IET Control Theory & Applications

Research Article

Distributed backstepping based control of multiple UAV formation flight subject to time delays

ISSN 1751-8644
Received on 4th October 2019
Revised 2nd February 2020
Accepted on 20th April 2020
doi: 10.1049/iet-cta.2019.1151
www.ietdl.org

Yusuf Kartal[1] ✉, *Kamesh Subbarao*[1], *Nicholas R. Gans*[2], *Atilla Dogan*[1], *Frank Lewis*[2]

[1] *Aerospace Engineering, University of Texas at Arlignton, 500 W 1st St, Arlington, TX, USA*
[2] *University of Texas at Arlington Research Institute, Forth Worth 76118, TX, USA*
✉ *E-mail: yusuf.kartal@mavs.uta.edu*

Abstract: In this study, the authors propose a backstepping-based, distributed formation control method that is stable independent of time delays in communication among multiple unmanned aerial vehicles (UAVs). Centralised formation control of UAVs requires each agent to maintain a separation distance from other agents, which burdens the communication network of the UAVs. To overcome this problem, the authors consider a distributed control scheme wherein each agent updates its attitude and position based on the state information gathered through its neighbours. Instead of directly controlling the thrust generated by the propellers, they partition the mathematical model of the UAV into two subsystems, a linear attitude control loop and a non-linear position control loop. A backstepping-based outer position controller is then designed that interfaces seamlessly with the inner attitude controller of the cascaded control system. The closed-loop stability is established using a rigorous Lyapunov–Krasovskii analysis under the influence of distributed network time delays. Using the directed graph topology and a distributed backstepping structure, it is shown that the stability criterion is delay-independent. The proposed control algorithms are verified in simulation and then implemented in hardware, and actual flight test experiments prove the validity of these algorithms.

1 Introduction

Inspired by the naturally occurring biological groups such as herds and flocks where each member acts only under the influence of its neighbours [1], formation flight of quadrotors and other unmanned aerial vehicles (UAVs) has drawn great attention in recent years, due to their capability to perform certain tasks such as transportation [2], surveillance and reconnaissance, [3] and target search and detection [4]. With the increase in demand for UAVs to work together to accomplish these tasks, several real-life challenges need be addressed that include the implementation, test, and validation of the control algorithms in flight test experiments. In particular, we examine the challenge of designing a cooperative controller for UAVs to provide robust capabilities like, performance despite communication time-delays in a leader–follower formation.

Formation control is a type of multi-agent architecture that relies on relative motion of agents [5]. There exist different approaches to ensure formation control for multi-agent systems in the control community. Three recognised categories are behaviour-based approaches [6], virtual structure-based approaches [7], and leader–follower approaches [8]. In the behaviour-based approaches, each agent of the formation acts according to predefined behaviour. This approach is behaviourally inflexible since motion is predefined. Alternately, virtual structure-based approach introduces a virtual vehicle for each vehicle in the formation and transforms the formation problem into a trajectory tracking problem. However, since the virtual vehicles are not exposed to any type of disturbance in the environment, there is a high chance that the followers break formation in the event of unexpected environmental disturbances. On the other hand, the leader–follower approach is easy to implement, and all agents react to any environmental change, but the network delay must be examined well to maintain formation.

Several works deal with linear dynamics of multi-agent systems [9–13]. Particularly, [10] reveals some of the necessary and sufficient conditions to achieve predefined time-varying formations with switching interaction topologies based on the algebraic Riccati equation. Rui *et al.* [13] propose a distributed adaptive control technique that uses adaptive gain scheduling to tune the coupling weights between the individuals of the multi-agent system. Whereas, [14] uses the same technique to satisfy prescribed H-∞ like performance and to manage the side effects of uncertainties in the system dynamics. However, since most physical systems are intrinsically non-linear, these linear cooperative control methods cannot be applied directly [15]. Therefore, considerable number of works studied non-linear dynamics of the multi-agent systems [15–19]. Neural-network based adaptive control and distributed impulsive control methods are examined to achieve leader–follower consensus with the class of non-linear multi-agent systems in [15, 16], respectively. The authors prove the stability of consensus error dynamics with well-known Lyapunov stability analysis techniques.

However, the problem of designing leader–follower formation strategies in which agents experience distributed network delays with pinning gain control requires more attention. One significant challenge associated with this relates to the construction of an appropriate compact non-linear mathematical model of the multi-UAV system. In [20], authors study the effect of commensurate time delays on the closed-loop stability using the retarded functional differential equation (RFDE) form. By modelling the dynamics of the UAV as a double integrator, [21, 22] treat each agent of the multi-UAV system as a point-mass system to apply time-varying consensus-based approaches on the multi-UAV system. This is a gross oversimplification of the UAV dynamics, especially for the quadrotor platforms considered in this paper. The inner/outer control loop partitioning allows us to deal with the delays occurring in the communication network of the UAVs, and to show that the stability of the outer control loop of the cascaded system is independent of delay. It is then shown that the closed-loop error dynamics of the whole system is also stable, independent of time delay.

Moreover, authors of [23–27] use non-linear backstepping control method to deal with recursive design structure. This approach enables designer to solve the stabilisation problem partially for each submodule of the system of interest. Particularly, Krsti'c *et al.* [23] propose the adaptive backstepping control scheme to possess stronger stability properties while dealing with parametric uncertainties. Skjetne *et al.* [24] extends the adaptive backstepping control scheme in [23], to achieve both convergence to the path and predetermined dynamic behaviour along the path

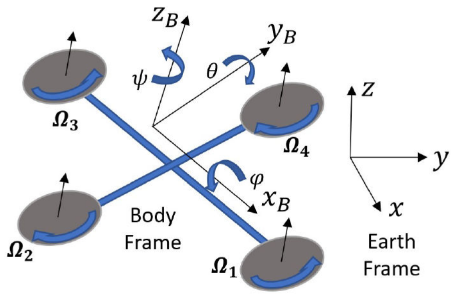

Fig. 1 *Coordinate systems of the quadrotor*

simultaneously. In [25], integrator backstepping and quaternion feedback is adopted to stabilise the attitude of a micro-satellite. Li and Zhao [27] work on improving transient response of the closed-loop system by presenting a generalised backstepping process, based on the solvability of virtual controllers.

There are three major contributions of this work. Firstly, we use the second-order non-linear dynamics of UAVs and synthesise a novel rigorous distributed backstepping control technique that has a form that easily extends to multiple UAV distributed control. Secondly, we partition the mathematical model of UAVs into two subsystems, an inner attitude control loop which is built into the quadrotor, and outer position controllers that consider relative motion of neighbours in formation flight. This allows us to rigorously analyse the delays occurring in the communication network of the UAVs. We prove that the stability of the outer control loop of the cascaded system is independent of delay, which implies that the closed-loop error dynamics of the whole system is also stable independent of delay. Lastly, we employ the directed graph topology to design formation control of the multi-UAV system, which reduces the work burden for the UAV communication network. It is rigorously shown that the stability criterion is delay-independent when each agent of the formation experiences distributed delays while communicating with its neighbours. The actual flight tests that show the validity and robustness of the developed control algorithms.

The rest of the paper is organised as follows. In Section 2, we provide the preliminaries of the mathematical model of the quadrotor and graph theory to understand the basics of the distributed control approach. Section 3 brings an analysis of the control structures proposed, which involves the inner attitude controller and backstepping-based position controller for the task of trajectory tracking. We first explain the attitude controller design procedure for the quadrotors. Then we show the stability analysis of the backstepping control method. Section 4 illustrates how to extend the backstepping control algorithm to control multiple agents using the distributed backstepping tracker, which has a stable delay-independent system structure under the influence of non-constant distributed delays. Sections 5 and 6 show the flight tests on a real UAV, where we illustrate the trajectories followed by AR.Drone 2.0 quadrotor with a full non-linear backstepping tracker and by multiple AR.Drone 2.0's with a distributed backstepping trackers.

2 Preliminaries

The goal of this paper is to design a distributed controller for multi-UAV systems. Flying in a formation requires the agents to maintain separate distance from each other, which burdens the communication network and induces communication delays. In this section, we give preliminaries of a mathematical model of the quadrotor and graph theory to clarify the idea of the backstepping-based, distributed formation control method. In the next section, we present backstepping control for a single UAV. Then in Section 4, we present the formation controller with the network delays.

2.1 Mathematical model

This section introduces the standard non-linear model of the quadrotor dynamics. To localise the quadrotor position, we use the Earth fixed frame. The origin of the three-dimensional (3D) axis system of the Body frame is assumed to be at the centre of mass of the quadrotor. The kinematics of the Euler angle rates can be expressed as

$$\boldsymbol{w_B} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\varphi & c\theta s\varphi \\ 0 & -s\varphi & c\theta c\varphi \end{bmatrix} \dot{\boldsymbol{\eta}} \tag{1}$$

where c and s refer to cosine and sine, respectively, and $\boldsymbol{w_B} \in \mathbb{R}^3$ is the angular velocity in the Body frame components. Particularly, p is the roll rate, q is the pitch rate, and r is the yaw rate defined in the Body frame. Moreover, $\boldsymbol{\eta} \in \mathbb{R}^3$ is the Euler angle vector (roll, pitch, and yaw), i.e. $\boldsymbol{\eta} = [\varphi\, \theta\, \psi]^T$. Note that positive directions of Euler angles determined by right-hand rule, which are shown in Fig. 1.

The rotational dynamics are given by

$$\boldsymbol{I_B}\dot{\boldsymbol{w}}_{\boldsymbol{B}} = \boldsymbol{S}(\boldsymbol{w_B})\boldsymbol{I_B}\boldsymbol{w_B} + \boldsymbol{\tau_B} \tag{2}$$

where $\boldsymbol{S}(\boldsymbol{w_B}) \in \mathbb{R}^{3\times 3}$ is the skew-symmetric matrix [22], $\boldsymbol{\tau_B} = \left[\tau_\varphi\, \tau_\theta\, \tau_\psi\right]^T$ is the torque vector and $\boldsymbol{I_B} \in \mathbb{R}^{3\times 3}$ is the inertia matrix defined in Body frame.

The translational dynamics of the quadrotor, ignoring any aerodynamic effects, is expressed in the Body frame is obtained to be

$$m\dot{\boldsymbol{U}} = \begin{bmatrix} 0 \\ 0 \\ \mu \end{bmatrix} + \boldsymbol{R}\boldsymbol{F_g} \tag{3}$$

where $\boldsymbol{U} = [u\, v\, w]^T$ is the velocity vector defined in the Body frame, μ is the total thrust produced by rotors in the Body frame z_B-axis. Note that m is mass of the rigid body, $\boldsymbol{F_g} = [0\,0\ -mg]^T$ is the gravitational force vector and $\boldsymbol{R} \in \mathbb{R}^{3\times 3}$ is the rotation matrix from the Earth frame to the Body frame. Moreover u, v, and w stand for the velocities of the quadrotor in Body-axis coordinate system given in Fig. 1. We obtain this rotation matrix using the yaw-pitch-roll (3-2-1) sequence. It is given by

$$\boldsymbol{R} = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ -c\varphi s\psi + s\varphi s\theta c\psi & c\varphi c\psi + s\varphi s\theta s\psi & s\varphi c\theta \\ s\varphi s\psi + c\varphi s\theta c\psi & -s\varphi c\psi + c\varphi s\theta s\psi & c\varphi c\theta \end{bmatrix} \tag{4}$$

Note that $\boldsymbol{R}$ belongs to the special orthogonal group and is of rank 3, or $SO(3)$, whose determinant is equal to 1.

