Towards A Realtime, Collision-Free Motion Coordination and Navigation System for a UAV Fleet

Adnan Ashraf, Amin Majd, Elena Troubitsyna
Faculty of Natural Sciences and Technology
Åbo Akademi University
Turku, Finland
first.last@abo.fi

Abstract

This paper presents a realtime, collision-free motion coordination and navigation system for an Unmanned Aerial Vehicle (UAV) fleet. The proposed system uses geographical locations of the UAVs and of the successfully detected, static and moving obstacles to predict and avoid: (1) UAV-to-UAV collisions, (2) UAV-to-static-obstacle collisions, and (3) UAV-to-moving-obstacle collisions. Our collision prediction approach leverages efficient runtime monitoring and Complex Event Processing (CEP) to make timely predictions. A distinctive feature of the proposed system is its ability to foresee a risk of a collision in realtime and proactively find best ways to avoid the predicted collisions in order to ensure safety of the entire fleet. We also present a simulation-based implementation of the proposed system along with an experimental evaluation involving a series of experiments. The results demonstrate that the proposed system successfully predicts and avoids all three kinds of collisions in realtime. Moreover, it generates efficient UAV routes, has an excellent runtime performance, efficiently scales to large-sized problem instances involving dozens of UAVs and obstacles, and is suitable for some densely populated, cluttered flying zones.

1 Introduction

An Unmanned Aerial Vehicle (UAV), or drone, is a semi-autonomous aircraft that can be controlled and operated remotely by using a computer along with a radio-link [3]. UAVs can be classified into different types based on their design, size, and flying mechanism. Among the existing types, the quadrotors or quadrocopters are particularly popular because of their simple design, small size, low cost, greater maneuverability, and the ability to hover-in-place. A quadrotor uses two pairs of identical, vertically oriented propellers of which one pair spins clockwise and the other spins counterclockwise. Commercially-available quadrotors are increasingly been used in a variety of applications such as monitoring and surveillance, search and rescue operations, photography and filming, media coverage of public events, and aerial package delivery.

With the growing popularity and use of UAVs for consumer applications, the number of incidents involving drones is also increasing dramatically. In the United States alone, the Federal Aviation Administration (FAA) receives more than 100 reports every month of unauthorized and potentially hazardous UAV activity reported by pilots, citizens, and law enforcement[1]. Ensuring a hazard-free, safe flight is also equally important for indoor applications. Therefore, motion safety of UAVs is a prime concern for UAV operators. It refers to the ability of the UAVs to detect and avoid collisions with static and moving obstacles in the environment.

Some of the contemporary, commercially-available quadrotors come with a limited support for detecting and avoiding static and moving obstacles. For example, DJI's Phantom 4 Pro[2] (released in November 2016) uses five-directional sensors to provide obstacle detection or sensing in five directions with a front and rear sensing range of up to 30 meters and up to 7 meters for left and right side. However, the left and right sensing capability does not work in most of the flight modes. Similarly, the obstacle avoidance mechanism does not work in all kinds of scenarios.

For larger and more complex applications and tasks which are either beyond the capabilities of a single UAV or can not be performed efficiently if only a single UAV is used, multiple UAVs can be used together in the form of a swarm or a fleet. In such scenarios, a safe operation can not be guaranteed without preventing the UAVs from colliding with one another and with static and dynamically appearing, moving obstacles in the flying zone. Therefore, in the context of a UAV fleet, ensuring motion safety entails devising and implementing a realtime motion path planning, coordination, and navigation system for multiple UAVs with an integrated support for collision prediction and avoidance.

In this paper, we present a realtime, collision-free motion coordination and navigation system for a UAV fleet. 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, and (3) UAV-to-moving-obstacle collisions. It comprises two main components: (1) a Complex Event Processing (CEP) and collision prediction module and (2) a 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 realtime and proactively find best ways to avoid the predicted collisions in order to ensure safety of the entire fleet.

