Coursework 2

UAV SWARM SIMULATION

January 29, 2016

Aidan Scannell

1 Abstract

This report will outline a software simulation of a UAV swarm tracking a pollutant cloud. A discrete time state space model of the world was produced and a finite state machine was used to add intelligence. It was assumed that the UAVs were at constant altitude and unaffected by wind. Their maximum turning curvature was 6 $°m^{-1}$ and their speed range was between 10 ms^{-1} and 20 ms^{-1}. They were limited to fly within a square of side length 2 km, could only sense their position via GPS subject to a $\pm 3m$ error, had an endurance of 30 minutes and were also capable of taking concentration measurements. The UAV swarm was required to locate any pollutant clouds in the area and track their 1 PPM contour while spreading out along the cloud to ensure the complete perimeter was tracked. The inputs to the UAVs were velocity and curvature and the dynamics were governed by the following equations: $\dot{x} = v \sin\theta$, $\dot{y} = v \cos\theta$ and $\dot{\theta} = v\mu$.

The simulation is capable of using multiple agents to locate and track the clouds. It uses several swarm behaviour features that reduce the number of collisions and enables the swarm to spread around the cloud. Unfortunately the system for deploying multiple waves of agents was poor and resulted in errors with the second and third waves.

2 Guidance System

The finite state machine guidance system is shown in Figure 1 and the states and transitions are described in Table 1. The discrete time model assumes that the velocity and curvature changes occur over the time step, so time steps too small can result in impossible accelerations and turns. Smaller time steps also increase simulation time and can result in the computation time being larger than the step size for the processor on board the UAVs. The communication systems has a 1 second delay, so reducing the time step below this would increase the complexity of the simulation. For these reasons a time step of 3.6 seconds was selected for the simulation.

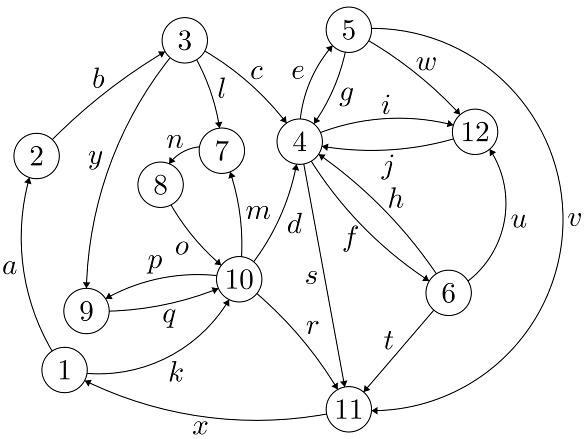

Figure 1: Finite State Machine (FSM)

2.1 Search Strategy

All UAVs are initially parked at the base in State 1 where a third of them are set to active. The number of agents defined at the start of the simulation is equivalent to three times the number initially deployed i.e. the guidance strategy consists of three waves. Active agents enter State 2 where they spread out across the map and then enter State 3 where their movement is defined by a spiral as can be seen in Figures 2 and 3 respectively. This offers an effective and efficient search strategy for searching the area for any pollutant clouds. Deploying multiple agents at the start results in a larger area being covered per unit time, which reduces the search time. It was assumed that the agents would not collide while taking off, which is realistic as operators can assist takeoff. The search strategy could be improved by implementing random movement once the agents have spiraled for a certain amount of time; this would increase the chances of pollutant clouds not within the

