Formation Control of UAVs with a Fourth-Order Flight Dynamics

Yasuhiro Kuriki* and Toru Namerikawa*

Abstract: This paper studies cooperative control problems with a multi-UAV (Unmanned Aerial Vehicle) system expressed as a fourth-order system using a consensus-based algorithm. How to model a linearized model of UAVs like quadrotors as a fourth-order system is described, and then a formation control algorithm for the fourth-order system is proposed after formulating a problem. The proposed control law is based on a consensus algorithm, and a leader-follower structure is also applied to the control law so that the leader can provide the quadrotors with commands such as their desired states. And then, the study shows that the proposed control algorithm can guarantee accurate formation keeping when fundamental assumptions about the network composed of the multiple UAVs are satisfied. Finally, the proposed approach is validated by some simulations.

Key Words: cooperative control, formation, consensus, UAVs.

1. Introduction

In recent years, cooperative control problems with a multi-vehicle system have attracted a lot of attention from many researchers [1]–[21]. These cooperative control technologies are expected to be applied to actual vehicles such as UAVs, artificial satellites, and autonomous mobile observation robots as well as sensor networks. There is a possibility that a multi-vehicle system can perform tasks more efficiently than a single highly functional vehicle does.

The authors have been studying cooperative control problems with a multi-vehicle system [17]–[21], especially focusing on formation control problems using a consensus algorithm. In [19], the authors proposed a control algorithm for multiple columnar vehicles to surround a moving columnar object cooperatively without depending on a network structure. This method, however, resulted in fixing the geometric configuration of the vehicles, with respect to a fixed system of coordinates, even when the object moved changing the traveling direction. Generally, every cooperatively controlled UAV flies aiming at a single cooperative objective. In other words, how the UAV flies is affected by the other UAVs, that is, the UAVs do not fly aiming at their own objectives. Therefore, each of the cooperatively controlled UAVs must establish an individual objective, which is different from the cooperative one, to change the geometric configuration of formation individually. Considering these points, we have already proposed a control algorithm that UAVs fly in formation and also change their geometric configuration arbitrarily [21]. The control algorithm, however, was intended for a first-order system which did not allow for dynamics.

Cooperative control problems using a consensus-based algorithm have specifically attracted a lot of attention from many researchers in recent years. And then, many algorithms have been developed for the problems under different assumptions. Authors in [2],[4]–[8],[10]–[12],[16] and [18]–[21] considered a first-order system, and authors in [5],[7] and [12] considered a second-order system, just to name a few. Overall, most works of the cooperative control problems using a consensus-based algorithm assume that vehicles are described as a first-order system. The results in a first-order system are directly applied to most works of a second-order system. Works of a linear system [3],[9],[13]–[15] and [17], however, are not directly extended from the results in a first-order system, because it is difficult to extend the algorithms for a first-order system directly to the problems with a linear system. Therefore, most authors who consider a linear system mainly use complicated ways such as an optimization approach and a LMI (Linear Matrix Inequality). A consensus-based algorithm has advantages in simplicity of finding appropriate controller gains and flexibility in network structure of multi-vehicle system.

It is well known that almost all of vehicles or UAVs contain linear or nonlinear dynamics. We generally describe the vehicles or the UAVs as a linear system when we consider control problems. For example, quadrotors, which is a type of UAV, can be modeled as a fourth-order system with change of the attitude and the position under ideal conditions. Note that a fourth-order system is a special case of a linear system. Recently, small unmanned helicopters with multiple rotors such as quadrotors have attracted attention. A quadrotor has four propellers powered by electric motors, and has good flight performance while the quadrotor is generally controlled only by the rotational speed of the four propellers. Making the best use of the advantages, many researchers have been carrying out studies on various problems using quadrotors [22]–[28]. In addition, there are some quadrotors which are sold for civil use such as air surveillance, aerial observation and research uses.

To take full advantage of a consensus-based algorithm, how we directly extend the remarkable results in a first-order system using a consensus-based algorithm to a fourth-order system with dynamics is a continuing issue. For the quadrotor expressed as a fourth-order system, we propose a control al-

* Faculty of Science and Technology, School of Integrated Design Engineering, Keio University, 3-14-1, Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan
E-mail: yasuhiro@nl.sd.keio.ac.jp, namerikawa@sd.keio.ac.jp
(Received April 21, 2013)
(Revised July 7, 2013)
algorithm to fly in formation, and also to change the geometric configuration of formation.

An outline of this paper is as follows. In section 2, modeling a quadrotor and a network structure among the UAVs are stated, and then a control objective is defined. In section 3, for the problem defined in section 2, we propose a control algorithm to achieve the control objective, and then we prove the theorem. Section 4 presents simulation results to validate the proposed approach, and finally, concluding remarks are stated in section 5.

2. Problem Statement

This section presents how a quadrotor is modeled as a fourth-order system and how a network structure among the UAVs is modeled using graph theory. After that, a control objective is defined.

