Stanisław KONATOWSKI[1], Sebastian TATKO[2]

Military University of Technology, Faculty of Electronics, Institute of Radio electronics
ORCID: 1. 0000-0002-3871-926X; 2. 0000-0001-9603-427X

doi:10.15199/48.2023.05.10

Behaviour of unmanned aircraft in formation

Abstract. *For the described model of the dynamics of the unmanned aircraft, made in the Matlab/Simulink environment, an innovative method of controlling the UAV formation and the laws of dynamics governing the complex system were presented. The software implementation of the UAV model shows the process of building the system and selecting the settings of MPC, PID and LQR regulators. The proposed swarming algorithm is based on measuring mutual distances and bearings between individual aircraft. The correctness of the algorithm's operation was tested in a simulation manner with the use of a large list of characteristics. The article also includes an analysis of the influence of measurement uncertainty on the formation behaviour in flight.*

Streszczenie. *Dla opisanego modelu dynamiki bezzałogowego statku powietrznego, wykonanego w środowisku Matlab/Simulink, przedstawiona została nowatorska metoda sterowania formacją BSP oraz prawa dynamiki rządzące układem złożonym. Implementacja programowa modelu BSP ukazuje proces budowy układu oraz doboru nastaw regulatorów MPC, PID oraz LQR. Zaproponowany algorytm rojowy bazuje na pomiarze wzajemnych odległości oraz namiarów między poszczególnymi statkami powietrznymi. Poprawność działania algorytmu została sprawdzona w sposób symulacyjny z wykorzystaniem licznego zestawienia charakterystyk. Artykuł zawiera także analizę wpływu niepewności pomiarów na zachowanie formacji w locie.* ***(Zachowanie się bezpilotowych statków powietrznych w formacji)***

Keywords: drone, Unmanned Aerial Vehicle (UAV), swarm, formation, behaviour of the swarm, swarm algorithm.
Słowa kluczowe: dron, Bezzałogowy Statek Powietrzny (BSP), rój, formacja, zachowanie się roju, algorytm rojowy.

Introduction

In recent years, the topic of unmanned aerial vehicles has become one of the most popular threads in the world of technology and science. The increasingly thriving electronics industry is presenting us with new drone applications year after year. Initially seemingly small, toy-like devices, over the years they have acquired greater size, enormity of functionality and autonomy.

The created potential of a single unmanned unit now contributes to new technologies and swarming algorithms. Drone formation is a relatively young field of knowledge that is currently being studied by many scientists and engineers [1-5,9]. Mapping the laws of nature is of great interest to the defence industry. The ability to search vast terrain, conduct large-scale reconnaissance, and the striking potential of UAVs are just some of the positives that the armies of many countries recognise. Minimising human interference in the operation of the system is also a very important factor. Such characteristics of the formation in the current reality make it possible for one soldier to perform a complex task, which was previously performed by a group of military men.

This paper presents the idea and software implementation of a novel UAV formation control algorithm. The main idea is to use procedures and algorithms that are commonly available, in which the acquisition of data during flight does not involve the use of complex measurement systems. For this purpose, the drones use only information about the distance from each other and mutual angular bearing for formation control. Such an assumption is mainly supported by the fact that the current electronics market offers a wide range of devices that can determine such flight data without problems.

The basis of the system's operation is the described mathematical model of the quadcopter and the flight controller system used. The flight controller was designed based on the application of three independent control techniques.

Mathematical representation of the Quadcopter

The quadcopter model used in this paper is an X-type structure, often called a plus configuration. In this architecture, one engine is assumed to be the front of the UAV. Another strategy is the cross configuration, in which two engines are adopted as the front of the UAV.

Figure 1 shows the initial position of the UAV, the torques M and thrust forces f of the individual motors, and the angles of roll Φ, pitch θ, and yaw ψ assigned to each axis.

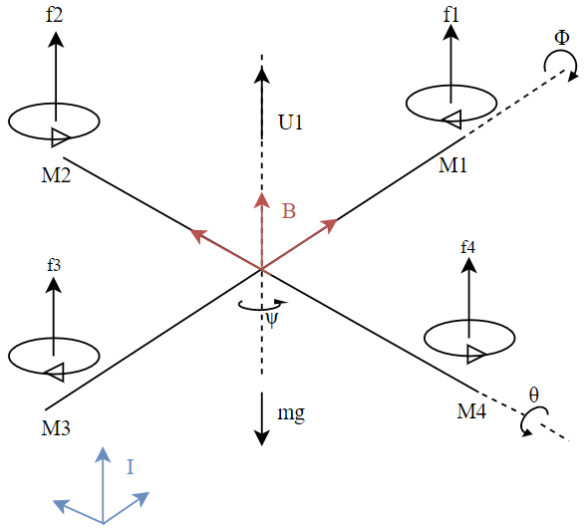

Fig. 1. Position and orientation of the quadcopter (own elaboration based on [3,4])

