adopts the POMDP framework to model the collision avoidance strategies of flying animals such as birds, bats, and bees, who seem to avoid mid-air collisions effortlessly even in incredibly dense situations and apparently without the complex structure and communications of civil systems such as the Traffic Alert and Collision Avoidance System (TCAS). While the dynamics of a bird and a plane are different, a comparison of animal strategies with TCAS might better inform the ongoing development of next generation TCAS systems. This is an active research area especially due to the recent progress in Unmanned Aerial Vehicles (UAVs) that spurred the need for a more reliable and robust TCAS that can handle more traffic [18].

Interestingly, POMDP framework is currently central to such efforts [8]. This coincidence is not accidental. Due to errors in sensing and control, an agent (e.g., pilot) may not know their exact state and the actions of the neighbouring entities. POMDP is designed to handle such types of uncertainty. Instead of finding the best action with respect to a single state, a POMDP solver finds the best action with respect to the set of states that are consistent with the available information so far. This set of states is represented as a probability distribution, called a belief b, and the set of all possible beliefs is called the belief space B. A POMDP solver calculates an optimal policy $\pi^* : B \rightarrow \mathcal{A}$ that maps a belief in B to an action in the set $\mathcal{A}$ of all possible actions the agent can perform, so as to maximize a given objective function. In TCAS, POMDP models the flying dynamics and sensing ability of an aircraft along with the errors and uncertainty of the system, to generate a robust collision-avoidance strategy for the aircraft.

Although solving a POMDP is computationally intractable in the worst case [14], recent developments of point-based POMDP approaches [3, 11, 15, 19] have drastically increased the speed of POMDP planning. Using sampling to trade optimality with approximate optimality for speed, point-based POMDP approach have moved POMDP framework from solving a 12 states problem in days to solving non-trivial problems with millions of states and even problems with 10 dimensional continuous state space within seconds to minutes [2, 6, 7, 9, 10]. This progress in POMDP solving is key to its recent adoption in TCAS [20], and to the feasibility of our proposed *in silico* system.

Leveraging this result, we adopt the POMDP model of TCAS and adjust the dynamics and sensing model to approximate those of flying animals. The solution to this POMDP problem is an optimal policy / motion strategy for the flying animal to avoid mid-air collision. We also develop a simulator that simulates motion strategies of the animal that uses the policy to avoid mid-air collision in various encounter scenarios. The encounter scenarios are generated based on data and information about flight plans of the flying animal under study. Biologists can then use the simulator to generate and observe various notions on how the flying animal avoid mid-air collision.

We have developed and tested our *in silico* system for characterising mid-air collision avoidance for honeybees. We have also compared the simulated bee motion generated by our system and the motion of 100 actual honeybees in avoiding mid-air collisions. Preliminary results are promising, with less than 1% difference in the collision rate, and an average difference of approximately 12mm in the minimum distance between two bees in 100 head-on encounters, which corresponds roughly to the average wingspan of the bees in our data.

Of course an *in silico* study of animal motion is no substitute for studying the motion of real animals. However, it may enable biologists to develop better initial hypotheses, and hence perform more focused and efficient studies on real animals, which can be much more costly and difficult compared to an *in silico* study.

2 Related Work

2.1 Motion Strategies for Mid-Air Collision Avoidance

Motion is a defining characteristic of an animal. Its analysis, however, is typically focused on the dynamics and loadings that drive the motion [1]. The decision making strategies behind these motions are typically made by using the observed trajectories [13] to determine gait model parameters that are then compared to hypothesized models and strategies that minimise energy or forces, for example.

In the case of mid-air flight steering and collision avoidance, analysis has ranged from Ros et al. [17] who studied manoeuvrability in pigeons to Groening et al. [5] who studied pairwise collision avoidance behaviour in bees flying through narrow tunnels. They discovered that bees actively avoid mid-air collisions when they are flying. Discovering such behaviour requires a large amount of data, which is often difficult to get. This work propose to alleviate such difficulty by developing an *in silico* system that generates trajectories similar to real animal trajectories, based on limited data and known information about the animal under study.