The translational dynamics of the quadrotor in the Earth frame is then formulated as

$$m\ddot{\boldsymbol{\xi}} = m\dot{\boldsymbol{V}} = \boldsymbol{F} + \boldsymbol{F_g} \tag{5}$$

where $\boldsymbol{\xi} = [x\, y\, z]^T$ and $\boldsymbol{V} \in \mathbb{R}^3$ denotes the position and velocity vectors in the Earth frame, respectively. $\boldsymbol{F} \in \mathbb{R}^3$ is the input force vector defined in the Earth frame. Then (3) and (5) give the following relation:

$$\boldsymbol{F} = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \boldsymbol{R}^T \begin{bmatrix} 0 \\ 0 \\ \mu \end{bmatrix} = \begin{bmatrix} \mu(s\varphi s\psi + c\varphi s\theta c\psi) \\ \mu(- s\varphi c\psi + c\varphi s\theta s\psi) \\ \mu(c\varphi c\theta) \end{bmatrix} \tag{6}$$

2.2 Graph theory

A graph is constructed with a pair $\boldsymbol{G} = (V, E)$, where the set $\mathrm{V} = \{\mathrm{v}_1, \ldots, \mathrm{v}_N\}$ defines the nodes or vertices, and E defines edges or arcs. The set E is composed of edge pairs $(\mathrm{v}_i, \mathrm{v}_j)$. If $(\mathrm{v}_i, \mathrm{v}_j)$ is equal to $(\mathrm{v}_j, \mathrm{v}_i)$ $\forall i, j \in [0, \mathrm{N}]$, $i \neq j$, then graph is said to be bidirectional. Each edge $\left(\mathrm{v}_j, \mathrm{v}_i\right) \in \mathrm{E}$, has a weight $a_{ij} > 0$ if and only if there exists a connection from node j to i. The graph is

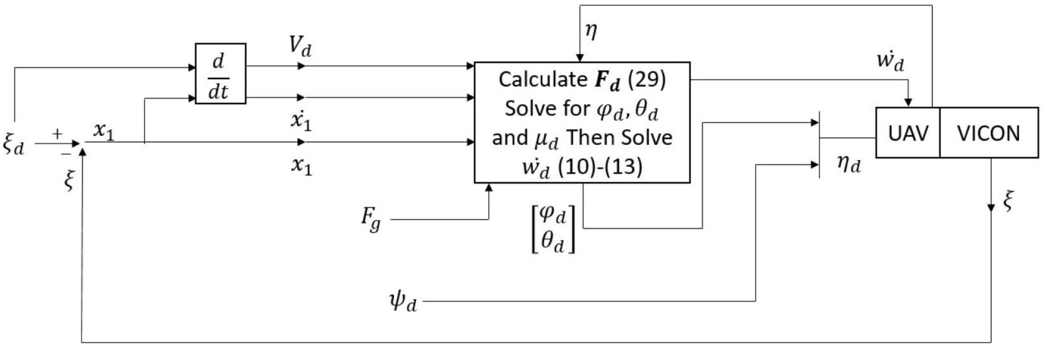
$\frac{d}{dt}$
V_d
$\dot{x}_1$
x_1
x_1
ξ_d
+
−
ξ
η
Calculate $\boldsymbol{F_d}$ (29) Solve for φ_d, θ_d and μ_d Then Solve $\dot{w}_d$ (10)-(13)
$\dot{w}_d$
UAV
VICON
η_d
F_g
$\begin{bmatrix}\varphi_d\\ \theta_d\end{bmatrix}$
ψ_d
ξ

Fig. 2 *Desired input states calculation for the attitude controller*

called undirected if $a_{ij} = a_{ji}$, $\forall i, j$.The undirected graph is said to be weight balanced, which leads to symmetric adjacency matrix $\boldsymbol{A}$.

The diagonal matrix $\boldsymbol{D}$ is the ith row sum of $\boldsymbol{A}$ or weighted in-degree. Then, the Laplacian matrix is defined as

$$\mathscr{L} = \boldsymbol{D} - \boldsymbol{A}\,. \tag{7}$$

In this paper, the edge weights represent the trust between quadrotors, which are nodes of the formation graph. We create a graph topology based on adjacency or connectivity matrix $\mathbf{A} = [a_{ij}]$, realising that $a_{ii} = 0$. The Laplacian matrices of all undirected graphs are real symmetric matrices. On the other hand, this is not valid for the digraphs. One of the contributions of this paper is proving consensus of the UAVs by adopting directed graph topology.

3 Backstepping control

This section explains the full non-linear backstepping control design for the quadrotor. The backstepping control structure derived here is shown in Fig. 2. This is a non-standard backstepping controller, has a novel form that allows direct extension to multiple interacting UAV control. Specifically, we derive the UAV error dynamics (33), which has a special form that is easily extended to multiple UAV formation control in Section 4.

To apply the backstepping control method to the system defined in (5), we begin by adding and removing $\boldsymbol{F_d}$, an ideal virtual force input, and obtain the Newtonian model in terms of the desired forces

$$m\ddot{\boldsymbol{\xi}} = m\dot{\boldsymbol{V}} = \boldsymbol{F_d} + \boldsymbol{F_g} + \widetilde{\boldsymbol{F}}_{\boldsymbol{d}} \tag{8}$$

where $\widetilde{\boldsymbol{F}}_{\boldsymbol{d}} = \boldsymbol{F} - \boldsymbol{F_d}$. In Section 3.1, we show how to obtain desired Euler angle vector $\boldsymbol{\eta_d} = [\varphi_d\, \theta_d\, \psi_d]^T$ to generate $\boldsymbol{F_d}$, and the time rate of change of desired vertical speed, $\dot{w}_d$. Then in Section 3.2, $\boldsymbol{\tau_B}$ in (2) is designed using $\boldsymbol{\eta_d} = [\varphi_d\, \theta_d\, \psi_d]^T$ and $\dot{w}_d$ to get $\widetilde{\boldsymbol{F}}_{\boldsymbol{d}} \to \boldsymbol{0}$. Lastly in Section 3.3, $\boldsymbol{F_d}$ is selected to get $\boldsymbol{\xi} \to \boldsymbol{\xi_d}$, where $\boldsymbol{\xi_d} = [x_d\, y_d\, z_d]^T$ is the given desired position vector in the Earth frame. Proof of stability and tracking error convergence is given in Section 3.3.

3.1 Desired Euler angles

In Section 3.3, we show how to compute desired force vector $\boldsymbol{F_d}$ to obtain position and velocity tracking. Herein we show how to compute $\boldsymbol{\eta_d}$ and $\dot{w}_d$ from the desired force data $\boldsymbol{F_d}$ by using the inverse kinematics approach. Suppose we are given desired force $\boldsymbol{F_d}$. Note that from (6)

$$\boldsymbol{F_d} = \begin{bmatrix} f_{xd} \\ f_{yd} \\ f_{zd} \end{bmatrix} = \begin{bmatrix} \mu_d(s\varphi_d s\psi_d + c\varphi_d s\theta_d c\psi_d) \\ \mu_d(-\, s\varphi_d c\psi_d + c\varphi_d s\theta_d s\psi_d) \\ \mu_d(c\varphi_d c\theta_d) \end{bmatrix} \tag{9}$$

where μ_d is the desired thrust in the Body frame. Equation (9) can be solved for the desired Euler angles

$$\begin{aligned} \tan\theta_d &= \frac{f_{xd}\cos\psi_d + f_{yd}\sin\psi_d}{f_{zd}}, \\ \theta_d &= \tan^{-1}\left(\frac{f_{xd}\cos\psi_d + f_{yd}\sin\psi_d}{f_{zd}}\right), \end{aligned} \tag{10}$$

$$\begin{aligned} \tan\varphi_d &= \frac{\cos\theta_d\,(f_{xd}\sin\psi_d - f_{yd}\cos\psi_d)}{f_{zd}}, \\ \varphi_d &= \tan^{-1}\left(\frac{\cos\theta_d\,(f_{xd}\sin\psi_d - f_{yd}\cos\psi_d)}{f_{zd}}\right), \end{aligned} \tag{11}$$

$$\mu_d = \frac{f_{zd}}{\cos\varphi_d\cos\theta_d}\,. \tag{12}$$

and $f_{zd} \neq 0$. Although it should be mentioned that $f_{zd} = 0$ only if θ_d and/or $\varphi_d = \pm\frac{\pi}{2}$ or $\mu_d = 0$. The condition $\theta_d = \pm\frac{\pi}{2}$ or $\varphi_d = \pm\frac{\pi}{2}$ correspond to singular orientations of the quadrotor and our domain of operation for θ and φ is $\left(-\frac{\pi}{2}\ \frac{\pi}{2}\right)$. Further $\mu_d = 0$ would correspond to zero total thrust. Furthermore, notice that ψ_d can be arbitrarily prescribed, and only the variables θ_d, φ_d and μ_d must be found. The inner loop control design of the backstepping method requires the time rate of change of the desired vertical speed in the Body frame, $\dot{w}_d$, which is calculated by using (3) such that

$$\dot{w}_d = \frac{\mu_d}{m} - g\cos\varphi_d\cos\theta_d\,. \tag{13}$$

Note that the information of time rate of change of the desired vertical speed or simply desired vertical acceleration acts as an input of the attitude controller loop will be given in Section 3.2. Derivation of this data is essential to control the height of the UAV, while accomplishing the path tracking objective accurately in 3D space.

3.2 Inner attitude control loop

In this section, we explain the inner attitude control of the backstepping method for the quadrotor. The attitude controller is generally built-in to the UAV and cannot be modified. This implies that the built-in attitude controller is assumed to track the quantities $\boldsymbol{\eta_d}$ and $\dot{w}_d$.

