

# Finding Optimal Modular Robots for Aerial Tasks

Jiawei Xu and David Saldaña

**Abstract**—Traditional aerial vehicles have limitations in their capabilities due to actuator constraints, such as motor saturation. The hardware components and their arrangement are designed to satisfy specific requirements and are difficult to modify during operation. To address this problem, we introduce a versatile modular multi-rotor vehicle that can change its capabilities by reconfiguration. Our modular robot consists of homogeneous cuboid modules, propelled by quadrotors with tilted rotors. Depending on the number of modules and their configuration, the robot can expand its actuation capabilities. In this paper, we build a mathematical model for the actuation capability of a modular multi-rotor vehicle and develop methods to determine if a vehicle is capable of satisfying a task requirement. Based on this result, we find the optimal configurations for a given task. Our approach is validated in realistic 3D simulations, showing that our modular system can adapt to tasks with varying requirements.

## I. INTRODUCTION

The last decade witnessed the development of multi-rotor vehicles which use thrust forces from the propellers to compensate for gravity and to maneuver. Dynamic models have been proposed in the literature to describe and control the motion of multi-rotor vehicles [1], [2], [3], [4]. Despite their numerous advantages, the lack of versatility remains a significant concern for multi-rotor vehicles. For instance, a light-weight quadrotor carrying a camera has limited potential to increase the payload and transport heavy objects. Furthermore, multi-rotor vehicles with vertical rotors are underactuated and require tilting for horizontal translation. These limitations in strength and actuation arise when the vehicle is designed specifically for certain tasks, thus limiting their versatility.

To address the lack of versatility in multi-rotor designs, researchers have present various approaches. For example, additional vertically-placed rotors in a multi-rotor can increase the strength of the vehicle, enabling it to transport heavier objects [5], [3]. Modular vehicles based on quadrotors have also been developed, where the quadrotors rigidly attach to each other to increase strength and form different shapes on the fly [6]. However, such aerial systems do not increase their actuated degrees of freedom (ADOF) as long as the rotors share the same orientation.

Incorporating modularity in multi-rotor vehicles is a scalable approach to improve their strength and actuation. In [7], the authors present a robot composed of modules actuated by pairs of actively tilting rotors. Our previous work presents H-ModQuad [8], a heterogeneous modular system composed of modules that fly autonomously and change their ADOF by reconfiguration. These approaches improve the versatility of multi-rotor vehicles by expanding their actuation capabilities, *i.e.*, the maximum force and torque they can generate. In a

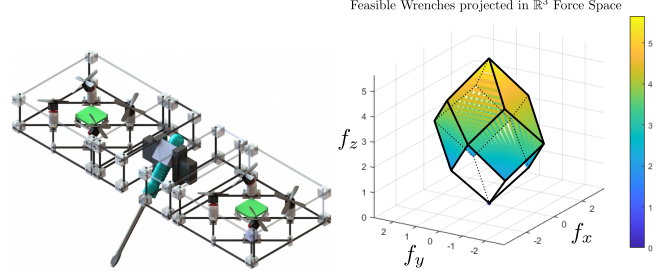


Fig. 1: A modular multi-rotor manipulating a screwdriver. The set of all forces that the vehicle can generate is represented by a polytope in the  $\mathbb{R}^3$  force space.

modular system, it is essential to determine whether there exists a configuration that satisfies the task requirements or not. If so, we need to find an optimal configuration that minimizes the number of modules.

In [9], the authors present a modular flying robot that can reconfigure by tilting the modules inside the structure to adapt to different trajectories; to determine whether a configuration satisfies a task requirement or not, the vehicle evaluates several configurations to track a trajectory in simulation. We propose a geometric approach to solve this problem by modeling the task requirements and the actuation capabilities of a multi-rotor vehicle using polytopes. In recent literature, we notice methods to describe the actuation capability of multi-rotor vehicles [10], [11], [12] that are commonly illustrated as a polytope within which forces and torques are achievable by the vehicles. Researchers also present optimizations for the design of multi-rotor vehicles with performance characteristics, both in theory [13] for the rotor orientation, and in practice [14] for the hardware components. These methods work well for either fixed designs or fixed tasks such as following a trajectory. In this paper, we develop a method that allows to check whether a modular multi-rotor vehicle satisfies a task requirement or not. We reconfigure the vehicle by leveraging on modularity to expand its capabilities and satisfy the task requirements. Fig. 1 illustrates a potential application of our method where we can find an appropriate multi-rotor configuration to operate a screwdriver. We consider the reconfiguration as an offline procedure which is not affected by any physical constraints such as the space limit [15].

The main contribution of this paper is threefold. First, we provide a formal definition of aerial task requirements present a geometric model that characterizes the feasible wrenches of a multi-rotor vehicle with uni-directional rotors. Second, we present a convex hull abstraction of the actuation capabilities of a multi-rotor vehicle, and an optimization-based algorithm to identify whether a configuration satisfies a set of task requirements. Third, we develop search based

J. Xu and D. Saldaña are with the Autonomous and Intelligent Robotics Laboratory (AIRLab), Lehigh University, PA, USA: {jix519, saldana}@lehigh.edu

method that makes a modular multi-rotor vehicle expand its capabilities to satisfy the task requirements. Although we analyze the methods with H-ModQuad modules, the abstractions and algorithms are applicable to any modular multi-rotor vehicles.

## II. BACKGROUND ON MODULAR MULTI-ROTORS

Integrating modularity into multi-rotor systems enhances their versatility. Modular systems can adapt to different tasks through self-reconfiguration. This concept has sparked a wide range of discussion, including the fractal assembly [16], cooperative transport [17], [18], novel structural design [19], [8], and various applications [6], [20].

In [6], the authors present a modular aerial multi-rotor vehicle design called ModQuad, where homogeneous quadrotor modules assemble in mid-air to compose a structure. Our previous work presents H-ModQuad [8], which integrates heterogeneous quadrotor modules with tilted rotors in different directions. This change allows the vehicle to increase the number of ADOF. This paper focuses on modular aerial multi-rotors composed of homogeneous modules with tilted motors.