Mid-air collision avoidance is also of great interest to air traffic. Recent advancements in Unmanned Aerial Vehicles (UAVs) means heavier air traffic is expected in the near future, which spurred the need for more reliable and robust TCAS. One of the key issues in increasing TCAS' reliability and robustness is in taking into account the various uncertainty affecting pilots or UAVs in avoiding mid-air collision. Therefore, POMDP has been proposed [4] and successfully applied to improve the reliability and robustness of today's TCAS system [2, 20]. The POMDP model for TCAS provided a good starting point for our work.

2.2 Background on POMDP

Formally, a POMDP model is defined by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, T, Z, R, \gamma, b_0 \rangle$, where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, and $\mathcal{O}$ is a set of observations. At each time step, the POMDP agent is at a state $s \in \mathcal{S}$, performs an action $act \in \mathcal{A}$, and perceives an observation $o \in \mathcal{O}$. Due to errors in its controller and the partially observed world dynamics, the next state the agent might be in after performing an action is uncertain. This uncertainty is modeled as a conditional probability function $T = f(s' \mid s, act)$, with $f(s' \mid s, act)$ representing the probability the agent moves from state s to s' after performing action act. Uncertainty in sensing is represented as a conditional probability function $Z = g(o \mid s', act)$, where $g(o \mid s', act)$ represents the probability the agent perceives observation $o \in \mathcal{O}$ after performing action act and ends at state s'.

Furthermore at each step, the agent receives a reward $R(s, act)$, if it takes action act from state s. The agents goal is to choose a suitable sequence of actions that will maximize its expected total reward, while the agents initial belief is denoted as b_0. When the sequence of actions may have infinite length, 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 $\pi \colon B \rightarrow \mathcal{A}$ assigns an action act 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

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

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

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

where η is a normalization constant.

3 The Model

In this work, our goal is to create a simulated flying animal that mimics the behaviour of the real animal in avoiding mid-air collisions with another flying animal. We refer to our simulated animal as the *outgoing* animal, while the other animal as the *incoming* animal. To take into account our lack of information about the behaviour and about the motion and sensing capabilities of the outgoing and incoming animals, we model the outgoing animal as a POMDP agent that needs to avoid collision with the incoming animal whose flight plan is not perfectly known.

In particular, we adopt the POMDP model of TCAS [2]. This model is based on a very general and simplified flight dynamics and sensing model of airplanes, such that when we simplify the flying dynamics and sensing capabilities of the animals under study to a similar level of simplification used in [2], the set of parameters used to model the airplane dynamics and sensing in [2] are similar to those used for flying animals. Of course, the values of the parameters would be different, and need to be adjusted. We describe the model in this section, and discuss the required adjustments for honeybees in Section 5.

3.1 Flying Dynamics

The state space $\mathcal{S}$ of our POMDP model is a continuous space that represents the joint flight state spaces of the two animals. A flight state of each animal is specified as (x, y, z, θ, u, v), where (x, y, z) is the 3D position of the animal, θ is the animal's heading angle with respect to the positive direction of X axis, u is the animal's horizontal speed, and v is the animal's vertical speed (Figure 1 shows an illustration).

The action space $\mathcal{A}$ represents the control parameters of only the outgoing animal. It is a joint product of vertical acceleration a and turn rate ω. Considering the heavy computation cost of solving the POMDP model, we restrict a to be in $\{-a_m, 0, a_m\}$ and ω to be in $\{-\omega_m, 0, \omega_m\}$, where a_m and ω_m are the maximum vertical acceleration and the maximum turn rate, respectively. Although the control inputs are continu-

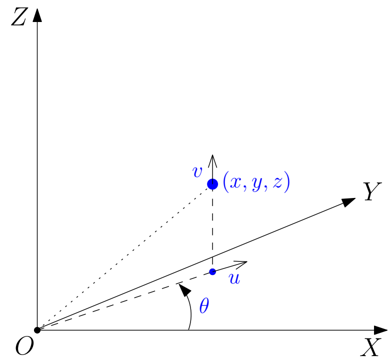

Figure 1: The flight state of one flying animal

ous, restricting their values to extreme cases is reasonable because when under the danger of near mid-air collisions, it is reasonable to assume that an animal will maximize its maneuvering in order to escape to a safe position. Figure 2 shows the 9 discrete actions. As the incoming animal's control inputs are unknown to the POMDP agent, we can either model them as uniformly randomized values, or as the controls of flying to some prescribed destinations, or based on information on the flight path of the animal under study.

