Development of the Local Navigation Algorithm of the Agricultural UAVs During Swarm Movement

Maksym Ogurtsov[1], Oleh Rybalchenko[1], Vyacheslav Korolyov[1] and Oleksandr Khodzinsky[1]

[1] *V.M. Glushkov Institute of Cybernetics of the National Academy of Sciences of Ukraine, Ac. Hlushkova str. 40, building 1, apt. 801, Kyiv, 03187, Ukraine*

Abstract

Every year, the scope of unmanned aerial vehicle use is growing, and the amount of such applications is expanding. To enhance efficiency, the focus of development is shifting from individual unmanned aerial vehicles to utilizing swarms for various applications such as agricultural surveys, irrigation, etc.

The purpose of this work is to develop an algorithm for building a local positioning system for a swarm of unmanned aerial vehicles to maintain a stable structure during the swarm movement while performing its tasks in two-dimensional and three-dimensional settings. At the same time, it should be assumed that the swarm is controlled by one operator, regardless of the number of swarm elements. An additional limitation of the developed algorithm should be the prevention of potential collisions of unmanned aerial vehicles during aerial maneuvers.

As part of the work, the task of forming and maintaining the structure and configuration of an agricultural unmanned aerial vehicle flock in two-dimensional and three-dimensional settings was set. The formalization of the given task has been completed. Considered possible methods of mapping the graph formed by an unmanned aerial vehicle swarm into two-dimensional Euclidean space by forming a basic triangle method to create a relative coordinate system. To determine the coordinates of the remaining graph vertices on the plane and correspondingly increase the accuracy of local positioning, the multilateration method is used and considered simplified options – application of the trilateration algorithm, triangulation and a simplified algorithm for the degenerate case of two drones.

For the problem in a three-dimensional setting, the possibility of applying the multidimensional scaling algorithm using the methods of multidimensional scaling/reduction of dimensions is considered. The developed practical implementation of the created algorithm showed its efficiency during practical experiments, allowing it to determine the local positioning of elements of a swarm containing three to twenty elements.

Keywords

UAV, drone, swarm, swarm control, local positioning, multilateration, triangulation, Euclidean space

1. Introduction

Every year, the scope of use of unmanned aerial vehicles (UAVs) is increasing and the spread of such applications is expanding. At the same time, the emphasis is gradually shifting from the individual UAVs use to the use of UAV swarms, since for many tasks (for example, surveying agricultural areas, irrigation, etc.) this is more effective than the use of individual UAVs [1]. The increase in efficiency is achieved by reducing the number of operators required to operate the same number of UAVs as swarm elements, instead of using them in the "one operator – one UAV" mode. In addition, when using swarms, it is possible to use group management with elements of artificial intelligence, which in many tasks exceeds the capabilities and speed of the human operator's response [2]. But at the same time, there are several problems. In some cases, the unavailability of

14th International Scientific and Practical Conference from Programming UkrPROG'2024, May 14-15, 2024, Kyiv, Ukraine

[*] Corresponding author.

[†] These authors contributed equally.

maksymogurtsov@gmail.com (M. Ogurtsov); rv.oleg.ua@gmail.com (O. Rybalchenko); korolev.academ@gmail.com (V. Korolyov); okhodz@gmail.com (O. Khodzinsky)

0000-0002-6167-5111 (M. I. Ogurtsov); 0000-0002-5716-030X (O. Rybalchenko); 0000-0003-1143-5846 (V. Korolyov); 0000-0003-4574-3628 (O. Khodzinsky)

GPS for agricultural drones that perform the task of processing agricultural land from pests or monitoring crops [3].

2. Goal

The purpose of this work is to develop an algorithm for building a system of local positioning of a swarm of UAVs to maintain a stable structure during the movement of the swarm while performing its assigned tasks in two-dimensional and three-dimensional settings. At the same time, we believe that the swarm is controlled by one operator regardless of the swarm elements number. The additional limitation of the developed algorithm: the prevention of potential UAV collisions during aerial maneuvers.

3. Analysis of previous developments

In the existing scientific works devoted to solving the given problem [1-8], it is necessary to single out approaches based on the use of the Internet of Things paradigm and artificial intelligence tools [4], although they are resource-intensive and require significant computing power, which limits the scope of possible application. The use of parallel genetic algorithms and other methods of combinatorial optimization [6] is also interesting, although it should be noted that, according to the authors, solving the given problem in several applications is possible without using similar methods, in a simpler way.

Another approach is the pre-planning of UAV swarm routes [7], but this significantly narrows the scope of swarms based on this approach.

The most popular option is the use of particle swarm optimization algorithms [7-8], although this method is also quite computationally demanding. Usually in similar works, the variant with the development of a system of swarm elements local mutual positioning, presented in this work, is not considered.

4. Main part

Let's first consider how the problem can be solved in a simplified, two-dimensional formulation, which is applicable when the swarm moves at approximately the same height and the height difference can be neglected. This option is applicable for a swarm of ground drones communicating in the zone of radio waves direct propagation. We will consider the swarm elements as homogeneous, although with the additional parameters and restrictions it is possible to use heterogeneous UAVs.

Meaningfully, we have a set of drones that should form a swarm. In this work, a swarm of up to 20 drones is considered. Each drone is equipped with a set of sensors (GPS, compass, accelerometer, altimeter) that will al-low determination of mutual position and forming and maintenance of the swarm structure. The task is to achieve the desired configuration of mutual the swarm elements placement in the minimum time and further maintenance of this configuration with minimum costs of the total flight resource of all swarm elements. The algorithm must function iteratively, with a frequency of repetition at least once every few seconds, to consider possible changes in the swarm elements relative position. The configuration to be formed by the swarm must provide equal distances between adjacent swarm elements, i.e. three UAVs will form an equilateral triangle, four – a square, and so on. Thus, we have:

Input data – data from the sensors of each element of the swarm:

- GPS coordinates.
- the direction of movement.
- speed.
- height.

Parameters:

- maximum UAV movement speed.
- the maximum permissible flight height of the UAV.
- minimum permissible UAV flight height.
- maximum UAV flight range.
- the minimum permissible distance between UAVs.