Table 1: States and transitions

State	Description	Transition Label	End State	Criteria	State	Description	Transition Label	End State	Criteria
1	Remain at base	a k	2 10	IF agent active and t == 0 IF agent active and t >1000s	7	Collision avoidance (turn 180 degrees)	n	8	IF turned 180 degrees
2	Spread agents	b	3	IF t >54s	8	Collision avoidance (travel in straight line)	o	10	IF collision flag off
3	Spiral agents	c l y	4 7 9	IF PPM >0.6 IF another agent within collision zone IF within 200m of boundary	9	Prevent agent from leaving boundary	q	10	IF not within 200m of boundary
4	Track cloud	e f s i	5 6 11 12	IF another agent present behind IF another agent present in front IF flight time >1400s IF within 200m of boundary	10	Move towards cloud	d m r	4 7 11	IF PPM >0.6 IF another agent in collision zone IF flight time >1400s
5	Speed up agent	g w v	4 12 11	IF no agent present behind IF within 200m of boundary IF flight time >1400s	11	Return to base	x	1	IF within 50m of base
6	Slow down agent	h u t	4 12 11	IF no agent present in front IF within 200m of boundary IF flight time >1400s	12	Prevent agent from leaving boundary	j	4	IF not with 200m of boundary and tracking flag == 1

spirals being located. The agents build a map of everywhere the agents have been as well as a map of the last measured concentration at every point. These could be used to make the agents search areas that haven't previously been searched or were searched a long time ago. The search strategy could also be implemented continuously so that agents are searching the area whilst other agents are tracking the cloud.

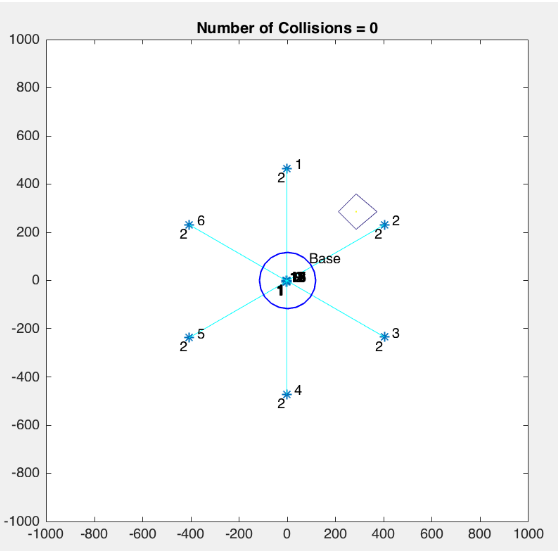

Figure 2: State 2 - Spread agents

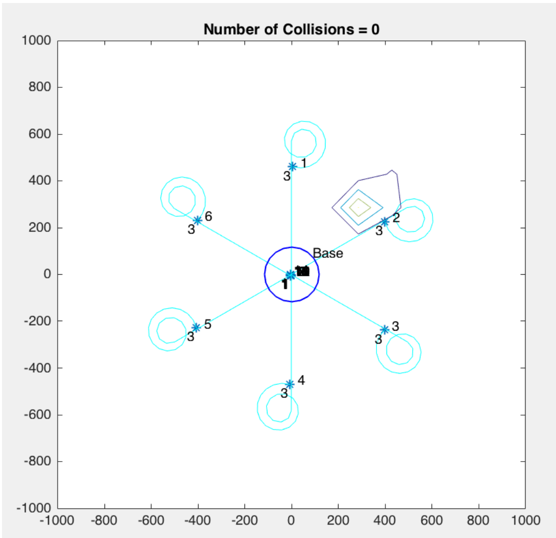

Figure 3: State 3 - Spiral agents

2.2 Tracking Strategy

The swarm is required to track the 1 PPM contour of the pollutant cloud, which is achieved using State 4. The tracking system originally consisted of 4 cases, switching between them dependent on whether or not the the agent was inside or outside of the 1 PPM contour. It used concentration gradients to determine the curvature and resulted in abrupt movements and poor overall tracking, as can be seen in Figure 4. This tracking system was not robust as it was difficult to implement spreading strategies, so a better system was required.

This was achieved using a PD controller and the results can be seen in Figure 5, showing a swarm tracking the 1 PPM contour for Cloud 2. The error between the measured concentration and the desired concentration (1 PPM) is calculated and used for the controller. The gains are dependent on the agents velocity, which is set to 12 ms^{-1} when tracking the cloud. This enables the agents within the swarm to speed up or slow down when they are tracking the cloud, allowing them to spread out. The controller offers an efficient solution allowing each agent to track the 1 PPM contour using one case, offering an efficient and robust solution.