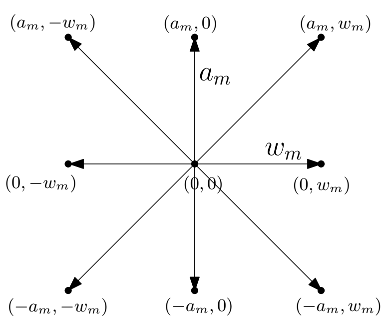

Figure 2: The action space contains 9 discrete actions.

We use a simplified model of flight dynamics in which each animal is treated as a point mass. Given a control (a, ω), the next flight state of an animal after a small time duration Δt is given by

$$\begin{aligned} x_{t+1} &= x_t + u_t \Delta t cos\theta, & \theta_{t+1} &= \theta_t + \omega \Delta t, \\ y_{t+1} &= y_t + u_t \Delta t sin\theta, & u_{t+1} &= u_t, \\ z_{t+1} &= z_t + v_t \Delta t, & v_{t+1} &= v_t + a\Delta t. \end{aligned} \tag{3}$$

Figure 3 demonstrates this transition process. In this model, we assume that during the encounter process, the horizontal speed is a constant.

3.2 Sensor Model

Although the outgoing animal has no prior information the incoming animal's flight path, it can noisily sense the location of the incoming animal. Given this noisy sensor input, the outgoing animal (i.e., the agent) manoeuvres to prevent near mid-air collisions by keeping a safe separation distance from the incoming animal.

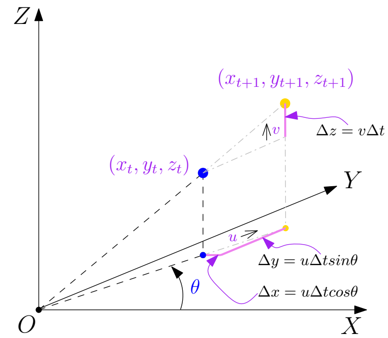

Figure 3: State transition from timestamp t to the next timestamp $t+1$, after a small time duration Δt.

We assume the animal has a visibility sensor with limited field of view and limited range. The field of view is limited in the elevation direction (both up and down) with a maximum elevation angle of θ_e, and is limited in the horizontal direction (both left and right) with a maximum azimuth of θ_a. The range limit is denoted as D_R.

The observation space $\mathcal{O}$ 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 its elevation and azimuth angles. Figure 4 illustrates this discretization. The observation space $\mathcal{O}$ is then these bins plus the observation `NO-DETECTION`, resulting in 17 observations in total.

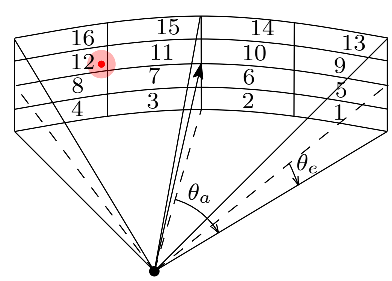

Figure 4: The sensor model. The black dot is the position of the agent; the solid arrow is the agent's flying direction. The red dot is the position of the incoming animal. Due to bearing error and elevation error, our agent may perceive the incoming animal at any position within the pink area.

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 cell.

For example, in Figure 4, the red dot represents the incoming bee, and it lies in 12. However, due to bearing error and elevation error, there will also be small probabilities that that agent observes the incoming animal to lie in cells 8, 7 and 11, respectively, and this brings uncertainties to the observation results. The bearing error is described by a normal distribution with zero mean and σ_b degree standard deviation; similarly, the elevation error is described by a normal distribution with zero mean and a standard deviation σ_e.