Restrictions:

Minimization of the total additional movement of all swarm UAVs required to restore the given configuration.

Output data:

GPS coordinates of the points where each UAV should move to restore the given swarm configuration.

4.1. Two-dimensional formulation of the problem

The two-dimensional formulation of the problem will involve mapping the graph formed by the UAV swarm first into three-dimensional Euclidean space [9] (Figure 1), which must then be reduced to two-dimensional space.

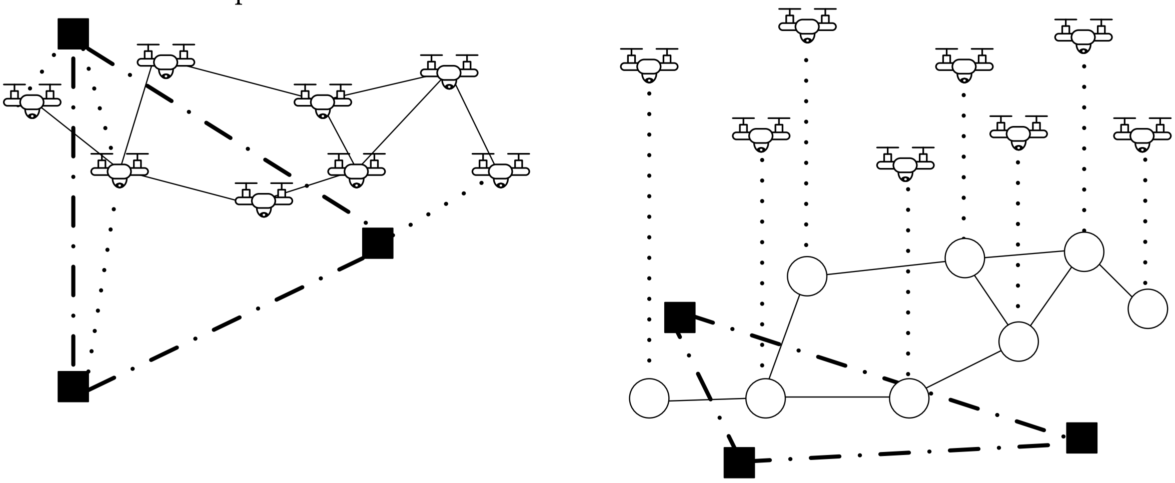

Figure 1: Mapping the graph formed by a swarm of UAVs into three-dimensional Euclidean space.

Let's consider *the problem formulation*:

$G(V,E)$ is a complete UAV graph, given

$n = |V|$ is the number of graph vertices, $V = \{v_1, \dots, v_n\}$ each vertex represents a swarm element, a separate UAV, which has a set of three coordinates

$(x_i, y_i, z_i), i = \overline{1,n}$ – coordinates of these vertices, where x_i, y_i are unknown, but z_i is known.

$E = \{e_{12}, e_{13}, \dots, e_{n-1,n}\}$ – the known lengths of the ribs.

Now consider the mapping of the graph from the three-dimensional Euclidean space to the problem on the plane (Figure 2).

It is easy to see from the figure that the length of the projection of the edge e_{ij} between the vertices v_i, v_j onto the XOY plane: $e'_{ij} = \sqrt{e_{ij}^2 - (z_j - z_i)^2}$

To form the desired configuration based on the swarm elements known coordinates, one must first choose the basic (largest) triangle. Its use will allow the formation of the desired configuration within the base triangle and will not require an increase in the maximum distance between any two swarm elements.

The basic triangle in the graph can be chosen according to the following characteristics:

- the largest sum of sides $a + b + c$, where a, b, c are the edges of the graph for some triplet of vertices (Figure 3):

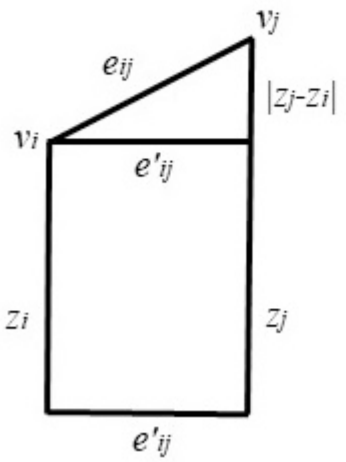

Figure 2: Mapping Reduction of a three-dimensional problem to a problem on a plane.

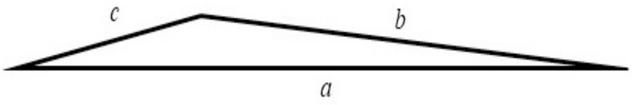

Figure 3: Basic triangle by the sum of the largest sides.

- the largest area $S = \sqrt{p(p-a)(p-b)(p-c)}$, where $p = \frac{a+b+c}{2}$is the semi-perimeter (Figure 4):

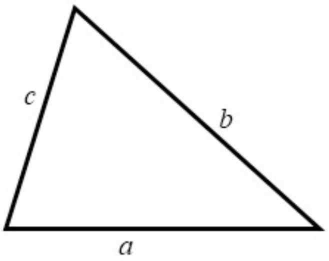

Figure 4: Basic triangle by the largest area.

In this work, the basic triangle with the largest area is used.

When choosing a base triangle, some swarm configurations will cause some swarm elements to be outside the base triangle (for example, a square for a four-element swarm). This is the standard situation; in the future, the position of the elements will be adjusted and they will move inside the base triangle.

4.2. Fixation of the basic triangle on the plane

After selecting the basic triangle, several degrees of freedom are preserved when placing the projection of the graph on the plane. Let's choose the following: the origin of the coordinate system (the first vertex of the triangle), the direction of the abscissa axis (the second vertex of the triangle), the direction of the ordinate axis (the third vertex of the triangle) – Figure 5.

