Obstacle Avoidance and Boundary Following Behavior of the Echolocating Bat

Laura Freyman and Scott Livingston
Faculty Mentors: Dr. Timothy K. Horiuchi, Dr. P. S. Krishnaprasad and Dr. Cynthia F. Moss

***Abstract*—Developing a robust and rapid system for obstacle avoidance and boundary following has long been a goal of robotics research. The echolocating bat, *Eptesicus fuscus*, has evolved to navigate through complex environments in pursuit of evasive prey. Given its robust flight behavior, active sensing through sonar, small size, and low power consumption, *E. fuscus* has much potential as a source of inspiration for bio-inspired engineering, especially in the context of unmanned aerial vehicles. In this paper, we take steps toward such engineering by investigating the obstacle avoidance and boundary interaction behavior of *E. fuscus*. The bat's obstacle avoidance behavior is compared to the neurally-inspired `open space` algorithm, a winner-take-all based heading direction selection method. Boundary interaction by the bat is examined by testing a hypothesis: in following a boundary, the bat's flight path curvature is a time-delayed version of the boundary curvature. Initial results are promising; however, future work will include a more rigorous statistical analysis of observed behavior.**

I. Introduction

In designing a mobile robot or other freely moving, autonomous vehicle, one of the primary concerns is successfully moving toward a target while avoiding collisions with possibly nonstationary obstacles. Another important task for mobile robots is continuous following of a boundary. Though this task may at first seem derivative of obstacle avoidance, it can be used to serve unique purposes that are greater than simply avoiding collisions, e.g., localization or map building. Robot navigation methods can be understood by imagining a spectrum spanning from entirely global methods that attempt to find the globally optimal path from the robot's current position to the target (for a globally convergent potential field method, see [1]), to entirely local methods that only use immediate sensory information to select appropriate actions and thus include a very limited internal world model (for a highly successful example, see the Dynamic Window Approach [2]; for a more theoretical treatment, see [3]). On the one hand, globally optimal path planning methods suffer from computational intractability or overly stringent assumptions, such as availability of a perfect world model and no sensor noise. On the other hand, local methods suffer from local minima in the search space, and may exhibit cyclic behavior or even fail to reach the target. These challenges are further complicated in autonomous unmanned aerial vehicles (UAVs), which can only use a very limited range of speeds in order to maintain lift.

A potentially rich source of inspiration for robot navigation methods is the flight behavior of the echolocating big brown bat, *Eptesicus fuscus*. The big brown bat uses sonar for sensing its environment [4] and has evolved sophisticated behavior for pursuing evasive prey [5]. During foraging in the night, the bat must navigate through complex wooded environments while competing with other bats for food. As a subject of study, the echolocating bat is very attractive because it uses stroboscopic, active sensing and thus, with appropriate equipment, one can infer approximately what the bat is sensing (based on vocalization recordings and trajectory data) and how it is reacting at specific instants of time. In particular, it is a superb organism for neuroethological, systems, and control theoretic research.

A recently developed and promising approach to analysis of navigation by biological systems is behavioral dynamics (see [6] for a review). In this approach, behavior is described by a dynamical system in which the specifics of goal-directed behavior emerge from agent-environment interactions, characterized by attractors and repellers, and thus are not explicitly planned by the agent. Notable features of behavioral dynamics include: avoidance of the assumption of an internal world model, computational simplicity and, depending on the descriptive system, improved scaling with increased environment complexity. For a concise treatment of the subject and an application to human walking behavior modeled using the dynamics of a spring with damping, see [6].

Recent work demonstrates the successful development of a sensorimotor feedback law, generating a type of pursuit called "constant absolute target direction" ([5], [7] and [8]), that describes prey pursuit behavior of the echolocating bat and has applications to, for example, missile guidance and UAV flight control. Encouraged by this work, we propose that an analogous approach can be taken to formulate sensorimotor feedback laws to describe the obstacle avoidance behavior and the boundary interaction behavior of the echolocating bat, and we further hope that biological models will lead to the improvement of existing approaches or even development of entirely new approaches to these two key functions in mobile robots. In addition, by finding a feedback law that accurately describes the behavior of the echolocating bat, control methods for UAVs, which have unique requirements to maintain lift, might be more easily engineered.

In the present paper, we take steps toward developing sensorimotor feedback laws for the obstacle avoidance and boundary following behaviors of the echolocating bat by conducting experiments with the big brown bat, *Eptesicus fuscus* and then analyzing experimental data. In the case of obstacle avoidance, we examine observed behavior through comparison to an

Research supported by NSF REU Grant CCF-0755224.

existing steering control method called the `open space` algorithm [9]. In the case of boundary following, we test the hypothesis that the bat's flight path curvature is a time-delayed version of the boundary curvature. Initial results are considered; however, a more comprehensive statistical analysis of our experimental data is needed before any conclusions can be achieved.

II. Background

In this section, we detail the "flight room" in which all experiments were conducted and describe equipment used for data recording. We then briefly review natural Frenet frames, used to model the bat's movement and flight controls. Finally, the `open space` algorithm is explained.

A. Flight Room

All experiments with the big brown bat, *Eptesicus fuscus*, were conducted in one of the flight rooms of the Auditory Neuroethology Lab, which is led by Dr. Cynthia F. Moss and located at the University of Maryland, College Park. The flight room has dimensions of 7.3 m length, 6.4 m width, and 2.5 m height. The walls and ceiling are covered with sound absorbent material that substantially reduces reverberations. Lighting during experiments is dim and long wavelength (red, $>$ 650 nm). *E. fuscus* is only weakly sensitive to this type of light and is thus forced to rely primarily on echolocation for navigation and other behaviors.

A diagram of the flight room is shown in Fig. 1. We used three pieces of recording equipment in our experiments. First, two high speed cameras with frame rate 250 fps (one frame per 4 ms), and sensitivity to infrared light enable video recording. Second, an ultrasonic microphone is placed on the floor during experiments for detailed recording of the bats' vocalizations. Third, a horizontal array of microphones allows for measuring peak intensities of vocalizations with respect to direction. For all baseline (flight in empty room) and obstacle avoidance experiments, the array consists of 16 microphones evenly spaced 1 meter apart along three walls. Each of these microphones feeds into a bandpass filter and peak envelope detector before the signal is recorded. During the course of our project, Ben Falk and Lasse Jakobsen of the Auditory Neuroethology Lab expanded the array to the fourth wall, resulting in 20 microphones total. Recording of vocalization intensity in any direction is now supported. All boundary following experiments were performed with the complete array, consisting of 20 microphones.

