Swarm-based Counter UAV Defense System

Matthias Brust (matthias.brust@uni.lu)
University of Luxembourg

Grégoire Danoy
University of Luxembourg

Daniel Stolfi
University of Luxembourg

Pascal Bouvry
University of Luxembourg

Research Article

Keywords: UAV, Swarm, Clustering, Counter-UAV, Defense System

Posted Date: December 1st, 2020

DOI: https://doi.org/10.21203/rs.3.rs-113835/v1

Version of Record: A version of this preprint was published at Discover Internet of Things on February 24th, 2021. See the published version at https://doi.org/10.1007/s43926-021-00002-x.

Noname manuscript No.
(will be inserted by the editor)

Swarm-based Counter UAV Defense System

Matthias R. Brust · Grégoire Danoy ·
Daniel H. Stolfi · Pascal Bouvry

Received: date / Accepted: date

Abstract Unmanned Aerial Vehicles (UAVs) have quickly become one of the promising Internet-of-Things (IoT) devices for smart cities thanks to their mobility, agility, and onboard sensors' customizability. UAVs are used in many applications expanding beyond the military to more commercial ones, ranging from monitoring, surveillance, mapping to parcel delivery. As governments plan using UAVs to build fresh economic potential for innovation, urban planners are moving forward to incorporate so-called *UAV flight zones* and *UAV highways* in their smart city design.

However, the high-speed and dynamic flow of UAVs needs to be monitored to detect and, subsequently, to deal with intruders, rough drones, and UAVs with a malicious intent. Autonomous defense systems consisting of collaboratively working UAVs as a swarm will gain increasing importance, since a manually conducted defense is bound to fail due to the high dynamics involved in UAV maneuvering.

This paper proposes a UAV defense system for intercepting and escorting intruders. The proposed UAV defense system consists of a defense UAV swarm, which is capable to self-organize their defense formation in the event of intruder detection, and chase the malicious UAV.

Matthias R. Brust
SnT, University of Luxembourg
E-mail: matthias.brust@uni.lu

Grégoire Danoy
SnT and FSTM-DCS, University of Luxembourg
E-mail: gregoire.danoy@uni.lu

Daniel H. Stolfi
SnT, University of Luxembourg
E-mail: daniel.stolfi@uni.lu

Pascal Bouvry
SnT and FSTM-DCS, University of Luxembourg
E-mail: pascal.bouvry@uni.lu

Our fully localized approach follows a modular design regarding the defense phases and it uses a newly developed balanced clustering to realize the intercept- and capture-formation. The resulting networked defense UAV swarm is resilient against communication losses. Finally, a prototype UAV simulator has been implemented. Through extensive simulations, we demonstrate the feasibility and performance of our approach.

Keywords UAV · Swarm · Clustering · Counter-UAV · Defense System

1 Introduction

Unmanned Aerial Vehicles (UAVs) have quickly found their way into the Internet-of-Things (IoT) ecosystem as part of smart cities thanks to their three-dimensional mobility, agility, and onboard sensors' customizability. UAVs are nowadays used in a wide scope of commercial applications ranging from monitoring, surveillance, mapping to parcel delivery.

As governments plan using UAVs to build fresh economic potential for innovation, urban planners are moving forward to incorporate so-called *UAV flight zones* and *UAV highways* in their smart city designs.

Initiatives like NASA UTM (3) or European U-Space (18) aim at providing a framework for a regulated and safe UAV traffic management in the yet unregulated class-G airspace. However these will not prevent potential intruders, rough drones, or UAVs with malicious intent to enter the shared airspace and potentially harm other UAVs or even citizens.

It is thus necessary to develop counter-UAV systems (C-UAV) that will permit to detect and neutralize malicious UAVs. C-UAV systems which mainly stem from the military domain, e.g. net guns or signal jammers, have a limited scalability and can induce potential collateral damage that is not acceptable in a smart city context.

Our approach proposes an innovative C-UAV system based on defense UAVs (dUAVs) which can autonomously and collaboratively act as a defense swarm. Malicious UAVs (mUAVs) can then be intercepted, captured and escorted out of the flight zone. Our swarm of dUAVs forms a three-dimensional cluster around the mUAV in a way that the latter has a minimum number of movement possibilities. Hereby, we assume that the mUAV is trying to avoid colliding with dUAVs to maintain its functioning. By enclosing the mUAV, its movement possibilities are enforced by the dUAVs such that the mUAV will be constrained to move outside the flight zone (see Fig. 1).

Contributions. We propose a comprehensive C-UAV system, which auto-deploys a swarm of autonomous dUAVs which creates an intercept- and capture-formation to handle mUAVs.

The most outstanding features and contributions of the presented approach are the balanced clustering to realize the intercept- and capture-formation. Additionally, the approach consists of a modular design containing the phases such as swarming, clustering, formation control, and capturing. All parts of

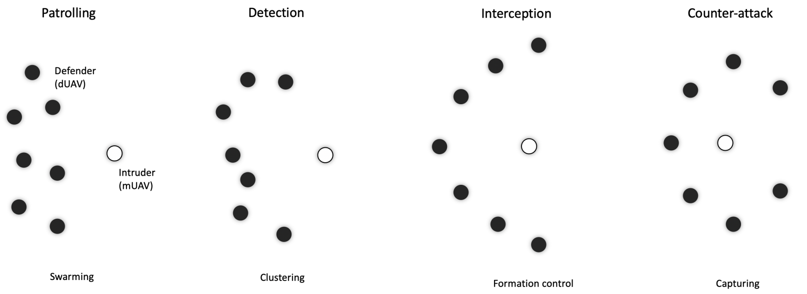


Fig. 1 Illustration of the self-organizing approach of the defense swarm from patrolling to the counter-attack.

the approach are fully localized, and the resulting networked defense UAV swarm is resilient against communication losses.

Results. Simulations have been conducted to understand the influence of the parameters on the performance of the C-UAV defense system. Results show that some parameters have a greater influence over the simulation time that is needed to escort the malicious UAVs than others. More precisely, in order to minimize this time, the following measures need to be taken: The wobbling of UAVs should be as low as possible, while the collision threshold of the dUAVs should be greater than the one of the mUAVs and the dUAV communication range should be as high as possible.

The remainder of this paper is organized as follows. Section 2 presents state-of-the-art. Section 3 gives a motivation to the problem statement, following by a listing of all used notations and the system model that will be used later on. We continue from Section 4.1 to 4.4 by describing each modular phases of our approach in detail. In Section 5, we conclude by showing the results gathered from our simulation experiments.

2 Related Work

This section describes related work to UAV defense systems and UAV defense swarms, but also to formation and positioning of UAVs in the 3D space.

2.1 UAV Defense Systems

UAV defense systems are used in order to protect against threats, being either intentional (e.g., hostile mission) or unintentional (e.g., trespassing in a restricted area).

Such defense systems are typically described as multi-layered or multi-steps systems, designed to cope with a variety of possible attacks (13). While the literature is replete with research on UAVs, until recently counter-UAVs systems have received less attention. Few surveys can be found (5; 2; 15; 10; 25), the most recent and comprehensive one being by Kang et al. (16).

In this work we propose to categorize UAV defense systems in two main classes: ground-based and air-based solutions. Both solutions rely on sensing and mitigation systems, the former allowing to detect and identify a mUAV while the latter permits to act, from simple warning to the neutralization of the mUAV.

A wide range of technologies can be used for the detection and the identification of a potential threat, such as video cameras, thermal cameras, radars, acoustics sensors, magnetic sensors or RF signal detection. These feature different levels of performance according to the type UAV to detect. Such technologies are out of the scope of this study, the interested reader can refer to (16) for detailed information.

We here focus on mitigation solutions. These can be divided into physical and non-physical systems, if they respectively impact or not on the physical integrity of the mUAV.

Existing non-physical systems have mainly focused on radio frequency (RF) and Global Navigation Satellite System (GNSS) signal jamming. These permit to disrupt the mUAV communications (22) and navigation system (21) respectively. While mainly developed in ground-based systems, jamming mitigation is also used by some air-based systems since the closer to the target the more effective the jamming is (19). Other non-physical solutions consider high power electromagnetic radiation (HPEM). Its purpose is to alter the electronics onboard a mUAV which will result in its crash (27). The last non-physical systems are lasers, which permit to neutralize the sensors of a mUAV or even destroy the mUAV itself if powerful enough (12).

When it comes to physical systems, which aim at neutralizing the mUAV, solutions range from projectiles to eagles, nets and UAVs. Counter-UAS projectiles include machine guns and missiles and are usually assisted by tracking systems like electro-optical sensors which might rely on AI. These are costly and the risk of collateral damage is high. The most known physical systems are nets, which permit to block the mUAV propeller(s) and thus stop its progression. Such nets can be thrown by guns/cannons (11) as part of ground-based solutions or directly launched from one UAV (9) or carried between a couple of UAVs (23) as part of an air-based solution. Such systems are well adapted to small mUAV and cause minimum collateral damage when nets are equipped with parachutes which make them more suitable in populated environments. However their range of action is limited. A few police forces have trained eagles to catch mUAVs (26). However this solutions remains marginal and potentially harmful for the birds For that reason, other solutions using drones as attackers (also referred to as collision UAVs) are now considered. Such UAVs are capable of flying at high speeds, are equipped with detection and tracking systems and can additionally carry explosives to increase the collision effect. Potential collateral damage is thus high.

All these solutions have considered countering a single malicious UAV. However, intruders can also be UAV swarms themselves (8). Since such enemy swarms are difficult to target and financially not worth being taken down, the U.S. military is investing on swarm-on-swarm warfare tactics in order to attack other enemy swarms (24). Such UAV defense swarms can be used for collapsing and trapping an enemy swarm. Collapsing is being done via communication jamming in order to disrupt the enemy swarm such that the individual drones get disintegrated and uncoordinated. The defense UAV system can also trap the enemy swarm to force it into a disadvantageous position such as an unfavorable area outside a critical zone. This strategy is similar to the approach presented in this paper.

2.2 Node Positioning Approaches

There exist some node positioning approaches to position UAV in the 3D-space which are related to our work. For example, Brust et al. (6) proposed VBCA, a virtual forces clustering algorithm, which imitates the VSEPR model (14) from molecular geometry for the arrangement of UAVs in a clustered swarm. The UAV's position is determined by the distance and role of its neighboring UAVs. VBCA assigns the role of a clusterhead to one UAV. This central UAV acts as a connector influencing the entire topology of the network geometry while individual UAVs are only affected by their direct neighbors. VBCA is maximizing the volume coverage, while maintaining *advanced connectivity* within the clustered UAV swarm.

Al-Turjman et al. (1) introduced a 3-D deployment strategy for relay nodes in WSNs (wireless sensor networks) for forestry applications maximizing network connectivity while maintaining a period of lifetime. The approach uses a two-layer hierarchical WSN architecture and determines the relay node positions on a 3-D grid to maximize connectivity with constraints on the available number of nodes and required lifetime.

Barnes et al. (4) addressed the problem of coordinating a swarm of Unmanned Ground Vehicles (UGV) with a UAV. The authors used a single UAV as a leader for the UGV swarm. The UGV formation is achieved by artificial potential fields. More specific swarm control–behaving according to a set constraints, formation, and member spacing was provided by the usage of limiting functions. A small configuration of three UGVs and one UAV was used to demonstrate their approach. In addition the scalability of this approach has not been validated or discussed.

Kim et al. (17) proposed a multiple UAV platform based on artificial potential function (similar to (4)), which guarantees collision avoidance. Their approach rely on optimization techniques to generate the optimal trajectory. Although the approach is demonstrated using a single constant point, the authors claim it can be applied for multiple target points.

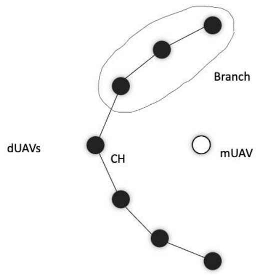


Fig. 2 The basic elements used in the development of the algorithms: Branch, cluster head (CH), intruder (mUAV) and defender (dUAV).

3 Problem Description and System Model

This section describes the notations and definitions as shown in Table 1 as well as the assumptions, properties, and communication model, which define the system model on which our proposed counter UAV system is based.

For this paper, we assume a malicious UAV (mUAV) has been detected in the flight zone and a number of defense UAVs (dUAVs) have been instructed to initiate the defense mission. The mUAV tends to escape when it detects the dUAVs (cf. Fig. 2).

Notation	**Definition**
$\|A\|$	number of elements in the set A
B	set of branches of a cluster head
β	enclosing angle
BM-A	Basic Message - Accept
BM-D	Basic Message - Discard
CH	Cluster Head
CM	Control Message
$\hat{d}$	normalization of vector $\vec{d}$ where $\hat{d} = \frac{\vec{d}}{\|\|\vec{d}\|\|}$
dUAV	defense UAV
ϵ_d	collision threshold of dUAV
ϵ_m	collision threshold of mUAV
Flight zone	restricted area/space
mUAV	malicious UAV
N	set of UAVs in the neighborhood
n_B	pre-defined number of branches of a CH
r_F	formation radius
$\|\|\vec{v}\|\|$	magnitude of $\vec{v}$
w_B	balanced clustering weight
w_K	KHOPCA weight

Table 1 Counter UAV system's notations and definitions

3.1 Assumptions

- The mUAV has already been detected by every dUAV.
- The mUAV has a slightly lower top speed than dUAVs in order to avoid static locking.
- Every UAV actively tries avoiding collisions with each other.
- mUAVs have a higher collision threshold than dUAVs.
- All UAVs have transmission, distance, relative positioning and absolute position sensing capabilities.
- A high-quality UAV monitoring system in place that is capable to detect and identify malicious UAVs.

3.2 Properties

- Every dUAV can have at most one parent and a child.
- CHs have no parent and can have up to n_B children.
- Every CH stores a n_B.
- A branch is a dUAV that has a CH as parent.
- The length of a branch x is defined by the number of dUAVs in parent-child relations starting from x. The length of x is denoted as $|x|$.
  - Example: CH$\rightarrow a \rightarrow b \rightarrow c \rightarrow d$. Then, $|a| = 4$.
- Every dUAV has a clustering weight w_B, which is initially set to 0.
- A leaf is the last dUAV in a branch, has the largest w_B in the branch and has no child.
- w_B of a dUAV is defined as its position on a branch.
  - Example: $CH \rightarrow a \rightarrow b \rightarrow c \rightarrow d$ with w_B assignment $(CH, 0), (a, 1), (b, 2), (c, 3), (d, 4)$
- The difference of the lengths of any two branches cannot exceed 1.

3.3 Communication Model

Every UAV is equipped with a network adapter that can be used to establish a communication channel between UAVs. The communication itself could be realized with infrastructure-less and self-configuring UAV Ad hoc Networks (UAANETs) (20) that are a subset of the well-known Mobile Ad hoc Network (MANET) paradigm. For the sake of simplicity, we assume that every UAV has the capability of periodically scanning the surroundings by using a *circular* transmission range. Furthermore, we assume a reliable communication channel.

4 Counter UAV Defense System: Approach

Our approach consists of a swarm of dUAVs that forms a three-dimensional cluster around the mUAV in such a way that the mUAV has a minimal set of movement possibilities. Hereby, we assume that the mUAV is trying to avoid collisions with dUAVs to maintain its functioning. By enclosing the mUAV, the the dUAVs are then able to escort the mUAV outside the flight zone.

The proposed approach follows a modular design, implementing four phases to realize the escort maneuver (task, problem), which are patrolling, detection, interception, and counter-attack.

The patrolling phase is simply explained by the defense UAVs surveying the flight zones rather passively until a detection of an anomaly triggers the next phase. The clustering and formation algorithms for detection and interception phases are executed simultaneously during the entire mission, whereas the transition between the interception and counter-attack phases are decided by the CH depending on the following conditions:

1. Chase phase to escort phase: The distance between the CH and the mUAV is lower than r_F.
2. Escort phase to chase phase: The distance between the CH and the mUAV is higher than two times r_F.

Details about the individual phases will be explained in further detail in the upcoming sections.

