UNSYS digital

The Journal of Instrumentation, Automation and Systems

Aircraft Swarm Intelligence: An Approach to Coordinated Swarming with Artificial Potential Functions & Gradient Descent

Justin T. Ruscoe and Tiauw Hiong Go

Florida Institute of Technology, Melbourne, FL 32901, USA

Abstract—This study presents an organized method of swarm coordination with the use of artificial potential functions (APFs) utilizing a first-order optimization gradient descent algorithm. With the emergence of an increasing need for Unmanned Aerial Vehicles (UAVs) system control, swarm coordination presents an approach to eliminate collisions and effectively achieve mission goal parameters.

The gradient descent algorithm begins with an initial configuration and implements a step, or iteration, in a direction that is opposite to the gradient. The APFs contain both repulsive and attractive potential functions that contribute to the gradient ultimately determining the states of the agent with respect to the distance from other agents and obstacles. Obstacles or other agents projected to be too close within the path of an individual agent affect the agent's path and dynamics.

Experimental simulations consisted of three, five, and ten agents with two obstacles arranged at different initial positions. Agents' dynamics were constrained to match the Boeing AH-6 Unmanned Little Bird (ULB). Simulations had shown each agent to effectively travel to a prescribed target location while avoiding obstacles and other agents simultaneously.

Keywords—Swarm, swarm intelligence, artificial potential functions, gradient descent.

I. Introduction

Swarm intelligence can be defined as the coordinated behavior of individuals in a single system to solve a problem that a single individual could not achieve at the same level of success. These individuals are referred to as agents and the coordinated behavior is known as swarming.

Biological inspirations have been a critical factor to leading developments in engineering technologies. One of the more notable examples of biologically inspired technology that has shaped history through the ages is flight. Earlier designs for manned flight were based on winged animals such as bats and birds, a complex anatomy to be later solved through the centuries with more simplistic designs.

Another observation of birds, ants, and even bacteria is their ability to move as one organized body. This collective behavior seen as a multi-agent system, where the agents operate to achieve a given task is referred to as swarming. The characteristics that define an effective swarm are the agent's ability to communicate and react. Information relayed between one more agents provides necessary instructions for agents to appropriately react given the situation.

Applications for multi-agent systems with swarming behavior have been of interest to researchers and engineers in an effort to achieve tasks that may not be possible with a single agent or rather optimize efforts with more agents. Area mapping, search and rescue, structure development, tracking, and many other tasks have been the highlight of multi-agent systems with swarming behaviors. When the ability to use a multi-agent system exists, benefits include increased flexibility and decreased likelihood to fail. Being a multi-agent system, failure of one agent may decrease the effectiveness of the system but the task will still be completed rather than fail if a single agent system was deployed.

The modelling of swarm intelligence roots from the observations of biological swarming behavior. This behavior was first observed and modelled in the fields of social psychology and artificial intelligence where the foundation set forth for swarming intelligence studies stems from the particle swarm optimization (PSO) and evolutionary computation. Kennedy and Eberhart are attributed to the emergence of the PSO [1] describing it as an approach to optimizing a set of nonlinear functions utilizing the particle swarm methodology. Soon after the release of the first PSO algorithm by Kennedy and Eberhart, the improvements to the original algorithm are released by Eberhart and Shi [2]. At this point, computational improvements [3] and advancements [4] to the algorithm were developed.

Expansion to other methods of swarming had broadened the mathematical concepts utilized to mimic the behavior. One of these mathematical models is known as artificial potential functions (APFs). An example of this type of swarm control most relevant to this study was from Vries and Subbarao who formulated a cooperative control of UAVs constructed as a potential based swarm [5]. Computer simulations employed for this study made use of multi-rotor vehicles. These multi-rotor vehicles exhibit a simplified dynamics that are preferable

Corresponding author: Tiauw Hiong Go (e-mail: tgo@fit.edu).
This paper was submitted on September 23, 2015; revised on November 11, 2015; and accepted on December 4, 2015.

compared to fixed-wing UAVs that are dynamically more complex systems.

Simple tasks such as lifting a box to move to a tabletop can be deemed a simple mission with a simple goal. If the box now has an increased weight that the ordinary person cannot carry then this task would be more difficult. If multiple individuals worked together with the common goal of placing the box atop the table then the task could be completed with far less effort. This concept of cooperative mission behavior is a desirable quality to multi-robotic or multi-vehicular performance.

The most relevant application to this document's approach to coordinated swarming would be aerial applications. More difficult in nature, swarming behavior in three-dimensions is noticeably more complex than swarms in two-dimensions or on-the-ground applications. Birds of a flock are typically agile and are able to maneuver quickly in reaction to the neighboring birds. Modern aviation has progressed significantly enough to where pilots are able to control the aircraft to a highly precise degree as well as increased maneuverability control. This benefit enables aerial formations and coordinated navigation. Swarming applications with pilot control can be deemed more dangerous, but removing the pilot from the aircraft could remove the human factor from the system. Fleets of piloted aircraft are aware of the movements of their neighbors through human sight. Increasing the amount of aircraft in a given fleet could cause chaos in the event of a single malfunction or unpredicted movement. Causes of disaster could be due to the limited reaction time of a pilot and reliance on sight for nearby aircraft.

