Abstract
The control problem of a multi-copter swarm, mechanically coupled through a modular lattice structure of connecting rods, is considered in this article. The system’s structural elasticity is considered in deriving the system’s dynamics. The devised controller is robust against the induced flexibilities, while an inherent adaptation scheme allows for the control of asymmetrical configurations and the transportation of unknown payloads. Certain optimization metrics are introduced for solving the individual agent thrust allocation problem while achieving maximum system flight time, resulting in a platform-independent control implementation. Experimental studies are offered to illustrate the efficiency of the suggested controller under typical flight conditions, increased rod elasticities and payload transportation.

Keywords Multi-agent swarms · Adaptive controls · Cooperative aerial vehicles · Aerial transportation

1 Introduction
Aerial transportation of an arbitrary weight and shape cargo using several collaborating agents demands their interaction with the environment [1]. This interaction can be accomplished with recent sensors and the on board computational capabilities. Classical controllers used in the autopilots need to be enhanced for maintaining the formation characteristics of the participating aerial-systems [2]. The advocated Unmanned Aerial Vehicle (UAV) formation dynamics must account for the physical connection between the aerial members in maintaining a proper flight behavior [3].

Aerial manipulators have also been used for cooperative manipulation and transportation [4, 5] where the dynamic coupling between the individual subsystems must be accounted. In [6], methods for collaboratively manipulating objects with autonomous aerial agents are studied, while in [7] cargo transportation using aerial vehicles equipped with dexterous robotic arms is studied. Similarly, in the [8] project, aerial manipulators interacting with the environment were employed.

Cooperative aerial transportation via use of cables has been heavily researched, due to the added benefit of minimization of interaction forces [9, 10]. Concept designs of aerial vehicles collaboratively transporting objects via contact forces have also been studied in [11].

In [12] modules of independently powered connected rotors are considered; these rotors are attached directly on objects in [13] and an estimation scheme is provided for computing the rotors’ positions and the moment areas, while the controller uses a pseudo-inverse thrust concept in its structure. The controller requires the utilization of individual IMUs placed at each rotor as well as at the transported payload.

Single Degree-of-Freedom (DoF) connected UAVs appeared in [3] and simulation studies validated the overall concept. In [14, 15] magnets were used for the copter-
attachments followed by a single DoF joint for controlling the system’s yaw. The suggested scheme assumed copters connected in sequential manner while complete knowledge of individual altitudes are required. Similarly, in [16] copters encountered in close proximity were attached relying on estimation schemes for computing the moment areas, while neural network controllers were tested. Advanced modules of rotor-wheel combinations were considered in [17]; these were attached to the carried objects and their optimal attachment positions were computed.

Overactuated aerial vehicles were considered in [18] and the thrust allocation problem was examined using a quadcopter on a dual-axes gimbal configuration followed by singularity avoidance methods. Using multicopter-systems [19, 20] for carrying large payloads allows the application of coupled yaw and thrust commands. Replacing these systems with single-rotor ones is a viable approximation without generating independent yaw-commands.

The main contributions of this paper are: 1) the design using primitive components (rods, polygons, attachment mechanisms) for interconnecting in a mechanical manner copters for arbitrary-load transportation, 2) derivation of dynamics for the multi-copter configuration while handling structural flexibilities, 3) the design of an adaptive controller robust against these flexibilities and handling the unknown payload properties, and 4) an optimization mechanism for adjusting the thrust to each agent using a-priori defined criteria (i.e., maximizing flight time). Compared to its earlier shortened version [21], this article extends the presented theoretical and experimental results by considering the effects of flexibilities in the dynamics and controllers, while carrying a payload with unknown properties (mass, moments of inertia), as well as improved optimization metrics and experimental validations.

This paper is structured in the following manner. Section 2 provides the kinematic analysis of the rigid-structure of the modular copter-structure. Section 3 presents the dynamics of the modular-copter including the flexibility effects, while Section 4 refers to the adaptive controller design. Experimental studies are presented in Section 5 followed by Concluding remarks.

2 Rigid Multi-Copter Kinematic Analysis

The kinematics of a zero-carried payload \((m_p = 0)\) multi-copter structure is considered. Assume a planar\(^1\) modular lattice comprising of \(n\) copters, \(r\), \((r \geq n)\) connecting rods and \(p\) polygons, as shown in Fig. 1. Each side of the rod is connected to a copter or polygon and its other side to another polygon; the rods cannot be deformed and are assumed rigid in this section.

Using the standard notation [22], let \(\text{Rot}(a, b)\) be the \(4 \times 4\) homogeneous transformation matrix of a rotation about axis \(a\) by angle \(b\) and \(\text{Trans}(a, b)\) the homogeneous matrix of a simple translation along axis \(a\) by a distance \(b\). Similarly, let a coordinate system described by its origin \(O_i\) and a triad of mutually orthogonal basis vectors denoted as \((\hat{x}_i, \hat{y}_i, \hat{z}_i)\), or for brevity \([O_i, (\hat{x}_i, \hat{y}_i, \hat{z}_i)]\).

Let \([O_E, (\hat{x}_E, \hat{y}_E, \hat{z}_E)]\) be the Earth-fixed frame, \([O_i, (\hat{x}_i, \hat{y}_i, \hat{z}_i)]\) the rigid structure’s coordinate frame and \([O_i, (\hat{x}_i, \hat{y}_i, \hat{z}_i)]\) the individual copter’s coordinate system; the common attribute between the \(s\)-coordinate system and all \(i\)-coordinate systems, \(i = 0, \ldots, n - 1\) is that their \(z\)-axes are parallel \((\hat{z}_i \parallel \hat{z}_0, \ldots, \parallel \hat{z}_{n-1})\). The origin of the \(i\)-th coordinate system is located at the center of mass of the \(i\)-th copter, the \(\hat{x}_i\)-axis of the IMU is along the line that connects the polygon with the \(i\)-th copter while the \(\hat{z}_i\)-axis is perpendicular to the copters’ plane. For notation simplicity, we assume that the \(p_0\) polygon is connected to the 0th-copter.

Each \(p_i\) equilateral-polygon has \(p_i, f_i\) faces (i.e., \(p_0\) (\(p_1\)) is a hexagon (square) and has \(f_0 = 6\) (\(f_1 = 4\)) faces) with rods departing from its vertices having an inner rod angle of \(-\frac{2\pi}{f_i}\). Each copter or polygon is relatively placed with respect to its connecting polygon at a vector with length \(l_i\) and angle \(\frac{2\pi}{f_i};\) let \(l_i^j\) be the length of the rod between polygons \(p_i\) and \(p_j\), \(i, j \in {1, \ldots, p - 1}\).

Having defined in a systematic manner the mechanical interconnections between polygons and copters the kinematic description for all copters \(C = \{c_i\}, i = 0, \ldots, n - 1\), polygons \(P = \{p_j\}, j = 0, \ldots, p - 1\), types of polygons \(F = \{f_0, \ldots, f_{p-1}\}\) and rods \(L = \{l_k \cup l_q^k\}, k = 0, \ldots, n - 1, q \neq r \in {0, \ldots, p - 1}\) can be expressed with respect to the \([O_0, (\hat{x}_0, \hat{y}_0, \hat{z}_0)]\) coordinate system. Using this notation, the top-down version of Fig. 1 appears in Fig. 2.