In the diagram shown, two types of coordinate systems are indicated. The first is a local system associated with the centre of gravity of the aircraft $\vec{\mathbf{B}} = [\vec{X}_L, \vec{Y}_L, \vec{Z}_L]$, however $\vec{\mathbf{I}} = [\vec{X}, \vec{Y}, \vec{Z}]$ means the inertial reference system related to the Earth. The orientation of the UAV is determined by the orthonormal rotation matrix **R**, which is also often referred to as the directional cosine matrix DCM. The knowledge of the drone's coordinates in one coordinate system and the DCM matrix allows a smooth transition to a second type of coordinate system [4-5] according to the relation

$$\mathbf{R} = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\Phi - s\psi s\Phi & c\psi s\theta c\Phi + s\psi c\Phi \\ s\psi c\theta & s\psi s\theta s\Phi + c\psi c\Phi & s\psi s\theta c\Phi - c\psi s\Phi \\ -s\theta & c\theta s\Phi & c\theta c\Phi \end{bmatrix}, \quad (1)$$

where c is the cosine function and s the sine function.

The quad-rotor UAV model analysed is a system controlled by independent speed $U_{1\text{-}4}$ of individual engines

with thrust b and rotation speeds $\Omega_{1\text{-}4}$. In order to make the model simpler and to get rid of redundant systems, such as the motor mixing algorithm, the following relationship was used

$$(2)\qquad \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b({\Omega_1}^2 + {\Omega_2}^2 + {\Omega_3}^2 + {\Omega_4}^2) \\ b({\Omega_4}^2 + {\Omega_3}^2 - {\Omega_1}^2 - {\Omega_2}^2) \\ b({\Omega_2}^2 + {\Omega_3}^2 - {\Omega_1}^2 - {\Omega_4}^2) \\ b({\Omega_1}^2 + {\Omega_3}^2 - {\Omega_2}^2 - {\Omega_4}^2) \end{bmatrix},$$

The signal U_1 is responsible for the hover of the UAV and is the value of the sum of the squares of the speeds of all motors [3-4,10].

This simplification simplifies the design process by enabling the selection of motor parameters to be bypassed, which can only be determined by the measurement process. To reflect the behaviour of the real system, the following values of the UAV parameters are assumed in table 1 [6].

Table 1. UAV parameters [6]

Parameter	Symbol	Value
Mass [kg]	m	1.295
Inertia for the axes OX/OY [Nms^2]	I_{xx}/I_{yy}	$15.19 \cdot 10^{-3}$
Inertia for the axes OZ [Nms^2]	I_{zz}	$15.06 \cdot 10^{-3}$
Rotor centre of gravity distance [m]	*l*	0.267

The mathematical representation of the quadcopter system is six differential equations. The first three of these are the equations that determine the linear acceleration of the UAV. Integrating these relationships once provides linear velocities, while repeating this process leads to an estimate of the position in three-dimensional space of the aircraft. The remaining three relations describe angular accelerations. The double integral of each of these expressions results in receiving the angles of roll (Φ), pitch (θ) and yaw (ψ):

$$(3)\qquad \begin{aligned} \ddot{x} &= (\sin\psi \sin\Phi + \cos\psi \sin\theta \cos\Phi)\frac{U_1}{m}, \\ \ddot{y} &= (-\cos\psi \sin\Phi + \sin\psi \sin\theta \cos\Phi)\frac{U_1}{m}, \\ \ddot{z} &= -g + (cos\theta \cos\Phi)\frac{U_1}{m}, \\ \ddot{\Phi} &= \frac{I_{yy}-I_{zz}}{I_{xx}} qr - \frac{J_{TP}}{I_{xx}} q\Omega + \frac{lU_2}{I_{xx}}, \\ \ddot{\theta} &= \frac{I_{zz}-I_{xx}}{I_{yy}} pr - \frac{J_{TP}}{I_{yy}} p\Omega + \frac{lU_3}{I_{yy}}, \\ \ddot{\psi} &= \frac{I_{xx}-I_{yy}}{I_{zz}} pq + \frac{U_4}{I_{zz}}, \end{aligned}$$

where: *g* – acceleration of gravity; *m* – mass of UAV; *p, q, r* – angular velocities of the UAV in the coordinate system *x, y, z; Ω* – rotation speed.

