**SWARMING BEHAVIOUR IN DRONES**

**(QUADCOPTERS)**

Department of Computer Science

University of Karachi

MCS Final

Project Supervisor: Sir Dr. S.M Khalid Jamal

Group Members:

Naveed Hussain

EP-19101063,

Osama Muhammad Shaikh

EP-19101066

&

Zohaib Ali

EP-19101095

1. **Abstract**

This paper proposes the detailed information about the Swarm behavior. Converting this swarm behavior in the technology and applying this technology to the UAVs for increasing the range of applications. In this paper we explain every stuff about swarm technology step by step. Application wise its types and use is explained and different type of swarming algorithms are also defined. We have also tried to convert this behavior in UAVs. As UAVs are performing vital role in social and defense sectors. We have developed algorithms for fulfilling the swarm behaviors like to avoid the collision to obstacles and to other drones we have developed Dynamic Collision-Free Motion Coordination and Navigation System .Due to some limitations we could not perform experiments, but this paper may help the coming students to precede the research in this technology.

1. **Introduction & Problem Statement**

The UAVs can be used in military missions, social causes and for commercial uses. For military missions it can be used in surveillance and reconnaissance. For social it may be used in search and rescue operations similarly it may perform significant role in image processing and infrastructure of country. These successes in UAVs field encouraged us to maximize the research in this field and enjoying the benefits of group of UAVs. Therefor we (Human Being) developed swarming behavior in the drones by being inspired from the social animals, such as insects, winged animals and fish that exhibit a collective intelligence which appears to achieve complex goal through simple rules and local interactions [1]. The main benefits of a swarm drones includes

1. Robustness:- Ability to cope with loss of individuals in the group of drones
2. Scalability:- Ability to handle the increasing of members in the group
3. Flexibility :- Ability to handle the sudden changes in the group of drones
4. Collision avoiding Position updating :- To avoid the collision between group members and other moving and static obstacles also updating the position to reach the destination
5. Full Duplex Communication to Base station and between members.

This technology market worth is about 447.2 million USD by 2030[2]. This technology is changing the way wars are fought. In February 20119, “the defense secretary said “Swarm Squadrons” will be deployed by the British armed forces in the coming year [3].

1. **Related Works**

**3.1 Swarm Intelligence (SI) Algorithms**

To achieve the swarming behavior in our machines we have various types of swarm intelligence algorithms designed by observing the various social animals *figure-01*. Here we are going to brief the some algorithms to gain the concept of swarm behavior.