---

[1] `https://www.faa.gov/uas/resources/uas_sightings_report/`

[2] https://www.dji.com/phantom-4-pro

We also present a simulation-based implementation of the proposed system along with an experimental evaluation involving a series of experiments. Our proposed navigation system, its implementation, experiments, and results are not based on or limited to a particular application of the UAV fleets. Instead, they are generic enough to be applicable to a wide range of applications such as search and rescue operations and aerial package delivery. We assume that the UAV fleet executes certain missions, in which each UAV flies from a starting location to a destination location while avoiding all three kinds of collisions. The results demonstrate that the proposed system successfully detects and avoids all three kinds of collisions in realtime. Moreover, it generates efficient UAV routes, has an excellent runtime performance, efficiently scales to large-sized problem instances involving dozens of UAVs and obstacles, and is suitable for some densely populated, cluttered flying zones.

We proceed as follows. Section 2 describes three safety requirements that must be satisfied to ensure a collision-free operation of a UAV fleet. The proposed realtime, collision-free motion coordination and navigation system for UAV fleets is presented in Section 3. In Section 4, we illustrate the main steps of our proposed approach on a small example. Section 5 presents a simulation-based implementation of the proposed system along with an experimental evaluation involving a series of experiments. Section 6 reviews important related works. Finally, we present our conclusions in Section 7.

2 Safety Requirements for a UAV Fleet

As stated in the previous section, it is assumed that the UAV fleet executes certain missions, in which each UAV flies from a starting location to a destination location while avoiding all three kinds of collisions. The proposed system not only predicts and avoids collisions, but also generates a complete path for each UAV in realtime. Unlike traditional motion path planning approaches that assume that all obstacles and their precise locations are known before the start of the mission, the proposed approach does not assume any a priori knowledge of the obstacles. Therefore, it does not require a preliminary, off-line motion planning phase to produce efficient paths for the UAVs. In our approach, the UAVs takeoff from their start locations and fly uninterruptedly towards their destinations until a collision is predicted, in which case our collision avoidance mechanism is triggered to ensure a collision-free, safe operation of the fleet. The safety requirements for a UAV fleet can be formulated as follows:

Req1: UAVs do not collide with static obstacles in the flying zone.

Req2: UAVs do not collide with one another.

Req3: UAVs do not collide with dynamically appearing, moving obstacles in the flying zone.

We assume that the terrain of the flying zone is not known beforehand. Therefore, the proposed system does not make any assumptions on the number

Table 1: Summary of concepts and their notations

Notation	Description
$AREA$	Three-dimensional flying zone
Cl	Wireless communication latency
dis	Distance between two consecutive UAVs
dis_s	Safe distance for the UAVs
$FLEET$	Set of drones
l_{fin}	Final or destination location of a UAV
l_i	A location in $AREA$
l_{in}	Initial or start location of a UAV
MOV_OBS	Set of moving obstacles
Pt	Obstacle detection and processing time
$route_i$	A UAV route
sen_r	Sensing range of the UAVs
Sp	Maximum speed of the UAVs
STA_OBS	Set of static obstacles

and locations of the static and dynamically appearing, moving obstacles. Moreover, since some commercially-available quadrotors come with built-in obstacle detection capability, we assume that each UAV is equipped with an appropriate obstacle detection capability and can successfully detect all static and dynamically appearing, moving obstacles in its surroundings. Therefore, the emphasis of this work is not on obstacle detection. Instead, we focus on collision avoidance and realtime motion coordination and navigation of a drone fleet. For clarity and convenience, the important concepts and notations used in this paper are summarized in Table 1.

Let the mission flying zone be represented by a finite set of locations $AREA = \{l_1, l_2, ..., l_M\}$, where each location l_i 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 sen_r of the UAVs and greater than or equal to the safe distance dis_s for the UAVs. For example, the front and rear sensing range sen_r 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 $l_i, l_j \in AREA \mid i \neq j$ should be less than or equal to 30 meters. The safe distance dis_s 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 dis_s can be estimated as

$$dis_s = 2 \cdot Sp\,(2 \cdot Cl + Pt) \tag{1}$$

which yields 9 meters. Therefore, in this example, the minimum distance between any two consecutive locations $l_i, l_j \in AREA \mid i \neq j$ should be greater than or equal to 9 meters. For simplicity, we assume that all consecutive locations in $AREA$ are a uniform, fixed distance apart from one another denoted as dis, such that $dis_s \leq dis \leq sen_r$. Hence, the flying zone $AREA$ can be viewed as a three-dimensional grid.