\(^1\) The copters can be placed in parallel planes and the requirement for a planar configuration can thus be relaxed.
where the polygons ($p_0$-hexagon, $p_1$-square) are filled with yellow color and the rods are shown with black-solid vectors.

Then, as an example, from Fig. 2, $[O_1, (\hat{x}_1, \hat{y}_1, \hat{z}_1)] = [O_0, (\hat{x}_0, \hat{y}_0, \hat{z}_0)] \times \text{Trans}(x, l_0) \times \text{Rot}(z, 60^\circ) \times \text{Trans}(x, -l_1)$, and $[O_2, (\hat{x}_2, \hat{y}_2, \hat{z}_2)] = [O_0, (\hat{x}_0, \hat{y}_0, \hat{z}_0)] \times \text{Trans}(x, l_0) \times \text{Rot}(z, 120^\circ) \times \text{Trans}(x, -l_2)$.

The needed information to be stored for this kinematic analysis are the rods’ length array $\lambda$, the polygon list $P \times F$ and the interconnection list for rods $I$ corresponding to a matrix $I_{(p) \times (n+p)}$ with elements $I_{i,j}^* = \sum_{j=1}^{n+p} I_{i,j} = 1$, where $I_{i,j} = \begin{cases} 1 & \text{if } \exists \text{ rod between } p_i \text{ and } c/p_j \text{.} \\ 0 & \text{otherwise} \end{cases}$. Inhere, the enumeration of rods regarding the polygons follows that of the copters; for the multi-copter shown in Fig. 2, $I = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}_{2 \times 8}$.

Let the center $O_s$ of the multi-copter system $O_s \in \text{Co}(C, P)$ coincide with the center of mass of the entire copter/polygon/rod structure. The $\hat{x}_s$-axis points to the copter $c_i \in C$ that has the largest distance from $O_s$; in Fig. 2 $\parallel O_sO_0 \parallel = \max_{i \in C} \| O_sO_i \|$, since each $i$-th copter is located at $O_i$.

This uniform framework of assigning coordinate frames for the collective structure is necessary in order to simplify the overall development of the presented controllers. It should be noticed for the case of carried payload $m_p \neq 0$, $[O_s, (\hat{x}_s, \hat{y}_s, \hat{z}_s)]$ changes and the adaptive algorithm estimates recursively this coordinate system.

Having assigned $[O_s, (\hat{x}_s, \hat{y}_s, \hat{z}_s)]$, the pose of all copters with respect to it, can be computed. Assume a homogeneous matrix $sA_i$ describing the pose of the $i$-th copter with respect to $[O_s, (\hat{x}_s, \hat{y}_s, \hat{z}_s)]$

$$sA_i = \begin{bmatrix} \cos(\alpha_i) & -\sin(\alpha_i) & 0 & 0 & c_{i,x} \\ \sin(\alpha_i) & \cos(\alpha_i) & 0 & 0 & c_{i,y} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

where $\alpha_i$ is the angle between the copter and the $\hat{x}_s$-axis and $(c_{i,x}, c_{i,y})$ is the planar displacement from $O_s$. $R_c(\alpha_i) \in SO(3)$ denotes the elementary rotation matrix about $z$-axis by an angle $\alpha_i$, adopted from [22].

The procedure initially computes the copters’ pose with respect to $[O_0, (\hat{x}_0, \hat{y}_0, \hat{z}_0)]$ followed by a transformation to $[O_s, (\hat{x}_s, \hat{y}_s, \hat{z}_s)]$.

The displacement vectors of all copters are required to solve the control allocation problem, while the orientation angles $\alpha_i$ are used by the attitude controller and re-adjusting the IMU-readings of the coupled copters.

3 Multi-Copter Structure Dynamic Model

During the aforementioned kinematic analysis of the multi-copter system, it was assumed that the rods were rigid. If elongated rods are used, this gives rise to certain bending flexibilities. These flexibilities can influence the flight dynamics and counteract the benefits enjoyed by the adoption of long rods. These benefits include less required effort per agent in generating torques for maneuvering the structure, Assuming the rod-model to correspond to that of a clamped-free (or clamped-clamped) beam where the clamped side is at the used polygon and the free one at the copter side, this provides significant static displacement at its tip (copter-side). As an example, in Fig. 3, a rod with the same attributes employed in our experimental studies (280mm length, 5mm diameter, ABS-material) is subjected to a 0.46 N thrust force by the attached copter at the end and results in a 10 mm bending at the copter side and an angle $\gamma \simeq 6.5$ degrees; the generated result is computed using the ComSol Multiphysics software.

This flexibility has adverse effects on the IMU-sensor readings of each copter, since these are being made with respect to the deformed (flexible) system $[O_i', (\hat{x}_i', \hat{y}_i', \hat{z}_i')]$ rather than the rigid one $[O_i, (\hat{x}_i, \hat{y}_i, \hat{z}_i)]$. Furthermore the control inputs $T_i, M_i$ of the $i$-th copter must be computed with respect to the $\hat{z}_i'$-axis. In the static configuration, the rotation between $[O_i, (\hat{x}_i, \hat{y}_i, \hat{z}_i)]$ and $[O_i', (\hat{x}_i', \hat{y}_i', \hat{z}_i')]$ is a rotation around the $y_i$-axis by a bending angle $\gamma_i$, while the

$\mathbf{2}$ The same analysis holds for non-planar drones, where there is a nonzero displacement in the $z_i'$-axis, resulting in the $(3,4)$ element for matrix $sA_i$ to be $s_{c_{i,z}}$ rather than 0.
relative displacement is modeled by an elevation \( \delta_i^z \) along the \( z_i \) axis. The transformation between these systems is \( R(z, \delta_i^z) \). In the sequel let the altitude and attitude of the centroid of the multi-copter system (including the carried payload) expressed with respect to the Earth coordinate system be denoted as \( E \mathbf{x}_c = [x_c, y_c, z_c]^T \) and \( E \Theta_c = [\phi, \theta, \psi]^T \). It should be noted that the coordinate system \( \{O_i, (\hat{x}_i, \hat{y}_i, \hat{z}_i)\} \) does not coincide with the coordinate system of the multicopter system, even when the roll, pitch and yaw angles are zero \( [\phi, \theta, \psi] = [0, 0, 0] \) for the case of a non zero asymmetric payload \( (m_p \neq 0) \).