2.1 Modeling a Quadrotor

Suppose that there are \( N \) quadrotors which have the same motion characteristics. Also, each of the quadrotors has four propellers and a controller. The controller can give each of the propellers control commands individually. To simply model the quadrotor [24],[28], let us make a few assumptions as follows. First, the quadrotor flies slowly enough to ignore external aerodynamic forces such as aerodynamic drag and blade vortex interface acting on the quadrotor. Second, the propellers respond to thrust commands fast enough to ignore time delay from when the controller gives the propellers the thrust commands till the propellers actually generate the commanded thrust forces. Lastly, the quadrotor flies at constant altitude and does not yaw. And besides, the controller does not give the quadrotor yaw moment commands. Under these assumptions, a longitudinal and a lateral linearized model of the quadrotor are given by

\[
\begin{align*}
\frac{d}{dt} \begin{bmatrix} x \\ u \\ \theta \\ q \end{bmatrix} &= \begin{bmatrix} u \\ -g\theta \\ \frac{M_0}{I_{xy}} \end{bmatrix}, \\
\frac{d}{dt} \begin{bmatrix} y \\ v \\ \phi \\ p \end{bmatrix} &= \begin{bmatrix} v \\ gg \phi \\ p \\ \frac{M_g}{I_{xx}} \end{bmatrix},
\end{align*}
\]  

(1)

where an equilibrium point of a nonlinear quadrotor system is the state that the quadrotor flies at a constant altitude and velocity. The meaning of the symbols in (1) is shown in Table 1.

Note that though the controller commands a magnitude of the moment about both a longitudinal and a lateral axis, a magnitude of the thrust which each of the propellers should generate can be derived from the magnitude of the moment using the geometric configuration of the propellers.

Next, we define new state variables for the quadrotor:

\[
\begin{align*}
\bar{\theta} &= -g\theta, \\
\bar{q} &= -gq, \\
\bar{M}_0 &= -\frac{g}{I_{xy}} M_0,
\end{align*}
\]  

(2)

\[
\begin{align*}
\bar{\phi} &= g\phi, \\
\bar{p} &= gp, \\
\bar{M}_g &= \frac{g}{I_{xx}} M_g.
\end{align*}
\]  

(3)

Using (1), (2) and (3), we can write new state space equations of motion for the quadrotor as

\[
\begin{align*}
\frac{d}{dt} \begin{bmatrix} x \\ \bar{\theta} \\ \bar{q} \end{bmatrix} &= \begin{bmatrix} u \\ \frac{M_0}{I_{xy}} \end{bmatrix}, \\
\frac{d}{dt} \begin{bmatrix} y \\ v \\ \bar{\phi} \end{bmatrix} &= \begin{bmatrix} v \\ \frac{M_g}{I_{xx}} \end{bmatrix},
\end{align*}
\]  

(4)

Next, we also define new state variables for the quadrotor:

\[
\begin{align*}
r_i^0 &= x, \\
r_i^{(1)} &= u, \\
r_i^{(2)} &= \bar{\theta}, \\
r_i^{(3)} &= \bar{q}.
\end{align*}
\]  

(5)

\[
\begin{align*}
r_i^0 &= y, \\
r_i^{(1)} &= v, \\
r_i^{(2)} &= \bar{\phi}, \\
r_i^{(3)} &= \bar{p}.
\end{align*}
\]  

(6)

where the superscript denotes the order of the variable. Using (4), (5) and (6), we can also write new state space equations of motion for the quadrotor as

\[
\begin{align*}
\frac{d}{dt} \begin{bmatrix} r_i^{(0)} \\ r_i^{(1)} \\ r_i^{(2)} \\ r_i^{(3)} \end{bmatrix} &= \begin{bmatrix} r_i^{(0)} \\ \bar{M}_0 \end{bmatrix}, \\
\frac{d}{dt} \begin{bmatrix} r_j^{(0)} \\ r_j^{(1)} \\ r_j^{(2)} \\ r_j^{(3)} \end{bmatrix} &= \begin{bmatrix} r_j^{(0)} \\ \bar{M}_g \end{bmatrix},
\end{align*}
\]  

(7)

Then, combining the longitudinal and the lateral variable which has the same order, we define new state variables as

\[
\begin{align*}
r_i^{(k)} &= [r_i^{(k)} \ r_i^{(k)}]^T, \quad k \in \{0, 1, 2, 3\}, \quad i \in \{1, 2, \cdots, N\},
\end{align*}
\]  

(8)

\[
\begin{align*}
M_i &= [\bar{M}_0 \quad \bar{M}_g], \quad i \in \{1, 2, \cdots, N\},
\end{align*}
\]  

(9)

where the subscript of \( i \) denotes the quadrotor \( i \).

