Movement of a group of unmanned aerial vehicles in formation

E A Heiss
Tula State University, 92, Lenin prospekt, Tula, 300012, Russia
E-mail: edheiss78@gmail.com

Abstract. This paper presents a method for organising the movement of unmanned aerial vehicles in formation. In the presented method of UAV collisions are prevented by a modification of the potential field method. The movement of the group is accomplished by incrementing the origin coordinates of the formation at each iteration of data exchange between UAVs. This paper presents a graph showing the negligible positioning error of drones in formation over time as the heterogeneous swarm of quadcopters moves towards the target position. Simulation results show that the flight time of a group of UAVs to the target swarm position using the proposed method is longer than using the master-slave method. The model of a Wi-Fi network as an information exchange network between UAVs was used in the developed simulation environment. This approach facilitates the implementation of the presented method on real aircrafts.

1. Introduction
Currently, multi-agent systems are increasingly being used for problem solving [1, 2]. For example, groups of unmanned aerial vehicles (UAVs) can be used for area photography, intelligence gathering, and targeting enemy targets. For this kind of mission to be successful, UAVs need to form a formation. The formation of UAVs in this paper refers to their mutual order in which a certain distance is maintained between the vehicles.

In a previous paper [3], a method was developed to ensure uniform distribution of UAVs within a given area. However, when the swarm reaches a steady state, the mutual arrangement of the drones remains chaotic. Also, this method lacks a component to ensure that the entire UAV group moves to a given position while maintaining the geometric structure.

One of the common solutions to the problem of UAV positioning in the system is the potential field method [4–6], the master-slave method [7–11], the swarm intelligence method [12], and the local voting method [13, 14]. Binary tree elements can be used to form the structure of the UAV [15]. To date, there are commercial proposals that provide formation of multiple UAVs using master-slave method.

2. Description of the master-slave method
In this paper, an agent is understood as an unmanned aerial vehicle (UAV) of the ATV type in a group capable of information interaction. The master-slave method for a quadcopter is somewhat simplified compared to an aircraft-type UAV due to the greater manoeuvrability of the former. The control system component responsible for collision avoidance is omitted for
simplicity, and agents are assumed to start moving from group 2 by 5 drones. In this paper, the control object model is of fourth order and the angular position and linear motion control system is a PID controller.

There is \( N = \{1, 2, \ldots, n\} \) number of agents. The target position for the first UAV in the group is the target position of the whole group. For the second UAV, the target position is the sum of the position of the first UAV and its velocity vector, as well as the target distance between the agents:

\[
\vec{N}_i = \vec{P}_{i-1} + \vec{V}_{i-1} \cdot T, \quad i = \{2, 3, \ldots, n\},
\]

where \( \vec{P}_i \)—the position of the \( i \)-th agent, \( \vec{V}_i \)—is the speed of the \( i \)-th agent, \( T \) —data exchange period.

The normalised directional vector has the form:

\[
\hat{d}_i = \frac{\vec{N}_i}{\|\vec{N}_i\|}.
\]

In this method, the slave keeps a set distance \( r \) from the master:

\[
\vec{L}_i = (r - \|\vec{N}_i\|) \cdot \hat{d}_i.
\]

It is also necessary to limit the speed of the first agent slightly more than that of the others to compensate for the constant error in the relative position of master and slave. Thus, the vector coming to the input of the position control system for the slave UAVs is of the form:

\[
\vec{Y} = \begin{cases} 
L_{\text{max}} \frac{\vec{L}}{\|\vec{L}\|} & \text{if } \|\vec{L}\| < L_{\text{max}}, \\
\vec{L} & \text{otherwise,}
\end{cases}
\]

where \( L_{\text{max}} \)—the maximum vector length for the position control system.

This method does not have a collision avoidance component. It also involves the creation of a hierarchy and has some limitations on the geometry, which makes it difficult to use in multi-agent systems. Thus, the aim of the work is to develop a method that ensures the mutual ordering of UAVs as well as their movement in formation.