The system’s dynamics can be computed by taking the extra transformation into account as:

\[
\begin{bmatrix}
\dot{x}_c \\
\dot{y}_c \\
\dot{z}_c
\end{bmatrix} = \begin{bmatrix}
0 \\
0 \\
-\frac{g}{m}
\end{bmatrix} + \frac{1}{m} E R_s (\phi, \theta, \psi) \\
\times \sum_{j=0}^{n-1} \begin{bmatrix}
0 \\
0 \\
T_j
\end{bmatrix}, \tag{1}
\]

\[
\begin{bmatrix}
\dot{\phi} \\
\dot{\theta} \\
\dot{\psi}
\end{bmatrix} = \begin{bmatrix}
I_s \dot{\phi} \\
I_s \dot{\theta} \\
I_s \dot{\psi}
\end{bmatrix} + \begin{bmatrix}
\tau_x^c \\
\tau_y^c \\
\tau_z^c
\end{bmatrix} + \begin{bmatrix}
\tau_x^c \\
\tau_y^c \\
\tau_z^c
\end{bmatrix}, \tag{2}
\]

with the torque control definition:

\[
\begin{bmatrix}
\tau_x^c \\
\tau_y^c \\
\tau_z^c
\end{bmatrix} = \sum_{j=0}^{n-1} \begin{bmatrix}
\delta_{j,j}^x \\
\delta_{j,j}^y \\
\delta_{j,j}^z
\end{bmatrix} + \begin{bmatrix}
R_z(\alpha_j) [0] \\
[0] \\
[T_j]
\end{bmatrix}
\times \begin{bmatrix}
0 \\
0 \\
T_j
\end{bmatrix}, \tag{3}
\]

\[
\begin{bmatrix}
\tau_x^c \\
\tau_y^c \\
\tau_z^c
\end{bmatrix} = r_p \times \left( E R_s \right)^T \begin{bmatrix}
0 \\
0 \\
-m_p g
\end{bmatrix}. \tag{4}
\]

In the previous formulation, \( g \) is the gravity, \( m \) the total mass (including the unknown payload), \( E R_s = R_z(\psi) R_y(\theta) R_x(\phi) \) is the 3 x 3 rotation matrix from the structure-fixed frame to the Earth, \( \tau_i^x (I_i) \) represents the unknown static torque (the moment of inertia) acting on the structure’s \( i \) th-axis, \( i \in \{x, y, z\} \) due to a Center of Mass (CoM) displacement, \( m_p, r_p \) the unknown mass and 3D displacement vector between structure CoM and payload CoM, \( \gamma_j, \delta_j^z \) the bending angle and elevation (deformation) experienced by copter-\( j \) respectively, \( T_j \) the total thrust produced by copter-\( j \) and \( M_j \) the total yaw moment produced by the propellers of copter-\( j \), \( j = 0, \ldots, n-1 \).

In the above dynamics, if \( \gamma_j \) goes to zero (i.e. fully rigid beams with no deflection) then nominal multi-rotor flight dynamic model is retrieved. In Eq. 3 the only torque \( (M_j) \) generated by each agent \( j \) is about its \( z_j \)-axis. This is a direct outcome of the controller design, as only the thrusts of the agents are selected to control the attitude of the structure, with each agent being commanded to produce zero individual torques about its own \( x_j, y_j \) axes.

Let \( s(\cdot) = \sin(\cdot), \ c(\cdot) = \cos(\cdot) \), then Eq. 1 can be condensed as

\[
\begin{bmatrix}
\dot{x}_c \\
\dot{y}_c \\
\dot{z}_c
\end{bmatrix} = -g \begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix} + \frac{1}{m} E R_s \begin{bmatrix}
-s(\gamma_0)c(\alpha_0) & \ldots & -s(\gamma_{n-1})c(\alpha_{n-1}) \\
-s(\gamma_0)s(\alpha_0) & \ldots & -s(\gamma_{n-1})s(\alpha_{n-1}) \\
-c(\gamma_0) & \ldots & -c(\gamma_{n-1})
\end{bmatrix}
\times \begin{bmatrix}
T_0 \\
\vdots \\
T_{n-1}
\end{bmatrix}, \tag{5}
\]

Contrary to the rigid-dynamics case, where the sum of thrusts is relevant for position control, in the “flexible”-rod system dynamics, the individual thrusts appear in Eq. 5.

Similarly the attitude dynamics can be compacted: consider \( \Sigma = E \Theta_c \), the diagonal inertia matrix \( J = \begin{bmatrix}
I_x & 0 & 0 \\
0 & I_y & 0 \\
0 & 0 & I_z
\end{bmatrix} \)

\[
\Sigma = \begin{bmatrix}
\Sigma_0 & \cdots & \Sigma_{n-1}
\end{bmatrix}^T, \tag{6}
\]

where

\[
\Sigma_i = \begin{bmatrix}
\delta_{i,j}^x s(\gamma_i) c(\alpha_i) + s_{i,j} c(\gamma_i) \\
\delta_{i,j}^y s(\gamma_i) c(\alpha_i) - s_{i,j} c(\gamma_i) \\
\delta_{i,j}^z s(\gamma_i) c(\alpha_i) - s_{i,j} c(\gamma_i)
\end{bmatrix},
\]
and the unknown static torque vector \( \tau^s = [\tau^x, \tau^y, \tau^z]^T \), then Eq. 3 can be rewritten as

\[
\mathbf{J} \dot{\Omega} = -\Omega \times (\mathbf{J} \Omega) + \Xi \begin{bmatrix} T_0 \\ \vdots \\ T_{n-1} \end{bmatrix} + \Psi \begin{bmatrix} M_0 \\ \vdots \\ M_{n-1} \end{bmatrix} + \tau^s. (7)
\]

It is apparent that the yaw moments of all copters are reoriented in a similar manner thus taking into advantage the adjustment of the thrusts-induced altitude control.

Although the re-projection of the thrusts due to the \( \Xi \) matrix seems complicated, due to the small numbers involved in the first two rows of each column \( \Xi_i \) since when \( \gamma_i \simeq 0^\circ \), then \( \Xi_i \simeq \begin{bmatrix} i c_{i,y}, -i c_{i,x} \end{bmatrix}^T \), and its effect can easily be quantified.

Under the assumption of a clamped-free bending beam model for each polygon-drone case, where the polygon-side corresponds to the clamped-end and the drone-side the free one, then the maximum deflection and slope for the rod can be computed. Using an Euler-Bernoulli formulation and elastic rods with uniform density and section moments of inertia, then the elastic-rod’s dynamics is

\[
\rho \frac{\partial^2 z_i(x,t)}{\partial t^2} + E I \frac{\partial^2 z_i(x,t)}{\partial x^2} = 0, \quad i \in [0, \ldots, n-1], \quad x_i \in [-l_i, 0] \quad \text{where} \quad \rho \quad \text{is the mass linear density}, \quad E \quad \text{is the Young’s modulus}, \quad I \quad \text{the moment of inertia, and} \quad z_i(x_i, t) \quad \text{the deflection of the rod along the} \quad x_j \quad \text{-axis}. \quad \text{The boundary conditions are} \quad z_i(-l_i, t) = 0, \quad \frac{\partial z_i(l_i, t)}{\partial x_i} = 0, \quad \frac{\partial^2 z_i(0, t)}{\partial x_i^2} = 0, \quad \text{and} \quad \frac{\partial^2 z_i(l_i, t)}{\partial x_i^2} = T_i(t) - m_i g. \quad \text{Assuming static loading conditions} \quad (T_i(t) = T_i), \quad \text{then} \quad z_i(x_i, t) = (T_i - m_i g) \frac{(x_i - l_i)^3}{6 E I} (3l_i - x_i), \quad \text{with a maximum deflection (slope) at the drone attachment}
\]

\[
\delta^z = \max z_i(l_i, t) = (T_i - m_i g) \frac{l_i^3}{6 E I}, \quad \gamma_i = (T_i - m_i g) \frac{l_i^2}{2 E I}. (8)
\]

These static ‘bending’ estimates of each rod, given the corresponding copter’s thrust will be used in the controller design as disturbances that need to be attenuated.

