2022 American Control Conference (ACC)
Atlanta, USA, June 8-10, 2022

Control of an Assembly of Aerial Vehicles Under Uncertainty

Mohamad T. Shahab, Kévin Garanger, and Eric Feron

Abstract— **In this paper, we consider the problem of controlling a rigid assembly of aerial vehicles under uncertainty. We consider the case when the positions of the vehicle modules in the assembly structure are unknown, but belong to a finite set. In addition, we consider that each module has only its own measurements available for feedback but not that of the whole assembly, so a decentralized control law is developed. We apply an adaptive switching control approach to control this uncertain system. Given a stabilizing controller for the case when there is no uncertainty, we show that the proposed adaptive approach achieves the control objective under uncertainty by presenting illustrative simulation examples; we provide a case study of a recently proposed novel modular flying system, namely a fractal tetrahedron assembly.**

I. Introduction

Systems of multiple robots play an important role in the field of robotics as they provide solutions to problems that would otherwise be hard to tackle by single robots [16]. Robotic assemblies, or modular robots, are such a type of multirobot systems which consist of several physically interconnected modules. Robotic assemblies are typically able to achieve different shapes and sizes depending on the number of modules constituting them and the way they are connected. This reconfiguring ability and the fact that robotic assemblies are made of self-contained elements allow for greater adaptability to the task performed and enhanced robustness via redundancy. Examples of such assemblies are found in [5], [13], [17] and [15].

As a consequence of the multiple possible configurations with potentially an arbitrary number of modules, and relying on the fact that each module has their own sensors and actuators, it is usually desirable to achieve a high level of decentralization of the control system, meaning that every module should rely on their own sensors to decide the proper actuation that will result in the desired behavior at the assembly level. The decentralization requirement implies the need for control methods that account for the uncertainty of the assembly state, as it is in general unreasonable to expect each module to be aware of the state of all the other modules. In practice, this uncertainty can manifest in different ways, such as the number of modules contained in an assembly, the configuration of the assembly, the place occupied by a module given a configuration, or the relative position of the other modules in cases where connections between modules have some degrees of freedom. The problem of reaching a global objective with limited information and interaction between modules, through emergent behavior, is common in multirobot systems and therefore applies to robotic assemblies [10].

Adaptive control is an approach that deals with uncertain or time-varying systems. A classical adaptive controller consists of a time-invariant control law together with a tuning algorithm which adjusts its parameters, e.g. see [6] and [9]. Adaptive control approaches to robotic assemblies are found in the literature, such as in [11], [4] and [21], where interconnected vehicles try to manipulate unknown payloads in a cooperative manner; although there, the vehicles usually need a common measurement of the states shared among all. For nonlinear systems, such as robotic systems, adaptive controllers typically need state feedback [20]; so a different approach may be used, such as one involving a switching mechanism, to side-step this constraint. In adaptive switching control approaches, such as Supervisory Control [14], [8], [7], the adaptive mechanism is typically responsible for switching from time to time between candidate controllers, where it selects the controller corresponding to the "best" performance. Adaptive switching approaches have been used to deal with cases of uncertainty in, for example, controlling underactuated vehicles [1], visual feedback control [3], the control of robotic manipulators [22], and robotic biped walking [2].

In this paper, we consider the problem of controlling, in a decentralized manner, a rigid assembly of aerial robotic vehicles under uncertainty. We assume that we are uncertain about the configuration of the assembly; in particular, the positions of the vehicle modules in the assembly structure are unknown. We employ an adaptive switching approach to show, under a suitable assumption, how a decentralized control law is applied to this uncertain system; we show, through illustrative simulation examples, that the system under uncertainty achieves the control objective, and with comparable closed-loop behavior to the case when there is no uncertainty. As far as the authors are aware, the proposed adaptive switching approach has never been applied in the context of controlling multiple vehicles.

Notation. We denote $\mathbb{R}$, $\mathbb{R}^+$, $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{Z}^+$ as the set of real numbers, nonnegative real numbers, natural numbers, integers and nonnegative integers, respectively. For $N \in \mathbb{N}$, define the set notation $\overline{N} := \{1, 2, \dots, N\}$. We will denote the Euclidean-norm of a vector by the subscript-less default notation $\|\cdot\|$. Let I_p denote the identity matrix of size p. Let $\mathbf{0}_{p\times q}$ denote the matrix of size $p \times q$ whose all of its entries are zero. Define the normal vector $\mathbf{e}_3 := \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top \in \mathbb{R}^3$.

II. The Setup

A. System Model

We consider N modules connected together in a rigid assembly; we label each module by a index $j \in \overline{N}$. We have a world frame defined by x, y and z axes, with the z-axis pointing upward. Let $\mathbf{p} = \begin{bmatrix} x & y & z \end{bmatrix}^\top \in \mathbb{R}^3$ be the coordinates of the center of mass of the whole assembly in the world frame. We consider that the center of mass of the assembly is situated at the origin of a body frame $\mathcal{B}$. The orientation of this body frame is represented by

M.T. Shahab and E. Feron are with the Computer, Electrical, and Mathematical Sciences and Engineering (CEMSE) Division at King Abdullah University of Science and Technology (KAUST), Thuwal 23955, Saudi Arabia. Emails: {mohamad.shahab,eric.feron}@kaust.edu.sa.

K. Garanger is with the Decision and Control Laboratory, Georgia Institute of Technology, Atlanta, Georgia 30332, USA. Email: kevin.garanger@gatech.edu.

Support for the work of M.T. Shahab and E. Feron was provided by the KAUST faculty baseline fund. Support for the work of K. Garanger was provided by a gift from General Atomics.