We begin with deriving the desired Euler rates $\boldsymbol{w_{Bd}} = [p_d\, q_d\, r_d]^T$ by using (1) and $\boldsymbol{\eta_d}$ such that

$$\begin{bmatrix} p_d \\ q_d \\ r_d \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s\theta_d \\ 0 & c\varphi_d & c\theta_d s\varphi_d \\ 0 & -s\varphi_d & c\theta_d c\varphi_d \end{bmatrix} \begin{bmatrix} \dot{\varphi}_d \\ \dot{\theta}_d \\ \dot{\psi}_d \end{bmatrix}. \tag{14}$$

Then the following PID controller is designed to generate changes in angular velocity of the propellers

$$rCl\begin{bmatrix}\Delta\Omega_{\varphi}\\ \Delta\Omega_{\theta}\\ \Delta\Omega_{\psi}\end{bmatrix}=\begin{bmatrix}P_{\varphi}(\varphi_d-\varphi)+D_{\varphi}(p_d-p)+I_{\varphi}\int(\varphi_d-\varphi)\\ P_{\theta}(\theta_d-\theta)+D_{\theta}(q_d-q)+I_{\theta}\int(\theta_d-\theta)\\ P_{\psi}(\psi_d-\psi)+D_{\psi}(r_d-r)+I_{\psi}\int(\psi_d-\psi)\end{bmatrix}. \quad (15)$$

Furthermore, (15) is used to obtain the desired angular velocity of each rotor [2] such that

$$\begin{bmatrix}\Omega_{1d}\\ \Omega_{2d}\\ \Omega_{3d}\\ \Omega_{4d}\end{bmatrix}=\begin{bmatrix}1&0&-1&1\\ 1&-1&0&-1\\ 1&0&1&1\\ 1&1&0&-1\end{bmatrix}\begin{bmatrix}\Omega_h+\Delta\Omega_{net}\\ \Delta\Omega_{\varphi}\\ \Delta\Omega_{\theta}\\ \Delta\Omega_{\psi}\end{bmatrix} \quad (16)$$

where Ω_{id}, $i = 1, 2, 3, 4$, corresponds to the desired angular velocities of the rotors and k is the thrust factor. Ω_h is the rotor speed required to hover such that

$$\Omega_h=\sqrt{\frac{mg}{4k}} \quad (17)$$

$\Delta\Omega_{net}$ is the outcome of desired vertical acceleration in the Body frame, $\dot{w}_d$ (13), in the form of

$$\Delta\Omega_{net}=\frac{m}{8\mathrm{k}\Omega_h}\dot{w}_d\,. \quad (18)$$

Notice that while producing Ω_h keeps the quadrotor at nominal condition (hover), $\Delta\Omega_{net}$ moves the UAV along z_B-axis. In addition, producing $\Delta\Omega_{\varphi}$, $\Delta\Omega_{\theta}$, and $\Delta\Omega_{\psi}$ deviates quadrotor from hover by resulting in roll, pitch and yaw, respectively. Finally, (16) yields the following torque vector expression, which stabilises the rotational dynamics (2):

$$\boldsymbol{\tau_B}=\begin{bmatrix}\tau_{\varphi}\\ \tau_{\theta}\\ \tau_{\psi}\end{bmatrix}=\begin{bmatrix}lk({\Omega_{4\mathrm{d}}}^2-{\Omega_{2\mathrm{d}}}^2)\\ lk({\Omega_{3\mathrm{d}}}^2-{\Omega_{1\mathrm{d}}}^2)\\ d({\Omega_{1\mathrm{d}}}^2+{\Omega_{3\mathrm{d}}}^2-{\Omega_{2\mathrm{d}}}^2-{\Omega_{4\mathrm{d}}}^2)\end{bmatrix} \quad (19)$$

where l is the lever length, and d is the drag factor. The direction of angular velocities for each rotor is given in Fig. 1, while first and third rotor turn anti-clockwise, the other two turn clockwise to cancel the yawing moments generated when the quadrotor is at nominal condition. The total thrust, μ_d, is equal to the sum of thrusts generated by each rotor, that is

$$\mu_d=k({\Omega_{1\mathrm{d}}}^2+{\Omega_{2\mathrm{d}}}^2+{\Omega_{3\mathrm{d}}}^2+{\Omega_{4\mathrm{d}}}^2)\,. \quad (20)$$

3.3 Outer position control loop

This section explains the outer position control loop of full non-linear backstepping design for the quadrotor. We derive an expression for $\boldsymbol{F_d}$ that guarantees the dynamics of (8) are stable. A main result is the form (33) for the error dynamics, which has a special structure that is directly extended to multiple quadrotors in Section 4.

Begin with defining the position and velocity errors in the Earth frame as

$$\begin{aligned}\boldsymbol{x}_1&=\boldsymbol{e_p}=\boldsymbol{\xi_d}-\boldsymbol{\xi},\\ \boldsymbol{x}_2&=\boldsymbol{e_v}=\boldsymbol{V_d}-\boldsymbol{V}\end{aligned} \quad (21)$$

where $\boldsymbol{V_d}=\dot{\boldsymbol{\xi}}_{\boldsymbol{d}}\in\mathbb{R}^3$ is the desired velocity vector in the Earth frame.

The objective of the paper is to demonstrate the performance of the cooperative controller for any Commercial off the Shelf quadrotor platform with an inbuilt attitude controller. In such a situation, the assumption is reasonable that the inbuilt attitude controllers (typically PID) will accomplish this task. Thus, the inner loop attitude controller is not analysed further. Then to prove the convergence of error dynamics (21), which is the second step of backstepping method given in Section 3, we make the following assumption.

Assumption 1: The inner attitude controller (19) tracks the Euler angles (10) and (11) and vertical acceleration (13). Hence, the equilibrium of inner attitude control loop is stable.

The next main theorem shows how to compute $\boldsymbol{F_d}$ to guarantee stable position and velocity tracking of (8).

Theorem 1: Under Assumption 1, the following control law, applied to the system governed by (8) ensures that the position and velocity tracking errors in (21) → 0 as $t\to\infty$

$$\begin{aligned}\boldsymbol{F_d}&=m\dot{\boldsymbol{V}}_{\boldsymbol{d}}-\boldsymbol{F_g}+(m\boldsymbol{K}_1+m\boldsymbol{K}_2)\dot{\boldsymbol{x}}_1\\ &\quad+(m\boldsymbol{I}_{3\times3}+m\boldsymbol{K}_1\boldsymbol{K}_2)\boldsymbol{x}_1\,.\end{aligned} \quad (22)$$

Proof: The error dynamics are derived as

$$\dot{\boldsymbol{x}}_1-\boldsymbol{x}_2+\boldsymbol{x}_2^v-\boldsymbol{x}_2^v=\boldsymbol{0} \quad (23)$$

where $\boldsymbol{x}_2^v$ is a virtual control signal. Moreover, by using (8) the error dynamics become

$$\dot{\boldsymbol{x}}_2-\dot{\boldsymbol{V}}_{\boldsymbol{d}}+\frac{\boldsymbol{F_d}}{m}+\frac{\boldsymbol{F_g}}{m}+\frac{\widetilde{\boldsymbol{F}}_{\boldsymbol{d}}}{m}=\boldsymbol{0}\,. \quad (24)$$

Then, define the following velocity error mismatch variable:

$$\tilde{\boldsymbol{x}}_2=\ \boldsymbol{x}_2^v-\boldsymbol{x}_2\,. \quad (25)$$

Substituting (25) in (23)

$$\dot{\boldsymbol{x}}_1-\boldsymbol{x}_2^v=\ -\tilde{\boldsymbol{x}}_2\,. \quad (26)$$

Now we pick $\boldsymbol{x}_2^v=\ -\boldsymbol{K}_1\boldsymbol{x}_1$ where $\boldsymbol{K}_1\in\mathbb{R}^{3\times3}$ is a diagonal positive definite matrix. Then (26) becomes

$$\dot{\boldsymbol{x}}_1+\boldsymbol{K}_1\boldsymbol{x}_1=\ -\tilde{\boldsymbol{x}}_2\,. \quad (27)$$

To examine the stability of (27), we pick the Lyapunov function candidate as follows:

$$\mathscr{V}=\frac{1}{2}{\boldsymbol{x}_1}^{\boldsymbol{T}}\boldsymbol{x}_1+\frac{1}{2}{\tilde{\boldsymbol{x}}_2}^{\boldsymbol{T}}\tilde{\boldsymbol{x}}_2\,. \quad (28)$$

Then the derivative of Lyapunov function candidate is derived using (24) and (26) as

$$\begin{aligned}\dot{\mathscr{V}}&={\boldsymbol{x}_1}^{\boldsymbol{T}}\dot{\boldsymbol{x}}_1+{\tilde{\boldsymbol{x}}_2}^{\boldsymbol{T}}\dot{\tilde{\boldsymbol{x}}}_2\\ &={\boldsymbol{x}_1}^{\boldsymbol{T}}(-\boldsymbol{K}_1\boldsymbol{x}_1-\tilde{\boldsymbol{x}}_2)\\ &\quad+{\tilde{\boldsymbol{x}}_2}^{\boldsymbol{T}}\left(\dot{\boldsymbol{x}}_2^v-\dot{\boldsymbol{V}}_{\boldsymbol{d}}+\frac{\boldsymbol{F_d}}{m}+\frac{\boldsymbol{F_g}}{m}+\frac{\widetilde{\boldsymbol{F}}_{\boldsymbol{d}}}{m}\right).\end{aligned} \quad (29)$$

To have strictly negative definite Lyapunov function derivative, we set $\boldsymbol{F_d}$ as

$$\begin{aligned}\boldsymbol{F_d}&=m\dot{\boldsymbol{V}}_{\boldsymbol{d}}-\boldsymbol{F_g}-m\dot{\boldsymbol{x}}_2^v+m\boldsymbol{x}_1-m\boldsymbol{K}_2\tilde{\boldsymbol{x}}_2\\ &=m\dot{\boldsymbol{V}}_{\boldsymbol{d}}-\boldsymbol{F_g}+(m\boldsymbol{K}_1+m\boldsymbol{K}_2)\dot{\boldsymbol{x}}_1\\ &\quad+(m\boldsymbol{I}_{3\times3}+m\boldsymbol{K}_1\boldsymbol{K}_2)\boldsymbol{x}_1\end{aligned} \quad (30)$$

where $\boldsymbol{K}_2\in\mathbb{R}^{3\times3}$ is a diagonal positive definite matrix. Then, (29) becomes

$$\begin{aligned}\dot{\mathscr{V}} &= -\boldsymbol{x}_1{}^T\boldsymbol{K}_1\boldsymbol{x}_1 - \tilde{\boldsymbol{x}}_2{}^T\boldsymbol{K}_2\tilde{\boldsymbol{x}}_2 + \tilde{\boldsymbol{x}}_2{}^T\frac{\tilde{\boldsymbol{F}}_d}{m},\\ &\le -\lambda_{min}(\boldsymbol{K}_1,\boldsymbol{K}_2)\|\,\tilde{\boldsymbol{x}}\,\|^2 + \tilde{\boldsymbol{x}}_2{}^T\frac{\tilde{\boldsymbol{F}}_d(0)}{m},\\ &\le -\lambda_{min}(\boldsymbol{K}_1,\boldsymbol{K}_2)\|\,\tilde{\boldsymbol{x}}\,\|^2 + \|\,\tilde{\boldsymbol{x}}_2\,\|\frac{\|\,\tilde{\boldsymbol{F}}_d(0)\,\|}{m}\end{aligned} \tag{31}$$

where $\tilde{\boldsymbol{x}} = \left[\boldsymbol{x}_1^T\,\tilde{\boldsymbol{x}}_2^T\right]^T$ and $\lambda_{min}(\boldsymbol{K}_1,\boldsymbol{K}_2)$ stands for min eigenvalue of $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$. Note that in the worst case scenario, $\lambda_{min}(\boldsymbol{K}_1,\boldsymbol{K}_2)$ must be bigger than $\|\,\tilde{\boldsymbol{F}}_d(0)\,\|/m$, which is a sufficient condition for asymptotic stability of origin. Moreover, from Assumption 1, the inner attitude control loop ensures that the quadrotor tracks the desired attitude angles φ_d, θ_d, and the desired thrust μ_d, i.e $\varphi \to \varphi_d$, $\theta \to \theta_d$, and $\mu \to \mu_d$. From the description of $\boldsymbol{F}$ and $\boldsymbol{F}_d$ in (6) and (9), respectively, we conclude that $\boldsymbol{F} \to \boldsymbol{F}_d$ and hence, $\tilde{\boldsymbol{F}}_d \to \boldsymbol{0}$. Then (31) becomes

$$\dot{\mathscr{V}} = -\boldsymbol{x}_1{}^T\boldsymbol{K}_1\boldsymbol{x}_1 - \tilde{\boldsymbol{x}}_2{}^T\boldsymbol{K}_2\tilde{\boldsymbol{x}}_2, \tag{32}$$

which is strictly negative definite since $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ are positive definite matrices. Note that $\tilde{\boldsymbol{x}}_2 \to \boldsymbol{0}$ implies $\boldsymbol{x}_2 \to \boldsymbol{x}_2^v$. Moreover, $\boldsymbol{x}_2^v \to \boldsymbol{0}$ as $\boldsymbol{x}_2^v = -\boldsymbol{K}_1\boldsymbol{x}_1$. Hence, $\boldsymbol{x}_2 \to \boldsymbol{0}$ and the origin $(\boldsymbol{0},\boldsymbol{0})$, which is the equilibrium of (21), is globally asymptotically stable. □