### 4 Controller Design

Classical open-source controllers (like the Ardupilot) cannot be easily used in this concept, since their firmware has been adjusted for specific configurations (coaxial, tricopter, traditional helicopter, tandem, et al.). For this reason the customized centralized controller architecture uses: 1) a position controller computing the desired total thrust force and attitude of the structure, and 2) an attitude controller that computes the necessary torque control inputs. Inhere, an optimizer finds individual agent thrusts and yaw moments \((T_j, M_j)\), in order to produce the desired total thrust and torques. These individual agent setpoints are communicated to each agent’s on-board flight controller to adjust the angular velocities \(\Omega^m_i\) of its individual motors.

The adopted control framework is shown in Fig. 4, where all variables and equations are explained in the sequel.

### 4.1 Positioning Controller

The controller avoids creating aggressive maneuvers, and the position controller is thus computed for the multicopter’s linearized dynamics around hovering [24]. Under the assumption of equal participation by each copter to the necessary thrust \((T_i^o = \frac{m g}{n})\), then the linearization dynamics from Eq. 5 around \(E \mathbf{X}_0 = \begin{bmatrix} x_0^o, y_0^o, z_0^o \end{bmatrix}, \quad \phi, \theta, \psi \end{bmatrix}^o = (0, 0, 0)\), where \((\cdot) = (0) + \Delta(\cdot)\), result in:

\[
\begin{bmatrix} E \Delta \mathbf{X}_0 \\ E \Delta \mathbf{X}_f \end{bmatrix} = \begin{bmatrix} E \Delta \mathbf{X}_0 \\ 0_{3 \times 1} \end{bmatrix} + \begin{bmatrix} 0_{3 \times 3} \mathbf{A}_o \end{bmatrix} \begin{bmatrix} \Delta \phi \\ \Delta \theta \\ \Delta \psi \end{bmatrix}
\]

\[
+ \begin{bmatrix} 0_{3 \times 3} \mathbf{B}_o \end{bmatrix} \begin{bmatrix} \Delta T_0 \\ \vdots \\ \Delta T_{n-1} \end{bmatrix}, \quad \text{where}
\]

\[
\mathbf{A}_o = \begin{bmatrix} 0 & -\sum_{i=0}^{n-1} c(\alpha_i) \gamma_i & -\sum_{i=0}^{n-1} \sum_{i=1}^{n-1} c(\alpha_i) r_{ij} \\
-\sum_{i=0}^{n-1} s(\alpha_i) \gamma_i & 0 & -\sum_{i=0}^{n-1} \sum_{i=1}^{n-1} s(\alpha_i) r_{ij} \\
\sum_{i=0}^{n-1} s(\alpha_i) \gamma_i & -\sum_{i=0}^{n-1} \sum_{i=1}^{n-1} s(\alpha_i) r_{ij} & 0
\end{bmatrix}, \quad (10)
\]

\[
\mathbf{B}_o = \begin{bmatrix} c(\alpha_0) \gamma_0 & \cdots & c(\alpha_0) \gamma_{n-1} \\
0 & \cdots & 0 \\
\frac{c(\alpha_0) \gamma_0}{m} & \cdots & \frac{c(\alpha_0) \gamma_{n-1}}{m}
\end{bmatrix}, \quad (11)
\]

For small angles \((\gamma_i \simeq 0^\circ)\), skew symmetric matrix \(\mathbf{A}_o\) and matrix \(\mathbf{B}_o\) degenerate to

\[
\mathbf{A}_o = \begin{bmatrix} 0 & 1 & -\sum_{i=1}^{n-1} s(\alpha_i) \gamma_i \\
0 & 0 & -\sum_{i=1}^{n-1} c(\alpha_i) \gamma_i \\
0 & 0 & 0
\end{bmatrix}
\]

\[
\mathbf{B}_o = \begin{bmatrix} c(\alpha_0) \gamma_0 & \cdots & c(\alpha_0) \gamma_{n-1} \\
0 & \cdots & 0 \\
\frac{c(\alpha_0) \gamma_0}{m} & \cdots & \frac{c(\alpha_0) \gamma_{n-1}}{m}
\end{bmatrix}
\]

It should be noted that the elements of the last row of matrix \(\mathbf{B}_o\) are independent of \(\gamma_i\), and thus the vertical acceleration of the multicopter system can easily be computed. In this case (small deflection angles),

\[
\Delta \mathbf{x}_e = g \Delta \theta + \sum_{i=0}^{n-1} \gamma_i \left[ \frac{s(\alpha_i) \Delta \psi}{n} - \frac{c(\alpha_i) \Delta T_i}{m} \right] = g \Delta \theta + \sum_{i=0}^{n-1} \gamma_i \Delta \mathbf{b}_i \quad (12)
\]
Fig. 4 Controller Diagram

\[ \Delta \hat{e}_z = -g \Delta \phi + \sum_{i=0}^{n-1} \gamma_i \left[ -c(\alpha_i) \Delta \psi - s(\alpha_i) \Delta T_i \right] = -g \Delta \phi + \sum_{i=0}^{n-1} \gamma_i \hat{e}_z^i \]

(13)

\[ \Delta \hat{e}_z = \sum_{i=0}^{n-1} \Delta T_i \frac{m}{2} + \sum_{i=0}^{n-1} \gamma_i \left( -g \Delta \phi + \frac{c(\alpha_i) \Delta \phi + s(\alpha_i) \Delta \theta}{n} \right) = \sum_{i=0}^{n-1} \Delta T_i \frac{m}{2} + \sum_{i=0}^{n-1} \gamma_i \hat{e}_z^i \]

(14)

**Theorem 1** The backstepping PD-alike altitude controller computes

\[ \sum_{i=0}^{n-1} \Delta T_i = m \left( -K_{z1} - K_{z2} \right) \hat{e}_z - \left[ 1 + K_{z1} K_{z2} \right] e_z - \sum_{i=0}^{n-1} \gamma_i \hat{e}_z^i \]

(15)

where \( e_z = z^o - z^d \), and \( K_{z1}, K_{z2} > 0 \). The control input including the feedforward term \( T_i^o \) and the differential thrusts \( \Delta T_i \) satisfying Eq. 15 forces \( e_z \to 0 \).

**Proof** Application of the backstepping control principle [25], results in abbreviated Lyapunov function \( V_z = \| e_z \|^2 + \| K_{z1} e_z \|^2 \) and application of Eq. 15 achieves \( \dot{V} \leq 0 \) rendering the closed-loop altitude system stable. An analytic proof of the above application of backstepping control is found in Appendix A.