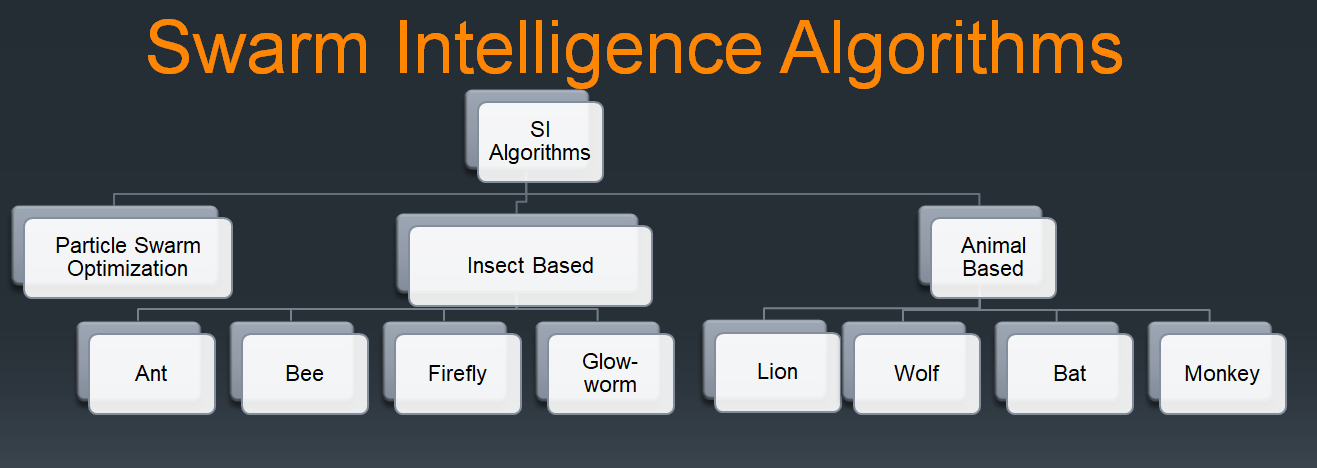


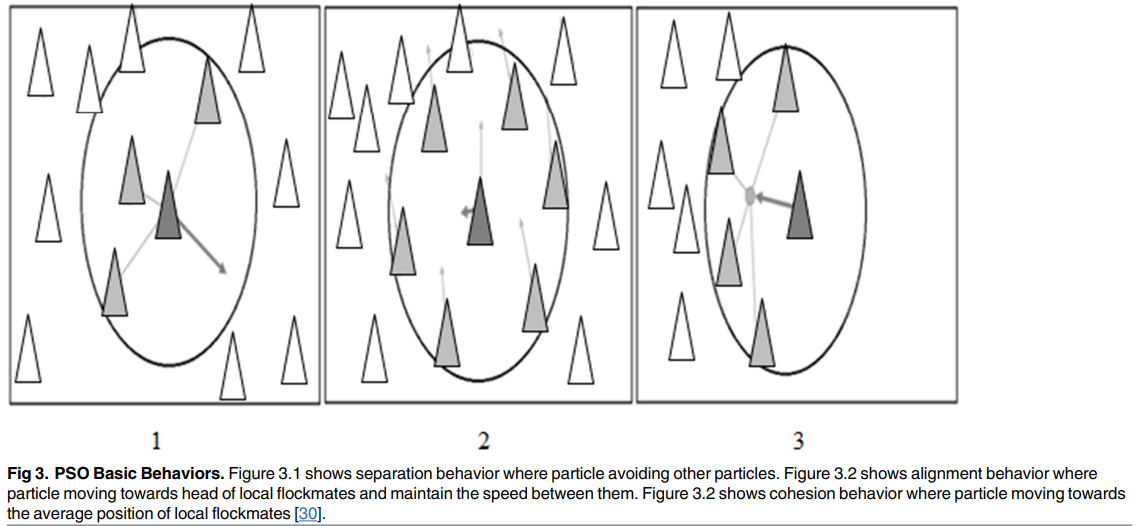
Figure-01

1. **Particle Swarm Optimization**

It uses a simple mechanism that mimics swarm behavior in birds flocking and fish schooling to guide the particle to search for global optimal solutions

Include 3 simple behaviors

1) Separation

2) Alignment

3) Cohesion

Figure-02[4]

**Application:**

****

Figure-03[5] Figure-04[6]

1. **Insect Base**

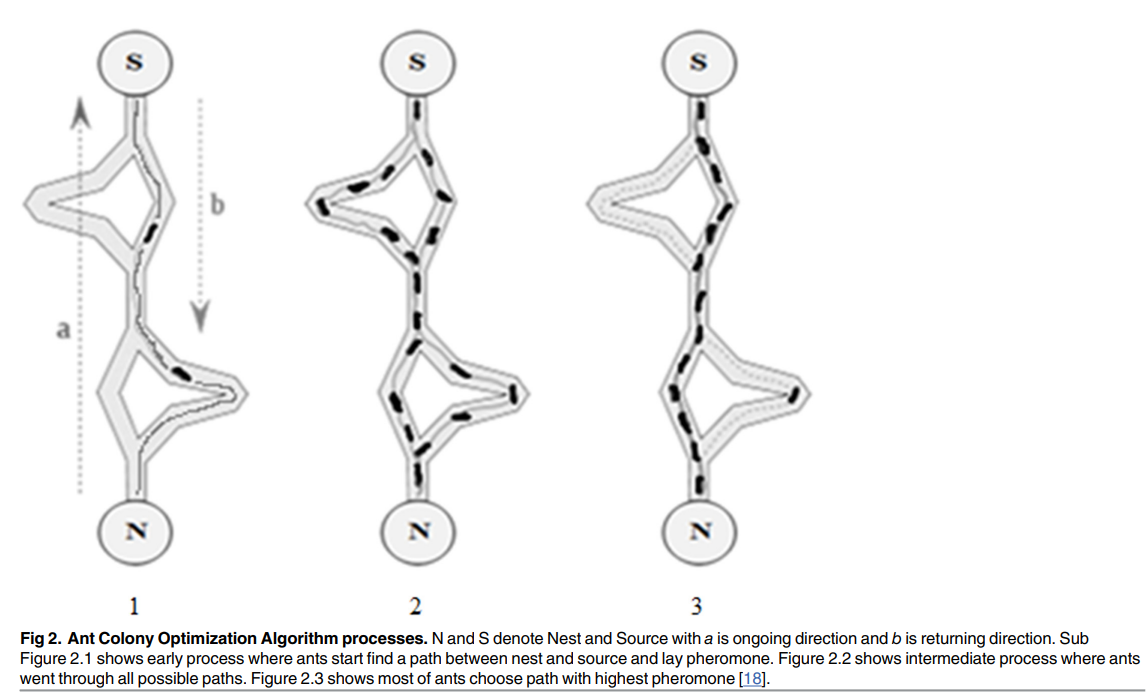
**Ant Colony Optimization:-**

* After inspiring from the foraging behavior of real ants we made this algorithms
* This algorithms consist of main four components
* Ant
* Pheromone
* Daemon action
* Decentralized Control

Application:

1. Finding shortest Path
2. Development of intelligent solutions for heavy transportation
3. Data mining
4. Vehicle Routing Problems
5. University time Table
6. Networking Problems etc.