Other factors that contribute to observation uncertainties are false negative and false positive errors. False positive error is the probability of perceiving the incoming animal when it is out of range; false negative error is the probability of not perceiving the incoming animal when it is in range. Our sensor model can be described by the parameters in Table 1.

Table 1: Sensor Parameters

	Parameter
Range limit	D_R
Azimuth limit	θ_a
Elevation limit	θ_e
Bearing error standard deviation	σ_b
Elevation error standard deviation	σ_e
False positive probability	p_{fp}
False negative probability	p_{fn}

3.3 Reward Model

We assume that the outgoing flying animal is a rational agent that minimizes its risk of mid-air collision with the incoming flying animal, while avoiding collision with static objects in the environment. Furthermore, we assume the flying animal tries to use as few manoeuvres as possible to avoid collision. To model such behaviour in our POMDP agent, we use the following additive reward function $R(s, act) = R_C(s)+R_W(s)+R_M(s, act)$, where $R_C(s)$ is the penalty imposed if at state $s \in \mathcal{S}$, the outgoing and incoming animals collide, $R_W(s)$ is the penalty imposed if at state $s \in \mathcal{S}$, the outgoing animal collides with one or more static objects in the environment, and $R_M(s, act)$ is the cost for the outgoing animal to perform action $act \in \mathcal{A}$ from state $s \in \mathcal{S}$.

4 The Simulator

Given the POMDP problem as modelled in Section 3, the motion strategy for the outgoing animal is generated by solving the POMDP problem. Any POMDP solver can be used. In this work, we use Monte Carlo Value Iteration (MCVI) [3], which has been shown to perform well on POMDP-based TCAS [2], the model we have adopted for modelling motion strategies of flying animals in avoiding mid-air collision. Our simulator simulates the behaviour of the flying animal that uses this motion strategy to avoid mid-air collision in various head-on encounter scenarios. The head-on encounter scenarios can be generated based on data or information about flight plans of the flying animal under study.

Biologists can then use the simulator to generate and observe how the agent avoids mid-air collision in various environments and encounter scenarios, to obtain an intuition on how the flying animals might avoid mid-air collisions.

One may argue that our simulator assumes that the flying animal acts rationally, in the sense that it tries to maximize a certain objective function, while the real animal may not act rationally. Indeed, this is true. However, our preliminary tests on honeybees data indicate that the underlying motion strategy of honeybees in avoiding mid-air collision may not be far from that of a rational agent (Section 5).

One may also argue that the objective function we set may not be the same as the objective function of the flying animal. Again, this is correct. However, if we acquire additional information that leads us to believe that the reward function needs to be modified, we can easily do so by revising the reward function in the model, regenerating the motion strategy, and revising the simulator to implement the new motion strategy.

5 Case Study on Honeybees

This case study is based on 100 honeybee encounters in a 3-dimensional tunnel space. The size of the tunnel space is $930mm \times 120mm \times 100mm$. The possible coordinate values for x, y, z are $-30 \leq x \leq 900$, $-60 \leq y \leq 60$, and $-50 \leq z \leq 50$. Each encounter consists of the trajectories of two bees, in the format of $(x_1, y_1, z_1, x_2, y_2, z_2)$ at each timestamp, where (x_1, y_1, z_1) is the position of the outgoing bee and (x_2, y_2, z_2) is the position of the incoming bee. The data is sampled at 25 frames per second. Figure 5 shows one example of an encounter scenario. In it, the path formed by circles is the outgoing bee's flying path, while the path formed by crosses is the incoming bee's flying path.

5.1 Setting the Parameters

In our *in silico* system, the outgoing bee is modelled as a POMDP agent as described in Section 3.