**Remark 1** For the rigid-case, where \( \gamma_i = 0 \), the controller degenerates to that of a PD-controller.

**Remark 2** The controller-formulation assumes knowledge of the carried payload in computing \( T_i^o \) and in Eq. 15. If this is unknown, an extra term should be added and the controller needs to be modified as follows

\[ \dot{\hat{e}}_z = -\sigma \left( \hat{e}_z + K_{z1} e_z \right) \left[ -K_{z1} - K_{z2} \right] \hat{e}_z - \left[ 1 + K_{z1} K_{z2} \right] e_z - \sum_{i=0}^{n-1} \gamma_i \hat{e}_z^i \]

(16)

\[ \dot{\gamma}_i = m \left[ -c(\alpha_i) \Delta \psi - s(\alpha_i) \Delta T_i \right] \]

(17)

**Theorem 2** The backstepping based PD-alike controller, adjusting the roll and pitch angles as

\[ \phi^d = [-K_{x1} - K_{x2}] \hat{e}_x - \left[ 1 + K_{x1} K_{x2} \right] e_x - \sum_{i=0}^{n-1} \gamma_i \hat{e}_x^i \]

(18)

where \( K_{x1}, K_{x2}, K_{y1}, K_{y2} > 0 \) renders the closed-loop altitude system stable yielding \( e_x \to 0 \) and \( e_y \to 0 \).

**Proof** Same as Theorem 1.

4.2 Attitude Controller

The development of the attitude controller is subject to the following assumptions

**Assumption 1** The flexibility effects on the system dynamics are small and terms like \( \delta_i \sin(\gamma_i) \) can safely be neglected.

**Assumption 2** The yaw-torques induced by matrix \( \Psi \) in Eq. 5 in the \( x \)- and \( y \)-axes can be neglected and \( \Psi \simeq \begin{bmatrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ 1 & \cdots & 1 \end{bmatrix} \). Essentially this is an indication of the relative lengths of the rods compared to the negligible flexibility effects since \( \delta_i, \gamma_i \simeq 0 \).
Given these assumptions and for small $\gamma$, Eq. 7 is transformed to:

$$\ddot{\mathbf{J}} = -\Omega \times (\mathbf{J} \Omega) + \dot{\mathbf{r}}^c + \dot{\mathbf{r}}' = -\Omega \times (\mathbf{J} \Omega) + \left[ \begin{array}{c} \tau_0^c \\ \vdots \\ \tau_n^c \end{array} \right] + \left[ \begin{array}{c} T_0 \\ \vdots \\ T_{n-1} \end{array} \right],$$

(19)

where $\zeta = \gamma \left[ s_{c_{i,y}} c(\alpha_i) - s_{c_{i,x}} s(\alpha_i) \right]$; the top-two rows of Eq. 20 is the $(x, y)$ thrust-to-torque allocation matrix.

The objective of the attitude controller is to compute the control vector $\tau^c$ in order to regulate the attitude dynamics. Subsequently the thrust optimizer assigns the individual thrusts $T_i$, while the total yaw moment is computed from Eq. 20 as:

$$\sum_{i=0}^{n-1} M_i = \tau_z^c - \sum_{i=0}^{n-1} \zeta_i T_i.$$  

(21)

Let the real moments of inertia and static torques acting on a given structure be constants, then the torque control vector is defined as

$$\tau^c = \Omega \times (\mathbf{J} \cdot \Omega) + \tilde{\mathbf{J}} (-K_\phi e_\phi - K_\omega e_\omega - K_a z_\phi) - \tilde{\tau}^s,$$

(22)

where $\tilde{\mathbf{J}}$ and $\tilde{\tau}^s$ represent adaptations for the unknown moment of inertia matrix and static torque vector, $K_\phi$, $K_\omega$ are diagonal positive gain matrices and $e_\phi$, $z_\phi$ are error vectors defined in Appendix C.

**Theorem 3** The controller Eq. 22 stabilizes the system dynamics Eq. 19 for small deflections.

**Proof** See Appendix C.

### 4.3 Individual Copter Thrust Computation

Having computed the $\tau^c_i$ and $\tau^d_i$ and the desired total thrust

$$T^d \triangleq \sum_{i=0}^{n-1} T_i = \sum_{i=0}^{n-1} (T^o_i + \Delta T_i),$$

the individual agent thrusts $T_i$, $i = 0, \ldots, n - 1$, and yaw torques need to be computed and transmitted to each agent. The thrusts are related through the allocation matrix equation

$$\left[ \begin{array}{c} \tau_x^c \\ \tau_y^c \\ T^d \end{array} \right] = \sum_{i=0}^{n-1} \left[ \begin{array}{c} s_{c_{i,x}} \\ s_{c_{i,y}} \\ 0 \end{array} \right] \times \left[ \begin{array}{c} 0 \\ 0 \\ T_i \end{array} \right] = \Gamma \mathbf{T} = \Gamma \left[ \begin{array}{c} T_0 \\ \vdots \\ T_{n-1} \end{array} \right].$$

(23)

For multi-copter systems, where $n > 3$, Eq. 23 has infinite solutions in computing $\mathbf{T}$. In most works, the pseudo-inverse is used [26]. Hence, an optimization procedure is devised for the computation of $\mathbf{T}$ while satisfying certain metrics.

---

**Fig. 5** Battery depletion voltage $B_i$ history for various $m_p$

**Fig. 6** Modular quad-copter configuration
Fig. 7 Modular quad-copter attitude controller response

Fig. 8 Quad-copter and T-copter responses
where $T^{\text{max}}$ the maximum thrust that can be provided by the agent. Equation 24 is solved in a centralized manner to compute the optimal thrust of every agent.

4.3.2 Flight Time Maximization with Maneuvering Efficiency

The system’s flight time is dictated by the minimum flight time among agents, since in case of an agent’s battery depletion the system needs to land. In [27] (section F), there is an experimental finding that relates the battery depletion to the generated thrust. Thus, maximization of the flight time can be achieved by minimization of the maximum thrust level among agents, or

$$f_T = \| \begin{bmatrix} T_0, \ldots, T_{n-1} \end{bmatrix}^T \|_{\infty}. \quad (25)$$

Other approaches include a quadrant-based thrust selection procedure [14] and the pseudo-inverse which in general result in a smaller mean thrust value among all agents, while the adoption of the $\infty$-norm results in the lowest maximum thrust among all agents.