978-1-6654-5197-0/$31.00 ©2022 AACC

the Euler angles in the world frame, namely, by the yaw angle ψ about the z-axis, the roll angle ϕ about the x-axis, and the pitch angle θ about the y-axis; so let us define $\boldsymbol{\varphi} := \begin{bmatrix}\phi & \theta & \psi\end{bmatrix}^\top \in \mathbb{R}^3$. We have the rotation matrix transforming coordinates from $\mathcal{B}$ to the world frame as

$$R(\boldsymbol{\varphi}) = \begin{bmatrix} c\psi c\theta - s\psi s\theta s\phi & -s\psi c\phi & c\psi s\theta + c\theta s\phi s\psi \\ s\psi c\theta + c\psi s\theta s\phi & c\psi c\phi & s\psi s\theta - c\psi s\phi c\theta \\ -c\phi s\theta & s\theta & c\theta c\phi \end{bmatrix},$$

with $c\chi$ and $s\chi$ short for $\cos\chi$ and $\sin\chi$, respectively. We now represent the dynamics of the system as a rigid-body. With T denoting the total thrust produced by the assembly, the translational dynamics can be represented by

$$m_{\mathcal{B}}\ddot{\mathbf{p}} = -m_{\mathcal{B}} g \mathbf{e}_3 + R(\boldsymbol{\varphi})\mathbf{e}_3 T, \tag{1}$$

with $m_{\mathcal{B}}$ and g denoting the mass of whole assembly and the gravitational acceleration, respectively. Denote the angular velocities of the body frame by $\Omega = \begin{bmatrix}p & q & r\end{bmatrix}^\top \in \mathbb{R}^3$; then, by defining $\mathbf{M} := \begin{bmatrix}M_x & M_y & M_z\end{bmatrix}^\top \in \mathbb{R}^3$ with M_x, M_y and M_z denoting the applied roll, pitch, and yaw moments, respectively, the angular dynamics can be represented by

$$J_{\mathcal{B}}\dot{\Omega} = -\Omega \times [J_{\mathcal{B}}\Omega] + \mathbf{M}, \tag{2}$$

with $J_{\mathcal{B}} \in \mathbb{R}^{3\times 3}$ denoting the moment of inertia of the assembly. Using

$$S(\boldsymbol{\varphi}) = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\theta & 0 & c\theta c\phi \end{bmatrix}, \tag{3}$$

we can transform between the angular velocities in the body frame and the angular velocities in the world frame by the relation:

$$\Omega = S(\boldsymbol{\varphi})\dot{\boldsymbol{\varphi}}. \tag{4}$$

We now consider model equations having multiple modules in the assembly. Let $\mathbf{p}_j^{\mathcal{B}} = \begin{bmatrix}x_j^{\mathcal{B}} & y_j^{\mathcal{B}} & z_j^{\mathcal{B}}\end{bmatrix}^\top \in \mathbb{R}^3$ be the coordinates of the center of mass of module j in the body frame. In this paper, we assume that the modules are connected together rigidly, and that each module's frame is parallel to the body frame. Then we can write the total thrust and moments exerted on the whole assembly in terms of module individual thrusts and moments in a simple manner; with T_j as the total thrust produced by module j, and $M_{j,x}$, $M_{j,y}$ and $M_{j,z}$ as the roll, pitch and yaw moments produced by module j, respectively, we have

$$\begin{aligned} &T = \textstyle\sum_{j=1}^N T_j, \quad M_x = \sum_{j=1}^N M_{j,x} + y_j^{\mathcal{B}} T_j, \\ &M_y = \textstyle\sum_{j=1}^N M_{j,y} - x_j^{\mathcal{B}} T_j, \quad M_z = \sum_{j=1}^N M_{j,z}; \end{aligned} \tag{5}$$

notice here that this relation allows modules to be on different planes in the assembly [13]. Also in this rigid assembly, we can represent the orientation of a module j with respect to the world frame by the same orientation of the whole assembly

$$\boldsymbol{\varphi}_j := \begin{bmatrix}\phi_j & \theta_j & \psi_j\end{bmatrix}^\top = \begin{bmatrix}\phi & \theta & \psi\end{bmatrix}^\top, \tag{6}$$

and similarly for the angular velocities,

$$\Omega_j := \begin{bmatrix}p_j & q_j & r_j\end{bmatrix}^\top = \begin{bmatrix}p & q & r\end{bmatrix}^\top. \tag{7}$$

Then, if we denote the coordinates of the center of mass of module j in the world frame by $\mathbf{p}_j = \begin{bmatrix}x_j & y_j & z_j\end{bmatrix}^\top \in \mathbb{R}^3$, then it is easy to see that

$$\mathbf{p}_j = \mathbf{p} + R(\boldsymbol{\varphi})\mathbf{p}_j^{\mathcal{B}}, \tag{8}$$

and so

$$\dot{\mathbf{p}}_j = \dot{\mathbf{p}} + \Omega \times [R(\boldsymbol{\varphi})\mathbf{p}_j^{\mathcal{B}}]. \tag{9}$$

B. The Control Problem

In this paper we want to control the assembly under uncertainty and in a decentralized manner. We want to stabilize the system and drive the position of the assembly $\mathbf{p}$ and its yaw angle ψ to some desired position $\mathbf{p}_d$ and angle ψ_d, respectively; we want to achieve this using a decentralized control law, i.e. module j should only use measurements available at module j in the control. Moreover, we are considering the case when module j is uncertain about its own position in the assembly structure, so this means there is uncertainty about the value of $\mathbf{p}_j^{\mathcal{B}}$; under such uncertainty, the control of the whole system is more complicated compared to the case when we know exactly the position of module j in the assembly. To this end, we consider that for each j, we have the measurements $\mathbf{y}_j = \begin{bmatrix}\mathbf{p}_j^\top & \dot{\mathbf{p}}_j^\top & \boldsymbol{\varphi}_j^\top & \Omega_j^\top\end{bmatrix}^\top \in \mathbb{R}^{12}$; this means we assume that each module knows its own position, orientation, and linear and angular velocities, but not those of the center of mass of the whole assembly. We consider for module j inputs available for control to be $\mathbf{u}_j = \begin{bmatrix}T_j & M_{x,j} & M_{y,j} & M_{z,j}\end{bmatrix}^\top \in \mathbb{R}^4$. We are considering here a challenging problem where we want to control the system without the availability of a common measurement of the center of mass of the assembly, without inter-module communication, and with uncertainty of each module's position in the said assembly.