Using (7), (8) and (9), we can get a following quadrotor model which is expressed as a fourth-order system:

\[
\begin{align*}
\frac{d}{dt} \begin{bmatrix} r_i^{(0)} \\ r_i^{(1)} \\ r_i^{(2)} \\ r_i^{(3)} \end{bmatrix} &= \begin{bmatrix} r_j^{(1)} \\ \bar{M}_i \end{bmatrix}, \\
& \quad i \in \{1, 2, \cdots, N\}.
\end{align*}
\]  

(10)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x, y )</td>
<td>Position (x-axis, y-axis)</td>
</tr>
<tr>
<td>( u, v )</td>
<td>Velocity (x-axis, y-axis)</td>
</tr>
<tr>
<td>( \phi, \theta )</td>
<td>Attitude angle (roll, pitch)</td>
</tr>
<tr>
<td>( p, q )</td>
<td>Angular velocity (roll, pitch)</td>
</tr>
<tr>
<td>( M_0, M_g )</td>
<td>(Nm) Control moment (roll, pitch)</td>
</tr>
<tr>
<td>( I_{xx}, I_{yy} )</td>
<td>Inertia (kgm²)</td>
</tr>
<tr>
<td>( g )</td>
<td>(m/s²) Gravity constant</td>
</tr>
</tbody>
</table>

Table 1: Definition of symbols.

2.2 Modeling a Network Structure among UAVs

A multi-UAV system is modeled as a group of dynamical system which exchanges information with one another. To describe this network composed of multiple UAVs, we use graph theory [29].

We use a graph \( \mathcal{G} = (\mathcal{V}, \mathcal{A}) \) to model information interaction among \( N \) UAVs, where \( \mathcal{V} = \{v_1, v_2, \cdots, v_N\} \) is a set of nodes, and \( \mathcal{A} \in \mathcal{V} \times \mathcal{V} \) is a set of edges. The edge \( (v_i, v_j) \) in the edge set of the graph denotes that there is a network path from the UAV \( i \) to the UAV \( j \). This means that the UAV \( j \) can obtain information from the UAV \( i \).

There are two types of graphs. One is an undirected graph where both the node \( i \) and the node \( j \) can obtain information with each other. The other is a digraph where the node \( i \) can obtain information from the node \( j \), but in the reverse direction, the node \( j \) cannot obtain information from the node \( i \). A graph has some important characteristics. First, a graph is defined as connected if there is a network path between the node \( i \) and the node \( j \) for every pair of different vertices. This means that all UAVs can obtain information with each other through a network. Second, a directed tree is a digraph where every node has
exactly one parent node except for one node, called a root. The root has no parent and has a directed path to every other node. Also, a directed spanning tree of \( G \) is a tree which contains all nodes of \( G \).

Let \( A \in \mathbb{R}^{N \times N} \), \( D \in \mathbb{R}^{N \times N} \) and \( L = L \) be an adjacency matrix, a degree matrix and a graph Laplacian matrix related to the graph \( G \), respectively. The component of the adjacency matrix \( A = [a_{ij}] \) is defined as

\[
a_{ij} = \begin{cases} 1, & \text{for } (v_j, v_i) \in A \\ 0, & \text{otherwise} \end{cases}.
\] (11)

This means that if the UAV \( i \) is obtaining information from the UAV \( j \) through a network, \( a_{ij} \) is set to one, otherwise \( a_{ij} \) is set to zero.

The degree matrix \( D \) is an in-degree matrix defined as

\[
D = \text{diag}(\text{deg}(v_1), \text{deg}(v_2), \cdots, \text{deg}(v_N)),
\] (12)

where \( \text{deg}(v_j) \) is a number of communication links arriving at the node \( v_j \).

The graph Laplacian matrix is defined as

\[
L = D - A.
\] (13)

The graph Laplacian \( L \) has the following properties: if a graph has a directed spanning tree, the graph Laplacian \( L \) has a single eigenvalue of zero, and all nonzero eigenvalues of the graph Laplacian have positive real part.

### 2.3 A Control Objective

In this paper, let us consider a mission that \( N \) quadrotors follow their leader, and both the quadrotors and the leader fly in formation as shown in Fig. 1. This figure shows three quadrotors follow their leader and fly in formation.

In order to achieve this mission, a control objective in this paper is defined as follows:

Each of the quadrotors follows their leader, and the position asymptotically converges to a desired relative position between the quadrotor and the leader. This is formulated as

\[
\lim_{t \to \infty} \| r_i(t) - (r_L(t) + d_i(t)) \| = 0, \quad i = 1, 2, \cdots, N,
\] (14)

where as shown in Fig. 2, \( r_i \) is a position of the quadrotor \( i \), \( r_L \) is a position of a leader, and \( d_i \) is a desired relative position between the quadrotor \( i \) and the leader. Note that the desired relative position for the quadrotor \( i \) should be different from it for any other quadrotor.