For the analysis of these equations, with almost stationary flight conditions, the values of the first derivatives of the angular accelerations will slightly affect the final values of the UAV roll. This results in the following simplifications:

$$(4)\qquad \ddot{\Phi} = \frac{lU_2}{I_{xx}},\ \ddot{\theta} = \frac{lU_3}{I_{yy}},\ \ddot{\psi} = \frac{U_4}{I_{zz}}.$$

The resulting equations described by relation (4) and the relations describing the second derivatives of the positions (3) form the basis of the control system and swarm application. Important from the point of view of object control and data retrieval about the current UAV behaviour is the state vector [10]

$$(5)\qquad \mathbf{x} = [\Phi\ \theta\ \psi\ p\ q\ r\ u\ v\ w\ x\ y\ z]^{\mathrm{T}},$$

containing Euler angles, angular velocities, linear velocities and the position of the UAV.

Altitude controller

A control signal U_1 (2) proportional to the sum of the rotational speeds of the four rotors should be used to maintain constant altitude. Control of the UAV's horizontal attitude can be provided by a proportional-differential-integral controller, commonly referred to as a PID controller.

The PID controller consists of three components. The proportional path is responsible for the gain in the control system. In the integrating path, the addition of the current value of the error signal to the previously stored value is realised - making the integrator block a memory system. The differentiating path, in turn, reflects the rate of change of this error. Using the error of the derivative change, the controller determines the rate at which the UAV approaches the chosen target.

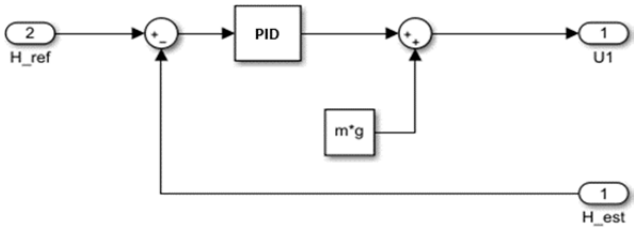

Fig. 2. Method of including the PID controller

The implementation of the controller in the Simulink environment is shown in figure 2. To balance the earth's gravity, the signal from the controller output is summed with the gravity acting on the UAV.

Linear-quadratic hover controller

In order to hold a constant position of the UAV above the Earth (in the so-called hover), the tilt and roll angles of the aircraft must also be subjected to stabilisation. For this purpose, a linear-quadratic LQR controller was chosen to further control the angular parameters θ and Φ [7].

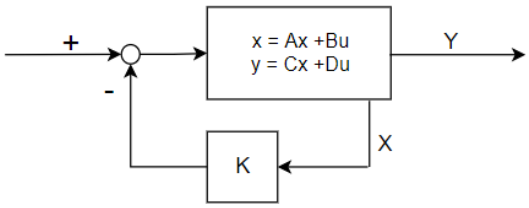

Fig. 3. LQR control scheme (own elaboration based on [7])

LQR is a control technique that allows the control of any process, with an appropriate selection of the parameters **Q**, **R** and the gain matrix **K**. The value of the LQR is determined by the relations [5]

$$(6)\qquad J = \int_0^{\infty} (\mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} + \mathbf{u}^{\mathrm{T}}\mathbf{R}\mathbf{u})\mathrm{d}t,\ \text{gdzie}\ \ \mathbf{u} = -\mathbf{K}\mathbf{x}.$$

The inclusion of the **K** matrix should result in a quick stabilisation of the overshoot, which means a normalisation of the UAV roll angles. To find the **K**-matrix, it is necessary to start the analysis with the selected control paths that will be controlled by the LQR. For this purpose, the values of $\ddot{\Phi}$ and $\ddot{\theta}$ should be determined (4).

In control theory, any object representable in the state space can be represented using the equation of state and observation:

(7) $$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}, \qquad \mathbf{y} = \mathbf{Cx} + \mathbf{Du}.$$

The essence of calculating the **K** matrix in the environment, in addition to knowing the values of **R** and **Q**, is also to find the state matrix **A** and the control matrix **B**. The calculation of both matrices is possible by transforming

(8) $$\frac{d}{dt}\begin{bmatrix}\phi\\ \theta\\ \dot{\phi}\\ \dot{\theta}\\ \int\phi\\ \int\theta\end{bmatrix} = \begin{bmatrix}0&0&0&1&0&0\\0&0&1&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\1&0&0&0&0&0\\0&1&0&0&0&0\end{bmatrix}\begin{bmatrix}\phi\\ \theta\\ \dot{\phi}\\ \dot{\theta}\\ \int\phi\\ \int\theta\end{bmatrix} + \begin{bmatrix}0&0\\0&0\\\frac{l}{I_{yy}}&0\\0&\frac{l}{I_{xx}}\\\frac{1}{m}&0\\0&\frac{1}{m}\end{bmatrix}\begin{bmatrix}U_2\\U_3\end{bmatrix}$$