The following tasks must be solved to organize the movement of an agent in a swarm:

- Create a method of UAV movement as part of a swarm with low network bandwidth to provide:
  - collision prevention;
  - the formation of a given geometric structure by the agents;
  - moving the entire group without disturbing the geometry of the structure.
- Compare the performance of the resulting method with the master-slave method.

The task should be solved under the following conditions:

- the number of swarm elements is unknown in advance and may change over time;
- the swarm is heterogeneous, consisting of quadcopters with different dynamic characteristics;
- the turbulence created by propeller groups of ATVs is not taken into account;
- the coordinates of an agent in space are determined without interference;
- the distance between the agents is known in advance.
3. Description of the proposed method

The potential field method with some changes is used to organize agent interaction. In this case, the potential field method calculates the position of the virtual leader of each actor the UAV is aiming at. Each agent has an internal state—the radius used to calculate the virtual leader position. Two agents interact with each other if the sum of their radii is greater than the distance between them:

$$ R_1 + R_2 \geq D, \quad (5) $$

where $D$—distance between agents.

Each agent has $N_j = \{1, 2, \ldots, j\}$ number of neighbors. The resulting direction vector that takes into account neighbors’ positions is calculated as follows:

$$ \vec{L}_N = k \sum_j (R + R_j - D_j) \cdot \hat{d}_j, \quad (6) $$

where $k$—certain tilt coefficient, $R$—own radius, $R_j$—neighbor radius, $D$—distance between agents, $\hat{d}_j$—direction vector from neighbor.

The described control system component solves the problem of UAV collision avoidance. In this case, the possibility of overlapping motion paths is not taken into account, which entails a reduction in the efficiency of the overall method.

Let the formation have a triangular shape arranged in a horizontal plane. All free space inside the triangle can be represented as a set of ordered points located at a given distance from each other. Each point has its own index indicating the row and column. Figure 1 shows the grey and white colours respectively for unacceptable and acceptable positions for UAVs in the formation. In addition, white dots can be either free or occupied. The busy state is determined by the presence of an agent in the vicinity of the point. If there is more than one UAV in the vicinity of a point, the point is occupied by the agent whose distance from the point is the shortest.

Each agent has a target point at any given time, which is defined as those coordinates of space that are taken into account when calculating the virtual leader’s position. Also, the agent at any given time occupies a point called a position point. An agent has a list of nearby points to the

![Figure 1. Grid of UAV positions in formation.](image-url)
position point, called an environment. In this paper, six points are included in the environment. The target points are selected according to the following rules:

• if there are free points in the environment, select the one that is closer to the beginning of the line, i.e. to the point with the index \( \{0,0\} \);
• if all points around are unacceptable, move to the start of the formation;
• if there are no free ones, select the one with the highest row index and the lowest column index.

The position point index is determined as follows:

\[
m = \left\lfloor \frac{s_z}{a_{mz}} \right\rfloor, \tag{7}
\]

\[
n = \left\lfloor \frac{s_x - m \cdot a_{mx}}{a_{nx}} \right\rfloor, \tag{8}
\]

where \( m \)—the row index, \( n \)—the column index, \( \{s_x, s_z\} \)—the coordinates of the agent relative to the start of the formation in the formation plane, \( \vec{a}_m \) and \( \vec{a}_n \)—the coordinate increments of a point, \([ \ ]\)—rounding operator to the nearest integer.

The point coordinate increments are determined based on the given distance between the agents:

\[
\vec{a}_m = \{-0.5 \cdot r, 0, 0.866 \cdot r\}, \tag{9}
\]

\[
\vec{a}_n = \{r, 0, 0\}. \tag{10}
\]

The distance of the agent from the target point is determined based on the row and column indices:

\[
\vec{L}_B = \vec{P}_0 + \vec{P}_p - \vec{P}, \tag{11}
\]

where \( \vec{P}_0 \)—the coordinates of the start of the formation, \( \vec{P}_p \)—removal of the agent’s target point from the start of the formation, \( \vec{P} \)—their own position in space.