Figure-06[8]

Figure-05[7]

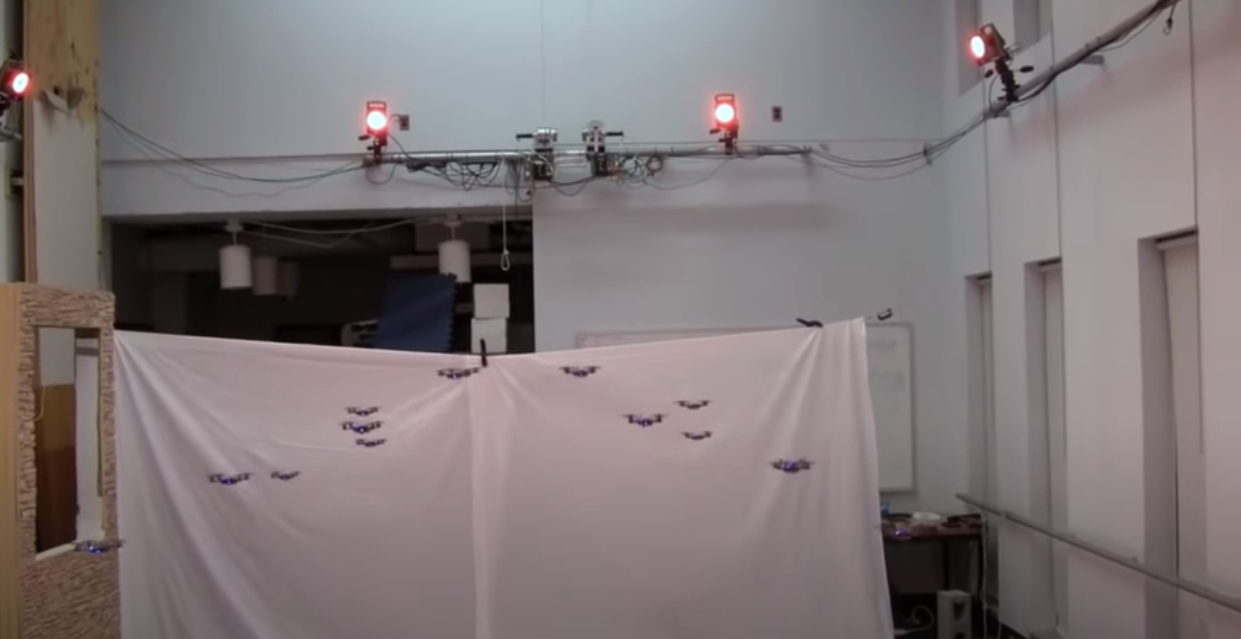
**3.2 Types of Swarming Environments**

We have divided the swarming in two types keeping their environment in consideration.

1) Indoor Swarming

2) Outdoor Swarming

**Indoor Swarming:-**

In indoor swarming we develop the swarming in drones and these drones are limited to some covered places like halls or compounds. Often in this method we use cameras to capture the motion of members of the swarm and generating velocity commands accordingly to update the position of individuals in group avoiding the collision. Some other techniques are also used to for navigation but in this type I have not found a range of applications. Therefor we have concentrated on the Outdoor swarming. As in figure below the drones are surrounded by cameras to capture the motion of the drones and generating paths for individuals with the help of central high processing power computer system. In future we may develop the useful applications of this type of swarming environment.[15] Figure-06[9]

**Outdoor Swarming:-**

In outdoor swarming we develop swarming behavior in the drones that are not limited to specific hall or compound because in this environment we use different algorithms to update position of the drones instead of motion capturing system using the cameras. Some of the algorithms used for position update and collision avoidance are

1. Swarm Gradient Bug Algorithm (SGBA). [10]
2. Particle Swarm Optimization (PSO) [11]
3. Linear Control Method [11]

**Applications:-**

1. Image Processing
2. Search & Rescue
3. Mapping for infrastructure
4. Countering antimissile system
5. Efficient searching and targeting enemies
6. The detection identification and surveillance of a target.



Figure-07[12]

1. **The proposed Swarm Navigation & Coordination**

In this paper, we present a dynamic, collision-free path generation and navigation system for swarms of UAVs. The proposed system uses geographical locations of the UAVs and of the successfully detected, static and dynamically appearing, moving obstacles to predict and avoid the following:

(1) UAV-to-UAV collisions

(2) UAV-to-static-obstacle collisions

(3) UAV-to-moving-obstacle collisions.

It comprises three main components:

(1) Complex event processing (CEP) and collision prediction module[13]

(2) Mutually exclusive locking mechanism

(3) Collision avoidance mechanism.

