UNIVERSITY OF WARWICK

UAVSI

UNMANNED AERIAL VEHICLE SWARM INTELLIGENCE

Project Specification

Antonio Brito

1 Abstract

This project explores simulating methods of unmanned aerial vehicles (UAVs) moving, pathfinding and completing tasks in bird-like groups (swarms), initially based off of Boids theory [1].

2 Background

This project combines previous research from several disciplines into one. Looking across these fields, we have **Swarm Intelligence**, which explores a group of agents interacting locally with one another and with their environment[2]. Boids theory is some of the initial work that fits into this field. This project aims at augmenting this with modern swarm intelligence metaheuristics. Aside from this, the aim is to explore the data analysis process from the point of sensing, tracking, state estimation, through to processing and decision making, autonomously. Lastly, we look at managing a UAV system in three dimensions using classical flight mechanics.

In the final report, we will also look back on findings to explore the parallels between the agents' behaviour and expected human behaviour when acting in swarms to evaluate the feasibility of future research on intelligent systems. The overall goal of the project is to combine existing technology in a novel way; in a field of research lacking open source resources.

3 Objectives

The aim of the project can be split down into four distinct sections, grouped by similarity and priority. For all objectives, several implementations will be explored and compared.

3.1 Swarm Operation

The aim here is to simulate a group of identical nodes (UAVs) operating cohesively as a swarm. This objective is further broken down into simulating the physics of the model and the environment as such:

3.1.1 Physics Model

As we are concerned with a 3D environment, the model of the agents will fit such an environment. The agents will be modelled as quadcopters[3], rather than winged aircraft, for simplicity. This is because of the following considerations[1]:

- High maneuverability is key in swarms to ensure high rates of survivability.
- Aircraft position can be fixed, allowing for greater analysis time.
- Vertical take-off and landing (VTOL) is possible.

As a trade-off, the quadcopter model will be slower and will likely be less stable in real-life scenarios. However, we will explore extending the project to looking at conventional winged aircraft flight. Using the Boids model, we will implement the three key rules for emergent swarm behaviour:

- **Node Separation**
  Simulating the ability for nodes to keep some minimum distance from other nodes, to prevent collisions.
- **Node Alignment**
  Simulating nodes tending to have similar velocity vectors to nearby nodes, such that they travel in the same direction.

[1] Quadcopter vs Drone: What's the Difference?

- **Node Cohesion**
  Simulating nodes moving towards the centre of nearby nodes to uphold the ’pack’-like behaviour.

Ideally, after the physics model is created, a working simulation of a group of agents will be demonstrated. The number of agents will be dynamic, as this may later affect optimisations. At this stage of development, the aim will be to have minimal to no collisions between agents with a swarm size of **8-16** drones. The relationship between swarm size and number of collisions will be modelled.

3.1.2 Environment Model

The environment will be continous and three dimensional. The aim will be to have a finite world size, with randomly generated terrain according to a continous heightmap, likely using Perlin noise[4]. Agents will have a start location and a goal location, generated randomly at every iteration of the simulation.

3.2 Pathfinding

To introduce and compare several global path planning algorithms, we will first use a 2D map to model both the **A***[5] and **D* Lite**[6] algorithms for individual node pathfinding. We will then explore a 3D implementation of the more useful algorithm for the project, likely D*, alongside a case study for the algorithm - the Mars Exploration Rovers[7]. Noticeably, a large part of the development timetable in Section 5 is devoted to this 3D implementation. This is due to a lack of resources avaliable for this specific implementation.

We will look at three separate implementations of pathfinding:

- **The Naive, Leaderless Approach**: Does chaos occur without an elected leader, leaving each agent to think for themselves?
- **A Combined Approach for Swarm Operation & Pathfinding**: We will attempt to model swarm pathfinding with collision avoidance and cohesion heuristics, such that the edge weights that bring nodes too close or too far to other nodes are high-cost.
- **D* with Leadership Behaviours**: Does electing a leader reduce computational power requirements at a trade-off for survivability?

For all of these implementations, we will consider the computational power required for simulation, aiming to minimise this so that larger numbers of agents can exist.

3.3 Obstacle Avoidance

Assuming a suitable method of pathfinding is found, the next objective will be to simulate an environment with obstacles and model obstacle avoidance within the swarm. The obstcle-environment relationship is classified by the following characteristsics:

- Obstacles are randomly generated from a set of types (such as Floating Cube, Mountain, Building).
- The size of obstacles will be random, within a Gaussian distribution.

This objective can be broken down into two parts:

3.3.1 Perfect Representation of the Environment

Initially, the goal is to ensure a high survivability rate amongst the swarm when given a perfect representation of the location and classification of obstacles. The project will explore the use of two main avoidance mechanisms:

1. Naive D* Avoidance using edge-weights
2. Simulated Magnetic Fields for Collision Avoidance[8]

3.3.2 Incomplete Representation of the Environment

The focus will then shift onto how the swarm handles receiving incomplete data regarding obstacles, such as position without a type classification, or vice versa. The goal is to explore autonomous behaviour when considering aspects of probability. At this stage, we consider obstacle density as another factor; in less challenging terrain, avoidance is easier.