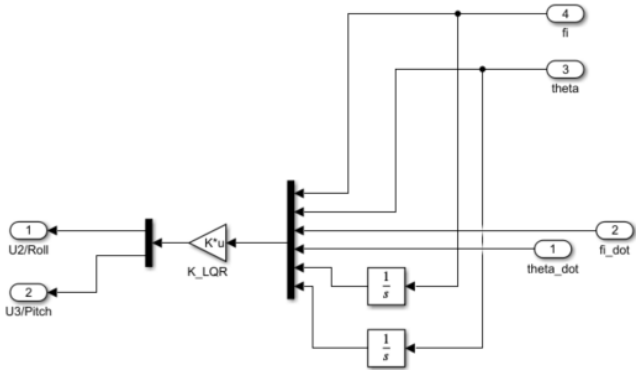

Fig. 4. Internal structure of the LQR block

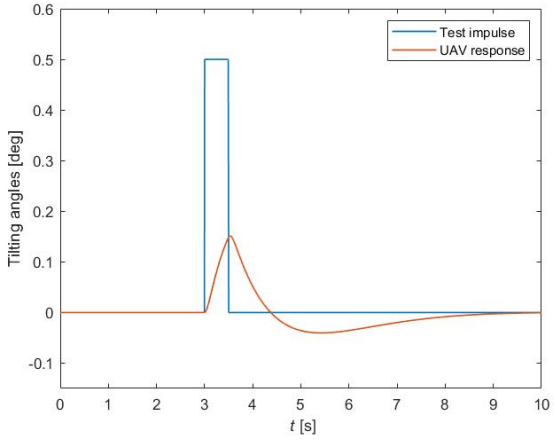

Fig. 5. LQR performance

The mathematical development of the system (8) is a mapping of the first equation of the linear representation of the system into the state space (7). This transformation made it possible to determine matrix **A** of dimension *6 x 6* and matrix **B** of dimension *6 x 2*. Having the necessary data such as parameter **R** of value 1 and identity matrix **Q** of dimension 6 x 6, it is possible to determine the values of the gain elements in the feedback loop from the following relations

(9) $$\mathbf{K} = \mathrm{lqr}(\mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{R}),$$

The Matlab function lqr results in the following **K** matrix values

(10) $$\mathbf{K} = \begin{bmatrix}1.79 & -1.63a & 1.09 & -3.03a & 1 & 1.22a\\ -1.40a & 1.78 & -1.22a & 1.09 & -1.99a & 1\end{bmatrix}$$

where: $a = e^{-16}$

The software implementation is shown in figure 4.

The effects of a successful design phase and software implementation are shown in figure 5.

At the time of the disturbance, the drone tilted forward, and then, due to the LQR regulation, a stabilization manoeuvre took place.

Prediction flight controller

The final phase in the selection and implementation of the UAV controller is the flight controller. Due to the complexity of the mathematical transition between rotational values for UAV roll, the MPC control was chosen for its positioning.

The MPC allows very good control even for highly unstable systems.

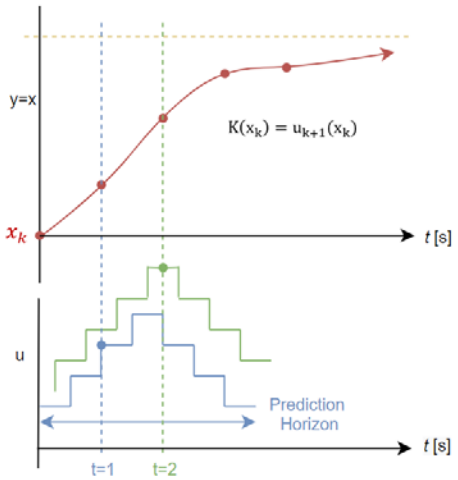

Fig. 6. MPC control (own elaboration based on [8])

In its functionality, the MPC controller offers engineers so-called constraints. They are a means of additional regulation, making it easier to provide limits for selected parameters. In order to protect the quadcopter's system from excessive tilting, the limit for the drone's tilt in all axes was set at ±30°.

The software implementation process consisted of using the MPC controller block and the *MPC designer* application built into the Matlab environment.

Table 2. MPC controller setting values

Parameter	Value
Sample time	0.1
Prediction horizon	30
Control horizon	3

The integration of the dynamic model with the three independent control techniques presented provides a complete controllable model of a single UAV.

UAV formation

The designed algorithm focuses on two tasks for a four-element UAV formation. The first is the close-group approach manoeuvre, while the second movement phase is a circular flight manoeuvre.