Using the control laws derived previously, from (30) the tracking error dynamics can be written in the state-space form as

$$\dot{\boldsymbol{x}} = \underbrace{\begin{bmatrix}\boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3}\\ -(\boldsymbol{K}_1\boldsymbol{K}_2 + \boldsymbol{I}_{3\times3}) & -(\boldsymbol{K}_1 + \boldsymbol{K}_2)\end{bmatrix}}_{\boldsymbol{J}}\boldsymbol{x} \tag{33}$$

where $\boldsymbol{x} = \left[\boldsymbol{x}_1{}^T \quad \boldsymbol{x}_2{}^T\right]^T$. Note that $\boldsymbol{J}$ is Hurwitz. This form is instrumental in designing formation controllers for multiple UAV in the next section.

4 Distributed backstepping position control loop of multiple UAV with network delays

This section provides the connection of outer position control loop of backstepping method defined in Section 3.3 to the distributed multi-agent case. The error dynamics (33) are in a novel form which is easily extended in this section to multiple UAV formation.

We first treat the distributed system as delay-less. Then we perform rigorous stability analysis when agents experience both constant and distributed delays. Define dynamics (5) for each agent as

$$m\ddot{\boldsymbol{\xi}}_i = m\dot{\boldsymbol{V}}_i = \boldsymbol{F}_i + \boldsymbol{F}_g, \quad \forall i = 1,\dots,N. \tag{34}$$

4.1 No communication delay

In this section, we first extend the error dynamics (33) to multiple quadrotors. If there is no communication delay, define the position-based consensus error

$$\boldsymbol{e}_{p_i} = \sum_{j\epsilon N_i} a_{ij}(\boldsymbol{\xi}_j - \boldsymbol{\Delta}_j - \boldsymbol{\xi}_i + \boldsymbol{\Delta}_i) + g_i(\boldsymbol{\xi}_0 - \boldsymbol{\xi}_i + \boldsymbol{\Delta}_i) \tag{35}$$

$\forall i = 1,\dots,N$ where g_i is the pinning gain, $\boldsymbol{\Delta}_i$ (and $\boldsymbol{\Delta}_j$) is the n-dim constant tracking offset vector of the ith (and jth) UAV with respect to the n-dim position of the leader, $\boldsymbol{\xi}_0 \in \mathbb{R}^3$, of the formation. Lastly, N is the number of the UAVs in the formation. Note that g_i only takes values different than zero, if the node i is directly connected to the leader node. For the sake of simplicity, we use following vector notations:

$$\begin{aligned}\boldsymbol{e}_p^c &= \left[\boldsymbol{e}_{p_1}{}^T\boldsymbol{e}_{p_2}{}^T\dots\boldsymbol{e}_{p_N}{}^T\right]^T, \boldsymbol{e}_p{}^c \in \mathbb{R}^{Nn}\\ \underline{\boldsymbol{\Delta}} &= \left[\boldsymbol{\Delta}_1{}^T\boldsymbol{\Delta}_2{}^T\dots\boldsymbol{\Delta}_N{}^T\right]^T, \underline{\boldsymbol{\Delta}} \in \mathbb{R}^{Nn}\\ \boldsymbol{\xi}^c &= \left[\boldsymbol{\xi}_1{}^T\boldsymbol{\xi}_2{}^T\dots\boldsymbol{\xi}_N{}^T\right]^T, \boldsymbol{\xi}^c \in \mathbb{R}^{Nn}\\ \dot{\boldsymbol{\xi}}^c = \boldsymbol{V}^c &= \left[\boldsymbol{V}_1{}^T\boldsymbol{V}_2{}^T\dots\boldsymbol{V}_N{}^T\right]^T, \boldsymbol{V}^c \in \mathbb{R}^{Nn}\\ \boldsymbol{F}^c &= \left[\boldsymbol{F}_1{}^T\boldsymbol{F}_2{}^T\dots\boldsymbol{F}_N{}^T\right]^T, \boldsymbol{F}^c \in \mathbb{R}^{Nn}.\end{aligned} \tag{36}$$

By using (5), the global system dynamics for followers can be written as

$$m\dot{\boldsymbol{V}}^c = \boldsymbol{F}^c + \boldsymbol{1}_N \otimes \boldsymbol{F}_g. \tag{37}$$

Then, by using (7) and noting the fact $\mathscr{L}\boldsymbol{1}_N = \boldsymbol{0}_N$ since the row sum of $\mathscr{L}$ is zero, re-write (35) as

$$\begin{aligned}\boldsymbol{e}_p{}^c &= -((\mathscr{L} + \boldsymbol{G}) \otimes \boldsymbol{I}_n)(\boldsymbol{\xi}^c - \underline{\boldsymbol{\Delta}}) + (\boldsymbol{G} \otimes \boldsymbol{I}_n)(\boldsymbol{1}_N \otimes \boldsymbol{\xi}_0)\\ &= -((\mathscr{L} + \boldsymbol{G}) \otimes \boldsymbol{I}_n)(\boldsymbol{\xi}^c - \underline{\boldsymbol{\Delta}} - \boldsymbol{1}_N \otimes \boldsymbol{\xi}_0)\end{aligned} \tag{38}$$

where $\boldsymbol{1}_N$ is N-dim vector whose all elements are ones and $\boldsymbol{G} \in \mathbb{R}^{N\times N}$ is a diagonal pinning gain matrix with the diagonal elements of g_i $\forall i = 1,\dots,N$. In addition, $\otimes$ stands for the Kronecker product. Define $\underline{\boldsymbol{\xi}}_0 = \boldsymbol{1}_N \otimes \boldsymbol{\xi}_0$. Then (38) becomes

$$\boldsymbol{e}_p{}^c = ((\mathscr{L} + \boldsymbol{G}) \otimes \boldsymbol{I}_n)(\underline{\boldsymbol{\xi}}_0 + \underline{\boldsymbol{\Delta}} - \boldsymbol{\xi}^c). \tag{39}$$

The velocity-based consensus error is

$$\boldsymbol{e}_v{}^c = ((\mathscr{L} + \boldsymbol{G}) \otimes \boldsymbol{I}_n)(\underline{\boldsymbol{V}}_0 - \boldsymbol{V}^c) \tag{40}$$

where $\underline{\boldsymbol{V}}_0 = \dot{\underline{\boldsymbol{\xi}}}_0 \in \mathbb{R}^{Nn}$, then the error dynamics are derived as

$$\begin{aligned}\dot{\boldsymbol{e}}_p{}^c &= \boldsymbol{e}_v{}^c,\\ m\dot{\boldsymbol{e}}_v{}^c &= ((\mathscr{L} + \boldsymbol{G}) \otimes \boldsymbol{I}_n)\left(m\dot{\underline{\boldsymbol{V}}}_0 - m\dot{\boldsymbol{V}}^c\right).\end{aligned} \tag{41}$$

Substituting (37) in (41) results in

$$m\dot{\boldsymbol{e}}_v{}^c = ((\mathscr{L} + \boldsymbol{G}) \otimes \boldsymbol{I}_n)\left(m\dot{\underline{\boldsymbol{V}}}_0 - (\boldsymbol{F}^c + \boldsymbol{1}_N \otimes \boldsymbol{F}_g)\right). \tag{42}$$

Now, we set the global desired force vector that contains desired force information for each agent of the formation (36) by using mixed-product property of Kronecker product, $(\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{C} \otimes \boldsymbol{D}) = (\boldsymbol{AC}) \otimes (\boldsymbol{BD})$ such that

$$\begin{aligned}\boldsymbol{F}^c = m\dot{\underline{\boldsymbol{V}}}_0 &- \boldsymbol{1}_N \otimes \boldsymbol{F}_g + m(\boldsymbol{I}_N \otimes (\boldsymbol{K}_1 + \boldsymbol{K}_2))\boldsymbol{e}_v{}^c\\ &+ m(\boldsymbol{I}_N \otimes (\boldsymbol{I}_{3\times3} + \boldsymbol{K}_1\boldsymbol{K}_2))\boldsymbol{e}_p{}^c.\end{aligned} \tag{43}$$

Note that $\boldsymbol{F}^c$ is the global form of (30). Moreover, $\underline{\boldsymbol{\xi}}_0 + \underline{\boldsymbol{\Delta}}$ and $\underline{\boldsymbol{V}}_0$ are the global form of $\boldsymbol{\xi}_d$ and $\boldsymbol{V}_d$, respectively. Then, we end up with the following second-order error dynamics such that:

$$\begin{aligned}\dot{\boldsymbol{e}}_p{}^c &= \boldsymbol{e}_v{}^c\\ \dot{\boldsymbol{e}}_v{}^c &= -((\mathscr{L} + \boldsymbol{G}) \otimes (\boldsymbol{K}_1\boldsymbol{K}_2 + \boldsymbol{I}_{3\times3}))\boldsymbol{e}_p{}^c\\ &\quad -((\mathscr{L} + \boldsymbol{G}) \otimes (\boldsymbol{K}_1 + \boldsymbol{K}_2))\boldsymbol{e}_v{}^c.\end{aligned} \tag{44}$$

The state-space form of (44) is

$$\dot{\boldsymbol{x}}^c(t) = \boldsymbol{J}^c\boldsymbol{x}^c(t) \tag{45}$$

where $\boldsymbol{J}^c \in \mathbb{R}^{2Nn\times2Nn}$ is the global system matrix such that

$$\boldsymbol{J}^c = \begin{bmatrix}\boldsymbol{0}_{Nn\times Nn} & \boldsymbol{I}_{Nn\times Nn}\\ -((\mathscr{L} + \boldsymbol{G}) \otimes (\boldsymbol{K}_1\boldsymbol{K}_2 + \boldsymbol{I}_{3\times3})) & -((\mathscr{L} + \boldsymbol{G}) \otimes (\boldsymbol{K}_1 + \boldsymbol{K}_2))\end{bmatrix}$$

(46)

and $\boldsymbol{x^c}(t) = \left[\boldsymbol{e_p}^{\boldsymbol{c}T} \quad \boldsymbol{e_v}^{\boldsymbol{c}T}\right]^T \in \mathbb{R}^{2Nn \times 1}$ is the global state vector. The global dynamics (45) are the combination of the single-agent dynamics (33) for the entire formation. Before we do the stability analysis for the closed-loop error dynamics given in (45), we make following assumption.