4.1 Detection

The clustering procedure is based on the KHOPCA clustering algorithm (7) (cf. Section 2) with the key difference that the structure of the cluster remains balanced. We use KHOPCA for three main reasons. Firstly, it provides a leader election algorithm that creates cluster heads, which is the entry point for our clustering algorithm. Secondly, KHOPCA does not require weights for the cluster head to be unique. Implementing a *simple* leader election would require such an assumption. Lastly, KHOPCA has been proven to be suitable for highly dynamic networks, such as the ones encountered in our problem scenario.

The cluster structure consists of the CH being in the middle of the cluster, acting as a coordinator of the whole cluster and a set of branches that originate from the CH.

The reason for maintaining a balanced structure is the formation. Our goal is to construct a clustering that is suitable for the desired formation that looks like a closed hemisphere where the CH tries to enclose its branches in order to catch the mUAV. Therefore, the branches should ideally have the same length for them to be balanced. We also introduced the notation of a branch since it simplifies the modeling of the formation by considering a sequence of interconnected dUAVs rather than single ones. The weighting constraint is defined as follows:

$$\forall b_i, b_j \in B \; : \; \mid b_i.length - b_j.length \mid \leq 1.$$

The weighting constraint states that the difference of the lengths of any two branches cannot exceed 1. In section 4.1.4 we illustrate how the re-balancing of the cluster works. Re-balancing is required due to unexpected connection losses. In section 4.2 we elaborate on the formation.

The clustering is done fully locally at each UAV. We can distinguish between the three different states that an UAV can be in: UAV, dUAV and CH. The difference between UAVs and dUAVs is that UAVs have no parent, hence are not in a cluster and are searching for a parent while dUAVs are cluster members that are being coordinated by the CH for performing the escort mission. Every other dUAV will adapt his weight according to its parent. The dUAVs with weight w_{B_i} are exactly w_{B_i} hops away from the CH. Note that the weight of the clustering is not the same as the weight that KHOPCA provides. We differentiate between w_B and w_K, where w_B is the weight of the balanced clustering algorithm and w_K is the one from the KHOPCA algorithm. We run both KHOPCA and our balanced clustering algorithm simultaneously. In the following, we elaborate on the different states of UAVs.

4.1.1 Behaviour: UAV

Initially every UAV is parent-less and scans the neighborhood for a parent. Every UAV does not accept children by default and is flying to the mean position of the neighborhood. This *flocking* ensures that UAVs nearby will be gathered together so that we can achieve bigger and fewer clusters. The flocking is described in the Algorithm 1 and the behavior of UAVs is described in the Algorithm 2.

Algorithm 1 Flocking algorithm

```
$sum \leftarrow \vec{0}$
for $n \in N$ do
    $sum \leftarrow sum + n.pos$
end for
$\mu_{pos} \leftarrow sum \,/\, |N|$
moveTo($\hat{\mu}_{pos}$)
```

Algorithm 2 Behaviour of UAVs (non-CH and parent-less)

```
DoFlocking($N$)
$parents \leftarrow \{n \in N \mid u.accept\}$
if $|parents| > 0$ then
    $duav$ = apply criterion to select $p \in parents$
    request connection with $duav$
end if
```

The UAVs scan the neighborhood for possible parents that accept children. If there exists more than one, we should consider to apply a criterion to choose one from $|parents|$. Our criterion is the minimal distance from the requesting UAV to the parent. Therefore, we sort the possible parent dUAVs in ascending order of distance. This enables a short communication channel and hence fewer potential connection losses. However, other criteria could be applied as well.

4.1.2 Behaviour: CH

As soon a UAV is elected as CH by the KHOPCA algorithm, it starts accepting children. CHs stop accepting further children if n_B is reached. CHs then inform the children that they can now start accepting an additional child. We distinguish between the following two message types:

1. **Basic Message**
   These messages are sent from the CH to its branches in order to trigger them into accepting an additional child (BM-A) or to discard (BM-D) the current child. Discarding a child at w_{B_r} of a branch r leads to the discarding of $|r| - w_{B_r} + 1$ dUAVs since the message will be passed recursively to all children.
2. **Control Message**
   CMs are recursively sent from dUAVs to the CH in order to notify about a new child.

CMs, as shown in Algorithm 3, have the purpose of knowing the length of the branches of a CH which is crucial for the balancing mechanism.

Algorithm 3 Behaviour of CHs

```
$accept \leftarrow true$
while $accept$ do
    if $|B| = n_B$ then
        send BM-A to all children
        $accept \leftarrow false$
    end if
end while
```

The algorithm runs until $|B| = n_B$. Then, CHs only act upon message receipt.

Upon receipt of a CM, CHs send a BM-A message to all its branches that have the minimal length among all branches. Hence, the leaves can start accepting a new child. This is how the CH ensures balancing. The balancing is a key feature that distinguishes our algorithm from KHOPCA. This behavior upon message receipt is described in the Algorithm 4.

4.1.3 Behaviour: dUAV

The dUAVs have a parent and hence are in a cluster. They wait for incoming messages from the CH and are ready for chasing and formation. The dUAVs

Algorithm 4 CH: upon receipt of a CM

```
$senderUAV \leftarrow sender\ of\ CM$
$b_s \leftarrow getBranch(senderUAV)$
if $b_s$ not null then
    $b_s.length \leftarrow |b_s| + 1$
    if $|B| = n_B$ then
        $min \leftarrow min(\{b \in B : |b|\})$
        for $b \in B$ do
            if $|b| = min$ then
                send BM-A to $b$
            end if
        end for
    end if
end if
```

that are leaves in a cluster might still accept children. Let d be a dUAV that has accepted a UAV c as a child. Then, the following steps are executed:

- c will join the cluster, hence become a dUAV
- $c.w_B \leftarrow d.w_B + 1$
- $d.accept = false$
- d will send a CM to its parent

Let d receive a CM. d will propagate the CM to his parent.

Let d receive a BM-A message. If d has a child, it will no longer accept children and propagate the BM-A message to its child. If d is a leaf, it will start accepting a child.

Let d receive a BM-D message. If d has a child, it will propagate the message to his child and discard the connection. d will take the state of a UAV, thus resetting its w_K, performing flocking and searching for a new parent.

4.1.4 Cluster Re-balancing

Due to connection losses, the cluster can lose its balance. Therefore, we implemented a self re-balancing mechanism that keeps the cluster balanced according to the weighting constraint. The re-balancing of a cluster is depicted in Fig. 3. Algorithm 5 shows the procedure of cluster re-balancing. Note that only CHs run the re-balancing algorithm.

Algorithm 5 Cluster Balancing

```
$min \leftarrow min(\{b \in B : |b|\})$
for $b \in B$ do
    if $|b| > min + 1$ then
        $b.removeChildAt(min + 2)$
        $b.length = min + 1$
    end if
end for
```

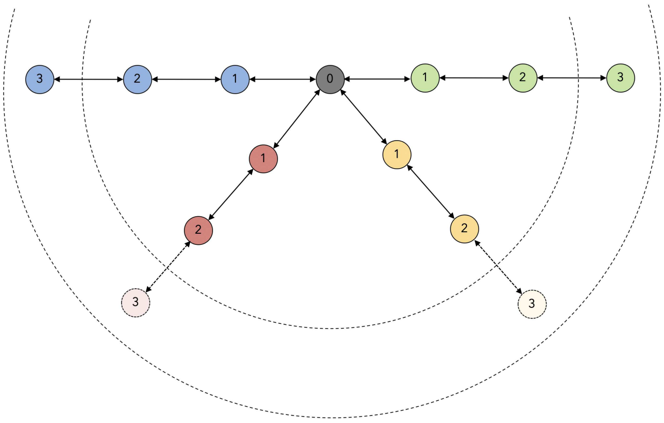


Fig. 3 Cluster balancing and re-balancing

Example: Suppose we have a cluster as shown in Fig. 3 and suppose there is a connection loss in the red branch between the dUAVs with the weights 1 and 2 respectively. It follows that the weighting constraint is violated since the difference of the lengths between the blue and red branches is 2. The re-balancing algorithm will run and search for the minimal branch length, which is $min = 1$ in this case. Iteratively, every branch b will be checked if $|b| > min + 1 = 2$. The green branch fulfills this condition, hence this branch will be ”cut” between the weight levels $min + 1 = 1 + 1 = 2$ and $min + 2 = 1 + 2 = 3$. This cutting of branches is done via BM-D messages. After the balancing of the cluster, the weighting constraint is respected again.

4.2 Interception Formation

With the aim of escorting in mind, the swarm of dUAVs chasing must ensure that the movement of the mUAV is restricted while the escort phase is under operation. The cluster of dUAVs must strive to restrict the mUAV to be able to only move in one direction. To achieve the following, a formation model must be realized. This model should be resistant to any disruptions caused by the mUAV. Following from the assumption that $\epsilon_m > \epsilon_d$, one constraint we need to ensure is that the distance between the dUAVs within the formation is not too high which could allow the mUAV to escape.

The formation shape chosen for this particular problem is an hemisphere. It is motivated by the fact that the objective is to enclose the mUAV within the cluster in order to simplify the escort mission. The formation begins to take place while approaching the mUAV and then encloses the mUAV when the cluster reaches a certain distance. The rest of this section focuses on how the formation calculations are being done.

4.2.1 Calculating the cluster formation radius

Our goal is to place dUAVs on a branch equidistant to each other according to ϵ_d in order to minimize the escape directions of the mUAV. Firstly, we need to determine the maximum length of the branches to derive r_F. Let $max = max(\{b \in B : b.length\})$ be this number. Suppose we inscribe a regular polygon into a circle. Then, the branch occupies only $\frac{1}{4}^{th}$ of an imaginary circle. Every member of the branch lies on the edges of this circle. Therefore, if we were to mirror the singular branch in 2D along the y-axis and then mirror the resultant along the x-axis, we would get a regular polygon with n sides. Here, $n = 4 \cdot max$. Any regular polygon can be inscribed within a circle. With this we can now find the r_F with the following formula

$$r_F = \frac{\epsilon_d}{2\sin(\frac{\pi}{4max})},$$

where r is the formation radius, a is the length of a side in the polygon which is equal to the ϵ_d.

4.2.2 Determining branch rotations

Now that we have r_F, we know how far from the cluster head the branches are going to exceed. However to determine the positions of each branch relative to the cluster head along the z-axis we need to rotate each point along the z-axis. To calculate the rotation positions we use the Rodrigues' rotation formula as follows

$$\vec{v_{rot}} = \vec{v}\cos\theta + (\vec{k} \times \vec{v})\sin\theta + \vec{k}(\vec{k}\cdot \vec{v})(1 - \cos\theta),$$

where $\vec{v}$ is the vector that needs to be rotated, $\vec{k}$ is the axis of rotation and θ is the angle by which the vector $\vec{v}$ needs to be rotated. Steps for calculating this rotation are listed below:

1. Calculate the angular separation theta between every branch. This is done by dividing 2π by the total number of branches.
2. From the origin of rotation, each branch is θ away from the previous branch. Let $b \in B$ be the current branch and i_b the current index of b. Then, b is an angle of $\theta_{rot} = \theta \cdot (i_b - 1)$ away from the origin.
3. Rotate b by θ_{rot} along the z-axis relative to the direction between the mUAV and the CH.
4. Following the Rodrigues rotation formula, the $\vec{v}$ which is the formation direction is a vector perpendicular to the direction heading from CH to the mUAV with a magnitude of ϵ_d.
5. The vector $\vec{v}$, once rotated along the axis of rotation $\vec{k}$, will yield the branch positions along the z-axis. Note that at this rotation step, the whole branch will not be in its correct position.

6. The axis of rotation $\vec{k}$ is the direction that is represented by tracing a vector from the CH to the mUAV and normalizing it so that a unit vector is obtained. The values $\vec{v}$ and $\vec{k}$ along with the rotation angle θ_{rot} for the concerned specific branch is substituted in the Rodrigues rotation formula to yield the positions for every branch.
7. Every branch member b_i occupies an angle of their corresponding θ_{rot_i} from the origin.
8. With the fractional angle, the actual positions of each branch member can be calculated. The x component is decomposed to be in the branch parent and tracing the fractional angle over the cluster radius to derive the magnitude. The z-axis is in the direction of the mUAV. This decomposition is shown in Fig 5.

   The rotation procedure run by the CHs is described in Algorithm 6.

Algorithm 6 Rotation Positions

```
$i = 0$
for $b \in B$ do
    $\theta \leftarrow \frac{2\pi}{n_B}$
    $\theta_{rot} \leftarrow i \cdot \theta$
    $\vec{d} \leftarrow mUAV.pos - b.pos$
    $\vec{v} \leftarrow \vec{d} \cdot \epsilon_d$
    $\vec{k} \leftarrow \hat{d}$
    $\vec{v} \leftarrow \vec{v}$ rodrigues rotation $(\vec{v}, \vec{k}, \theta_{rot})$
    rotate $b$ by $\vec{v}$
    $\theta_{frac} \leftarrow \frac{\beta}{n_B}$
    $b$.sendRotationMessage($\theta_{frac}$, $\vec{d}$, $\vec{v}$, $r_F$, $pos$)
    $i \leftarrow i + 1$
end for
```

Let b be a dUAV receiving a rotation message. Let c be the child of b. Then, b will run Algorithm 7.

Algorithm 7 Children rotation positions

```
procedure doRotation($\theta_{frac}$, $\vec{d}$, $\vec{v}$, $r_F$, $t$)
    $\alpha_b \leftarrow \theta_{frac} \cdot w_{B_b}$
    $x \leftarrow r_F \cdot \cos(\alpha_b)$
    $z \leftarrow r_F \cdot \sin(\alpha_b)$
    $x' \leftarrow \hat{v} \cdot x$
    $z' \leftarrow \hat{d} \cdot z$
    $t_{new} \leftarrow t - \vec{d}$
    moveTo($\hat{t}_{new}$)
    send rotation message to $c$ with $(\theta_{frac}, \vec{d}, \vec{v}, r_F, pos)$
end procedure
```

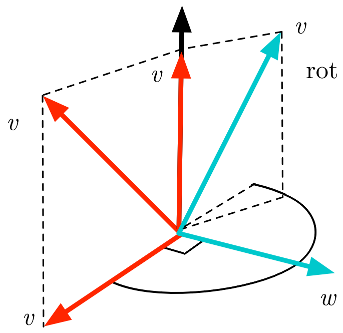


Fig. 4 Rotating a vector $\vec{v}$ by θ along $\vec{w}$

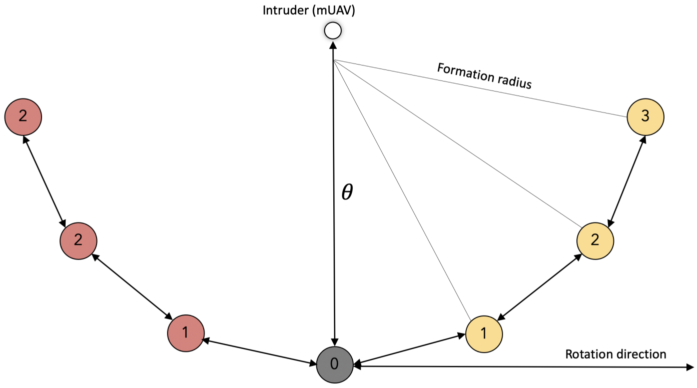


Fig. 5 Branch children placement

4.3 Interception Mobility

As part of our C-UAS system, a UAV Monitoring and Interception System (UMIS) permits to detect and identify the approximate location of the mUAV in the restricted area. More precisely, if the UMIS detects a mUAV, it will trigger the UAV defense system, the dUAVs deployment and then, initiate the creation/generation of the UAV defense swarm.

We here consider a UMIS running on-board each dUAV, in a distributed manner, to support autonomous decision making and to prevent a single point of failure.

In the context of the proposed C-UAS system, the goal of the mUAVs is to intercept the mUAV. The optimization criteria is hereby to minimize the time between detection and interception.

One strategy to improve the interception process consists in predicting the future position of the mUAV. For this, two positions at different timestamps can be compared to each other, forming a movement vector which then can

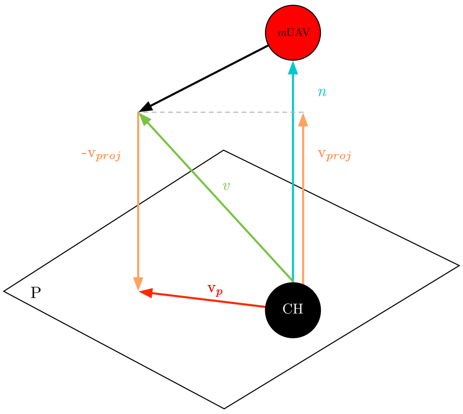


Fig. 6 mUAV Movement Prediction

be multiplied by a certain factor in order to obtain the next predicted mUAV position. If needed, multiple points with weight distributions can be used to increase the accuracy of the prediction.