B. Natural Frenet Frames

In our analysis of experimental data, we model the bat as a particle and consider its movement using the natural Frenet frame (also called Relatively Parallel Adapted Frame (RPAF)) [11], [8], [12] and [13]. In short, natural frames specify a right-handed coordinate system that moves with the bat as it flies through space, allowing us to examine the speed and curvatures taken by the bat at every time instant. The curvatures of the flight path, as found in the natural Frenet frame, represent steering controls available to the bat. Understanding how speed selection and steering in the observed behavior relate to sensory information, such as range and directional data obtained from an ultrasonic vocalization, is critical in finding sensorimotor feedback laws that describe obstacle avoidance and boundary following behavior and in the eventual application of the control methods to mobile robots. A more detailed treatment of the natural Frenet frame follows.

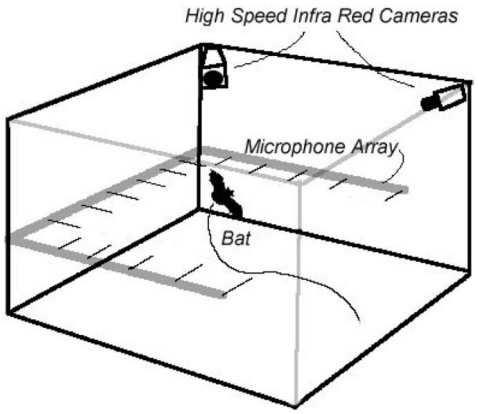

Fig. 1. The experimental flight room. Dimensions are $7.3 \times 6.4 \times 2.5\ m^3$. The microphone array is used for directional analysis of the bat's vocalizations. The two high speed cameras facilitate 3D flight path reconstruction. Finally, a high fidelity microphone is placed on the floor for recording vocalizations. The figure is copied from [10].

We represent the bat's flight path as a vector γ parameterized by time t; that is, $\gamma : [0, T] \rightarrow \Re^3$, where T is the last time considered. We assume that $\forall t$, $\dot{\gamma}(t) \neq 0$; such a curve is called *regular*. Note that the "dot" operator over the vector indicates the derivative with respect to time, that is, $\dot{\gamma} = \frac{d\gamma}{dt}$. In addition, we assume that the curve γ is twice differentiable and continuous for all $0 \leq t \leq T$. At time t, the arc length of the curve is expressed by

$$s(t) = \int_0^\tau \|\dot{\gamma}(\tau)\| d\tau, \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean norm. Given the arc length s, we thus have the speed $\nu = \frac{ds}{dt}$.

We next proceed to specify the moving coordinate system that is the "frame." Let $\mathbf{T}$ be a unit vector in the direction of the instantaneous velocity vector:

$$\mathbf{T}(t) = \frac{\dot{\gamma}(t)}{\|\dot{\gamma}(t)\|} = \frac{1}{\nu}\dot{\gamma}(t). \tag{2}$$

The natural Frenet frame is thus defined for all t,

$$\dot{\mathbf{T}}(t) = \nu(t)(u(t)\mathbf{M_1}(t) + v(t)\mathbf{M_2}(t)), \tag{3}$$
$$\dot{\mathbf{M}}_\mathbf{1}(t) = -\nu(t)u(t)\mathbf{T}(t), \tag{4}$$
$$\dot{\mathbf{M}}_\mathbf{2}(t) = -\nu(t)v(t)\mathbf{T}(t), \tag{5}$$

where $\mathbf{M_1}$ and $\mathbf{M_2}$ are unit vectors such that

$$\mathbf{M_1}(t) \perp \mathbf{M_2}(t),$$
$$\mathbf{M_1}(t) \perp \mathbf{T}(t),$$
$$\mathbf{M_2}(t) \perp \mathbf{T}(t) \quad \forall t,$$

and u and v are called natural curvatures. These curvatures can be understood as gyroscopic steering controls available to the moving particle (in our case, a bat). Therefore, given an initial frame $(\mathbf{T}(0), \mathbf{M_1}(0), \mathbf{M_2}(0))$ and a path curve γ, all positions of the frame along the path (for $t \in (0, T]$), speed ν and curvatures u and v are uniquely determined.

In this project, we use the natural Frenet frame in order to reconstruct the bat's speed and curvature (or steering) at every time instant. Given the frame rate of the high speed infrared cameras used for recording (250 fps) time in experimental data is given in discrete steps of 4 milliseconds. Determination of the initial frame orientation and all curvatures and speed throughout the course of a flight is a nontrivial problem. We use the Regularized Inversion algorithm developed by Puduru Viswanadha Reddy to approximate an optimal solution [14]. The MATLAB software that implements the algorithm was developed by P. V. Reddy in the Intelligent Servosystems Lab, which is led by Dr. P. S. Krishnaprasad.

C. `Open Space` Algorithm

The `open space` algorithm is a method for obstacle avoidance proposed by Dr. Timothy K. Horiuchi [9]. It is briefly introduced here.

Assume that the only sensory information available for steering is range and directional data, which can be understood by imagining a set of relative heading angle (in radians) and distance (in meters) pairs. The angle is relative to the current sensor orientation or, in the case of a bat, the head direction. The `open space` algorithm operates under the assumption of constant speed (an important requirement to maintain lift in many aerial vehicles) and thus focuses on selecting a new bearing. Each potential direction θ is evaluated according to the equation

$$E(\theta) = E_0 + g \cdot e^{\frac{-(\theta-\theta_g)^2}{\sigma_g^2}} - \sum_{i=1}^{N} \frac{1}{r_i} \cdot e^{\frac{-(\theta-\theta_i)^2}{\sigma(r_i)^2}} \qquad (6)$$

where E_0 is an initial bias, which may encode prior knowledge such as actuation limits, g is the center height of a goal Gaussian, which if no obstacles were detected, would cause flight toward the goal, and the final term is a summation of subtractive Guassians caused by detected distances to obstacles (possibly revealing the presence and proximity of obstacles). See Fig. 2 for a graphical depiction of a sample evaluation. Two noteworthy elements are the inverse obstacle weighting with respect to range (i.e., $\frac{1}{r_i}$), and the range-dependent variance $\sigma(r_i)$. This system has been successfully implemented in analog VLSI using spiking neurons that "race" to win and are inhibited by echoes from obstacles (which contain implicit obstacle distance data given echo return times), thus making a winner-take-all architecture [9].

III. Obstacle Avoidance

The first behavior of the big brown bat of interest is obstacle avoidance. In this project, we began our work in the Auditory Neuroethology Lab, in which we designed and conducted experiments in order to gather behavioral data on the echolocating bat, and then moved on to data analysis, including: 3D flight path determination, vocalization directional analysis, calculation of various flight trajectory statistics, and curvature extraction. Most of our analysis was performed in the Intelligent Servosystems Lab. As mentioned earlier in the paper, both laboratories are located at the University of Maryland, College Park, and are led by Dr. C. F. Moss and Dr. P. S. Krishnaprasad, respectively.