The process of acquiring data about its formation neighbours is enabled by the main flight parameter matrix **MM** (11), which contains information about three position components and three velocity components

(11) $$\mathbf{MM} = \begin{bmatrix}X_{d1} & X_{d2} & X_{d3} & X_{d4}\\ Y_{d1} & Y_{d2} & Y_{d3} & Y_{d4}\\ Z_{d1} & Z_{d2} & Z_{d3} & Z_{d4}\\ V_{xd1} & V_{xd2} & V_{xd3} & V_{xd4}\\ V_{yd1} & V_{yd2} & V_{yd3} & V_{yd4}\\ V_{zd1} & V_{zd2} & V_{zd3} & V_{zd4}\end{bmatrix}.$$

Ongoing updating of the matrix allows each drone to calculate the distance and course to its partners. The matrix presented in this way should be regarded as a measurement matrix, in which the most important preconceived flight data take their place.

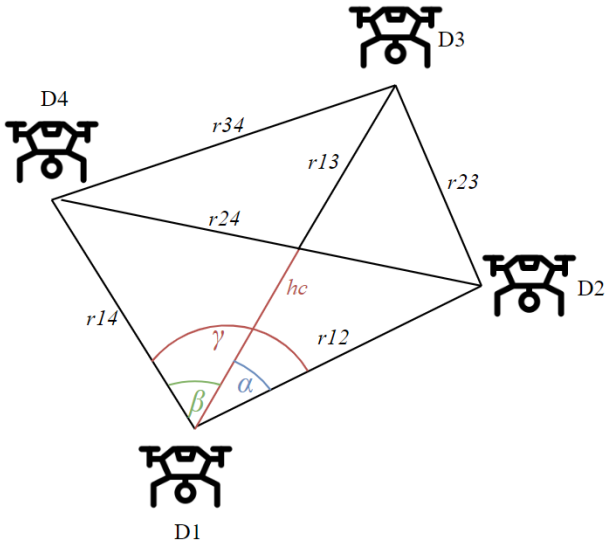

Fig. 7. Procedure for UAV interconnection

The manoeuvre of approaching in a tightly concentrated group is to cluster the drone formation into a square-shaped formation. To perform such a grouping, the diagonal of the formation must first be determined.

Knowing all the distances between the drones allows the angles α, β, γ to be determined. Thus, at any time, drone D1 has up-to-date information about the angular position of the other three drones of the formation. The measure of the largest angle informs the location and numbers of its edge (outer) neighbours. This excludes them from the algorithm and points unambiguously to the drone opposite D1.

The next step of the algorithm is to determine a virtual rally point D (x_{pz}, y_{pz}) that will allow the drones to approach (fig 8).

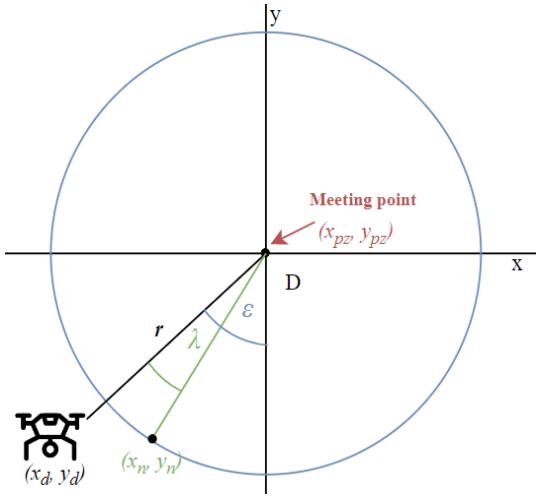

Fig. 8. Formation turnover procedure

Knowing the value of the length of the segment h_c and the angular bearing of UAV1 to UAV2, allows a virtual meeting point to be determined for all drones:

(12) $$\alpha = \arccos\left(\frac{{r_{12}}^2+{r_{13}}^2-{r_{23}}^2}{2\cdot r_{12}\cdot r_{13}}\right),$$

(13) $$P_{\Delta ABC} = P_{\Delta ABD} + P_{\Delta ACD},$$

(14) $$r_{12}r_{14}sin\gamma = r_{12}h_c sin\alpha + r_{14}h_c sin\beta,$$

(15) $$h_c = \frac{r_{12}r_{14}sin\gamma}{r_{12}sin\alpha+r_{14}sin\beta}.$$

Performing formation rotation becomes possible by planning future flight points by each UAV. Then the position of the n-drone is determined according to the equations:

(16) $$\begin{aligned} x_n &= x_{pz} + [r\cdot\cos(\varepsilon+\lambda)] \\ y_n &= y_{pz} + [r\cdot\sin(\varepsilon+\lambda)] \end{aligned}$$