As depicted in Fig. 6, the goal is to project the vector v onto the plane P, defined by the normal vector n. The resulting vector v_p is then added on top of the main heading direction n. The exact procedure is shown in Algorithm 8. Note that $\alpha_{x,y}$ represents the angle between x and y.

Algorithm 8 Chase procedure of CHs including mUAV Movement Prediction

```
$\vec{n} \leftarrow m\text{UAV}.pos - pos$
$\vec{v} \leftarrow m\text{UAV}_{predicted} - pos$
$\vec{v}_{proj} \leftarrow \vec{n}.setMag(\cos(\alpha_{\vec{n},\vec{v}}) \cdot \|\vec{v}\|)$
$\vec{v}_p \leftarrow \vec{v} - \vec{v}_{proj}$
$applyForce(\vec{v}_p, w_0)$
$applyForce(\vec{n}, w_1)$
```

In practice, detecting the position of the mUAV might depend on the dUAVs sensors and the precision of the detection depends on the distance between the dUAV and mUAV.

For modeling purposes, we assume that the dUAV has several types of sensors, which are complementary in sensing different information about the location of the mUAV. The sensor reading depends on the distance between dUAV and mUAV. Incorporating the sensor configuration and information in a interception strategies model, we introduce different zones or regions surrounding the dUAV. In each regions, the dUAV can perform different motions in order to increase its performance. According to the sensing capabilities, we split the regions surrounding the dUAV to three classes which are Line of Field View (LOFV), Line of Sight (LOS), and Line of Control (LOC).

We have three assumptions upon which we mainly rely on for the response design:

1. When mUAV is inside the LOFV, we assume that dUAV can deduce the distance between dUAV and mUAV, and an angle which forms a spherical sector within which the mUAV could possibly reside
2. When mUAV is inside the LOS, we assume that dUAV can deduce the exact location of mUAV
3. When B enters inside LOC, dUAV can catch mUAV

Line of Field View (LOFV): The LOFV distance forms a sphere around the dUAV with dLOFV as radius. This region between LOFV and LOS is called the field view region. If the distance between dUAV and mUAV is reduced below dLOFV, according to our assumptions 1, dUAV can deduce both the distance and the angle of the intruder mUAV.

Line of Sight (LOS): The dLOS is the LOS distance, and this forms a sphere around dUAV with dLOS as radius. This region between LOS and LOC is called clear sight region. If the distance between dUAV and mUAV is reduced below dLOS, it means that dUAV falls inside the LOS region. According to our assumption 2, dUAV can deduce the exact location of mUAV.

Line of Control (LOC): The dLOC is the LOC distance, and this forms a sphere around dUAV with dLOC as radius. This region of the sphere is called the control region. When the distance between dUAV and mUAV is reduced below dLOC, the dUAV is assume to have mUAV intercepted (assumption 3) and the enclosement angle of the cluster formation is enlarged, thus trapping the mUAV inside the resulting spherical structure and triggering the escort phase.

Based on this model, strategies for the dUAV can be proposed depending on mUAV regional location. Future work will be dedicated in testing mobility strategies for dUAVs.

4.4 Counter-Attack

The counter-attack consists - in the context of this paper - of an escorting procedure. This phase is aims to bring the previously trapped mUAV outside the regular flight zone. According to our assumptions, the mUAV will try to avoid any collision with nearby UAVs, and thus will be forced to move with them as shown in Fig. 12. Optionally, the branches could actively perform anti-escaping maneuvers in order to avoid loosing the mUAV due to larger holes in the formation.

During the process, the CH is in charge of the heading, while its branches maintain their relative positions to the CH. Usually, the shortest path to the flight zone border is taken. If needed, this can be freely adjusted depending on the end goal of the mission.

5 Prototypical Implementation

As a proof of concept, we developed a tool that simulates the whole process of the escort mission. Initially, the simulator creates a set of UAVs spread through a flight zone and a mUAV as shown in Fig. 7.

5.1 Wobbling

Every UAV is able to slightly deviate within a given radius from its anchor point. For this, they continuously generate random Perlin Noise[1] values for their three movement axes. The resulting pseudo-random movement is supposed to represent the real-world floating instability of UAVs, e.g. windy weather conditions. In the case of the mUAV, the wobbling can be used in order to simulate a spontaneous and unpredictable movement, making the chase and escort phases less trivial and thus resulting in a more realistic scenario.

5.2 Separation

Let u_1 and u_2 be two UAVs. Let $d = \|u_1.pos - u_2.pos\|$ be the distance between u_1 and u_2. Then, if $d < \epsilon_d$, a force vector parallel to d and of amplitude $\epsilon_d - d$ is applied to $u_{1,2}$, resulting in a separation. When more UAVs are involved, the sum of all produced vectors is applied. If desired, a constant c can be added to each force vector in order to push the UAVs even further apart, making them less likely to stay at the exact borders of the threshold radius. Note that the defense UAVs and the malicious UAV work similarly in terms of collision avoidance principles, that is, they can only move in a given direction if there is no other UAV.

5.3 Cumulative force movement logic

Multiple forces of different origins may act on some UAV at the same time. For instance, a UAV may at the same time try to head towards a certain direction and actively try to avoid a collision with another UAV. A force can be described as a directional vector v and a weight w. At the end of an update cycle, all executed forces are added together, with respect of their weights, to form a cumulative force v_{sum}. The amplitude of v_{sum} cannot exceed the maximal velocity of the UAV. The wobbling effect of a UAV is the only movement component that is not translated into force as it is not produced by the UAV itself but by its environment (e.g. wind), implying that the combination of v_{sum} and the produced wobbling movement may exceed the maximal velocity of that UAV.

[1] Perlin Noise: https://mzucker.github.io/html/perlin-noise-math-faq.html

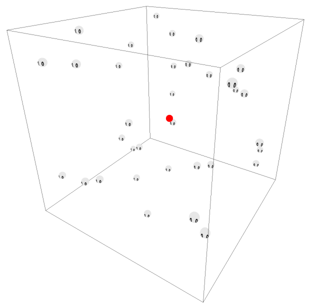


Fig. 7 Initial view

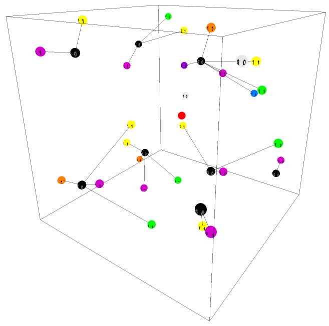
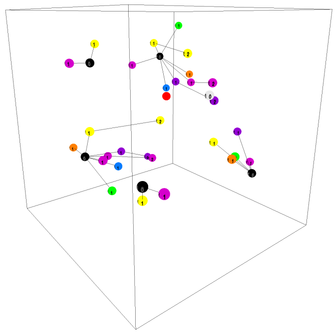
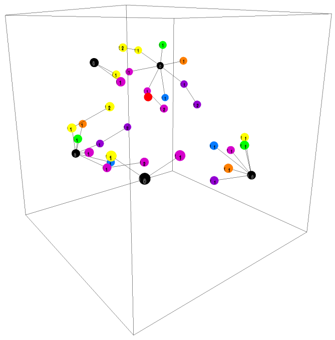


Fig. 8 Clustering view



Formation (a)



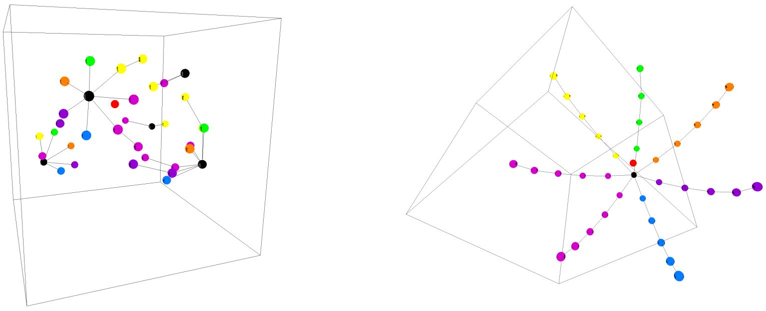
Formation (b)

5.4 Simulator GUI

Initially all UAVs are randomly distributed within the flight zone. The malicious UAV is represented by the red sphere in the middle of the flight zone. Black spheres represent CHs which are elected by KHOPCA.

Observe that the cluster branches tend to direct themselves towards the mUAV, preparing for enclosing the mUAV in the next steps.

The last Fig. 9(d) shows a single cluster as opposed to the other figures. This is due to a merging mechanism between several clusters which is triggered when at least two CH are within their communication range. If n CHs are within a common communication range, then $n-1$ CHs will be degraded to UAVs by KHOPCA which will lead to the destruction of the individual clusters. This enables a new construction of a bigger cluster.



Formation (c)

Formation (d)

Fig. 9 Formation - View

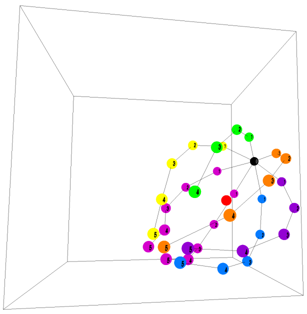


Fig. 10 Chase View - Enclosure 1

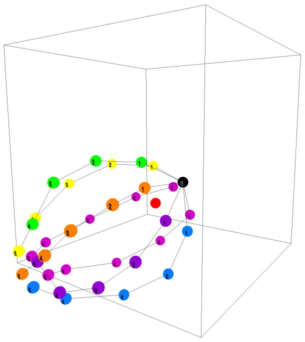


Fig. 11 Chase View - Enclosure 2

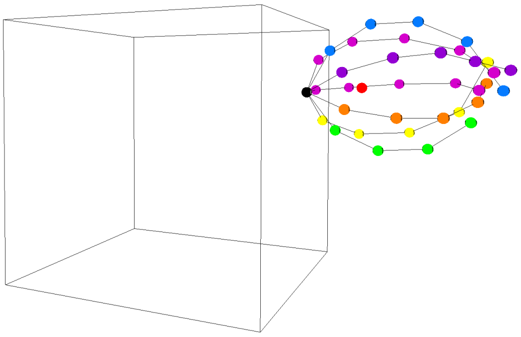


Fig. 12 Escort - View

Flight zone dimensions	$500 \times 500 \times 500$
Number of dUAVs	20
Communication Range	100
dUAV Wobbling Radius	50
mUAV Wobbling Radius	150
dUAV Collision Threshold	40
mUAV Collision Threshold	60
Number of Branches	3
Angular separation θ	$\pi/2$
UAV speed	0.8
UAV radius	10

6 Simulation Study

6.1 Simulation Setup

The parameters used for the simulation are communication range, number of branches, wobbling radius, collision threshold, UAV speed, UAV size, UAV deployment (positions), flightzone design (shape and size), number of dUAVs, and the formation shape. For all simulations, we consider the existent of one mUAV in the center of the flight zone, whereas dUAVs are uniformly positioned within an outer flight zone side.

For each experiment, 100 independent simulation runs have been conducted to ensure a statistical significance. If one simulation exceeds a simulation time of 10000 steps, it is aborted and considered as failed.

6.2 Metrics

Escort time. The performance of the experiments can be measured by the time needed to successfully escort the mUAV outside the flight zone. This time is measured starting from the very beginning of the simulation and should be minimized.

6.3 Experiments and Results

By performing the Anderson-Darling normality test (Fig. 13) over a sample of 100 simulations, we get a `p-value` of 0.247, thus giving us an evidence that the data follows a normal distribution.

In Fig. 14, one can observe that the impact of the number of dUAVs on the simulation times is more or less negligible. However, the number of clusterless dUAVs increases proportional to the total amount of dUAVs on the flight zone. Since these UAVs are not contributing to the escort mission, resources could be saved by deploying fewer dUAVs.

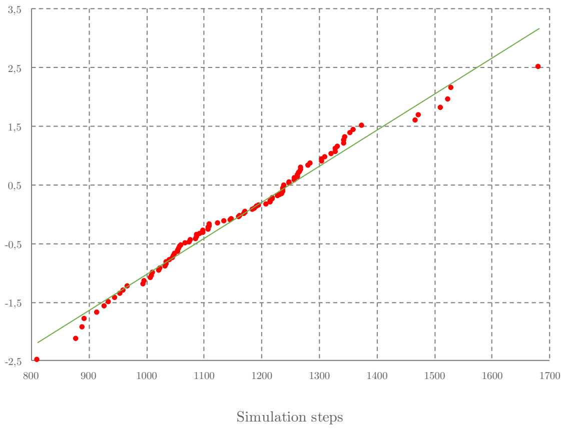


Fig. 13 Normality test

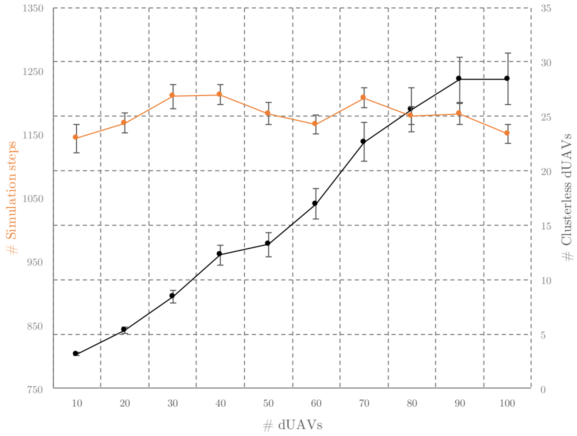


Fig. 14 Dependency between number of dUAVs and simulation times

The performance of the simulation noticeably improves when choosing a higher communication range between dUAVs until reaching a point of diminishing returns like depicted on Fig 15. As expected, the number of clusterless dUAVs decreases with the increased communication range as every UAV is able to locate its neighbors.

From Fig. 16 and 17, it is clear that the wobbling of UAVs has an negative impact on the simulation times and therefore should be minimized as much as possible.

Fig. 18 shows that the simulation fails when choosing a dUAV collision threshold over 60. Similarly, the same also happens for the collision threshold of mUAV when below 40 (cf. Fig. 19). When looking at the initial configuration, we see that the collision threshold for dUAV and mUAV has been chosen to be

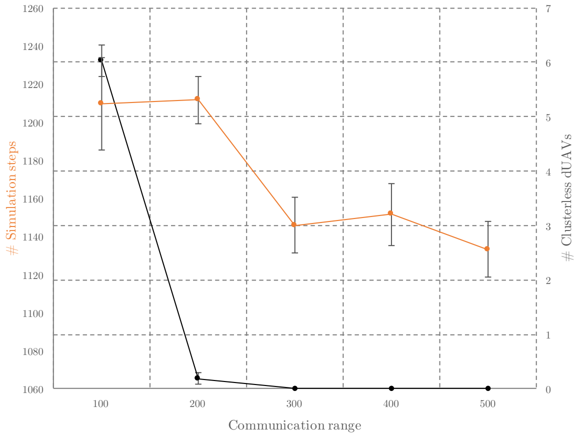


Fig. 15 Dependency between communication range and simulation times

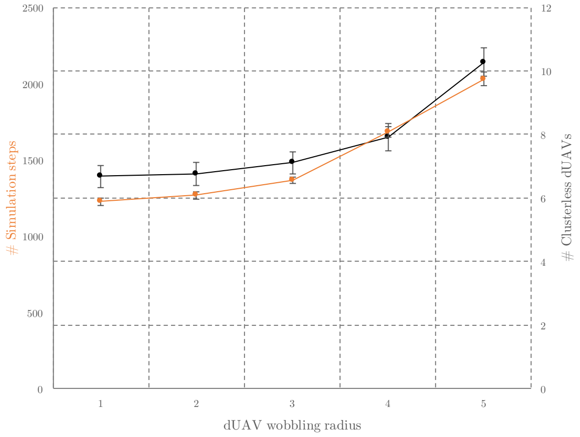


Fig. 16 Dependency between dUAV wobbling radius and simulation times

at 40 and 60 respectively. In fact, as soon as the value of the mUAV surpasses the one of the mUAV, the simulation fails since the dUAVs are unable to push the mUAV outside its current position.