Reducing the system’s maneuvering response is equivalent to generating large torques from the agents. Since the $T^{\text{max}}$ is fixed, these torques can be generated by copters away from the system’s center of mass [14, 28]. This is explained since in Eq. 23, copters with larger $s_{c_i,x}, s_{c_i,y}$ generate the same torques with smaller thrust adjustment. Thus, agents with large values of $\gamma_1(1, i)$ and $\gamma_1(2, i)$ should participate more in the control effort. Since a minimization scheme is employed, then agents with small values of $\Gamma(1, i)$ and $\Gamma(2, i)$ should be rewarded. To further emphasize this reward, some agents with low torques are considered inactive and the ones with high torques are further emphasized; variables $\epsilon_x$ and $\epsilon_y$ are introduced for this reason, defined as

$$\epsilon_x(\alpha_{\text{min}}, \alpha_{\text{max}}) = \text{sat} \left( \frac{\tau_x}{\tau_{x,\text{max}}}, \alpha_{\text{min}}, \alpha_{\text{max}} \right),$$

$$\epsilon_y(\alpha_{\text{min}}, \alpha_{\text{max}}) = \text{sat} \left( \frac{\tau_y}{\tau_{y,\text{max}}}, \alpha_{\text{min}}, \alpha_{\text{max}} \right).$$
Fig. 11  Path following by the hexa-copter configuration

where the saturation function returns 0 when the normalized input \( \frac{\tau_c x}{\tau_{c,\text{max}}} \) drops below \( \alpha_{\text{min}} \) and 1 above \( \alpha_{\text{max}} \). \( 0 \leq \alpha_{\text{min}} < \alpha_{\text{max}} \leq 1 \) and \( \tau_{c,\text{max}} \) is the largest anticipated control effort in the \( x \)-direction. A similar function is defined for the \( y \)-component, named \( \epsilon_y (\alpha_{\text{min}}, \alpha_{\text{max}}) \). Then the maneuvering efficiency metric encapsulating the aforementioned is

\[
f_M =\sum_{i=0}^{n-1} \left( \frac{\epsilon_x}{|\Gamma(1, i)|} + \frac{\epsilon_y}{|\Gamma(2, i)|} \right) T_i.
\]  

(26)

A combined version of the previous metrics maximizes the flight time while reducing the response time as

\[
f_E = \epsilon f_T + (1 - \epsilon) f_M, \quad 0 \leq \epsilon \leq 1.
\]  

(27)

This metric utilizes agents further from the centroid, while maneuvering, while optimally distributes the thrust requirements among the agents in near-hovering conditions.

**4.3.3 Battery Time Optimization**

The individual agent’s battery consumption \([29, 30]\) depends on several factors, including the rotor angular velocity, the motor-propeller combination, while frequently ignore aerodynamic effects of rotary-wing aircraft. Blade-element-momentum theory is used for describing these effects while the remaining battery capacity is calculated using Peukert model. The resulting model is quite complicated and rather empirical and results in polynomial methods to express the battery capacity. The adopted battery time metric is formed along this empirical method and the adopted metric is

\[
f_B = \sum_{i=0}^{n-1} \left( \frac{T_i^2}{1 - e^{(\delta - B_i)}} \right),
\]  

(28)

where \( B_i \) is the battery voltage reading of agent \( i \). The \( \delta \)-constant is the lowest operational voltage of the battery and depends on the payload and the induced temperature. This metric penalizes agents with battery levels \( B_i \approx \delta \) and

Fig. 12  Asymmetric Pentacopter System
assumes that the battery is depleted according to the square of the thrust.

For all metrics, the YALMIP optimization toolbox [31] was utilized along with the MOSEK solver [32] to provide a solution to the optimization problem. The typical optimized solution could be found in 5 msec and updated this solution (thrust and yaw moment setpoints) on the controller component every 5 msec.

5 Experimental Studies

5.1 Control Framework Implementation

In order to validate the proposed controllers, the Crazyflie quadrotors [33] vehicles were used, in order to create flying structures. Hexagons and square connecting devices were used, while the structure rods are identical with \( l_i = 0.14 \) m.

The weight of each rod is 3.5 g, the hexagon’s is 9 g and the square’s is 7 g. Powerful neodymium magnets were inserted at each rod-end capable of providing a 7 N attractive force.

The Crazyflie-ROS stack [34] is used for communication with all agents and the ground station.

The centralized nature of the controller demands an estimate of the state of the multi-copter system and requires the solution of the nonlinear optimization problem Eq. 24. The flight time Eq. 25 or battery time Eq. 28 maximization cases result in a quadratic optimization problem. The computation of its constrained optimum can be achieved within 7 simple iterations requiring 53.95MFlops\(^3\). The overall position of the system is provided by a motion capture system operating at 120 Hz, while an Extended Kalman Filter (EKF) estimates the system’s orientation. The prediction step is performed by using the latest control commands, while the attitude update uses measurements received from all agents’ IMUs [35].

Once the controller has computed the attitude control and subsequently the thrust and yaw moment required for each agent, the setpoint command

\[
\begin{bmatrix}
\phi^d_i,
\theta^d_i,
T_i,
M_i
\end{bmatrix}^T
\]

is generated. \( \phi^d_i \) and \( \theta^d_i \) are extracted from \( R_z(\alpha_i)R_B^WR_z(\alpha_i) \), where \( R_B^W \) is the rotation matrix representation of the current estimate of the multi-copter’s attitude maintained at the ground station’s EKF and \( \alpha_i \) is defined in Section 2. Indirectly, the rotation of agent-\( i \) as seen from its own IMU is used, thus leading to commands \( \phi^d_i, \theta^d_i \) equal to the roll and pitch angle estimates maintained at the on-board of the \( i \)th agent autopilot. The setpoints are transmitted to each Crazyflie every 5 msec, while the commands to the position controller are transmitted every 20 msec.

\(^3\) An Intel i7-8850H computer can complete the search in 1.3msec, while an on-board Cortex A53 requires 10msec.
5.2 Crazyflie Battery Characterization

The operational battery voltage characteristics for the Crazyflie quadcopters is depicted in Fig. 5, where $B^{\text{max}}_i = 4\text{Volt}$ and $\delta = 2.75\text{ Volt}$ for 80% of the maximum allowable payload $(m^{\text{max}}_p = 13.125\text{ g})$ and $\delta = 2.6\text{ Volt}$ for $m_p = 0\text{ g}$.

The battery depletion history for two Crazyflie quadcopters in a hovering mode is shown in Fig. 5. In all cases, there is a sudden drop from the non-operating voltage, followed by a slow voltage drop leading to a sudden voltage drop before the need to land these quadcopters. The non-carrying quadcopter’s batteries lasted until 430 seconds (blue line) in comparison with the loaded agent which lasted 270 seconds. Different $\delta$ values were recorded depending on the payload. The maximum value of all $\delta$s for varying payloads was used; $\delta = 2.9\text{ Volt}$ for $m_p = 90\%$ of the maximum available carrying payload.

The empirical battery depletion versus the carried payload which corresponds to the needed thrust is also validated in Fig. 5, hence the agent operating at maximum thrust among the system, dictates the system’s flight time due to the faster depletion of its battery.

5.3 Flight Experiments with Rigid Copter-Structure

In this section, experiments are presented, showcasing the efficiency of the proposed control methods in achieving cooperative flight. Cases spanning $n = 3, \ldots, 6$ copters are presented with short rods which correspond to a rather rigid structure, since $\max \delta^* = 0.0025\text{m}$ and $\max \gamma = 1.37^\circ$.

5.3.1 Quad-copter & T-copter

A symmetric quad-copter ($n = 4$) was configured, as shown in Fig. 6, in order to perform the initial evaluation of the flight controller.

The attitude response of the quad-copter is evaluated first, where the copter is in an altitude hold mode having the $z$-axis component of the position controller as active. Pulse com-

![Fig. 14 Agent thrust comparison for the penta-copter](image-url)
mands are issued for the roll and pitch reference angles and a step command for the yaw. As seen in Fig. 7, the vehicle is capable of fast-tracking these roll and pitch commands while the response for the yaw-command is slower, as anticipated from [15].