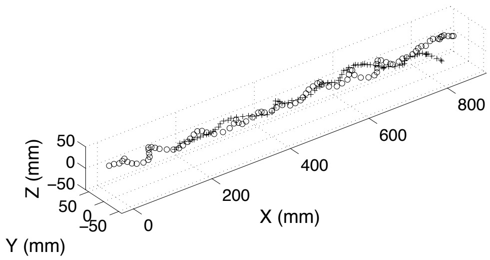

Figure 5: The 3D tunnel space in which the path formed by circles represents the outgoing bee's flying trajectory and the path formed by crosses represents the incoming bee's flying trajectory.

In the POMDP model, we assume the outgoing bee and the incoming bee share the same flying dynamics, i.e., they have the same horizontal velocity u, the same maximum/minimum vertical acceleration $\pm a_m$ and the same maximum/minimum turn rate $\pm\omega_m$. This is a reasonable assumption considering that both bees are of the same species, i.e., honeybees, and both behave in the same environment, i.e., the tunnel space. To get the exact values for these parameters, we perform statistical analysis on the data set. From this analysis, we can set $u = 300mm/s$, $a_m = 562.5mm/s^2$, and $w_m = 375deg/s$.

Now, we set the sensing parameters (Table 1). Since bees can see quite 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 bees 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.

We use the reward model as described in Section 3.3 to model the risk of near mid-air collisions and the risk of colliding with the tunnel boundaries.

We consider a state $s = (x_1, y_1, z_1, \theta_1, u_1, v_1, x_2, y_2, z_2, \theta_2, u_2, v_2) \in \mathcal{S}$ to be a collision state whenever the centre-to-centre distance between two parallel body axes is smaller than the wing span of the bee. By analysing the data and based on the biologist's observations on when collision occurs, we set this centre-to-centre distance (or wing span) to be 12mm. And define a state to be in collision when the two bees 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 $\|x_1 - x_2\| \leq 5$. We assign collision penalty to be -10,000 as suggested in [2], i.e., $R_C(s) = -10,000$ whenever s is a collision state. In addition, to discourage unnecessary manoeuvres, we also assign a small penalty of -0.1 as suggested in [2], i.e., $R_M(s, act) = -0.1$ when act has a non-zero vertical speed or non-zero turn rate.

Bees have a tendency to fly in the centre of the tunnel. To mimic this flying tendency, we impose a penalty $R_W(s)$ when the bee is too close to the tunnel walls. Specifically, in the Y-axis, when our agent bee flies in the centre area of the tunnel ($-20 \leq Y \leq 20$), no penalty applies; beyond that, a penalty applies linearly proportional to the distance to the wall; when our agent hits walls, a maximum penalty $-10,000$ is imposed. Similarly, the gradient-based penalty mechanism is also applied in the z-axis.

5.2 Experimental Setup

The goal of this experiment is to measure the resemblance of the trajectories produced by the original outgoing bee and trajectories produced by POMDP for the outgoing bee. For this comparison, we use two measurements, derived from the necessary conditions for the two trajectories to be the equivalent. The first measurement is *Collision rate*, which is the percentage of colliding encounters (among the 100 encounters). The second measurement is *Minimum Encounter Distance (MED)*, which is the smallest Euclidean distance between the outgoing bee and the incoming bee during one whole encounter process. If the trajectories produced by POMDP are similar to the trajectories of the original outgoing bee, then both collision rate and MED should be similar too.

To generate the trajectories, we first need to solve the POMDP problem. For this purpose, we implement our POMDP problem in C++ and solve it using MCVI [3]. Since MCVI is a randomized algorithm, we generate 30 different policies to get reliable measurements. To reliably capture the effect of stochastic uncertainty on the collision avoidance strategy, for each policy, we run 100 simulations. Each simulation consists of 100 different encounter processes, where in each encounter, the simulated incoming bee follows one of the trajectories observed from a real bee. Each simulation run produces a collision rate. The average collision rate of the trajectories generated by the *in silico* system is then the average collision rate over the 30×100 simulation runs. The average MED for a particular encounter situation is then the average MED over 30×100 simulation runs too.

All experiments are carried out on a Linux platform with a 3.6GHz Intel Xeon E5-1620 and 16GB RAM.