Furthermore, let $FLEET = \{d_1, d_2, ..., d_N\}$ be a set of drones or UAVs in the fleet. The static obstacles are represented by the set $STA\_OBS = \{so_1, so_2, ..., so_O\}$. Similarly the dynamically appearing, moving obstacles are represented by the set $MOV\_OBS = \{mo_1, mo_2, ..., mo_P\}$. Each drone occupies a certain location in $AREA$. The drones takeoff from their start locations and fly towards their destination locations. A drone route or path is a sequence of locations from drone's start location to drone's destination location. For a drone d_i, $route_i =< l_{in}, ..., l_{fin} >$ such that $ran(route_i) \subseteq AREA$ and where l_{in} is the initial or start location and l_{fin} is the final or destination location of d_i. Similarly, each static and moving obstacle occupies a certain location in $AREA$. Moreover, the moving obstacles keep on moving arbitrarily until they leave the flying zone.

Since the proposed system does not assume any a priori knowledge on the numbers and locations of the static and moving obstacles and does not depend on a preliminary, off-line motion planning phase, none of the safety requirements can be verified before the start of the mission. For the first safety requirement **Req1** concerning static obstacles, it is necessary that the drones do not fly into a location where a static obstacle is situated. Our proposed system helps the drones to avoid all successfully detected static obstacles in realtime by providing a collision avoidance mechanism. Similarly, for the second safety requirement **Req2** concerning collisions with other drones, it is required that at any given time t each location is occupied by at most one drone. Therefore, the proposed system stops the drones from flying into other drones in the vicinity and provides a collision avoidance and motion coordination and navigation mechanism that allows them to bypass any locations occupied by other drones at time t or to hover-in-place until the route is cleared. For the third safety requirement **Req3** concerning collisions with dynamically appearing, moving obstacles, the proposed system uses a similar approach as used for **Req2** that helps the drones to avoid all successfully detected moving obstacles in realtime.

3 Collision-Free Motion Coordination and Navigation

Figure 1 presents the high-level system architecture and overview of the proposed realtime, collision-free motion coordination and navigation system for a UAV fleet. The main components of the proposed system include: (1) a CEP

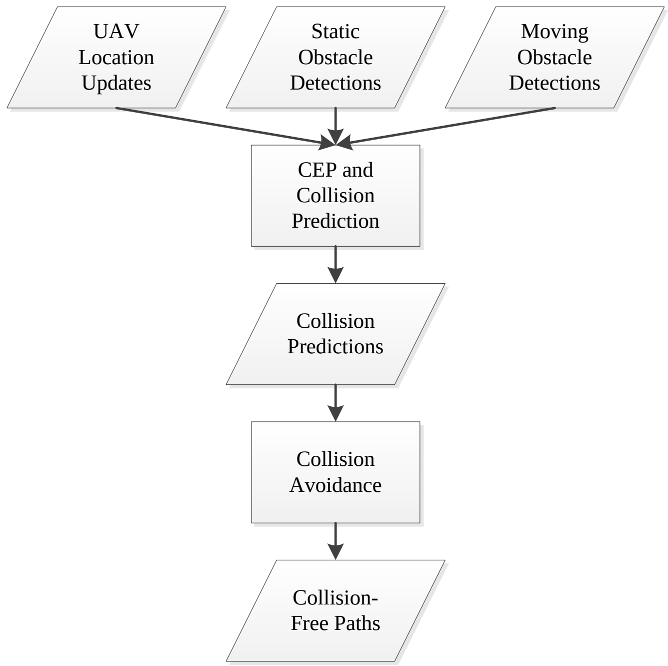


Figure 1: Overview of the proposed realtime, collision-free motion coordination and navigation system for a UAV fleet.