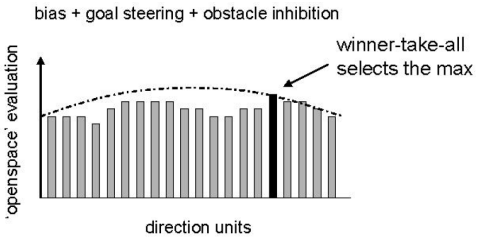

Fig. 2. A sample `open space` evaluation. The horizontal axis represents potential steering directions for the agent, and the vertical axis represents the desirability of each possible direction. The dotted curve indicates an initial bias due to, for example, direction to the goal and any prior knowledge, such as actuation limits. The two dips in the plot represent reduced desirability caused by detected obstacles. Finally, the direction with the largest evaluation is selected. The image is copied from [9].

In attempting to understand the obstacle avoidance behavior of the big brown bat, we compare experimental results to simulations of a bat operating under the `open space` algorithm in an environment that approximates the experimental setup.

A. Experiment: Materials & Methods

All obstacle avoidance experiments were conducted in a flight room in the Auditory Neuroethology Lab (simply the "BatLab"). Two bats of the species *Eptesicus fuscus* were used in the experiments, one male and the other female. In all experiments, the basic task for the bat is to find and capture a tethered meal worm (i.e., a stationary target), which is hung at the end of a thin string attached to the ceiling. We performed 36 baseline (empty flight room, no obstacles) trials before proceeding to obstacle avoidance experiments.

In our experiments, the obstacles were artificial trees constructed by previous BatLab researchers. The basic structure of an artificial tree consists of mist net strung on the edges of two metal rings, which form the top and bottom of the tree. When hung from the ceiling, each of these artificial trees approximates the shape of a cylinder with radius 25.7 cm and spans nearly the entire height of the flight room. Mist net is a thin and light weight black net that permits visibility of the bat by the flight room cameras, even if the bat is flying behind several artificial trees. However, despite its thin structure, the echolocating bat can still detect the mist net, and accordingly, the trees successfully act as obstacles.

We designed and conducted a total of five different obstacle avoidance experiments, in which the number of artificial trees is 10, 13, or 14. Table I specifies dates, numbers of trees, and number of trials completed. A top view of the flight room for an experiment in which 10 trees were used is shown in Fig. 3, and similarly, Fig. 4 depicts the setup for an experiment in which 14 trees were used. Notice the use of a barrier at

TABLE I
OBSTACLE AVOIDANCE EXPERIMENTS: NUMBER OF TRIALS

Date	Number of trees	Bat HP77	Bat OR41
06/13/2008	10	20	11
06/18/2008	10	0	9
06/18/2008	13	20	0
06/23/2008	14 + barrier	22	4
06/24/2008	14 + barrier	0	7

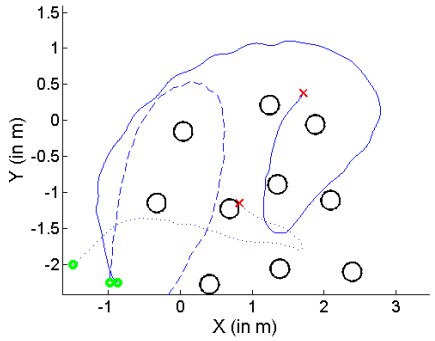

Fig. 3. Top view of the arrangement of artificial tree obstacles for one of our experiments, conducted on June 13, 2008. The obstacles are approximately cylindrical, and thus appear as circles in this top view. Each artificial tree has a radius of 25.7 cm. Three sample recorded flight paths are drawn. For each path, a small circle marker indicates the starting position, and a small "X" marker indicates the final position.

the edge of the artificial "forest." The barrier is a sheet of mist net that connects a tree at the edge of the forest to one wall of the flight room. This construction prevents the bat from continuously circling around the outside of the forest and thus forces it to enter and navigate amidst the artificial trees in search of the meal worm. During performance of all obstacle avoidance experiments, we initially hung the meal worm somewhere among the artificial trees and then moved the meal worm to a new, random location after every two or three trials. The location of the meal worm is changed frequently to discourage development of spatial memory by the bat.

B. Data Analysis

After completion of all experiments, the next step is analysis of the data. In particular, the software tools D3, Sunshine, and Moonbeam, all developed in the Auditory Neuroethology Lab by Kaushik Ghose, Murat Aytekin, and others, enable 3-dimensional flight path reconstruction (using raw video data), directional analysis of vocalizations (using microphone array data), and trial animation generation, respectively. This analysis is important because it takes raw experimental data and transforms it into a much more accessible form: a series of (x, y, z) coordinates in time steps of 4 ms (as a result of camera rates of 250 fps), the (x, y) coordinates of the artificial trees, and normalized vocalization intensities read by the microphone array. The vocalization intensities are tagged with time of occurrence, and can thus be easily used to create an emission beam. This beam is important because it provides insight into what the bat might have been able to sense at that moment of flight. In particular, we can infer obstacle distances

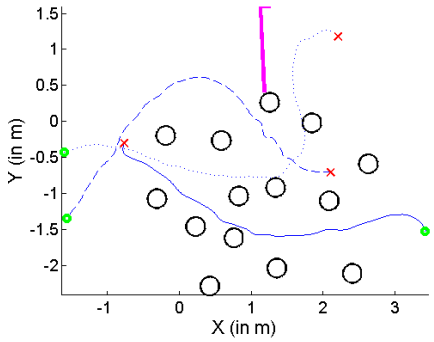

Fig. 4. Top view of the arrangement of artificial tree obstacles for one of our experiments, conducted on June 23, 2008. The obstacles are approximately cylindrical, and thus appear as circles in this top view. Each artificial tree has a radius of 25.7 cm. The thick line toward the top of the figure represents a mist net barrier constructed to prevent the bat from circling around the outside of the artificial forest. The barrier spans from the edge of one tree to the flight room wall (not depicted). Three sample recorded flight paths are drawn. For each path, a small circle marker indicates the starting position, and a small "X" marker indicates the final position.