3.4 Obstacle Detection

An extension to the project will consist of exploring methods of obstacle detection. This objective, as well as the next objective, begins to look at a simulated implementation of state estimation & tracking models. The inital steps for this objective will be looking at methods of sensing, primarily LiDAR[9], to detect obstacle positions.

Conequently, we will explore how sensor placement within the swarm affects survivability and whether an increased number of sensors leads to better results (using data fusion techniques).

3.5 Hostile & Moving Obstacles

A final possible extension will be looking at the detection and avoidance of hostile and moving obstacles. To complete this, hostile obstacles are obstacles with the capability of firing projectiles or signal jamming across some unknown (random) radius. Moving obstacles traverse the terrain linearly in a given (random) direction. This could be further extended to implement random non-linear walks.

The aim of this objective is to explore autonomous behaviour in environments where predictions need to occur when considering danger travelling towards the swarm.

4 Methods

The proejct will be completed in Unity due to the lack of licensing requirements, ease of use and scripting support[10], likely using C++. Consequently, the simulation will be a Unity3D environment. The codebase in its entirety, including source code and version control, will be managed by a cloud-based GitHub repository[11].

The **waterfall** software methodology will be used in development. This is evident in the timetable in Section 5. The waterfall approach lends itself to structured stages of development, where one stage leads naturally into the next. For the purposes of this project, these stages are outlined in bold in the timetable in Section 5.

For this approach, the model splits the development process into the following sections, specific to this project:

1. **System** - the project specification outlining the tasks to be completed with rough timelines.
2. **Analysis** - a section in the final report detailing the models used and techniques compared.
3. **Coding** - the development of the system.
4. **Testing** - validating the system to ensure the project's objectives have been met.

5 Timetable

Generally, the aim is to split the objectives above to be completed according to the timetable below:

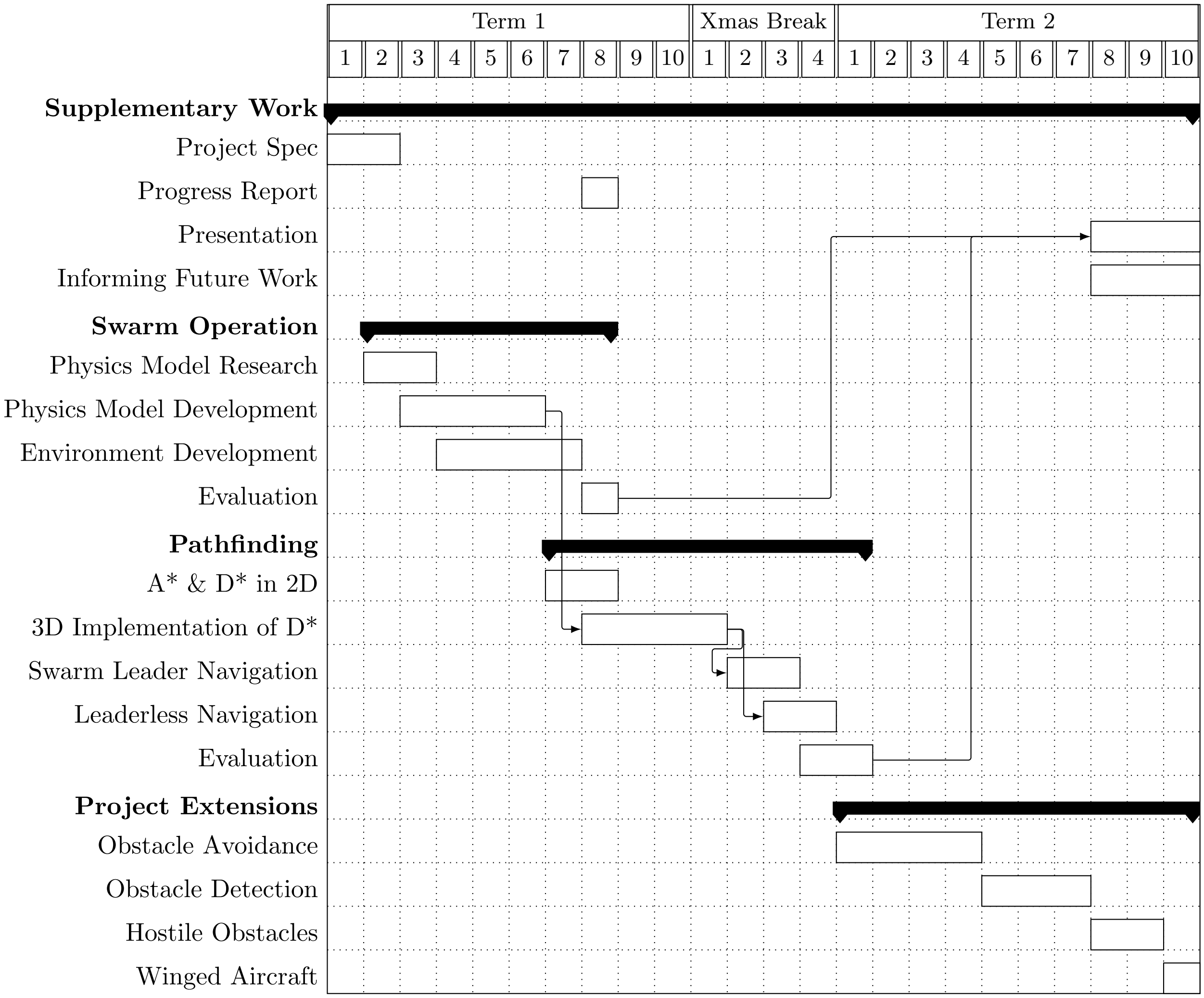

Figure 1: Gantt Chart showing main objectives. Numbers shown are week numbers.

Noticeably, the timetable leaves a large chunk of time between the scheduled completion of essential objectives (Term 2 Week 1) and the presentation (Term 2 Week 9). This is to leave a margin of error induced by risks and unexpected events, as discussed in the next section.

6 Risks & Issues

The major risks to the project are as follows:

1. **Stages in the project fall behind schedule.**
   This may occur when a critical task takes too long to complete. The solution to this issue is to ensure alternatives exist at all critical points and the timetable lends itself to having leeway.

2. **Insufficient computational power.**
   Naturally, simulation lends itself to requiring high computing specifications. A side aim of the project is optimising the simulation environment to ensure it is able to be ran on my personal computer. If all else fails, a wise option is to consider the Department's access to computational resources.

3. **Project work is lost and/or unretrievable.**
   As all development will occur on a personal computer, it is essential to ensure backup copies exist elsewhere. Hence, all written reports, specifications and source code will be backed up to GitHub as a minimum.

4. **Insufficient findings for a complete report.**
   There is often more to write about regarding failures than successes, as all other reasonable options must be exhausted before declaring failure! No useful product is guaranteed, but findings certainly are.

5. **Libraries or resources become unavaliable during the project process.**
   Whilst this is low risk, to mitigate this, local copies will be used wherever possible and backed up alongside the codebase.

6.1 Ethical Issues

A research project involving autonomous technology comes with vital ethical considerations. All research will be conducted in line with the Department's ethical guidelines[2], as no human imvolvement is required.

It is important to ensure there is no ethical ambiguity for the purposes of the project. As such, the intended research goals as detailed in the problem statement will be kept to those without ethical ambiguity or favouring. The aim through this project is to inform future research for using intelligent autonomous systems defensively.

[2] Warwick DCS Ethical Guidelines

References

[1] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, 21(4):25–34, aug 1987.

[2] Junyan Hu, Ali Emre Turgut, Tomáš Krajník, Barry Lennox, and Farshad Arvin. Occlusion-based coordination protocol design for autonomous robotic shepherding tasks. *IEEE Transactions on Cognitive and Developmental Systems*, 14(1):126–135, 2022.

[3] Mohd Shahiq Khan. Quadcopter flight dynamics. *International Journal of Scientific & Technology Research*, 3:130–135, 2014.

[4] Ken Perlin. An image synthesizer. *SIGGRAPH Comput. Graph.*, 19(3):287–296, jul 1985.

[5] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[6] Sven Koenig and Maxim Likhachev. D*lite. pages 476–483, 01 2002.

[7] Joseph Carsten, Arturo Rankin, Dave Ferguson, and Anthony Stentz. Global path planning on board the mars exploration rovers. In *2007 IEEE Aerospace Conference*, pages 1–11, 2007.

[8] Fang Wu, Akash Vibhute, Gim Song Soh, Kristin L. Wood, and Shaohui Foong. A compact magnetic field-based obstacle detection and avoidance system for miniature spherical robots. *Sensors*, 17(6):1231, May 2017.

[9] Arne Devos, Emad Ebeid, and Poramate Manoonpong. Development of autonomous drones for adaptive obstacle avoidance in real world environments. In *2018 21st Euromicro Conference on Digital System Design (DSD)*, pages 707–710, 2018.

[10] Ismail Buyuksalih, Serdar Bayburt, G. Buyuksalih, A. Baskaraca, Hairi Karim, and Alias Rahman. 3d modelling and visualization based on the unity game engine - advantages and challenges. volume IV-4/W4, pages 161–166, 11 2017.

[11] John Blischak, Emily Davenport, and Greg Wilson. A quick introduction to version control with git and github. *PLoS computational biology*, 12:e1004668, 01 2016.