Let $\Theta_j \in \mathbb{R}^p, j \in \overline{Q}, Q \in \mathbb{N}$, be a vector that contains the uncertain parameters of the system, and it belongs to the finite set of possible parameters,

$$\mathcal{P} := \{\Theta_1, \Theta_2, \ldots, \Theta_Q\},$$

which is clearly compact; in this paper, we have $\Theta_j = \mathbf{p}_j^{\mathcal{B}} \in \mathbb{R}^3$ and $Q = N$. With state vector $\mathbf{x} \in \mathbb{R}^{12}$ defined by

$$\begin{aligned}\mathbf{x} &= \begin{bmatrix}x & y & z & \dot{x} & \dot{y} & \dot{z} & \phi & \theta & \psi & p & q & r\end{bmatrix}^\top \\ &= \begin{bmatrix}\mathbf{p}^\top & \dot{\mathbf{p}}^\top & \boldsymbol{\varphi}^\top & \Omega^\top\end{bmatrix}^\top,\end{aligned}$$

it is convenient for control to write the dynamics of the system as follows: from (1), (2), (4), (5), (8), (9), (6) and (7), we obtain

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\left(\sum_{j=1}^N \overline{B}(\Theta_j)\mathbf{u}_j\right), \tag{10a}$$

$$\mathbf{y}_j = h(\mathbf{x}, \Theta_j), \quad j = 1, 2, \ldots, N, \tag{10b}$$

with the functions $f : \mathbb{R}^{12} \to \mathbb{R}^{12}$, $g : \mathbb{R}^{12} \to \mathbb{R}^{12\times 4}$, $\overline{B} : \mathcal{P} \to \mathbb{R}^{4\times 4}$ and $h : \mathbb{R}^{12} \times \mathcal{P} \to \mathbb{R}^{12}$ defined by

$$f(\mathbf{x}) := \begin{bmatrix} \dot{\mathbf{p}} \\ -g\mathbf{e}_3 \\ S(\boldsymbol{\varphi})^\top \Omega \\ J_{\mathcal{B}}^{-1}\left(-\Omega \times [J_{\mathcal{B}}\Omega]\right) \end{bmatrix}, \quad g(\mathbf{x}) := \begin{bmatrix} \mathbf{0}_{3\times 4} \\ [\frac{1}{m_{\mathcal{B}}} R(\boldsymbol{\varphi})\mathbf{e}_3 \quad \mathbf{0}_{3\times 3}] \\ \mathbf{0}_{3\times 4} \\ [\mathbf{0}_{3\times 1} \quad I_3] \end{bmatrix},$$

$$\overline{B}(\Theta_j) := \begin{bmatrix} 1 & 0 & 0 & 0 \\ y_j^{\mathcal{B}} & 1 & 0 & 0 \\ -x_j^{\mathcal{B}} & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad h(\mathbf{x}, \Theta_j) := \begin{bmatrix} \mathbf{p} + R(\boldsymbol{\varphi})\mathbf{p}_j^{\mathcal{B}} \\ \dot{\mathbf{p}} + \Omega \times [R(\boldsymbol{\varphi})\mathbf{p}_j^{\mathcal{B}}] \\ \boldsymbol{\varphi} \\ \Omega \end{bmatrix}.$$

We now discuss an assumption on the controllability of the system. Define

$$\overline{\mathbf{u}} := \sum_{j=1}^{N} \overline{B}(\Theta_j)\mathbf{u}_j, \tag{11}$$

so we can write (10a) as follows:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\overline{\mathbf{u}}; \tag{12}$$

notice that (11) is equivalent to $\overline{\mathbf{u}} = \begin{bmatrix} T & M_x & M_y & M_z \end{bmatrix}^\top \in \mathbb{R}^4$, i.e. $\overline{\mathbf{u}}$ represents the total thrust and moments produced by the whole system. In this paper we **assume** the existence of a controller that stabilizes the system in (12): there exists a function $K : \mathbb{R}^{12} \to \mathbb{R}^4$ such that if we set $\overline{\mathbf{u}} = K(\mathbf{x})$ in (12), then the system $\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})K(\mathbf{x})$ has an exponentially stable equilibrium point at $\begin{bmatrix} \mathbf{p}_d^\top & \mathbf{0}_{1\times 5} & \psi_d & \mathbf{0}_{1\times 3} \end{bmatrix}^\top$. This assumption says that we have at hand a controller that stabilizes the aerial rigid body provided that we have the full state $\mathbf{x}$ available for feedback. This assumption covers a wide range of hovering based controllers, normally used to control multirotors, e.g. see [12]. Of course in this paper we assume we do not have the full state for feedback, and we control thrust and moments of the whole body only indirectly and under uncertainty about the assembly structure.

C. A Decentralized Control Law

By assumption, there exists a stabilizing controller $\overline{\mathbf{u}} = K(\mathbf{x})$; now rewrite (11) as

$$\underbrace{\begin{bmatrix} \overline{B}(\Theta_1) & \overline{B}(\Theta_2) & \cdots & \overline{B}(\Theta_N) \end{bmatrix}}_{=:\widetilde{B}} \underbrace{\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_N \end{bmatrix}}_{=:\mathbf{u}} = \overline{\mathbf{u}};$$

we want to find values for $\mathbf{u}_j$, $j = 1, 2, \ldots, N$, satisfying the above. We observe that $\widetilde{B} \in \mathbb{R}^{4\times 4N}$ and $\mathbf{u} \in \mathbb{R}^{4N}$; so for the case when $N > 1$ we have an *under-determined* linear system with four equations but $4N$ unknowns; hence, we possibly have infinite number of solutions for $\mathbf{u}$. A solution can be obtained by calculating the pseudoinverse of $\widetilde{B}$; indeed a solution for $\mathbf{u}$ is found by calculating

$$\mathbf{u} = \widetilde{B}^\top \left(\widetilde{B}\widetilde{B}^\top\right)^{-1} \overline{\mathbf{u}}. \tag{13}$$

The solution in (13) is just one possible solution; actually this is the solution of the optimization problem $\arg\min_{\mathbf{u}}\{\|\mathbf{u}\| : \widetilde{B}\mathbf{u} = \overline{\mathbf{u}}\}$; one may choose different costs other than $\|\mathbf{u}\|$ to achieve alternative control actuation objectives. Observe that by the structure of $\widetilde{B}$ we can write (13) in a distributed manner and substitute in $\overline{\mathbf{u}} = K(\mathbf{x})$ yielding:

$$\mathbf{u}_j = \overline{B}(\Theta_j)^\top \left(\widetilde{B}\widetilde{B}^\top\right)^{-1} K(\mathbf{x}), \quad j = 1, 2, \ldots, N. \tag{14}$$

It is easy to check that the matrix $\widetilde{B}\widetilde{B}^\top$ is non-singular; observe that the matrix $\left(\widetilde{B}\widetilde{B}^\top\right)^{-1}$ is static, and merely a function of the geometry of the assembly, i.e. it is a function of the whole set $\mathcal{P}$ which is assumed to be known to all modules.