Fix the coordinates of the vertex v_b as the origin of the coordinates: $v_b = (0{,}0)$

1. Let the side $v_b v_c$ be part of the axis OX: $v_c = (e'_{bc}, 0) = (a, 0)$
2. y_a– v_a determine the coordinate of the vertex from the area of the triangle:

$$S = \sqrt{p(p-a)(p-b)(p-c)} = \frac{a y_a}{2}$$

3. x_a– v_a we determine the coordinate of the vertex according to the Pythagorean theorem:

$x_a = \sqrt{c^2 - y_a^2}$

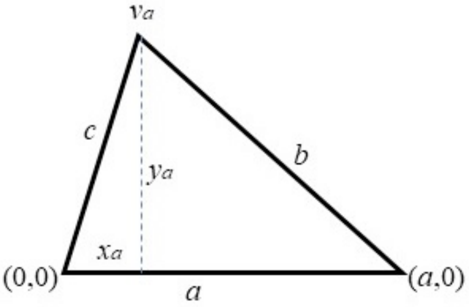

Figure 5: Choosing the swarm's coordinate plane.

To determine the coordinates of the remaining vertices of the graph on the plane, the multilateration method can be applied [10-13].

Since satellite positioning data is available, the task of maintaining the swarm structure can be reduced to maintaining the position of the UAV in each location of a geometric figure based on the coordinates' absolute values (Figure 6).

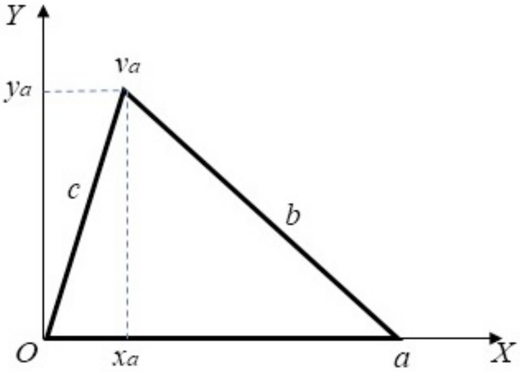

Figure 6: Determination of the basic triangle vertices coordinates.

But in the case of small distances be-tween UAVs, the accuracy of determining coordinates may be insufficient and the mini-mum distance between UAVs, set to prevent collisions, may be violated.

4.3. The problem of increasing the accuracy of local positioning

Global positioning often does not provide a sufficient level of accuracy for the task at hand – potential errors that, in the case of two fast-moving UAVs, can overlap and add up, causing swarm elements to collide. One of the options for solving this problem is the multilateration methods use – local positioning, which does not depend on global positioning satellite systems and is more accurate than GPS (see Figure 7) [14].

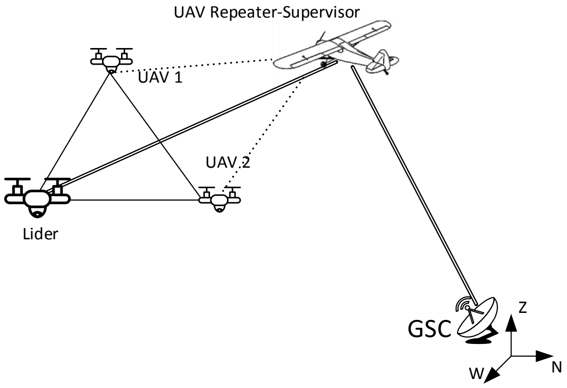

Figure 7: Illustration for the task of determining the co-ordinates of objects by the distances between them (Z – height, W – latitude, N – longitude).

Three drones in Figure 7 have their local system used for autonomous flight, and the ground station control (GSC) is tied to the absolute geographic coordinate system.

A simplified trilateration algorithm consists of the following steps:

Stage 0. Calibration of the distance be-tween nodes by power [15-16].

Input: Calibration based distance table.

Output: distance between nodes.

Stage 1. Each network node will receive information about the signal strength from other nodes and transmit its data to these nodes.

Stage 2. Each node for all visible nodes calculates distances according to the table and founds local coordinates by solving the system of trilateration equations. The received values are sent to all nodes [17].

The next stage is the application of triangulation algorithms.

4.4. A degenerate case of two drones

1. When creating a swarm of two, their mutual location and distance are determined. The parameters of the mutual local positioning means are stored, which corresponds to the sensors data for such mutual location, determined by the means of global positioning.
2. One of the drones becomes the main one, the master.
3. The master relays control commands from the operator to the slave (or slave receives them directly and repeats them). At the same time, a survey is conducted once every ten seconds (data exchange to determine the signal strength and the corresponding distance between the drones).
4. If the distance has changed, correction is made by changing the flight direction and re-measurement.

4.5. The problem in a three-dimensional setting

Let us now consider the three-dimensional formulation of the problem. We have up to twenty objects, the distances between them are known, and the GSC to which the distance is also known. In addition, all objects (1-20 pcs.) can be separately controlled remotely any time (that is, telemetry is received on the GSC and the distance is measured by signal strength and time from editing/receiving data packets) when the operator needs it. Absolute coordinates are known for GSC. Thus, we have the following formulation of the problem:

A matrix of distances between two to twenty-one points in three-dimensional space is given. For one of these points, its absolute coordinates are known. It is necessary to develop an algorithm for building a relative coordinate system for these points.

4.5.1. Algorithm of multidimensional scaling

The task of determining the coordinates of objects by the distances between them can be reduced to the well-known task of reducing the dimensionality of input data. A general approach to solving the problem of data dimensionality reduction consists of performing a sequence of steps to reduce the number of variables describing this data, and the result of the work of the corresponding algorithms is an ordered set of main variables.

The methods of multidimensional scaling/reduction of dimensions (Multidimensional Scaling – MDS) are applied to various types of data [18].

4.5.2. Mathematical formulation of MDS

The basic idea of MDS is to find a general process in a low-dimensional space that minimizes the variance D in the high-dimensional space and the dependent variance *d* in the low-dimensional space. Mathematically, this can be expressed as follows: given a dissimilarity or distance matrix *D*, representing the difference between points *i* and *j* in space, where D_{ij} is the Euclidean distance, MDS tries to find a set of coordinates x_i and x_j. This is done in a low-dimensional space so that the Euclidean distance between points *i* and *j* (denoted d_{ij}) is as close as possible to the difference D_{ij}.