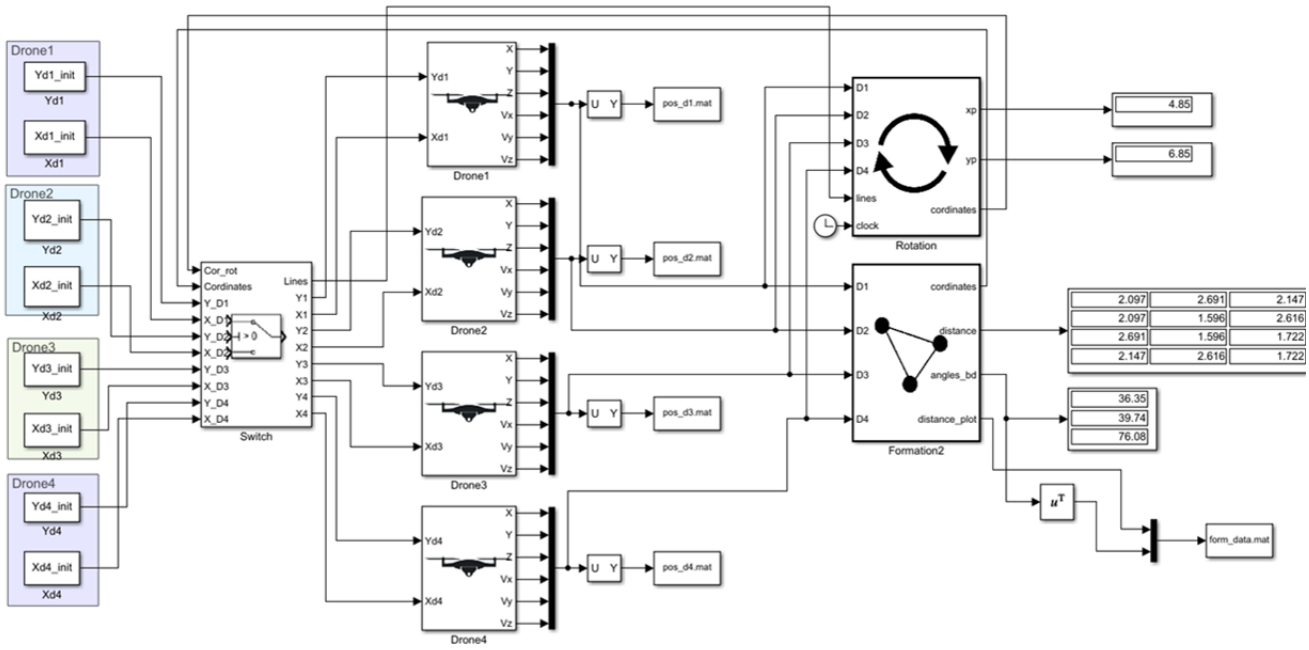

Fig. 9. Programme structure of the UAV formation

Implementation of the swarming algorithm

The first step in the implementation of the UAV formation, was to multiply the quadcopter model with its controller four times. In the next step, new software blocks *Rotation* and *Formation2* were created.

The testing of the integrated formation structure was based on the selection of dozens of different take-off points of individual UAVs. In order to represent the phenomena and dependencies that occurred, it was decided to evaluate one of the possible variants of drone formation (table 3).

Table 3. Starting points of individual UAV

UAV Number	Starting position	
	x_d [m]	y_d [m]
D1	1	3
D2	12	1
D3	8	10
D4	1	10

Swarm simulation results

The representation of the formation's flight path from a bird's-eye perspective is a horizontal projection of the movement of individual UAVs.

