Introduction

Insects, while searching for mates or food, must explore their environment to detect sparse cues. Pheromones, carbon dioxide, and changes in temperature or humidity are all signals used to localize a target. Localization often has a temporal constraint, i.e., males compete to find the signaling female, or the larger host prey must be found before they move great distance. Mosquitoes and other insects solve this problem by employing two robust behaviors: 1) *casting*, a zigzagging crosswind flight, and 2) *surging*, a fast upwind flight. Casting is employed to gather information about the source location, while surging is employed to exploit estimations about source location. Together, these behaviors optimize search flight. It is unknown what signal information governs changes in behavior that allows for efficient source localization. The processing that leads to behavioral changes is also unknown, as well as what information is required to exploit the gathered information, *versus* further exploring the environment.

Thermosensation in mosquitoes provides a potentially exceptional model for search with minimal parameters because the system has only 4 sensors—a warm and cold sensor on the end of each antenna. Thermosensation does gate CO2 detection, and vice versa, but the ability to search with thermosensitive cues alone is unknown. This system success means we could develop a plume searching model with the smallest network possible—a single input (senor) and a single output (command).

We sought to model this system, by creating an agent-based model that searches for the highest intensity pixel in a spatiotemporally dynamic coordinate plane. Such a system would not only allow us to build hypotheses as to 1) the function and signal input while casting or surging and, 2) the decision rules which govern changes between the two search states. We also aim to augment our model such that the path flights are similar to, and similarly constrained as, mosquito flight. Thus, further developments will require intricate measurement of thermally driven mosquito flight.

Our agent model is built using a cascade of leaky integrate-and-fire (LIF) neuron, and exhibits flight characteristics (speed, angular velocity) matched to mosquito flight. The agent navigates a spatiotemporally varying intensity field, derived from videos of smoke plumes (courtesy of Mark Willis). The first LIF neuron, a sensor neuron, encodes the stimulus intensity along the flight path. The agent employs a casting behavior, governed by a crosswind sinusoidal pattern. The cast amplitude follows a second integrator, which rapidly decreases with the firing rate of the sensor neuron, but relaxes back to a maximum amplitude with insufficient inputs. When the amplitude of the cast approaches zero, the model surges.

Keeping our model as simple as possible, we chose four parameters to augment. Marching through parameter space allows us to target the range of values for which the agent can quickly and accurately localize the source. The agent system and parameters are described as follows.

Model

**I. Movement**

The agent movement on the coordinate plane is governed by a set of parametric equations:

|  |  |  |
| --- | --- | --- |
|  |  | (1) |
|  |  | (2) |

where, *cmax* is the maximum speed the agent can move, *a*(*t*) is the amplitude and *ωy* is the angular velocity in the *y*-axis oscillation, and *vwind* is the mean wind speed (0.45 cm s-1). Equation 1 acts to make the mean speed of the agent constant, regardless of the size of amplitude. That is, when the amplitude is large, the *x*-axis speed is low, and *vice versa*. The agent speed minimum, *cmin*, is equal to the wind speed. Thus, at this point in development, the agent may become stationary, but not move downwind.

The amplitude, *A*(*t*) is given an upper limit, *amax*:

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

This constraint exists for technical reasons—*Amax* values that are greater than *x*(*t*) result in imaginary speeds. Thus, we maintain a maximum amplitude, that is less than *cmax* but greater than some small difference *(delta?)*. *Amax* is also set such that the agent cannot oscillate out of the coordinate plane boundary. The model is ends a search if the agent travels out of the bounds of the coordinate plane.

**II. Sensory LIF Neuron**

The amplitude parameter is decreases over time, based on input through a sensory leaky integrate-and-fire (LIF) neuron. The sensor, *E*, integrates the intensity of its position (*I*(*x*,*y*,*t*)) as follows:

|  |  |  |
| --- | --- | --- |
|  | , | (4) |

where *τE* is the decay constant of the cell or membrane. Sensor spiking events occur when *E*(*t*) passes some threshold, θ. The responses (spike events) are tracked by *R*, which also resets *E* in order to impart a refractory period. The spikes are described as a Dirac delta function at time *t*. Spikes also change the centering of the *y*-axis oscillation, by changing the value of *yoff*.

|  |  |  |
| --- | --- | --- |
|  |  | (5) |
|  |  |  |

The output from E (spikes) control several outputs and behaviors. Recentering the *y*-axis oscillation, and signaling changes in the output neuron, *A*. Recentering is controlled by *yaim*, which in turn changes the *yoff*(*t*) in Eq. 1.

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

The change in *yoff* occurs exponentially, as is described by:

|  |  |  |
| --- | --- | --- |
|  |  | (7) |
|  |  | (7) |

where τy is the rate of change that the offset can maximally take. This parameter is held fixed in our simulations.

**II. Amplitude (Output) Control LIF Neuron**

Each spike from the sensory neuron E diminishes the amplitude of the movement, described in Equations 1 and 2. This behavioral output is controlled by a second LIF neuron, *A*.

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

Amax is determined as described in Equation 3; *τA* is the membrane constant of the neuron. To smooth changes in targeting and amplitude, and prevent them from being unnaturally sharp, the rate of change of the amplitude is constrained by giving the rate of change a Gaussian slope, as described by α:

|  |  |  |
| --- | --- | --- |
|  |  | (8a) |
|  |  | (8b) |

where *t’* is *l* timesteps or temporal width of the filter, σ is the standard deviation of the Gaussian, and *s* is the amount of change in amplitude that can maximally happen with each response event. We have empirically found Eq. 8b. Note that the filter can be temporally summed. If there are events in quick succession, the filter is summed or stacked, which preserves response to individual events and increases change in amplitude.

When the A(t) approaches zero, the oscillation in the y-axis (Equation 2) is zero, and only upwind flight occurs. This results in surging behavior.

Simulation

With this model, we can alter four main parameters: 1) the decay constant of the sensor, *τE*, 2) the response threshold, θ, 3) the decay constant of the output neuron, *τA*, and 4) the scalar change in amplitude, *s*. To do this, we have marched through parameter space. We run through instantiations of the search model in sets of 50 *iterations*, which we call *trials*. We run a trial in each of four videos, each analyzed separately. The trials were run with the following conditions:

1. **Initial positions-** There are three possible downwind positions to test, two outside of the plume on either side, and one in the middle. Three trials are carried out, at each position.



1. **Initial phases**- For each of the 50 iterations in a trial, there is a new initial phase chosen from the range 0:2π.
2. **Maximal or constrained speed**- We set the speed to the mean speed of mosquitoes flying in a wind tunnel with a carbon dioxide plume, approximately 30 cm s-1.
3. **Coordinate plane**- The videos have the same x and y dimensions. The length (number of frames) is set to the same value.

We ran trials with 3 values each of s and θ, and 25 values of *τA* and *τE*, and compared the times of successful searches. Once the search time is minimized, we will compare success rates.

Results