Finally, Fig. 19 shows that starting from 2, the number of branches does not have a significant impact on the simulation times for the specified initial configuration.

6.4 Discussions

One should note that while the experiments show a certain behaviour of the simulation while changing different parameters one by one, there may be de-

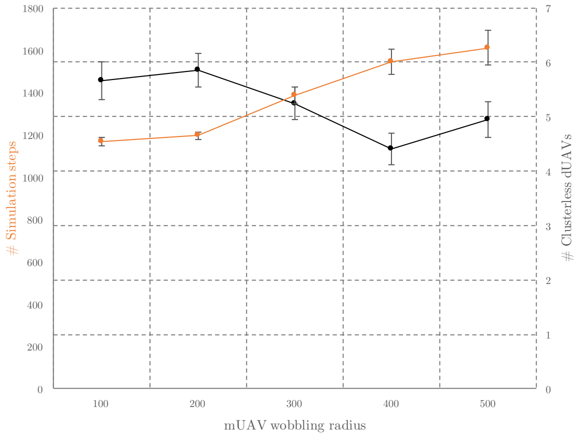


Fig. 17 Dependency between mUAV wobbling radius and simulation times

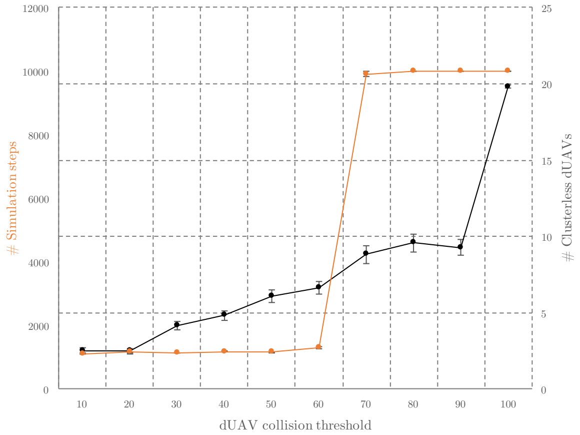


Fig. 18 Dependency between dUAV collision threshold and simulation times

pendencies between the parameters themselves, meaning that those results may vary depending on the initial configuration choice. Some additional experiments would be required in order to further understand the impact of every individual parameter.

Flocking is a great mechanism to realize a natural swarm behavior and to gather nearby clusterless UAVs in order to form a bigger cluster. At the moment the flocking is realized by moving to the center position of the nearby UAVs. This leads sometimes to unexpected behavior such as preventing the cluster from chasing the mUAV by blocking it. This is because of the UAVs that try to move to the CH which is located in the center of a cluster, while only the outer leaves of certain branches are accepting additional children.

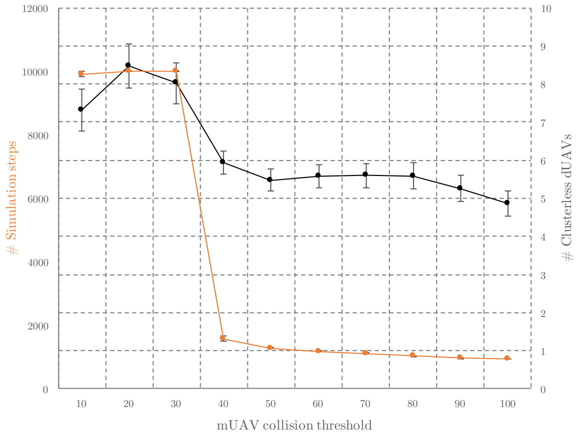


Fig. 19 Dependency between mUAV collision threshold and simulation times

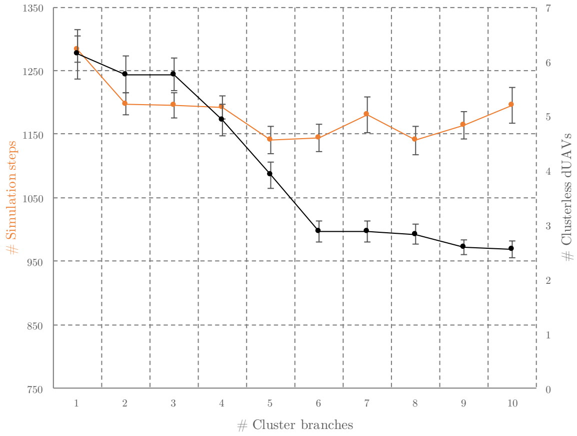


Fig. 20 Dependency between number of branches and simulation times

The merging of clusters is being realized only when the CHs are within a communication range. Several cluster chasing the same mUAV might slow down the escort mission due to hindering each other. One might consider merging the clusters as soon as possible. This would require an inter-cluster communication over branch nodes and leaves.

Our approach can be tested and augmented to successfully escort multiple independent mUAVs or a swarm of mUAVs with its own behavior out of the flight zone.

Concerning the problem with the flocking that is blocking the movement of the clusters, we need a communication mechanism that tells the UAVs where a branch leaf is located that accepts children. Thus, the UAVs can try to move to that position rather than blocking the whole cluster from moving. We plan to solve this issue in the future.

Inter-cluster communication can be realized to improve the cluster merging such that an early merge can be performed to prevent a hindering between several clusters. This requires special message types that are sent from branch nodes or leaves up to the CH to inform about a nearby cluster. Thus the CHs can agree upon merging which can be realized with a cluster destruction and flocking combination.

7 Conclusions

UAV-based autonomous defense systems consisting of collaboratively working dUAVs as a swarm will gain increasing attention, since a manually conducted defense is bound to fail due to the high dynamic involved in UAV maneuvering.

The most outstanding features and contributions of the presented approach are the balanced clustering to realize the intercept- and capture-formation. Additionally, the approach consists of a modular design containing the phases deployment, clustering, formation, chase, escort. All parts of the approach are fully localized, and the resulting networked defense UAV swarm is resilient against communication losses.

Acknowledgements This work is partially funded by the joint research programme UL/SnT-ILNAS on Digital Trust for Smart-ICT.

References

1. Al-Turjman, F.M., Hassanein, H.S., Ibnkahla, M.A.: Connectivity optimization with realistic lifetime constraints for node placement in environmental monitoring. In: 2009 IEEE 34th Conference on Local Computer Networks, pp. 617–624 (2009). DOI 10.1109/LCN.2009.5355140
2. Altawy, R., Youssef, A.M.: Security, privacy, and safety aspects of civilian drones: A survey. ACM Trans. Cyber-Phys. Syst. **1**(2) (2016). DOI 10.1145/3001836. URL https://doi.org/10.1145/3001836
3. Aweiss, A., Homola, J., Rios, J., Jung, J., Johnson, M., Mercer, J., Modi, H., Torres, E., Ishihara, A.: Flight demonstration of unmanned aircraft system (uas) traffic management (utm) at technical capability level 3. In: 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), pp. 1–7 (2019). DOI 10.1109/DASC43569.2019.9081718
4. Barnes, L., Garcia, R., Fields, M., Valavanis, K.: Swarm formation control utilizing ground and aerial unmanned systems. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4205–4205 (2008). DOI 10.1109/IROS.2008.4651260
5. Birch, G.C., Woo, B.L.: Counter unmanned aerial systems testing: Evaluation of vis swir mwir and lwir passive imagers. Tech. rep., Sandia National Laboratory, USA (2017). DOI 10.2172/1342469
6. Brust, M.R., Akbas, M.I., Turgut, D.: Vbca: A virtual forces clustering algorithm for autonomous aerial drone systems. In: IEEE SysCon (2016)
7. Brust, M.R., Frey, H., Rothkugel, S.: Dynamic multi-hop clustering for mobile hybrid wireless networks. In: Proceedings of the 2nd international conference on Ubiquitous information management and communication, pp. 130–135. ACM (2008)

8. CNBC: A swarm of armed drones attacked a Russian military base in Syria (accessed November 14, 2020). https://www.cnbc.com/2018/01/11/swarm-of-armed-diy-drones-attacks-russian-military-base-in-syria.html
9. Delft Dynamics: Drone Catcher (accessed November 14, 2020). https://dronecatcher.nl/
10. Ding, G., Wu, Q., Zhang, L., Lin, Y., Tsiftsis, T.A., Yao, Y.: An amateur drone surveillance system based on the cognitive internet of things. IEEE Communications Magazine **56**(1), 29–35 (2018). DOI 10.1109/MCOM.2017.1700452
11. Droptec: The Dropster Net gun (accessed November 14, 2020). https://www.droptec.ch/product
12. Extance, A.: Military technology: Laser weapons get real. Nature **521**, 408–410 (2015). DOI 10.1038/521408a
13. Gabriel C., B., John C., G., Matthew K., E.: Uas detection, classification, and neutralization: Market survey 2015. Tech. rep., Sandia National Laboratories, USA (2016)
14. Gillespie, R.: Fifty years of the vsepr model. Coordination Chemistry Reviews **252**(12-14), 1315–1327 (2008)
15. Guvenc, I., Koohifar, F., Singh, S., Sichitiu, M.L., Matolak, D.: Detection, tracking, and interdiction for amateur drones. IEEE Communications Magazine **56**(4), 75–81 (2018). DOI 10.1109/MCOM.2018.1700455
16. Kang, H., Joung, J., Kim, J., Kang, J., Cho, Y.S.: Protect your sky: A survey of counter unmanned aerial vehicle systems. IEEE Access **8**, 168671–168710 (2020). DOI 10.1109/ACCESS.2020.3023473
17. Kim, H., Ahn, H.: Realization of swarm formation flying and optimal trajectory generation for multi-drone performance show. In: 2016 IEEE/SICE International Symposium on System Integration (SII), pp. 850–855 (2016). DOI 10.1109/SII.2016.7844106
18. Lappas, V., Zoumponos, G., Kostopoulos, V., Shin, H., Tsourdos, A., Tantarini, M., Shmoko, D., Munoz, J., Amoratis, N., Maragkakis, A., Machairas, T., Trifas, A.: Eurodrone, a european utm testbed for u-space. In: 2020 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 1766–1774 (2020). DOI 10.1109/ICUAS48674.2020.9214020
19. Li, A., Wu, Q., Zhang, R.: Uav-enabled cooperative jamming for improving secrecy of ground wiretap channel. IEEE Wireless Communications Letters **8**(1), 181–184 (2019). DOI 10.1109/LWC.2018.2865774
20. Maxa, J.A., Mahmoud, M.S.B., Larrieu, N.: Survey on uaanet routing protocols and network security challenges. Ad Hoc & Sensor Wireless Networks (2017)
21. Noh, J., Kwon, Y., Son, Y., Shin, H., Kim, D., Choi, J., Kim, Y.: Tractor beam: Safe-hijacking of consumer drones with adaptive gps spoofing. ACM Trans. Priv. Secur. **22**(2) (2019). DOI 10.1145/3309735. URL https://doi.org/10.1145/3309735
22. Pärlin, K., Alam, M.M., Le Moullec, Y.: Jamming of uav remote control systems using software defined radio. In: 2018 International Conference on Military Communications and Information Systems (ICMCIS), pp. 1–6 (2018). DOI 10.1109/ICMCIS.2018.8398711
23. Rothe, J., Strohmeier, M., Montenegro, S.: A concept for catching drones with a net carried by cooperative uavs. In: 2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), pp. 126–132 (2019). DOI 10.1109/SSRR.2019.8848973
24. Scharre, P.: Counter-swarm: A guide to defeating robotic swarms - war on the rocks (2017). URL https://warontherocks.com/2015/03/counter-swarm-a-guide-to-defeating-robotic-swarms/
25. Shi, X., Yang, C., Xie, W., Liang, C., Shi, Z., Chen, J.: Anti-drone system with multiple surveillance technologies: Architecture, implementation, and challenges. IEEE Communications Magazine **56**(4), 68–74 (2018). DOI 10.1109/MCOM.2018.1700430
26. The New Indian Express: Now, eagles to take down illegal drones in Telangana (accessed November 14, 2020). https://www.newindianexpress.com/states/telangana/2020/aug/01/now-eagles-to-take-down-illegal-drones-in-telangana-2177572.html
27. Zohuri, B.: High-Power Microwave Energy as Weapon, pp. 269–308. Springer International Publishing, Cham (2019). DOI 10.1007/978-3-030-20794-$6_4. URLhttps://doi.org/10.1007/978-3-030-20794-6_4$

Figures

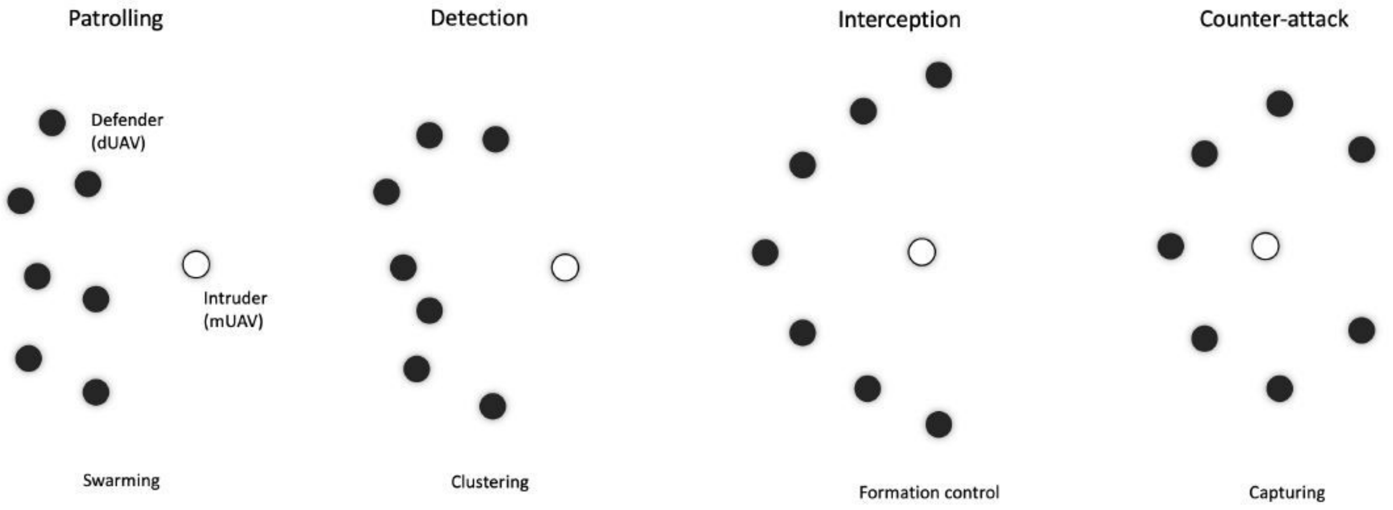


Figure 1

Illustration of the self-organizing approach of the defense swarm from patrolling to the counter-attack.

