UDC 004.855.5(045)
DOI:10.18372/1990-5548.87.20906

[1]Denys Trotsyuk,
[2]Kyrylo Lesohorskyi

TOPOLOGY MANAGEMENT OF A SWARM OF UNMANNED AERIAL VEHICLES

[1]Department of Avionics and Control Systems, Faculty of Air Navigation Electronics and Telecommunications, State University "Kyiv Aviation Institute", Kyiv, Ukraine
[2]Faculty of Informatics and Computer Science, National Technical University of Ukraine "Ihor Sikorsky Kyiv Polytechnic Institute," Kyiv, Ukraine
E-mails: [1]hofmann.denys@gmail.com ORCID 0009-0006-2256-7843
[2]lesogor.kirill@gmail.com ORCID 0000-0003-2773-7398

***Abstract*—*The article is devoted to the problem of controlling the topology of a swarm of unmanned aerial vehicles (UAVs). The primary objective of swarm operation is maintaining a dynamic topology, i.e., stable information exchange and structural consistency between swarm elements in a constantly changing environment. It is shown that existing approaches rely on a global navigation satellite system (GPS) for UAV positioning, such as the Global Positioning System (GPS). This approach is unacceptable, as UAVs can suddenly experience loss of GPS signals while performing their missions, potentially resulting in a lack of location information. To support functionality, including the use of geographic routing protocols, maintaining connectivity in dynamic network conditions, and adapting to topological changes in the UAV swarm, this paper utilizes virtual coordinates. This paper develops a virtual coordinate system that forms the basis of the proposed method for UAV swarm topology management. This eliminates the need for global coordinates, a centralized controller, motion model coordination, and pre-calibration of the swarm formation. The system operates solely based on local distance measurements to neighbors, making it universal, scalable, and resilient to the loss of individual UAVs. Algorithms for merging and separating swarms based on the virtual coordinate system have been developed.*

Keywords—Dynamic topology; fault tolerance; geographic routing protocols; swarm robotics; unmanned aerial vehicles; virtual coordinate systems.

I. INTRODUCTION

In the current context of unmanned aerial systems development, technologies for the group and collective use of unmanned aerial vehicles (UAVs) are becoming especially important. Unlike single aircraft platforms, a UAV swarm can improve mission efficiency through collective behavior, the distribution of functions among elements, adaptability to external conditions, and high fault tolerance, collecting and processing information for decision-making in real time [1] – [3].

The primary objective of a swarm's operation is to maintain a dynamic topology – that is, stable information exchange and structural consistency between swarm elements in a constantly changing environment. In practice, this requires the swarm's ability to maintain connectivity, reorganize its spatial structure, adaptively merge into larger groups, or, conversely, split into sub-groups depending on operational requirements.

The paper [4] proposes the use of cellular networks as a communication infrastructure for UAV swarms to support dynamic topology and proposes its own high-level architecture for swarm autonomy. At a basic level, a UAV swarm is a floating dynamic wireless network, commonly known as a wireless mesh network. A mesh network is a network topology in which each node transmits data for the network. All nodes collaborate in distributing data within the network. A mesh network whose nodes are connected to each other is a fully connected network. A mesh network can be designed using a routing method or a message propagation method. With the routing method, a message propagates along a path, hopping from node to node until it reaches its destination. To ensure the availability of all its paths, a routing network must provide continuous connections and reconfigure around broken or blocked paths using self-healing algorithms. The self-healing capability allows a routing-based network to function when one node fails or the connection is interrupted. As a result, the network is generally quite reliable, since there is often more than one path between a source and a destination in the network.

A key issue in swarm flight is ensuring reliable communication between UAVs, which creates a number of technical challenges, such as high communication overhead associated with route discovery and maintenance algorithms, swarm localization and control in flight, UAV coordination within the swarm, and the level of UAV autonomy in navigation. Traditional approaches, such as table-based or on-demand routing protocols, struggle to effectively manage communication in large networks and have difficulty dealing with the dynamic nature of mobile nodes, resulting in significant overhead that makes such protocols impractical [5]. Among the most promising solutions facilitating data exchange in dynamic swarms is the geographic routing protocol, which uses the geographic location of UAVs to route information within the swarm [6], [7]. However, this approach depends on precise geographic locations, traditionally provided by GNSS, such as the Global Positioning System (GPS). However, UAV missions often occur in challenging environments – such as combat zones, natural clutter such as mountain ranges and dense forests, dense urban areas with complex radar structures, or indoor (GNSS-denied) environments – where GNSS signals are either weak or unavailable, requiring alternative strategies for maintaining effective communication and coordination within the swarm.