The distance of the agent’s target point from the start of the formation is calculated as follows:

\[
\vec{P}_p = m \cdot \vec{a}_m + n \cdot \vec{a}_n. \tag{12}
\]

The resulting directional vector of the virtual leader is limited:

\[
\vec{Y} = \begin{cases} L_{\text{max}} \hat{d}, & \text{if } \|\vec{L}_N + \vec{L}_B\| < L_{\text{max}}, \\ \vec{L}_N + \vec{L}_B, & \text{otherwise}, \end{cases} \tag{13}
\]

where \( L_{\text{max}} \)—the maximum distance from the agent to the virtual leader, \( \hat{d} \)—normalised directional vector \( \vec{L}_N + \vec{L}_B \).

The target point coordinates obtained are used by the UAV control system together with the coordinates obtained by the collision warning component.

Movement of the UAV formation is carried out by changing the position of the beginning of the formation towards the target position of the whole group (figure 2). The formation is said to have reached the target position when the point with index \( \{0,0\} \) is in some predetermined vicinity of the target position (TP).

The formation movement is accomplished by incrementing the origin coordinates of the formation towards the target position. The origin coordinate increment is carried out by each agent with the frequency of data exchange. Since the dynamic characteristics of the agents may
Figure 2. Diagram of UAV movement in formation towards the target position.

differ, there is a need to synchronise the start of the formation. Synchronization is performed by averaging the eigenvalues of the row origin coordinates with the values of the neighbors:

\[
P_{0}^{k+1} = \frac{1}{j+1} \left( P_{0}^{k} + \sum_{j} P_{0j}^{k} \right),
\]

(14)

where \( k \)—calculation iteration step, \( P_{0}^{k} \)—eigen-coordinates of the origin formation, \( P_{0j}^{k} \)—coordinates of the neighbour’s origin formation.

The speed synchronization looks similar:

\[
\vec{V}^{k+1} = \frac{1}{j+1} \left( \vec{V}^{k} + \sum_{j} \vec{V}_{j}^{k} \right),
\]

(15)

where \( k \)—calculation iteration step, \( \vec{V}^{k} \)—the intrinsic speed of the onset increment the intrinsic speed of the origin formation increment, \( \vec{V}_{j}^{k} \)—the increment rate of the neighbor.

The incremental vector of the origin formation is limited in length and depends on the period of data exchange:

\[
\vec{V}_{e} = \begin{cases} 
V_{\text{max}} T \hat{d}_{V}, & \text{if } \| \vec{V} \| < V_{\text{max}} T, \\
\vec{V} & \text{otherwise}
\end{cases},
\]

(16)

where \( V_{\text{max}} \)—the maximum speed of the origin formation increment, \( \hat{d}_{V} \)—direction from the origin formation to the target position, \( T \)—the period of data exchange between agents.

4. Modelling

Software has been created for modeling the method. The Quadrocopter model for linear motion changes the values of roll, pitch and yaw angles; motors are represented by a gain. The software also allows modeling data transfer between agents via Wi-Fi network, where at each moment of time only one agent performs broadcasting. Time of transmission of one message over the network is 6 ms. The data exchange between the agents is performed with a period of 0.2 s. In the simulation 10 agents were used, and the value of the set distance between UAVs is 15 m.

The quality criterion for the algorithm is the error of the set distance between neighbours as the formation moves towards the target position. Tracking the distance between neighbours in this method is done indirectly by tracking the target point and synchronising the start of
Figure 3. Positioning error of the UAV relative to the target point.

The simulation results are shown in figure 3, where $n$—simulation environment calculation iterations, $E$—UAV positioning error relative to the target point.

The graph shows that the maximum deviation during movement is just over 1 metre. So, the method allows for fairly accurate positioning of UAVs in formation, given the fact that the dynamics of the agents are different and the frequency of data exchange over the Wi-Fi network is low.