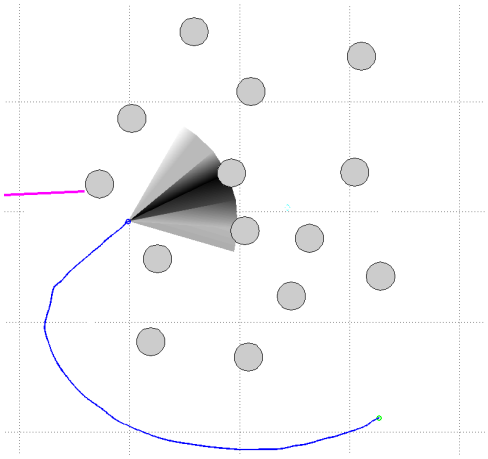

Fig. 5. A sample bat vocalization from one of the obstacle avoidance experiments (performed on June 23, 2008). Shaded circles represent obstacles, and the short, straight line attached to the left-most tree is a mist net barrier. The fan shape represents the echolocation chirp, with darker gradients indicating greater intensity. Notice that the bat initially flies outside of the artificial forest and, after detecting the barrier, turns into the forest. Based on the vocalization intensity at this moment in time, one may suppose that the bat sensed the distances of the trees immediately in front of it and also trees in the upper right section of the artificial forest.

known by the bat. A sample vocalization beam, including flight path, is shown in Fig. 5

After determination of the bat's flight path, which is represented as a series of (x, y, z) coordinates in time steps of 4 ms, an optimal natural Frenet frame is found for the data. Curvature extraction is completed using the Regularized Inversion software developed by P. V. Reddy [14], leading to speed, curvatures, and frame orientation for every point along the flight path in time steps of 2 ms. The extracted speed and natural curvatures from a particular obstacle avoidance trial are shown in Fig. 6.

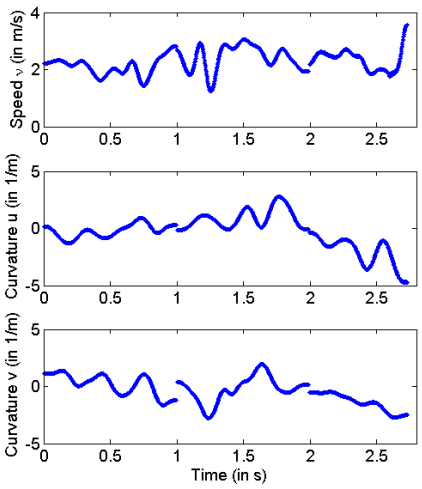

Fig. 6. Sample speed and curvature plots for one of our obstacle avoidance trials (specifically, June 23, 2008, trial 8). Sudden steps in the curves are a result of dividing the trial into three disjoint sections prior to curvature extraction. This must be done because analysis of an entire trial spanning over several seconds (at a resolution of 2 ms) is very computationally expensive.

C. Comparison to `Open Space` Simulations

In order to begin to understand the obstacle avoidance behavior of the echolocating bat, we compare the actual bat flight data to that generated by the `open space` algorithm in simulation. Due to time limitations, all of the analysis herein pertains only to a limited number of trials from our complete data set.

First, as noted in the Background section of this paper, there are several parameters in the `open space` algorithm that can be altered to change the model behavior. We modified the existing simulation software, written by Dr. T. K. Horiuchi, to make the simulation more like the experimental environment. To that end, we wrote a MATLAB function called `simecho` which, given an obstacle field of circles of arbitrary radius and a resolution for range scanning, returns approximate distance data. Thus, instead of treating each obstacle as a point in simulation, we are able to deal directly with distance information of non-point obstacles (e.g., artificial trees). Various other updates were made to existing software in order to more closely model the experimental setup. For example, time steps were reduced from 100 ms to 4 ms in order to reflect the real bat flight data.

During initial use of the software and while attempting to adjust controller parameters to enable successful operation in a simulation environment nearly equivalent to the experimental setup of our obstacle avoidance trials, we discovered the difficulty of selecting a good combination of parameters. A stochastic search method is proposed, in which the parameter space (4 dimensional because we considered only 4 parameters) is searched by adding Gaussian noise to competing sets of parameters. We label the parameters a, b, c, and d (their use will be explained later). Then, given a total number of iterations `num_iter` and number of offspring to consider in each iteration `num_offspring`, the stochastic search method can be expressed in the following pseudocode:

```
Initialize a set of combinations of
 initial positions, initial orientations,
 and goal positions.
Initialize the parameters {a,b,c,d}
For i in 1 to num_iter
   Select an initial position and goal
    location.
   For j in 1 to num_offspring
      Add Gaussian noise (mean 1, var 0)
       to parameters.
      Perform trial using these parameters
      Score this offspring's performance
   Select two offspring with highest score
   Randomly combine parameters from these
    two offspring to generate a new
    parameter set.
```

The cost function computed to score the offspring (i.e., candidate parameter sets) is, given parameters a, b, c, and d,

$$w(a,b,c,d) = n_{coll} + 0.1 \cdot (T + d_g) \tag{7}$$

where n_{coll} is the number of collisions (the agent is permitted to pass through artificial trees and continue the trial), T is total time of the trial (where trials stop if either the goal is reached or the maximum allowed trial length is reached), and d_g is the final distance to the goal (i.e., distance between agent and goal point at end of trial).

We describe where the parameters appear in the software implementation of the `open space` algorithm. First, in biasing of direction desirability based on prior goal knowledge, each direction θ is initialized using

$$E(\theta) = 5 + \frac{a}{\max(0.5, d_g - b)} e^{\frac{-(\theta + g_{bias})^2}{49}}, \tag{8}$$

where d_g is the current distance to the goal, g_{bias} is a goal-induced directional bias, and a and b are free parameters. Recall that the initial biasing is performed *prior* to acquisition of distance data. Let the obstacle distance data be represented by a function that maps angle values to distance values: $f : [-\frac{pi}{4}, \frac{pi}{4}] \rightarrow [0, \infty)$. Iterating through each direction, which is discretized to 33 angular steps in software, the update rule is composed of two steps. First, for each direction θ, a value d_{inv} that decays exponentially with distance is calculated,

$$d_{inv} = c \cdot e^{\frac{-f^2(\theta)}{d^2}}. \tag{9}$$

Second, given the result of Eq. (9), the evaluations of all directions θ_i are updated by

$$E(\theta_i) := E(\theta_i) - d_{inv} \cdot e^{\frac{-(\theta - \theta_i)^2}{(1.5 + d_{inv}/2)^2}}. \tag{10}$$

After all direction evaluations are updated given detected distance, the direction with maximum evaluation (or "desirability") is selected.

For use in the proposed parameter selection method, combinations of initial bat position and orientation and goal position are created and used for training. Fig. 7 provides an example of progression of performance during the learning process. We only compare here two real bat obstacle avoidance trials

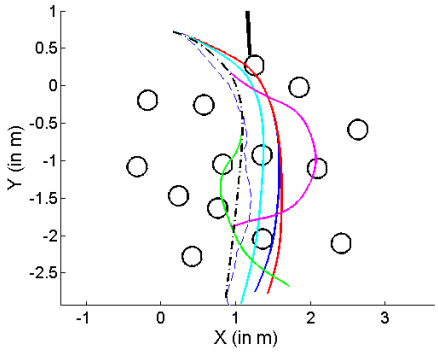

Fig. 7. Progressive example of performance of `open space` algorithm-generated flight paths. The real bat (measured experimentally) flight path is indicated by the thin dashed line. All thick lines indicate sample paths generated during the learning process. After 300 training iterations, the simulated bat's flight path is the thick dash-dotted line. Note that all simulated paths are initialized to the real bat's starting position and heading, and the goal is fixed at the terminal point of the real bat's flight path. The experimental setup is that of the obstacle avoidance experiment performed on June 23, 2008 (see Fig. 4 for details).