II. VIRTUAL COORDINATE SYSTEMS

To ensure that the geographic routing protocol operates in environments where geographic coordinates may be unavailable or unreliable, virtual coordinates, which were first proposed in [8], can be used as a replacement.

A virtual coordinate is a position assigned to a node based on its relative distance to neighboring nodes. It should not coincide with the geographic coordinate. In article [8], three cases of finding virtual coordinates are considered successively: perimeter nodes know their exact geographic coordinates (a relaxation algorithm is used to solve the problem); perimeter nodes know that they are actually located on the perimeter, but we exclude the assumption that these perimeter nodes know their exact geographic location (relaxation and triangulation algorithms are used to solve the problem); there is no information about the coordinates of the perimeter nodes, but the perimeter nodes know that they are on the perimeter (relaxation and triangulation algorithms are used to solve the problem).

The resulting virtual network topology mirrors the physical network, preserving its structure and connectivity. Once all UAVs (network nodes) in the swarm (network) are assigned virtual coordinates, a geographic routing protocol can be implemented. Importantly, virtual coordinates allow the swarm to maintain its connectivity in a flexible and decentralized manner, even when actual geographic data is unavailable.

Most of the work on VCS design uses "always-on" static nodes to serve as topology "anchors" and is therefore not suitable for routing in dynamic mesh environments; the overhead generated by these algorithms is often comparable to that of table-based routing protocols [9].

In article [10], a gradient-aware virtual coordinate system (GAR) for multi-node mesh networks is proposed. It creates a routing gradient in the network without the need to obtain location information from external sources such as GPS or cellular infrastructure and approximates the physical topology up to the rotation and translation operations of the network graph. The proposed VCS distributes and dynamically assigns logical addresses to nodes based on information periodically exchanged in physical layer beacons or specially designated packets, as well as (possibly erroneous) distance estimates to nearest neighbors obtained using the received signal strength (RSS). The system dynamically adapts to changing network conditions, including network mergers and splits, node on/off behavior, and node mobility.

The main goal of the proposed GARVCS system is to enable a node to calculate its virtual GAR address based only on locally available knowledge in order to minimize the overhead of distributing information throughout the network.

III. PROBLEM STATEMENT

We consider a network of n UAVs located in some two-dimensional space, $n < \infty$ [10]. Each UAV cannot determine the geographic coordinates of its location. Each individual UAV (network node) has a unique identification number$i$, $i = 1, 2, \ldots$. Node i is able to receive a signal from other nodes within a certain radius. Based on the received signal strength (RSS), node i can estimate the physical distance d_i, between itself and another node j.

Let us denote the set of one-hop neighbors of node i as N_i. Here, one-hop neighbors are adjacent nodes that are directly reachable via the air interface without any intermediate relay nodes. Two-hop neighbors of node i are nodes that require a one-hop relay to transmit data to node i. The set of two-hop neighbors of node i is denoted as $M_i = \left(\bigcup_{j\in N_i} N_{j}{}_{i}\right)\Big/N$. The sets of one-hop and two-hop neighbors do not overlap.

Given that the nodes' geographic coordinates are unknown, we aim to construct a virtual topology that mirrors the physical topology up to rotation, translation, and scaling. In the virtual topology, node i has virtual coordinates $s_i = (x_i, y_i)$. Since nodes are constantly moving in the network, their physical coordinates change over time, as do their virtual coordinates. For this reason, we introduce another index – the time step *t*. The goal is to find the virtual coordinates $s_t = (x_t, y_t)$ of node *i* at time step *t*.

To construct a virtual topology that mirrors the physical topology, including rotation, translation, and scaling, we assume that the following information is available at each node:

- a list of unique identifiers (IDs) of destinations;
- a list of unique IDs of one-hop neighbors, their GAR addresses, and distance estimates to them, obtained, for example, using RSS;
- lists of two-hop neighbors, i.e., for each one-hop neighbor, a list of IDs of its neighbors and their GAR addresses.

We restrict it to two-step neighbors, providing a trade-off between the overhead and accuracy of the algorithm.

Most of the above information is obtained from neighboring nodes through the exchange of signaling packets, while distance estimates can be obtained by measuring the RSS and then applying a free-space propagation loss model or by any other available method, including an empirical estimate based on the number of common neighbors. Identifiers should be pre-stored as a phonebook. The remaining parameters should be defined for each node individually or globally, depending on the type and characteristics of the radio interface used and the overall considerations for the network in question.

There are no specific restrictions on nodes or the network's purpose. For example, nodes can move freely, be temporarily turned on and off, etc.