and collision prediction module and (2) a collision avoidance mechanism. The inputs to the system are the UAV location updates, 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, and (3) UAV-to-moving-obstacle collisions. Our collision avoidance mechanism tries to find best ways to avoid or bypass collisions and computes collision-free routes for UAVs in realtime. In a densely populated and cluttered flying zone, it might not be possible to immediately compute a bypass route for all drones. In such scenarios, the proposed system might put some of the drones into the hover-in-place mode until the situation improves and the routes clear. Additionally, it may also let some UAVs to temporarily retreat or backtrack to find more suitable, collision-free routes.

The proposed system implements a safety-first approach. Therefore, a hazard-free, 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.

3.1 CEP and Collision Prediction

CEP is a technique for realtime, 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 [15, 12]. 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 realtime 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 engine[3], 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 $d_i \in FLEET$, UAV location l_i 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 $so_i \in STA\_OBS$ and the location $l_i \in AREA$ 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 $mo_i \in MOV\_OBS$, the location $l_i \in AREA$ 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

[3]http://www.espertech.com/esper/

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.

Listing 1: An EPL example from the proposed system

```
select A.droneName 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)
```

3.2 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 realtime. Based on the severity of the predicted collision, its surroundings, and the overall situation of the $FLEET$ and of the successfully detected static and moving obstacles (STA_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, and (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 can not 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

```
redirect the UAV into another direction
if not successful then
    activate the hover-in-place mode until the UAV route is cleared
end if
if not successful then
    temporarily retreat or backtrack the UAV to find a more suitable,
    collision-free route
end if
```

last resort, it temporarily retreats or backtracks some UAVs to find more suitable, collision-free routes. It should be noted that all three collision avoidance techniques incur some overhead, which might extend the routes and increase the flight durations for some of the UAVs. However, as explained in Section 3, this is inevitable for a safety-first approach. The pseudocode of the proposed collision avoidance mechanism is given as Algorithm 1.

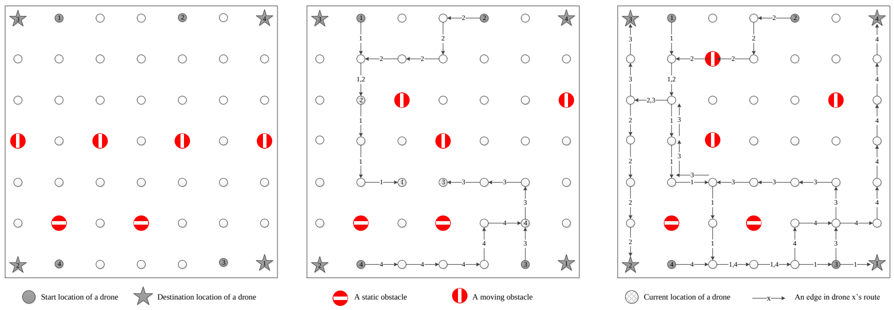
4 An Illustrative Example

In this section, we present a small example to illustrate the main components and steps of the proposed realtime, 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



(a) Before the start of the mission.

(b) A snapshot of the flying zone after five time intervals have elapsed.

(c) After the completion of the mission.

Figure 2: A simple, illustrative example with four UAVs, two static obstacles, and four moving obstacles in a two-dimensional flying zone.

the collision avoidance mechanism. 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 labelled, directional edges in Figure 2b show the collision-free UAV routes generated by the proposed system in realtime. For example, in the top left corner of Figure 2b, the first downward edge labelled 1 means that UAV 1 flew in the downward direction. Similarly, the next edge in the same direction labelled 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 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.

5 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 backtracks 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.

5.1 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.