The metric of the MDS algorithm is the stress function S (objective function), which defines the difference between the distances in the source space and the distances in the subspace. MDS for dimensionality reduction can be defined as follows:

$$S = \sqrt{\left[W_{ij} * \left(d_{ij} - D_{ij}\right)^{2}\right]}$$

where W_{ij} is the weight that can be used to emphasize or de-emphasize certain distances.
d_{ij} is the Euclidean distance between data points i and j in the lower dimensional space.
D_{ij} is the difference (distance) between data points i and j in a high-dimensional space.

The stress function S is defined as:

$$S = \sqrt{\frac{1}{n^{2}}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(l_{i,j} - \hat{l}_{i,j}\right)^{2}}$$

where l_{ij} is the distance between data points *i* and *j* in the original space, is the distance between data points *i* and *j* in the lower dimensional space, and *i* is the number of data points.

The stress function S is a measure of the deviation of the distances in the lower dimensional space from the distances in the original space and is used to evaluate the quality of the projection.

To create a relative coordinate system for the number of points from two to twenty-one in three-dimensional space based on the matrix of distances and absolute coordinates of one point, we used a modification of the MDS algorithm developed by us. Additionally, the output coordinates had to be adjusted to include the known absolute coordinates of a single point. At the same time, the emphasis was on performing calculations on single-board computers of the Raspberry type *Pi* 4 B onboard drones [19].

For the practical implementation of the task, a Python script was created based on the developed algorithm, which uses the Scikit-learn library for the case of 10 UAVs. They are in the local coordinate system and one control object is tied to the global coordinate system.

The MDS algorithm applied to the drone distance matrix allowed us to find the relative coordinates of ten points in three-dimensional space (the local coordinate system of the swarm) and then adjust these relative coordinates to include the known absolute coordinates of a single point (Figure 8, Figure 9).

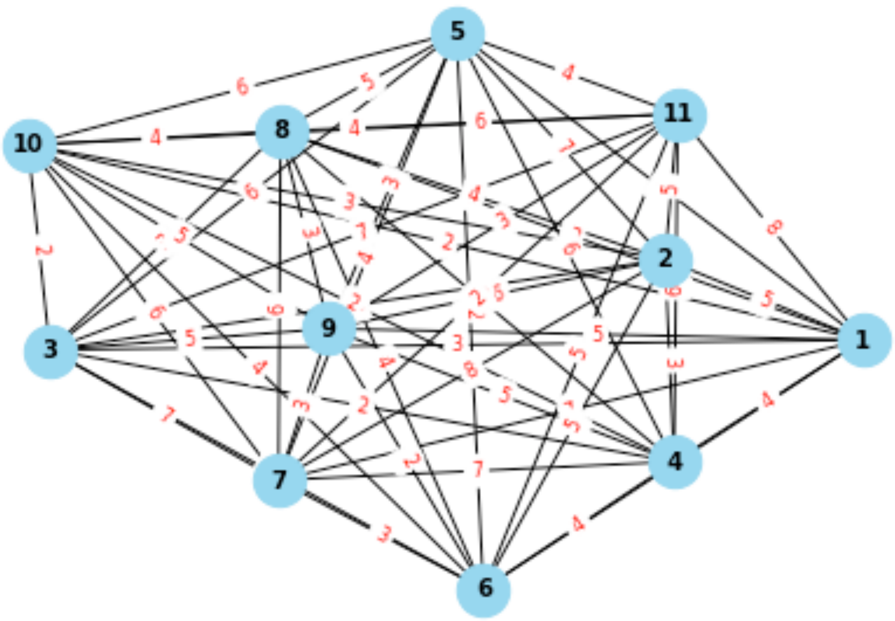

Figure 8: Representation of the matrix of distances be-tween eleven objects in the form of an undirected fully connected graph.

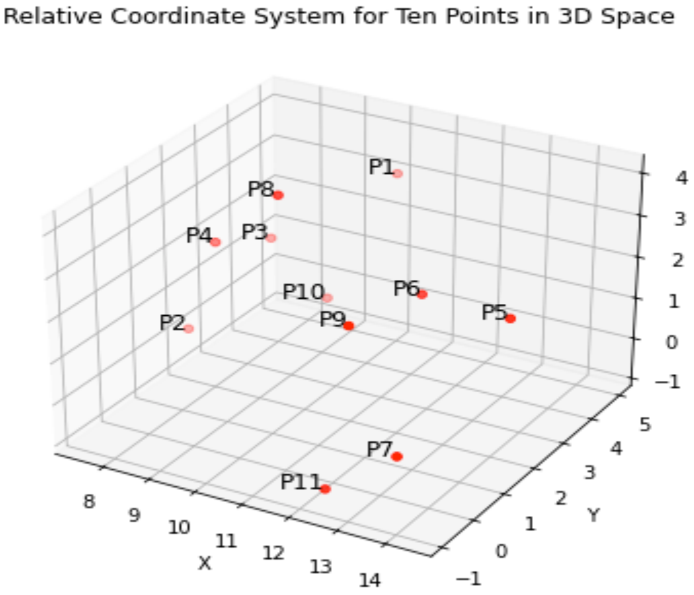

Figure 9: Representation of the distance matrix between eleven objects in the form of an undirected fully connected graph.

Thus, ten objects are in the local coordinate system without being tied to the absolute geographic coordinate system.

The developed script applies MDS to a distance matrix to find the relative coordinates of points (ten in the example) in 3D space and then adjusts those relative coordinates to include the known absolute coordinates of a single point.

Finally, it plots the points on a 3D plot using Matplotlib, with each point labeled for identification. You can configure the distances_matrix array and the absolute_coordinates variable to represent different distances between points and different known absolute coordinates, respectively (Figure 10, Figure 11).