The CEP and collision prediction module leverages efficient runtime monitoring and CEP to make timely predictions. The mutually exclusive locking mechanism prevents multiple UAVs from attempting to fly to the same location at the same time. The collision avoidance mechanism tries to find best ways to prevent the UAVs from colliding into one another with the successfully detected static and moving obstacles in the flying zone. A distinctive feature of the proposed system is its ability to foresee potential collisions and proactively find best ways to avoid the predicted collisions in order to ensure safety of the entire swarm. In contrast to the existing works our proposed system does not depend on a planning phase and produces efficient, collision-free paths in an online manner. We focus on collision prediction and avoidance and online path generation and navigation for swarms of UAVs.[14]

The proposed system not only provides support for online collision prediction and avoidance, it also generates complete routes for all UAVs in the swarm dynamically. Unlike other motion path planner this method does not know about the flying zone beforehand. In this approach the drone take off from their start locations and fly uninterruptedly towards their destination until the proposed system predicts a collision and triggers our collision avoidance mechanism to prevent the predicted collision.

**Terminology and Notations**

Area Three-dimensional flying zone

Cl Wireless communication latency

Dis Distance between two consecutive UAVs

Diss Safe distance for the UAVs

SWARMSwarm of drones

lfin Final or destination location of a UAV

li A location in AREA

lin Initial or start location of a UAV

MOV\_OBS Set of moving obstacles

Pt Obstacle detection and processing time

routei A UAV route

senr Sensing range of the UAVs

Sp Maximum speed of the UAVs

STA\_OBS Set of static obstacles

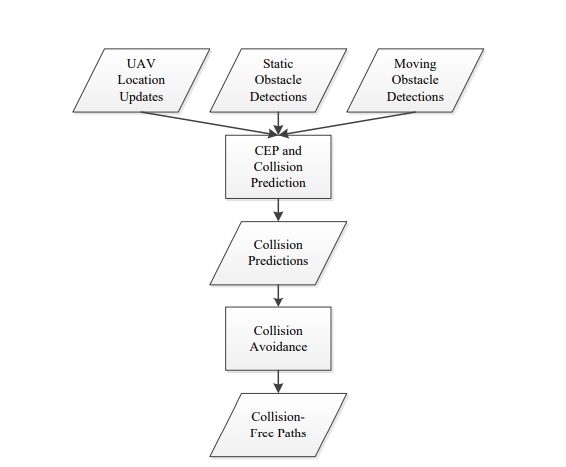
**Minimum Distance between Drones:**

Let the mission flying zone be represented by a finite set of locations AREA = {l1, l2, ..., lM}, where each location li is represented as a point in a three dimensional space (x, y, z). In an outdoor mission, the dimensions x, y, z may correspond with latitude, longitude, and altitude or elevation. To ensure a suitable formation of the fleet, it is assumed that the distance between any two consecutive locations in AREA is less than or equal to the sensing range senr of the UAVs and greater than or equal to the safe distance diss for the UAVs. For example, the front and rear sensing range senr of Phantom 4 Pro UAV is up to 30 meters. Therefore, if the fleet comprises Phantom 4 Pro UAVs, the maximum distance between any two consecutive locations li , lj ∈ AREA | i ≠ j should be less than or equal to 30 meters. The safe distance diss for UAVs depends on their maximum speed Sp, obstacle detection and processing time Pt, and wireless communication latency Cl [5]. For example, if two UAVs are found heading towards each other at a maximum speed Sp of 5 meter per second each and with an obstacle detection and processing time Pt of 0.5 seconds and a wireless communication latency Cl of 0.2 seconds, the safe distance diss can be estimated as

diss = 2 · Sp (2 · Cl + P t) Here minimum distance is 9 meter. Hence diss ≤ dis ≤ senr

* 1. **Collision-Free Motion Coordination and Navigation**

Figure below present the architecture and overview of the proposed real time, collision free motion coordination and navigation system for a swarm of UAV.



Main components of above architecture are

1. CEP & Collision Prediction module
2. Collision Avoidance Mechanism

Inputs to the system are UAV location update, static obstacle detections and moving obstacle detections. Based on these three inputs, the CEP and collision prediction module predicts

(1) UAV-to-UAV collisions

(2) UAV-to-static-obstacle collisions

(3) UAV-to-moving-obstacle collisions.

The proposed system implements a safety-first approach. Therefore, a hazardfree, safe operation of the UAV fleet takes precedence over other objectives including lengths of the UAV routes, timely arrival of the UAVs to their destinations, and achievement of any other mission-specific goals. As a consequence, we do not formulate the problem as an optimization problem. Instead, we use a simple greedy approach for computing UAV routes. In the proposed system, the UAVs takeoff from their start locations and fly uninterruptedly towards their destinations until the CEP and collision prediction module predicts a collision in which case our collision avoidance mechanism is triggered to bypass or avoid the collision by redirecting the UAVs into some other directions, putting them into the hover-in-place mode, or letting them to temporarily retreat or backtrack. The two main components of the proposed system are described in the following subsections.