**Definition 1 (Module).** A module is an aerial vehicle composed of a quadrotor within a cuboid frame. The propellers do not have to be vertical with respect to the base of the module, and their orientation determines the actuation properties of the module.

In contrast to our previous work on heterogeneous modules, this work is focused on using a single type of module to assemble a versatile robot. Based on an attaching mechanism on the frame, a pair of modules can dock and create a rigid connection by aligning their vertical faces. Via docking actions, we assemble a *structure*.

**Definition 2 (Structure).** A structure is a group of  $n \geq 1$  rigidly-connected modules docked horizontally that function as a single multi-rotor vehicle.

We denote the unit basis in  $\mathbb{R}^3$  by  $e_1 = [1, 0, 0]^\top$ ,  $e_2 = [0, 1, 0]^\top$ , and  $e_3 = [0, 0, 1]^\top$ . The  $i$ -th module in the structure has a module reference frame,  $\{M_i\}$ , with its origin in the module's center of mass (COM). The four propellers are in a square configuration and located on the  $xy$ -plane. A structure has  $4n$  rotors. For each rotor  $k \in \{1, 2, \dots, 4n\}$ , we define a propeller frame  $\{P_k\}$  with its  $z$ -axis pointing in the direction of the propeller force and its  $x$ -axis points towards the front of the module. The coordinate frames of a module are illustrated in Fig. 2, where  $\{W\}$  stands for the world reference frame. The structure frame, denoted by  $\{S\}$ , has its origin in its COM. Without loss of generality, we align all the module frames in the structure and define the  $x$ -,  $y$ - and  $z$ -axes of  $\{S\}$  as in parallel to the  $x$ -,  $y$ - and  $z$ -axes of all modules in the structure.

Once a configuration is assembled, the position and orientation of all rotors are fixed with respect to  $\{S\}$ . We assume all  $4n$  rotors on the structure are identical, and each of them can generate a thrust force  $f_k$  and a torque from the air drag. The actuation of the rotors generates a wrench that applies on the structure, which is described by a linear mapping,  $w = Au$ , where the input  $u = [f_1, f_2, \dots, f_{4n}]^\top$

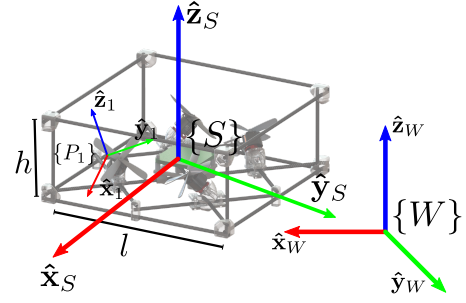


Fig. 2: A module with its coordinate frames and dimensions labelled.  $\{P_1\}$ ,  $\{S\}$ ,  $\{W\}$  are the frames of the 1<sup>st</sup> rotor, the structure, and the world reference, respectively. Note that this module composes a structure, thus  $\{S\}$  aligns with  $\{M_1\}$ .

is the thrust force of all rotors, and the wrench vector  $w = [f^\top \tau^\top]^\top$  is a  $6 \times 1$  vector of force and torque vectors. The configuration matrix,

$$A = \begin{bmatrix} \cdots & {}^S R_k e_3 & \cdots \\ \cdots & p_k \times {}^S R_k e_3 + {}^S R_k (-1)^{1+k} c_\tau e_3 & \cdots \end{bmatrix}, \quad (1)$$

is a  $6 \times 4n$  matrix that maps the input forces into the total wrench in  $\{S\}$ , where  ${}^S R_k \in \text{SO}(3)$  and  $p_k \in \mathbb{R}^3$  describe the orientation and position of the  $k$ -th rotor with respect to  $\{S\}$ , respectively.  $c_\tau$  is a coefficient related to the motors and propellers that can be obtained experimentally.

In traditional multi-rotor vehicles, for instance, a quadrotor, the matrix  $A$  has dimension  $6 \times 4$  and is fixed after building the robot [21], [22]. In our case, the dimension of the matrix can grow in groups of four columns by increasing the number of modules, and its values will depend on the structure configuration. The configuration matrix describes the actuation properties of a structure. For instance, the number of ADOF is equal to the rank of the configuration matrix [8].

## III. PROBLEM STATEMENT

The configuration of a multi-rotor structure dictates how the input forces are mapped to the total wrench, which depends on the *rotor configuration*, i.e., the rotors' position and orientation of the multi-rotor structure.

**Definition 3 (Rotor Configuration).** The rotor configuration of a structure,  $\mathcal{D}$ , is defined as a set of tuples  $\mathcal{D} = \{(p_k, {}^S R_k) | k = 1, 2, \dots, 4n\}$  that represents each rotor's position and orientation with respect to the structure frame.

The actuation capability of a multi-rotor vehicle is defined as the set of all *feasible wrenches* based on its configuration matrix  $A$  and the actuation limit of the input force,  $f_{max}$ .

**Definition 4 (Feasible Wrenches).** The set of feasible wrenches,  $\mathcal{W}$ , defines all available wrenches  $\mathcal{W}$  that a structure with a configuration matrix  $A$  can generate, i.e.,

$$\mathcal{W} = \{Au | 0 \preceq u \preceq f_{max}, u \in \mathbb{R}^{4n}\}, \quad (2)$$

where the operator " $\preceq$ " represents element-wise comparison. Considering a structure with uni-directional motors, all propellers are identical, and they can produce continuous non-negative thrust force which has an upper bound  $f_{max}$ . Aerial

tasks involve physical interaction with the environment and require to generate certain wrenches.

**Definition 5** (Task Requirement). A task requirement is a set of wrenches  $\mathcal{T} \subset \mathbb{R}^6$  that a robot needs to generate to perform a task.

A structure with feasible wrenches  $\mathcal{W}$  satisfies a task requirement  $\mathcal{T}$ , if  $\mathcal{T} \subseteq \mathcal{W}$ . For instance, to manipulate a screwdriver, a multi-rotor needs to generate both a torque for rotation and a force for translating the screwdriver. Thus, the task requirement is  $\mathcal{T} = \mathbf{w}_1, \mathbf{w}_2$ , where  $\mathbf{w}_1$  is the wrench for translation and alignment, and  $\mathbf{w}_2$  is for driving the screw. Our objective is to determine whether a given structure satisfies a task requirement, and if not, to enhance the structure capabilities by adding modules until it satisfies the requirement. We formally define the problem as follows.