IV. Algorithm for Determining the Virtual Coordinates of a Node

The algorithm constructs an optimization objective function that includes distance estimates to neighbors one hop away and penalties based on neighbors two hops away. The optimization problem is solved to obtain the most suitable virtual address.

Consider the following optimization function

$$F_0\left(s_{k,j}\right)=\sum_{j\in N_i}\left[F_1\left(p\left(s,s_{k,j}\right),d_{i,j}\right)+\sum_{j\in M_i}F_2\left(p\left(s,s_{k,j}\right),d_{i,j}\right)\right], \qquad (1)$$

where $p\left(s_i,s_j\right)=\sqrt{\left(x_i-x_j\right)^2+\left(y_i-y_j\right)^2}$; $s = (x, y)$ is the resulting pair of virtual coordinates for the node in question, which it must adopt at the current step.

Optimization of function (1) consists of finding the virtual coordinates of node *i* by minimizing the penalty assigned to it for violating a set of rules.

$$F_1\left(d,d_n\right)=\frac{A}{d_n}\left(d-d_n\right), \qquad (2)$$

$$F_2\left(d,d_n\right)=\begin{cases}\dfrac{A}{d_n}\left(d-d_n\right)d\le d_n,\\ 0,\ d>d_n.\end{cases} \qquad (3)$$

where d_n is a predefined limit.

The updated virtual coordinate of the current node is calculated as follows:

$$s_{k+1,i}=s_{k,i}+\left[\sum_{j\in N_i}\frac{s_{k,j}-s_{k,i}}{p\left(s_{k,i},s_{k,j}\right)}F_1\left(p\left(s_{k,i},s_{k,j}\right),d_{i,j}\right)+\sum_{j\in M_i}\frac{s_{k,j}-s_{k,i}}{p\left(s_{k,i},s_{k,j}\right)}F_2\left(p\left(s_{k,i},s_{k,j}\right),d_{i,j}\right)\right]\Delta. \qquad (4)$$

In equation (4) $s_{k+1,i}$ is the virtual coordinate of node *i* at iteration $k + 1$, recursively updated from $s_{k,i}$, and Δ is a parameter defining the relative convergence step. The functions $F_1(d,d_n)$ and $F_2(d, d_n)$ control the force with which the current node is pushed or pulled by its neighbors.

Fault tolerance is one of the basic properties of intelligent swarm systems. In real-world operating conditions, UAVs can fail due to mechanical damage, loss of power supply, interference with the communication channel, or enemy action. Despite this, the swarm must continue to perform its assigned mission without losing controllability and coordination.

1) Principles of ensuring fault tolerance

Fault tolerance is achieved due to:

- Distributed control architecture. There is no single decision-making center, so the loss of one node does not disrupt the system.
- Automatic topology reconfiguration. After the node disappears, the connectivity graph *G*(*t*) is updated, and neighbors rebuild communication routes.
- Local state analysis. Each agent stores a list of active neighbors in the Hello table, which is regularly updated. If a node does not respond within TtimeoutT_{timeout}Ttimeout, it is removed from the list.
- Self-recovery of communications. UAVs search for new connections using the Neighbor Discovery mechanism or transmitter power expansion to temporarily overcome the network gap.

2) Swarm reconfiguration model

After losing a node, the swarm transitions to a new topology state:

$$G'(t+\Delta t) = G(t)\{v_k\}.$$

To Preserve Integrity, the following condition must be met:

$$\lambda_2\left(L\left(G'(t+\Delta t)\right)\right) > 0,$$

where λ_2 is the second eigenvalue of the Laplace matrix, which guarantees the connectivity of the graph.

If $\lambda_2 \to 0$, the local reconstruction procedure is activated: the drones increase the communication range, move towards the nearest nodes and form new edges in the graph.

3) Local stabilization mechanism

Each UAV checks its neighbors with a frequency fcheckf_{check}fcheck (typically 1–5 Hz). If one or more neighbors disappear, a local stabilization cycle is activated:

$$S_{n+1,i} = S_{n,i} + \eta \sum_{j \in N_i'} \left(S_{n,j} - S_{n,i}\right),$$

where N_i' is the updated set of neighbors; η is the correction coefficient.

This mechanism allows the swarm to “tighten” the created gap without the intervention of an external controller.

4) Detection and compensation of communication loss

In addition to the physical loss of the drone, it is critically important to restore the logical connection. For this purpose, the following are used:

- Multi-Path Routing: data can be transmitted via several alternative paths.
- Link Quality Estimation: the system selects the optimal connection based on RSSI and SNR.
- Role reassignment: if the leader of the local swarm is lost, the neighbors select a new one based on the centrality or energy reserve metric.