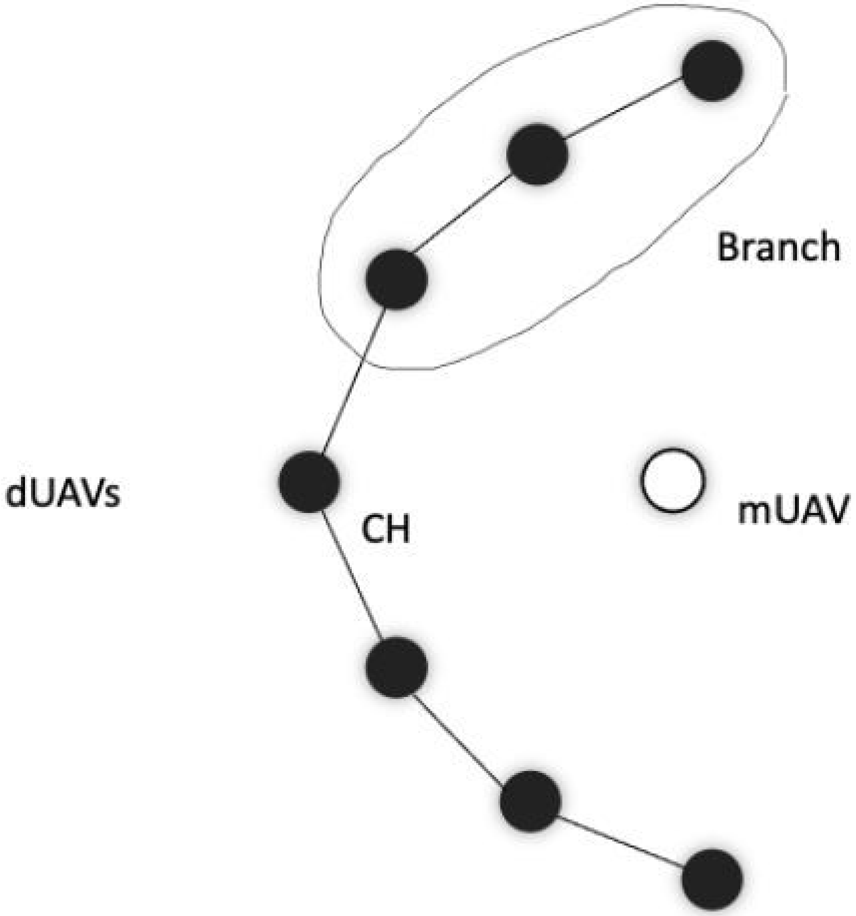


Figure 2

The basic elements used in the development of the algorithms: Branch, cluster head (CH), intruder (mUAV) and defender (dUAV).