1. **CEP & Collision Prediction module**

CEP is a technique for real time, fast processing of a large number of events from an event stream to derive some complex events and to identify important patterns in the event stream. CEP has found its applications in a variety of business domains such as retail management, health-care, and cloud computing. The basic or primitive events in CEP are processed into complex or composite events by means of event processing queries, analysis rules, and patterns, which are written in a Structured Query Language (SQL)-like language. Therefore, CEP provides a similar functionality for real time event streams to that of a relational database management system for persistent data. One of the most widely used CEP tools is the Esper CEP engine3, which uses Event Processing Language (EPL) for writing event processing queries and patterns. There are three main steps for using Esper CEP engine. In the first step, event types and sources of events are registered with the CEP engine. An event class in Esper is written as a Plain Old Java Object (POJO). The second step requires event processing queries, analysis rules, and patterns to be implemented in EPL. Finally, in the third step, event sinks are implemented which can be used to perform some suitable control and repair actions. The CEP and collision prediction module in our proposed system uses a CEP engine to monitor and keep track of the current location of the UAVs and of the successfully detected static and moving obstacles. The UAVs generate and send location update events on regular intervals, for example every 50 milliseconds. A UAV location update event contains UAV name or identification number of the concerned UAV di ∈ F LEET, UAV location li in the three-dimensional flying zone AREA, and the event time t. The CEP engine receives and processes these events to predict possible UAV-to-UAV collisions in the fleet. Similarly, for each successfully detected static obstacle, a static obstacle detection event is generated and sent to the CEP engine. A static obstacle detection event contains obstacle identification number of the static obstacle soi ∈ STA OBS and the location li ∈ AREA of the static obstacle. The CEP engine processes all UAV location update events and static obstacle detection events to predict UAV-to static-obstacle collisions. Finally, for successfully detected moving obstacles, moving obstacle detection events are generated and sent to the CEP engine. A moving obstacle detection event contains obstacle identification number of the moving obstacle moi ∈ MOV OBS, the location li ∈ AREA of the moving obstacle, and the event time t. The CEP engine processes UAV location update events and moving obstacle detection events to predict UAV-to-moving-obstacle collisions.

Listing1 presents an example EPL query from the proposed system. The query in Listing1 uses two drone location update events to see if two drones are in close proximity of each other. If a match is found, the CEP engine triggers the concerned event sink, which may predict a UAV-to-UAV collision and then invoke the collision avoidance mechanism to prevent the UAVs from colliding into each other.

Select A. drone Name as aName, A. x as aX,

A. y as aY , A . z as aZ ,

B. droneName as bName , B . x as bX ,

B . y as bY , B . z as bZ ,

from DroneLocationEvent . win : time (1 sec ) A ,

DroneLocationEvent . win : time (1 sec ) B

where A . droneName != B . droneName

and A . x in [ B .x -2: B . x +2] and A . y in [ B .y -2: B . y +2]

and A . z in [ B .z -2: B . z +2]

and ( A . x = B . x or A . y = B . y or A . z = B . z )

1. **Collision Avoidance Mechanism**

Whenever the CEP and collision prediction module predicts a collision, it invokes our collision avoidance mechanism which tries to find best ways to avoid or bypass collisions and computes collision-free routes for UAVs in real time. Based on the severity of the predicted collision, its surroundings, and the overall situation of the F LEET and of the successfully detected static and moving obstacles (ST A OBS and MOV OBS) in AREA, our collision avoidance mechanism uses one of the three collision avoidance techniques in the following order:

(1) Redirecting the UAV into another direction

(2) Putting the UAV into the hover-in-place mode until the route is cleared

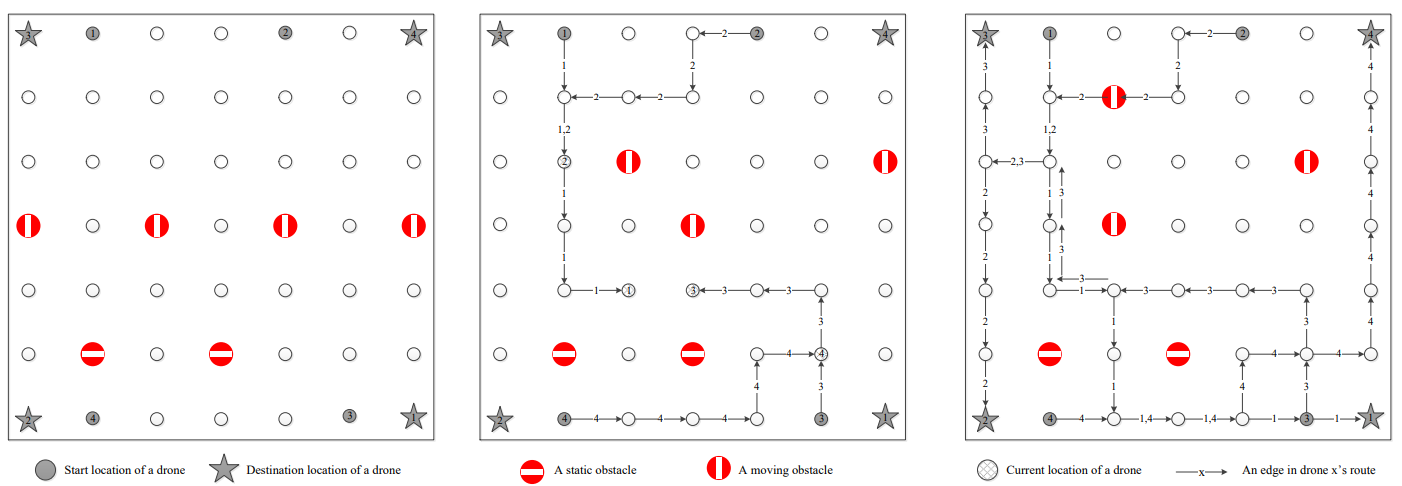
(3) Temporarily retreating or backtracking the UAV to find more suitable, collision-free routes.

