Modelling Social Biology with Drones Algorithm

Bishop Clark

University of South Florida

# Introduction

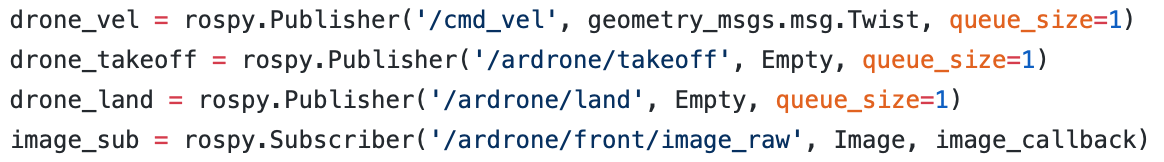
These series of programs are used to model sociobiological animals like bird flocks, wolf packs, schools of fish, and even ants. It involves a lead member followed by one or more other members of that group. The scripts use the ROS platform and Python programming language to model these groups.

## Objective

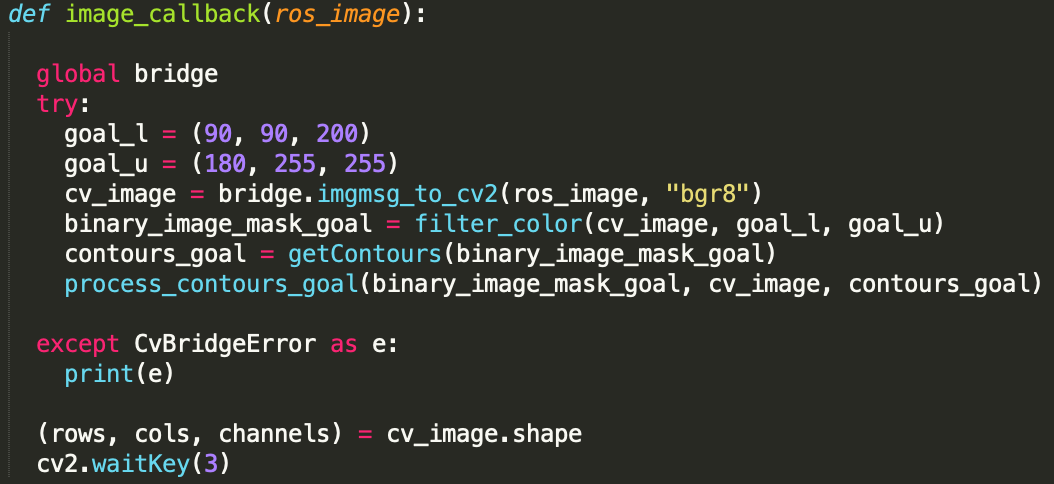
To design a collaborative navigation algorithm that accurately models a sociobiological animal group using the ROS platform and drones.

## Algorithm Design

The algorithm involve few of the same characteristics as a bug algorithm in that beta drones: take off, search for the objective (goal for each drone), moves toward the objective, and waits for further instruction. If the objective decides to move, the beta drone will move toward the objective. If the objective is motionless, so too is the beta drone. Motion control used the ROS platforms’ preexisting linear and angular motion functions. The following figure initializes the drones ability to traverse, take off, land, and use the front camera respectively.



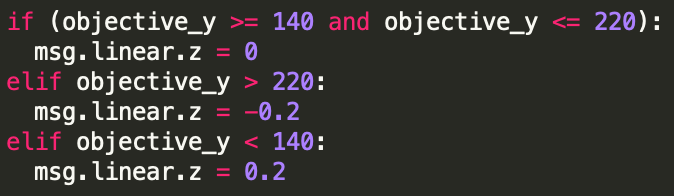
We will explain the most important part of the algorithm first, the camera feedback. The camera is initialized by the last line in the above figure. It allows the drone to transfer images to the host machine (Ubuntu) and the host will process these images and depending on the condition it will command the drone to perform certain actions. Because each drone is run by separate computers, each drone also has an exclusive script. If modelling a eusocial ant colony, the “alpha” should not be followed by the ant preceding the ant behind it. In other words, a drone should follow the drone in front of it, not two drones in front. This requires each drone to follow a specified color so that no beta drone will follow the incorrect drone.

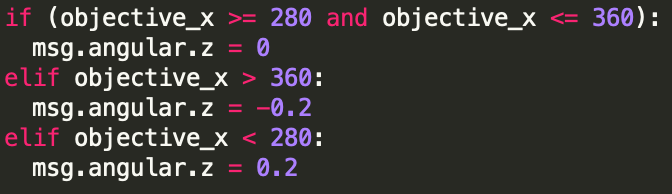
The following figure is an image call back function that allows the color specified to be processed and contours to hone on. The lists are the upper and lower HSV values of pink and green used in

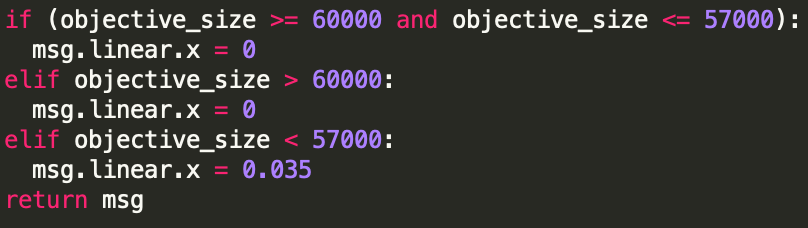
experimentation. The lines beneath create an OpenCV image on the host and creates binary images that designate something to be or not to be the object. It will then process the contours of its specified color. It sends information like the area and perimeter of the contour, the x and y values of the contour on the image, it can even tell you how many contours are in the image.