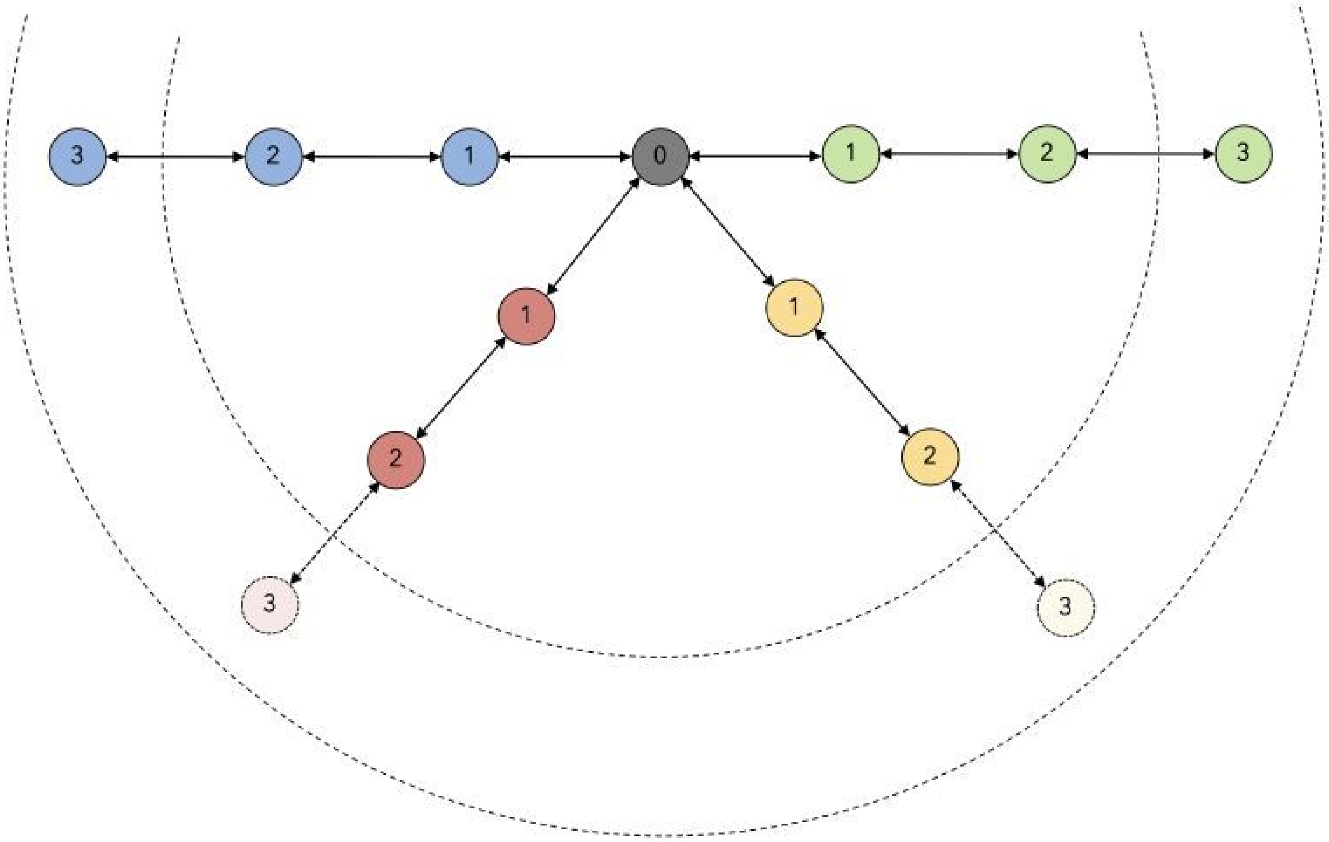


Figure 3

Cluster balancing and re-balancing

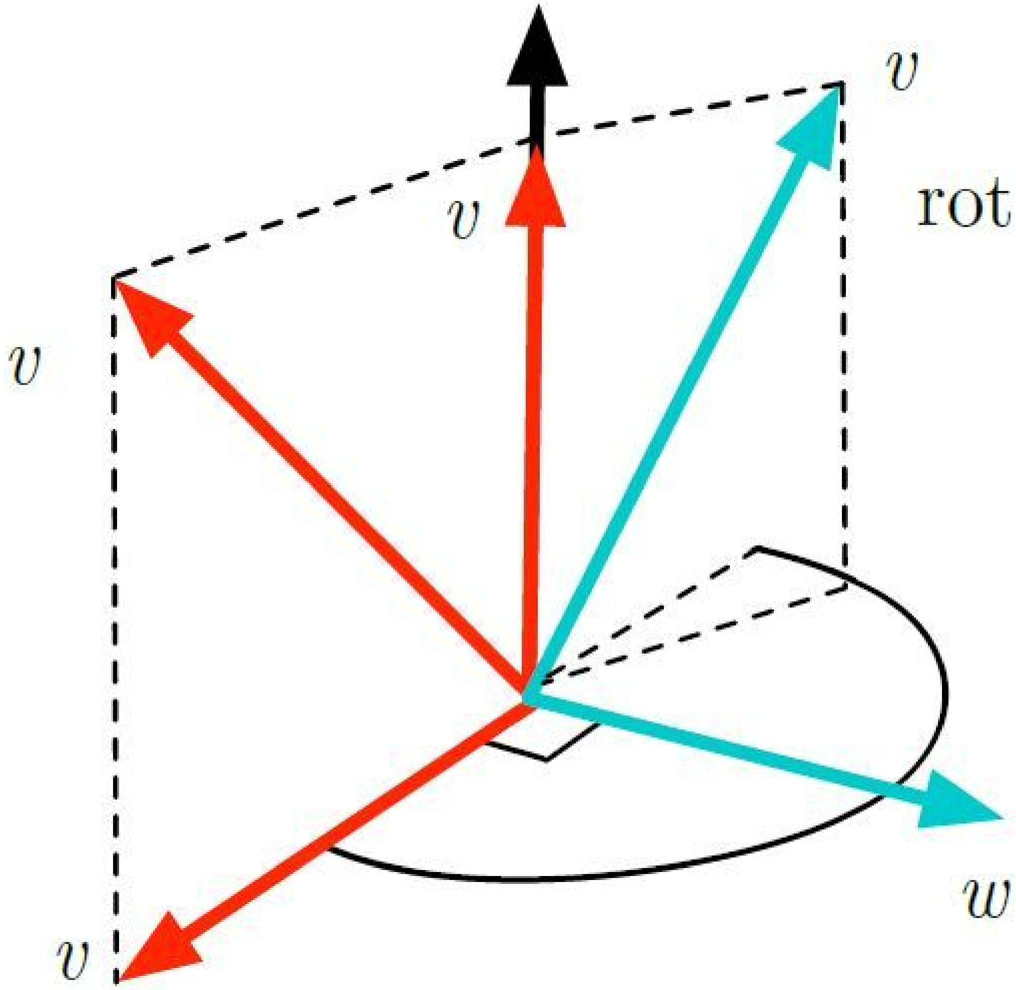


Figure 4

Rotating a vector v by θ along w

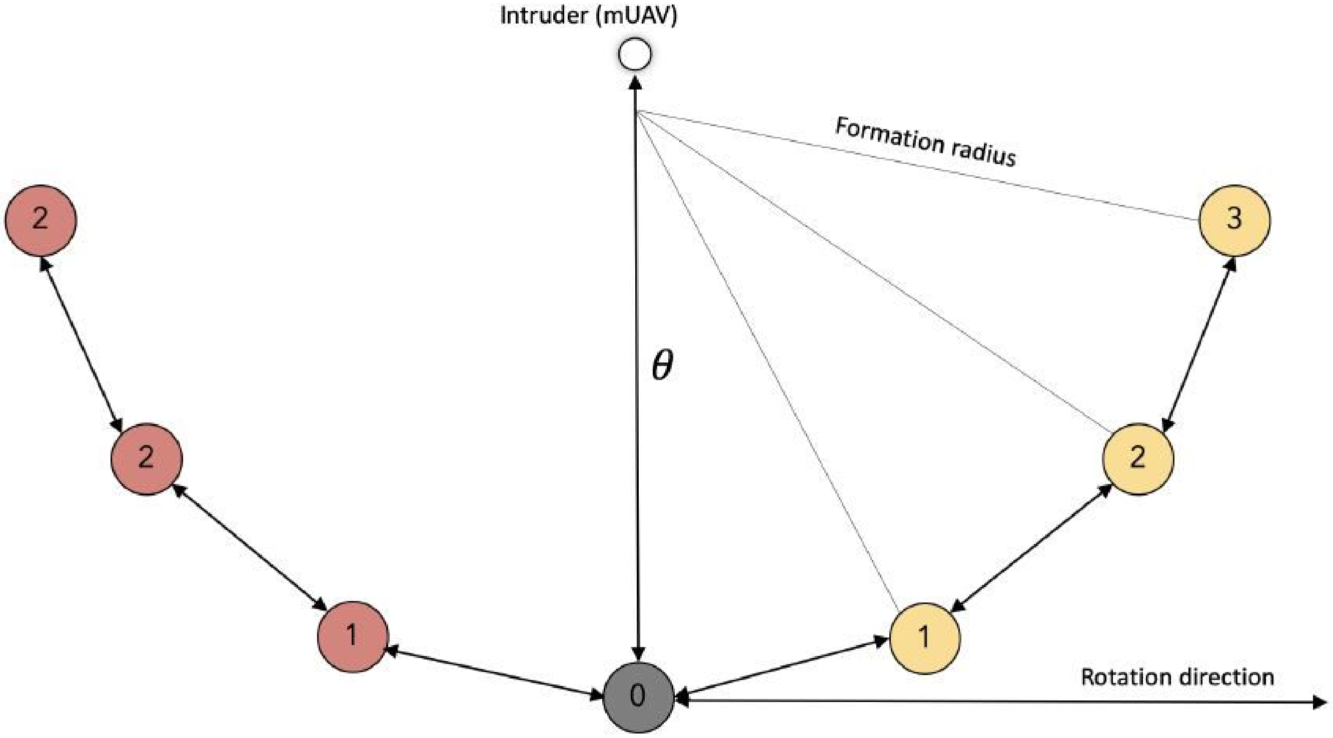


Figure 5

mUAV Movement Prediction

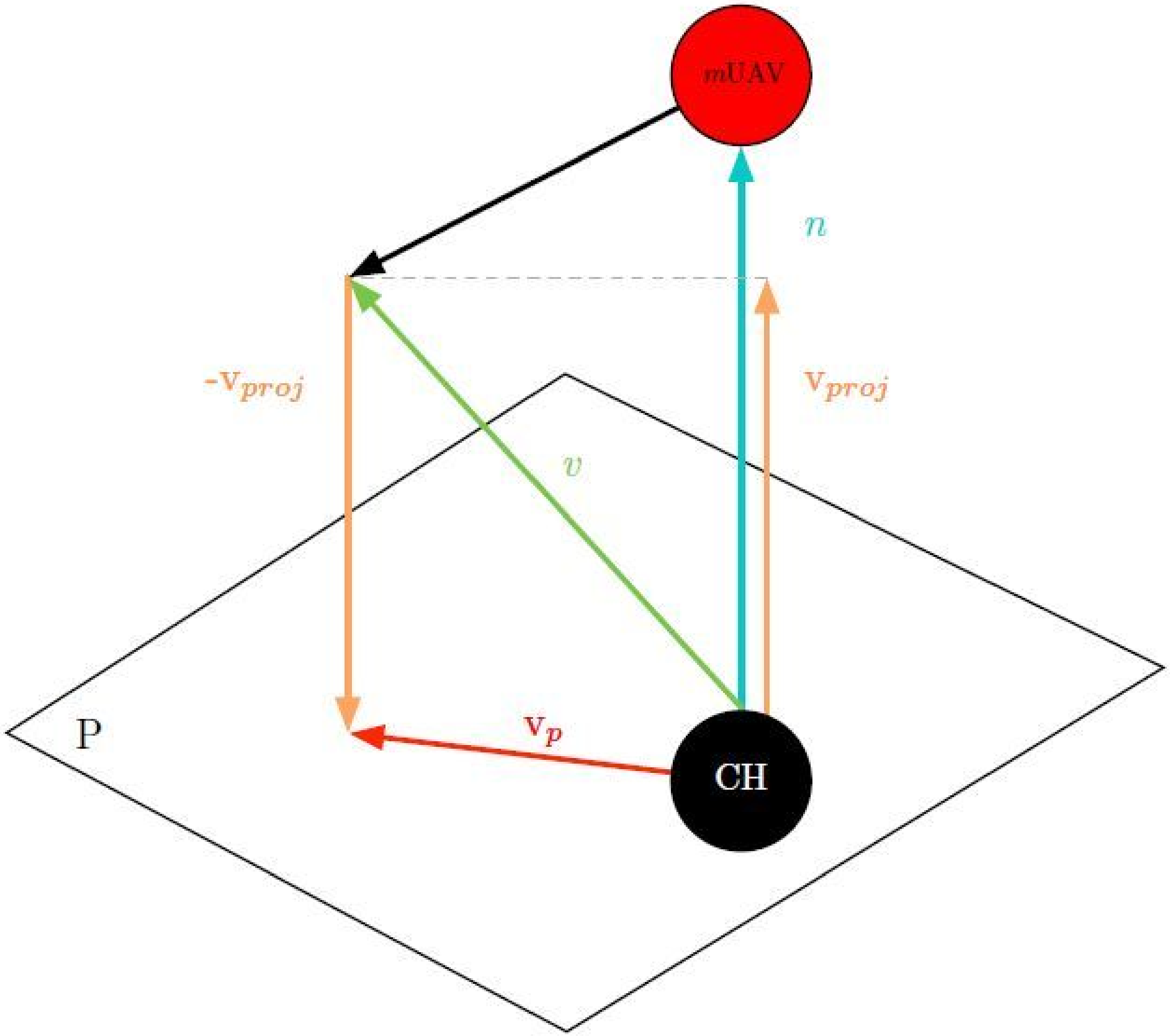


Figure 6

mUAV Movement Prediction

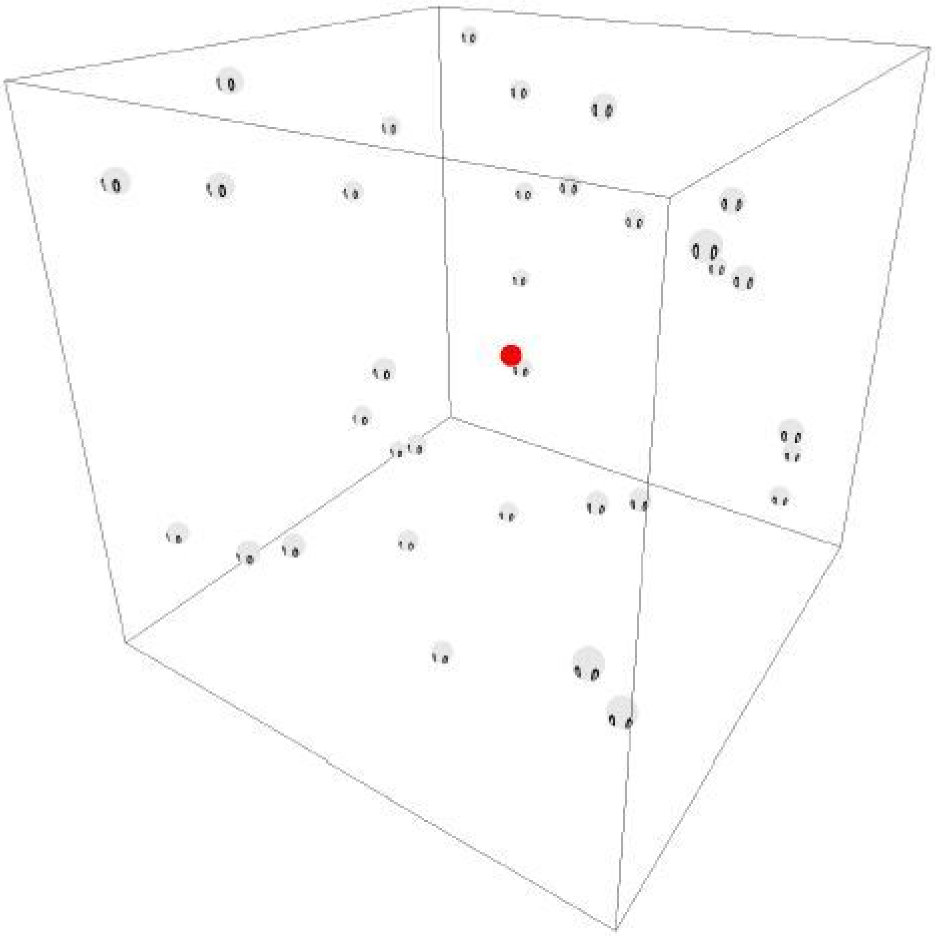


Figure 7

Initial view

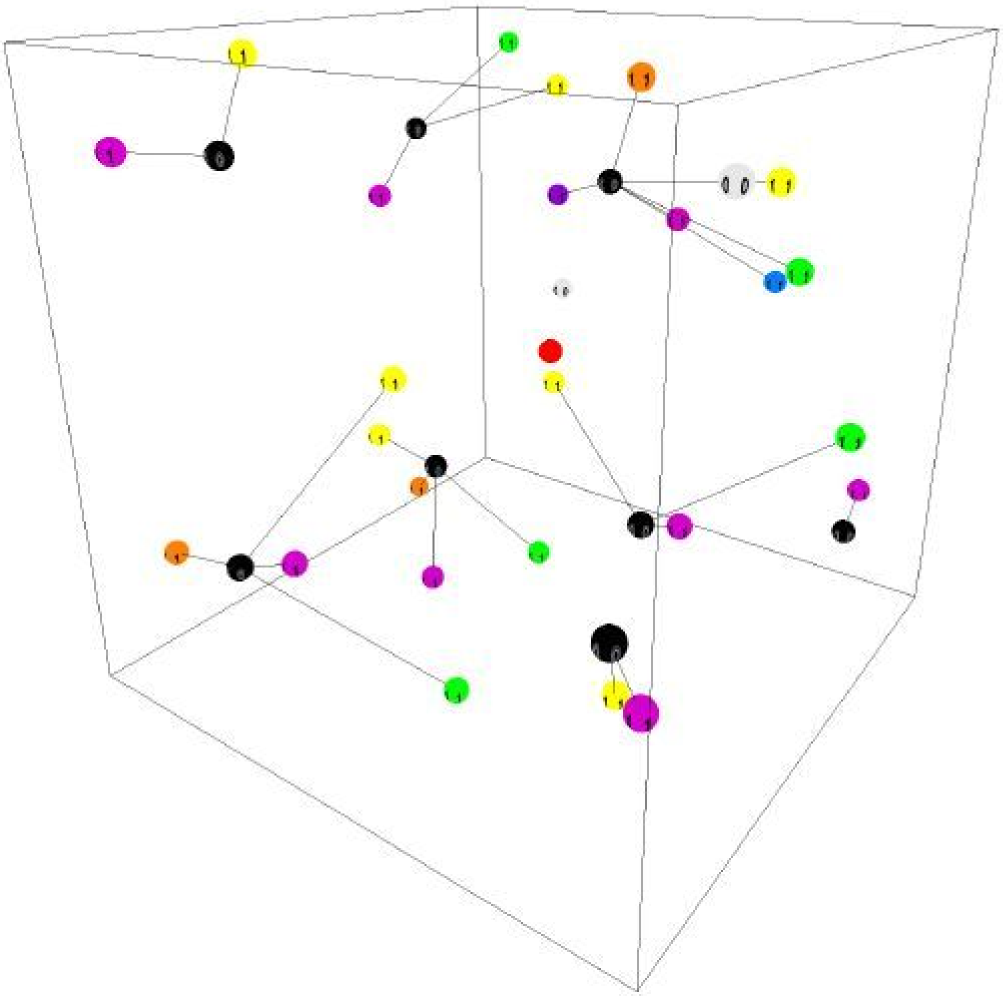
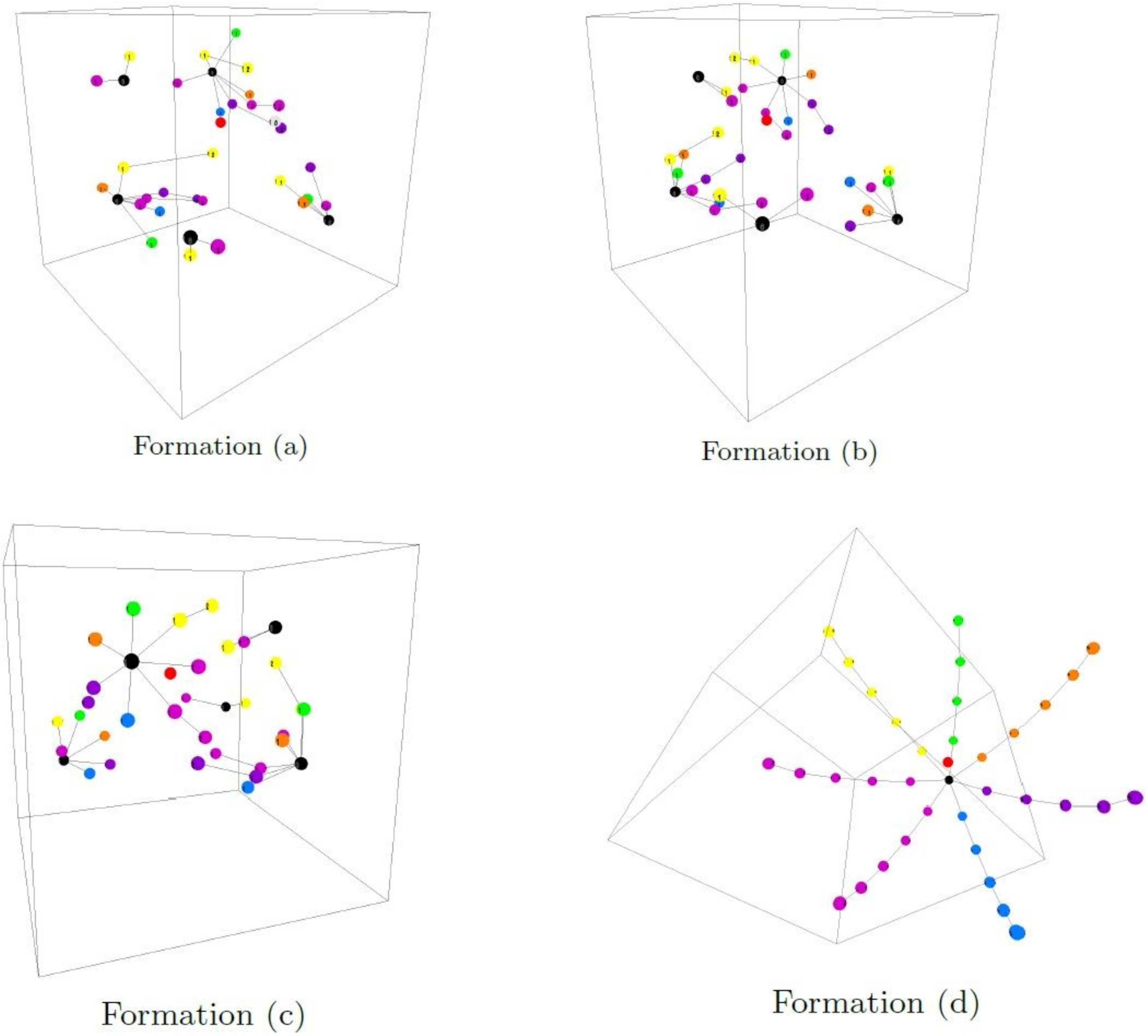


Figure 8

Clustering view



Formation (a)

Formation (b)

Formation (c)

Formation (d)