Eliminating the human factor from the skies, the Unmanned Aerial Vehicle (UAV) was created. While currently most UAVs are not fully autonomous, such control does exist and poses a valuable asset in the near future. Placing autonomous, coordinated, swarm control on a fleet of highly maneuverable UAVs presents a method of collision avoidance and synchronized actions to complete a certain goal with little risk. Most applicable to this type of control would be military application. Take for example a search and rescue in a hostile battlefield, where a single aircraft is subject to ground fire, projectiles, and other enemy aircraft. With the aid of additional aircraft, fire power and search support this would significantly increase mission effectiveness. As simple as activating a swarm control, this could connect an entire fleet of aircraft for precise, systematic area mapping while maintaining obstacle avoidance from known and unknown risks, given avionics effectiveness. The desirable unmanned aircraft in this type of system control would be a vertical takeoff and landing (VTOL) rotorcraft due to the increased maneuverability in a crowded airspace.

The primary goal of this study is to develop an approach of coordinated swarming to be applied to VTOL UAVs using APFs for navigation and the gradient descent for implementation of the swarm model.

All vehicles are subject to the limitations of their motion, thus the swarm model explores the extent of such dynamics restrictions to a particular unmanned helicopter known as the Unmanned Little Bird (ULB). Additionally, collision avoidance is another primary goal of all swarm simulations to prevent system failure.

Simulations of a centralized and decentralized version of the swarm model are evaluated to assess the effectiveness of both types of swarming as well as to demonstrate the value of APFs in such application.

Beginning with Section I, the inspirations and past works involving swarm intelligence have been discussed. Moreover, the purpose and objectives of this study are also outlined to provide the reader with an understanding of the primary goals of the presented swarm model.

Section II contains the outline of the swarming model. This section explains the components of the model and how it was developed. Simulation environment setup is also detailed as well as the different types of simulations that were performed.

The agent dynamics constraints are discussed and explained in Section III. This portion reviews the incorporation of aircraft to the swarm model.

The results section, Section IV, explores the simulation results of the different model types with differently sized agent swarms. Analysis and measure of success is determined based on the criteria presented.

Concluding with Section V, the conclusions drawn and future work possibilities are outlined.

II. Swarm Model Design & Control

The theory and implementation behind the concepts and formation of the swarm model begins with the foundation in which it was inspired by Chiew, Zhao and Go in *Swarming Coordination with Robust Control Lyapunov Function Approach* [6]. In this study, a multi-agent system modeled to UAVs utilizes Robust Control Lyapunov Function to control each agent to attain the desired heading consensus and separation distance. The simulations made use of an artificial potential field to demonstrate the viability, to operate within the rules presented by Reynolds' boids model.

Following the inspiration behind the ideals for this model, is the approach to the model itself. For this document, as the primary objective states, the goal is to contribute to the artificial potential function swarming techniques by developing an original method to swarming coordination with artificial potential functions making use of the gradient descent algorithm for implementation of the approach. A well-known approach to using APFs to navigate through space with gradient descent is a robot navigation technique presented by Howie Choset in *Principles of Robot Motion: Theory, Algorithms, and Implementation* [7]. The theory presented by Choset is followed most closely in development of the presented swarm model.

By making use of the APFs described by Choset, a coordinated multi-agent three-dimensional, navigating swarm was created. For simplicity of the model, the agents are point

mass systems. In the model, the attractive potential navigates the agents to the target location while the repulsive potential repels agents from each other and obstacles within a given distance. This defined distance is the bordering location to whether the repulsive potential is seen or ignored. Each agent traveling through space is only aware of the locations of other agents when in the neighborhood of other agents, thus the system is decentralized. On the other hand, when introducing an aggregation attractive potential, all agents of the swarm are always aware of each other's locations and are said to be a centralized system. All agents in all situations are always aware of the target location.

A. Artificial Potential Functions

The attractive potential, $U_{att,}$ presented by Choset mimics the way realistic vehicles will approach an objective. In the case of the objective or target location at a far distance, the agent will start at a larger speed to minimize the distance. When the agent is approaching the target location, q_{goal}, it will slow down at a certain distance. Therefore, a potential that quadratically grows with distance is selected to match this effect. The potential is formulated as:

$$U_{att}(q) = \frac{1}{2}\zeta d^2(q, q_{goal}) \tag{1}$$

where the current location or position of the agent is denoted simply as q and the term $d^2(q, q_{goal})$ represents the square of distance between the target location, q_{goal}, and the current location, q. In order to control the magnitude of attraction potential, an attraction scaling parameter, or attractive gain, ζ, is introduced.