This information is set as a global variable on the host so that separate functions can review this data and process it accordingly. In experimentation, it was necessary to focus on one contour only (the objective) and remove any color from the environment that resembled the objective color.

The important data like the x and y coordinates and objective area are used by the motion algorithm. We use the x, y coordinates to determine where the objective is relative to the orientation of the beta drone. For example, if the objective is located at position (420, 80) the beta drone is instructed to rotate right until the objective (contour) is between 280 and 360 pixels and descend downward until the objective is between 140 and 220 pixels. The next data member the drone uses is the area. The area is calculated simply by taking the radius of the contour and squaring it, then multiplying by . The beta drone should approach the objective until the area is greater than or equal to 57000.







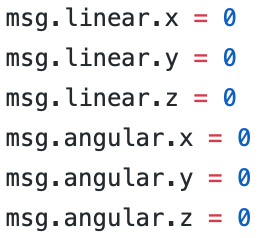
## Complications and Solutions

There were several complications experienced during testing. The most notable ones, and the ones likely to occur for future researchers are red lights on rotors, unstable movement, inability to connect to drone.

Notice that if the drone crashes into an object with enough force, the it will immediately deactivate, fall to the earth, and red lights will show under each rotor. This means that the drone went in to safety mode and cannot fly any longer. To resolve this, disconnect the battery and reconnect it. The “drone.launch” file will lose connection and begin searching again. Eventually it will reconnect, and the drone should be able to fly again.

Occasionally the drone may collide into an object with enough force to damage and compromise rotor integrity by misaligning blades or gears. This incident is considered fatal and the drone should not be flown as its movements are sporadic and dangerous. In order to fix this problem, you will need to purchase tools and parts to repair the rotor.

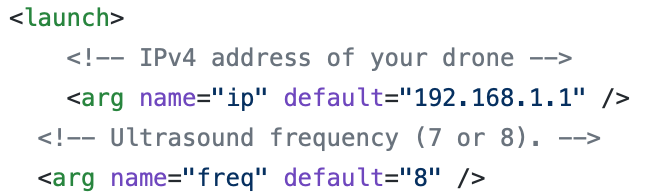
Unstable movement is a very common problem when designing drone motion control. It should be noted that any time a msg.linear or msg.angular member is modified from 0, the drone will be out of stability mode.



The above image is stability mode. In this mode the drone will hover in a single spot after takeoff. Any modification may induce unstable movement to the drone creating undesired results. This is why it is important to stabilize the drone frequently during execution to provide a more stable flight. Another reason the drone flight may be unstable is low battery power. This usually happens when battery power is below 20%. The rotors will not generate as much force to propel the drone any longer and as such, the battery should be recharged for further flight.

Another way to stabilize the drone is by calibration. This requires a smartphone. Download the free AR.FreeFlight app on the App Store, power on the drone, plug in the battery, connect to the drone using its WiFi connection. Open the app on the smartphone and enter the “Piloting” tab. On the bottom of the screen you will see a “Rescue” button, tap it. You can calibrate by tapping both of the bottom two buttons on this interface. This will calibrate all four motors on the drone and the drone should be more stable on takeoff and during movement.

The final and perhaps most frustrating complication is inability to connect to the drone through the “roscore” command on the terminal. To resolve this, the parameter in the “drone.launch” file may need to be changed.



The IP address may need to be changed to “192.168.1.2”. If this does not work, then the computer’s host name needs to be changed. Make sure the computer is wirelessly connected to the drone’s access point. On a new terminal enter the command: “ipconfig -a”, then retrieve IP address. I refer to a Google search to learn how to change your host name to your retrieved IP address.

## External Resources

These resources provided exceptional aid to this project. They also give troubleshooting and solutions to common problems that continuing researchers may have.

* ROS Wiki Documentation (wiki.ros.org)
* GaiTech EDU (https://edu.gaitech.hk/drones/ar\_parrot\_2/ar-parrot-2-ros.html)
* Udemy (https://www.udemy.com/ros-essentials/?couponCode=ROS1GAITECHEDU)

## Conclusion

This study exhibits a successful modelling of various socio biological systems. It does however have its limitations. The algorithm stipulates fixed-size objects. Because the area is calculated, a small objective that is very close to the drone would appear the same size on the image mask as a large objective if the drone were very far away. Another limitation are the computers it requires to run all of the drones. A future revision would be to use a USB Wi-Fi module to simultaneously connect to Wi-Fi networks sending each drone separate commands from the host computer. Nevertheless, this project demonstrated many of the applications that can be utilized on the drone creating a complete alpha-beta motion model.