**Problem 1** (Finding structure configurations). Given a task requirement  $\mathcal{T}$ , find the structure configuration  $\mathcal{D}$  with the minimum number of modules that is able to satisfy the task requirement.

#### IV. STRUCTURE-TASK REQUIREMENT EVALUATION

The main challenge to solve Problem 1 is that the set  $\mathcal{W}$  has infinite elements, making it computationally difficult to explore. Notice that the constraints on the input vector  $0 \preceq \mathbf{u} \preceq \mathbf{f}_{max}$  represent the intersection of  $2 \cdot 4n$  half-spaces in the  $\mathbb{R}^{4n}$  input space, which forms a convex polyhedron [23]. Since the element-wise comparison provides both upper and lower bounds for the polyhedron along each axis, the polyhedron of constraints is a bounded convex polytope [24]. The feasible wrenches are obtained by mapping the polytope with an affine function  $\mathbf{A}$  to the  $\mathbb{R}^6$  wrench space. Thus,  $\mathcal{W}$  is a convex polytope in  $\mathbb{R}^6$ .

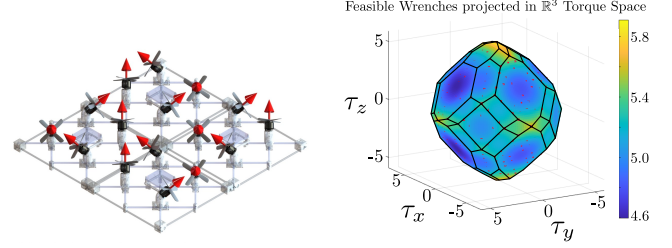
A closer inspection on (2) shows that  $\mathcal{W}$  is the Minkowski sum of the column vectors of  $\mathbf{A}$  whose size is scaled by  $\mathbf{f}_{max}$ , which means  $\mathcal{W}$  is a zonotope. Such a zonotope is defined as a set in  $\mathbb{R}^n$  constructed from generator vectors  $\mathbf{g}_i$ ,  $i = 1, \dots, 4n$ , of which an element can be expressed as  $\sum_i a_i \mathbf{g}_i$ ,  $0 \leq a_i \leq 1$ . Different choices of  $a_i$  result in different elements, and the zonotope is the union of all such elements [25], [26], [27], [28], [29]. The generator vectors of the feasible wrenches  $\mathcal{W}$  are the column vectors of  $\mathbf{A}$ . Based on this insight, we provide two methods to determine whether the condition  $\mathcal{T} \subseteq \mathcal{W}$  is satisfied or not.

##### A. Convex Hull Generation for Zonotopes

Since a zonotope is a polytope and the convex hull of a convex polytope is the polytope itself, we characterize the feasible wrenches  $\mathcal{W}$  with the convex hull of  $\mathcal{W}$ , denoted as  $\mathcal{H}(\mathcal{W})$ . Furthermore, the convenience of the representation of a zonotope with only its generators allows us to generate  $\mathcal{H}(\mathcal{W})$  directly from  $\mathbf{A}$ . Thereby, one approach based on the concept of convex hull is to find  $\mathcal{H}(\mathcal{W})$  using  $\mathbf{A}$  and then check whether all required wrenches belong to it or not.

We start by introducing a discretization approach to construct the convex hull of the zonotope. First, discretize the feasible wrenches for binary input vectors which contain either  $\mathbf{f}_{max}$  or 0, and group them in a set

$$\mathcal{S} = \{\mathbf{A}\mathbf{u} \mid \mathbf{u} = \{0, \mathbf{f}_{max}\}^{4n}\}. \quad (3)$$



(a) A 2x2 structure is composed of four modules. The red arrows represent the direction of the rotors. (b) The polytope of the feasible wrenches  $\mathcal{W}$  of the structure, projected in the torque space.

Fig. 3: A structure composed of four modules and a visualization of its feasible wrenches projected in the torque space.

Then, find the convex hull

$$\mathcal{H}(\mathcal{S}) = \{\sum_i \lambda_i \mathbf{s}_i \mid \sum_i \lambda_i = 1, \lambda_i \geq 0, \mathbf{s}_i \in \mathcal{S}\} \quad (4)$$

of the set of points  $\mathcal{S}$  using existing convex hull generation algorithms such as the Quickhull algorithm [30].

**Proposition 1.** The convex hull of the set  $\mathcal{S}$ ,  $\mathcal{H}(\mathcal{S})$ , is equivalent to  $\mathcal{W}$ , if  $\mathcal{W}$  and  $\mathcal{S}$  are described by (2) and (3), respectively.

*Proof.* To prove the equivalency, we show  $\mathcal{H}(\mathcal{S}) \subseteq \mathcal{W}$ , then  $\mathcal{W} \subseteq \mathcal{H}(\mathcal{S})$ . First, by definition of the convex hull, for any arbitrary point  $\mathbf{v} \in \mathcal{H}(\mathcal{S})$ , there exists a vector  $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_{4n}]^T$  such that  $\mathbf{v} = \sum_i \lambda_i \mathbf{s}_i$  for all  $\mathbf{s}_i \in \mathcal{S}$  where  $\lambda_i \geq 0$ , and  $\sum_i \lambda_i = 1$ . By definition of  $\mathcal{S}$ , any element in  $\mathcal{S}$  can be expressed as  $\mathbf{A}\mathbf{u}$  for some  $\mathbf{u} \in \{0, \mathbf{f}_{max}\}^{4n}$ . Substituting all  $\mathbf{s}_i$  with  $\mathbf{A}\mathbf{u}_i$ , we obtain  $\mathbf{v} = \sum_i \lambda_i \mathbf{s}_i = \sum_i \lambda_i (\mathbf{A}\mathbf{u}_i) = \mathbf{A} \sum_i (\lambda_i \mathbf{u}_i)$  for any  $\mathbf{v} \in \mathcal{H}(\mathcal{S})$ . Let us define a vector  $\boldsymbol{\sigma} \in [0, \mathbf{f}_{max}]^{4n}$ . We can express  $\sum_i \lambda_i \mathbf{u}_i = \boldsymbol{\sigma}$ ,  $0 \preceq \boldsymbol{\sigma} \preceq \mathbf{f}_{max}$  because  $\sum_i \lambda_i = 1$  and  $\mathbf{u}_i \in \{0, \mathbf{f}_{max}\}$ . Therefore, for any arbitrary  $\mathbf{v} \in \mathcal{H}(\mathcal{S})$ , there always exists  $\mathbf{u}$  such that  $\mathbf{v} = \mathbf{A}\mathbf{u}$ ,  $0 \preceq \mathbf{u} \preceq \mathbf{f}_{max}$ , meaning that  $\mathbf{v} \in \mathcal{W}$ , which shows  $\mathcal{H}(\mathcal{S}) \subseteq \mathcal{W}$ .