An example of this gradient effect is seen in **Figure 1** [7]. In (a), the location of the goal is placed at the center. The gradient vectors surrounding the goal location represent the different points nearby the goal. To visually aid in the physical representation of the potential, (b) displays a contour plot with individual circles to exhibit where $U_{att}(q)$ is constant. A surface plot of the attractive potential is shown in (c).

Attractive potential for this model is broken into a piecewise function such that as the agent approaches the threshold distance from the target, d^*_{goal}, the attractive potential switches to from the quadratic to conic potentials. The definition is as follows:

$$U_{att}(q) = \begin{cases} \frac{1}{2}\zeta d^2(q, q_{goal}), & d(q, q_{goal}) \le d^*_{goal} \\ d^*_{goal}\zeta d(q, q_{goal}) - \frac{1}{2}\zeta (d^*_{goal})^2, & d(q, q_{goal}) > d^*_{goal} \end{cases} \tag{2}$$

As the gradient is what is of most concern, $\nabla U_{att}(q)$ is also defined as follows:

$$\nabla U_{att}(q) = \begin{cases} \zeta(q - q_{goal}), & d(q, q_{goal}) \le d^*_{goal} \\ \frac{d^*_{goal}\zeta(q - q_{goal})}{d(q, q_{goal})}, & d(q, q_{goal}) > d^*_{goal} \end{cases} \tag{3}$$

$$\nabla U_{att}(q) = \begin{cases} \zeta(q - q_{goal}), & d(q, q_{goal}) \le d^*_{goal} \\ \frac{d^*_{goal}\zeta(q - q_{goal})}{d(q, q_{goal})} + \frac{d^*_{goal}\nu(q - s_{center})}{d(q, s_{center})}, & d(q, q_{goal}) > d^*_{goal} \end{cases} \tag{4}$$

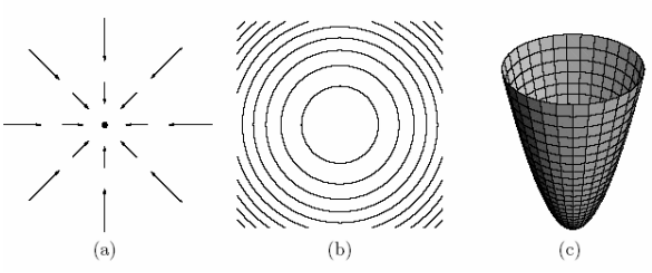

Figure 1 Attractive potential [7]

Just as ordinarily used with robot navigation, the attractive potential is placed at the target location to direct the agents to the desired direction. One difference applied to the centralized case is the aggregation attractive potential that affects the swarm prior to entering the threshold distance to the goal.

Note that when $d(q, q_{goal}) = d^*_{goal}$, for (3) the transition between quadratic and conic sections is continuous because the values will be equivalent. Rather, in (4) there is a discontinuity about $d(q, q_{goal}) = d^*_{goal}$ which will inhibit an oscillatory or "jumping" behavior. To mitigate this, the aggregate potential gain, ν, must be sufficiently small enough to avoid large jumps crossing the threshold distance to the target. Aggregate potential gain, ν, can be solved iteratively or selected on an *ad hoc* basis.

Repulsive potential enables the swarm model to contain separation characteristics similar to the way Choset's robot navigation model avoids obstacles. Essentially, the swarm agents of the model are considered obstacles by placing a repulsive potential at their location throughout the path they travel.

As with the attractive potential, the strength of repulsion must be relative to the distance from the obstacle or other agents. The closer an agent is to an obstacle or another agent, the force opposing its motion should increase similar to how the repulsive behavior of a magnet works.

Again, a piecewise function represents the repulsive potential:

$$\nabla U_{rep}(q) = \begin{cases} \eta\left(\frac{1}{Q^*} - \frac{1}{D(q)}\right)\frac{1}{D^2(q)}\nabla D(q), & D(q) \le Q^* \\ 0, & D(q) > Q^* \end{cases} \tag{5}$$

where η, is the gain on the repulsive gradient and Q^* is the threshold distance to the obstacle. $\nabla D(q)$, is defined as:

$$\nabla D(q) = \frac{q - c}{d(q, c)} \tag{6}$$

A surface plot of the repulsive potential can be seen in **Figure 2** where Q^* shows the boundary between where the obstacle or agent is seen or ignored.

It is possible for repulsive potentials to overlap creating an even larger potential than if the individual repulsive potentials were further away from each other. This affect is desirable as this increase in potential implies an even greater threat to the approaching agent in motion.