Assumption 2: The graph topology of the multi-agent system contains a spanning tree with the root node being the leader node. This means that there is a directed path (not necessarily unique) from the leader node to every follower node.

The next theorem extends the single-agent result in Theorem 1 to the multi-agent case by using M-matrix properties of the digraphs [28].

Theorem 2: Given the Assumption 2, $\mathscr{L} + \boldsymbol{G}$ is an irreducible M-matrix and has all eigenvalues strictly in the open right-half plane [28]. Then, the equilibrium of closed-loop error dynamics given in (45) is globally asymptotically stable point meaning that $\boldsymbol{J^c}$ is Hurwitz.

Proof: Use the fact that Kronecker product of a positive diagonal matrix and an M-matrix has all eigenvalues strictly in the open right-half plane [29, 30]. Then, re-write (45) in the form of the second-order differential equation such that

$$\begin{aligned} \ddot{\boldsymbol{e}}_{\boldsymbol{p}}{}^{\boldsymbol{c}} &+ ((\mathscr{L} + \boldsymbol{G}) \otimes (\boldsymbol{K}_1 + \boldsymbol{K}_2))\dot{\boldsymbol{e}}_{\boldsymbol{p}}{}^{\boldsymbol{c}} \\ &+ ((\mathscr{L} + \boldsymbol{G}) \otimes (\boldsymbol{K}_1\boldsymbol{K}_2 + \boldsymbol{I}_{3 \times 3}))\boldsymbol{e_p}{}^{\boldsymbol{c}} = \boldsymbol{0}\,. \end{aligned} \tag{47}$$

Notice that all coefficient matrices of the characteristic polynomial of (47) have eigenvalues at open right half plane, hence the origin is globally asymptotically stable equilibrium by Routh-Hurwitz test. Note that if the graph topology was undirected, algorithms proposed in this paper would still work because $\mathscr{L} + \boldsymbol{G}$ would be positive definite symmetric matrix with the assumption of there exists a path from the leader node to every follower node. □

4.2 Communication delays

In this section, first we consider the system with constant communication delay, which occurs while the local positioning system shares the position of each agent to their neighbours. This delay may be created by processing time of the positioning system, data header analysis, storage at routers, and so on. Note that this delay is upper bounded by the practical limitations. Then (45) is written in the form of RFDE [20] such that

$$\dot{\boldsymbol{x}}^{\boldsymbol{c}}(t) = \boldsymbol{J^c}\boldsymbol{x^c}(t) + \beta\boldsymbol{J^c}\boldsymbol{x^c}(t - \gamma) \tag{48}$$

where γ is the network delay, β is the gain of delayed term, and $\boldsymbol{J^c}$ is the system matrix (46).

Theorem 3: For the system in (48), the origin is stable equilibrium for $\beta \in (-1, 1]$ as the system matrix $\boldsymbol{J^c}$ is Hurwitz by Theorem 2.

Proof: As $\gamma \to 0$, β must be greater than -1 so that the overall system is stable, which is the lower bound of β.

To find the upper bound, use the fact that $\rho\left((jw\boldsymbol{I} - \boldsymbol{J^c})^{-1}\boldsymbol{J^c}\right) < 1\, \forall w > 0$ as given in [31] where $\rho(\,.\,)$ denotes the spectral radius of a matrix and w denotes the frequency.

First assume that $\rho\left((jw_t\boldsymbol{I} - \boldsymbol{J^c})^{-1}\boldsymbol{J^c}\right) = 1,\, \forall w_t > 0$, which implies $e^{j\sigma_t}$ is the eigenvalue of the matrix $(jw_t\boldsymbol{I} - \boldsymbol{J^c})^{-1}\boldsymbol{J^c}$ for $\sigma_t \in [0, 2\pi]$. Then, $\det\left(\boldsymbol{I} - (jw_t\boldsymbol{I} - \boldsymbol{J^c})^{-1}\boldsymbol{J^c}e^{j\gamma_t w_t}\right) = 0$ for $\gamma_t = \frac{\sigma_t}{w_t}$, or equivalently by using matrix determinant lemma

$$\det\left(jw_t\boldsymbol{I} - \boldsymbol{J^c} - \boldsymbol{J^c}e^{j\gamma_t w_t}\right) = 0. \tag{49}$$

Hence, (48) is not stable independent of delay with this assumption.

Next, assume that $\rho\left((jw_t\boldsymbol{I} - \boldsymbol{J^c})^{-1}\boldsymbol{J^c}\right) > 1,\, \forall w_t > 0$. Since $\rho(\,.\,)$ is continuous function of w_t and

$$\lim_{w_t \to \infty} \rho\left((jw_t\boldsymbol{I} - \boldsymbol{J^c})^{-1}\boldsymbol{J^c}\right) = 0. \tag{50}$$

Then $\exists w_t \in (w, \infty)$ such that $\rho\left((jw_t\boldsymbol{I} - \boldsymbol{J^c})^{-1}\boldsymbol{J^c}\right) = 1$, which makes (48) not stable independent of delay as this ends up with (49). Consequently, we show that $\rho\left((jw_t\boldsymbol{I} - \boldsymbol{J^c})^{-1}\boldsymbol{J^c}\right) < 1,\ \forall w_t > 0$. By using Gelfand Corollaries, this results in

$$\begin{aligned} \rho\left((jw_t\boldsymbol{I} - \boldsymbol{J^c})^{-1}\boldsymbol{J^c}\right) &\leq \rho\left((jw_t\boldsymbol{I} - \boldsymbol{J^c})^{-1})\rho(\boldsymbol{J^c})\right) \\ &\leq \frac{\|\boldsymbol{J^c}\|_\infty}{\sqrt{\|\boldsymbol{J^c}\|_\infty^2 + w_t^2}}\,. \end{aligned} \tag{51}$$

Then the upper bound of β must be 1 by (51) to have $\rho\left(\beta(jw_t\boldsymbol{I} - \boldsymbol{J^c})^{-1}\boldsymbol{J^c}\right) < 1$. To this end, we proved that as $\beta \in (-1, 1]$, the system represented in (48) is stable independent of delay. □

Now, if we consider the system with non-constant distributed delays, (45) can be written as

$$\dot{\boldsymbol{x}}^{\boldsymbol{c}}(t) = \boldsymbol{J^c}\boldsymbol{x^c}(t) + \beta\int_{-\gamma}^{0}\boldsymbol{J^c}\boldsymbol{x^c}(t + s)\,\mathrm{d}s \tag{52}$$

where $s \in [-\gamma, 0]$, β is the gain of delayed term and γ is the maximum delay.

Theorem 4: The origin is an asymptotically stable equilibrium of (52) when there exist distributed delays in the communication network with gain $\beta \in [0, 1]$, which is a sufficient condition for asymptotic stability.

Proof: As shown in Theorem 3, the origin is stable equilibrium for $\beta \in (-1, 1]$ since $\boldsymbol{J^c}$ is proven to be Hurwitz in Theorem 2. With this in mind, pick Lyapunov–Krasovskii functional as

$$\begin{aligned} \boldsymbol{V}(\boldsymbol{x}_t^{\boldsymbol{c}}) &= \boldsymbol{x}^{\boldsymbol{c}\boldsymbol{T}}(t)\boldsymbol{P}\boldsymbol{x^c}(t) \\ &+ \beta\int_{t-\gamma}^{t}\left[\int_{s}^{0}\boldsymbol{x}^{\boldsymbol{c}\boldsymbol{T}}(\ell)\boldsymbol{S}\boldsymbol{x^c}(\ell)\,\mathrm{d}\ell\right]\mathrm{d}s \end{aligned} \tag{53}$$

where $\boldsymbol{P} \in \mathbb{R}^{2Nn \times 2Nn}$ and $\boldsymbol{S} \in \mathbb{R}^{2Nn \times 2Nn}$ are positive definite, symmetric matrices. To have strictly positive definite Lyapunov–Krasovskii functional (53), the sufficient condition is $\beta > 0$. Before taking the derivative of Lyapunov–Krasovskii functional and developing stability analysis, we use change of variable $\boldsymbol{f}(T) = \boldsymbol{x^c}(t + T)$ for arbitrary T, to simplify the stability analysis. Then, (52) and (53) become

$$\dot{\boldsymbol{f}}(0) = \boldsymbol{J^c}\boldsymbol{f}(0) + \beta\int_{-\gamma}^{0}\boldsymbol{J^c}\boldsymbol{f}(s)\,\mathrm{d}s \tag{54}$$

$$\begin{aligned} \boldsymbol{V}(\boldsymbol{f}) &= \boldsymbol{f}^{\boldsymbol{T}}(0)\boldsymbol{P}\boldsymbol{f}(0) \\ &+ \beta\int_{-\gamma}^{0}\left[\int_{s}^{0}\boldsymbol{f}^{\boldsymbol{T}}(\ell)\boldsymbol{S}\boldsymbol{f}(\ell)d\ell\right]ds\,. \end{aligned} \tag{55}$$

Furthermore, by using the Leibniz Integral Rule and (54), the derivative of Lyapunov–Krasovskii functional (55) becomes

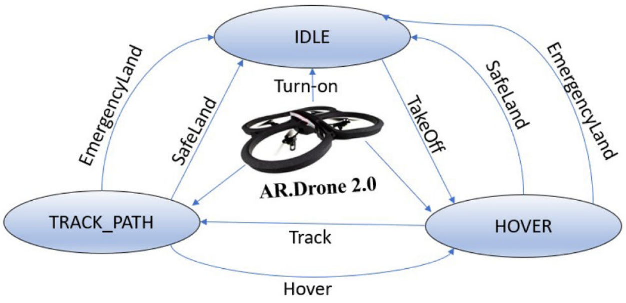

Fig. 3 *Flight controller FSM design*

$$\begin{aligned}\dot{V}(\boldsymbol{f}) &= \boldsymbol{f}^{\boldsymbol{T}}(0)\left[\boldsymbol{P}\boldsymbol{J}^c + \boldsymbol{J}^{c\boldsymbol{T}}\boldsymbol{P} + \gamma\beta\boldsymbol{S}\right]\boldsymbol{f}(0)\\ &+2\boldsymbol{f}^{\boldsymbol{T}}(0)\int_{-\gamma}^{0}\boldsymbol{P}\boldsymbol{J}^c\boldsymbol{f}(s)\,\mathrm{d}s\\ &-\beta\int_{-\gamma}^{0}\boldsymbol{f}^{\boldsymbol{T}}(s)\boldsymbol{S}\boldsymbol{f}(s)\,\mathrm{d}s\,.\end{aligned} \tag{56}$$