Second, we show  $\mathcal{W} \subseteq \mathcal{H}(\mathcal{S})$ . For an arbitrary point  $\mathbf{w} \in \mathcal{W}$ , by the definition of  $\mathcal{W}$ , there exists an input vector  $\mathbf{u}$  such that  $\mathbf{w} = \mathbf{A}\mathbf{u}$ ,  $0 \preceq \mathbf{u} \preceq \mathbf{f}_{max}$ . The following procedure shows that any input can be rewritten as  $\mathbf{u} = \sum_i \lambda_i \mathbf{v}_i$  such that there are  $\epsilon \leq 4n$  unique elements in  $\mathbf{u}$ ,  $\mathbf{v}_i \in \{0, 1\}^{4n}$ ,  $0 \leq \lambda_i \leq \mathbf{f}_{max}$ , and  $\sum_i \lambda_i \leq \mathbf{f}_{max}$ . **a.** if  $\mathbf{u} = \mathbf{0}$ , terminate; else mark the current iteration as  $i$ . **b.** Find the smallest entry in  $\mathbf{u}$ , mark its amplitude as  $\lambda_i$ . **c.** the  $j$ -th element in  $\mathbf{v}_i$  is assigned as  $v_{ij} = 1$  if and only if  $u_j \geq \lambda_i$ , otherwise,  $v_{ij} = 0$ . **d.**  $\mathbf{u} = \mathbf{u} - \lambda_i \mathbf{v}_i$ , go to step **a.** In the procedure, we can see  $\mathbf{v}_i \in \{0, 1\}^{4n}$  because their components are either 0 or 1. There is no repetitive appearance of  $\mathbf{v}_i$  vectors because otherwise the identical vectors can merge by adding up the corresponding amplitudes. Therefore,  $\mathbf{A}\mathbf{v}_1, \dots, \mathbf{A}\mathbf{v}_\epsilon \in \mathcal{S}$ , and  $\frac{\mathbf{f}_{max}}{\sum_i \lambda_i} \mathbf{A}\mathbf{u} \in \mathcal{H}(\mathcal{S})$  by the definition of the convex hull. Since  $\mathbf{0} \in \mathcal{H}(\mathcal{S})$ ,  $\mathbf{w} = \mathbf{A}\mathbf{u}$  is on the segment between  $\mathbf{0}$  and  $\frac{\mathbf{f}_{max}}{\sum_i \lambda_i} \mathbf{A}\mathbf{u}$ ,  $\mathbf{w} \in \mathcal{H}(\mathcal{S})$ , which shows  $\mathcal{W} \subseteq \mathcal{H}(\mathcal{S})$ . Thus,  $\mathcal{H}(\mathcal{S}) \equiv \mathcal{W}$ .  $\square$

Computing all combinations of the binary inputs, i.e., combinatorial operations of columns vectors in  $\mathbf{A}$ , induces exponential time complexity with respect to the number of

---

**Algorithm 1:** Divide-and-conquer approach to find the convex hull of the feasible wrenches for a multi-rotor structure

---

```

1 Function ConstructHull ( $\mathbf{A}$ );
   Input :  $\mathbf{A}$ : configuration matrix of the structure
   Output:  $\mathcal{H}(\mathcal{W})$ : convex hull of the feasible wrenches
2 if  $\mathbf{A}.\text{columns} \leq 1$  then
3    $\mathcal{S} := \{\mathbf{A}\mathbf{u} | \mathbf{u} \in \{0, f_{max}\}^{\mathbf{A}.\text{columns}}\}$ 
4   return QuickHull( $\mathcal{S}$ )
5 else
6    $\mathcal{H}_1 := \text{ConstructHull}(\text{first } \lceil \frac{\mathbf{A}.\text{cols}}{2} \rceil \text{ columns of } \mathbf{A})$ 
7    $\mathcal{H}_2 := \text{ConstructHull}(\text{last } \lceil \frac{\mathbf{A}.\text{cols}}{2} \rceil \text{ columns of } \mathbf{A})$ 
8    $\mathcal{V} := \emptyset$ 
9   for  $\mathbf{v}_1 \in \mathcal{H}_1.\text{vertices}, \mathbf{v}_2 \in \mathcal{H}_2.\text{vertices}$  do
10     $\mathcal{V} \leftarrow \{\mathbf{v}_1 + \mathbf{v}_2\} \cup \mathcal{V}$ 
11  end
12  return QuickHull( $\mathcal{V}$ )
13 end

```

---

modules and rotors if not optimized. Therefore, we present a divide-and-conquer algorithm that constructs  $\mathcal{H}(\mathcal{S})$  with an upper bound of exponential time complexity in Algorithm 1. If  $\mathbf{A}$  is a single-column matrix, we calculate the corresponding convex hull, whose vertices are the  $\mathbf{0}$  vector and the column vector itself, as the base case using the **QuickHull** algorithm as shown on lines 2 to 4. On lines 5 to 12, if the input configuration matrix has more than 1 column, we first divide the configuration matrix into two halves, then recursively compute the convex hulls based on the two halved configuration matrices. Based on the definition of  $\mathcal{S}$ , the convex hull for the input configuration matrix is equal to the convex hull of the Minkowski-sum of the convex hulls for the two half configuration matrices, which is also a polytope. Since the Minkowski-sum of convex polytopes is commutative [31], the convex hull of the Minkowski-sum of two convex polytopes is equivalent to the convex hull of the Minkowski-sum of the vertices of each polytope, which is captured on lines 9 to 11 in Algorithm 1.