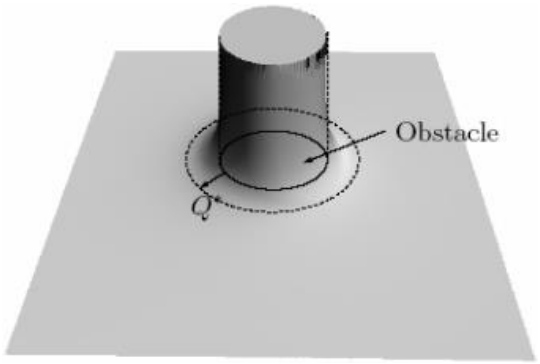

Figure 2 Repulsive potential [9]

In order to effectively use both the repulsive and attractive potentials, the total potential is determined by the sum of all potential in the system. This total potential, *U(q)*, is as follows:

$$U(q) = U_{att}(q) + U_{rep}(q) \quad (7)$$

where the gradient sum is represented simply as:

$$\nabla U(q) = \nabla U_{att}(q) + \nabla U_{rep}(q) \quad (8)$$

The swarm model presented for both the centralized and decentralized cases only contain repulsive and attractive potentials. These potential sums are assigned to each agent rather than the system as a whole for more individual effects.

B. Gradient Descent & Model Types

The three-dimensional swarm model presented contains a core implementation of the APFs and gradient descent while exhibiting two behavioral methods referred to as centralized and decentralized.

A swarm is said to be decentralized if the agents operate through local intelligence and neighboring agents. One key factor to being decentralized is that the agents do not work under a single commanding guidance. Typically this means the agents will only communicate when in close contact with each other. A centralized swarm is one that is controlled overall by a higher commanding guidance. Additionally, the agents are self-aware of each agent location at all points in time.

In the case of the decentralized swarm, all agents are not aware of their locations nor are they aware of any obstacles in the path. One method to navigate through the APFs is by gradient descent. To implement the gradient descent algorithm, there must exist a gradient to travel "downhill" from, thus all agents are aware of a target location and nothing else. Due to being aware of the target location from the beginning but unaware of the other agents' locations and obstacles, this model can be more appropriately considered mission-centralized, swarm-decentralized. But because the goal of the model is not to reach a target location but to achieve a level of swarming behavior before arrival, the swarming model itself is termed decentralized.

The centralized swam model also allows the agents to be aware of the target travel location. Additionally, each agent is always aware of the desired convergence location as the swarm progresses in time. This difference lies within the attractive potential between the two cases. Both models utilize the attractive potential that navigates the agents to the target location; whereas, the centralized model also implements an aggregate attractive potential in addition to the navigation attractive potential. In theory, the decentralized approach should be able to achieve a level of swarming without this aggregate attractive potential as the agents will eventually converge to the target path.

Implementation and creation of the decentralized model is through scripting software of MATLAB [8].

Gradient descent requires the initial start point of each agent $q_k(1) = [x_{initial}, y_{initial}, z_{initial}]$, where *k* refers to agent *k* for $k = 1, 2, 3 \ldots n$. The selected initial locations should be within the nearby vicinity of all agents in the system but not too close such that the agents are within the threshold distance of an obstacle or other agent, Q^*. The proposed swarm model exists in three dimensions because two-dimensional aerial swarming applications are unlikely due to the three-dimensional kinematic characteristics of aircraft.

Next, parameter initialization is performed. **TABLE I** provides the variables and their definitions that need to be initialized prior to beginning the swarm progression. The gains η, ζ, and ν are scaling parameters that control the size of the attractive and repulsive forces. These gains should be selected to appropriately match the size of the environment in which the agents will travel. Similarly, the threshold distances Q^* and d^*_{goal} should also match the environment size. The size of an obstacle and operating agent should be accounted for when assigning the threshold distances.

TABLE I PARAMETER VARIABLE DEFINITIONS

Parameter	Definition
η	Repulsive potential gain
d^*_{goal}	Threshold distance to goal
ζ	Attractive potential gain
Q^*	Threshold distance to obstacle
ν	Aggregate attractive potential gain
α	Iteration step size
i	Step counter

Step size, α, is applied to the gradient descent when the new location is determined. It is important to choose a step size small enough to avoid "jumping" into an obstacle by applying a potential force too large based on the environment size. Equation (9) represents how this parameter is used.

$$q_k(i+1) = q_k(i) + \alpha \nabla U\big(q_k(i)\big) \quad (9)$$

If desired, obstacle placements require initial conditions as well. $obs_l(1) = [x_{initial}, y_{initial}, z_{initial}]$, where *k* refers to

obstacle l for l = 1, 2, 3 … m. In this model, the obstacles are treated as spherical to best encase all shapes and sizes without defining a complex function to represent the actual shape. For this reason, radii of the obstacles, r_l, as well as the agents, r_k, must be defined.