5.3 Experimental Results

Table 2: Collision Rates

Policy	Collision rate	Margin of error (95% Conf.)
Bee	3%	—
POMDP	3.84%	±0.13%

Applying the collision definition of our POMDP agent to the 100 bee data, we found the collision rate of this set of data is 3%. Table 2 shows the collision rates of the original data and the average collision rate of the POMDP-based *in silico* system. The two collision rates are less than 1% difference, which indicates that the trajectories produced by the POMDP-based *in silico* system are similar to trajectories produced by the actual bees in terms of collision rate.

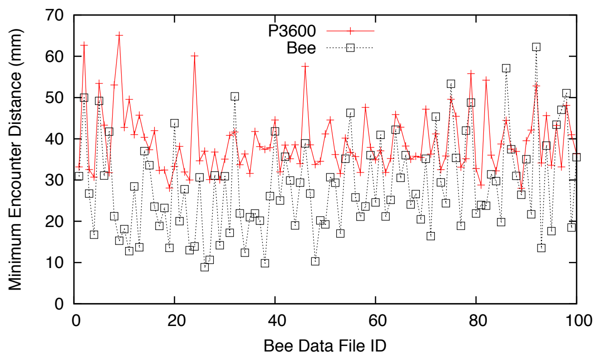

Figure 6: *Minimum Encounter Distance (MED)* for the bee data and the simulated encounter in our POMDP-based *in silico* system.

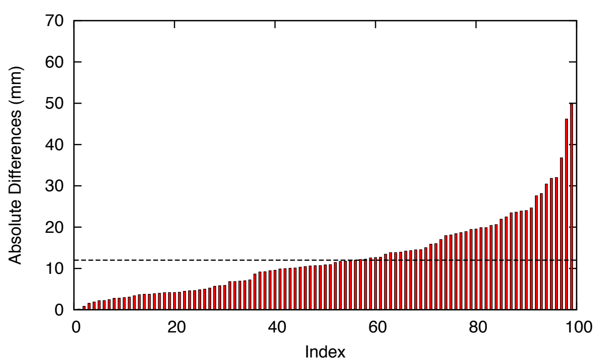

Figure 7: The sorted absolute difference between the *MED* measure of real bee encounters and the average *MED* of the simulated encounters generated by the *in silico* system

Figure 6 shows the *Minimum Encounter Distance (MED)* for the real bee data and the simulated encounter in our POMDP-based *in silico* system. Figure 7 shows a histogram depicting the increasing sorted absolute differences between the *MED* measurement of the real bee data and the average *MED* measurement of the simulated encounters in our POMDP-based *in silico* system. This histogram shows that in 98% of the encounters, the absolute difference between the *MED* of the real bee data and that of the simulated encounters is less than 35mm, while in 60% of the encounters, the absolute difference between the *MED*s is less than 12mm. In fact, the average absolute difference between the *MED* measure of the real bee data and the average *MED* measure of the POMDP-based *in silico* system is 12.60, with 95% confidence interval of 1.85.This 12mm average absolute difference is roughly equivalent to the estimated average wing span of the honeybees in our data, which implies that in terms of *MED* measure, our POMDP-based *in silico* system produces similar results to the original bee data.

6 Conclusions

In this paper, we propose a POMDP-based *in silico* system to help biologists study mid-air collision avoidance strategies of flying animals. Our system is distinct from flying animal dynamics and trajectory simulation, as the focus is on the strategy behind the observed motions, rather than the specific motions in space. Our *in silico* system consists of a model and a simulator. We model the animals as decision making agents under the POMDP framework. The solution to this POMDP problem is an optimal motion strategy for the agent to avoid mid-air collision with another flying animal. Our simulator simulates the behaviour of a flying animal that uses this motion strategy in various head-on encounter scenarios. The head-on encounter scenarios are generated based on data and information about flight plans of the flying animal under study.

