UAV 3D FORMATION FLIGHT USING THE RELATIVE STATE SPACE METHOD

Tagir Z. Muslimov, Rustem A. Munasypov
The Institute for Aerospace Technology, Ufa State Aviation Technical University, Russian Federation
E-Mail: tagir.muslimov@gmail.com

ABSTRACT
In this paper the problem of fixed-wing unmanned aerial vehicles (UAVs) 3D formation flight was solved using the relative state space method. Using this method, the UAVs formation becomes an autonomous decentralized multi-agent system since the system functional order is generated by the interaction of its agents. The solution was tested in MATLAB/Simulink using full dynamic models of the vehicles.

Keywords: UAV control, formation flight, group control, UAV formation, multi-agent system, decentralized control, bio-inspired algorithm, decentralized system.

INTRODUCTION
One of the promising directions for the development of autonomous unmanned aerial vehicles (UAVs) applications is a group control. Flying in formation, i.e. precise holding of certain specified relative positions during the flight of a group both improves the efficiency of certain types of missions and for a number of tasks becomes a prerequisite for their solution. The examples include the localization of radar [1], the overcoming of enemy air defense with the false targets help, the construction of antenna arrays from UAV [2, 3], the wind profiles measurement for meteorological studies [4], automatic refueling in the air [5], the increase of useful load or range by reducing the lift-induced drag in the case of flight in tight formations [6], etc.

There are several approaches to solving the UAVs formation flight problem. The most common are the following: Leader-Wingman method [7, 8] and an approach based on virtual structures [9]. The disadvantages of Leader-Wingman method are the absence of feedback from Wingman vehicles and centralization of the system, which means the Leader UAV failure leads to the formation loss. The approach based on the virtual structures in the original version also does not involve feedback from the control objects and, in addition, is largely sensitive to external disturbances (for example, wind disturbances), thereby losing the accuracy of maintaining the formation.

In our article there is used the relative state space method for the three-dimensional formation UAVs control which is decentralized control of the multi-agent system, fault-tolerant in the sense that the failure of individual agents does not lead either to the failure of the entire system or to the inability to further build and maintain the formation. Each agent has autonomy, i.e. the ability to control part of the system’s global state. This method is a bio-inspired algorithm based on the model of living organisms’ motor neurons network. In comparison with the Leader-Wingman method, the relative state space approach involves the construction of a control hyper surface in the relative state space instead of just following Leader’s commands.

UAV MODEL
For the UAVs dynamic model there are used two coordinate frames: the inertial "north-east-down" (NED) with the index \(^n\) and the body frame with the index \(^b\).

UAV coordinates are specified as follows:
\[
p^n = (p_n, p_e, p_d)^T,
\]
where \(p_n\) is the northern coordinate of the UAV position in the inertial coordinate frame; \(p_e\) is the eastern UAV position in the inertial coordinate frame; \(p_d\) is the UAV coordinate along the axis directed to the center of the Earth in the inertial coordinate frame.

The orientation of the UAV is specified using Euler angles:
\[
\Theta = (\phi, \theta, \psi)^T,
\]
where \(\phi\) – roll angle, \(\theta\) – pitch angle, \(\psi\) - yaw angle.

UAV speed’s components in the body frame:
\[
v^b = (u, v, w)^T,
\]
where \(u\) is the velocity component along the axis directed to the vehicle’s nose, \(v\) is the velocity component along the axis directed along the right vehicle wing, \(w\) is the velocity component along the axis directed from the vehicle bottom top.

The angular velocities
\[
\Omega^b = (p, q, r)^T
\]
rotate around the body frame axes. The control signals’ input vector
\[
u = (\delta_e, \delta_a, \delta_r, \delta_t)^T,
\]
where \(\delta_e\) is the elevator deflection, \(\delta_a\) is the aileron deflection, \(\delta_r\) is the rudder deflection, \(\delta_t\) is the throttle deflection.
Linearized equations of UAVs lateral motion in the state space [10]:

\[
\begin{bmatrix}
\dot{\psi} \\
\dot{\rho} \\
\dot{\phi} \\
\dot{\psi}
\end{bmatrix} =
\begin{bmatrix}
Y_v & Y_p & Y_r & g \cos \theta^* \cos \phi^* & 0 \\
L_v & L_p & L_r & 0 & 0 \\
N_v & N_p & N_r & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\psi \\
\rho \\
\phi \\
\psi
\end{bmatrix} +
\begin{bmatrix}
\dot{Y}_v \\
\dot{L}_v \\
\dot{N}_v \\
\dot{\phi}
\end{bmatrix}
\]

Linearized equations of UAVs longitudinal motion in the state space:

\[
\begin{bmatrix}
\dot{w} \\
\dot{q} \\
\dot{h}
\end{bmatrix} =
\begin{bmatrix}
X_w & X_q & X_r & -g \cos \theta^* & 0 \\
Z_w & Z_q & Z_r & -g \sin \theta^* & 0 \\
M_w & M_q & M_r & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
w \\
q \\
h
\end{bmatrix} +
\begin{bmatrix}
\dot{X}_w \\
\dot{Z}_w \\
\dot{M}_w \\
\dot{h}
\end{bmatrix}
\]

Table-1 shows linearized parameters calculated in MATLAB for the “Zagi” UAV.