5.1.1 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

Table 2: Summary of the results from Experiment 1

Flying zone dimensions	10x10x10
Number of UAVs	20
Number of static obstacles	20
Number of moving obstacles	20
Number of UAV-to-UAV detections	30
Number of static obstacle detections	59
Number of moving obstacle detections	5
Number of UAV-to-UAV collisions	0
Number of UAV-to-static-obstacle collisions	0
Number of UAV-to-moving-obstacle collisions	0
Average UAV route length (moves)	≈19
Standard deviation of route length (moves)	≈4
Number of UAVs put into hover-in-place mode	2
Maximum number of time intervals a UAV hovered-in-place	1
Length of a time interval (milliseconds)	50
Total simulation time (seconds)	2

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.

5.2 A Larger Problem Instance

Experiment 2 was designed to evaluate the proposed system against a larger problem instance. The experiment used a 20x20x20 flying zone with 50 drones, 50 static obstacles, and 50 moving obstacles. All drones and obstacles were placed randomly in a similar way as in Experiment 1.

5.2.1 Collision Prediction and Avoidance

A summary of the results from Experiment 2 is presented in Table 3. The results show that there were a total of 21 UAV-to-UAV detections, 64 static obstacle detections, and 2 moving obstacle detections. The proposed system successfully avoided all UAV-to-UAV, UAV-to-static-obstacle, and UAV-to-moving-obstacle collisions and successfully routed all drones to their destinations.

Table 3: Summary of the results from Experiment 2

Flying zone dimensions	20x20x20
Number of UAVs	50
Number of static obstacles	50
Number of moving obstacles	50
Number of UAV-to-UAV detections	21
Number of static obstacle detections	64
Number of moving obstacle detections	2
Number of UAV-to-UAV collisions	0
Number of UAV-to-static-obstacle collisions	0
Number of UAV-to-moving-obstacle collisions	0
Average UAV route length (moves)	≈32
Standard deviation of route length (moves)	≈7
Number of UAVs put into hover-in-place mode	10
Maximum number of time intervals a UAV hovered-in-place	2
Length of a time interval (milliseconds)	50
Total simulation time (seconds)	3

5.2.2 Route Lengths and Runtime Performance

The average and standard deviation of the route lengths were ≈32 and ≈7, respectively. Therefore, the results show that the system-generated routes and their lengths were quite reasonable for a 20x20x20 flying zone. Moreover, 10 UAVs were put into hover-in-place mode and each one of them hovered for a maximum of 2 time intervals. The length of a time interval was the same as in Experiment 1. The simulation run completed in 3 seconds. It shows that the proposed system is also highly suitable for larger problem instances.

5.3 Densely Populated, Cluttered Flying Zone

Experiment 3 was designed to evaluate the proposed system in a densely populated, cluttered flying zone. It used a similar experiment design as in Experiment 1, but with twice as many static and moving obstacles. Therefore, the experiment used a 10x10x10 flying zone with 20 drones, 40 static obstacles, and 40 moving obstacles. All drones and obstacles were placed randomly in a similar way as in Experiment 1 and 2.

5.3.1 Collision Prediction and Avoidance

Table 4 presents a summary of the results from Experiment 3. There were a total of 25 UAV-to-UAV detections, 132 static obstacle detections, and 8 moving obstacle detections. In spite of the fact that the flying zone was cluttered with

Table 4: Summary of the results from Experiment 3

Flying zone dimensions	10x10x10
Number of UAVs	20
Number of static obstacles	40
Number of moving obstacles	40
Number of UAV-to-UAV detections	25
Number of static obstacle detections	132
Number of moving obstacle detections	8
Number of UAV-to-UAV collisions	0
Number of UAV-to-static-obstacle collisions	0
Number of UAV-to-moving-obstacle collisions	0
Average UAV route length (moves)	≈24
Standard deviation of route length (moves)	≈13
Number of UAVs put into hover-in-place mode	2
Maximum number of time intervals a UAV hovered-in-place	2
Length of a time interval (milliseconds)	50
Total simulation time (seconds)	4

obstacles, the proposed system successfully avoided all UAV-to-UAV, UAV-to-static-obstacle, and UAV-to-moving-obstacle collisions. As a result, all drones successfully completed their maneuvers.