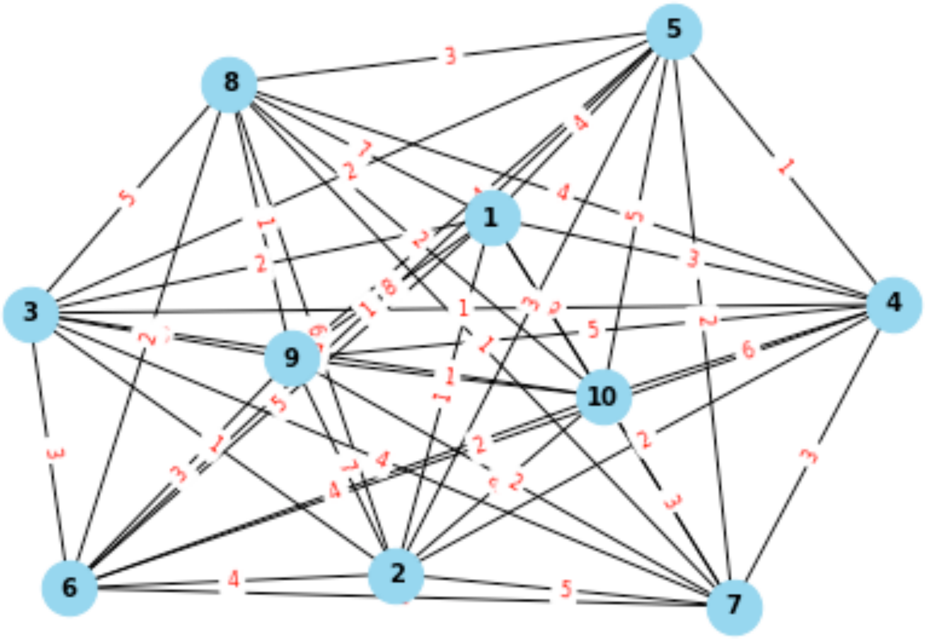

Figure 10: Representation of the distance matrix be-tween ten objects in the form of an undirected fully connected graph.

Thus, the modified MDS algorithm applied to the matrix of distances between drones allows finding the relative coordinates of points in three-dimensional space (local coordinate system of the swarm) for cases from one to twenty UAVs.

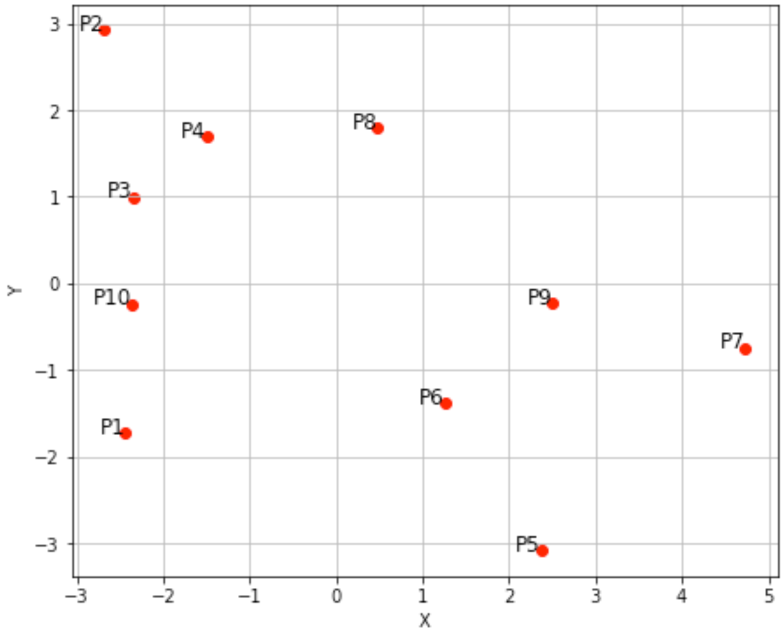

Figure 11: The result of solving the MDS problem for ten objects with the construction of a local coordinate system.

With its application, the construction of a local system for positioning swarm elements is less computationally complex than in alternative formulations [1-8].

The MDS algorithm applied to the matrix of distances between drones made it possible to find the relative coordinates of ten points in three-dimensional space (local coordinate system of the swarm). Experiments with other well-known dimensionality reduction methods, such as PCA, t-SNA, did not show satisfactory results for Euclidean metrics, since the idea of construction is based on the statistical characteristics of the data.

It should be noted that MDS should not be used to obtain the position of drones in the absolute coordinate system, since MDS does not allow obtaining the correct orientation in space and has reproduction errors even for the local coordinate system (as it is shown on example – Figure 12, Figure 13).

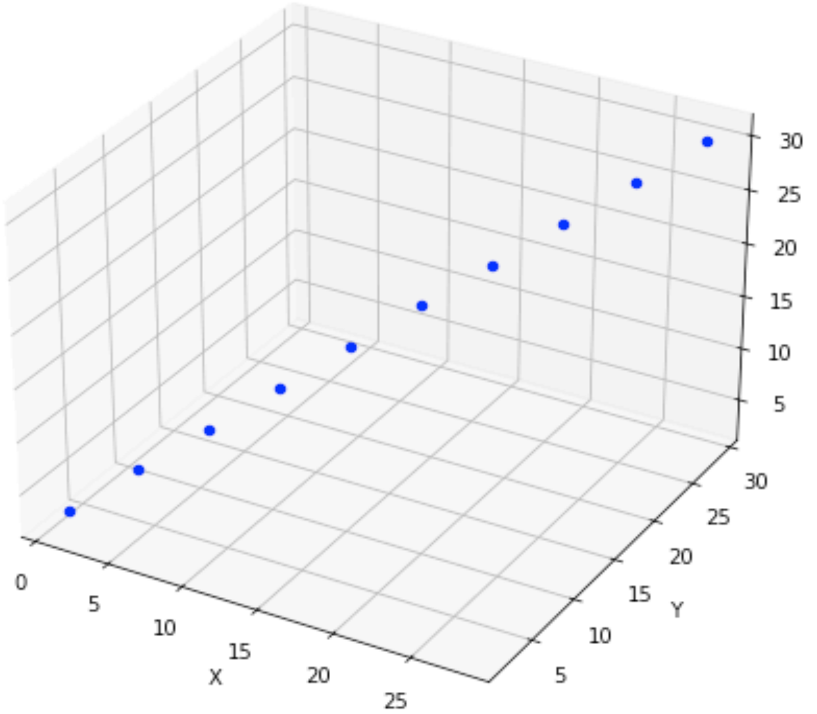

Figure 12: An example of a test for ten points.