<table>
<thead>
<tr>
<th>Linearized parameters for lateral motion</th>
<th>Linearized parameters for longitudinal motion</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Y_v [s^{-1}])</td>
<td>-1.3407</td>
</tr>
<tr>
<td>(Y_p [m \cdot s^{-1}])</td>
<td>0.6728</td>
</tr>
<tr>
<td>(Y_r [m \cdot s^{-1}])</td>
<td>-12.967</td>
</tr>
<tr>
<td>(L_v [m \cdot s^{-1}])</td>
<td>-2.0701</td>
</tr>
<tr>
<td>(L_p [s^{-1}])</td>
<td>-3.2756</td>
</tr>
<tr>
<td>(L_r [s^{-1}])</td>
<td>1.708</td>
</tr>
<tr>
<td>(N_v [m \cdot s^{-1}])</td>
<td>4.4056</td>
</tr>
<tr>
<td>(N_p [s^{-1}])</td>
<td>0.2346</td>
</tr>
<tr>
<td>(N_r [s^{-1}])</td>
<td>-4.3912</td>
</tr>
<tr>
<td>(Y_{\delta} [m \cdot s^{-2}])</td>
<td>0</td>
</tr>
<tr>
<td>(Y_{\delta} [m \cdot s^{-2}])</td>
<td>-3.0234</td>
</tr>
<tr>
<td>(L_{\delta} [s^{-2}])</td>
<td>18.624</td>
</tr>
<tr>
<td>(L_{\delta} [s^{-2}])</td>
<td>24.1101</td>
</tr>
<tr>
<td>(N_{\delta} [s^{-2}])</td>
<td>14.0735</td>
</tr>
<tr>
<td>(N_{\delta} [s^{-2}])</td>
<td>-7.0607</td>
</tr>
<tr>
<td>(X_w [s^{-1}])</td>
<td>0.1805</td>
</tr>
<tr>
<td>(X_q [m \cdot s^{-1}])</td>
<td>-0.9258</td>
</tr>
<tr>
<td>(Z_w [s^{-1}])</td>
<td>-1.1564</td>
</tr>
<tr>
<td>(Z_q [s^{-1}])</td>
<td>-4.9198</td>
</tr>
<tr>
<td>(M_w [m \cdot s^{-1}])</td>
<td>12.967</td>
</tr>
<tr>
<td>(M_q [m \cdot s^{-1}])</td>
<td>0.3311</td>
</tr>
<tr>
<td>(M_r [m \cdot s^{-1}])</td>
<td>-4.6373</td>
</tr>
<tr>
<td>(X_{\delta} [s^{-1}])</td>
<td>-7.2717</td>
</tr>
<tr>
<td>(X_{\delta} [m \cdot s^{-2}])</td>
<td>-0.4559</td>
</tr>
<tr>
<td>(X_{\delta} [m \cdot s^{-2}])</td>
<td>8.5311</td>
</tr>
<tr>
<td>(Z_{\delta} [m \cdot s^{-2}])</td>
<td>6.3863</td>
</tr>
<tr>
<td>(M_{\delta} [s^{-2}])</td>
<td>-79.5245</td>
</tr>
</tbody>
</table>

The autopilot of the longitudinal and lateral motion of a single UAV based on the successive loop closure method was synthesized.
ALGORITHM OF UAVS GROUP CONTROL
BASED ON THE RELATIVE STATE SPACE
METHOD

In the articles [11, 12] in accordance with the
concepts of oscillatory neural networks of living
organisms that generate motor signals of locomotion the
model of a decentralized, autonomous system of
interacting agents was proposed. Based on this approach
we define a group of autonomous UAVs as a system of
this type.

We consider a multi-agent autonomous system as
a graph where each agent is a vertex in the graph and the
interaction is an edge. Let \( n \) be the number of UAVs, \( N \) is
the number of interactions between them,
\( \xi = (\xi_1, \ldots, \xi_n) \) - the vector of current states,
\( \zeta = (\zeta_1, \ldots, \zeta_N) \) - vector of relative states, \( A \) is
the incidence matrix of the graph. In this case, their
relationship is defined as follows:

\[
\zeta = A^T \xi.
\]

Let the dynamic equation of the \( i \)-th UAV:

\[
\frac{d\xi_i}{dt} = f_i(\xi_i, \xi_{i1}, \ldots, \xi_{in}),
\]

where \( \xi_i, \ldots, \xi_{in} \) is the relative states of \( i \)-th, \( i_{1} \)-th, \( i_{n} \)-th
agents which directly interact with \( i \)-th, \( f_i \) - differentiable
function. Therefore, the equation of relative states
dynamics can be represented as follows:

\[
\frac{d\zeta}{dt} = A^T f,
\]

where \( f = (f_{i1}, \ldots, f_{in})^T \).