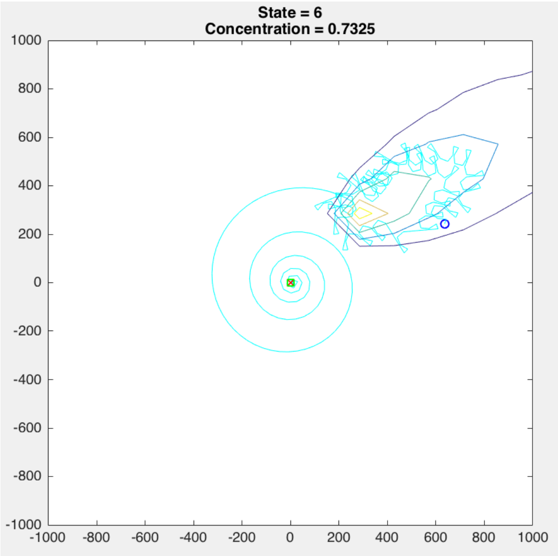

Figure 4: Original single agent tracking

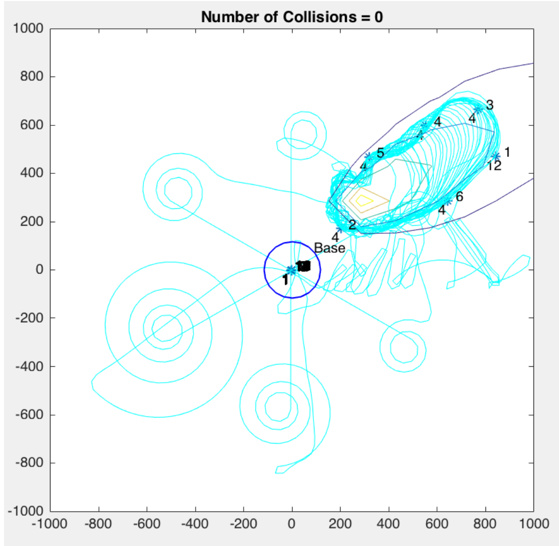

Figure 5: State 4 - Six agents tracking

2.3 Spreading Strategy

The tracking controller is beneficial for the swarm as it reduces complexity and enables the agents to work cooperatively. They are spread around the cloud by speeding up and slowing down the agents using States 5 and 6 respectively. Agents are slowed down from 12 ms^{-1} to 10 ms^{-1} when there is an agent present in front of them and are accelerated from 12 ms^{-1} to 14 ms^{-1} when an agent is present behind them. This is achieved by making every agent send their current position via the communications channel and determining if these points are in front or behind of each agent. Figure 6 shows the zones in front and behind of each each agent that are used to spread them evenly along the 1 PPM contour. This swarm behaviour ensures that the whole perimeter of the cloud is tracked, preventing the agents from clustering at certain points and helping prevent collisions between the agents tracking the cloud. Tuning the PD controller for different velocities ensures that agents traveling faster don't oscillate about the 1 PPM contour and as a result travel slower around the perimeter of the cloud than the slower agents. This allows the spreading strategy to be implemented effectively. As the cloud increased it was observed that the spreading strategy is less effective and that more agents are required to ensure good spreading over the entire perimeter.

Table 2 shows the effect of removing the spreading strategy on a number of performance metrics. The data was obtained for the simulation with Cloud 2 for the first 1500 seconds. This was due to problems with the simulation after 1500 seconds, which will be discussed later. There is no effect on any of the performance metrics other than the number of collisions and the range of distances between tracking agents. The number of collisions increases significantly due to the fact that the spreading strategy acts as the only collision avoidance for agents tracking the cloud. This results in large numbers of collisions within the cloud. The range of distances between agents tracking the cloud increases, significantly reducing the ability of the swarm to spread around the 1 PPM contour. This shows that the spreading strategy is extremely effective and that it is crucial for the simulation to perform well.