To facilitate further development, (56) is written as

$$\begin{aligned}\dot{V}(\boldsymbol{f}) &= \boldsymbol{f}^{\boldsymbol{T}}(0)\left[\boldsymbol{P}\boldsymbol{J}^c + \boldsymbol{J}^{c\boldsymbol{T}}\boldsymbol{P}\right]\boldsymbol{f}(0)\\ &+\int_{-\gamma}^{0}\boldsymbol{v}^{\boldsymbol{T}}\begin{bmatrix}\beta\boldsymbol{S} & \boldsymbol{P}\boldsymbol{J}^c\\ \boldsymbol{J}^{c\boldsymbol{T}}\boldsymbol{P} & -\beta\boldsymbol{S}\end{bmatrix}\boldsymbol{v}\,\mathrm{d}s\end{aligned} \tag{57}$$

where $\boldsymbol{v} = \left[\boldsymbol{f}(0)^{\boldsymbol{T}} \quad \boldsymbol{f}(s)^{\boldsymbol{T}}\right]^{\boldsymbol{T}}$. Notice that $V(\boldsymbol{f}) \geq \varepsilon\|\,\boldsymbol{f}(0)\,\|^2$ is satisfied for sufficiently small $\varepsilon > 0$. And, $\left[\boldsymbol{P}\boldsymbol{J}^c + \boldsymbol{J}^{c\boldsymbol{T}}\boldsymbol{P}\right] \leq -\varepsilon\boldsymbol{I}$ since $\boldsymbol{J}^c$ is proved to be Hurwitz by Theorem 2. In addition, assuming $\exists\boldsymbol{P} = \boldsymbol{P}^{\boldsymbol{T}} > \boldsymbol{0}$ and using the linear matrix inequality [13], the negative definiteness of a matrix

$$\begin{bmatrix}\beta\boldsymbol{S} & \boldsymbol{P}\boldsymbol{J}^c\\ \boldsymbol{J}^{c\boldsymbol{T}}\boldsymbol{P} & -\beta\boldsymbol{S}\end{bmatrix}$$

implies $\dot{V}(\boldsymbol{f}) \leq \varepsilon\|\,\boldsymbol{f}(0)\,\|^2$. Therefore, all conditions of the asymptotically stability by analysing the derivative of Lyapunov–Krasovskii functional given in [23], have met meaning that the origin is asymptotically stable equilibrium. □

5 Experiment design and flight test details

This section addresses the crucial elements of our experiments, which are lab environment, flight controller design and simulations. An actual flight test is conducted in Section 6.

5.1 Lab environment

Equipment used are the Vicon, Parrot AR.Drone 2.0, and the master computer. Vicon is a motion capture system that provides the position of the UAVs.

The communication between master computer and Vicon is done via User Datagram Protocol (UDP). The frequency of the UDP Packets taken from the Vicon motion capture system is 100 Hz. The AR.Drone 2.0 has a built-in gyroscope and Inertial Measurement Unit (IMU) sensor suite. In practical applications, many quadrotors are designed with a built-in attitude controller and AR.Drone has its own attitude controller. This controller takes the desired values of φ_d, θ_d and ψ_d as inputs. The communication between the master computer and the AR.Drone is done via UDP. The frequency of UDP packages is set to 500 Hz. MATLAB-Simulink is used to create UDP nodes that are communicating with AR.Drone and Vicon. The receiver and the sender UDP nodes are inserted to the Simulink model in the form of S-functions. The controller and the trajectory generation algorithms are implemented in the model. Simulink-Desktop Real Time Add-on is used to send the real time commands to the quadrotor. The UDP nodes tolerate up to 10% packet loss rate, which is necessary to handle communication channel noise created by the lab environment.

Table 1 Inner attitude control loop PID parameters

Gains	Φ(Roll)	θ(Pitch)	ψ(Yaw)
$\boldsymbol{P}$	6.42	6.42	4.82
$\boldsymbol{D}$	5.54	5.54	7.89
$\boldsymbol{I}$	1.85	1.85	0.11

5.2 Flight controller design

The flight controller is a high-level decision-making mechanism that activates different modes of operation depending on the state of the UAV. We recognise three modes of operation in our MATLAB implementation, which are IDLE, HOVER, and TRACK_PATH as shown in Fig. 3. The quadrotor enters the IDLE mode, when either we turn-on the AR.Drone manually or it is landed by receiving the Land command. While in the IDLE, the UAV is actively receiving the data packets via UDP port communication and is ready to get the TakeOff command.

When the UAV reaches the desired height, z_d, the flight controller switches to the HOVER mode. In this mode, the UAV is at the nominal condition, its attitude is parallel to ground and motionless in the air. If the Track command is received in the HOVER mode, the flight controller switches to the PATH_TRACK mode. This mode is triggered after 10 s passed from the transmission of TakeOff command. In this mode, AR.Drone begins to track the predetermined trajectory. The flight control algorithm first reads the IMU sensor buffers and then Vicon buffer to construct the close loop error dynamics. To calculate desired Euler angles, (10)–(12) are used. Notice herein the desired yaw angle command is set to an arbitrary constant.

To land the quadrotor on the ground, we use either EmergencyLand or SafeLand commands. The difference of these two commands is the timing of stopping propeller movements. If we send the SafeLand command to the quadrotor, it reduces the propellers' speed till the height is in the range of 0–0.1 m and shuts down the propellers. Else if we send the EmergencyLand to the quadrotor, it directly stops the propellers and lands on the ground. The appropriate structure for implementing the flight controller is the finite state machine (FSM) since the mode switching event is driven as shown in Fig. 3.

5.3 Simulations

The aim of this section is to verify control algorithms proposed in this paper by conducting different test scenarios. Before we implement the distributed backstepping control algorithm in the actual hardware, mathematical model in Section 2.1, backstepping control method in Section 3, and distributed backstepping trackers in Section 4 are implemented in the Simulink. We first verified inner attitude control loop design in Section 3.2, the PID gains are given in Table 1.

To derive PID gains given in Table 1, thrust, drag factors, mass and arm length of the AR.Drone 2.0 must be measured. Mass and arm length of the quadrotor are measured as $m = 0.467\,\mathrm{kg}$ and $l = 0.1785\,\mathrm{m}$, respectively. To determine the thrust factor k, we first measure the angular velocity of a propeller with tachometer when the quadrotor is in hover. Then, we used (17) to derive k, which is found as $8 * 10^{-6}\,\mathrm{N} * \mathrm{s}^2/\mathrm{rad}^2$. After that, using the thrust ratio analysis for small UAVs, the thrust factor d is derived as $2 * 10^{-7}\mathrm{N} * \mathrm{m} * \mathrm{s}^2/\mathrm{rad}^2$. Moreover, the diagonal elements of $\boldsymbol{K}_1$, $\boldsymbol{K}_2$ are tuned as 2, 2, 3 and 1.5, 1.5, 3, respectively.

To construct the desired circular trajectory for the formation leader, x_d is set to $cos(\omega_t(t - t_{\text{track}}))$ and y_d is set to $\sin(\omega_t(t - t_{\text{track}}))$ where t is simulation time, t_{track} is the time at which the UAV begins to track circular trajectory, and ω_t stands for the frequency of the sinusoidal function. In our simulations, we pick t_{track} as 15 s and ω_t as 0.5 rad/s. Note that before formation leader begins to track circular trajectory, the x_d value is linearly increased by 1 m

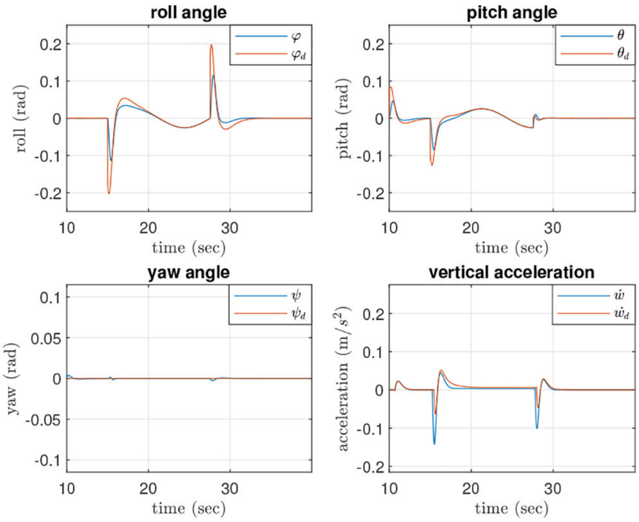

Fig. 4 *Attitude control inputs of leader UAV*

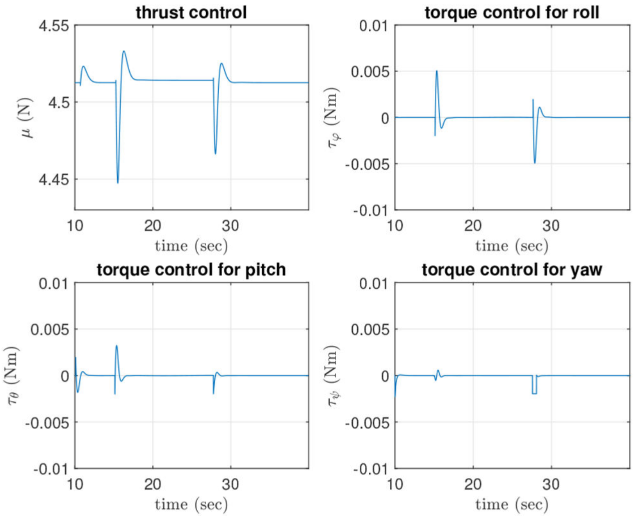

Fig. 5 *Torque and thrust controls of the leader UAV*

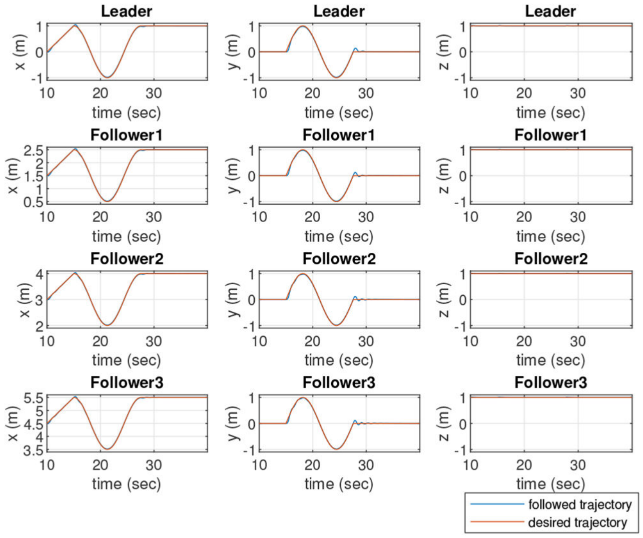

Fig. 6 *Path tracked by the UAVs when there exists no delay and undirected graph topology is used*

for $t \in [10, 15]$. Therefore, the leader UAV begins to track circular trajectory at 15 s. In Section 6.1, we double value of ω_t for y_d setting to construct the eight figure trajectory.