Subsequently, the performance of the position controller is also evaluated. The symmetric quad-copter and an asymmetric T-copter \((n = 3)\) were used for comparison, where the T-copter is created by removing one rod/copter. The same waypoints were transmitted to both 4-copter and 3-copter systems, which had the same controller employed relied on the pseudo-inverse in solving Eq. 23. As shown in Fig. 8, the quadcopter achieves slightly faster transitions and has less turbulence during its take-off phase which is anticipated due to its symmetry.

Figure 9 shows the evolution of the \(x, y\) static torque and the \(x, y\) diagonal elements of the inertia matrix adaptations for the T-copter configuration. In asymmetric configurations, the need to offer adaptations is apparent since there are different converging values between \(\hat{J}_{xx}\) and \(\hat{J}_{yy}\) and similarly between \(\tau_{sx}\) and \(\tau_{sy}\).

5.3.2 Highly Asymmetric Hexacopter

The hexacopter of Fig. 1, seen mid-flight in Fig. 10, was configured to test the controller design. The connecting rod between the hexagon to the square element exhibited significant vibrations which affected the system’s overall response. The system’s center of mass does not lay on any of the connecting elements (rods or polygons).

Moreover, the agents thrust exceeded 70% of their maximum throttle for lifting the structure’s weight.

For a similar waypoint navigation, proper flight can be achieved, as seen in Fig. 11 (for three waypoints), showcasing the effectiveness of the proposed controller, in flying arbitrary structures.

5.4 Thrust Allocation Comparisons

A pentacopter shown in Fig. 12 was created, in order to compare the different thrust control allocation methods proposed and their preferred usage; the copter enumeration is shown
in the same Figure. In the subsequent cases, the criterion of maximizing the flight time and decrease the maneuvering response time is used from Eq. 27, where $\epsilon = 0.67$, $\alpha_{\text{min}} = 0.1$, $\alpha_{\text{max}} = 1$ and $\tau_{x,\text{max}} = 0.09\text{Nm}$, $\tau_{y,\text{max}} = 0.09\text{Nm}$.

### 5.4.1 $f_E$-metric vs. Pseudo-Inverse Response

The first experiment was conducted to compare the performance of the thrust allocation controller of Eq. 27 with that of the simple pseudo-inverse. A series of step maneuvers were commanded to the pentacopter. The achieved flight history is shown in Fig. 13 and the corresponding thrusts commanded by the controller for each agent are seen in Fig. 14.

The proposed $f_E$-minimization controller outperforms the pseudo-inverse, achieving a smoother flight with less oscillations by examining the thrust commands in Fig. 14, looking
Fig. 18  Collaborative payload transportation position-history

Fig. 19  Agent thrusts during collaborative payload transportation
into the high frequency thrust component of the copters. For the $f_E$-thrust allocation controller, short thrust excursions are observed during its maneuvers; these occur when $\epsilon_x$ or $\epsilon_y \neq 0$, which corresponds to 4% of the flight time. Furthermore, the 'pseudoinverse'-controller can result in infeasible commands that need to be saturated.

5.4.2 Battery-Life Optimizer

The efficiency of the thrust allocation controller of Eq. 28 was examined for the penta-copter platform. Initially, all batteries were fully charged at 4.1 Volt, except for the battery on agent 3, which was depleted at 3.85 Volt. A take-off followed by a hovering experiment was conducted to quantify the proposed optimizer’s efficiency compared to the ‘pseudoinverse’ approach.

The control allocation shown in Fig. 15 commands on the average 10% less total thrust from agent-3 compared to the pseudo-inverse, due to its awareness of the battery voltage level. This is a significant reduction, given that an individual Crazyflie agent was operating at 59% of its thrust capacity in autonomous hovering.

The effects of the reduced thrust requirement on agent 3 are visible in Fig. 16, where the evolution of the battery voltage readings is shown. For all other agents, except the third one, there is a faster rate of voltage decrease, resulting in small voltages $B_i, i = 1, 2, 4, 5$ after the completion of the experiment.

5.5 Payload Transportation Experiments

In this section, experiments are conducted to showcase the usage of the designs in cooperative payload transportation of an asymmetric payload of an L-shape, similar to the one in [17, 28]. Second generation lightweight (less than 0.4g) joints have been fabricated and tested for this reason. T-
and L-junctions were incorporated to form the structure’s skeleton and sturdily connect polygons and carbon rods in a lightweight manner, as shown in Fig. 17.

Typical waypoint navigation flight segments for the structure are plotted in Fig. 18, along with the thrusts commanded to each agent by the thrust allocation optimization controller in Fig. 19.

5.6 Flight Experiments with Flexible Copter-Structure

A T-copter structure with four copters \((n = 4)\) is used in this Section; the fourth one is at an elevated\(^4\) centered position. Two rods of \(l_1 = l_2 = 14\text{cm}\) are used while the third one has 30 cm length. The long rod resulted in a flexible structure as shown in Fig. 20. Due to the flexibility of the third rod \(\delta_3^e = 0.02\text{m}\) and \(\gamma_3 = 5.73^\circ\), while the elevated one was at 6 cm vertical distance from the remaining ones. The \(f_E\)-metric optimizer was used for the thrust allocation controller, while the parameter \(\gamma_i\) used in Eqs. 17 and 18 was computed from Eq. 8.

The static flexibility effects Eq. 8 of the 30-cm clamped free rod was examined for various thrusts. Figure 21 shows the time-history of \(\gamma_3\) angle for the (rod/copter) that had \(m_3 = 37\text{ g}\). The applied thrust \(T_3\) is shown in a dotted line, while the static (actual) \(\gamma_3\) angle is shown in blue (red) color. The actual response exhibits significant oscillations due to the unaccounted modes of vibration.

The altitude response of the system appears in Fig. 22, where there is significant reduction in the system oscillations caused by the system’s flexibility.

\(^4\) The developed method is also valid for copters that are parallel-placed at different altitudes.

6 Conclusions

In this paper, a generalized framework for flying arbitrary modular multi-copter structures was presented. A novel control approach for collaboratively flying such interconnected aerial systems was proposed, relying on the combination of the total thrust produced by each agent, while taking into account the structure’s flexibility. The feasibility of the proposed scheme has been experimentally validated using prototype copters and custom designed connecting structure elements.

Acknowledgements This research was in part performed by using NYUAD’s Core Technology Platform Kinesis lab motion capture system. The authors thank Mr. Nikolaos Giakoumidis and Dr. Oraib Al Ketan for their technical support and insights.

Author Contributions All authors contributed to the study conception. The primary design was made by D. Chaikalis (DC), followed by modifications made by DC and A. Tzes (AT). The experiments were performed by DC and N. Evangelion. The first draft of the manuscript was written by DC and edited by AT. All authors read and approved the final manuscript.

Funding The authors declare that no funds, grants, or other support were received during the preparation of this manuscript. This work is supported in part by the NYUAD Center for Artificial Intelligence and Robotics (CAIR), funded by Tamkeen under the NYUAD Research Institute Award CG010.

Declaration

Conflicts of interest The authors have no relevant financial or non-financial interests to disclose.