5.3.2 Route Lengths and Runtime Performance

The average and standard deviation of the route lengths were ≈24 and ≈13, respectively. The high standard deviation means that some drone routes were significantly longer than the others. The longest system-generated drone route comprised 72 moves, which shows that some drones encountered too many obstacles in their ways and were forced to use longer routes to their destinations. In this experiment, only 2 UAVs were put into hover-in-place mode and each one of them hovered for a maximum of 2 time intervals. The length of a time interval was the same as in Experiment 1 and 2. The simulation run completed in 4 seconds. It shows that the proposed system is also suitable for some densely populated, cluttered flying zones.

6 Related Work

The problem of motion safety of semi-autonomous robotic systems is currently attracting significant research attention. A comprehensive overview of the problems associated with autonomous mobile robots is given in [14]. The analysis carried out in [7] shows that the most prominent routing schemes do not guarantee motion safety. Our approach resolves this issue and ensures not only safety but also provides efficient, realtime routing.

Macek et al. [11] proposed a layered architectural solution for robot navigation. They focused on the problem of safe navigation of a vehicle in an urban environment. They also distinguished between global route planning and collision avoidance control. However, in their work, they focused on the safety issues associated with the navigation of a single vehicle and did not consider the problem of collision-free motion coordination and navigation in the context of fleets or swarms of robots.

Aniculaesei et al. [1] presented a formal approach that employs formal verification to ensure motion safety. They used UPPAAL model checker[4] to verify that a moving robot engages brakes and safely stops upon detection of an obstacle. Since our proposed system does not assume any a priori knowledge on the numbers and locations of the static and moving obstacles and does not depend on a preliminary, off-line motion planning phase, the safety requirements can not be verified before the start of the mission. Therefore, we did not employ formal verification. The solution proposed in our work is fast and flexible as it dynamically generates and recomputes the drone routes in realtime and avoids unnecessary stopping of the drones.

Petti and Fraichard [13] proposed an approach that relies on partial motion planning to ensure safety. They state that a calculation of an entire route is such a complex and compute-intensive problem that the only viable solution is a computation of the next safe states and navigation within them. Their solution supports navigation of a single vehicle. In our work, we have discretized the flying zone and have developed a highly efficient system that computes the next safe states for an entire fleet and provides a mechanism for realtime motion coordination and collision avoidance.

A comprehensive literature review on motion planning algorithms for UAVs can be found in [8]. The approaches reviewed in [8] are applicable to a preliminary, off-line motion planning phase to plan and produce an efficient path or trajectory for a UAV before the start of the mission. Our proposed system does not depend on a planning phase and produces efficient, collision-free paths for an entire fleet in realtime. A more recent survey on motion planning of UAVs can be found in [9].

Augugliaro et al. [2] presented an algorithm for generating collision-free trajectories for a quadrotor fleet. However, they focused on a planned approach that generates feasible paths ahead of time. In contrast, we presented a realtime, collision-free motion coordination and navigation system.

Olivieri [5] and Olivieri and Endler [6] presented an approach for movement coordination of swarms of drones using smart phones and mobile communication networks. They used CEP, but only to analyze and evaluate the formation accuracy of the swarm. Moreover, their work focuses on the internal communication of the swarm and does not provide a solution for collision-free path generation.

[4] http://www.uppaal.org/

Barry and Tedrake [4] proposed an obstacle detection algorithm for UAVs that allows to detect and avoid collisions in realtime. Similarly, Lin [10] presented a realtime path planner for UAVs that detects and avoids moving obstacles. These approaches are only applicable for individual UAVs and they do not provide support for a UAV fleet. In our work, we assumed that each UAV is equipped with an appropriate obstacle sensing and detection capability and does not require any additional support for obstacle detection. Therefore, we focused on collision prediction and avoidance and realtime motion coordination and navigation of a UAV fleet.

7 Conclusion

In this paper, we described three safety requirements that must be satisfied to ensure a collision-free operation of an Unmanned Aerial Vehicle (UAV) fleet and presented a realtime, collision-free motion coordination and navigation system for a UAV fleet. 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, and (3) UAV-to-moving-obstacle collisions. It comprises two main components: (1) a Complex Event Processing (CEP) and collision prediction module and (2) a 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 realtime and proactively find best ways to avoid the predicted collisions in order to ensure safety of the entire fleet.