The first collision avoidance technique namely redirecting the UAV into another direction means changing the flying direction of the UAV. For example, if a UAV is flying in the x dimension of AREA, but the CEP and collision prediction module predicts a collision due to the presence of an obstacle or another UAV on the path, then the UAV cannot continue a hazard-free flight in the x dimension any more. Therefore, the collision avoidance mechanism redirects the UAV to fly in the y or z dimension so the UAV may be able to avoid or bypass the collision. However, in a densely populated and cluttered flying zone, the collision avoidance mechanism might not be able to immediately compute a bypass route for all drones. Therefore, in such scenarios, the proposed collision avoidance mechanism activates the hover-in-place mode for some of the UAVs until the situation improves and the routes clear. Additionally and as a

|  |
| --- |
| Algorithm 1 Collision avoidance mechanism |
| 1: redirect the UAV into another direction  2: **if** not successful **then**  3: activate the hover-in-place mode until the UAV route is cleared  4: **end if**  5: if not successful then  6: temporarily retreat or backtrack the UAV to find a more suitable, collision-free route  7: **end if** |

1. **An Illustrative Example**

In this section, we present a small example to illustrate the main components and steps of the proposed real time, collision-free motion coordination and navigation system. Although the proposed system works for a realistic, three dimensional flying zone, it is difficult to illustrate and demonstrate a three dimensional flying zone on paper. Therefore, we use a two-dimensional flying zone for a simpler illustration. Figure 2 presents an illustrative example with four UAVs, two static obstacles, and four moving obstacles in a two-dimensional flying zone. The flying zone in our example is shown as a 7x7 grid, where all consecutive locations are a uniform, fixed distance apart from one another. The start and destination location of each drone is also highlighted. The goal is to route the drones from their start locations to their destination locations while avoiding collisions with static and moving obstacles and with the other drones in the fleet. It should be noted that the knowledge of the precise locations of the obstacles in this example is only for illustrative purposes. The proposed system does not make any assumptions on the number and locations of the static and moving obstacles in the flying zone. Similarly, although Figure 2a shows that all moving obstacles are present in the flying zone before the start of the mission, in a realistic scenario some moving obstacles may dynamically appear in the flying zone during the execution of the mission. As described in Section 2, the proposed system relies on obstacle sensing and detection capability of the drones in the fleet. Therefore, each drone detects obstacles on its way and in its surroundings. Moreover, on every successful detection of a static or a moving obstacle, appropriate events are sent to the CEP and collision prediction module which may predict a collision and invoke the collision avoidance mechanism.

* (a) Before the start of the mission. (b) After 5 intervals
* (c) After completion of mission