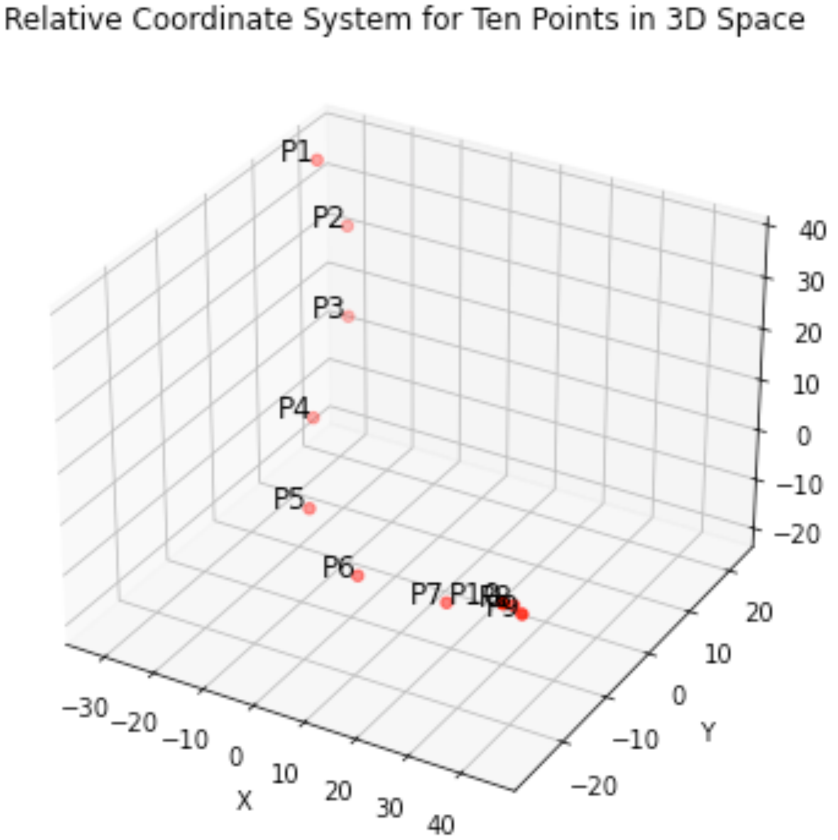

Figure 13: Example of reproduction of coordinates for the test presented in Figure 12.

4.6. Modeling the use of UWB devices for swarms of agricultural UAVs

The use of Ultra Wideband (UWB) radio modules in agricultural UAVs has the following features:

Using UWB on UAVs increases the communication range compared to indoors, but also increases the delay in receiving data. For plant and soil monitoring, it is advisable to increase the distance between swarm drones to more than 100 meters.

The use of radio modules on UAVs raises the question of the effect of Doppler shift on the speed, distance and stability of data transmission.

The duration of the UWB pulse is much shorter (less than 1 millisecond) [21] than the symbol transmission time in other technologies, so the UWB technology provides high accuracy (approximately 0.01 m) and low delay in determining the distance between radio modules. Positioning accuracy and delay for the most used technologies are [21]: Bluetooth (1-5 m, over 3 sec.), WiFi (5-15 m, over 3 sec.), RFID (1 m., 1 sec.) , GPS (5-20 m., 100 millisec.), 5G (10 m., less than 1 sec.).

UWB radio modules are based on the IEEE 802.15.4a protocol [22]. The Matlab 2024a software package contains tools for modeling the physical level of this protocol. Modern UWB radio modules use channel #5 (6489.6 MHz) and channel #9 (7987.2 MHz). Since the 6.5 MHz band is close to WiFi frequencies that can be suppressed by anti-unauthorized drone protection systems, consider the effect of Doppler shift on channel #9, which is less busy.

Typical speeds of drones are 50-70 km/h. To assess the influence of the Doppler shift, it is advisable to consider a three-fold reserve relative to the average speed. For the transmitter carrier frequency of 7987.2 MHz, the values of the Doppler shift will be as follows: 50 km/h: 370 Hz; 60 km/h: 444 Hz; 70 km/h: 518 Hz; 100 km/h: 740 Hz; 150 km/h: 1110 Hz. The maximum Doppler shift of 1110 Hz was considered in the Matlab 2024a UWB channel simulator [21] and showed the applicability of radio modules for the task of positioning a swarm of agricultural UAVs.

4.7. Analysis of advantages and disadvantages of statistical evaluation methods for improving drone navigation

Kalman filtering methods [23], particle filter methods [24], deep learning of neural networks [25] and combinations of the listed methods [24] are used to improve the navigation of individual drones or a swarm of drones.

1. Kalman filter:
- Estimates the state of a dynamic system based on a series of measurements.
- Advantages: accurate, optimal, recursive.
- Disadvantages: not suitable for non-linear systems.
- Possible variations for non-linear non-Gaussian systems.

2. Particle filter:
- Works with nonlinear and non-Gaussian systems.
- Advantages: universal.
- Disadvantages: Less accurate than the Kalman filter.

3. Neural networks:
- Can smooth out noise and emissions.
- Advantages: flexibility, adaptability.
- Disadvantages: difficulty of training, the result may not converge to the true values.

Resume:

- The Kalman filter gives the best results for typical tasks.
- Neural networks can be useful for noise smoothing.
- A particle filter can be used for complex nonlinear systems.

Figure 14 shows the result of positioning data filtering simulation for different types of filters with a noise variance of 10%. The FilterPy library for the Python language was used to model the filters, and TensorFlow was used to model the neural network.