The last location to be defined is the target location. This is the location that the agents of the swarm will choose to travel towards. Target location, $q_{goal} = [x_{goal}, y_{goal}, z_{goal}]$, should be placed sufficiently far enough from the initial locations to allow sufficient time for agents to capture full behavior. Again, the threshold distances need to be accounted for to allow time for the agents to travel outside and within threshold of an obstacle.

Lastly, the initial total gradient of potential, $\nabla U_k(q_k = 1)$ for each agent can be calculated. Note that the initial potential for the centralized and decentralized cases differ and calculation is discussed in the next section.

With the initial conditions defined, the loop statement in the gradient descent method can be executed. To begin, the loop statement must be defined. In the algorithm provided the statement "**do** $\nabla U_k(q_k(i)) \neq 0$ **while**," would be sufficient in an ideal situation yet it is highly unlikely the gradient potential will ever actually reach zero, therefore a tolerance condition, ϵ, can be introduced to solve this issue. By assigning a value small enough, the previous statement can be rewritten as "**do** $\nabla U_k(q_k(i)) < \epsilon$ **while**." Using this condition presents an issue to the swarming model considering there exists more than one agent trying to achieve this condition. Since the condition $\nabla U_k(q_k(i)) = 0$ is true at the target location, it is likely that agents will reach this condition prior to others. This causes oscillatory movement and unrest to the agents that arrive at the target location before the others. An approach to mitigate this is to terminate the gradient descent when one agent arrives rather than all. The downfall to altering the condition to a single agent is the fact that all agents may not have converged as close to the target location as desired. Because the goal of this study is not for all the agents to reach the target location but to achieve a swarming behavior, the modified condition for a single agent will suffice so as long as the full extent of the behavior in the movement can be captured prior to termination. If desired, a simple modification of the attractive potential function can be utilized:

$$\nabla U_{att}(q) = \begin{cases} \zeta(q - q_{goal}), & d_{stop} < d(q, q_{goal}) \leq d^*_{goal} \\ \dfrac{d^*_{goal}\zeta(q - q_{goal})}{d(q, q_{goal})}, & d(q, q_{goal}) > d^*_{goal} \\ 0, & d(q, q_{goal}) \leq d_{stop} \end{cases} \quad (10)$$

This function will act similar to the attractive gradient potential in (3), with the exception of the new threshold distance introduced, d_{stop}. This stopping distance should be smaller than the goal threshold distance to allow for sufficient slowing behavior during $d_{stop} < d(q, q_{goal}) \leq d^*_{goal}$ before stopping when $d(q, q_{goal}) \leq d_{stop}$ is met.

After looping through the gradient descent method and reaching the desired condition, a three-dimensional position plot can be used to represent the path the agents take.

III. Agent Dynamics Constraints

In order to be presented as a proof of concept for systematic aircraft control, restrictions on the kinematics of the agents must be presented. The model presented allows for altering of the kinematics of each agent. By trial and error the model can be matched to that of an actual UAV. In this document, the aircraft used is the Boeing AH-6 whose unmanned variant of the model is referred to as the Unmanned Little Bird (ULB). In order to capture the kinematics of the ULB the constraints of the aircraft are presented.

Path planning is a well-known issue in emerging UAV autonomous systems. Tsourdos, White, and Shanmugavel explain these very constraints for UAV path planning in *Cooperative Path Planning of Unmanned Aerial Vehicles* [9]. Defiance of the constraints to a UAV could cause mission failure or even catastrophic consequences. First and foremost are the obstacles within the path of the UAV. In this model, the repulsive potential is attached to known obstacles and obstacles that become within range. The flyable paths must meet the dynamics requirements of the aircraft such as velocity, accelerations, rate of climb, and maximum speeds. Maneuverability constraints also exist such as minimum turn radius for fixed wing aircraft. In comparison to fixed wing aircraft such constraints are much more lenient on rotorcraft. Turn radius is more of a constraint for fixed wing aircraft. For rotorcraft turning is subject to the constraints of the helicopter structure and the pilot at a given speed.

According to the *Helicopter Flight Training Manual* [10], for a given airspeed, the greater the angle of bank causes the greater the rate of turn, the smaller the radius of turn, and the greater the load factor. Additionally, with an increased airspeed at a given angle of bank this causes a lower rate of turn and a larger radius of turn.

For a helicopter to achieve the smallest radius of turn, the lowest bank angle for a given airspeed should be obtained. Structural limitations will prevent the helicopter from undergoing harsh bank angles from increased speeds as "the load factor for any helicopter is 2 G regardless of its airspeed" [11]. A 2 G load equates to approximately a 60 degree bank angle as seen in **Figure 3** where 0 to 30 degrees produces the safest load factor for the helicopter.