Therefore, we illustrate the main steps while assuming that the drones do not require any additional support or steps for obstacle detection. Figure 2b presents a snapshot of the flying zone after five time intervals have elapsed since the start of the mission. It shows that each UAV started flying from its start location and flew towards its destination location while randomly choosing to fly in the horizontal or vertical dimension in each time interval. Figure 2b also shows that the left most moving obstacle from Figure 2a left the flying zone during the execution of the mission and that the remaining moving obstacles moved to some new arbitrary locations within the flying zone. Although the moving obstacles moved in an arbitrary fashion either horizontally or vertically, in five time intervals each moving obstacle moved only one step, that is, only to a next consecutive location in the flying zone. Therefore, the moving obstacles moved slower than the drones. This is a reasonable assumption because if the moving obstacles move faster than the drones, even the most advanced and fastest collision detection, prediction, and avoidance mechanisms might not be able to avoid UAV-to-moving-obstacle collisions. The labeled, directional edges in Figure 2b show the collision-free UAV routes generated by the proposed system in real time. For example, in the top left corner of Figure 2b, the first downward edge labeled 1 means that UAV 1 flew in the downward direction. Similarly, the next edge in the same direction labeled 1,2 shows that UAV 1 and 2 used the same edge. However, two UAVs using the same edge does not mean a UAV-to-UAV collision. A UAV-to-UAV collision on an edge can happen when two UAVs fly at the same edge at the same time. In this example, UAV 1 and UAV 2 flew on the same edge, but in different time intervals. UAV 1 left the edge before UAV 2 arrived there and hence there was no collision-hazard between the two UAVs. Figure 2b also shows the current locations of the UAVs after five time intervals. It can be seen that 10 all UAVs except UAV 3 flew five steps. UAV 3 flew four steps and then hovered in the fifth time interval because the system could not find a collision-free move for UAV 3. UAV 1 in Figure 2b started flying vertically in the downward direction and continued towards its destination until it detected a static obstacle. At this stage, our CEP and collision prediction module predicted a UAV-to-static obstacle collision and invoked our collision avoidance mechanism, which redirected the UAV into the horizontal, rightward direction so the drone could continue flying towards its destination. However, in the same time interval, UAV 3 tried to fly into the same location where UAV 1 was headed. The two UAVs detected each other and the CEP and collision prediction module predicted a UAV-to-UAV collision. As a result, our collision avoidance mechanism was invoked, which tried to redirect UAV 3 in the vertical, upward direction, but the UAV detected a moving obstacle at that location and the CEP and collision prediction module predicted a UAV-to-moving-obstacle collision. Therefore, the collision avoidance mechanism activated the hover-in-place mode for UAV 3, but let UAV 1 to continue flying. Hence, UAV 3 flew only four steps in five time intervals. In this example, UAV 2 and 4 did not encounter a collision-hazard and flew normally towards their destinations. Figure 2c shows a snapshot of the flying zone after the completion of the mission. It shows that how each drone found its way to its destination while avoiding obstacles and other drones on its way. Once again, the remaining three moving obstacles moved to some new arbitrary locations within the flying zone. In the sixth time interval, UAV 1 was redirected in the downward direction to avoid a collision with UAV 3. Similarly, after flying downwards for two time intervals, UAV 1 reached at the end of the flying zone and was once again redirected to the horizontal, rightward direction. Finally, after flying for a few more intervals in the rightward direction, UAV 1 reached its destination. As can be seen in Figure 2c, all other UAVs found their ways in similar ways.

1. **Experimental Evaluation**

To demonstrate and evaluate our proposed system, we have implemented it in a Java-based simulation. This section briefly describes some important implementation details along with an experimental evaluation involving a series of experiments. As described in Section 3.1, the implementation of the first main component of the proposed system called the CEP and collision prediction module is based on Esper, which is a Java-based CEP engine. The second component, called the collision avoidance mechanism, implements Algorithm 1. It is not based on a particular tool. However, its implementation is currently at an early stage and does not support the third collision avoidance technique, which temporarily retreats or back tracks a UAV to find a more suitable, collision-free route. We hope to complete the implementation of this technique in our future work. In the current implementation, each drone randomly flies into one of the three dimensions (x, y, z) as long as they are in the direction of the drone’s destination location. However, when the collision avoidance mechanism is activated, then it follows one of the first two collision avoidance techniques as described in Section 3.2. We have implemented a simple, controlled simulation platform that does not take into account complex physical phenomena and uncontrolled environment variables such as gravity and wind. The objective is to test and evaluate the proposed system in an ideal scenario while ignoring and minimizing the effects of the external, uncontrolled factors. Therefore, it is easier to analyze and interpret the results. The implementation also assumes that all drones fly at the same speed and that there were no internal drone failures during the execution of the mission.

1. **A Smaller Problem Instance**

Experiment 1 was designed to evaluate the collision prediction and avoidance capability and effectiveness of the proposed system. In particular, we wanted to see how many collisions of each type are successfully predicted and avoided. Moreover, we also measured route lengths of the system-generated UAV routes and runtime performance of the system. The experiment used a 10x10x10 flying zone with 20 drones, 20 static obstacles, and 20 moving obstacles. All drones and obstacles were placed randomly. However, to ensure that the drones do not collide during takeoff, unique start locations were used and no obstacles were placed at the drone start locations. Similarly, the destination locations for the drones were also chosen randomly, but it was ensured that all destination locations are unique and that no obstacles were present at the destination locations.

**5.1 Collision Prediction and Avoidance**

Table 2 presents a summary of the results from Experiment 1. The results show that there were a total of 30 UAV-to-UAV detections, that is, events when a UAV detected another UAV in its close proximity. Similarly, a total of 59 static obstacle detections and 5 moving obstacle detections. For each UAV and obstacle detection, appropriate events were sent to the CEP and collision prediction module which predicted a possible collision and accordingly invoked the collision avoidance mechanism to prevent the UAVs from colliding into one another and into the successfully detected static and moving obstacles. As a result, all UAV-to-UAV, UAV-to-static-obstacle, and UAV-to-moving-obstacle collisions were avoided and all drones successfully completed their maneuvers.