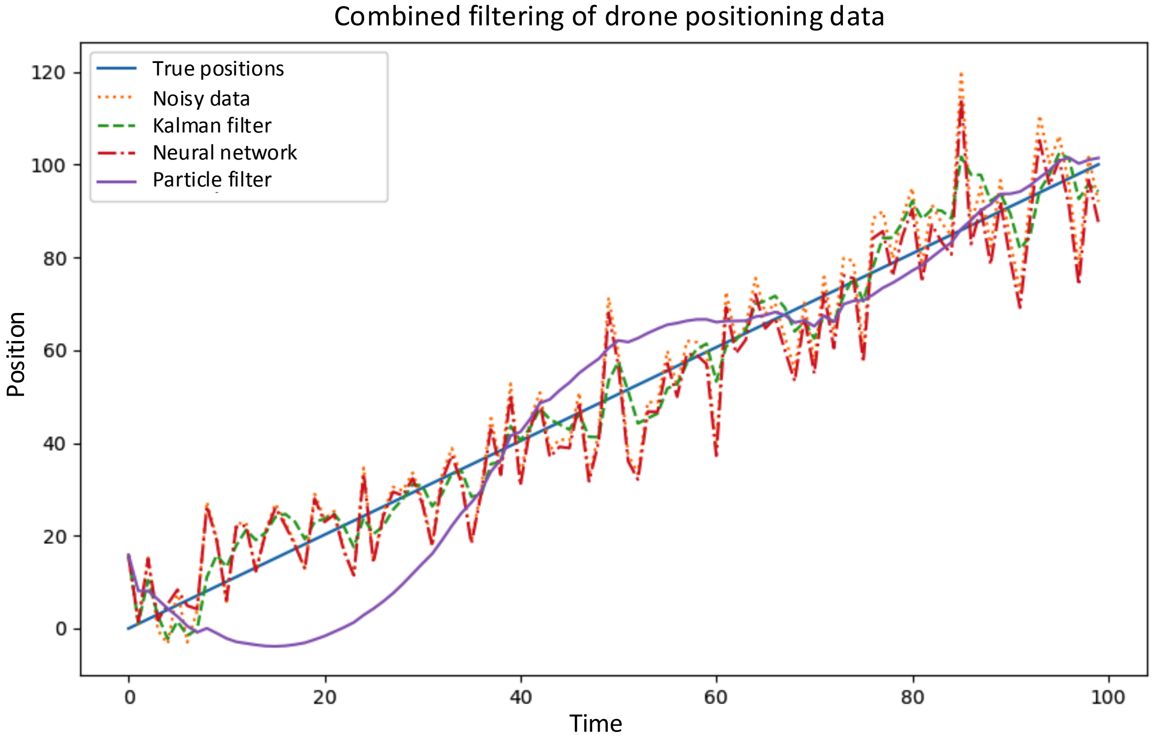

Figure 14: The result of modeling noisy navigation data and applying different types of filters.

Simulations have shown that the Kalman filter is the most stable and accurate when using the FilterPy library for Python and TensorFlow for neural networks. Neural networks and particle filters also showed significant results, but with some differences in accuracy and stability.

The following combined approaches are also used to improve the positioning and navigation of drones [26].

1. Integration with simultaneous localization and mapping technologies – Simultaneous Localization and Mapping (SLAM):

Use of Kalman filtering methods and its variants in combination with SLAM algorithms to improve simultaneous localization and mapping. This allows drones to navigate more effectively in unfamiliar or dynamic environments.

2. Use of hybrid methods:

Combination of deep learning methods with traditional filtering methods. For example, learning neural networks to predict nonlinear system dynamics and using Kalman filters for final correction. This increases positioning accuracy and stability.

3. Integration of multisensory information:

Fusion of data from different types of sensors: laser location (Light Detection and Ranging – LiDAR), cameras, inertial measurement devices using filtering methods. This approach allows to improve navigation accuracy in difficult conditions where one type of sensor may not be effective enough.

4. Advanced optimization methods:

Use of adaptive algorithms, such as Particle Swarm Optimization (PSO) or genetic algorithms, to adjust filter parameters in real time. This makes it possible to increase the adaptability and robustness of the filter in conditions of a changing environment.

Conclusion: The Kalman filter showed the best result, stable over several simulation runs. The neural network provided little smoothing of the outliers of the random process, while the particle filter demonstrated over-smoothing for this example. However, in other simulations, its convergence to true values was at the level of a neural network. The integration of new methods and technologies, such as multi-sensor fusion, can significantly improve the accuracy and reliability of drone navigation in various environments.

5. Conclusions

As part of the work, the task of forming and maintaining the structure and configuration of a flock of agricultural UAVs in two-dimensional and three-dimensional settings was set. The formalization of the given task has been completed. Considered possible methods of mapping the graph formed by a swarm of UAVs into a two-dimensional Euclidean space by forming a basic triangle to create a relative coordinate system. To determine the coordinates of the graph's remaining vertices on the plane and correspondingly increase the accuracy of local positioning, the multilateration method is used and considered simplified options − application of the trilateration algorithm, triangulation and a simplified algorithm for the degenerate case of two drones.

For the problem in a three-dimensional setting, the possibility of applying the multidimensional scaling algorithm using the methods of multidimensional scaling/reduction of dimensions is considered. The developed practical implementation of the created algorithm showed its efficiency during practical experiments, allowing it to determine the swarm elements local positioning, containing one to twenty elements.

The direction of further research is the development of an algorithm that allows, based on determined local coordinates, to eliminate disturbances introduced into the swarm structure due to internal or external factors, for example, the loss of one of the swarm elements [20].

References

[1] Y. Zhou, R. Bin, W. Wei, UAV swarm intelligence: Recent advances and future trends, IEEE Access 8 (2020): 183856-183878. doi: 10.1109/ACCESS.2020.3028865

[2] M. Campion, R. Prakash, F. Saleh, UAV swarm communication and control architectures: a review, Journal of Unmanned Vehicle Systems, 2018. 7.2. pp. 93-106. doi: 10.1139/juvs-2018-0009

[3] Modern methods of digital monitoring in crop production: monograph / Zozulya O.L. etc. K.: KNU named after Taras Shevchenko, 2023. 254 p.

[4] X. Wang, G. Chen, H. Gong, J. Jiang, UAV swarm autonomous control based on Internet of Things and artificial intelligence algorithms, Journal of Intelligent & Fuzzy Systems, 2021. 40(4). pp. 7121-7133. doi: 10.3233/JIFS-189541

[5] N.R. Frantz, Swarm intelligence for autonomous UAV control. Diss. Monterey California. Naval Postgraduate School, 2005.