Table 2: Evaluation of Spreading Strategy (Cloud 2, t = 1500s, 6 Agents)

Feature		**Number of Collisions**	**Time to Find Cloud (s)**	**Time for All Agents to Track Cloud (s)**	**Range of Distances Between Tracking Agents (m)**	**Computation Time (s)**
Spreading Agents	On	1	151.2	1060	smallest - 205, largest - 312	105
In Cloud	Off	200	151.2	1060	smallest - 10, largest - 998	100

The spreading behaviour makes it harder for agents not tracking the cloud to enter the cloud and start tracking. In future the agents could spread around the cloud but group together to allow extra agents to start tracking the cloud. This cooperation would reduce the number of collisions when agents start tracking the cloud and would increase the number of agents able to track the cloud.

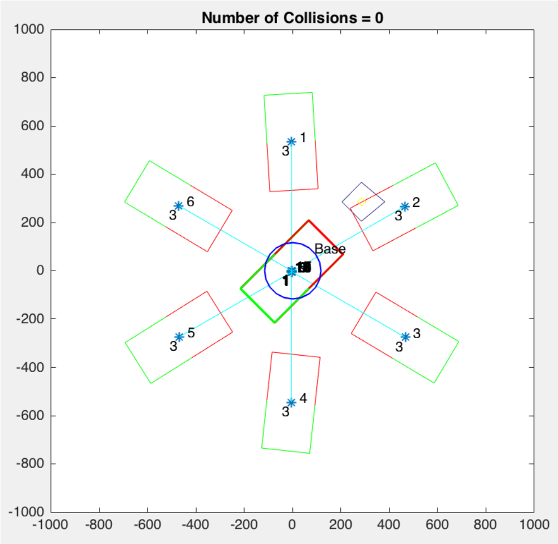

Figure 6: Zones used for spreading agents within cloud

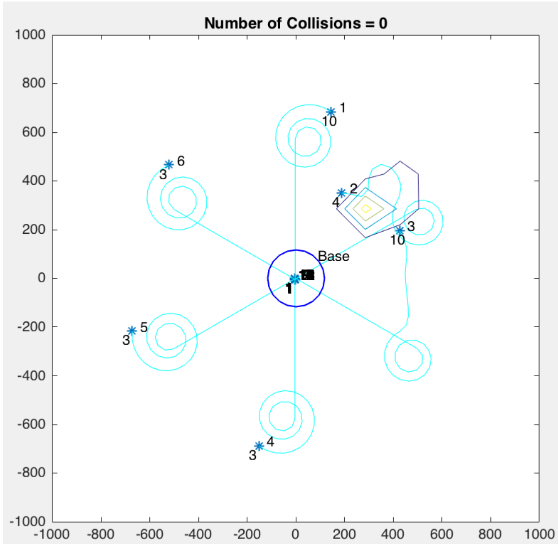

Figure 7: Closest agents moving towards cloud

Figure 7 shows how the swarm continues searching the area whilst the agents closest to the pollutant cloud moved towards it. Agents only move towards the cloud if they are the nearest, excluding the ones tracking the cloud and if all of the tracking agents within the cloud are separated by 100 m or more. This enables the swarm to locate multiple pollutant clouds if they are present near the start of the simulation. This cooperation between the agents also prevents the area surrounding the cloud from becoming overcrowded whilst the cloud is small. This makes it easier for the agents to start tracking the cloud without collisions and allows the agents tracking the cloud to spread out. It also prevents too may agents attempting to track the cloud whilst it is small, which is advantageous.

Table 3 shows the effect of removing the feature where only the nearest agent fly's towards the cloud once it is located. The time for all of the agents to start tracking the cloud is reduced as all of the agents move directly towards the cloud once it is located. The number of collisions increases due to the area surrounding the cloud becoming over populated. The range of distances between tracking agents is large due to the fact that agents start tracking the cloud too close to other agents. This results in agents flying directly on top of each other and are therefore unable to spread. This is not possible in real life so it can therefore be concluded that the behaviour allowing only the closest agent to fly to the cloud reduces the number of collisions and improves the swarms performance.