Notice that for the problem considered here the control law (14) is not yet suitable as $\mathbf{x}$ is not available to the modules. But we see from (10b) that we can easily obtain an inverse function transforming between module measurements and the state of system: we define the function $h^{-1} : \mathbb{R}^{12} \times \mathcal{P} \to \mathbb{R}^{12}$ by

$$h^{-1}(\mathbf{y}_j, \hat{\Theta}) := \begin{bmatrix} \mathbf{p}_j - R(\boldsymbol{\varphi}_j)\hat{\Theta} \\ \dot{\mathbf{p}}_j - \Omega_j \times [R(\boldsymbol{\varphi}_j)\hat{\Theta}] \\ \boldsymbol{\varphi}_j \\ \Omega_j \end{bmatrix}, \quad j \in \overline{N},\ \hat{\Theta} \in \mathcal{P}. \tag{15}$$

It is obvious that

$$\mathbf{x} = h^{-1}(\mathbf{y}_j, \Theta_j), \quad j \in \overline{N},$$

so now we re-write (14) in a fully decentralized form:

$$\mathbf{u}_j = \overline{B}(\Theta_j)^\top \left(\widetilde{B}\widetilde{B}^\top\right)^{-1} K\big(h^{-1}(\mathbf{y}_j, \Theta_j)\big), \quad j = 1, 2, \ldots, N. \tag{16}$$

In the next section we present the main contribution of the paper. We provide an approach to control the system when modules are uncertain about their own location in the assembly structure, or in other words, Θ_j is unknown.

III. The Adaptive Switching Controller

For ease of notation, let us define for any $p \in \overline{N}$ and $q \in \overline{N}$ the function $F : \mathbb{R}^{12} \times \mathcal{P} \to \mathbb{R}^4$ by

$$F(\mathbf{y}_p, \Theta_q) := \overline{B}(\Theta_q)^\top \left(\widetilde{B}\widetilde{B}^\top\right)^{-1} K\big(h^{-1}(\mathbf{y}_p, \Theta_q)\big); \tag{17}$$

in the ideal case when there is no uncertainty about module j's own perception in the assembly, $q = j$ when $p = j$; however, here Θ_j is unknown but belongs to $\mathcal{P}$.

A. Switching Control Law

Let us now define for each $j \in \overline{N}$ the signal $\sigma_j : \mathbb{R}^+ \to \overline{N}$ to denote the index with which module j perceives itself to be at any given point in time. So under uncertainty we apply the following control:

$$\mathbf{u}_j = F(\mathbf{y}_j, \Theta_{\sigma_j}), \quad j = 1, 2, \ldots, N. \tag{18}$$

However, under uncertainty a module j does not know if their perception is correct or not; hence, we employ here an adaptive mechanism to set $\sigma_j, j = 1, 2, \ldots, N$, which we present in the following.

Now define $\boldsymbol{\sigma} := (\sigma_1, \sigma_2, \ldots, \sigma_N)$; we see that the value of $\boldsymbol{\sigma}$ belongs to the set of all N-tuple permutations of the set $\{1, 2, \ldots, N\}$. For example, for $N = 3$, we have $\boldsymbol{\sigma} \in \big\{(1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2), (3,2,1)\big\}$. Now, for an assembly consisting of N modules, let $\mathcal{I}$ denote the set of all possible N-tuple permutations of the set $\{1, 2, \ldots, N\}$. So, each $\boldsymbol{i} \in \mathcal{I}$ corresponds to a candidate collection of module perceptions about their own position in the assembly, which we want to choose correctly. We let $\boldsymbol{i}^* \in \mathcal{I}$ to label the correct permutation; indeed the modules labels of $1, 2, \ldots$ and N are arbitrary in the first place, so we let $\boldsymbol{i}^* := (1, 2, \ldots, N)$; of course, this correct choice is unknown to the modules.

B. Multi-Estimators

Similar, but not identical, to the approach used in [8], we now construct systems that act as estimators for each candidate $\boldsymbol{i} \in \mathcal{I}$. This eventually will help us set $\boldsymbol{\sigma}$.

Before proceeding, partition each $\boldsymbol{i} \in \mathcal{I}$ naturally as $\boldsymbol{i} =: (i_1, i_2, \ldots, i_N)$, and define

$$\hat{\mathbf{x}}_{\boldsymbol{i},j} := h^{-1}(\mathbf{y}_j, \Theta_{i_j}), \quad j \in \overline{N}; \tag{19}$$

note that $\hat{\mathbf{x}}_{\boldsymbol{i}^*,j} = \mathbf{x}$ for all $j \in \overline{N}$. Then with $\lambda_e < 0$, for each $\boldsymbol{i} \in \mathcal{I}$, construct the following N systems: with $\mathbf{x}^e_{\boldsymbol{i},j}(0) = \mathbf{0}_{12\times 1}$,

$$\begin{aligned}\dot{\mathbf{x}}^e_{\boldsymbol{i},j} &= \lambda_e(\mathbf{x}^e_{\boldsymbol{i},j} - \hat{\mathbf{x}}_{\boldsymbol{i},j}) + \\ &f(\hat{\mathbf{x}}_{\boldsymbol{i},j}) + g(\hat{\mathbf{x}}_{\boldsymbol{i},j})\left(\sum_{q=1}^{N} \overline{B}(\Theta_{i_q})F(\mathbf{y}_q, \Theta_{\sigma_q})\right), \quad j \in \overline{N}.\end{aligned} \tag{20}$$

Associated with these systems, we define prediction errors for each $\boldsymbol{i} \in \mathcal{I}$ by

$$e_{\boldsymbol{i},j} := \mathbf{x}^e_{\boldsymbol{i},j} - \hat{\mathbf{x}}_{\boldsymbol{i},j}, \quad j \in \overline{N}. \tag{21}$$

It is easy to verify that

$$\dot{e}_{\boldsymbol{i}^*,j} = \lambda_e e_{\boldsymbol{i}^*,j} \quad \text{for all } j \in \overline{N};$$

this means that the prediction errors associated with the correct choice $\boldsymbol{i}^*$ goes to zero exponentially fast; this is true irrespective of the control input applied to the system at any given point in time. With this fact in mind, we show next how to choose $\boldsymbol{\sigma}$ from time to time.