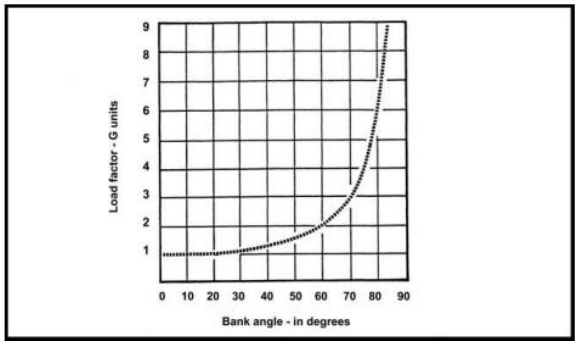

Figure 3 Load factors produced by a given angle of bank

A quick and relatively accurate way of estimating the bank angle is by using the following formulation, assuming that the airspeed is in miles per hour:

$$Angle\ of\ Bank \cong \frac{TAS(\text{mph})}{10} + 5 \quad (11)$$

In 2004, the ULB made its first autonomous flight for demonstrative purposes. The ULB is a Boeing AH-6 based of the MD 500 family of rotorcraft. Some characteristics that specify the performance of the ULB model is as follows in **TABLE II**.

TABLE II – PERFORMANCE SPECS

Parameter	Maximum Value at Sea Level
Maximum Cruise Speed	$v_{cruise_{max}} = 154$ mph [68 m/s]
Maximum Range	$R_{max} = 237$ miles [381 km]
Maximum Rate of Climb	$ROC_{max} = 2069$ fpm [10 m/s]
Height	$h = 8$ft 9in [2.48 m]
Rotor Diameter	$d_{rotor} = 27$ ft 4in [8.33 m]

Through the basic constraints reviewed, the swarm model can have agents made to match the dynamics restraints of the ULB. It should be noted that the overall motion of a helicopter is relatively complex, thus these constraints assume a simple helicopter with the full ability to move at any given time in any given direction. These constraints are provided as a means of limitation to dynamics of the ULB but do not fully embody the actual motion thus it is assumed any difference from actual dynamics is small or negligible.

IV. EXPERIMENTAL SIMULATION RESULTS

Within this section, the centralized and decentralized swarm models are simulated for 3, 5 and 10 agents. The simulation starts at randomly selected initial positions traveling to a target location with obstacles in the traveling path. To best exemplify the differences in the centralized and decentralized model, the agents' initial locations remain unchanged throughout. Parameter changes are allocated to sustain desired swarming characteristics as well as maintaining the dynamics constraints set forth per agent.

The criteria that define swarming behavior are used here to best assess the performance and success of the swarm simulations. In these swarm models the following will be assessed:

i) Collision Avoidance
ii) Speed Agreement
iii) Heading Agreement
iv) Separation Agreement

Though boids rules are not explicitly stated as a factor to evaluation, the above assessment list inherently guarantees that boids rules (i.e. separation, cohesion, alignment) are satisfied.

Collision avoidance is an essential factor to the primary objective of this document thus any collisions deem the simulation failure. Here, collisions are detected in the MATLAB script (similar to the dynamics restrictions) based on two given cases.

Case 1: Rotor blade alignment $d_{ij} \geq$ 9m for $q_i\hat{\imath} = q_j\hat{\imath}$

Case 2: Above/below rotor blades $d_{ij} \geq$ 5m for $q_i\hat{\imath} \neq q_j\hat{\imath}$

Since this simulation does not account for turbulent effects of the rotor blades, the agents can be as close as possible to each other without any effect thus it is assumed no turbulent effects exist in the simulation. This however, is a highly oversimplified assumption of helicopter flight but for the purposes of proving the cohesion and separation characteristics of the model, the agents are allowed to freely move within the defined constraints.

The remaining assessment criteria are based on distributed agreement that says that the agents' states, $x_i(t)$, will converge to a consensus state, x_c, in time. For speed, heading, and separation the following consensus states can be evaluated:

Speed agreement (Reconnaissance speed) $\lim_{t\to\infty} v_i(t) = v_c = 30$ m/s

Heading agreement $\lim_{t\to\infty} \|\theta_i(t) - \theta_j(t)\| = \theta_c \leq 5°$

Separation agreement $\lim_{t\to\infty} \|d_i(t) - d_j(t)\| = d_{ij} \leq 30$ m

The following plots exemplify the 10 agent case for both centralized and decentralized models.