Table 3: Evaluation of Nearest Agent Fly Towards Cloud Feature (Cloud 2, t = 1500s, 6 Agents)

Feature		**Number of Collisions**	**Time to Find Cloud (s)**	**Time for All Agents to Track Cloud (s)**	**Range of Distances Between Tracking Agents (m)**	**Computation Time (s)**
Nearest Agent Fly Towards Cloud	On	1	151.2	1060	smallest - 205, largest - 312	105
	Off	200	151.2	915	smallest - 3, largest - 350	138

2.4 Collision Avoidance

Collision avoidance is implemented for each agent by determining if any agents are present within an area in front of them and turning the agent by 180° if there is another agent present. The detection zone used can be seen in Figure 8 and is effective for preventing head on collisions and collisions between agents joining the cloud and agents tracking the cloud. Figure 5 shows the path of the agents before they start tracking the cloud and it can be seen that agents trying to enter the cloud move around the edge of the cloud until they are able to successfully enter the cloud. This is due to the collision avoidance preventing any agents from colliding. The behaviour of the swarm is designed to minimise how often the agents need to use the collision avoidance. Allowing only the nearest agent to fly towards the cloud prevents all of the agents from colliding while trying to start tracking the cloud. The spreading of agents within the cloud reduces the need for collision avoidance within the cloud, although it should be noted that if an agent manages to enter the cloud too close to another agent there is the possibility for collisions. This must be reduced before developing the real life system.

Table 4 shows the effect of removing collision avoidance from the simulation. The table shows that the number of collisions increases but it should be noted that it doesn't have as large an impact as removing other features, such as only letting the nearest agent fly to the cloud. This emphasises the fact that swarm behaviour is more important than crude collision avoidance on its own. Collision avoidance also reduces the amount of time for all of the agents to be tracking the cloud but this has no benefit due to the number of collisions. It also reduces the computation time but not by a large amount.

The collision avoidance could be improved with inspiration from nature, where agents fly along paths determined by their task. This would enable the swarm to work cooperatively and move to achieve an end goal, as opposed to crude collision avoidance if they detect another agent. The collision avoidance cannot detect real life obstacles such as birds due to the lack of sensors and this should be considered when designing a real life system.

Collision avoidance is affected by the time step, with lower time steps improving the performance of the collision avoidance. The 3.6 second time step made it difficult to remove all collisions and should be reduced if the processor on board the UAVs allows so. The velocity and curvature changes must also be considered when reducing the time step.

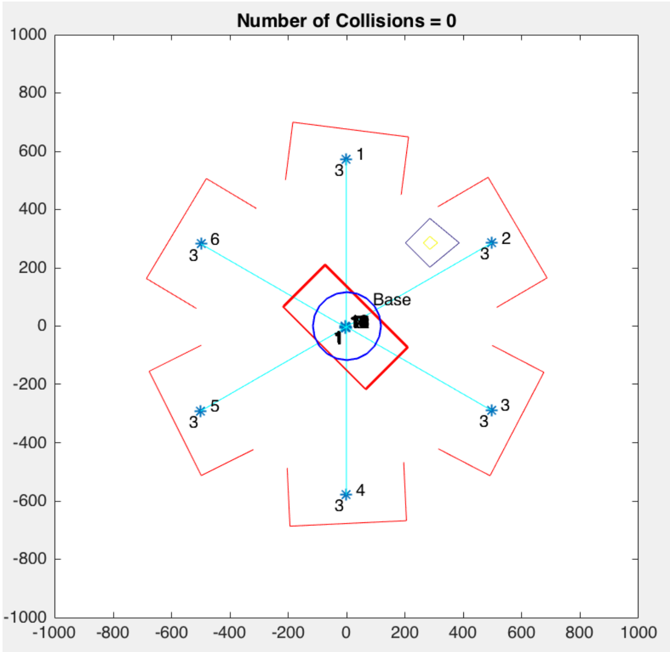

Figure 8: Collision zones

Table 4: Evaluation of Collision Avoidance (Cloud 2, t =1500s, 6 Agents)

Feature		**Number of Collisions**	**Time to Find Cloud (s)**	**Time for All Agents to Track Cloud (s)**	**Range of Distances Between Tracking Agents (m)**	**Computation Time (s)**
Collision Avoidance	On	1	151.2	1060	smallest - 205, largest - 312	105
	Off	8	151.2	464	smallest - 203, largest - 372	94

2.5 Returning to Base & Preventing Agents Leaving Flight Zone

The agents are prevented from traveling out of bounds by checking that their GPS position is not within 200 m of the boundary and turning them back towards the base if they are. They are then able to transition into further states and move towards the cloud.