We tested our system on 100 honeybee encounters. We measure how close our *in silico* system to the actual bee using two measurements —collision rate and minimum encounter distance— that are derived from the necessary conditions for the two systems to be equivalent. Preliminary results indicate that our POMDP-based *in silico* system is a promising tool to study mid-air collision avoidance strategies of flying animals, *in silico*. Such a tool may help biologists better understand mid air collision-avoidance strategies of flying animals faster and with much less cost, which in turn may benefit the robotics community in developing better mid air collision-avoidance system.

Many avenues are possible for future work. First is the measurement to determine if the *in silico* system generates similar trajectories as the real data. In this work, we have used measurements derived from the necessary conditions. A better measurement should be derived from the sufficient and necessary condition. Second is to test the system on more data and various different scenarios. Third is to expand the system to handle more complex encounter scenarios.

References

[1] R. M. Alexander. *Animal Mechanics.* University of Washington Press, Seattle, 1969.

[2] H. Bai, D. Hsu, M. J. Kochenderfer, and W. S. Lee. Unmanned aircraft collision avoidance using continuous-state pomdps. *Robotics: Science and Systems VII*, page 1, 2012.

[3] H. Bai, D. Hsu, W. S. Lee, and V. A. Ngo. Monte carlo value iteration for continuous-state pomdps. In *Algorithmic Foundations of Robotics IX*, pages 175–191. Springer, 2011.

[4] J. P. Chryssanthacopoulos and M. J. Kochenderfer. Accounting for state uncertainty in collision avoidance. *Journal of Guidance, Control, and Dynamics*, 34(4):951–960, 2011.

[5] J. Groening, L. McLeod, N. Liebsch, I. Schiffner, and M. V. Srinivasan. When left is right and right is wrong: Collision avoidance in honeybees. *Frontiers in Behavioral Neuroscience*, (235), 2012.

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

[7] K. Hsiao, L. Kaelbling, and T. Lozano-Perez. Grasping POMDPs. In *ICRA*, pages 4685–4692, 2007.

[8] M. J. Kochenderfer and J. P. Chryssanthacopoulos. Robust airborne collision avoidance through dynamic programming. *Massachusetts Institute of Technology, Lincoln Laboratory, Project Report ATC-371*, 2011.

[9] H. Kurniawati, T. Bandyopadhyay, and N. Patrikalakis. Global motion planning under uncertain motion, sensing, and environment map. *Autonomous Robots: Special issue on selected papers from RSS 2011*, 30(3), 2012.

[10] H. Kurniawati, Y. Du, D. Hsu, and W. Lee. Motion planning under uncertainty for robotic tasks with long time horizons. *IJRR*, 30(3):308–323, 2011.

[11] H. Kurniawati, D. Hsu, and W. S. Lee. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Proc. Robotics: Science and Systems*, 2008.

[12] K. Y. Ma, P. Chirarattananon, S. B. Fuller, and R. J. Wood. Controlled flight of a biologically inspired, insect-scale robot. *Science*, 340(6312):603–607, 2013.

[13] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001.

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

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

[16] M. Raibert, K. Blankespoor, G. Nelson, R. Playter, and the BigDog Team. Bigdog, the rough-terrain quadruped robot. In *Proc. of the 17th IFAC World Congress*, 2008.

[17] I. Ros, L. Bassman, M. Badger, A. Pierson, and A. Biewener. Pigeons steer like helicopters and generate down-and upstroke lift during low speed turns. *Proceedings of the National Academy of Sciences*, 108(50):19990–19995, 2011.

[18] K. A. Smith, M. Kochenderfer, W. A. Olson, and A. E. Vela. *Collision Avoidance System Optimization for Closely Spaced Parallel Operations through Surrogate Modeling.* American Institute of Aeronautics and Astronautics, 2013.

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

[20] S. Temizer, M. J. Kochenderfer, L. P. Kaelbling, T. Lozano-Pérez, and J. K. Kuchar. Collision avoidance for unmanned aircraft using markov decision processes. In *Proc. AIAA Guidance, Navigation, and Control Conference*, 2010.

[21] M. S. Triantafyllou and G. S. Triantafyllou. An efficient swimming machine. *Scientific American*, 272(3):64–70, 1995.