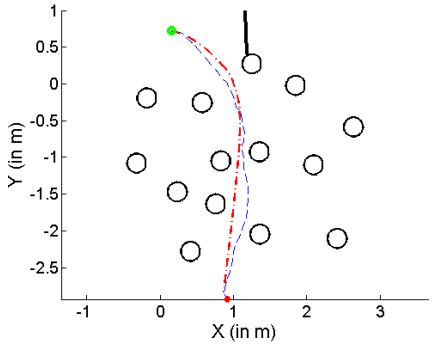

Fig. 8. This figure shows the actual flight path of an echolocating bat, indicated by the dashed line, in one of our obstacle avoidance experiments. Using parameters learned by our stochastic search method (same used in Fig. 9), also shown is the predicted path (indicated by thick dash-dotted line) generated by the `open space` algorithm with a fixed speed of 1.89 m/s. The bat's speed varied from 0.358 to 3.31 m/s, with a mean of 1.88 m/s. The start position is marked by a small circle near the top of the figure, and the goal is marked by a small dot at the bottom of the figure. For simulation, the goal is set to be the final position of the real bat.

to the `open space` predicted paths. After learning good parameters (by minimization of cost Eq. (7)) for the `open space` algorithm, simulated trials were generated by giving the agent the same initial position and heading as the real bat in the experimental trials under consideration and by fixing the goal at the final position of the real bat (i.e., last measured point in the bat's flight path). Ideally, the goal point considered would be the meal worm (recall the given task of the experiments), but we did not record the locations of the meal worm throughout our experiments.

Comparisons of actual to predicted flight paths are shown in Figs. 8 and 9. Parameters were learned by random selection of two sets of initial bat position and orientation and goal location performed over 300 iterations of our stochastic search method. In examining these initial results, it is important to note that the `open space` algorithm operates under constant speed and thus focuses solely on steering control. By contrast, the echolocating bat changes both its speed and steering during flight (see Fig. 6 for an example of this behavior).

D. Discussion & Conclusion

In summary, we conducted 99 experimental trials investigating obstacle avoidance behavior of the big brown bat, *Eptesicus fuscus*. With the goal of eventually describing this behavior using sensorimotor feedback laws, which may be later applied to improved autonomous UAV or mobile robot control, we performed an initial evaluation of the `open space` algorithm as a model for the observed behavior. Furthermore, we proposed an automated method for searching the parameter space of the algorithm. This stochastic search method finds the cost of potential parameter sets by considering the number of collisions, time to reach goal, and, if the goal is not reached, distance between agent and goal in simulated trials. The cost function definition probably influences the final parameter selection, and a thorough investigation of its effects, as well as consideration of machine learning techniques, is a potential direction for future research.

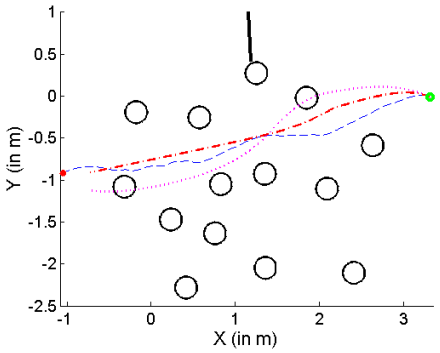

Fig. 9. This figure shows the actual flight path of an echolocating bat, indicated by the dashed line, in one of our obstacle avoidance experiments. Using parameters learned by our stochastic search method (same used in Fig. 8), also shown are two predicted paths generated by the `open space` algorithm under two fixed speeds: 0.9 m/s (thick dash-dotted line) and 1.89 m/s (thick dotted line). The bat's speed varied from 0.858 to 2.77 m/s, with a mean of 2.14 m/s. The start position is marked by a small circle near the right edge of the figure, and the goal is marked by a small dot at the left edge of the figure. For simulation, the goal is set to be the final position of the real bat. Notice that the predicted flight path with fixed speed of 1.89 m/s has two collisions, whereas that with speed of 0.9 m/s successfully navigates the artificial forest.

Comparisons to only two experimental trials are described in this paper (see Figs. 8 and 9). After reducing simulation speed by half in the second trial, the predicted paths are topologically equivalent to the actual bat flight paths. However, it is interesting to observe the predicted paths are smoother, that is, the simulated bat's path has fewer sudden turns. Therefore, these initial results are promising and speak favorably to use of the `open space` algorithm as a candidate model for obstacle avoidance behavior of the echolocating bat. Nonetheless, a comprehensive statistical study of the wealth of experimental data obtained is needed before any claims can be made. In future work, we will also attempt to find `open space` parameters that lead to a best-fit model of the experimental data, rather than optimizing them in terms of

speed and collision avoidance (irrespective of experimental data) as in the present study.

IV. Boundary Interaction

A. Hypothesis Statement

In this phase of the research, we hypothesize that while navigating around a curved boundary, the echolocating bat will utilize a time delayed version of the boundary curvature in controlling its own flight path. This hypothesis is inspired by a curvature-based control law for boundary following [15] and an experimental study in application to obstacle avoidance [16].

B. Experiment: Materials & Methods

To test this hypothesis, we performed 72 experimental boundary following trials among two bats. In these experiments, we used a mist net that was hung on the ceiling to create a curved boundary for the bat to follow. A meal worm was hung at different places along the boundary to encourage the bat to follow the curved net. After every four or five trials, the shape of the boundary was changed to prevent the bat from utilizing its spatial memory in navigating toward the worm and around the net. The boundary shape alternated between curved and straight configurations, and the flight trajectory and microphone array data were recorded for each trial. Pieces of white and reflective tape were attached to the net so that in analysis, these pieces of tape could be identified as points in a three dimensional space and the boundary could be reconstructed.

C. Data Analysis

1) Flight Path Determination: We have only analyzed 3 trials in which the bat appeared to follow the boundary. In these trials, the bats' flight trajectories were determined in a MATLAB program called D3, which is developed and maintained in the Auditory Neuroethology Lab. This program transforms the raw video data into a three dimensional reconstruction of the bat's flight path and the boundary location. In order to approximate the boundary shape, we performed Delaunay triangulation. Each point chosen on the boundary is represented by a coordinate in space, and this algorithm forms triangles by connecting each point so that no point is located within the circumcircle of any of these triangles. This process allows for a fairly accurate reconstruction of the complex boundary surface within our experiments, as shown in Fig. 10.