When an agent's flight time approaches its endurance limit it returns to base and is replaced by another agent. The agents work cooperatively and only return to base if they are the closest to the base, which reduces the number of collisions. The agents at the base use the communications channel to determine when they should take off and require no external assistance. The implementation of the replacement of agents is poor and the second wave of agents get stuck in collision mode with agents parked in the base. This could be removed to enable the second wave to successfully deploy and track the cloud. Removing the feature where only the closest agent returns to base significantly increases the number of collisions as returning agents fly straight through the center of the cloud and into other agents. It is clear that cooperative behaviours such as this are vital for the success of the swarm.

3 Results & Evaluation

The performance metrics in Table 5 were obtained from the first 1500 seconds of the simulation with cloud 2. It can be seen that as the number of agents increases the search time reduces, the number of agents tracking the cloud increases but the number of collisions increases along with the computation time. With 9 agents it is not possible for all of the agents to be tracking the cloud, indicating that the cloud is too busy, as can be inferred from the large number of collisions. The computation times were obtained on a 2.3 GHz Intel Core i7 processor and provides a good comparison between different set ups. It should be noted that computation time may vary on the UAVs processor. The computation time increases with the number of agents but could be decreased significantly if the second and third waves of agents did not run the code until they received a signal to activate them. Increasing the number of agents also improved the spread of agents around the cloud. For these reasons waves consisting of six agents were selected.

Table 5: Evaluation of Swarm Size on Simulation Performance

Number of Agents	Number of Collisions	Time To Locate Cloud (s)	Number of Agents Tracking Cloud	Range of Distances Between Tracking Agents (m)	Computation Time (s)
3	0	360	3	smallest - 246, largest - 712	74
6	1	151.2	6	smallest - 203, largest - 372	137
9	17	115.2	7	smallest - 201, largest - 232	202

In future the number of agents could be dependent on the perimeter of the cloud by using the concentration map that the agents populate. This would prevent collisions by not over populating the cloud and would enable larger numbers of agents to track larger clouds. The simulation could also be improved by having different groups of agents, a searching group and a tracking group both operating at the same time. Swarms provide tolerance to failures but do not completely remove them and so the simulation could be improved by acknowledging collisions and deploying new agents to replace them.

Table 6: Evaluation of Simulation With Noise (Cloud 2, t = 1500s, 6 Agents)

Feature	Number of Collisions	Time to Find Cloud (s)	Time for All Agents to Track Cloud (s)	Range of Distances Between Tracking Agents (m)	Computation Time (s)
Noise In Actuators	12	151.2	1120	smallest - 195, largest - 394	130
Noise In Sensors	10	147.6	740	smallest - 213, largest - 386	135

In real life there will be errors in the actuators and sensors, which should be included to offer a realistic simulation. A 10% error was added to the states and concentration measurement in order to simulate this. Table 6 shows the results of including noise and demonstrates how the guidance strategies performance reduces. The time for all of the agents to be tracking the cloud is reduced due to the error in the concentration resulting in agents switching into tracking mode when they aren't actually close enough to the cloud. The number of collisions increases, which indicates that larger collision avoidance zones would need to be researched before developing a real life system.

Overall the simulation did not meet the requirements for development into a real life system. However, it can locate a pollutant cloud and track the 1 PPM contour, it can spread the agents around the perimeter of the cloud and it minimises the number of collisions by utilising swarm behaviour as well as crude collision avoidance. Unfortunately the three wave strategy aimed to prevent agents running out of battery resulted in increased computation time and introduced problems when the second wave of agents started to deploy.