We analyze the time complexity of Algorithm 1 using Master theorem [32]. In each recurrence, the function calls itself with half of the original problem size. Then, it traverses through the vertices found on both hulls and calls **QuickHull** on the collected vertices to merge the divided problems. The complexity of the merging linearly increases with the number of vertices found on the sub-polytopes. Assume the number of columns is  $c$  in the current iteration, the recurrence can be expressed as  $T(c) = 2T(\frac{c}{2}) + h(c, d)$ , where  $d$  represents the number of dimensions of the hull and  $h(c, d)$  is a function that is polynomial when  $d \leq 3$ , and converges to an exponential function of  $c$  as  $d$  increases according to the authors of the **QuickHull** algorithm [30]. Applying the Master theorem, we obtain  $T(n) = \Theta(h(n, d))$ , which shows our algorithm of constructing the convex hull has a time complexity converging to exponential. When  $d = 6$ ,  $h$  is subexponential, *i.e.*, polynomial. In addition, we denote  $t = |\mathcal{T}|$  as the number of elements in  $\mathcal{T}$ , then determining whether  $\mathcal{T} \subseteq \mathcal{W}$  has time complexity of  $O(tf)$ , where  $f$  is the number of faces on  $\mathcal{H}(\mathcal{S})$  which is in the order of  $O(n^6)$ . Thus, the overall complexity of this approach is  $\Theta(h(n, d))$ .

## B. Optimization

Finding  $\mathcal{H}(\mathcal{W})$  involves the geometric construction of a convex hull, whose computational complexity suffers from the curse of dimensionality [33], especially due to the function  $h(n, d)$ . In fact, since the generators of the zonotope  $\mathcal{W}$  are in  $\mathbb{R}^6$ , the number of the intermediate vertices, *i.e.*,  $\mathbf{v}_1 + \mathbf{v}_2$  on lines 9 to 11 in Algorithm 1 are most likely to end up being the vertices of  $\mathcal{H}(\mathcal{W})$  [34], which is unoptimizable using any technique without losing precision. When the number of elements in  $\mathcal{T}$  is small, generating  $\mathcal{H}(\mathcal{W})$  then check against the elements in  $\mathcal{T}$  may be a waste of computational power.

Inspired by the optimization method used in zonotope containment problems [35], we develop a optimization-based method. The optimization-based approach to determine if a structure of configuration matrix  $\mathbf{A}$  satisfies a task requirement  $\mathcal{T}$  iterates through vectors  $\mathbf{w} \in \mathcal{T}$  and check if they belong to  $\mathcal{W}$ . Using the convexity of a zonotope, instead of dealing with the infinite set  $\mathcal{W}$ , we can compare the maximum magnitude on the boundary point of  $\mathcal{W}$  in the direction of  $\mathbf{w}$ . Let  $\mathbf{w} = |\mathbf{w}|\hat{\mathbf{w}}$  where  $|\mathbf{w}|$  and  $\hat{\mathbf{w}}$  are the magnitude and unit direction vector of  $\mathbf{w}$ , respectively. By maximizing the wrench in the direction of  $\hat{\mathbf{w}}$ , *i.e.*,

$$\begin{aligned}
& \underset{\mathbf{u}}{\text{maximize}} && \lambda, \\
& \text{subject to} && 0 \preceq \mathbf{u} \preceq f_{max}, \\
& && \mathbf{A}\mathbf{u} = \lambda\hat{\mathbf{w}},
\end{aligned} \tag{5}$$

By solving (5) for all  $\hat{\mathbf{w}}$  that is on the unit 6-sphere, we obtain the maximum magnitude of all possible wrenches achievable by the multi-rotor, which bound  $\mathcal{H}(\mathcal{W})$ . Fig. 3b shows one such convex hull for the structure shown in Fig. 3a. A specific representation of the force actuation is presented in [11]. Their approach focuses on the force capabilities, constraining the torque to  $\mathbf{0}$  when generating the polytope of all feasible forces. In our general representation, we can take such constraints into consideration by adding  $\boldsymbol{\tau} = \mathbf{0}$  as an additional equality constraint in (5). we obtain the maximum feasible actuation  $\lambda$  in the direction of  $\hat{\mathbf{w}}$ . If  $\lambda \geq |\mathbf{w}|$ , the vehicle can generate the wrench  $\mathbf{w}$  since it is within its actuation limits. We repeat the procedure for all elements in  $\mathcal{T}$  to determine if  $\mathcal{T} \subseteq \mathcal{W}$ .

The maximization problem in (5) finds the wrench vector on the boundary of  $\mathcal{W}$  in the direction of  $\hat{\mathbf{w}}$ . When  $\mathbf{u} = \mathbf{0}$ , all rotors of the multi-rotor stop rotating. In this case, the equality constraint  $\mathbf{A}\mathbf{u} = \lambda\hat{\mathbf{w}}$  yields  $\lambda = 0$ , which gives a lower bound  $\lambda \geq 0$  for all  $\hat{\mathbf{w}}$ . We apply the norm operator on both sides of the equality constraint, which yields  $\|\mathbf{A}\mathbf{u}\| = \|\lambda\hat{\mathbf{w}}\|$ . Since  $\|\hat{\mathbf{w}}\| = 1$  and  $\lambda \geq 0$ , we obtain  $\lambda = \|\mathbf{A}\mathbf{u}\|$ . Thus, the objective of maximizing  $\lambda$  in (5) is equivalent to maximizing  $\lambda^2 = \mathbf{u}^\top \mathbf{A}^\top \mathbf{A} \mathbf{u}$ , which shows that it is equivalent to a convex quadratic programming optimization problem. Therefore, we can obtain the solution in time complexity of  $O(((4n)^2 + 4n \cdot 4n)^2 (4n)^4) = O(n^8)$  with extended Karmarkar's projective algorithm [36], [37] in polynomial time. Determining whether  $\mathcal{T} \subseteq \mathcal{W}$  has time complexity of  $O(tn^8)$ . In comparison to the convex hull based approach, although this approach is less time-efficient when checking  $\mathcal{T} \subseteq \mathcal{W}$ , it does not require a pre-computation, *i.e.*, generating the convex hull in Algorithm 1.