2) Vocalization Directional Analysis: The vocalizations recorded by the microphone array were analyzed using a MATLAB program called Sunshine (developed by Kaushik Ghose). The microphone array recorded the bat's vocalizations, which were analyzed using Sunshine, based on intensity and the time at which each was recorded, taking into account speed of sound and humidity within the room. A fan plot was generated from this analysis, which depicted the area of highest vocalization intensity and approximated the direction in which the bat's head was pointing when it vocalized.

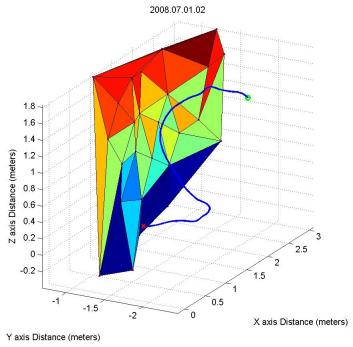

Fig. 10. The Delaunay triangulation reconstruction of the curved boundary plotted with the bat's flight path for trial 2 conducted on July 1, 2008. The circle represents the bat's starting position and the cross represents the bat's ending position.

3) Boundary Reconstruction: To best represent the net boundary on a two dimensional plane, points on the boundary within a limited range corresponding to the height of the bat's trajectory were selected and projected onto the xy-plane. Within the 3 trials that have been analyzed, we used a range of $1 < z < 1.4$ meters. This range resulted in a set of either 5 or 6 data points that most closely represented the curvature of the net boundary at the bat's height, given available boundary data resolution. The MATLAB function `linspace` was used in order to generate a vector of 100 points which were linearly spaced between the first and last data points on the boundary. A best-fit sixth order polynomial was then used to reconstruct the boundary and generate a more complete set of data points to be used for comparison to the bat's flight curvature. The sixth order polynomial was used because of its qualitative similarities to the original curved boundary, especially regarding its S-shaped appearance. The equation [17]:

$$|\kappa| = \left| \frac{y''}{(1+y'^2)^{3/2}} \right| \tag{11}$$

was used to extract absolute value of the curvature from the boundary approximation, in which y' and y'' are the first and second derivatives, respectively, of the best-fit polynomial reconstruction of the boundary. The three polynomials generated are

$$\begin{aligned} p_{trial2}(t) &= 0.0775t^6 - 0.5941t^5 + 1.498t^4 - 1.2581t^3 \\ &\quad -0.6699 \\ p_{trial13}(t) &= 0.0772t^6 - 0.4555t^5 + 0.902t^4 - 0.7558t^3 \\ &\quad +1.3094t - 1.6218 \\ p_{trial14}(t) &= -0.0616t^6 + 0.4228t^5 - 0.867t^4 + 1.3706t^2 \\ &\quad -1.4171 \end{aligned}$$

The polynomial representation of the boundary was sectioned off into subdivisions based on the x-coordinates of the original points of the boundary that were used to make the polynomial fit. These points were chosen as dividers for the boundary sections as they represented the x-coordinates that truly existed on the original boundary, and not coordinates that

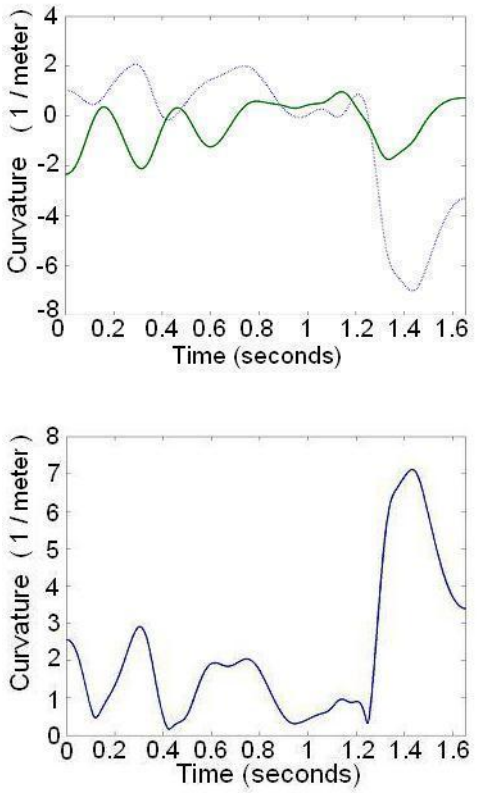

Fig. 11. Top: a plot of the u and v curvatures for trial 2 of July 1, 2008. The sharp decrease in curvature between time 1.2 and 1.4 seconds represents the point at which the bat abruptly changes directions, as seen in Fig. 12, where the bat's flight trajectory is plotted. Bottom: a plot of the w_b curvature for trial 2 of July 1, 2008. This is the nonnegative, normalized summation of the u and v curvatures.

were generated by the `linspace` MATLAB function, or by the polynomial approximation.

4) Determining Corresponding Points: In order to determine corresponding points on the bat's curvature and the approximated boundary, the bat's vocalizations were categorized based on (1) whether the vocalization was pointing toward the boundary at all and (2) which subdivision of the boundary the vocalization was directed toward. The vocalizations which were not aimed toward the boundary were not used in the correlation process, as they represented instances in which the bat was probably not attempting to use the boundary curvature in its navigation process. When vocalizing, the bat tended to focus its sonar beam direction on either the first half or the last half of each net section. Fig. 12 depicts the reconstructed boundary as well as the bat's trajectory and four sequential vocalizations within one trial, demonstrating one instance in which the bat focuses solely on the last half of the second section of the curved boundary.

The vocalizations depicted in Fig. 12 are in support of the previous statement that the bat's beam directions tended to focus on either the first or last half of the boundary sections. Curvatures were chosen based on each vocalization, thus the number of vocalizations and the number of curvatures extracted from the net were equal. In order to avoid using repeated curvatures on the net, and for simplicity of analysis, when the bat directed its sonar beam primarily on the first half of the net section, the sequential curvatures beginning at the first curvature defined in this section were used in the correlation process. When the bat directed its beam primarily on the last half of the net section, the sequential curvatures ending at the last curvature defined for this section were used. For example, if one section was grouped along with nine of the bat's vocalizations, and the bat directed its beam on the first half of that section, then the first nine net curvatures would be used in the correlation process. Similarly, if the bat directed its beam on the last half of the section, the last nine net curvatures would be used. The section of the net in which the bat was vocalizing toward was determined by a close examination of each of the vocalization intensity beams. The Sunshine generated fan plots for each vocalization were studied, and the intersection between the beams of highest intensity and the boundary were used in order to determine approximately where on the net the bat's beam direction was aimed.

The method of Regularized Inversion [14] was used to extract curvatures, denoted by u and v, from the bat's flight trajectory. In order to effectively compare the curvature of the bat's flight with the curvature of the net, the u and v curvatures were combined to form a scalar, w_b, as shown in Fig. 11, by the formula

$$w_b = \sqrt{v^2 + u^2} \tag{12}$$

which is a nonnegative representation of the steering control employed by the bat [18]. This resulted in a set of scalar curvature values that represented the overall curvature for the bat's position every 2 milliseconds in time.