We also presented a simulation-based implementation of the proposed system along with an experimental evaluation involving a series of experiments. Our proposed navigation system, its implementation, experiments, and results are not based on or limited to a particular application of the UAV fleets. Instead, they are generic enough to be applicable to a wide range of applications. We assumed that the UAV fleet executes certain missions, in which each UAV flies from a starting location to a destination location while avoiding all three kinds of collisions. The results demonstrated that the proposed system successfully predicts and avoids all three kinds of collisions in realtime. Moreover, it generates efficient UAV routes, has an excellent runtime performance, efficiently scales to large-sized problem instances involving dozens of UAVs and obstacles, and is suitable for some densely populated, cluttered flying zones.

As part of our future work, we plan to complete the implementation of our proposed collision avoidance mechanism to support the third collision avoidance technique which temporarily retreats or backtracks a UAV to find a more suitable, collision-free route. We also plan to implement the proposed system in a more realistic simulation environment that allows to take into account complex physical phenomena and uncontrolled environment variables. Moreover, we want to test and evaluate our system for heterogeneous drones that may have diverse capabilities and fly at different speeds. Finally, appropriate support and realtime mechanisms to handle and control the situations arising from internal drone failures during mission execution are also planned as future works.

Acknowledgements

The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions. The work was supported by the DIMECC Cyber Trust Program (http://cybertrust.fi).

References

[1] A. Aniculaesei, D. Arnsberger, F. Howar, and A. Rausch. Towards the verification of safety-critical autonomous systems in dynamic environments. In *Proceedings of the The First Workshop on Verification and Validation of Cyber-Physical Systems*, pages 79–90, 2016.

[2] F. Augugliaro, A. P. Schoellig, and R. D'Andrea. Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1917–1922, 2012.

[3] R. Austin. *Unmanned Aircraft Systems: UAVS Design, Development and Deployment*. Aerospace Series. Wiley, 2010.

[4] A. J. Barry and R. Tedrake. Pushbroom stereo for high-speed navigation in cluttered environments. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3046–3052, 2015.

[5] B. J. O. de Souza. An approach for movement coordination of swarms of unmanned aerial vehicles using mobile networks. Master's thesis, Pontifical Catholic University of Rio de Janeiro, 2015.

[6] B. J. O. de Souza and M. Endler. Coordinating movement within swarms of UAVs through mobile networks. In *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pages 154–159, 2015.

[7] T. Fraichard. A short paper about motion safety. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 1140–1145, 2007.

[8] C. Goerzen, Z. Kong, and B. Mettler. A survey of motion planning algorithms from the perspective of autonomous uav guidance. *Journal of Intelligent and Robotic Systems*, 57(1):65, 2009.

[9] F. Kendoul. Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics*, 29(2):315–378, 2012.

[10] Y. Lin. *Moving Obstacle Avoidance for Unmanned Aerial Vehicles*. PhD thesis, Arizona State University, 2015.

[11] K. Macek, D. Vasquez, T. Fraichard, and R. Siegwart. Safe vehicle navigation in dynamic urban scenarios. In *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pages 482–489, 2008.

[12] A. Mdhaffar, R. B. Halima, M. Jmaiel, and B. Freisleben. Reactive performance monitoring of cloud computing environments. *Cluster Computing*, pages 1–13, 2016.

[13] S. Petti and T. Fraichard. Partial motion planning framework for reactive planning within dynamic environments. In *Proceedings of the IFAC/AAAI International Conference on Informatics in Control, Automation and Robotics*, 2005.

[14] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza. *Introduction to Autonomous Mobile Robots*. Intelligent robotics and autonomous agents. MIT Press, 2011.

[15] E. Wu, Y. Diao, and S. Rizvi. High-performance complex event processing over streams. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, SIGMOD '06, pages 407–418, New York, NY, USA, 2006. ACM.