**Algorithm 2:** Configuration generation method that utilizes the property of symmetry to ensure torque-balance.

---

```

1 Function GenerateConfigSymmetry( $\mathcal{D}$ ,  $n_{max}$ );
   Input :  $\mathcal{D}$ : current structure configuration
            $n_{max}$ : maximum number of modules added
   Output:  $\mathcal{S}_1, \dots, \mathcal{S}_{n_{max}}$ : all possible configurations,
           grouped by the number of modules
2  $\mathcal{S}_0 := \{\mathcal{D}\}$  for  $n \in \{0, 1, 2, \dots, n_{max} - 1\}$  do
3   for  $\mathcal{D}_i \in \mathcal{C}_n$  do
4      $\mathcal{S}_{n+1} := \{\}$ 
5     for  $s \in \text{attachable surface on } \mathcal{D}_i$  do
6        $s_{sym} := \text{the attachable surface on } \mathcal{D}_i \text{ that is}$ 
            $\text{centrosymmetric to } s$ 
7        $\mathcal{D}_{temp} := \mathcal{D}_i \text{ with one more module attached}$ 
            $\text{at } s \text{ and } s_{sym}$ 
8        $\mathcal{S}_{n+1} = \{\mathcal{D}_{temp}\} \cup \mathcal{S}_{n+1}$ 
9     end
10  end
11 end

```

---

Therefore, when  $|\mathcal{T}|$  is small, the optimization-based approach provides better time-complexity.

## V. TASK ADAPTATION

The main challenges in finding a suitable configuration that satisfies a set of task requirements include the uncertain number of modules and the quadratically increasing number of possible configurations as the number of modules increases. Such a problem is different from a pattern formation problem such as in [38], or a reconfiguration problem such as in [39] that are well studied in the multi-agent literature. In our case, the cost of locomotion is ignored and the destination configuration is unknown.

We present two methods to adapt to a task requirement. We can add rotors to a multi-rotor one by one to reconfigure, but defining the position and orientation of each rotor with respect to the new structure during the process of reconfiguration would lead us to a difficult problem. Therefore, we leverage the modularity. In our previous work [8], we defined *torque-balanced modules* that form structures. Similar to modules, structures satisfy the torque-balance property; so they generate zero torque when all motors have the same input. It is feasible to control a multi-rotor vehicle that is not torque-balanced [40]. However, the torque-balance property reduces the complexity in its structural configuration. In this paper, we use a module that is torque-balanced and expands its feasible wrenches after connecting with other modules.

**Definition 6** (*T*-Module). A *T*-module is a module with its rotors tilted at an angle  $\eta_j$  around an axis defined by their arm vector  $\mathbf{p}_j$ ,  $j = 1, \dots, 4$ . The angles satisfy  $\eta_1 = \eta_3 = -\eta_2 = -\eta_4$ . Making  $\eta = \eta_1$ , we characterize a *T*-module with a single parameter  $\eta$ .

We use homogeneous *T*-modules with  $\eta = \frac{\pi}{4}$  to compose structures. In Sec. IV, we provide methods of checking if a vehicle satisfies the required wrenches. We utilize these methods to adapt a structure configuration to a task requirement by reconfiguration.

### A. Adaptation by exhaustive search

We present the exhaustive search-based approach to find the configuration with minimum number of modules given the

task requirement  $\mathcal{T}$ , which evaluates *all* possible structures with up to  $n_{max}$  additional modules. Given  $\mathcal{T}$  and the initial design  $\mathcal{D}$ , we dock the modules one by one to the attachable surfaces of the possible current design to ensure the structure is connected, and check if the structure *i*) is torque-balanced, and *ii*) satisfies  $\mathcal{T}$ . The process constrains the relative position of the new module to the structure during the docking so that the new module aligns one of its surfaces with a module in the structure.

To ensure all possible configurations are evaluated, every time a new module is added, the algorithm iterates through all designs from the previous step as possible current designs, and evaluates if connecting a new module to any one of them makes it a structure that satisfies *i*) and *ii*) using the methods introduced in Sec. IV. Therefore, this approach guarantees minimum number of modules added to the initial design.

The exhaustive search evaluates all possible configurations. However, as modules are added to the configuration, the COM of the structure changes, and the potential asymmetry may degrade the control quality. To avoid these problems, we take a heuristic approach in practice.

### B. Heuristic task adaptation

When adding modules to a structure, certain patterns apply to ensure that a structure is torque-balanced. Moreover, adding modules to a structure may cause the COM to shift with respect to the structure frame. Some aerial tasks do not allow the COM to shift. For instance, a structure is equipped with a special mechanism to carry a payload, and we want to increase the maximum weight of the payload the structure can carry. We have to make sure the new configuration does not change the COM. Otherwise, the structure may lose control when transporting the payload. Therefore, we introduce the centrosymmetry pattern that can help reduce the search space and makes sure the COM stays in place during the reconfiguration.

When generating all possible designs, instead of adding modules one by one and iterating through all attachable surfaces of all possible designs from the previous step, we call **GenerateConfigSymmetry** in Algorithm 2 that applies the heuristics of centrosymmetry to the exhaustive search. On line 6 of Algorithm 2, in addition to adding the module at an attachable surface  $s$  on the structure, we also find the surface  $s_{sym}$  that is centrosymmetric to  $s$  with respect to the COM of the existing configuration, and add another module at  $s_{sym}$ . Since all modules are homogeneous, the COM of the new structure remains the same. Note that centrosymmetry is not a necessary, but a sufficient condition for the assembled structure to be torque-balanced and to have a unshifted COM. It is possible that a non-symmetric structure is torque-balanced, or reconfiguring a structure without following the centrosymmetry rule causes no shift on the COM.