D. Testing the Hypothesis

After extracting the curvatures from both the bat's flight trajectory and the boundary and determining the corresponding points on each, time delays were added and correlations were computed to determine the best time delay for each trial. The "best" time delay is represented by that which maximizes the correlation coefficient between the bat and boundary curvatures. The best time delay represents the amount that the bat's curvature should be shifted so that its corresponding points on the net boundary best correlate. A time delay of 2 milliseconds represents a shift of one "step" in the bat's overall curvature, or w_b, vector. A MATLAB script was created in order to calculate the correlations for a large span of time delays, ranging from 2 milliseconds to 350 milliseconds. The w_b vector was traversed depending on the time delay, so a delay of 2 milliseconds would traverse the vector by 1 step, whereas a delay of 350 milliseconds would traverse the vector by 175 steps. In this study, the equation for correlation coefficient is

$$r = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \overline{y})^2}} \tag{13}$$

where x_i is the bat's curvature, $\overline{x}$ is the mean of the bat's curvatures, y_i is the boundary curvature, $\overline{y}$ is the mean of the boundary curvatures, and N is the number of curvatures considered. This equation was used in order to calculate the

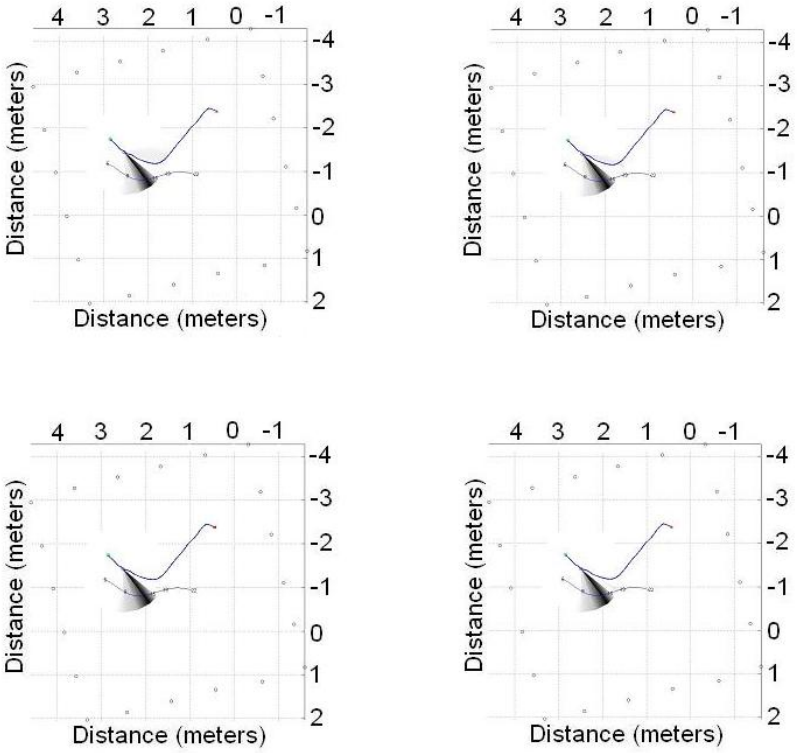

Fig. 12. Vocalizations 10 through 13 in trial 2 of July 1, 2008, plotted with the best fit polynomial approximation of the curved boundary. The highest intensity vocalization beams are directed toward the end of the second section of the reconstructed boundary. Shifting of the bat's sonar beam direction is barely noticeable within the bat's beam directions, thus the use of consecutive curvatures on the boundary approximation is a reasonable estimation.

correlation coefficient between the two sets of curvatures. The correlation coefficients were plotted against the varying time delays. The maximum correlation coefficient was calculated along with its corresponding time delay.

E. Discussion & Conclusion

Results of this study have thus far been inconclusive, yet promising. While the time delay from the analysis of the first trial was 98 ms with a correlation coefficient of 0.5630, the time delay from the second trial was 302 milliseconds with a correlation coefficient of 0.8856 and the time delay from the third trial was 340 milliseconds with a correlation coefficient of 0.7655, as illustrated in Figs. 13-15. However, all of these results indicate that the bat may be following a time delayed version of the boundary curvature. A more in-depth analysis of the experimental trials that were performed should not only give a more statistically sound set of results, but should also better test the time delayed curvature hypothesis among a wider range of data.

One of the problems with this experiment was that there had to be a goal or incentive that provided a reason for the bat to follow the curvature of the boundary. In this case, a meal worm was hung along the net boundary. However, while tracking the meal worm, the bat vocalized to determine the position of its target. It was thus difficult to determine

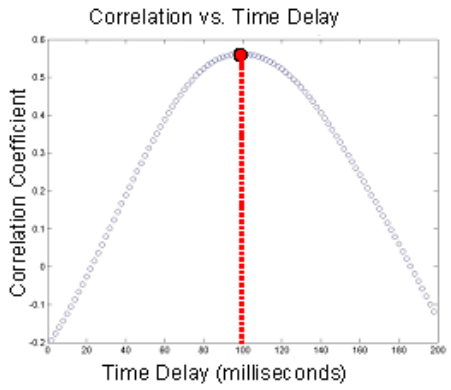

Fig. 13. The correlation versus time delay plot for trial 2 of July 1, 2008. The correlation coefficient is maximized at 0.5630 with a time delay of 98 milliseconds.

which vocalizations were used to determine the position of the meal worm versus which vocalizations were used to follow or avoid the net boundary. One possible explanation as to why the three trials are not similar is that the vocalizations that were directed toward the net boundary were used initially to determine where the meal worm was located. Thus, depending on the location of the meal worm and the time it took for

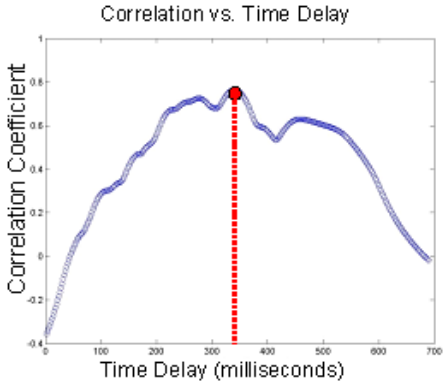

Fig. 14. The correlation versus time delay plot for trial 13 of July 1, 2008. The correlation coefficient is maximized at 0.8865 with a time delay of 302 milliseconds.