To achieve this control objective, we make a set of assumptions as follows:

**Assumption 1.** Every quadrotor must be connected from a leader on the network, but all of the quadrotors are not necessarily directly connected from the leader. In addition, there must be an undirected path, which is an interactive communication link, between every pair of distinct quadrotors.

**Assumption 2.** Movement of the leader must be independent from any quadrotor, that is, movement of the leader is not affected by any quadrotors.

### 3. Proposed Approach

In this section, we propose a control algorithm to achieve the control objective as mentioned in section 2.3.

To achieve the control objective, we consider an objective of a group of quadrotors and an objective of each of the quadrotors separately. The first one is that a group of quadrotors cooperatively flies in formation. The other one is that each of the quadrotors makes geometric configuration of the formation. We apply a consensus-based cooperative control algorithm to achieve the former cooperative object, and also apply a leader-follower structure to achieve the latter individual object. The leader provides the only directly connected quadrotors with their own positions and desired positions for every quadrotor as commands. The advantage of the proposed control algorithm is that all of the quadrotors are not necessarily directly connected from the leader. For example, the minimum network structure satisfying the assumptions 1 and 2 is enough to achieve the control objective.

A control law which should be applied to the quadrotor \( i \) is given by

\[
M_i(t) = - \sum_{j=1}^{N+1} a_{ij} \left[ \sum_{k=0}^{3} \beta_k r_i^{(k)} - r_j^{(k)} \right],
\] (15)

\[
a_{ij} = \begin{cases} 1, & \text{for the UAV } i \text{ connected from the UAV } j \\ 0, & \text{otherwise} \end{cases}
\] (16)

\[
r_j^{(k)} = r_j^{(k)} - d_j^{(k)}, \quad j = 1, 2, \cdots, N + 1, \\
\quad k = 0, 1, 2, 3,
\] (17)

where \( \beta_k \in \mathbb{R} \), for \( k \in \{0, 1, 2, 3\} \) is positive controller gains, the subscript of \( N + 1 \) denotes a leader, and \( r_j^{(k)} \), for \( k \in \{0, 1, 2, 3\} \)
is defined as a vector subtracting a desired relative state for the quadrotor \( j \) from a state of the quadrotor \( j \).

Note that a network structure of the multi-UAV system must be obtained to define the controller gains in advance.

In addition, \( a_{ij} \) is a value which indicates whether the UAV \( i \) obtains information from the UAV \( j \) or not, that is, if obtaining some information, \( a_{ij} = 1 \), otherwise \( a_{ij} = 0 \). Also, the control objective is achieved among a team of the quadrotors and the leader when \( \| r_k(t) - r_k(t) \| \to 0 \), for all \( i = 1, \ldots, N \), as \( t \to \infty \).

For the control input (15) and the multi-UAV system composed of the quadrotors and the leader, the following theorem concerning desired convergence is derived.

**Theorem 1.** Suppose that a multi-UAV system composed of \( N(\geq 1) \) quadrotors expressed as (10) and a leader. Also, the assumptions 1 and 2 are satisfied. Let a nonzero minimum eigenvalue of the graph Laplacian of the network composed of the quadrotors and the leader be \( \lambda_{\text{min}} \). When the control protocol (15) is applied to each of the quadrotors, and when the controller gains \( \beta_k \), for \( k \in \{0, 1, 2, 3\} \) are selected so as to satisfy new conditions, then the control objective (14) is asymptotically achieved.

\[
\beta_k > 0, \quad \forall k \in \{0, 1, 2, 3\},
\]

\[
\lambda_{\text{min}} > \frac{\beta_1^2}{\beta_1 \beta_3 - \beta_0 \beta_3^2},
\]

\[
\beta_1 \beta_3 > \beta_0 \beta_3.
\]

**Proof.** Applying the control protocol (15) to the quadrotor \( i \) expressed as (10), we get

\[
\dot{r}_i^{(3)} = -\sum_{j=1}^{N+1} a_{ij} \beta_k (p_j^{(3)} - r_j^{(3)}), \quad i \in \{1, 2, \ldots, N\}.
\]

Expanding the right-hand-side with respect to \( k \), we get

\[
r_i^{(3)} = -\sum_{j=1}^{N+1} a_{ij} \beta_0 (p_j^{(0)} - r_j^{(0)}) + \beta_1 (p_j^{(1)} - r_j^{(1)}) + \beta_2 (p_j^{(2)} - r_j^{(2)}) + \beta_3 (p_j^{(3)} - r_j^{(3)}), \quad i \in \{1, 2, \ldots, N\}.
\]