## VI. EVALUATION

We implement the algorithms introduced in Sec. IV and V<sup>1</sup>. We provide a task requirement for both the exhaustive search-based and the heuristic approaches to find structure configurations that satisfy the requirement. The task

<sup>1</sup>The source code can be found on <https://github.com/Jarvis-X/HModQuad-sim>

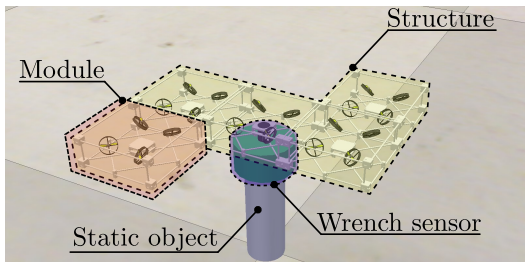


Fig. 4: Setup for the simulation in CoppeliaSim.

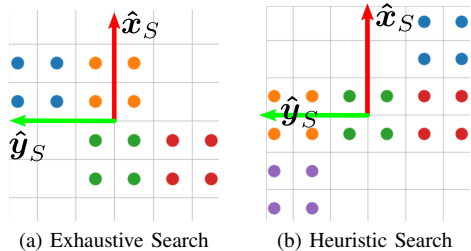
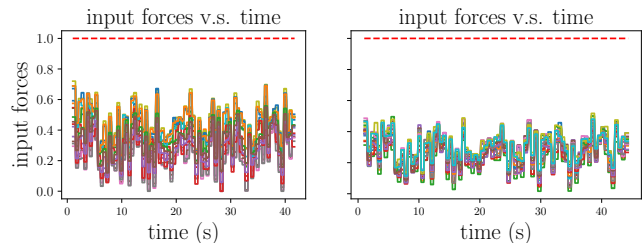


Fig. 5: Configurations found by our methods.

requirement is composed of randomly generated wrenches with an offset towards more force along the positive  $z$ -axis,  $f_z$ , to emulate the actuation to compensate the gravity. For both algorithms, we start with an initial structure of one module and set the maximum number of modules  $n_{max} = 7$ . We assume the maximum thrust force for each rotor  $f_{max} = 1$ , and the side length  $l = 0.4$  for each module.

After the algorithms return structure configurations that satisfy the requirement, we construct the structures in CoppeliaSim [41] environment. Since we want to show that the structure is able to generate the wrenches in the task requirement, we use a “force sensor” component to rigidly connect the COM of the structure to a static wall. All wrenches generated by the structure are transferred through the force sensor to the wall, which measures the wrench generated by the structure without causing any motion. Fig. 4 shows an example of such setup in CoppeliaSim. We randomly generate a task requirement  $\mathcal{T}$  that is composed of 80 wrenches. Each element of the wrenches is a random value uniformly chosen on  $(-0.5, 0.5)$ , and we multiply 30 on  $f_z$  to emphasize the gravity.  $\mathcal{T}$  is the input to both exhaustive and heuristic search algorithms. The exhaustive search-based approach evaluates 9 configurations and returns a configuration of 4 modules as shown in Fig. 5a. The heuristic approach evaluates 4 configurations and returns a configuration of 5 modules as shown in Fig. 5b.

We actuate the constructed structures. In our previous work [8], we present control strategies for a fully-actuated structure. Here, since the structures are rigidly connected by a force sensor to a static wall, their motion properties are ignored. Therefore, we apply a simplified approach. For a configuration, we calculate the desired input  $\mathbf{u}^* = \mathbf{A}^\dagger \mathbf{w}$  for each  $\mathbf{w} \in \mathcal{T}$  every 500 milliseconds. Since each rotor can produce thrust force between 0 and 1, we truncate the input vector  $\mathbf{u}^*$  outside the range of  $[0, 1]$ . The truncated input vector  $0 \leq \mathbf{u} \leq 1$  is then taken as the effective thrust forces the rotors of a structure generate. As shown in Fig. 6a and 6b, the configurations provided by both algorithms can generate



(a) The rotor input of the configuration found by the *exhaustive search*. (b) The rotor input of the configuration found by the *heuristic search*.

Fig. 6: The structure configurations found by both algorithms satisfy the task requirement without input saturation. The dot line represents  $f_{max}$ , and the solid lines represent the input forces of the rotors.

all required wrenches without input saturation.

The simulations for the random task requirement show that the structure configurations found by both algorithms can generate the required wrenches without saturating the motors as shown in Fig. 6, which validates our methods. Fig. 5 illustrates the configurations, where every four circles of the same color represent rotors of one module, and the green circles belong to the initial module. The exhaustive search-based algorithm outputs the configuration in Fig. 5a, which is centrosymmetric but with shifted COM from the initial configuration. It provides the optimal solution to the task requirement as for the less number of modules in comparison to the configuration found by the heuristic method in Fig. 5b.

## VII. CONCLUSION AND FUTURE WORK

In this paper, we have presented a comprehensive framework for analyzing the actuation capabilities of modular multi-rotor vehicles and determining their ability to satisfy task requirements. Our contributions include a formal definition of aerial task requirements, a geometric model of the feasible wrenches of a multi-rotor vehicle with uni-directional rotors, and a convex hull abstraction of the actuation capabilities of a multi-rotor vehicle. We have also introduced an optimization-based algorithm to identify whether a configuration satisfies a set of task requirements, as well as a search-based method for expanding the capabilities of modular multi-rotor vehicles to satisfy task requirements.

Our methods have been demonstrated using the H-ModQuad platform, which features tilted rotors to increase the number of ADOF. However, our abstractions and algorithms are applicable to any modular multi-rotor vehicle. We have shown that our approach is effective in finding the optimal configuration of homogeneous modules that satisfy task requirements.

For future work, we plan to extend our methods to heterogeneous modules and test our structures with real robots. We would also like to explore the possibility of incorporating the dynamics of the structures into the task requirements to enable motion during evaluation. By doing so, we can further improve the adaptability and versatility of modular multi-rotor vehicles in various applications.

## ACKNOWLEDGEMENT

The authors thank Diego S. D’Antonio at Lehigh Univer-

sity for the module design and renderings.