**5.1.2 Route Lengths and Runtime Performance**

Table 2 also shows average and standard deviation of the UAV route lengths measured in terms of drone moves among consecutive locations or edges traversed in the flying zone. The average and standard deviation of the route lengths were ≈19 and ≈4, respectively. Therefore, the results show that the system-generated routes and their lengths for a 10x10x10 flying zone were quite reasonable. Moreover, only 2 UAVs were put into hover-in-place mode and each UAV hovered for a maximum of 1 time interval. The length of a time interval was 50 milliseconds and the simulation run completed in 2 seconds. The results show that the proposed system has an excellent runtime performance and it is highly suitable for smaller problem instances. The performance and scalability of the proposed system are further demonstrated in the next experiment involving a larger problem instance. [16]

**Table 2: Summary of the results from Experiment 1**

|  |  |
| --- | --- |
| Flying zone dimensions  Number of UAVs  Number of static obstacles  Number of moving obstacles | 10 X 10 X10  20  20  20 |
| Number of UAV-to-UAV detections  Number of static obstacle detections  Number of moving obstacle detections | 30  59  5 |
| Number of UAV-to-UAV detections  Number of static obstacle detections  Number of moving obstacle detections | 0  0  0 |
| Average UAV route length (moves)  Standard deviation of route length (moves) | ≈19  ≈4 |
| Number of UAVs put into hover-in-place mode  Maximum number of time intervals a UAV hovered-in-place | 2  1 |
| Length of a time interval (milliseconds)  Total simulation time (seconds) | 50  2 |

1. **CONCLUSION**

In this paper, we discussed about the Swarming behavior of different social animals. We explained the Swarming Behavior and discussed to convert this behavior in to machines. We defined the different types of swarming intelligence algorithms to make a base about the swarming intelligence. Then we discussed the swarming in two environments

1) Indoor Swarming: - having limited flying zone often use motion capturing system to generate the path for the movement of drones.

2) Outdoor Swarming: - No limited flying zone but can fly on open sky by coordinating and generating paths for position updating without collision. We listed different types of algorithms used for navigation in outdoor swarming like SGBA Swarm Gradient Bug Algorithm.

Then we proposed our method for Coordination and Navigation of drones participating in the swarm of drones. We described three safety requirements that must be satisfied to ensure a collision-free operation of an Unmanned Aerial Vehicle (UAV) swarm and presented a real time, collision-free motion coordination and navigation system for a UAV swarm. The proposed system uses geographical locations of the UAVs and of the successfully detected, static and dynamically appearing, moving obstacles to predict and avoid:

(1) UAV-to-UAV collisions

(2) UAV-to-static obstacle collisions

(3) UAV-to-moving-obstacle collisions.

It comprises two main components:

(1) Complex Event Processing (CEP) and collision prediction module

(2) collision avoidance mechanism.

The CEP and collision prediction module leverages efficient runtime monitoring and CEP to make timely predictions. The collision avoidance mechanism tries to find best ways to prevent the UAVs from colliding into one another and with the successfully detected static and moving obstacles in the flying zone. Therefore, a distinctive feature of the proposed system is its ability to foresee a risk of a collision in real time and proactively find best ways to avoid the predicted collisions in order to ensure safety of the entire swarm.

This proposed method has excellent performance and suitable for some densely populated, cluttered flying zones.

As part of our future work we are planning to implement this proposed system in a more realistic simulation environment. Also planning to implement this algorithm on real Drones to complete the implementation of our proposed collision avoidance mechanism to support third collision avoidance technique which temporarily retracts or backtracks a UAV to find a more suitable, collision free route.

Refrences

1. Swarm coordination of mini-UAVs for target search using imperfect sensors
2. <https://www.marketsandmarkets.com/PressReleases/swarm-intelligence.asp>
3. <https://www.bbc.com/news/technology-47555588>
4. <https://www.researchgate.net/figure/PSO-Basic-Behaviors-Figure-31-shows-separation-behavior-where-particle-avoiding-other_fig14_277084736>
5. <https://unanimous.ai/social-optima/>
6. <https://www.youtube.com/watch?app=desktop&v=KhDEEN4gcpI>
7. <https://link.springer.com/article/10.1007/s40747-020-00138-3>
8. <https://slideplayer.com/slide/5950228/>
9. <https://www.youtube.com/watch?v=YQIMGV5vtd4>
10. Minimal navigation solution for a swarm oftiny flying robots to explore an unknown environment
11. SWARM INTELLIGENCE FOR AUTONOMOUS UAV CONTROL
12. <https://www.embention.com>
13. <http://www.espertech.com/esper/>
14. <https://www.hindawi.com/journals/sp/2020/8530763/>
15. Learning Vision-based Flight in Drone Swarms by Imitation
16. Towards A Realtime, Collision-Free Motion Coordination and Navigation System for a UAV Fleet