Here, the graph Laplacian \( \mathcal{L} \in \mathbb{R}^{(N+1) \times (N+1)} \) of the network composed of the \( N \) quadrotors and the leader is given by

\[
\mathcal{L} = \begin{bmatrix}
\sum_{j=1}^{N+1} a_{ij} & -a_{i1} & \cdots & -a_{iN} & -a_{i(N+1)} \\
-a_{1j} & \ddots & \cdots & \vdots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
-a_{nj} & \cdots & -a_{nj} & \ddots & -a_{n(N+1)} \\
0 & \cdots & \cdots & 0 & 0
\end{bmatrix},
\]

where the leader is regarded as the UAV (\( N + 1 \)). We define a new state vector \( \dot{p}^{(k)} = [p_1^{(k)T} \ p_2^{(k)T} \ \cdots \ p_{N+1}^{(k)T}]^T \in \mathbb{R}^{2(N+1)} \), \( \dot{p}^{(k)} = [r_1^{(k)T} \ r_2^{(k)T} \ \cdots \ r_N^{(k)T} \ \mathbf{0}_N^T]^T \in \mathbb{R}^{2N+1} \), for \( k \in \{0, 1, 2, 3\} \), where \( \mathbf{0}_N \in \mathbb{R}^N \) is a \( n \)-dimensional zero vector.

Using this graph Laplacian \( \mathcal{L} \), \( \dot{p}^{(k)} \) and \( \dot{p}^{(k)} \) to express (22) in a matrix form, we rewrite (22) as

\[
\frac{d}{dt} \dot{p}^{(3)} = -\beta_0 (\mathcal{L} \otimes I_2) p^{(0)} - \beta_1 (\mathcal{L} \otimes I_2) p^{(1)} - \beta_2 (\mathcal{L} \otimes I_2) p^{(2)} - \beta_3 (\mathcal{L} \otimes I_2) p^{(3)},
\]

where \( \otimes \) is the Kronecker product, and \( I_n \in \mathbb{R}^{n \times n} \) is a \( n \)-dimensional unit matrix.

Here, the vector \( \dot{p}^{(k)} \) consists of the states of the quadrotors and commands from the leader. Therefore, we separate the states of the quadrotors from the others. The following identities concerning the rows of the graph Laplacian \( \mathcal{L} \) always hold.

\[
a_{i(N+1)} = \sum_{j=1}^{N+1} a_{ij} - a_{i1} - a_{i2} - \cdots - a_{iN},
\]

\[
a_{ii} = 0, \quad i \in \{1, 2, \ldots, N\}.
\]

Now, using \( d_k^{(k)} = 0 \), for \( k \in \{0, 1, 2, 3\} \) and (25), (24) is separately expressed as

\[
\begin{align*}
\dot{r}^{(3)} &= -\beta_0 \dot{N} (0) - \beta_1 \dot{N} (1) - \beta_2 \dot{N} (2) - \beta_3 \dot{N} (3) \\
&+ \beta_0 \dot{N} (N+1) + \beta_1 \dot{N} (N+1) + \beta_2 \dot{N} (N+1) + \beta_3 \dot{N} (N+1) \\
&+ \beta_0 \dot{N} (N+1) + \beta_1 \dot{N} (N+1) + \beta_2 \dot{N} (N+1) + \beta_3 \dot{N} (N+1),
\end{align*}
\]

(26)

where the matrix \( \mathcal{M} \in \mathbb{R}^{2(N+1) \times 2(N+1)} \) is defined as (27), and \( \dot{M} = \dot{M} \otimes I_2 \). And also, \( r^{(k)} = [r_1^{(1)} \ r_2^{(1)} \ \cdots \ r_N^{(1)}]^T \in \mathbb{R}^{2N} \)

\[
\dot{r}^{(k)} = [\dot{r}_1^{(k)} \ \dot{r}_2^{(k)} \ \cdots \ \dot{r}_N^{(k)}]^T \in \mathbb{R}^{2N}, \text{ for } k \in \{0, 1, 2, 3\}.
\]

Note that the matrix \( \mathcal{M} \) is similar to the graph Laplacian of the multi-UAV system, but not equal to the matrix.

\[
\mathcal{M} = \begin{bmatrix}
The following equation is rewritten in a matrix-vector form as

\[
\begin{bmatrix}
\dot{r}^{(0)} \\
\dot{r}^{(1)} \\
\dot{r}^{(2)} \\
\dot{r}^{(3)}
\end{bmatrix} =
\begin{bmatrix}
0 & I_2 & 0 & 0 & 0 & 2N \\
0 & 0 & I_2 & 0 & 0 & 2N \\
0 & 0 & 0 & I_2 & 0 & 2N \\
0 & 0 & 0 & 0 & I_2 & 2N
\end{bmatrix}
\begin{bmatrix}
\dot{r}^{(0)} \\
\dot{r}^{(1)} \\
\dot{r}^{(2)} \\
\dot{r}^{(3)}
\end{bmatrix} +
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 2N \\
0 & 0 & 0 & 0 & 0 & 2N \\
0 & 0 & 0 & 0 & 0 & 2N \\
0 & 0 & 0 & 0 & 0 & 2N
\end{bmatrix}
\begin{bmatrix}
\dot{r}^{(0)} \\
\dot{r}^{(1)} \\
\dot{r}^{(2)} \\
\dot{r}^{(3)}
\end{bmatrix} +
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 2N \\
0 & 0 & 0 & 0 & 0 & 2N \\
0 & 0 & 0 & 0 & 0 & 2N \\
0 & 0 & 0 & 0 & 0 & 2N
\end{bmatrix}
\begin{bmatrix}
\dot{r}^{(0)} \\
\dot{r}^{(1)} \\
\dot{r}^{(2)} \\
\dot{r}^{(3)}
\end{bmatrix}
\end{align*}
\]