5.3.1 No delay, undirected graph: The aim of this section is to show validity of the proposed algorithms, when the undirected graph topology is adopted to design leader–follower formation control while multi-UAV system does not experience any delays. We share the control histories for the formation leader in Figs. 4 and 5. Particularly, in Fig. 4, we show attitude control inputs (10)–(13) of the leader UAV. In Fig. 5, we show torque (19), and thrust (20) controls defined in the body frame of the formation leader. For this test scenario, we pick adjacency, pinning gain matrices and offset vector as follows:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad (58)$$

$$\underline{\boldsymbol{\Delta}}^T = [1.5 \quad 0 \quad 0 \quad 3 \quad 0 \quad 0 \quad 4.5 \quad 0 \quad 0].$$

Fig. 6 shows the leader and follower positions with the adjacency and pining gain matrices given in (58), when there is no communication delay in the multi-UAV communication network.

5.3.2 No delay, directed graph: For this test scenario, along with the offset vector given in (58), we pick adjacency and pinning gain matrices that do no contradict Assumption 2, as follows:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \qquad (59)$$

Note that Fig. 7 is the same as Fig. 6 since the graph topology with the adjacency and pinning gain matrices given in (59) contains a spanning tree.

5.3.3 With delay, directed graph: For the last test scenario of simulations, along with the offset vector given in (58), we pick adjacency and pinning gain matrices given in (59). And, to test the stable independent of delay structure of the algorithms presented in Section 4, we added 2 s delay as a communication delay. By looking at Fig. 8, one can conclude that stable independent of delay property of the proposed algorithms is verified.

6 Actual flight test results

This section reveals the flight test results obtained with single and multiple UAVs under the influence of time delays. We share the graphs of the desired and followed trajectories when both the proposed methods in Section 3.3 and 4 are used.

6.1 Controller behaviour with a single quadrotor

In this section, we present the performance of the backstepping control algorithm proposed in Section 3 by using both circular and figure-eight trajectories.

For the backstepping controller designed in Section 3.3, Figs. 9 and 10 show the desired path of the UAV and the path traced by the UAV when both circular and eight-figure are desired trajectories. Notice that the tracking error is maintained inside the acceptable bounds, showing the performance of designed backstepping controller in terms of path following. Note that diagonal elements of $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$, are assigned, respectively, as 2, 2, 3 and 1.5, 1.5, 3.

To observe the network communication delay, we plot time versus desired and tracked positions in 3D space as shown in Fig. 11. Notice that communication delay is about 2 s and tends to be commensurate through path following experiment. Moreover, when the quadrotor is following a non-linear trajectory such as eight-figure and circular path, there exists time-delay between the desired position and followed position as shown in Fig. 11. This time delay is the summation of reaction time of UAV and the network delay caused by the local positioning system. That is why, the error seems to be bounded. However, after finishing the complete eight or circular figure trajectory, the error goes to zero since the desired position is constant at that time and quadrotor's

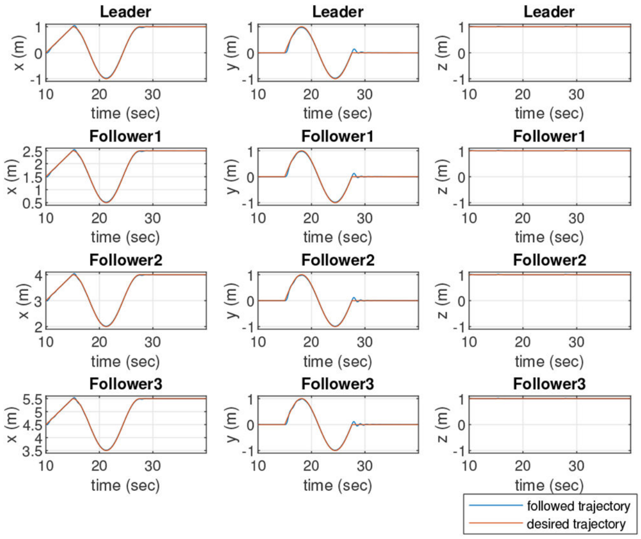

Fig. 7 *Path tracked by the UAVs when there exists no delay and directed graph topology is used*

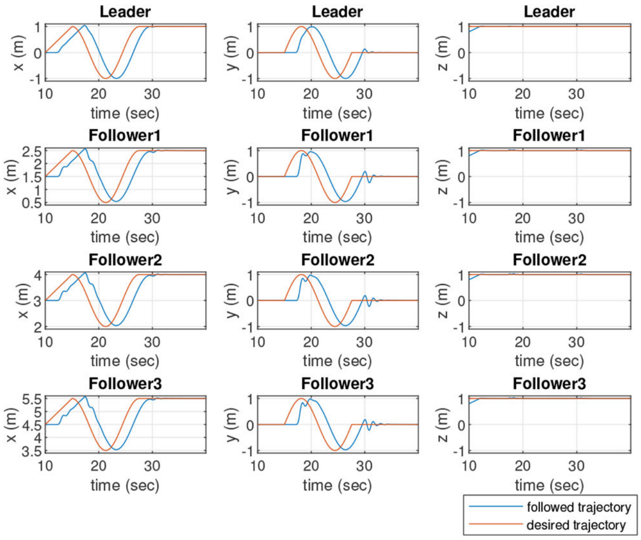

Fig. 8 *Path tracked by the UAVs when there exists 2 s delay and directed graph topology is used*

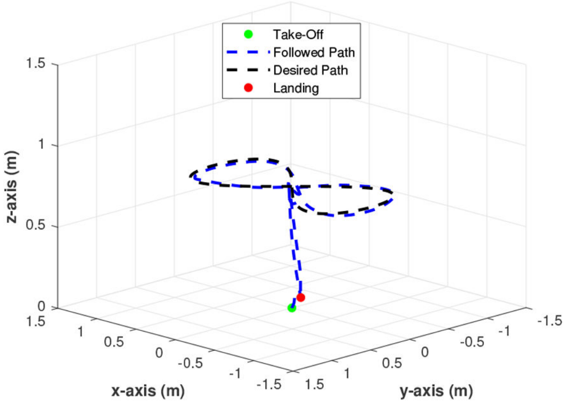

Fig. 9 *Path tracked by the UAV with the backstepping controller using eight-figure trajectory*

position is the same as the desired position. This can be seen clearly from Figs. 9 and 10.

6.2 Controller behaviour with multiple quadrotors

This section shows the formation control performance of the distributed backstepping control method given in Section 4. In the experiments of this section, the task of followers is to track the formation leader with a certain position offset. We show the leader and followers positions in the Figs. 12 and 13 while the leader of the multi-agent system is following both eight-figure and circular trajectories, respectively. Adjacency and pinning gain matrices along with offset vector used in this actual hardware implementation are

$$\begin{gathered}\boldsymbol{A} = \begin{bmatrix}0 & 0\\ 1 & 0\end{bmatrix}, \boldsymbol{G} = \begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix},\\ \underline{\boldsymbol{\Delta}}^T = [0 \quad -3 \quad 0 \quad 0 \quad -6 \quad 0].\end{gathered} \tag{60}$$

Figs. 14 and 15 show the desired path of the UAVs and the path tracked by the UAVs when the leader is following both circular and eight-figure type trajectories. Note that the delay experienced by each agent of the formation is slightly different than each other. However, if the position offset of the quadrotors gets bigger, agents of the formation would experience more distributed delays.

We record a video of the experiments described in this section, the interested reader can use the link ‘https://www.youtube.com/watch?v=rmY1LK42oPk’ to have a visual understanding of the paper. Notice in the movie, formation control using the distributed backstepping method is influenced by the very strong wind effect that is produced by quadrotors themselves. This demonstrates the robustness of the proposed control method.

7 Conclusion

In this paper, a method of distributed backstepping method to have formation flight of multiple quadrotors with distributed time delays is discussed. The proposed algorithms are validated by using Vicon Tracker, AR.Drone 2.0 and a master computer. Through rigorous experimentation and stability analysis, we showed that distributed backstepping control method, provides a guaranteed performance for follower agents to track the leader agent with a predetermined position offset. We give the trajectories followed by the single quadrotor under the influence of commensurate delay and by the multiple quadrotors under the influence of distributed delays. Further research can be conducted to investigate how aerodynamic forces that interferes with the fuselage affect the algorithms suggested in this paper.

8 Acknowledgments

This work is supported by ONR Grant N00014-17-1-2239, ONR Grant N00014-18-1-2221, NSF Grant ECCS-1839804. The author Mr. Kartal thanks Turkish Aerospace for the scholarship granted.

Q2

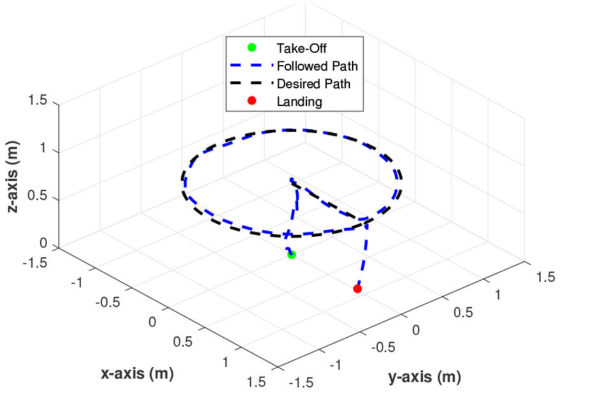

Fig. 10 *Path tracked by the UAV with the backstepping controller using a circular trajectory*

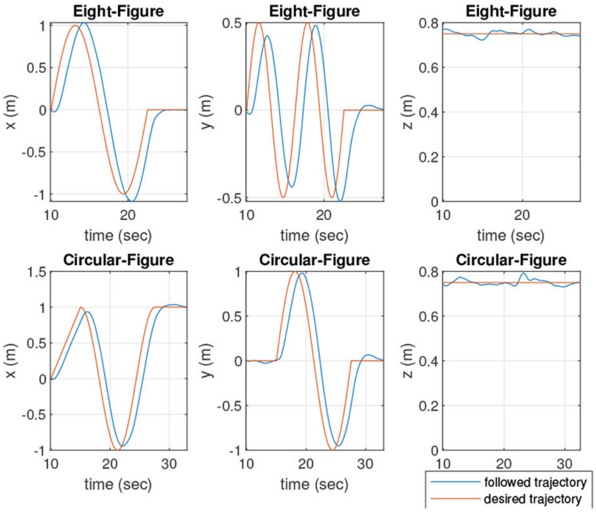

Fig. 11 *Controller behaviour with time delay*

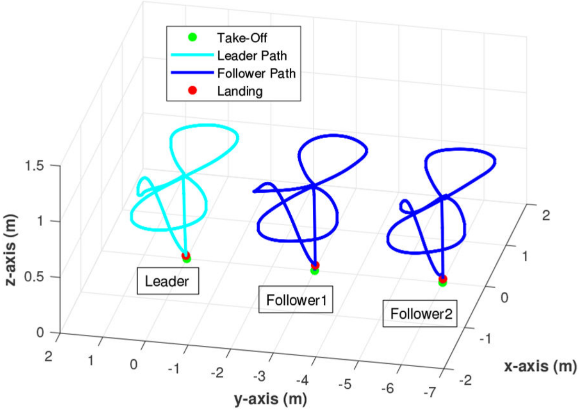

Fig. 12 *Path tracked by the UAVs with the distributed backstepping controller when the leader follows an eight-figure trajectory*