Ethical standard No ethical approval is required for publishing his research.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Appendix A

Let \(\dot{z} = z^\circ + \Delta z\), and \(T = T^\circ + \sum_{i=0}^{n-1} \Delta T_i\), where \(z^\circ\) and \(T^\circ\) are constants, then \(\dot{\dot{z}} = \Delta \dot{z}\). According to the backstepping control principle, the controller design relies on computing a virtual control signal to stabilize each sub-state sequentially.
Similar to [36], assume the positive definite Lyapunov function $V_1 = \frac{e_z^2}{2}$, where $e_z = z - z^d$ with $z^d$ corresponding to the desired altitude. Then, $\dot{V}_1 = e_z \dot{e}_z = e_z (\Delta \dot{z} - \dot{z}^d)$. Let the virtual input $\dot{z}^*$ capable of stabilizing the altitude error sub-state $\dot{z}^* = \dot{z}^d - K_z e_z$, and the velocity error term $s_z = \Delta \dot{z} - \dot{z}^*$. Then $\dot{V}_1$ can be rewritten as $\dot{V}_1 = e_z (s_z - K_z e_z)$. Let the augmented Lyapunov function $V_z = V_1 + \frac{1}{2} s_z^2$, then

$$\dot{V}_z = -K_z e_z^2 + s_z (e_z + \Delta \ddot{z}^* + K_z (s_z - K_z e_z)), \quad (29)$$

where it was assumed for simplicity that $\dot{z}^d = 0$. Given the altitude dynamics Eq. 13 and the altitude controller (15) applied to Eq. 29 results in $\dot{V}_z = -K_z e_z^2 - K_z s_z^2 \leq 0$. Hence $e_z \to 0$, and $s_z \to 0$, implying that $\dot{z} \to z^d, \ddot{z} \to \dot{z}^d$, since the system is asymptotically stable under the proposed controller.

Likewise, asymptotic stability of $x$ and $y$ can be proven.

**Appendix B**

In the case when there is external payload of unknown mass, the controller needs to be augmented with an online adaptation term for this mass. Assume in Eq. 15, the term $m$ is replaced by its estimate $\hat{m}$, then the Lyapunov function derivative is

$$\dot{V}_z = -K_z e_z^2 - K_z \hat{s}_z^2 + \frac{\hat{m} - m}{m} \hat{s}_z (s_z - K_z \hat{s}_z)$$

$$-e_z - K_z s_z - \sum_{i=0}^{n-1} \gamma_i \xi_i^2.$$ 

Let the positive quantity

$$V_m = \frac{(\hat{m} - m)^2}{2 \sigma m}, \quad \sigma > 0 \quad (30)$$

and the adaptation evolution rule

$$\dot{\hat{m}} = -\sigma \dot{s}_z (s_z - K_z \hat{s}_z) - e_z - K_z s_z - \sum_{i=0}^{n-1} \gamma_i \xi_i^2.$$ 

(31)

Then the augmented Lyapunov function $V_z + V_m$ has derivative $\dot{V} = \dot{V}_z + \dot{V}_m = -K_z e_z^2 - K_z \hat{s}_z^2 \leq 0$. We should note that there is no guarantee that $\dot{m} \to m$ but simply that $e_z$ and $s_z$ converge to zero.

**Appendix C**

Given the system’s attitude dynamics Eq. 19, the attitude control input $\tau^c$ Eq. 22 is computed using adaptive control principles, in order to guarantee the stability of the vehicle’s attitude.

Given desired roll, pitch and yaw angles $(\phi^d, \theta^d, \psi^d)$, let the: a) attitude error vector $e_\phi = [\phi, \theta]^T - [\phi^d, \theta^d, \psi^d]^T$, b) ideal angular velocity as $\Omega^* = \Omega^d - K_\phi e_\phi$, where $K_\phi$ is a diagonal positive gain matrix, c) velocity error vector $z_\phi = \Omega - \Omega^d$, d) adaptation estimates $\hat{J}$, $\hat{\xi}$ for inertia matrix and asymmetric torques acting on the structure, and e) error matrix $\tilde{E} = I_3 - J^{-1} \tilde{J}$, where $I_3$ is the $3 \times 3$ identity matrix.

Based on the backstepping principle, a composite Lyapunov function is defined, incorporating attitude errors and errors in unknown estimates

$$V = \frac{1}{2} e_\phi^T \phi^* + \frac{1}{2} z_\phi^T \phi^* + \frac{1}{2} \tr \left( J^T \Lambda^{-1} \tilde{E} \right)$$

$$+ \frac{1}{2\sigma} (\tau^\phi - \tau^c)^T J^{-1} (\tau^\phi - \tau^c), \quad (32)$$

where $\Lambda$ is a diagonal positive gain matrix and $\sigma$ is a positive constant.

Substituting Eqs. 22 to 32 and using the adaptation evolutions

$$\dot{\hat{J}} = \Lambda^T \phi^* \left[ -K_\phi (z_\phi - K_\phi e_\phi) - e_\phi - K_\omega z_\phi \right]^T, \quad (33)$$

$$\dot{\hat{\xi}} = \sigma z_\phi, \quad (34)$$

then the derivative of the Lyapunov composite function is

$$\dot{V} = -\phi^T K_\phi e_\phi - z_\phi^T K_\phi \phi_\phi \leq 0. \quad (35)$$

In this derivation the symmetry and positive definiteness of the inertia matrices is used. The inertia matrix estimate $\hat{J}(0)$ is computed using the application of the parallel axis theorem on the agent masses and $\hat{\xi}(0) = 0$. Similarly, the same initial estimate for the system inertia matrix is used for the feedforward component of (22).

**References**

1. Gkountas, K., Tzes, A.: Leader/follower force control of aerial manipulators. IEEE Access 9, 17584–17595 (2021)
2. Preiss, J.A., Honig, W., Sukhatme, G.S., Ayanian, N.: Crazywarm: a large nano-quacopter swarm. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), Marina Bay Sands, Singapore. pp. 3299–3304 (2017)
3. Chen, M., Zhou, Z., Wang, R.: Multi-rigid-body dynamics modeling and controller design for a connected UAV formation. In: 2019 IEEE 10th International Conference on Mechanical and Aerospace Engineering (ICMAE), Brussels, Belgium, pp. 187–192 (2019)
Dr. Nikolaos Evangeliou is a graduate (B.Eng & M. Eng) of the ECE Department, University of Patras, Greece, 2010. He received his doctorate in the area of Surgical Robotics at the University of Patras, Greece, 2017. His research interests include Unmanned Aerial Systems (UAVs), Surgical Robotics, Unmanned Ground Vehicles (UGVs), and Control Applications.

Professor Anthony Tzes is the Head of the Electrical Engineering program at NYU Abu Dhabi. Concurrently he is the Lead Investigator of the NYUAD Center for AI and Robotics. His research interests include distributed collaborative control of autonomous mobile agents.

Farshad Khorrami is a professor of ECE at NYU, with research interests in controls, robotics, machine learning, and cyber-physical system security. He has published over 320 papers, has 14 U.S. patents, and has co-directed the Center for AI and Robotics.