(28)

where \( 0_n \in \mathbb{R}^{n \times n} \) is a \( n \)-dimensional zero matrix.

Now, let us consider a solution of the differential equation (28). This differential equation means that the states of the quadrotor are affected not only by the controller gains and the network structure but also by the commands from the leader.

First, to examine the stability of (28), let us consider a solution of the homogeneous equation of (28) expressed as

\[
\dot{r} = \mathcal{N} r,
\]

where \( \mathcal{N} = [r^{(0)T} r^{(1)T} r^{(2)T} r^{(3)T}]^T \) and the matrix \( \mathcal{N} \) is expressed as
\[
N = \begin{bmatrix}
0_{2N} & I_{2N} & 0_{2N} & 0_{2N} \\
0_{2N} & 0_{2N} & I_{2N} & 0_{2N} \\
0_{2N} & 0_{2N} & 0_{2N} & I_{2N} \\
-\beta_0M & -\beta_1M & -\beta_2M & -\beta_3M
\end{bmatrix}.
\] (30)

The stability of (29) is determined by the eigenvalues of the Matrix \(N\). First, let \(\sigma\) and \(s\) be an eigenvalue and an eigenvector of the Matrix \(M\) respectively, that is, \(MS = \lambda S\). Also, let \(\mu\) and \(\sigma\) be an eigenvalue and an eigenvector of the Matrix \(N\) respectively, that is, \(N\sigma = \mu\sigma\). Then, next equality is derived from Eq. (29):

\[
\begin{bmatrix}
0_{2N} & I_{2N} & 0_{2N} & 0_{2N} \\
0_{2N} & 0_{2N} & I_{2N} & 0_{2N} \\
0_{2N} & 0_{2N} & 0_{2N} & I_{2N} \\
-\beta_0M & -\beta_1M & -\beta_2M & -\beta_3M
\end{bmatrix}
\begin{bmatrix}
s \\
\mu s \\
\rho s \\
\rho^2 s
\end{bmatrix}
= \mu \sigma,
\] (31)

where \(\sigma\) is expressed as

\[
\sigma = \begin{bmatrix}
s \\
\mu s \\
\rho s \\
\rho^2 s
\end{bmatrix}.
\] (32)

From the lowest row block of (31) and \(MS = \lambda S\), we get

\[
\mu^4 + \beta_1\mu^3 + \beta_2\mu^2 + \beta_1\lambda\mu + \beta_0\lambda = 0.
\] (33)

Equation (33) shows the relationship of the eigenvalues between the matrix \(M\) and \(N\). For example, this shows the matrix \(N\) has four eigenvalues \(\mu\) for each of the eigenvalues \(\lambda\) of the matrix \(M\).

Here, let us examine the relationship. It follows from the definition of the two matrix that one of the eigenvalues of the graph Laplacian \(\mathcal{L}\) is zero, and the rest of the eigenvalues \(\mathcal{L}\) are equal to the eigenvalues of the matrix \(M\). The graph Laplacian \(\mathcal{L}\) has a single eigenvalue at zero, and all nonzero eigenvalues of the graph Laplacian have positive real part when the graph has a directed spanning tree. Satisfying the assumptions 1 and 2 indicates that the graph has a directed spanning tree and the matrix \(M\) is symmetric. Hence, the graph Laplacian \(\mathcal{L}\) has a single eigenvalue at zero, and all nonzero eigenvalues of the graph Laplacian \(\mathcal{L}\) have real positive number. As a result, all of the eigenvalues \(\lambda\) of the matrix \(M\) have real positive number.

Now, to achieve the control objective, all of the real parts of the eigenvalues of the matrix \(N\) must be negative. To satisfy this condition, using the Routh-Hurwitz criterion and the result that all of the eigenvalues \(\lambda\) of the matrix \(M\) have real positive number, after carrying out computations, we get

\[
\beta_k > 0, \quad \forall k \in \{0, 1, 2, 3\},
\] (34)

\[
\lambda_{\min} > \frac{\beta_1}{\beta_1\beta_3 - \beta_0\beta_3},
\] (35)

\[
\beta_0\beta_2 > \beta_0\beta_3,
\] (36)

where \(\lambda_{\min}\) is the nonzero minimum eigenvalue of the graph Laplacian \(\mathcal{L}\).