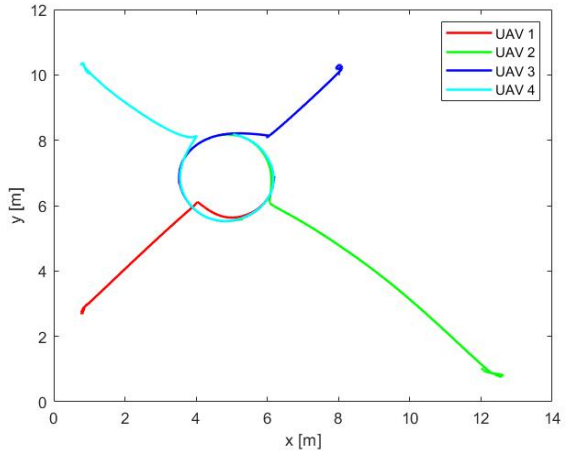

Fig. 10. Formation trajectory in the horizontal plane

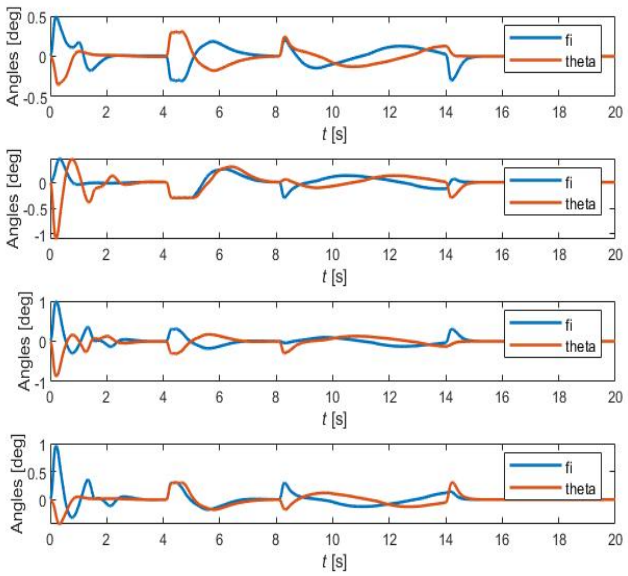

Fig. 11. Tilting angles of individual UAV

In the scenario analysed, it is the angle ψ that takes on the largest values. Such a phenomenon is information for drone D1 that its neighbour opposite is drone D3. Confirmation of the alignment of the formation into a square shape (fig 11), is provided by the data generated in the eighth second of flight. The largest angle value between the outer UAVs (fig. 12) is 90º, while the angle values between the immediate neighbours are 45º.

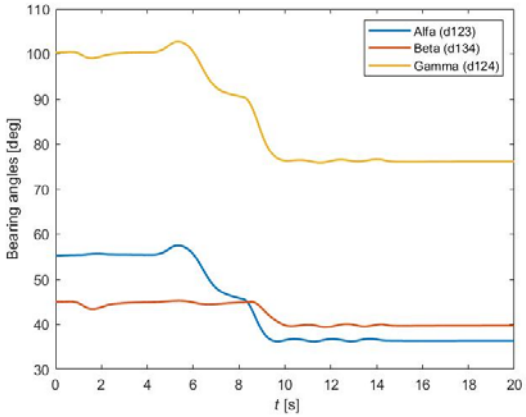

Fig. 12. Bearing angles in relation to individual UAV

In order to present the flight procedure in three-dimensional space, a suitable dynamic visualisation was made. The stages of the UAV coming together in a concentrated formation and the final procedure of circular movement are shown in figure 13.

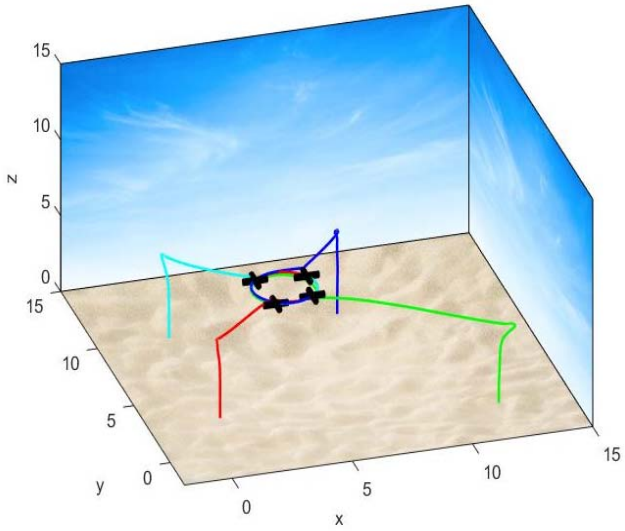

Fig. 13. Visualisation of the formation turnover procedure

The numerous simulations carried out, indicate the correctness of the formation control techniques adopted and the correctness of the UAV formation algorithm implemented.

Impact of measurement inaccuracies on formation behaviour

Measurement imprecision characterises the result of a measurement that is only an approximation of the true value. Measuring instrument inaccuracies, external disturbances, calculation and modelling errors are the main causes of measurement inaccuracies.

The basis of the designed control algorithm is the measure of the distance between the individual drones. Therefore, the impact of distance measurement inaccuracies on the in-flight behaviour of the formation was analysed. Software consideration of measurement inaccuracies amounts to adding an error value by which the algorithm will bias the flight controller. The errors are

particularly noticeable when the UAV formation focusing algorithm switches to circular flight mode.

