UAV Swarm Mapping Using a Fully Distributed Control Approach

Tristan Sherman, Aerospace Engineering

Mentor: Dr. Scott Boskovich

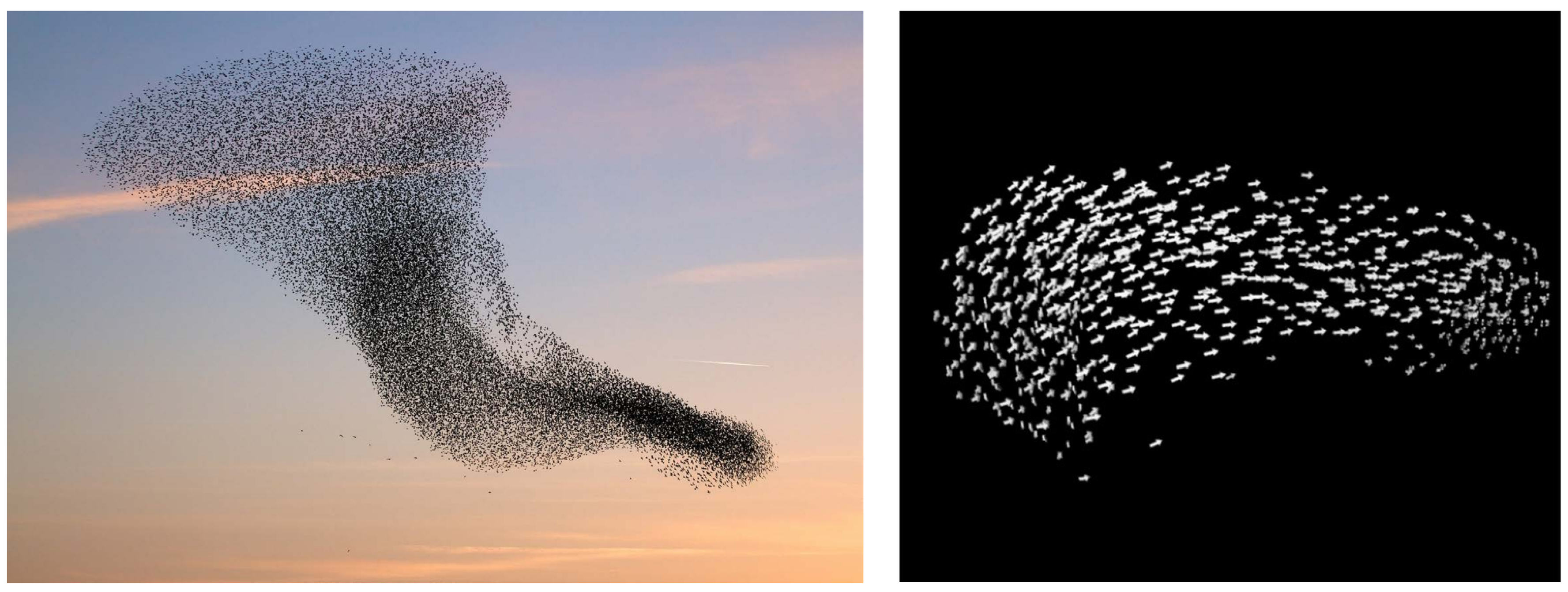
Kellogg Honors College Capstone Project

Abstract

Since their emergence, small unmanned aerial vehicles (UAVs) have been widely used in applications ranging from photogrammetry to search and rescue and have the ability to increase the efficiency of many others involving dull or dangerous tasks. Typically a single UAV is used, however the use of many UAVs at the same time (a UAV swarm) can increase the scope of the mission and reduce overall time to completion while improving redundancy in case of a failure of an individual vehicle. This research explores the use of fully distributed control in a UAV swarm for the purpose of terrain mapping. The Boids algorithm is used as the core control behavior to allow basic navigation and collision avoidance between all UAVs. Additional behaviors including eccentricity, velocity gradients, boundary avoidance and boundary cohesion are added to increase efficiency of the system. The UAV swarm is simulated in V-Rep and connects to Python scripts through a remote API to log mapped areas. Combinations of the additional behaviors are tested against each other to determine which are most efficient when applied to mapping. When compared to a single vehicle, most behaviors of the UAV swarm reduced the total mapping time by more than half.