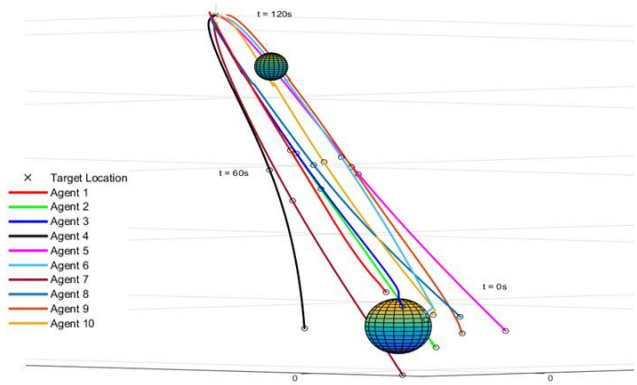

Figure 4 Sample 10-Agent centralized swarm path

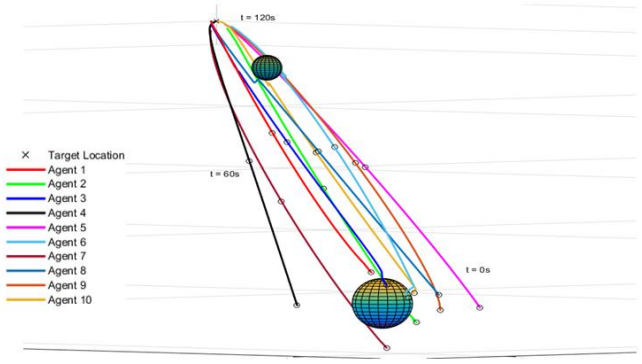

Figure 5 Sample 10-Agent decentralized swarm path

Initial parameters selected for the simulations were based on numerous factors when deciding the best possible combination in trial and error. To travel at 30 m/s the attractive gain was set to 150. To obtain a 30 m/s speed the step size, α, and the time per iteration must be set to 0.001 and 0.05s respectively. Using

a step size larger would cause the agents to jump into the obstacles or other agents whereas using a step size smaller would cause the agents to travel much slower within the environment created. Both produced undesirable behavior leading to the final chosen step size of 0.001. In the centralized model, the aggregate attractive gain was selected to be 1.5. This value was selected off of the oscillatory observations the agents exhibited when increasing this value.

The threshold distance of the goal location was set to a safe 10 meters whereas the threshold to the obstacles was set to 200 meters. Setting the threshold to the obstacle large was based on path planning of the agents. If a smaller threshold was chosen, all agents would travel in the straightest path possible which was relatively unrealistic as compared to actual motion of swarms. By providing a threshold in which all agents were always in, a more desirable effect was found coupled with a repulsive gain of 1000.

TABLE III ERROR RESULTS

Agent Amount	3		5		10	
Model Type	C*	D**	C	D	C	D
Average Heading Error (t > 20s)	0.30°	0.30°	0.50°	0.63°	0.40°	0.53°
Average Relative Velocity Error	0.11 m/s	0.11 m/s	0.13 m/s	0.13 m/s	0.09 m/s	0.06 m/s
Average Relative Distance Error	1.99 m	0.65 m	8.02 m	2.27 m	4.39 m	0.53 m

*C = Centralized **D = Decentralized

TABLE III contains all the error in the average relative heading, speed, and distance between other agents. Relative speed error shows the most minimal error with a deviation maximum of all simulation cases as 0.11 m/s. Average relative speed error in comparison between the centralized and decentralized cases show no difference except for the 10 agent simulation.

The average heading error, similar to average relative distance error, shows slight difference between the centralized and decentralized cases for t > 20s. These values would be much larger if incorporation of the initial heading changes in the first 20 seconds of the simulations were included. Allowing the agents to attain swarming after 20 seconds shows an error of no less than 10 times smaller than if the initial heading error was included. For all cases, the heading remains well within the desired constraints.

Relative distance error shows that for the decentralized case, the error is less than the centralized case for all agent number simulations. This implies increased movement in the centralized case whereas the decentralized case tends to be less prone to smaller changes of movement.

Evaluation criteria previously described was based off of collision avoidance, speed agreement, heading agreement, and separation agreement. Each criterion is examined for all model simulations.

A. Gradient Descent & Model Types

A collision was detected if the distance between the agents was less than the specified criteria for two cases:

Case 1: Rotor blade alignment $d_{ij} \geq 9\text{m}$ for $q_i\hat{\imath} = q_j\hat{\imath}$

Case 2: Above/below rotor blades $d_{ij} \geq 5\text{m}$ for $q_i\hat{\imath} \neq q_j\hat{\imath}$

In all simulations, the minimal distance was $d_{ij} = 6.5\text{m}$ for $q_i\hat{\imath} \neq q_j\hat{\imath}$ and $d_{ij} = 11\text{m}$ for $q_i\hat{\imath} = q_j\hat{\imath}$. Thus in no situation during simulations were the agents ever subject to possible collisions. Again, no turbulent affects were accounted for in the simulations thus it may not be entirely possible for the ULB to fly at such close quarters.

With the chosen parameters for the simulations, the agents react smoothly to maneuvering around the obstacles in the given path to the target yet in some situations a slight jagged trajectory effect is seen. This effect is responsible for the agent velocity decrease and increase spikes as the agent quickly attempts to slow down to avoid the collision and quickly speed back up to the swarm speed. While these spikes in movement are not ideal, they fit within realistic accelerations of actual aircraft.