5) Self-Healing Algorithm

The algorithm works in four stages:

a) Loss detection: if a neighbor does not respond within $T_{timeout}$, the drone marks it as inactive.

b) Impact assessment: it is analyzed whether the loss of a node caused the graph to become unconnected.

If $\lambda_2(L(G)) < \lambda_{\min}$, a rebuild is started.

c) Search for new neighbors: the Extended Neighbor Discovery mode is activated, in which the signal range is increased.

d) Topology rebuild: the neighboring drones shift their positions according to the updated set of coordinates in the VCS until a stable connection is restored.

e) Full isolation scenario

If an individual UAV becomes completely isolated, it enters the autonomous survival mode (Survival Mode).

f) In this mode, the drone:

- reduces the transmitter power to save energy;
- periodically sends broadcast search signals;
- when new nodes are detected, it initiates the creation of a new local swarm GnewG_{new}Gnew;
- synchronizes its VCS with the obtained neighbors using the Merging procedure.

Thus, even a single node can become the nucleus for the formation of a new swarm, which ensures self-renewal of the system at the architecture level.

g) Assessing the effectiveness of fault tolerance

The following metrics are used to assess the effectiveness:

- Connectivity retention ratio (CR):

$$\mathrm{CR} = \frac{N_{active}}{N_{total}},$$

where N_{active} is the number of nodes that remain connected after failures.

- *Reconfiguration time (T_reconf):* the interval between the loss of a node and its restoration.
- *Bandwidth loss (ΔBW):* the difference between the initial and current bandwidth after the incident.

For stable swarms, the following values are characteristic:

$$\mathrm{CR} > 0.9,\ T\_reconf < 3\ \mathrm{s}$$

h) Advantages of distributed fault tolerance

- absence of a single point of failure;
- ability to self-reconfigure without operator intervention;
- maintaining collective control even with partial loss of equipment;
- adaptation to dynamic battlefield conditions or catastrophic environments.

V. RESULTS

The algorithm for calculating virtual coordinates for a swarm consisting of five fixed-wing UAVs flying on a given horizontal course was tested using mathematical models of longitudinal and lateral motion dynamics. Perturbations caused by atmospheric turbulence were introduced into the equations of the dynamic models. The virtual coordinates were calculated according to expressions (2), (3). The positions of the UAVs in the swarm were adjusted based on the virtual coordinates, taking into account the current distances between adjacent UAVs.

VI. CONCLUSION

In this paper, we implemented an approach to control the topology of a swarm of UAVs based on the calculation of virtual coordinates in conditions of atmospheric turbulence and the absence of GPS information. This approach will be further validated using a real UAV swarm.

REFERENCES

[1] Yongkun Zhou, Bin Rao, and Wei Wang, "UAV Swarm Intelligence: Recent Advances and Future Trends". In: IEEE Access 8, (2020), pp. 183856–183878. https://doi.org/10.1109/ACCESS.2020.3028865

[2] Abhishek Sharma, Shraga Shoval, Abhinav Sharma, and Jitendra Kumar Pandey. "Path Planning for Multiple Targets Interception by the Swarm of UAVs based on Swarm Intelligence Algorithms: A Review". In: *IETE Technical Review*, (2021), pp. 1–23. https://doi.org/10.1080/02564602.2021.1894250

[3] Alejandro Puente-Castro, Daniel Rivero, Alejandro Pazos, and Enrique Fernandez-Blanco. "A Review of Artificial Intelligence Applied to Path Planning in UAV swarms," In: *Neural Computing and Applications* (2021), pp. 1–18. https://doi.org/10.1007/s00521-021-06569-4

[4] Aaron Sempf. Swarm Commnication. https://medium.com/c%D3%95lus-concept/swarm-communication-33cffc47db6d, Apr 20, 2015

[5] Rahul Malhotra, Nitin Nikesh, and Sangeeta Monga."Analysis of Table Driven and On-demand Routing Protocols for Mobile Adhoc Networks". In: Indian Journal of Computer Science and Engineering 2.3 (2011), pp. 391–401.