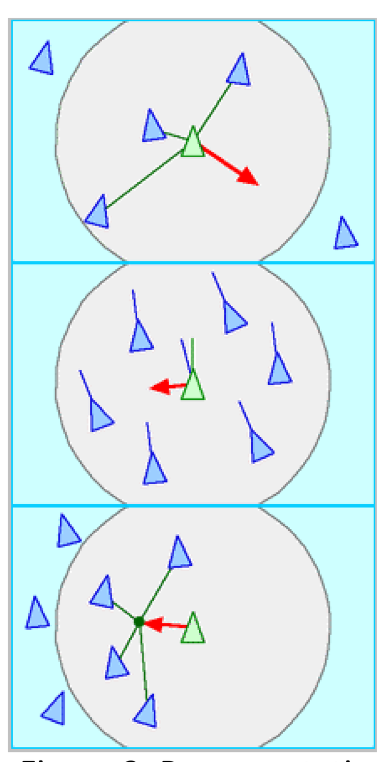
Background

The Boids algorithm, developed by Craig Reynolds[1] in 1986 was chosen as a base control algorithm for its simplicity and size. By continually adding together three simple steering behaviors (cohesion, separation, and alignment), it is able to model the complex flocking behaviors that occur in nature. It is a fully distributed control approach meaning each UAV in the swarm makes movement decisions independent of an external computer. Because of this the processing power is much lower and the vehicles size can be lowered.



(a) (b)

Figure 1: Bird swarms in nature (a) and an implementation of the Boids algorithm modeling the swarm behavior (b)



Separation

This vector steers the UAV away from any other UAVs in its neighbor group to avoid collisions. The magnitude of the vector increases inversely with the distance between UAVs

Alignment

The alignment vector motivates the UAV in the average direction of all UAVs in its neighbor group.

Cohesion

The cohesion vector attracts all the UAVs in a neighbor group towards the average of their center locations.

Figure 2: Representation of the three Boids behaviors and their corresponding steering vectors.[1]

Mapping Behaviors

The three behaviors used in the Boids algorithm are useful for swarm modeling, however they are unoptimized for any application outside of that. To apply this algorithm to mapping, four more behaviors were created to increase efficiency of the system.

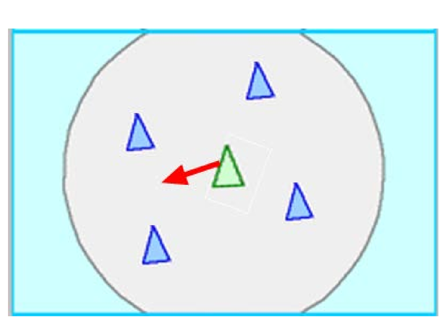


Figure 3: Eccentricity vector pointing in a random direction out of the cluster

Eccentricity

The eccentricity vector is implemented to reduce the tendency to conglomerate by adding a randomness factor to the velocity. After a UAV has been part of a small cluster of vehicles for a certain period of time, it will gain a large pull in a random direction until it has left the sphere of influence of the cluster.

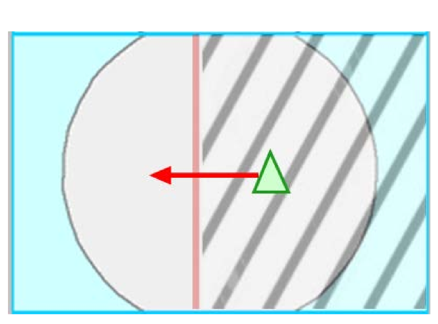


Figure 4: UAV outside the map area with a velocity vector towards the middle

Boundary Avoidance

The boundary avoidance behavior forces the UAVs to remain inside a specified area. It is assumed that the mapped area will always be provided, and therefore the boundary conditions are known. When a UAV passes the border of the boundary it gains a vector pointing normal to the surface towards the inside of the area to be mapped.

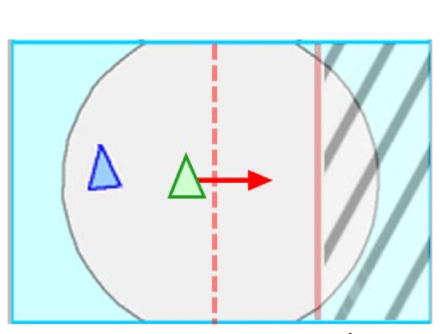


Figure 5: Vector returning the UAV closer to the boundary

Boundary Cohesion

With the boundary avoidance behavior enabled, the algorithm has a tendency to miss small portions of map closest to the border. Boundary cohesion is implemented to reduce this inclination. When a cluster of UAVs is near a boundary, the closest vehicle to the boundary will be forced to stay within a small distance from it, allowing the edges to be mapped quickly.