C. The Switching Algorithm

We update $\boldsymbol{\sigma}$ only every $\tau_c > 0$, so the switching algorithm is a process that produces its output at discrete time instants. To this end, define the sequence of switching times by $\hat{t}_k := k\tau_c$, $k \in \mathbb{Z}^+$; so here the signal $\boldsymbol{\sigma}$ is a piecewise constant signal of the form

$$\boldsymbol{\sigma}(t) = \boldsymbol{\sigma}(\hat{t}_k), \quad t \in [\hat{t}_k, \hat{t}_{k+1}), \ \ k \in \mathbb{Z}^+. \tag{22}$$

With the design parameter $\tau_c > 0$, we guarantee that there are no infinite number of switches during any finite time intervals; in practical situations, this means that a large enough τ_c avoids chattering while also achieves the desired closed-loop behavior. More discussion about this parameter is provided in [14]. We present now the algorithm to compute $\boldsymbol{\sigma}(\hat{t}_k)$ for $k = 0, 1, 2, \ldots$ and so on.

To proceed, for each $\boldsymbol{i} \in \mathcal{I}$ we define a performance signal $\mathcal{J}_{\boldsymbol{i}} : \{\hat{t}_0, \hat{t}_1, \hat{t}_2, \cdots\} \to \mathbb{R}^+$ by

$$\mathcal{J}_{\boldsymbol{i}}(\hat{t}_k) := \int_{\hat{t}_k}^{\hat{t}_{k+1}} \left(\sum_{j=1}^{N} \|e_{\boldsymbol{i},j}(\tau)\|^2\right) d\tau, \quad k \in \mathbb{Z}^+. \tag{23}$$

This quantity measures how large are the errors in the prediction for candidate $\boldsymbol{i}$ in between switching instants; the errors $e_{\boldsymbol{i}^*,j} \to 0$, $j \in \overline{N}$ as $t \to \infty$, so we expect that the candidate with least amount of errors to be the best choice. To this end, in the following we discuss the switching algorithm used in the present paper. This algorithm was used in [18] and [19] where the problem of tracking for a possibly non-minimum phase plant with a compact uncertainty set is considered. At each switching time $\hat{t}_k$ we have an admissible index set $\mathcal{I}(\hat{t}_k)$: we initialize $\mathcal{I}(\hat{t}_0) = \mathcal{I}$, and we obtain $\mathcal{I}(\hat{t}_{k+1})$ from $\mathcal{I}(\hat{t}_k)$ by removing all $\bar{\jmath} \in \mathcal{I}(\hat{t}_k)$ satisfying $\mathcal{J}_{\boldsymbol{\sigma}(\hat{t}_k)}(\hat{t}_k) \leq \mathcal{J}_{\bar{\jmath}}(\hat{t}_k)$, i.e. we keep all candidates in the admissible index set for which the performance signal is "better" than the one we are currently using; clearly $\bar{\jmath} = \boldsymbol{\sigma}(\hat{t}_k)$ satisfies this bound, but more $\bar{\jmath}$'s may as well; if this results in $\mathcal{I}(\hat{t}_{k+1})$ being empty, then we *reset* $\mathcal{I}(\hat{t}_{k+1})$ to be $\mathcal{I}$. This switching algorithm is summarized as follows: with $\boldsymbol{\sigma}(\hat{t}_0) \in \mathcal{I}$ and $\mathcal{I}(\hat{t}_0) = \mathcal{I}$:

$$\check{\mathcal{I}}(\hat{t}_k) = \left\{\boldsymbol{i} \in \mathcal{I} \ : \ \mathcal{J}_{\boldsymbol{i}}(\hat{t}_k) < \mathcal{J}_{\boldsymbol{\sigma}(\hat{t}_k)}(\hat{t}_k)\right\}, \tag{24a}$$

$$\mathcal{I}(\hat{t}_{k+1}) = \begin{cases}\mathcal{I} & \text{if } \mathcal{I}(\hat{t}_k) \cap \check{\mathcal{I}}(\hat{t}_k) = \varnothing \\ \mathcal{I}(\hat{t}_k) \cap \check{\mathcal{I}}(\hat{t}_k) & \text{otherwise,}\end{cases} \tag{24b}$$

$$\boldsymbol{\sigma}(\hat{t}_{k+1}) \in \underset{\boldsymbol{i} \in \mathcal{I}(\hat{t}_{k+1})}{\operatorname{argmin}} \ \mathcal{J}_{\boldsymbol{i}}(\hat{t}_k), \quad k \in \mathbb{Z}^+. \tag{24c}$$

Definition 1. We define the **index set reset times** as those $\hat{t}_k$, $k \in \mathbb{Z}^+$, for which $\mathcal{I}(\hat{t}_k) = \mathcal{I}$.

The algorithm in (24) has the following desirable property.

Lemma 1 ([19]). *If $\hat{t}_{\underline{k}}$ and $\hat{t}_{\bar{k}}$ are two consecutive index set reset times, then there exists a $k^* \in \{\underline{k}, \underline{k}{+}1, \ldots, \bar{k}{-}2, \bar{k}{-}1\}$ such that:*

$$\mathcal{J}_{\boldsymbol{\sigma}(\hat{t}_{k^*})}(\hat{t}_{k^*}) \leq \mathcal{J}_{\boldsymbol{i}^*}(\hat{t}_{k^*}).$$

Remark 1. *The result in Lemma 1 holds even if there are bounded noise/disturbance entering the system, and/or unmodelled dynamics. This is due to the logic of the algorithm in* (24) *irrespective of the definition of $\mathcal{J}_{\boldsymbol{i}}(\cdot)$; for more insight, see the proof of Lemma 1 in [19].*

In Lemma 1 we do not make any claim that $\boldsymbol{\sigma}(t) = \boldsymbol{i}^*$ any time; it only makes an indirect statement about the size of the prediction errors. It turns out that this is enough to achieve the desired closed-loop behavior. We show this by simulation; illustrative simulation examples, that show the robustness of the approach, are provided in the next section.