When the controller gains \(\beta_k\), for \(k \in \{0, 1, 2, 3\}\) are selected so as to satisfy the conditions (34) - (36), all of the eigenvalues of the matrix \(N\) have negative real part. Applying these appropriate gains to the controller produces following convergence:

\[
\lim_{t \to \infty} \rho = 0.
\] (37)

Next, a particular solution of (28) is given by

\[
\begin{bmatrix}
\rho^{(0)} \\
\rho^{(1)} \\
\rho^{(2)} \\
\rho^{(3)}
\end{bmatrix}
= \begin{bmatrix}
\tilde{p}^{(0)}_{N+1} \\
\tilde{p}^{(1)}_{N+1} \\
\tilde{p}^{(2)}_{N+1} \\
\tilde{p}^{(3)}_{N+1}
\end{bmatrix} + \begin{bmatrix}
d^{(0)} \\
d^{(1)} \\
d^{(2)} \\
d^{(3)}
\end{bmatrix}, \quad \text{as } t \to \infty.
\] (38)

This validity can be confirmed by substituting (38) for (28). To confirm this validity, we use \(\tilde{p}^{(k)}_{N+1} = 0\) and \(d^{(k)} = 0\).

The general solution of the non-homogeneous differential equation (28) is the sum of the particular solution and the general solution of the homogeneous equation. Hence, when the controller gains \(\beta_k\), for \(k \in \{0, 1, 2, 3\}\) are appropriately selected, that is, these gains are selected so as to satisfy the conditions (34) - (36), the general solution of (28) asymptotically converges to

\[
\begin{bmatrix}
\rho^{(0)} \\
\rho^{(1)} \\
\rho^{(2)} \\
\rho^{(3)}
\end{bmatrix}
\to \begin{bmatrix}
\tilde{p}^{(0)}_{N+1} \\
\tilde{p}^{(1)}_{N+1} \\
\tilde{p}^{(2)}_{N+1} \\
\tilde{p}^{(3)}_{N+1}
\end{bmatrix} + \begin{bmatrix}
d^{(0)} \\
d^{(1)} \\
d^{(2)} \\
d^{(3)}
\end{bmatrix},
\] (39)

This means that the states of the quadrotors will converge only to commands from the leader.

From the element of the first row block in (39), it is proved that the control objective (14) is asymptotically achieved when the control protocol (15) with appropriate controller gains \(\beta_k\) is applied to the quadrotor \(i\) expressed as (10).

\[\square\]

4. Simulation Results

In this section, we present some simulation results to validate the performance of the proposed control algorithm.

4.1 Simulation Setup

Simulations are carried out for three cases as shown in Table 2. We consider a group of three quadrotors and a leader is trying to fly in formation.

The initial positions are as follows: the first quadrotor is \([-5 \rightarrow 1]\)(m), the second quadrotor is \([10 \rightarrow 3]\)(m), and the third quadrotor is \([0 5 1]\)(m). Details are described in the following sections.

Table 2 Conditions of numerical simulations.

<table>
<thead>
<tr>
<th>Controller ([\beta_0, \beta_1, \beta_2, \beta_3])</th>
<th>CASE I</th>
<th>CASE II</th>
<th>CASE III</th>
</tr>
</thead>
<tbody>
<tr>
<td>([1.5 13.3])</td>
<td>([1.5 9.013])</td>
<td>([1.5 7.53])</td>
<td></td>
</tr>
</tbody>
</table>

4.1.1 Network Structure

We consider a single network which has a directed spanning tree and has interactive communication links between every pair of the distinct quadrotors as shown in Fig. 3. The main characteristics is that the only first quadrotor can obtain information directly from their leader, and that the other quadrotors obtain information from their leader through the others indirectly. The nonzero minimum eigenvalue of the graph Laplacian of this network is \(\lambda_{\min} = 0.2\).

4.1.2 Controller Gain

We consider three pairs of controller gains as shown in Table 2. First, the controller gains are selected so as to satisfy all the requirements of (18) - (20) in the case of CASE I. This case will produce a desired stable result. Second, the controller gains are selected so as not to satisfy the only requirement of (19) in
the case of CASE II, that is, the controller gains are selected to satisfy $\lambda_{\min} = \beta_2^2/(\beta_1\beta_2\beta_3 - \beta_0\beta_3^2)$. This case will produce a balanced result. Last, the controller gains are selected so as not to satisfy the only requirement of (19), that is, the controller gains are selected to satisfy $\lambda_{\min} < \phi_2^2/(\phi_1\phi_2\phi_3 - \phi_0\phi_3^2)$. This case will produce an unstable result.

4.1.3 Leader’s Path

We consider a single leader’s path. The leader flies at a constant altitude and flies in an elliptical orbit: the major axis is 40(m), the minor axis is 35(m), the cycle is 200(s), and the initial position is [0 0 1](m).