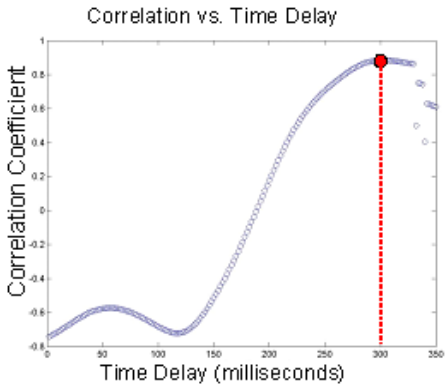

Fig. 15. The correlation versus time delay plot for trial 14 of July 1, 2008. The correlation coefficient is maximized at 0.7655 with a time delay of 340 milliseconds.

the bat to capture the worm, the time delay in boundary following would vary substantially. It thus becomes difficult to meaningfully correlate the bats curvature with the boundary curvature when there are others objects that the bat may be focusing its attention, or vocalizations, upon. Although it is very plausible that the bat was able to use a single vocalization for both tasks, it did not appear to do so within these analyzed trials. More analysis of the boundary following experiments that were conducted, as well as further study into the bat's neural processes and auditory abilities would be required in order to determine whether or not the bat is able to track prey and follow boundary curvatures simultaneously.

In the future, a different technique in reconstructing the boundary as a 2-D projection onto the xy-plane would be beneficial, as the current process is not as rigorous as is preferred. Although the sixth order best-fit polynomial approximates the boundary in a fairly reasonable manner, a more robust method would be more accurate in generating representations for various boundary shapes. The sixth order approximation may have been a reasonable fit for the S-shaped boundary, however the reasons for its use were largely subjective and based on a qualitative representation, as there were not many points used in generating this shape. In addition, a better method of identifying the boundary within the experimental trials would be useful as well. With a denser array of points within the bat's height range, more data points could be used to reconstruct the two dimensional representation of the boundary and thus the best fit polynomial would be significantly more accurate.

Additionally, selecting points on the boundary to be used for curvature extraction proved to be somewhat unreliable. Stating that the vocalizations followed the boundary curve sequentially was an assumption that was largely influenced by time constraints. In future steps, a more in-depth analysis of the exact boundary location which the bat was vocalizing towards would be a more reliable method of choice. A method such as summing the Sunshine analyzed vocalization beams to produce a beam-direction vector would be a better estimate of the position that the bat was vocalizing toward. Instead of using sequential curvatures on the net boundary, the sum of the directional beams for each vocalization would be calculated and the point at which it intersects with the boundary would be used in the correlation process.

V. Future Work

In this project, we have collected a large amount of experimental data for the obstacle avoidance and boundary interaction behavior of the big brown bat, *Eptesicus fuscus*. We have also taken initial steps in analyzing this data, comparing the obstacle avoidance behavior to the `open space` algorithm (as a model), and testing the time-delayed boundary curvature hypothesis. Future work will entail more detailed analysis, including a statistical exploration of our initial findings. In addition, we hope that these results will lead to development of feedback control laws that both describe the echolocating bat's behavior and inspire new navigation methods for mobile robots, especially autonomous unmanned aerial vehicles (UAVs).

VI. Acknowledgments

This research was made possible by NSF REU Grant CCF-0755224. The authors wish to thank Dr. T. K. Horiuchi, Dr. P. S. Krishnaprasad, and Dr. C. F. Moss for their guidance and excitement for the project. Working with them has been truly inspirational . Graduate students Ben Falk and Chen Chiu of the Auditory Neuroethology Lab (the "BatLab"), and Kevin Galloway and Matteo Mischiati of the Intelligent Servosystems Lab provided valuable mentoring and support throughout the summer.

References

[1] E. Rimon and D. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, October 1992.

[2] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, pp. 23–33, March 1997.

[3] P. Ogren and N. Leonard, "A convergent dynamic window approach to obstacle avoidance," *IEEE Transactions on Robotics and Automation*, vol. 21, no. 2, pp. 188–195, April 2005.

[4] D. R. Griffin, *Listening in the Dark: The Acoustic Orientation of Bats and Men*. Yale University Press, 1958.

[5] K. Ghose, T. K. Horiuchi, P. S. Krishnaprasad, and C. F. Moss, "Echolocating bats use a nearly time-optimal strategy to intercept prey," *PLoS Biology*, vol. 4, no. 5, pp. 865–873, 2006.
[6] W. H. Warren, "The dynamics of perception and action," *Psychological Review*, vol. 113, no. 2, pp. 358–389, 2006.
[7] P. Reddy, E. Justh, and P. Krishnaprasad, "Motion camouflage with sensorimotor delay," *Proc. 46th IEEE Conf. Decision and Control*, pp. 1660–1665, December 2007.
[8] E. W. Justh and P. S. Krishnaprasad, "Steering laws for motion camouflage," *Proceedings of The Royal Society A*, vol. 462, pp. 3629–3643, 2006.
[9] T. Horiuchi, "A neural model for sonar-based navigation in obstacle fields," *Proceedings of the International Symposium on Circuits and Systems (IEEE ISCAS 2006)*, pp. 4543–4546, May 2006.
[10] K. Ghose and C. F. Moss, "Steering by hearing: A bat's acoustic gaze is linked to its flight motor output by a delayed, adaptive linear law," *J. Neurosci.*, vol. 26, no. 6, pp. 1704–1710, 2006. [Online]. Available: http://www.jneurosci.org/cgi/content/abstract/26/6/1704
[11] E. Justh and P. Krishnaprasad, "Natural frames and interacting particles in three dimensions," *Proc. 44th IEEE Conf. Decision and Control, and 2005 European Control Conference, CDC-ECC '05*, pp. 2841–2846, Dec. 2005.
[12] P. Reddy, E. Justh, and P. Krishnaprasad, "Motion camouflage in three dimensions," *Proc. 45th IEEE Conf. Decision and Control*, pp. 3327–3332, December 2006.
[13] R. L. Bishop, "There is more than one way to frame a curve," *Amer. Math. Monthly*, vol. 82, no. 3, pp. 246–251, March 1975.
[14] P. V. Reddy, "Steering laws for pursuit," Master's thesis, Department of Electrical and Computer Engineering, University of Maryland (College Park, MD), August 2007.
[15] F. Zhang, E. Justh, and P. Krishnaprasad, "Boundary following using gyroscopic control," *Proc. 43rd IEEE Conf. Decision and Control*, vol. 5, pp. 5204–5209, December 2004.
[16] F. Zhang, A. O'Connor, D. Luebke, and P. Krishnaprasad, "Experimental study of curvature-based control laws for obstacle avoidance," *Proc. IEEE Int'l Conference on Robotics and Automation*, vol. 4, pp. 3849–3854, May 2004.
[17] E. W. Weisstein, "Curvature," 2008, read on 6 August 2008. [Online]. Available: http://mathworld.com/curvature.html
[18] E. Wei, "How do bats navigate within curved boundary?" May 2008, unpublished.