Remark 2. *The control law in* (18) *applied in module j is a function of $\mathbf{y}_j$ and σ_j; observe that the switching signal σ_j is the output of the switching mechanism in* (19)–(24) *which requires all of the measurements $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N$, rendering the whole controller not fully decentralized. However, we would like to emphasize that this is done without the knowledge of the state $\mathbf{x}$ and without the knowledge of the correct assembly configuration. Apart from their own measurements, each module j receives only updates for the value of σ_j, and only every τ_c time units.*

Remark 3. *We see that the number of candidate choices in $\mathcal{I}$ is $N!$, which can be large for a larger number of modules in an assembly. However, observe that the switching mechanism in* (19)–(24) *contains only simple computations and logic, which is manageable for a not very large assembly. Also using a similar approach to that found in [1], the systems in* (20) *may be replaced with a single global system that generates a state shared by all of candidates $\boldsymbol{i}$ to compute the errors $e_{\boldsymbol{i},j}$.*

IV. A Case Study: The Tetrahedron Assembly

In this case study, the example of the fractal tetrahedron Unmanned Air System (UAS) assembly is considered. The fractal tetrahedron UAS assembly is a modular robotic system whose modules are identical quadrotorcraft with a regular tetrahedron frame, allowing them to combine in groups of four by forming a new tetrahedron of side length twice the one of the modules. This assembling process, inspired by the Sierpinski tetrahedron, can be repeated with assemblies of same size, allowing the creation in theory of arbitrary large multirotor vehicles. The motivation behind this particular tetrahedral shape comes from two reasons; the first reason is that it results in a highly symmetric three-

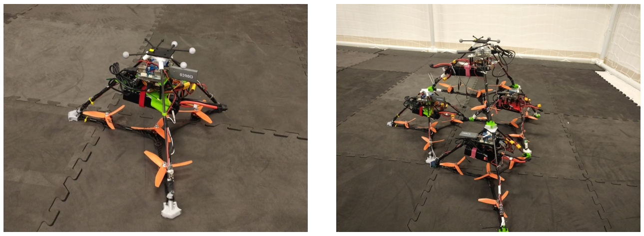

Fig. 1. A single module (left); the four-module assembly (right).

dimensional structure, providing rigidity in all directions and limiting deformations under stress; the second reason is that the assembling rule ensures that no rotor is positioned under another rotor, limiting rotor wake interactions that would otherwise decrease the performance of the UAS assembly. For more about the fractal tetrahedron assembly, see [5].

The case study focuses on the first generation of the fractal tetrahedron assembly, which is constituted of four independent modules, so $N = 4$; the modules are identical, and each have a mass of m and a moment of inertia $J \in \mathbb{R}^{3\times 3}$. See Figure 1 showing the system being developed at the Robotics, Intelligent Systems, and Control (RISC) Lab at King Abdullah University of Science and Technology (KAUST). Each module $j \in \{1, 2, 3, 4\}$ generates the moments $\mathbf{M}_j = \begin{bmatrix} M_{j,x} & M_{j,y} & M_{j,z} \end{bmatrix}^\top$ and thrust T_j by changing the angular velocities $\omega_{j,i}, i \in \{1, 2, 3, 4\}$ of its four propellers; the moments and thrusts are given by

$$\mathbf{M}_j = \sum_{i=1}^{4} \left(k_T \mathbf{r}_i \times \mathbf{e}_3 + k_M (-1)^i \mathbf{e}_3 \right) \omega_{j,i}^2, \quad T_j = k_T \sum_{i=1}^{4} \omega_{j,i}^2,$$

where k_T is the rotor thrust constant, k_M is the rotor drag constant, and $\mathbf{r}_i$ is the position of rotor i relative to the module's center of mass. Notice that the relationship between a module's generated moments and thrust and its rotors' angular velocities is one-to-one, justifying using $\begin{bmatrix} T_j & M_{j,x} & M_{j,y} & M_{j,z} \end{bmatrix}^\top$ directly as the control input of module j. In practice, each rotor's angular velocity is bounded between the real positive constants $\omega_{\min}$ and $\omega_{\max}$. For the system to be stabilizable, it must be able to generate enough thrust to compensate for its weight, that is $4 k_T \omega_{\max}^2 > mg$.

If each module has an edge of length L, then define $R := \frac{1}{2}\sqrt{\frac{3}{2}} L$, so to have in this case the parameter uncertainty set of

$$\mathcal{P} := \left\{ \begin{bmatrix} 0 \\ R\sqrt{2} \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{R}{\sqrt{2}} \\ -\frac{R}{\sqrt{2}} \\ 0 \end{bmatrix}, \begin{bmatrix} -\frac{R}{\sqrt{2}} \\ -\frac{R}{\sqrt{2}} \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ R \end{bmatrix} \right\},$$

which contains module candidate positions relative to the assembly's center of mass. We calculate the mass and moment of inertia of the four-module tetrahedral assembly as follows [5]: $m_{\mathcal{B}} = 4m$ and $J_{\mathcal{B}} = 4(J - \frac{2}{3} m R^2 I_3)$; we assume that the axes of the module frame are so that $J = \operatorname{diag}(J_{xx}, J_{yy}, J_{zz})$, with $J_{xx}, J_{yy}, J_{zz} \geq 0$.

A. The Control Law

For each $j \in \overline{N}$, we apply the control law (18) with the switching signals $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_N)$ chosen by the switching mechanism (19)–(24). We now present the control law $K(\cdot)$ as required (see discussion in Section II-B). Before proceeding, for each $\boldsymbol{i} = (i_1, i_2, \ldots, i_N) \in \mathcal{I}$, and each $j \in \overline{N}$, partition $\hat{\mathbf{x}}_{\boldsymbol{i},j}$, defined in (19), naturally by $\hat{\mathbf{x}}_{\boldsymbol{i},j} =: \begin{bmatrix} \hat{\mathbf{p}}_{i_j,j}^\top & \dot{\hat{\mathbf{p}}}_{i_j,j}^\top & \boldsymbol{\varphi}_{i_j,j}^\top & \Omega_{i_j,j}^\top \end{bmatrix}^\top$. Now we define the function $K(h^{-1}(\mathbf{y}_p, \Theta_q))$ used in inside the definition of $F(\cdot,\cdot)$ in (17). Here we use a control law used for hovering control adapted from that in [13]. First, define the target trajectories for the modules as follows: for each $j \in \{1, 2, 3, 4\}$, $\ddot{\mathbf{p}}_{T,j} := K_P(\mathbf{p}_d - \hat{\mathbf{p}}_{\sigma_j,j}) - K_D \dot{\hat{\mathbf{p}}}_{\sigma_j,j}$, and $\dddot{\mathbf{p}}_{T,j} := -K_P \dot{\hat{\mathbf{p}}}_{\sigma_j,j} - K_D \ddot{\mathbf{p}}_{T,j}$, with K_P and K_D, 3×3 diagonal matrices with positive diagonal entries; with $\ddot{\mathbf{p}}_{T,j} = [\ddot{x}_{T,j} \;\; \ddot{y}_{T,j} \;\; \ddot{z}_{T,j}]^\top$ and $\dddot{\mathbf{p}}_{T,j} = [\dddot{x}_{T,j} \;\; \dddot{y}_{T,j} \;\; \dddot{z}_{T,j}]^\top$,