Figure 9

Formation - View

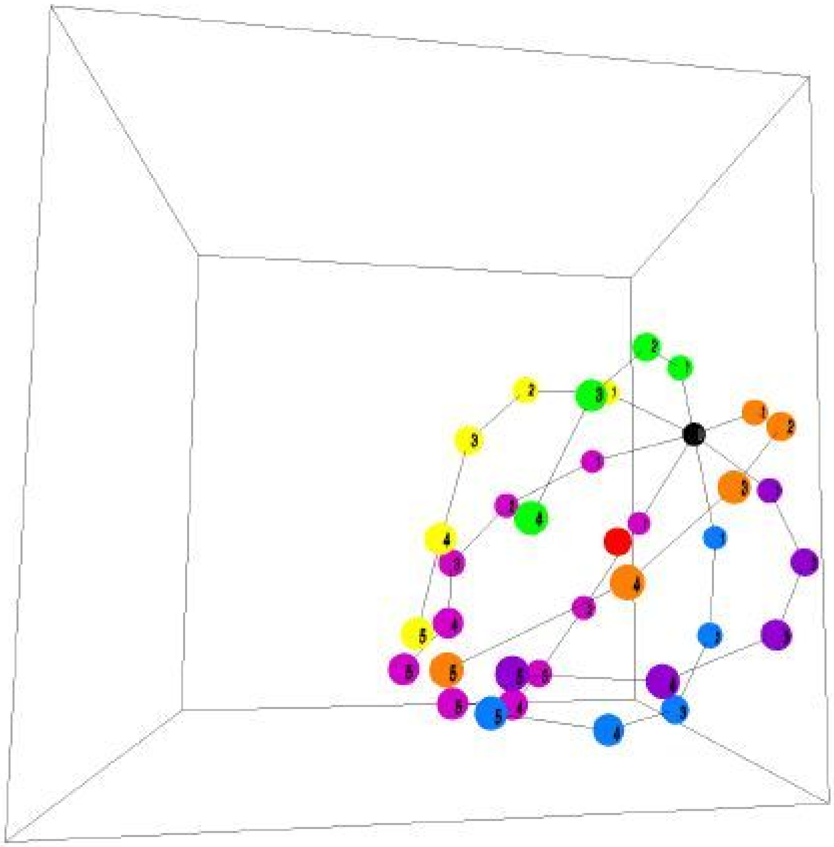


Figure 10

Chase View - Enclosure 1

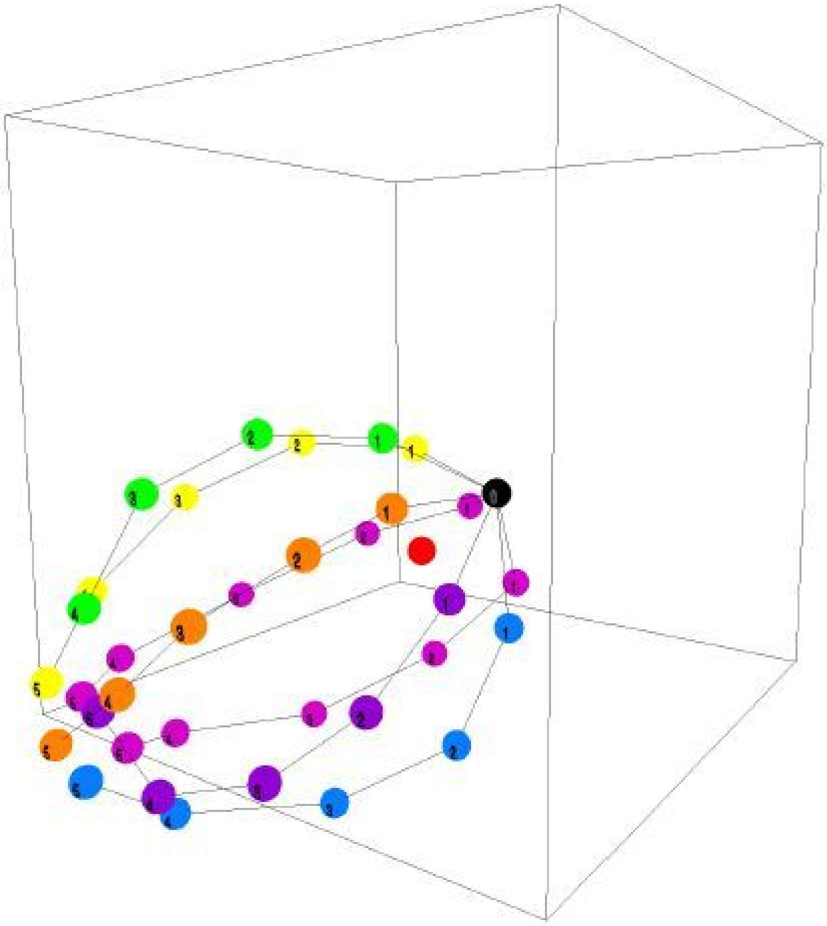


Figure 11

Chase View - Enclosure 2

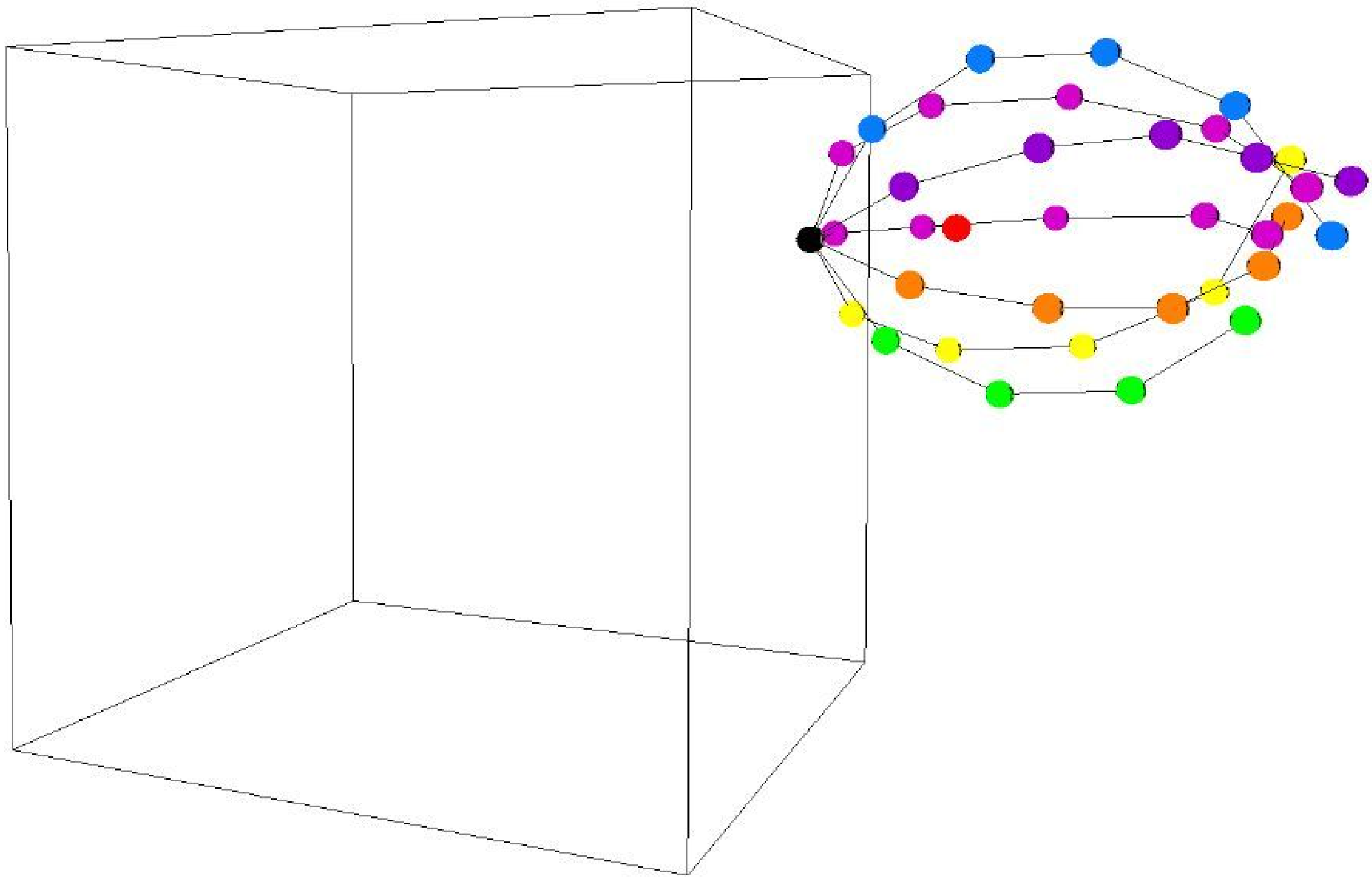


Figure 12

Escort - View

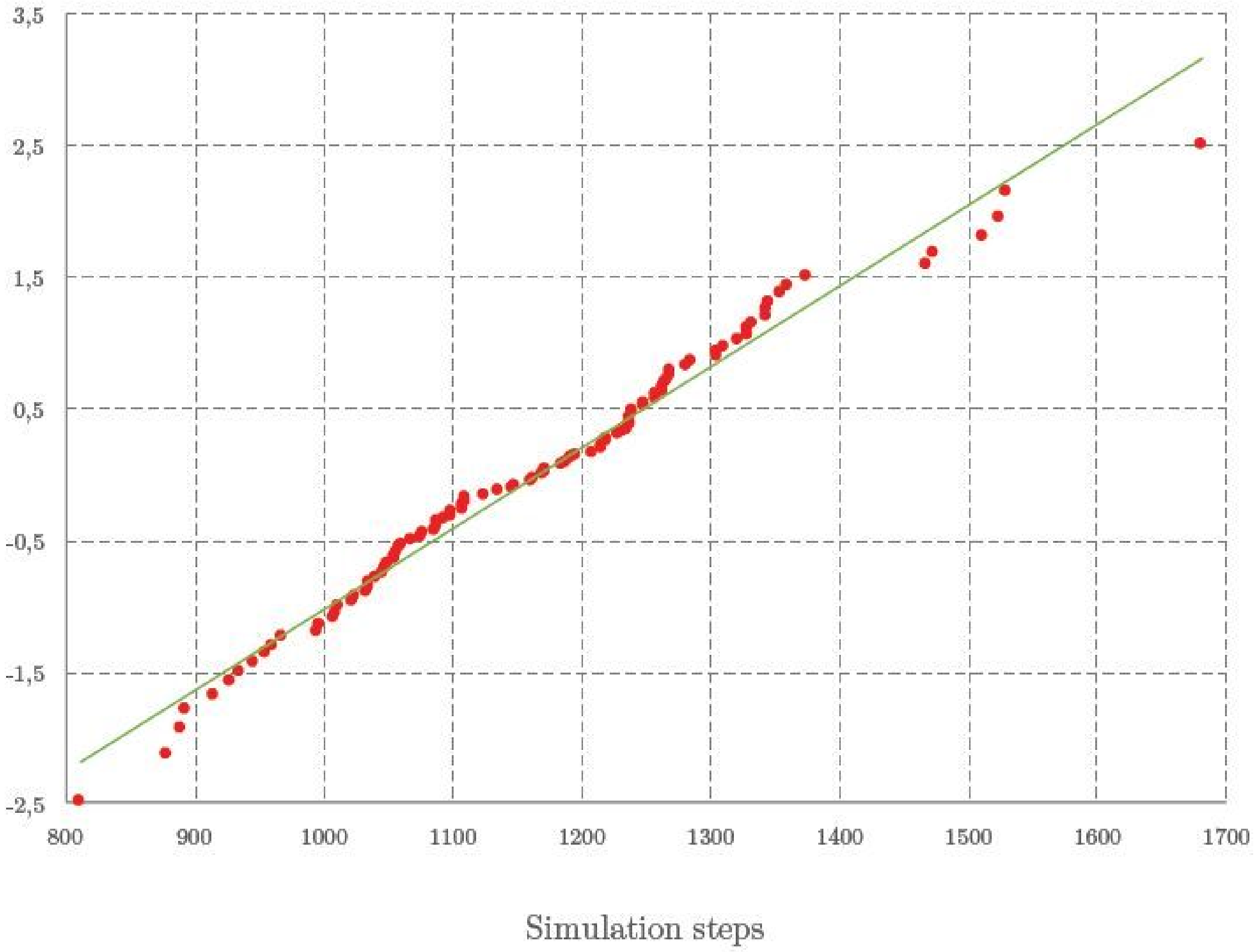


Figure 13

Normality test

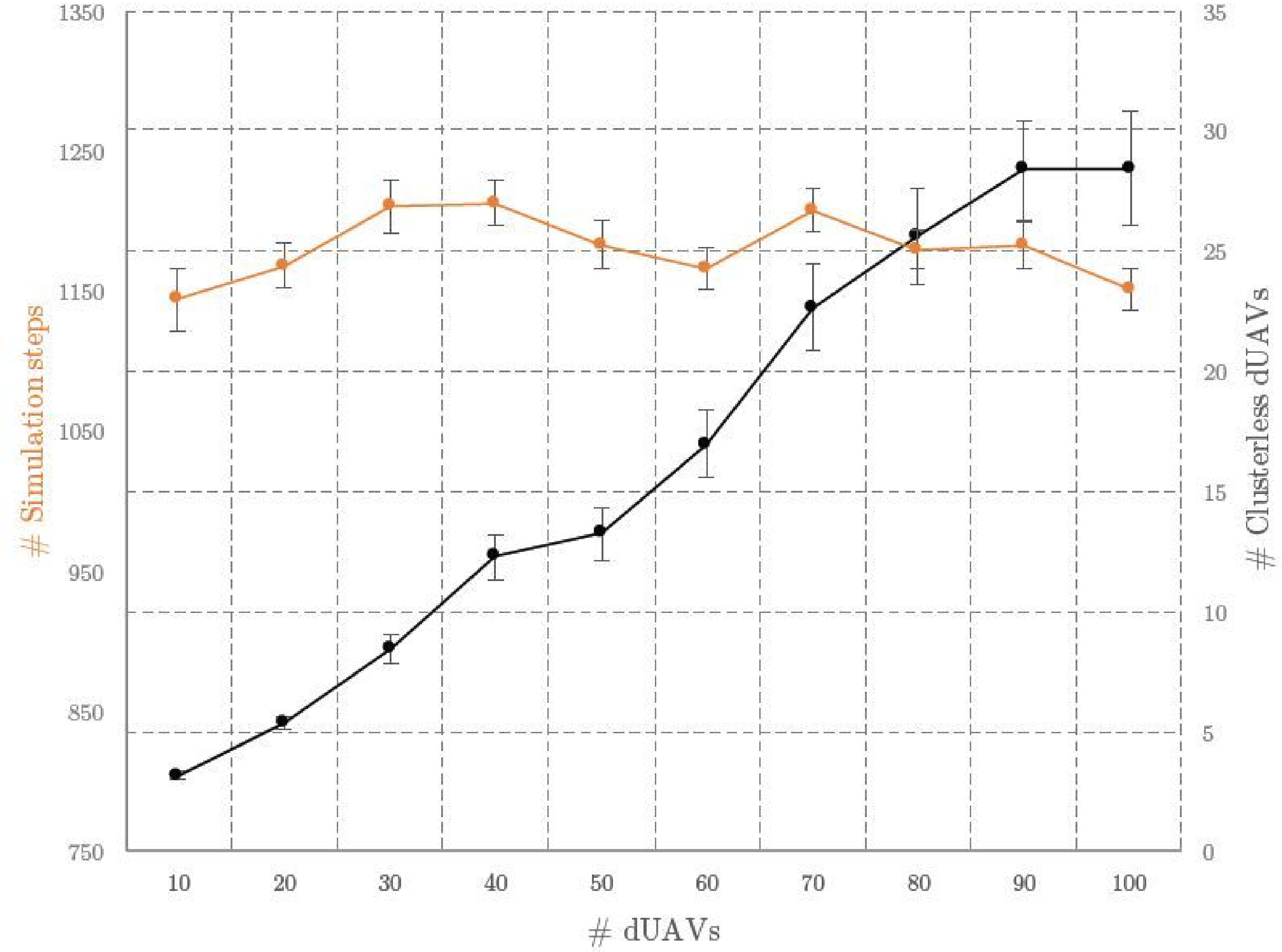


Figure 14

Dependency between number of dUAVs and simulation times

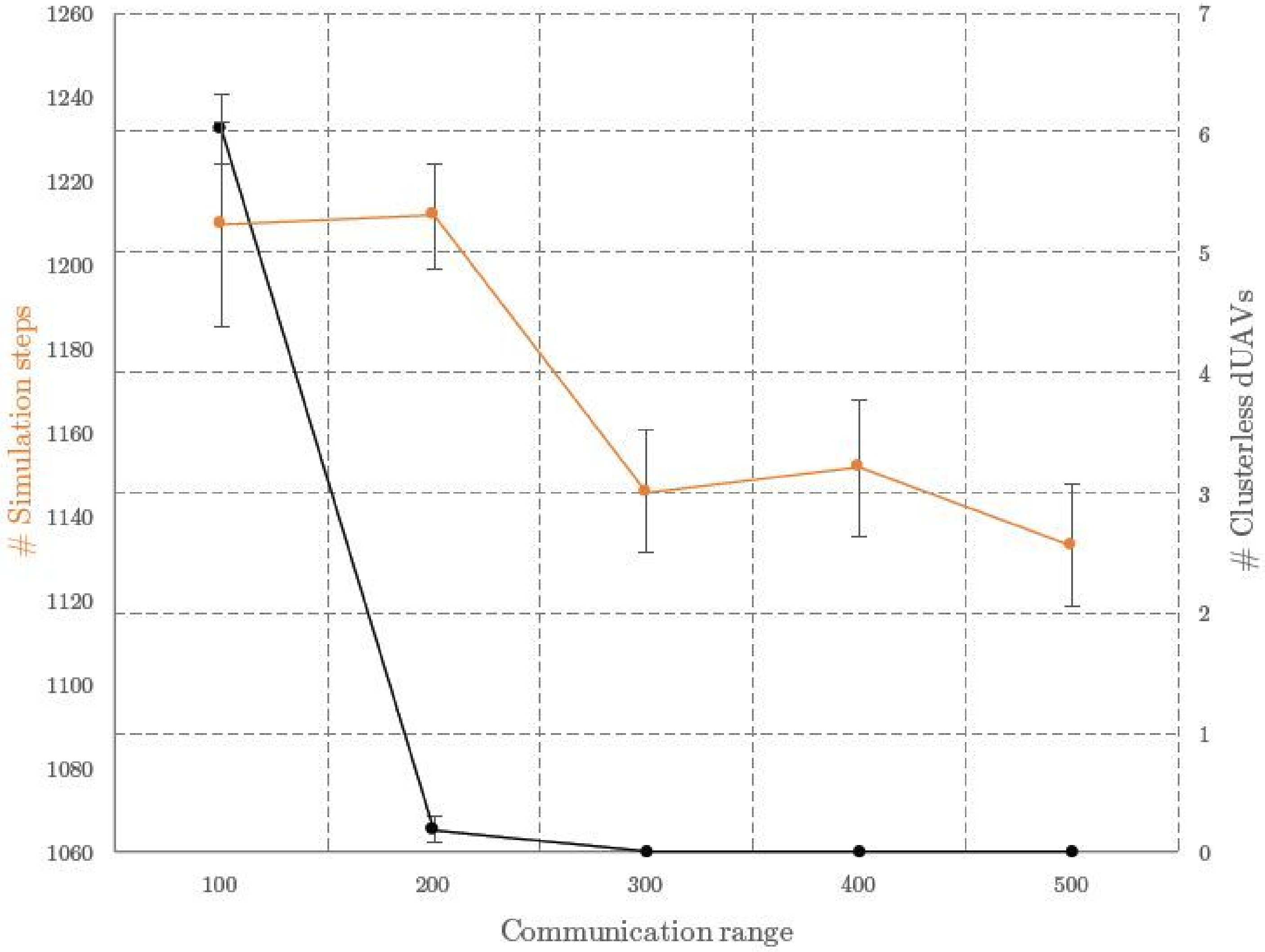


Figure 15

Dependency between communication range and simulation times