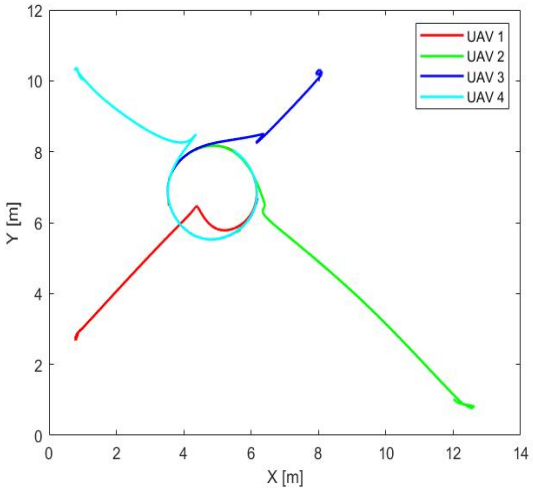

Fig. 14. Characteristics with an imprecision of 25 [cm]

The formation flight parameter values analysed in table 4 are data taken at the eighth second of UAV movement for two values of distance inaccuracy.

Table 4. Formation parameter values

Parameter	Reference value	Distance error	
		10 [cm]	25 [cm]
Distance D1-D3	2.820 [m]	2.951 [m]	3.101 [m]
Distance D1-D2	2.007 [m]	2.131 [m]	2.28 [m]
Angle γ	90.02 [°]	87.75 [°]	83.91 [°]
Angle α	45 [°]	42.85 [°]	41.23 [°]

Conclusions

The increase in measurement inaccuracies of the mutual distances between UAVs, results in positioning errors of individual UAVs. These errors directly affect the shape of the formation by deforming it. If the distance measurement inaccuracy of 25 [cm] is affected, the shape of the formation starts to look like a parallelogram. Both the distances between the D1-D2 and D2-D3 drones increased by 28 [cm] with respect to the reference system. The measurement errors occurring significantly affect the algorithm for clustering the UAVs into a close group.

The designed and implemented swarming algorithm works correctly. The application of numerous decision conditions and the chosen formation control strategy ensures the correctness of swarm formation, regardless of its initial orientation. The control technique used, consisting of three independent controllers, guarantees efficient aircraft movement. Occasional UAV fluctuations may be due to minor irregularities during the selection of controller parameters.

Acknowledgments

This work was financed by Military University of Technology under research project UGB for 2023.

***Authors**: dr inż. Stanisław Konatowski, prof. WAT, Military University of Technology, Institute of Radio Electronics, ul. Gen. Sylwestra Kaliskiego 2, 00-908 Warszawa, E-mail: stanislaw.konatowski@wat.edu.pl; mgr. inż. Sebastian Tatko, Military University of Technology, Faculty of Electronics, ul. Gen. Sylwestra Kaliskiego 2, 00-908 Warszawa.*

REFERENCES

[1] Ambroziak L., Gosiewski Z., Two stage switching control for autonomous formation flight of unmanned aerial vehicles, Aerospace Science and Technology, 46 (2015), pp. 221–226, http://dx.doi.org/10.1016/j.ast.2015.07.015

[2] Bieda R., Wyznaczanie orientacji IMU w przestrzeni 3D z wykorzystaniem macierzy tensora rotacji oraz niestacjonarnego filtru Kalmana. *Przegląd Elektrotechniczny*, R. 89 NR 12/2013

[3] Zawiski R., Control-oriented modelling of quadrotor UAV platform. Ph.D. dissertation, Politechnika Śląska, 2012, pp. 147

[4] Paiva-Pedro E., Soto J., Salinas J., Ipanaqu W., Modeling and PID cascade control of a Quadcopter for trajectory tracking, IEEE ICA/ACCA 2016, Vol. 22, pp. 7

[5] Giernacki W., Stępień S., Chodnicki M., Wróblewska A., Hybrid Quasi-Optimal PID-SDRE Quadrotor Control. Energies. 2022; 15(12):4312. doi.10.3390/en15124312

[6] Shahid, Programming a Voice Controlled Drone Using Node and ARDrone, 2019, https://medium.com/@codeforgeek/-programming-a-voice-controlled-drone-using-node-and-ardrone-159b5c4ad4ec

[7] Douglas B., Linear Quadratic Regulator (LQR) Optimal Control, 2019, youtube.com/watch?v=E_RDCFOlJx4

[8] Brunton S.L., Model predictive control, University of Washington, 2018, youtube.com/watch?v=YwodGM2eoy4

[9] Quesada W., Rodriguez J., Murillo J., Cardona J., Yanguas-Rojas D., Jaimes L., Calderón J., Leader-Follower Formation for UAV Robot Swarm Based on Fuzzy Logic Theory, Springer International Publishing AG, part of Springer Nature 2018, L. Rutkowski et al. (Eds.): ICAISC 2018, LNAI 10842, pp. 740–751, 2018, https://doi.org/10.1007/978-3-319-91262-2_65

[10] Sabatino F., Quadrotor control: modeling, nonlinear control design, and simulation. Master's Degree Project Stockholm, XR-EE-RT 2015:XXX, 2015, pp. 67

[11] Aström K.J., Murray R.M., Feedback System. An Introduction for Scientist and Engineers, 2019