B. Speed Agreement

Speed agreement: (Reconnaissance speed) $\lim_{t\to\infty} v_i(t) = v_c = 30\ \text{m/s}$

Here, the speed selected was 30 m/s to mimic a slow reconnaissance speed of the ULB. In all simulation cases, the speed agreement always stayed within 30 m/s within a small range in deviations.

Even though all agents had the same attractive gradient assigned, the small consistent deviations in speed are likely due to the differing repulsive effects at different locations in the simulation environment.

Bank angle was calculated based on the airspeed of the agents which is the reason for the similarity in the agent velocity and bank angle plots. Here the bank angle plots show that during any turn the agent, or ULB, was physically capable of the maneuver. This applies to all simulations in the centralized and decentralized cases.

C. Heading Agreement

Heading agreement: $\lim_{t\to\infty} \|\theta_i(t) - \theta_j(t)\| = \theta_c \leq 5°$

Within the first 15 seconds of all simulations, the relative heading was below 10 degrees. After 25 to 30 seconds, the relative heading was below 5 degrees. In both swarm types of centralized and decentralized swarming the relative heading appeared similar. In all differently numbered agent cases the centralized swarming converged to 5 degrees faster than the decentralized cases yet relatively unnoticeable.

D. Separation Agreement

Separation agreement: $\lim_{t\to\infty} \|d_i(t) - d_j(t)\| = d_{ij} \leq 30\ \text{m}$

Separation distances between the centralized and decentralized cases grow different as $t \to \infty$. As seen in the overall path figures, the paths exhibit different types of curvature. In the centralized case all agent simulations always

converge closer together and appear more unified at $t = 60$ s. In the decentralized cases, the agents appear less unified and are further separated than their centralized counterpart. In most cases, the average distance is always less than 30 m or is decreasing to a smaller value until reaching the target location at $t = 120$ s.

V. Conclusion & Future Work

The purpose to this document was to develop an approach to coordinated swarming to be applied to VTOL UAVs using APFs for navigation and the gradient descent for implementation. This swarming application was proven for 3, 5 and 10 agent swarms in both centralized and decentralized swarming simulations.

From the resulting simulations, both centralized and decentralized swarming demonstrated a unique behavior for swarming. Neither model type is better than the other but offer different properties within that may be better fit for different applications.

Given the dynamic limitations of the agents, the agents were able to maneuver within these constraints of the Unmanned Little Bird to achieve a successful level of swarming. The application of other dynamic constraints of other vehicles could be applied for additional simulation purposes.

Possibilities of future work include many other methods of expanding and extending the existing model. There exist other situations that the model could be simulated for. Currently only two obstacles with a single target was the constant throughout this document. Multiple targets could be setup as well as a vast increase in obstacles. Other simulating objects could have moving obstacles added to simulate unknown threats such as missiles or other aircraft moving through the environment.

As mentioned previously, the dynamics constraints set forth for the agents are not meant to fully embody the motion of the helicopter but to constraint its limitations. In further application this model could be built into a state space controller and the actual equations of motion for the aircraft could be incorporated. This would guarantee a more realistic simulation in the movement and swarming behavior.

One particular portion that could be improved would be the aggregate attraction potential function. This function was created as a means to centralize the model with an additional attraction field yet still contains a discontinuity that induces oscillatory motion in some cases that the gain is too large.

References

[1] Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks IV. pp. 1942–1948. CrossRef

[2] Shi, Y., & Eberhart, R. (1998, May). A modified particle swarm optimizer. In Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Conference on (pp. 69-73). IEEE. CrossRef

[3] Angeline, P. J. (1998, May). Using selection to improve particle swarm optimization. In *Proceedings of IEEE International Conference on Evolutionary Computation* (Vol. 89). CrossRef

[4] Ozcan, E., & Mohan, C. K. (1999). Particle swarm optimization: surfing the waves. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on* (Vol. 3). IEEE. CrossRef

[5] De Vries, E., & Subbarao, K. (2011, January). Cooperative control of swarms of unmanned aerial vehicles. In 49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Orlando, USA, 4-7 January 2011; AIAA 2011-78. American Institute of Aeronautics and Astronautics (AIAA). CrossRef

[6] Chiew, S. H., Zhao, W., & Go, T. H. (2013). Swarming Coordination with Robust Control Lyapunov Function Approach. *Journal of Intelligent & Robotic Systems*, 1-17.

[7] Choset, H. M. (Ed.). (2005). Principles of robot motion: theory, algorithms, and implementation. MIT press.

[8] MATLAB, The MathWorks, Inc., Natick, Massachusetts, United States.

[9] Tsourdos, A., White, B., & Shanmugavel, M. (2010). *Cooperative path planning of unmanned aerial vehicles* (Vol. 32). John Wiley & Sons. CrossRef

[10] *Helicopter Flight Training Manual*. (2006). Department of Transport Canada.

[11] MD 530F Performance Specifications. MD Helicopters.