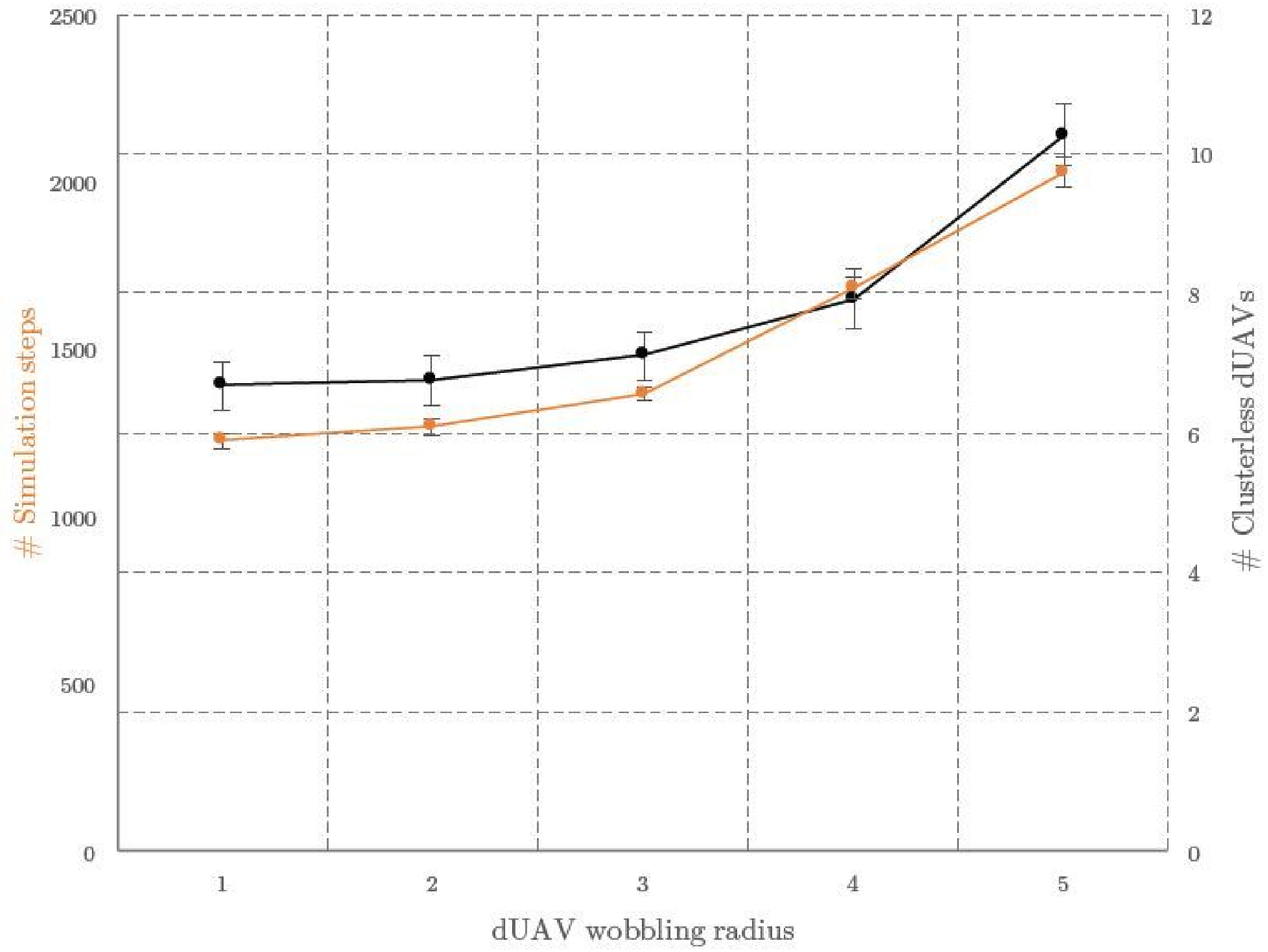


Figure 16

Dependency between dUAV wobbling radius and simulation times

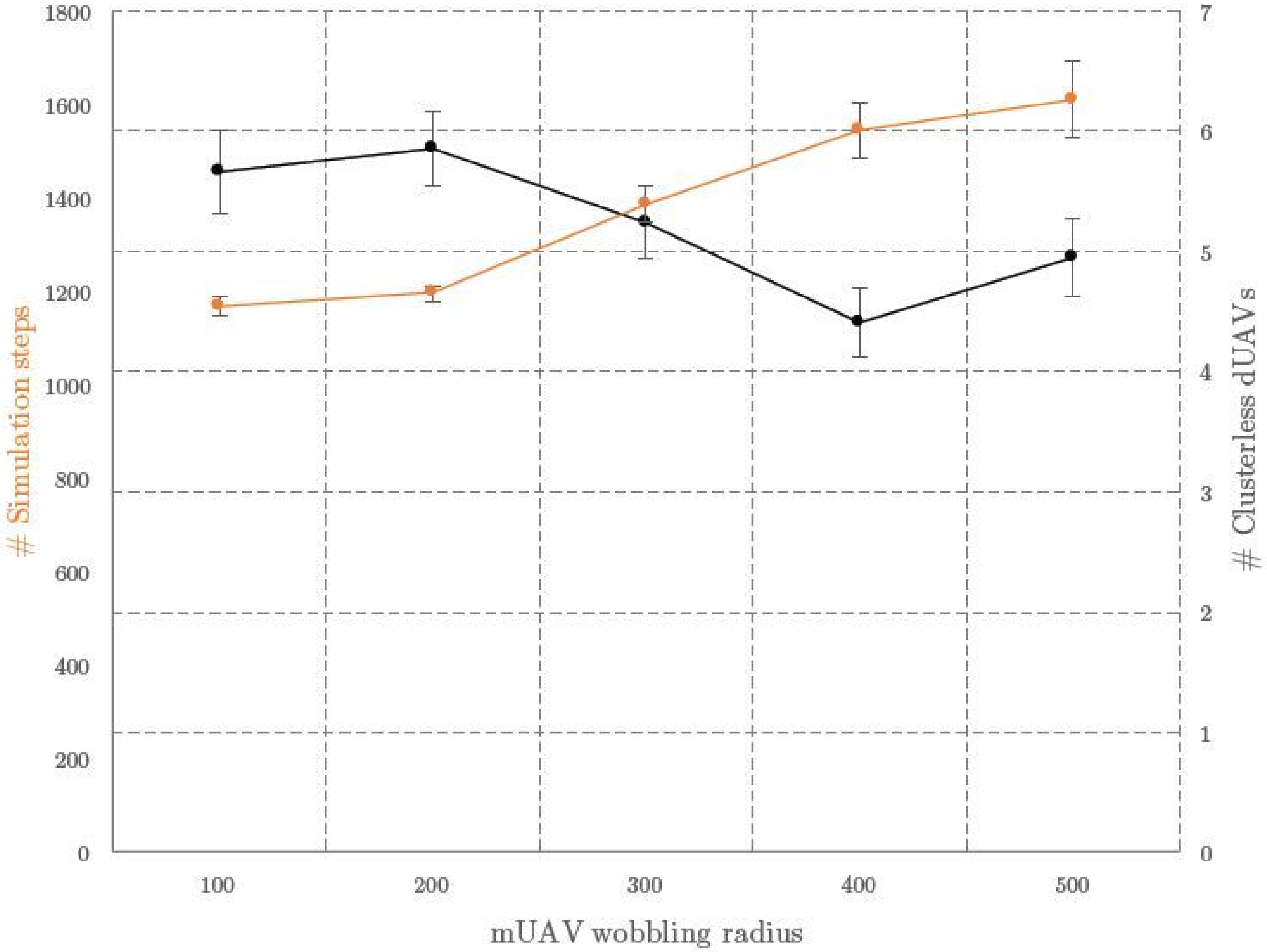


Figure 17

Dependency between mUAV wobbling radius and simulation times

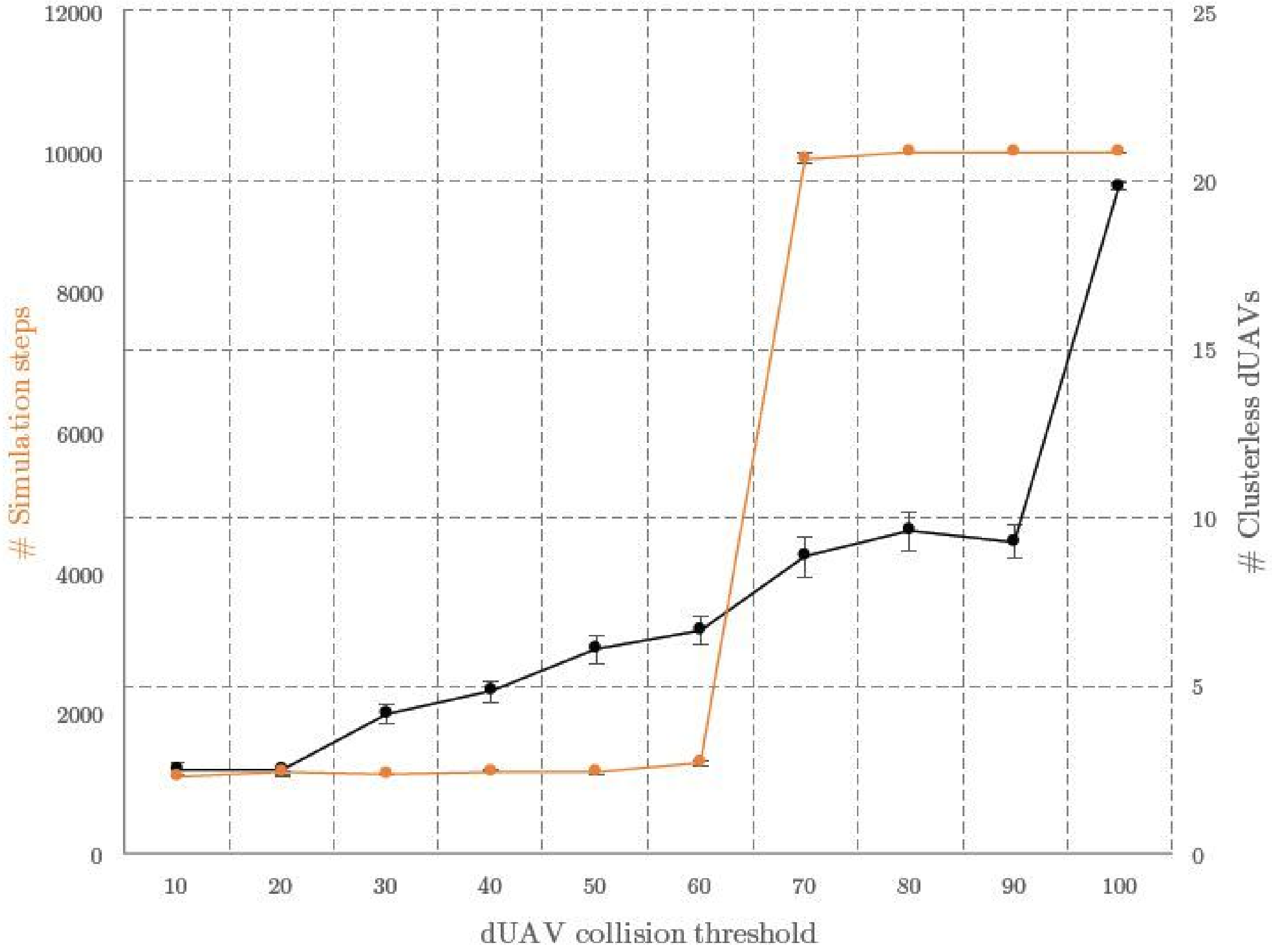


Figure 18

Dependency between dUAV collision threshold and simulation times

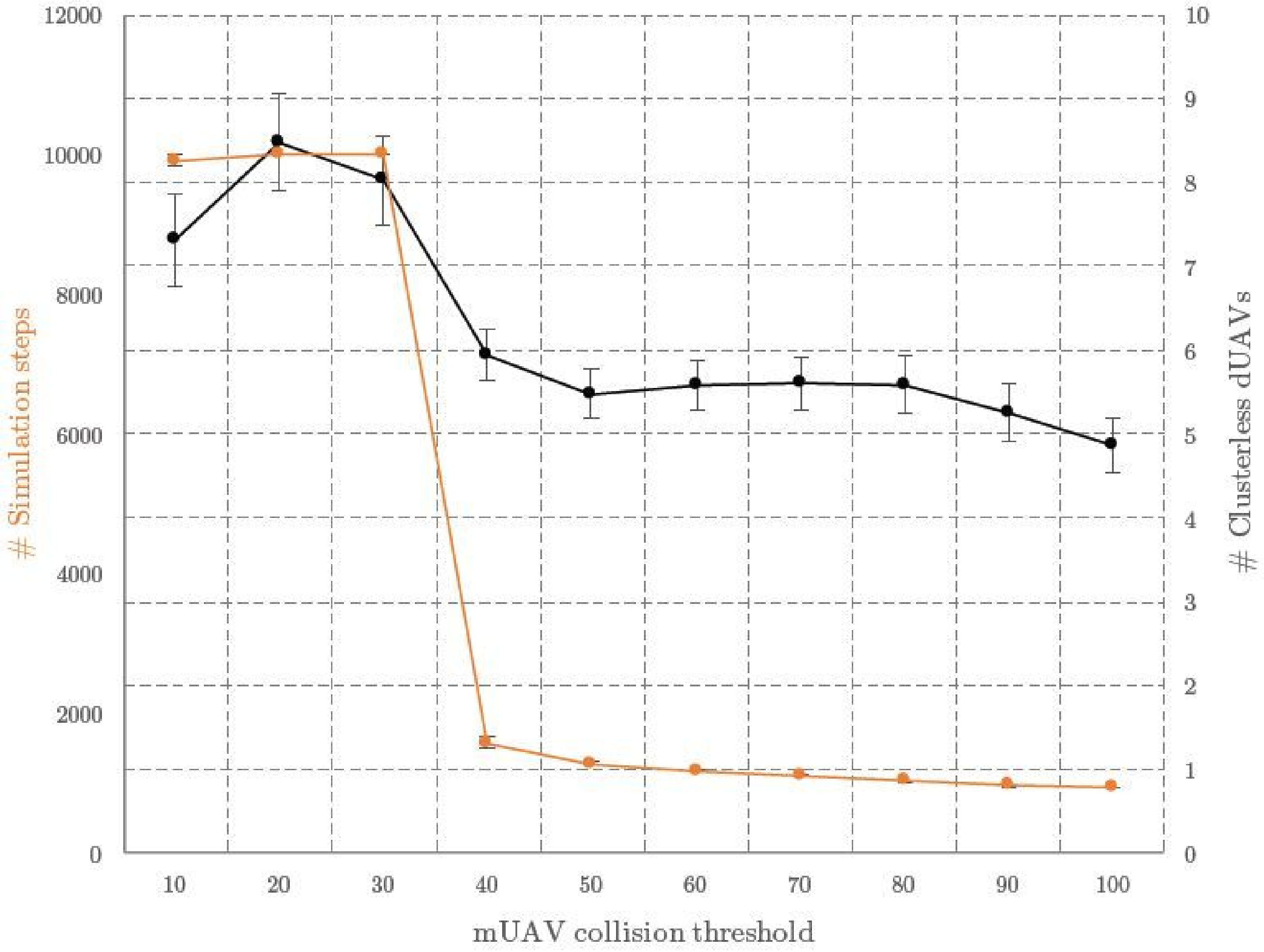


Figure 19

Dependency between mUAV collision threshold and simulation times

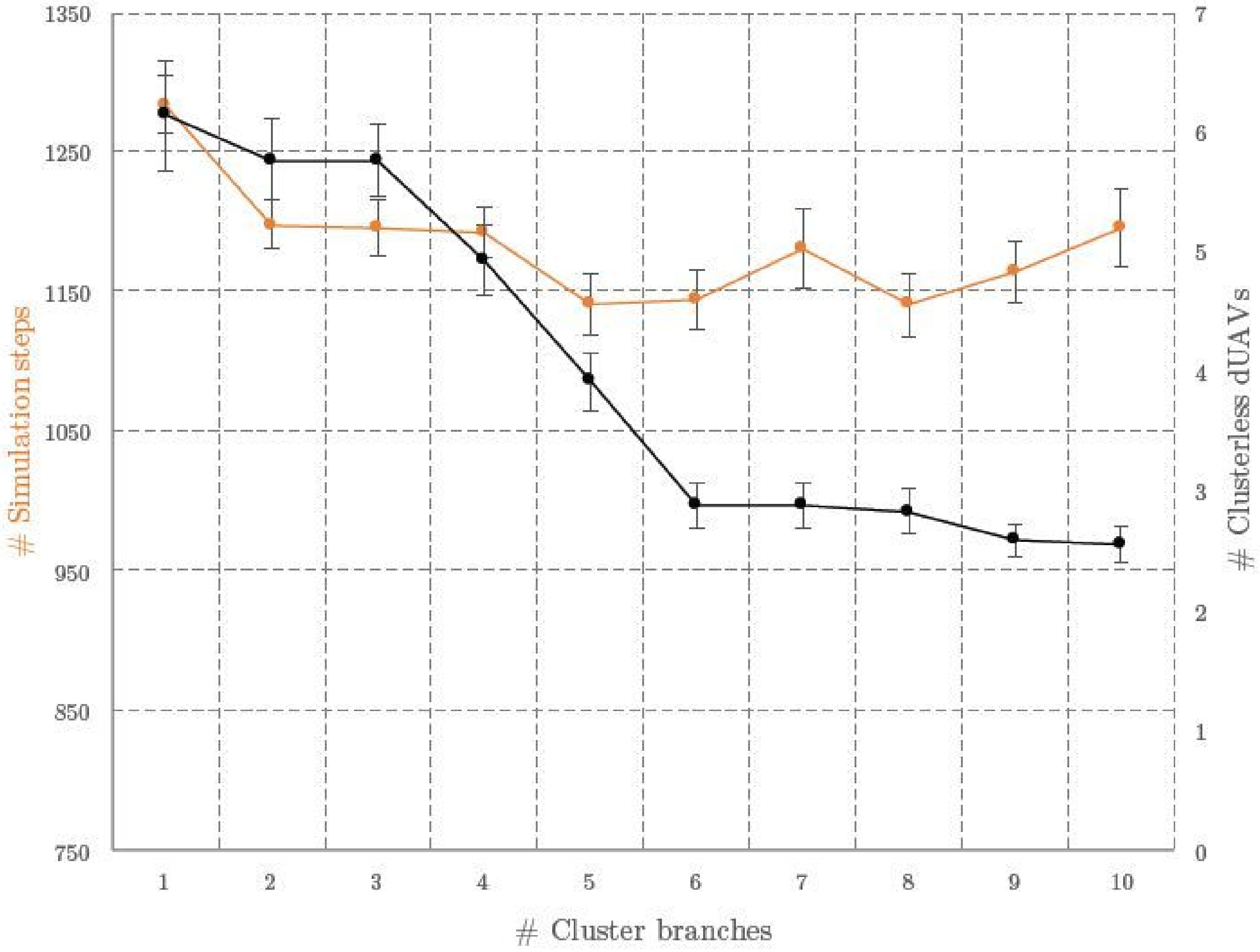


Figure 20

Dependency between number of branches and simulation times