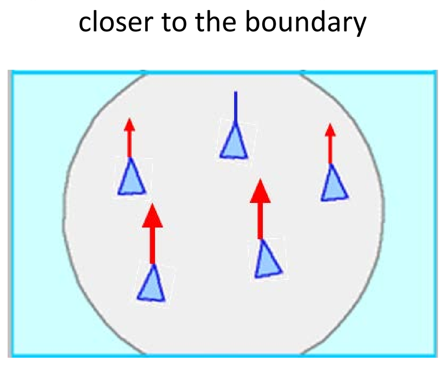


Figure 6: UAVs in the back of the cluster have larger velocities than UAVs in front

Velocity Gradient

The velocity gradient vector adds speed to UAVs in the back of a traveling neighbor group depending on how many UAVs are ahead of it. Vehicles near the back will travel faster and vehicles in front will travel slower. This causes the swarm to assume a line instead of a cluster while traveling.

Simulation

To determine the effectiveness of the custom mapping algorithms a swarm of 10 UAVs was simulated in V-Rep. Each UAV was represented by a green sphere accounting for position variance and was connected to a Python script which implemented all the algorithms. The simulation included a 10 by 10 square which represented the area that was to be mapped. In the Python script a corresponding matrix stored the locations where the simulated UAVs had traveled to determine when mapping was complete. The simulation and matrix can be seen in figure 7.