4.1.4 Formation Configuration

We consider a single formation configuration. Each quadrotor keeps a certain distance from the leader, and also keeps a certain azimuth from the traveling direction of the leader. Note that the traveling direction of the leader is formulated as a unit velocity vector $\hat{r}_N + \dfrac{1}{||\hat{r}_N||} r_0$. We describe this information as a desired relative position $d_i$ for the quadrotor $i$. This is given by

$$
\begin{cases}
  d_i = \dfrac{||d_i||}{||r_{N+1}||} R(\theta_i) r_{N+1} & (||r_{N+1}|| \geq 0.1) \\
  d_i = \dfrac{||d_i||}{||r_{N+1}||} R(\theta_i) r_0 & (||r_{N+1}|| < 0.1)
\end{cases}
$$

where $||d_i||$ is a relative distance between the quadrotor $i$ and the leader, $\theta_i$ is an azimuth which goes counterclockwise from the leader’s traveling direction as shown in Fig. 4, and $r_0 = [1 0 0]^T$. Also, $R(\theta_i)$ is a two-dimensional rotation matrix given by

$$
R(\theta_i) = \begin{bmatrix}
  \cos(\theta_i) & -\sin(\theta_i) \\
  \sin(\theta_i) & \cos(\theta_i)
\end{bmatrix}.
$$

Note that the desired relative position $d_i$ is set to a certain fixed point which does not depend on the leader’s traveling direction while the velocity is slower than 0.1(m/s).

The values of $||d_i||$ and $\theta_i$ are shown in Table 3.

<table>
<thead>
<tr>
<th>Quadrotor</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative Distance $</td>
<td></td>
<td>d_i</td>
<td></td>
</tr>
<tr>
<td>Azimuth $\theta_i$(deg)</td>
<td>0°</td>
<td>120°</td>
<td>240°</td>
</tr>
</tbody>
</table>

4.2 Simulation Results

We show the results of the simulation which is carried out for 60(s) under the cases shown in Table 2. Figures 5, 6 and 7 show the results of the case CASE I, Figs. 8 and 9 show the results of the case CASE II, and Figs. 10 and 11 show the results of the case CASE III.

Figures 5, 8, and 10 show the trajectories of the quadrotors and the leader, and also show their positions every 10 seconds. Figures 6, 9 and 11 show the difference from the desired position, and Fig. 7 shows all the state variables of the first quadrotor.

Now, let us examine the results. The results of the CASE I show that the control objective is achieved when the appropriate controller gains are applied. However, the results in the CASE II show that the states of the quadrotors are in equilibrium, and the results in the CASE III show that the states of the quadrotors are in divergence as mentioned in section 4.1.2. These results also indicate the validity of the conditions for the controller gains.
5. Conclusions and Future Work

The authors showed that a linearized model of UAVs like quadrotors can be expressed as a fourth-order system, and that a network structure among the UAVs can be modeled using graph theory, then proposed a formation control algorithm for the quadrotors expressed as a fourth-order system which allows for dynamics. We directly extended the formation control algorithm for a first-order system which did not allow for dynamics, and also applied a leader-follower structure so that a leader can provide the quadrotors with commands to make geometric configuration of the formation. There are four controller gains for the formation control algorithm. We can easily find the controller gains to achieve the control objectives. The network composed of the quadrotors and the leader must satisfy the requirements that one of the quadrotor directly obtains commands from the leader, and that the network has interactive communication links between every pair of the distinct quadrotors. In other words, the network satisfying the requirements is enough to achieve the control objective, even when all of the quadrotors are not directly connected from the leader.

Our extensive simulation results showed that the proposed algorithm was validated and effective for formation flying among a group of the quadrotors with a fourth-order dynamics and the leader.

We did not consider the collisions between the quadrotors in this paper. Hence, studying and designing a control algorithm considering collision avoidance will be a challenging problem for future work in this area.

References


2004.

---

**Yasuhiro KURIKI**

He received his B.S., and M.S. degrees from The University of Tokyo, Japan, in 1999, and 2001, respectively. In 2012, he joined Keio University, where he is currently a Ph.D. student. His research interests include cooperative control problems, especially formation control problems. He is a member of JSME and JSASS.

---

**Tofu NAMERIKAWA (Member)**

He received the B.E., M.E and Dr. of Engineering degrees in Electrical and Computer Engineering from Kanazawa University, Japan, in 1991, 1993 and 1997, respectively. From 1994 until 2002 he was with Kanazawa University as an Assistant Professor. From 2002 until 2005, he was with Nagaoka University of Technology as an Associate Professor. From 2006 until 2009, he was with Kanazawa University again. In April 2009, he joined Keio University, where he is currently an Associate Professor at the Department of System Design Engineering. His main research interests are cooperative and distributed control theories, and their application to power network systems and networked robotic system. He is a member of ISCE and IEEE.