[6] Sudesh Kumar, Ram Shringar Raw, Abhishek Bansal, and Pawan Singh. "UF-GPSR: Modified geographical routing protocol for flying ad-hoc networks". In: Transactions on Emerging Telecommunications Technologies 34.8 (2023), e4813. https://doi.org/10.1002/ett.4813

[7] M. M Pandith, N. K Ramaswamy, M. Srikantaswamy, and R. K. Ramaswamy,"A comprehensive review of geographic routing protocols in wireless sensor network". In: Information Dynamics and Applications 1.1 (2022), pp. 14–25. https://doi.org/10.56578/ida010103

[8] Ananth Rao, Sylvia Ratnasamy, Christos Papadimitriou, Scott Shenker, and Ion Stoica. "Geographic routing without location information". In: Proceedings of the 9th annual international conference on Mobile computingand networking. ACM. 2003, pp. 96–108. https://doi.org/10.1145/938985.938996

[9] A. Awad, R. German, and F. Dressler, Exploiting virtual coordinates for improvedrouting performance in sensor networks, IEEE Trans. Mob. Comput. 10 (9), (2010), 1214–1226. https://doi.org/10.1109/TMC.2010.218

[10] Andrey Samuylov, Dmitri Moltchanov, Roman Kovalchukov, Anna Gaydamaka, Alexander Pyattaev, Yevgeni Koucheryavy GAR: Gradient assisted routing for topology self-organization in dynamic mesh networks Computer Communications 190 (2022) 10–23. https://doi.org/10.1016/j.comcom.2022.03.023

Received: January 05, 2026
Accepted: January 21, 2026
Published: March 06, 2026

Trotsyuk Denys. ORCID 0009-0006-2256-7843. Postgraduate Student.
Department of Avionics and Control Systems, Faculty of Air Navigation Electronics and Telecommunications, State University "Kyiv Aviation Institute", Kyiv, Ukraine.
Education: National Aviation University, Kyiv, Ukraine, (2023).
Research area: Identification of complex systems.
E-mail: hofmann.denys@gmail.com

Lesohorskyi Kyrylo. ORCID 0000-0002-3297-9060. Postgraduate Student.
Department of Information Systems, Faculty of Informatics and Computer Science, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.
Education: National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine, (2022).
Research interests: artificial neural networks, artificial intelligence, distributed computing.
Publications: 20.
E-mail: lesogor.kirill@gmail.com

Д. О. Троцюк, К. С. Лесогорський. Керування топологією рою безпілотних літальних апаратів

Статтю присвячено проблемі керування топологією роя безпілотних літальних апаратів (БПЛА). Основним завданням при функціонуванні рою є підтримання динамічної топології, тобто стійкого інформаційного обміну і структурної узгодженості між елементами рою в умовах постійних змін середовища. Показано, що використання відомих підходів передбачає використання глобальної супутникової навігаційної системи для позиціонування БПЛА, наприклад, системи глобального позиціонування (GPS), що є неприйнятним, оскільки при виконанні своїх завдань БПЛА можуть раптово стикатися з втратою сигналів GPS, що може призвести до відсутності інформації про місцезнаходження. Для підтримки функціональності, що включає використання географічних протоколів маршрутизації, підтримка зв'язності в динамічних мережевих умовах та адаптацію до топологічних змін у рої БПЛА, у роботі використовуються віртуальними координатами. У роботі розроблена віртуальна координатна система, що є основою запропонованого методу управління топологією рою БПЛА, що забезпечує відсутність наявності: глобальних координат, централізованого контролера, узгодження моделей руху, попередньої калібрування формації рою. Система функціонує виключно на основі локальних вимірювань відстаней до сусідів, що робить її універсальною, масштабованою та стійкою до втрат окремих БПЛА. Розроблено алгоритми об'єднання та поділу роїв на основі віртуальної координатної системи.

Ключові слова: безпілотні літальні апарати; віртуальні системи координат; географічні протоколи маршрутизації; динамічна топологія; робототехнічні рої; відмовостійкість.

Троцюк Денис Олександрович. ORCID 0009-0006-2256-7843. Аспірант.
Кафедра авіоніки та систем управління, Факультет аеронавігації, електроніки і телекомунікацій, Державний університет «Київський авіаційний інститут», Київ, Україна.
Освіта: Національний авіаційний університет, Київ, Україна, (2023).
Напрям наукової діяльності: ідентифікація складних систем.
E-mail: hofmann.denys@gmail.com

Лесогорський Кирило Сергійович. ORCID 0000-0003-2773-7398. Аспірант.
Кафедра штучного інтелекту, Інститут прикладного системного аналізу, Національний технічний університет України «Київський Політехнічний Інститут імені Ігоря Сікорського», Київ, Україна.
Освіта: Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», (2022).
Напрям наукової діяльності: штучні нейронні мережі, штучний інтелект, розподіленні обчислення.
Кількість публікацій: 20.
E-mail: lesogor.kirill@gmail.com