$$\varphi_{T,j} := \begin{bmatrix} \frac{1}{g}(\ddot{x}_{T,j} s\psi_d - \ddot{y}_{T,j} c\psi_d) \\ \frac{1}{g}(\ddot{x}_{T,j} c\psi_d + \ddot{y}_{T,j} s\psi_d) \\ \psi_d \end{bmatrix}, \quad \Omega_{T,j} := \begin{bmatrix} \frac{1}{g}(\dddot{x}_{T,j} s\psi_d - \dddot{y}_{T,j} c\psi_d) \\ \frac{1}{g}(\dddot{x}_{T,j} c\psi_d + \dddot{y}_{T,j} s\psi_d) \\ 0 \end{bmatrix};$$

then set the desired thrust and moments for each module $j \in \{1, 2, 3, 4\}$ to be:

$$\begin{aligned} T_j^* &:= Nm(\ddot{z}_{T,j} + g), \\ \mathbf{M}_j^* &:= K_{\varphi,P}(\varphi_{T,j} - \varphi_j) + K_{\varphi,D}(\Omega_{T,j} - \Omega_j), \end{aligned}$$

with $K_{\varphi,P}$ and $K_{\varphi,D}$, 3×3 diagonal matrices with positive diagonal entries. Finally, we set the applied control input:

$$\mathbf{u}_j = \overline{B}(\Theta_{\sigma_j})^\top \left(\widetilde{B} \widetilde{B}^\top \right)^{-1} \begin{bmatrix} T_j^* \\ \mathbf{M}_j^* \end{bmatrix}, \quad j = 1, 2, 3, 4.$$

B. Simulation 1

In this simulation, we have module specifications as $m = 250\,\mathrm{g}$, $J_{xx} = J_{yy} = 5 \times 10^{-2}\,\mathrm{kg}\cdot\mathrm{m}^2$, $J_{zz} = 5 \times 10^{-1}\,\mathrm{kg}\cdot\mathrm{m}^2$ and $L = 50\,\mathrm{cm}$. We also add drag forces to the model; to this end, in this simulation, Equation (10a) becomes

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x}) \left(\sum_{j=1}^{N} \overline{B}(\Theta_j) \mathbf{u}_j \right) - \frac{1}{2} \begin{bmatrix} \mathbf{0}_{3\times 1} \\ R(\boldsymbol{\varphi}) \begin{bmatrix} \dot{x}^2 \\ \dot{y}^2 \\ \dot{z}^2 \end{bmatrix} \\ p^2 \\ q^2 \\ r^2 \\ \mathbf{0}_{3\times 1} \end{bmatrix}.$$

We set the controller gains $K_P = K_D = I_3$, $K_{P,\varphi} = 4I_3$, and $K_{D,\varphi} = 2I_3$. We set $\lambda_e = -10$ and $\tau_c = 50\,\mathrm{msec}$ in the switching mechanism. The desired position and yaw angle are $\mathbf{p}_d = \begin{bmatrix} 0 & -5 & 10 \end{bmatrix}^\top$ and $\psi_d = -\frac{\pi}{4}$, respectively. The system is initialized at position $\mathbf{p}(0) = \begin{bmatrix} 0 & 0 & 5 \end{bmatrix}^\top$, zero orientations, and zero linear and angular velocities. Set $\boldsymbol{\sigma}(0) = (4, 3, 2, 1)$. We simulate both cases of when there is uncertainty and when there is none. The result is in Figure 2. With a demanding maneuver for the assembly, you can see that the system achieves the control objective nevertheless, with the adaptive algorithm settling on the correct choice quickly.

C. Simulation 2

Here, we consider noise added to modules' measurements $\mathbf{y}_j$. In particular, we add sinusoidal noise to the altitude z measurement, so in this simulation we set Equation (8) to be

$$\mathbf{p}_j = \mathbf{p} + R(\boldsymbol{\varphi}) \mathbf{p}_j^{\mathcal{B}} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{10}\cos(2\pi j t) \end{bmatrix}, \quad j = 1, 2, 3, 4.$$

We use the same module specifications for m, J, and L as in Simulation 1. We apply the same controller with the same gains, the same λ_e, and the same $\tau_c = 50\,\mathrm{msec}$. The desired position and yaw angle are $\mathbf{p}_d = \begin{bmatrix} 0 & 0 & 5 \end{bmatrix}^\top$ and $\psi_d = 0$, respectively. The system is initialized at position $\mathbf{p}(0) = \begin{bmatrix} 5 & 0 & 10 \end{bmatrix}^\top$, zero orientations, and zero linear and angular velocities. Set $\boldsymbol{\sigma}(0) = (3, 4, 1, 2)$. Similarly, we

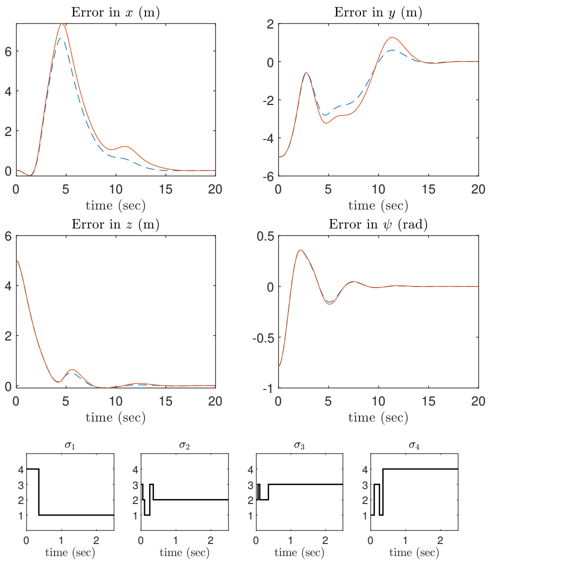

Fig. 2. This shows Simulation 1 results. The top four plots show the errors in the position and the yaw angle when the adaptive switching controller is used (solid), and when the ideal controller is used without uncertainty (dashed). The bottom four plots show the first 2.5 seconds of the switching signal $\boldsymbol{\sigma}$.