Then in accordance with [11] the following theorem holds:

**Theorem.** In the relative state space there is a
potential function if and only if the vector-function \( f \) is
defined as follows:

\[
f_i = \tilde{f}_i(\theta_i) + \tilde{f},
\]

where \( \theta_i = \sum_{k=1}^n (\xi_k - \xi_i) \), \( \tilde{f} \) is dynamics common to all
agents. Then the potential function in the relative state
space:

\[
V(\zeta) = \sum_{i=1}^n \int \tilde{f}_i(\theta_i) d\theta_i.
\]

Thus, the interaction between subsystems
(agents) generates the system order itself making it a
gradient system. In this case, the final equilibrium is in the
global minimum of the potential function \( V(\zeta) \) in
the relative state space. In the event of a change in the goal or
the surrounding situation, this potential function will
change and interactions between agents will change
accordingly reflecting the new functional order
construction. In addition, it should be noted the system is
decentralized because a supervisor is not required to make
a formation since each agent determines its own behavior
to achieve the final goal depending on the agents’ behavior
interacting with it. If one of the agents experiences
external perturbations then the others adjust to it keeping the
shape.

The control strategy for \( i \)-th UAV relative to the
coordinate frame eastern axis is as follows:

\[
\frac{dp_i}{dt} = \sum_{j \in J_i} \tau_{ij}(p_j - p_i) + u_{ei},
\]

where \( p_{ei} \) is the northern coordinate of the \( i \)-th UAV in
the inertial frame,
\( \tau_{ij} \) - coefficient of interaction between the \( i \)-th and \( j \)-th
agent,
\( J_i \) - the set of UAVs interacting with the \( i \)-th agent,
\( u_{ei} \) - control action experienced by the \( i \)-th UAV.

Similarly, control strategies are defined for the
northern axis of coordinates and altitude:

\[
\frac{dp_n}{dt} = \sum_{j \in J_i} \tau_{ij}(p_n - p_n) + u_{en},
\]

\[
\frac{dh}{dt} = \sum_{j \in J_i} \tau_{ij}(h_j - h_1) + u_{eh},
\]

where \( h = -p_d \) is the height of the \( i \)-th UAV above sea
level.

The kinematic equations of the UAV are as follows:

\[
\begin{bmatrix}
\dot{p}_n \\
\dot{p}_e \\
\dot{h}
\end{bmatrix} = \begin{bmatrix}
\cos \chi \cos \gamma \\
\sin \chi \cos \gamma \\
\sin \gamma
\end{bmatrix} v_g,
\]

where \( v_g \) is the velocity of the UAV in the inertial frame,
\( \chi \) is the course angle between the velocity vector in
the inertial frame and the north axis of the same coordinate
frame, \( \gamma \) is the flight path angle between the horizontal
plane and the velocity vector in the inertial frame.

Thus, in accordance with equations (1)-(4), the
command for the course angle \( \chi \), the airspeed \( v_g \) and the
flight path angle \( \gamma \) can be expressed as follows:
\[ \chi_i = \arctg \left( \frac{u_{ni}}{u_{ei}} \right), \]

\[ \nu_i = \sqrt{u_n^2 + u_e^2 + u_h^2}, \]

\[ \gamma_i = \arctg \frac{u_{nh}}{\sqrt{u_n^2 + u_e^2}}. \]

Control vector \( U_e = (u_{e1}, ..., u_{en}) \) for the eastern axis of the inertial frame can be found from the following equation:

\[ U_e = B_e p_e + D, \]

where \( D = -B_e H_e^{-1} \left( p_{ed}^T, \hat{P}_e \right)^T \) - control vector of the system in the relative state space, \( H_e \) is a matrix defined as follows:

\[ H_e = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{pmatrix}, \quad q_i = (..., 1, ..., -1, ...), \quad i < n, \quad q_n = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \]

wherein \( H_e \in \mathbb{R} \times n, \quad q \in \mathbb{R} \times 1, \quad p_{ed} \) is the vector of the desired relative positions along the eastern axis of the inertial frame, \( \hat{P}_e = \sum_{k=1}^{n} p_{nk} \) is the sum of the current UAV coordinates in the inertial frame, \( B_e = \begin{pmatrix} m_{y} \tau_{g} \end{pmatrix} \in \mathbb{R} \times n \) - a matrix obtained from the interaction matrix \( M \) which in turn can be represented in various ways depending on the type of interaction between the agents.

For example, for four UAVs in the case of "each-with-each" interaction:

\[ M_1 = \begin{pmatrix} 3 & 1 & 1 & 1 \\ 1 & -3 & 1 & 1 \\ 1 & 1 & -3 & 1 \\ 1 & 1 & 1 & -3 \end{pmatrix}, \]

In the case of the interaction "neighbor with neighbor":

\[ M_2 = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}. \]

Control vectors \( U_n \) and \( U_h \) can be found similarly.

**THE SIMULATION RESULTS**

Figure-1 shows UAVs formation building and maintaining.
Figure-2 shows graphs of the UAVs relative positions errors.

CONCLUSIONS
This article describes the successful application of the relative state space method for the constructing and maintaining the UAV formation. Further research will focus on the algorithm optimization and improvement.

REFERENCES


