ME 586: Biology-Inspired Robotics

University of Washington, Winter 2022

Instructor: Prof. Sawyer B. Fuller

term project overview and project topic ideas

This document lists a few ideas for project topics in the general area of biology-inspired robotics and includes a suggested format for your poster presentation. You can think of the project as a sort of advanced problem set, equal in work to 2–3 regular problem sets, but more open ended, with a refined poster presentation of the results. The basic criteria for success is that your simulation/robot have some new capability, new way of doing something, or new understanding of an observed process, rather than for example wiggle randomly.

Poster format An easel and foam board backing will be provided for each team/project. The exact size of the poster can vary, but a suggested format is to pin/tape four standard printed sheets of paper together ($\sim 17'' \times 22''$). Elements to include will largely be in bullet point format (keeping words to a minimum):

1. **Summary** ($<$100 words) short summary of work you did, including main problems and how you solved them.

2. **Introduction and technical need.** Your area and its relation to other work. Here you will state what is not yet known, and describe who or what would benefit from a better understanding.

   For example, if your topic is ball catching, you might summarize the physics of a ball in flight and catching, and then describe (and give citations for) previous suggestions for how it is done or learned by humans and what is not yet known and how such an understanding might help train catchers.

3. **Results.** Describe the work you have done. Here you would include figures, equations, and graphs that explain any data you have collected from simulation results.

   For example, you could say that you have implemented a simulation of a ball-intercepting feedback law that shows that a simple servo-feedback controller exhibits characteristics similar to human ball catching.

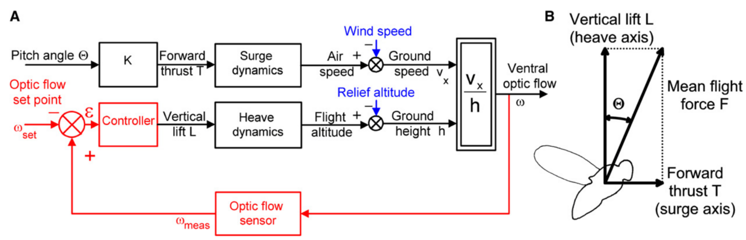
4. **Conclusions and future work**. What did you learn? What was most difficult? What was most surprising?

   For example, this might be that you found that by adding some element to your model it fits behavior better, and what the parameters are that match human behavior.

Associated papers that can be found on the course web page here.

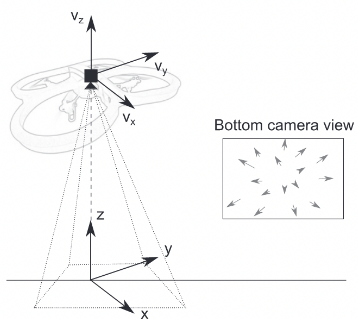
Simulation ideas that build on the crazyflie simulator you have been working with in the problem sets:

1. **Altitude regulation I.** It is unknown how insects regulate their altitude above the ground or distance to obstacles. It is likely they use optic flow, which is a measure of the speed of motion as seen by a camera. But optic flow does not provide a direct measurement of the distance to obstacles. One hypothesis suggests that insects may regulate their altitude by coupling *vertical thrust* to optic flow, rather than *forward thrust*. This can lead to a steady-state altitude. Such a controller would also descend in a headwind and rise in a tailwind, something that has anecdotally observed in flying insects. How well does this work on a helicopter, and in the wind?



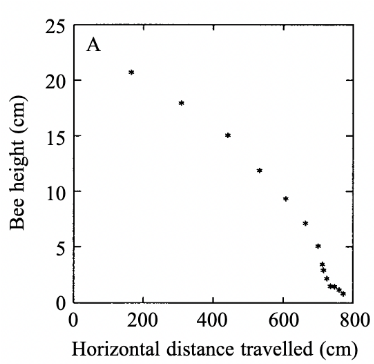
Reference: Franceschini2007: "A Biology-inspired robot sheds light on insect piloting abilities, *Current Biology.*

2. **Altitude regulation II.** Another possibility for altitude regulation holds that they may use a velocity feedback loop that has a time delay (caused by for example sensing + neural processing). When combined with a high feedback gain, this can cause instability, which is manifested by unstable oscillation. If the animal could detect these oscillations, it could use that to estimate distance. Can you implement the same behavior to estimate altitude on a simulated crazyflie that does not have a laser rangefinder? (Flying animals somehow get by without a laser rangefinder!)



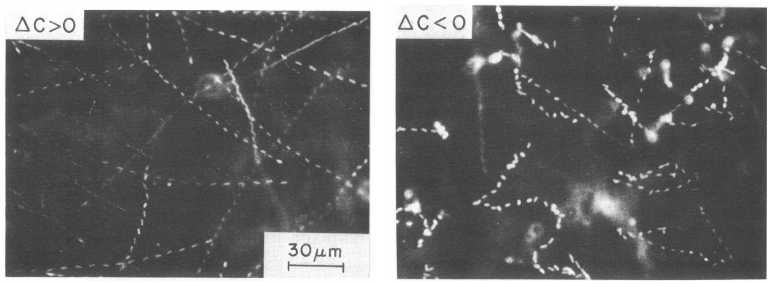
Reference: deCroon2016, "Monocular distance estimation using optic flow," *Bioinspiration & Biomimetics.*

3. **Grazing landings.** Evidence suggests that bees perform landing by keeping the optic flow below them constant. As they descend, this results in a slowing groundspeed, dropping to zero just as the bee intersects the ground. Can you implement this control law, without the laser rangefinder, to perform simulated landings?



Reference: Srinivasan1996: "Honeybee navigation en route to the goal," *Journal of Experimental Biology.*

4. **Search.** Bacteria are thought to find their way toward food by performing a "run-and-tumble" algorithm: go straight if they detect the concentration is going up ("$\Delta C > 0$"), and tumble to a random new orientation if they detect the concentration is going down ("$\Delta C < 0$"). Can you program a simulated robot to find a light source using a similar algorithm?



Reference: Macnab1972: "The Gradient-Sensing Mechanism in Bacterial Chemotaxis," *Proc Nat Acad Sci.*

Non-flying topic ideas:

1. **Improved Braitenberg Vehicle.** In problem set 1, we explored creating a simulation of "Braitenburg Vehicles": simple robots whose wheel speed depends on light or other environmental stimuli. In this project, you would add additional sensor types, actuator types, memory, or other capabilities. The idea is to build in one or two significant new features into the vehicle to make it exhibit more sophisticated behavior such as alternating between different food sources, or for example to use a neural network to learn how to find food sources. Reference: Braitenberg1984.

2. **Catching Simulation.** In Paper 0:McLeod1996, the algorithm by which fielders catch a ball was explored. In this project, you will write a simulation of the ball's trajectory, subject to wind drag and disturbances. Then, use your simulation to investigate the findings of the paper: what components are necessary to intercept the ball if they use the algorithm suggested in the paper? Does your simulation predict the behavior shown in Figure 3? If not, can you propose a plausible additional mechanism or dynamics model (that is, one that could realistically be implemented by a human, such as a proportional, or proportional-integral controller, for example) that can reproduce the observed behavior?