[6] V. Roberge, M. Tarbouchi, G. Labonté, Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. IEEE Transactions on industrial informatics, 2012. 9(1). pp. 132-141. URL: http://dx.doi.org/10.1109/TII.2012.2198665

[7] S. Shao, Y. Peng, C. He, Y. Du, Efficient path planning for UAV formation via comprehensively improved particle swarm optimization. ISA transactions , 2020. 97. Pp. 415-430. doi: 10.1016/j.isatra.2019.08.018

[8] M.D. Phung, Q.P. Ha, Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. Applied Soft Computing, 2021. 107. p. 107376. http://dx.doi.org/10.1016/j.asoc.2021.107376

[9] D. Sikeridis, E. EleniTsiropoulou, M. Devetsikiotis, S. Papavassiliou, Self-adaptive energy efficient operation in UAV-assisted public safety networks, 2018 IEEE 19th international workshop on signal processing advances in wireless communications (SPAWC) 2018, June, IEEE. 2018. pp. 1-5. URL: https://sigport.org/documents/self-adaptive-energy-efficient-operation-uav-assisted-public-safety-networks-3

[10] S.N. Othman, Node positioning in ZigBee network using trilateration method based on the received signal strength indicator (RSSI). European Journal of Scientific Research, 2010. 46(1). pp. 048-061. URL: https://godieboy.com/wp-content/uploads/2012/05/ejsr_46_1_05.pdf

[11] Y. Zheng, Y. Xie, J. Li, Multi-UAV Collaboration and IMU Fusion Localization Method in Partial GNSS-denied Scenarios. IEEE Access, 2023. P. 105499. http://dx.doi.org/10.1109/ACCESS.2023.3315870

[12] O. Hernandez, V. Jain, S . Chakravarty, P. Bhargava, Position location monitoring using IEEE 802.15.4, ZigBee technology. 2009. URL: http://www. freescale. com_files_microcontrollers_doc_brochure_PositionLocationMonitoring. pdf

[13] S.C. Kim, J.K. Lee, US, Patent No. 8,478,292. Washington, DC: US Patent and Trademark Office. 2013.

[14] F. He, T. Zhou, W. Xiong, SM Hasheminnasab, A. Habib, Automated aerial triangulation for UAV-based mapping, Remote Sensing, 10(12). 2018, p. 1952. URL: http://dx.doi.org/10.3390/rs10121952

[15] J. Yan, H. Zhao, X. Luo, C. Chen, X. Guan, RSSI-based heading control for robust long-range aerial communication in UAV networks, IEEE Internet of Things Journal, 2018. 6(2). pp. 1675-1689. URL: http://dx.doi.org/10.1109/JIOT.2018.2875428

[16] D. Lymberopoulos, Q. Lindsey, A. Savvides, An empirical characterization of radio signal strength variability in 3-D IEEE 802.15. 4 networks using monopole antennas, Wireless Sensor Networks: Third European Workshop, EWSN 2006, Zurich, Switzerland, February 13-15, Springer Berlin Heidelberg, 2006, Proceedings 3, Pp. 326-341. doi: 10.1007/11669463_24 URL: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/lymberopoulos_EWSN_06.pdf

[17] B. Yang, E. Yang, L. Yu, A. Loeliger, High-precision UWB-based localization for UAV in extremely confined environments, IEEE Sensors Journal, 2021. 22(1). pp. 1020-1029. doi: 10.1109/JSEN.2021.3130724 URL: https://pureportal.strath.ac.uk/files/131073493/Yang_etal_Sensors_2021_High_precision_UWB_based_localisation_for_UAV_in_extremely_confined_environments.pdf

[18] R. Chen, B. Yang, W Zhang, Distributed and collaborative localization for swarming UAVs, IEEE Internet of Things Journal, 2020. 8(6). pp. 5062-5074. doi: 10.1109/JIOT.2020.3037192

[19] M. Ogurtsov, V. Korolyov, O. Khodzinskyi, Improving the Productivity of UAV Operations Based on Granular Computing and Fuzzy Sets, 2021 IEEE 6th International Conference on Actual Problems of Unmanned Aerial Vehicles Development (APUAVD) 19-21 Oct. 2021. 2021, pp. 33-36. URL: https://doi.org/10.1109/APUAVD53804.2021.9615419

[20] V.P. Horbulin, L.F. Hulianytskyi, I.V. Sergienko, Planning of Logistics Missions of the “UAV+Vehicle” Hybrid Systems, Cybernetics and System Analysis, 59, 2023. pp. 733–742. URL: https://doi.org/10.1007/s10559-023-00609-8

[21] What Is Ultra-Wideband (UWB)? URL: https://www.mathworks.com/discovery/ultra-wideband.html

[22] "IEEE Standard for Low-Rate Wireless Networks--Amendment 1: Enhanced Ultra Wideband (UWB) Physical Layers (PHYs) and Associated Ranging Techniques," in IEEE Std 802.15.4z-2020 (Amendment to IEEE Std 802.15.4-2020), pp.1–174, 25 August 2020, doi: 10.1109/IEEESTD.2020.9179124.

[23] C.K. Chui, G. Chen. Kalman Filtering: with Real-Time Applications. Springer Cham. Cham. Switzerland. 2017. p. 247. URL: https://doi.org/10.1007/978-3-319-47612-4

[24] L.A. Klein. Sensor and Data Fusion: A Tool for Information Assessment and Decision Making, Second Edition. SPIE -- The International Society for Optical Engineering. Washington. USA. 2012. p. 512. URL: https://doi.org/10.1117/3.928035

[25] C.C. Aggarwal. Neural Networks and Deep Learning. Springer Cham. Cham. Switzerland. 2023. p. 529.URL: https://doi.org/10.1007/978-3-031-29642-0

[26] Drones, Volume 8, Issue 4 (April 2024) – 51 articles. Drones, EISSN 2504-446X, Published by MDPI. URL: https://www.mdpi.com/2504-446X/8/4