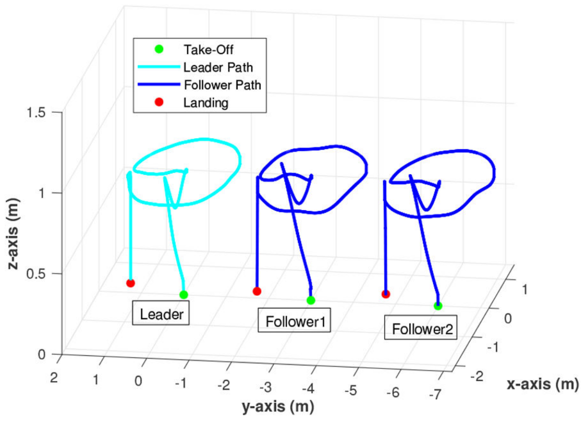

Fig. 13 *Path tracked by the UAVs with the distributed backstepping controller when the leader follows a circular trajectory*

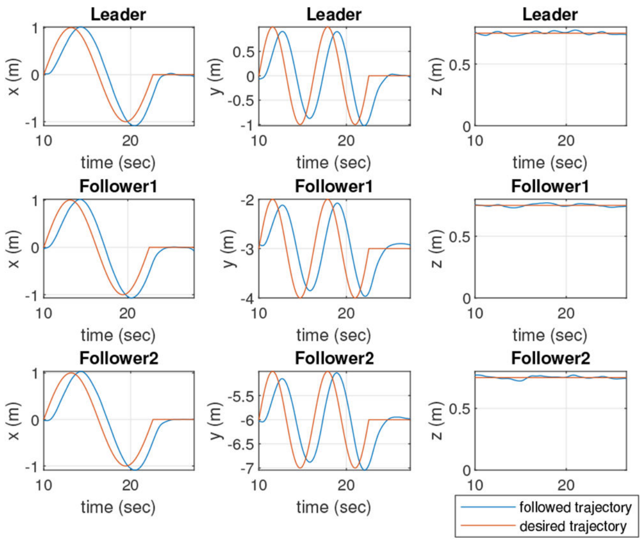

Fig. 14 *Observation of time delay graph when the leader follows an eight-figure trajectory*

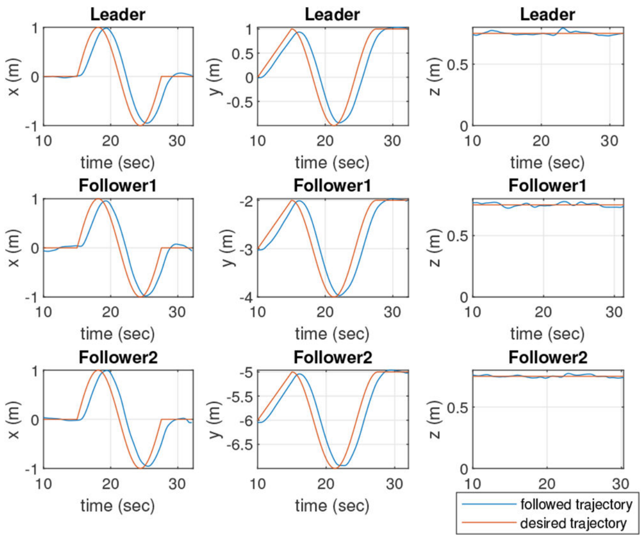

Fig. 15 *Observation of time delay graph when the leader follows a circular trajectory*

9 References

[1] Parrish, J.K., Viscido, S.V., Grunbaum, D.: 'Self-organized fish schools: an examination of emergent properties', *Biol. Bull.*, 2002, **202**, (3), pp. 296–305

[2] Mellinger, D., Shomin, M., Michael, N., *et al.*: 'Cooperative grasping and transport using multiple quadrotors'. Distributed Autonomous Robotic Systems 2013, pp. 545–558

Q3 [3] Kopfstedt, T., Mukai, M., Fujita, M., *et al.*: 'Control of formations of UAVs for surveillance and reconnaissance missions', *IFAC Proceedings Volumes*, 2008, **41**, (2), pp. 5161–5166

[4] Waharte, S., Trigoni, N., Julier, S.: 'Coordinated search with a swarm of UAVs'. IEEE Annual Communications Society Conf. on Sensor, Mesh and ad hoc Communications and Networks Workshops, June 2009, pp. 1–3

Q4 [5] Maza, I., Ollero, A., Casado, E., *et al.*: '*Classification of multi-UAV architectures' Handbook of unmanned aerial vehicles*' (2015), pp. 953–975

Q5 [6] Lee, G., Chwa, D.: 'Decentralized behavior-based formation control of multiple robots considering obstacle avoidance', *Intelligent Service Robotics*, 2018, **11**, (1), pp. 127–138

[7] Zhou, D., Wang, Z., Schwager, M.: 'Agile coordination and assistive collision avoidance for quadrotor swarms using virtual structures', *IEEE Trans. Robot.*, 2018, **34**, (4), pp. 916–923

[8] Duan, H., Qiu, H.: 'Unmanned aerial vehicle distributed formation rotation control inspired by leader-follower reciprocation of migrant birds', *IEEE Access*, 2018, **6**, pp. 23431–23443

[9] Xi, J., Cai, N., Zhong, Y.: 'Consensus problems for high-order linear time-invariant swarm systems', *Phys. A, Stat. Mech. Appl.*, 2010, **389**, (24), pp. 5619–5627

[10] Dong, X., Zhou, Y., Ren, Z., *et al.*: 'Time-varying formation control for unmanned aerial vehicles with switching interaction topologies', *Control Eng. Pract.*, 2016, **46**, pp. 26–36

[11] Carvalho, J.F., Pequito, S., Aguiar, A.P., *et al.*: 'Composability and controllability of structural linear time-invariant systems: distributed verification', *Automatica*, 2017, **78**, pp. 123–134

[12] Ma, C.Q., Zhang, J.F.: 'Necessary and sufficient conditions for consensusability of linear multi-agent systems', *IEEE Trans. Autom. Control*, 2010, **55**, (5), pp. 1263–1268

[13] Rui, W., Xiwang, D., Qingdong, L., *et al.*: 'Adaptive time-varying formation control for high-order LTI multi-agent systems'. 2015 34th Chinese Control Conf. (CCC), July 2015, pp. 6998–7003

[14] Baghbani, F., Akbarzadeh-T, M.R., Sistani, M.B.N.: 'Cooperative adaptive fuzzy tracking control for a class of nonlinear multi-agent systems'. 2017 Joint 17th World Congress of Int. Fuzzy Systems Association and 9th Int. Conf. on Soft Computing and Intelligent Systems (IFSA-SCIS), June 2017, pp. 1–6

[15] Wen, G.X., Chen, C.P., Liu, Y.J., *et al.*: 'Neural-network-based adaptive leader-following consensus control for second-order non-linear multi-agent systems', *IET Control Theory Applic.*, 2015, **9**, (13), pp. 1927–1934

[16] He, W., Chen, G., Han, Q.L., *et al.*: 'Network-based leader-following consensus of nonlinear multi-agent systems via distributed impulsive control', *Inf. Sci.*, 2017, **380**, pp. 145–158

[17] Sun, J., Geng, Z.: 'Adaptive consensus tracking for linear multi-agent systems with heterogeneous unknown nonlinear dynamics', *Int. J. Robust Nonlinear Control*, 2016, **26**, (1), pp. 154–173

[18] Defoort, M., Polyakov, A., Demesure, G., *et al.*: 'Leader-follower fixed-time consensus for multi-agent systems with unknown non-linear inherent dynamics', *IET Control Theory Applic.*, 2015, **9**, (14), pp. 2165–2170

[19] Du, H., Cheng, Y., He, Y., *et al.*: 'Second-order consensus for nonlinear leader-following multi-agent systems via dynamic output feedback control', *Int. J. Robust Nonlinear Control*, 2016, **26**, (2), pp. 329–344

[20] Daly, J.M., Ma, Y., Waslander, S.L.: 'Coordinated landing of a quadrotor on a skid-steered ground vehicle in the presence of time delays', *Auton. Robots*, 2015, **38**, (2), pp. 179–191

[21] Dong, X., Yu, B., Shi, Z., *et al.*: 'Time-varying formation control for unmanned aerial vehicles: theories and applications', *IEEE Trans. Control Syst. Technol.*, 2014, **23**, (1), pp. 340–348

[22] Abdessameud, A., Tayebi, A.: 'Formation control of VTOL unmanned aerial vehicles with communication delays', *Automatica*, 2011, **47**, (11), pp. 2383–2394

[23] Krstić, M., Kanellakopoulos, I., Kokotović, P.V.: 'Adaptive nonlinear control without overparametrization', *Syst. Control Lett.*, 1992, **19**, (3), pp. 177–185

[24] Skjetne, R., Fossen, T.I., Kokotović, P.V.: 'Adaptive maneuvering, with experiments, for a model ship in a marine control laboratory', *Automatica*, 2005, **41**, (2), pp. 289–298

[25] Kristiansen, R., Nicklasson, P.J.: 'Satellite attitude control by quaternion-based backstepping'. Proc. of the 2005 American Control Conf., June 2005, pp. 907–912

[26] Pavlichkov, S.S., Dashkovskiy, S.N., Pang, C.K.: 'Uniform stabilization of nonlinear systems with arbitrary switchings and dynamic uncertainties', *IEEE Trans. Autom. Control*, 2016, **62**, (5), pp. 2207–2222

[27] Li, Z., Zhao, J.: 'Co-design of controllers and a switching policy for nonstrict feedback switched nonlinear systems including first-order feedforward paths', *IEEE Trans. Autom. Control*, 2018, **64**, (4), pp. 1753–1760

[28] Lewis, F.L., Zhang, H., Hengster-Movric, K., *et al.*: '*Cooperative control of multi-agent systems: optimal and adaptive design approaches*' (Springer Science & Business Media, 2013)

Q6 [29] Brewer, J.: 'Kronecker products and matrix calculus in system theory', *IEEE Trans. Circuits Syst.*, 1978, **25**, (9), pp. 772–781

[30] Plemmons, R.J.: 'M-matrix characterizations. I—nonsingular M-matrices', *Linear Algebr. Appl.*, 1977, **18**, (2), pp. 175–188

[31] Gu, K., Chen, J., Kharitonov, V.L.: '*Stability of time-delay systems*' (Springer Science & Business Media, 2003)

IET-CTA20191151

Author Queries

Q Please make sure the supplied images are correct for both online (colour) and print (black and white). If changes are required please supply corrected source files along with any other corrections needed for the paper.

Q All equations too long to fit within a single column are automatically floated to the bottom of the page and a '(see equation below)' or '(see (equation number) and (equation number)' left in their place. This is done automatically by our new XML transform, so please specify in your corrections very clearly where they should be broken as your paper is edited by non-experts.

Q1 Please provide department name if available in affiliation 2.

Q2 As per journal style references are renumbered in the text and reference list. Please confirm.

Q3 Please provide conference location in Ref. [2].

Q4 Please provide conference location in Refs. [4, 13, 14, 25].

Q5 Please provide publisher name and location in Ref. [5].

Q6 Please provide the location of the publisher (country) in Refs. [28, 31].