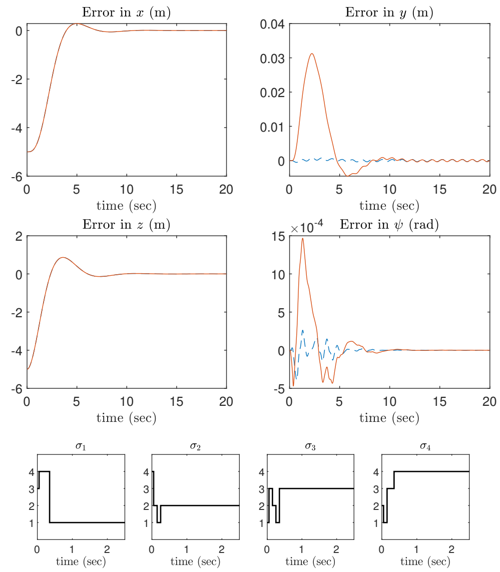

Fig. 3. This shows Simulation 2 results. The top four plots show the errors in the position and the yaw angle when the adaptive switching controller is used (solid), and when the ideal controller is used without uncertainty (dashed). The bottom four plots show the first 2.5 seconds of the switching signal $\boldsymbol{\sigma}$.

simulate both cases of when there is uncertainty and when there is none. The result is in Figure 3. The measurement noise has some effect on the performance but not significant.

V. Summary and Conclusions

We consider the problem of the control of an assembly of aerial vehicles under uncertainty; we assume that positions of the vehicle modules in the assembly structure is unknown. An adaptive switching controller is proposed; we consider a decentralized control law where each module only uses its own measurements for feedback and a switching signal which is set to determine from time to time which parameters to plug into the corresponding control law. Given a stabilizing controller for the case when there is no uncertainty, we show that the proposed adaptive approach achieves the control objective. Illustrative simulation examples are provided where a novel assembly structure is considered, namely the tetrahedron fractal assembly.

We are working on validating the proposed approach experimentally. In the future, we want to extend the approach to include other types of uncertainty; for example, after a mid-air assembly, the number of modules in an assembly, N, is itself possibly unknown, so we would like to incorporate this into the adaptive approach.

References

[1] A. P. Aguiar and J. P. Hespanha, "Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty," *IEEE Trans. Autom. Control*, vol. 52, no. 8, pp. 1362–1379, 2007.

[2] P. Chand, S. Veer, and I. Poulakakis, "An adaptive supervisory control approach to dynamic locomotion under parametric uncertainty," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 2443–2449.

[3] W.-C. Chang and A. S. Morse, "Control of a rigid robot using an uncalibrated stereo vision system," in *Proc. Amer. Control Conf.*, vol. 3, 1997, pp. 2103–2107.

[4] P. Culbertson and M. Schwager, "Decentralized adaptive control for collaborative manipulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 278–285.

[5] K. Garanger, J. Epps, and E. Feron, "Modeling and experimental validation of a fractal tetrahedron uas assembly," in *Proc. IEEE Aerospace Conf.*, 2020, pp. 1–11.

[6] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and Control*. New York, NY, USA: Dover Publications, Inc., 1984.

[7] J. P. Hespanha, D. Liberzon, and A. S. Morse, "Overcoming the limitations of adaptive control by means of logic-based switching," *Systems & Control Letters*, vol. 49, no. 1, pp. 49–65, 2003.

[8] J. P. Hespanha and A. S. Morse, "Supervision of families of nonlinear controllers," in *Proc. 35th IEEE Conf. Decision and Control*, vol. 4, 1996, pp. 3772–3773 vol.4.

[9] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1995.

[10] S. Kernbach, *Structural self-organization in multi-agents and multi-robotic systems*. Logos Verlag Berlin GmbH, 2008.

[11] H. Lee and H. J. Kim, "Constraint-based cooperative control of multiple aerial manipulators for handling an unknown payload," *IEEE Trans. Industr. Inform.*, vol. 13, no. 6, pp. 2780–2790, 2017.

[12] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se(3)," in *Proc. IEEE Conf. Decis. Control*, 2010, pp. 5420–5425.

[13] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative grasping and transport using multiple quadrotors," in *Distributed Autonomous Robotic Systems*. Springer, Berlin, Heidelberg, 2013, pp. 545–558.

[14] A. S. Morse, "Supervisory control of families of linear set-point controllers–Part 1: Exact matching," *IEEE Trans. Autom. Control*, vol. 41, no. 10, pp. 1413–1431, 1996.

[15] R. Oung, F. Bourgault, M. Donovan, and R. D'Andrea, "The distributed flight array," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2010, pp. 601–607.

[16] L. E. Parker, D. Rus, and G. S. Sukhatme, "Multiple mobile robot systems," in *Springer Handbook of Robotics*. Springer, 2016, pp. 1335–1384.

[17] D. Saldaña, B. Gabrich, G. Li, M. Yim, and V. Kumar, "Modquad: The flying modular structure that self-assembles in midair," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2018, pp. 691–698.

[18] M. T. Shahab and D. E. Miller, "Adaptive Set-Point Regulation using Multiple Estimators," in *Proc. IEEE Conf. Decis. Control*, Dec. 2019, pp. 84–89.

[19] M. T. Shahab and D. E. Miller, "Asymptotic Tracking and Linear-like Behavior Using Multi-Model Adaptive Control," *IEEE Trans. Autom. Control*, 2021.

[20] J.-J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice hall, Englewood Cliffs, NJ, 1991.

[21] S. Thapa, H. Bai, and J. Á. Acosta, "Cooperative aerial manipulation with decentralized adaptive force-consensus control," *J. Intell. Robot. Syst.*, vol. 97, no. 1, pp. 171–183, 2020.

[22] T.-C. Tsao and M. G. Safonov, "Unfalsified direct adaptive control of a two-link robot arm," *Int. J. Adapt. Control Signal Process.*, vol. 15, no. 3, pp. 319–334, 2001.