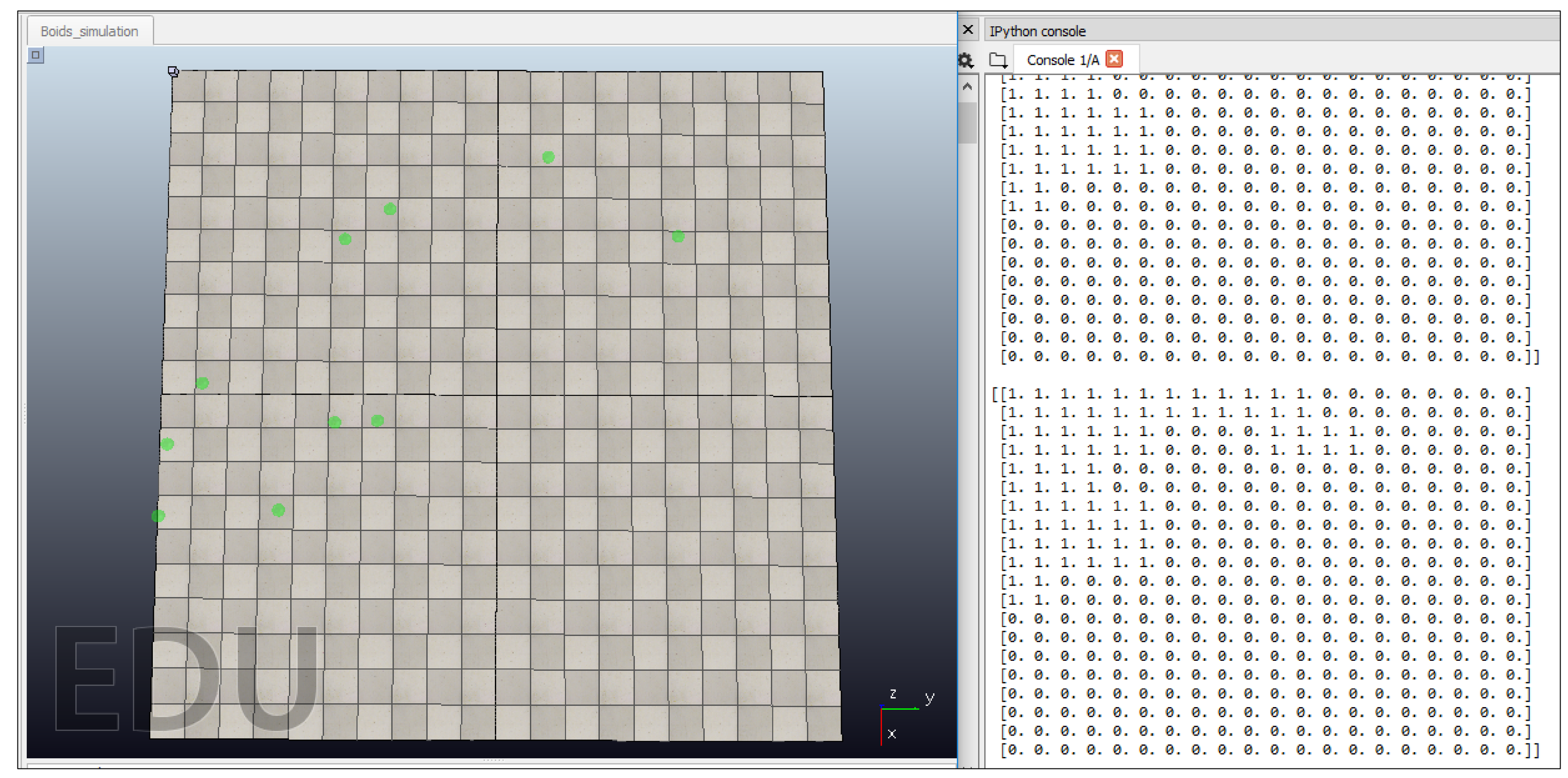


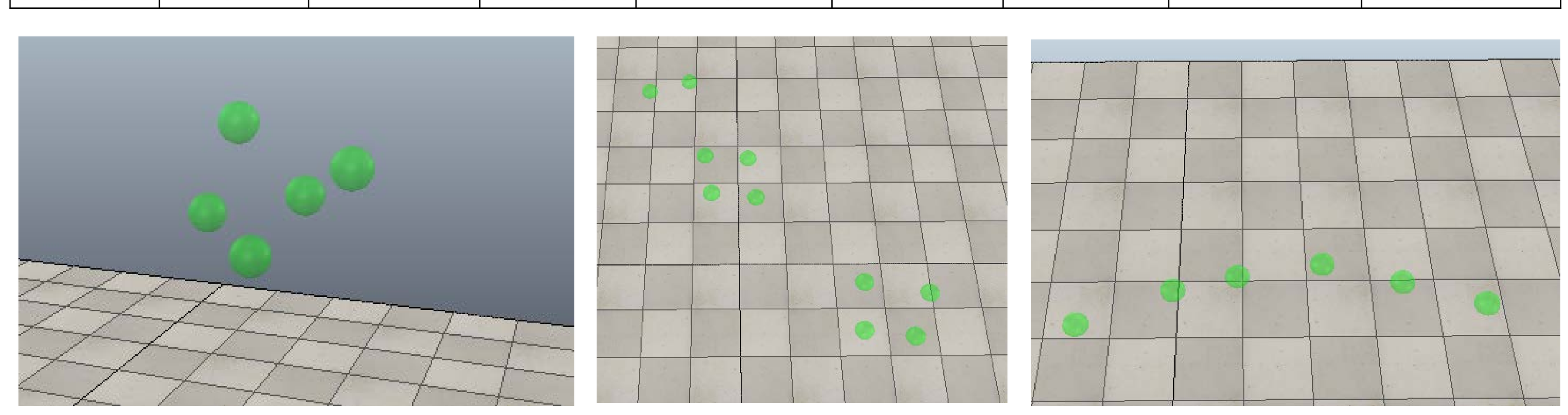
Figure 7: V-Rep simulation (left) and corresponding matrix showing mapped area (right)

Testing and Results

To determine the effectiveness of a UAV swarm, the time taken by the swarm to map the square was compared to the time taken by one UAV to map the same area. Different combinations of the mapping behaviors were combined showing how each affected the efficiency. Table 1 shows the complete results from testing. Five tests were done for each combination that included the eccentricity behavior and their results averaged to account for randomness.

Table 1: Results of testing where "X" means the behavior was included in the test

Test	Cohesion	Separation	Alignment	Boundary Avoidance	Boundary Cohesion	Velocity Gradient	Eccentricity	Time to Map (seconds)
Baseline	Linear mapping with one UAV							66.7
1	X	X	X	X				70.2
2	X	X	X	X	X			40.6
3	X	X	X	X		X		23.0
4	X	X	X	X			X	33.3
5	X	X	X	X	X	X		29.7
6	X	X	X	X	X		X	45.3
7	X	X	X	X		X	X	42.0
8	X	X	X	X	X	X	X	36.6



(a) (b) (c)

Figure 8: Boids implemented in 3D (a) and Boid clusters in 2D (b). Then a UAV cluster with velocity gradient behavior (c)

Discussion and Future Work

As can be seen in Table 1 the UAV swarm was able to map the area faster than a single UAV for every combination of behaviors except simple Boids with boundary avoidance. However, the time reduction achieved with the swarm was not significant enough to justify using 10 UAVs instead of one in most cases. Even so, the results show promise for the use of a distributed control algorithm for mapping if further developed.

The time to map with these behaviors is highly dependent on other parameters within the algorithm such as neighbor distance and separation distance. With further tuning and optimization the swarm will be able to map much faster than the current results show. Future work will not only include behavior optimization, but also the creation of additional behaviors that add awareness of the mapped/unmapped areas. One example of this could be attraction towards unmapped areas.

References

1. Reynolds, C. W. (1987) Flocks, Herds, and Schools: A Distributed Behavioral Model, in Computer Graphics, 21(4) (SIGGRAPH '87 Conference Proceedings) pages 25-34.
2. Petráček, Pavel and Martin Saska. "Decentralized Model of a Swarm Behavior Boids in ROS." (2017)