The two methods were compared in the following way. At the initial moment of time, the UAV group is located at the origin of coordinates in the horizontal plane. The group then starts moving by a pre-determined method over a predetermined distance. The movement time of the entire group to the set position is recorded in the Table 1.

| Distance between the group’s target position and the starting point, m | Group travel time to target position, s |
|---|---|
| The master-slave method | The proposed method |
| 500 | 185 | 184 |
| 1000 | 324 | 347 |
| 2000 | 602 | 670 |

Table 1 shows that at short distances the flight times are almost the same, but with increasing distance the difference between the two methods grows. The advantage of the proposed method at short distances can be explained by the fact that at the initial moment of time all agents start moving towards the target position simultaneously. In the current master-slave implementation, the slave agent does not start moving until there is a distance error from the master, which causes the entire group to stretch into a string.

5. Discussion

Of course, the use of this method is somewhat limited. For example, an aircraft-type UAV, for which many more suitable algorithms have been developed, has difficulty with a sharp change in direction of motion in contrast to the ATV. Therefore, the aircraft type UAV will need to upgrade the method presented.

In addition, this method limits the movement rate of the entire group. The maximum group movement speed is directly proportional to the frequency of data exchange and a given distance
between agents. If the speed of the group is too high, the UAV may be displaced from its position in the group, which may cause the geometric structure to collapse. One feature of the proposed method is noticed during the simulation. Scaling the swarm by adding new agents in the vicinity of the beginning of the formation can lead to reconfiguration of the entire geometric structure, which is not always appropriate.

6. Conclusion
Thus, a method is presented to ensure that the UAV is positioned in formation, and that the entire group moves towards the target position. It goes without saying that a method correction will be required if an aircraft-type UAV is used. Nevertheless, when using a quadcopter-type UAV, the method shows good performance even with a relatively low data rate. With the appropriate communication hardware, this method makes swarm scaling relatively easy. A comparison between the master-slave method and the proposed method is also made.

References
[1] Rastgoftar H and Atkins E M 2018 Aerospace Science and Technology 82 105–18
[2] Xu Chu Ding, Rahmani A and Egerstedt M 2010 IEEE Transactions on Robotics 26 256–68
[3] Heiss E A 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1029 012081
[4] Muslimov T Z 2019 Algorithms for controlling the building of autonomous unmanned aircraft type using the vector field method Systems of Control, Communication and Security 4 187–214
[5] Medvedev M Yu and Lazarev V S 2016 A method for planning the movement of a group of moving objects using dynamic repellers and target distribution Scientific Bulletin of NSTU 66 41–52
[6] Chen Y B, Luo G C, Mei Y S, Yu J Q and Su X L 2014 Int. J. Syst. Sci. 45 1407–20
[7] Tang J, Piera M A and Guasch T 2016 Transp. Res. Part. C: Emerg. Technol. 67 357–77
[8] Ruchti J, Senkbeil R, Carroll J, Dickinson J, Holt J and Biaz S 2014 J. Aerosp. Inf. Syst. 11 140–4
[9] Efroev V N, Mizin S V and Neretina V V 2014 Flight control of uav stroyuna based on coordination of interaction of a group of aircraft Vestnik UGATU 62 114–21
[10] Ilichev K V and Mantserov S A 2017 Development of a scalable mobile robotic system PNRPU Bulletin. Electrotechnics, Informational Technologies, Control Systems 21 91–108
[11] Karpov V E and Karpova I P 2014 Procedia Engineering 100 1459–68
[12] Ivanov D Ya 2011 Using the principles of swarm intelligence to control the purposeful behavior of mass-used microrobots in extreme conditions Proceedings of Higher Educational Institutions.chine Building 9 70–8
[13] Efroeva V A, Ivanskiy Yu V and Kijyaev V I 2015 Controlling a swarm of dynamic objects based on a multi-agent approach Computer Tools in Education journal 6 34–42
[14] Jiabro Ren, Baofang Wang, Mingjie Cai and Jinping Yu 2020 IEEE Access 8 25557–69
[15] Daifeng Zhang and Haibin Duan 2019 Journal of the Franklin Institute 356 835–59