## REFERENCES

- [1] S. Bouabdallah, P. Murrieri, and R. Siegwart, "Design and control of an indoor micro quadrotor," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 5, 2004, pp. 4393–4398 Vol.5.
- [2] S. Rajappa, M. Ryll, H. H. Bühlhoff, and A. Franchi, "Modeling, control and design optimization for a fully-actuated hexarotor aerial vehicle with tilted propellers," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4006–4013.
- [3] A. Alaimo, V. Artale, C. Milazzo, A. Ricciardello, and L. Trefiletti, "Mathematical modeling and control of a hexacopter," in *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2013, pp. 1043–1050.
- [4] R. Beniak and O. Gudzenko, "Control methods design for a model of asymmetrical quadcopter," *Journal of Automation Mobile Robotics and Intelligent Systems*, vol. 10, 2016.
- [5] R. Oung and R. D'Andrea, "The distributed flight array," *Mechatronics*, vol. 21, no. 6, pp. 908–917, 2011. [Online]. Available: <http://dx.doi.org/10.1016/j.mechatronics.2010.08.003>
- [6] D. Saldaña, B. Gabrich, G. Li, M. Yim, and V. Kumar, "Modquad: The flying modular structure that self-assembles in midair," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 691–698.
- [7] M. Zhao, T. Anzai, F. Shi, X. Chen, K. Okada, and M. Inaba, "Design, modeling, and control of an aerial robot dragon: A dual-rotor-embedded multilink robot with the ability of multi-degree-of-freedom aerial transformation," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1176–1183, 2018.
- [8] J. Xu, D. S. D'Antonio, and D. Saldaña, "H-modquad: Modular multirotors with 4, 5, and 6 controllable dof," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 190–196.
- [9] B. Gabrich, D. Saldaña, and M. Yim, "Finding structure configurations for flying modular robots," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 6970–6976.
- [10] M. Ryll, D. Bicego, and A. Franchi, "Modeling and control of fast-hex: A fully-actuated by synchronized-tilting hexarotor," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1689–1694.
- [11] M. Hamandi, F. Usai, Q. Sablé, N. Staub, M. Tognon, and A. Franchi, "Design of multirotor aerial vehicles: a taxonomy based on input allocation," 2021.
- [12] R. Naldi, F. Forte, and L. Marconi, "A class of modular aerial robots," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 3584–3589.
- [13] J. Strawson, P. Cao, T. Bewley, and F. Kuester, "Rotor orientation optimization for direct 6 degree of freedom control of multirotors," in *2021 IEEE Aerospace Conference (50100)*, 2021, pp. 1–12.
- [14] O. Magnussen, G. Hovland, and M. Ottestad, "Multicopter uav design optimization," in *2014 IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, 2014, pp. 1–6.
- [15] D. Halperin, M. v. Kreveld, G. Miglioli-Levy, and M. Sharir, "Space-aware reconfiguration," in *International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2020, pp. 37–53.
- [16] K. Garanger, J. Eppe, and E. Feron, "Modeling and experimental validation of a fractal tetrahedron uas assembly," in *2020 IEEE Aerospace Conference*, 2020, pp. 1–11.
- [17] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative grasping and transport using multiple quadrotors," *Springer Tracts in Advanced Robotics*, vol. 83 STAR, pp. 545–558, 2012.
- [18] D. S. D'Antonio, G. A. Cardona, and D. Saldaña, "The catenary robot: Design and control of a cable propelled by two quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3857–3863, 2021.
- [19] M. J. Duffy and T. C. Samaritano, "The lift! project-modular, electric vertical lift system with ground power tether," in *33rd AIAA Applied Aerodynamics Conference*, 2015, p. 3013.
- [20] B. Gabrich, D. Saldaña, V. Kumar, and M. Yim, "A flying gripper based on cuboid modular robots," 05 2018.
- [21] M. Ryll, H. H. Bühlhoff, and P. R. Giordano, "Modeling and control of a quadrotor uav with tilting propellers," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 4606–4613.
- [22] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro-uav testbed," *IEEE Robotics Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.
- [23] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge U. press, 2004.
- [24] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.
- [25] K. Fukuda, "From the zonotope construction to the minkowski addition of convex polytopes," *Journal of Symbolic Computation*, vol. 38, no. 4, pp. 1261–1272, 2004.
- [26] W. Kühn, "Rigorously computed orbits of dynamical systems without the wrapping effect," *Computing*, vol. 61, no. 1, pp. 47–67, 1998.
- [27] H. S. M. Coxeter, *Regular polytopes*. Courier Corporation, 1973.
- [28] M. Beck and S. Robins, *Computing the continuous discretely*. Springer, 2007, vol. 61.
- [29] B. Grünbaum, V. Klee, M. A. Perles, and G. C. Shephard, *Convex polytopes*. Springer, 1967, vol. 16.
- [30] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software (TOMS)*, vol. 22, no. 4, pp. 469–483, 1996.
- [31] R. Schneider, *Convex bodies: the Brunn–Minkowski theory*. Cambridge U. press, 2014, no. 151.
- [32] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Intro. to algorithms*. MIT press, 2009.
- [33] A. Hinrichs, E. Novak, and H. Woźniakowski, "The curse of dimensionality for the class of monotone functions and for the class of convex functions," *Journal of Approximation Theory*, vol. 163, no. 8, pp. 955–965, 2011. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0021904511000311>
- [34] H. Raynaud, "Sur l'enveloppe convexe des nuages de points aléatoires dans  $\mathbb{R}^n$ ," *Journal of Applied Probability*, vol. 7, no. 1, p. 35–48, 1970.
- [35] A. Deza and L. Pournin, "A linear optimization oracle for zonotope computation," *Computational Geometry*, vol. 100, p. 101809, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925772121000651>
- [36] Y. Ye and E. Tse, "An extension of karmarkar's projective algorithm for convex quadratic programming," *Mathematical programming*, vol. 44, no. 1, pp. 157–179, 1989.
- [37] K. M. Anstreicher, "A monotonic projective algorithm for fractional linear programming," *Algorithmica*, vol. 1, no. 1, pp. 483–498, 1986.
- [38] A. Dutta, P. Dasgupta, and C. Nelson, "Distributed configuration formation with modular robots using (sub)graph isomorphism-based approach," *Autonomous Robots*, vol. 43, no. 4, pp. 837–857, Apr. 2019, publisher Copyright: © 2018, Springer Science+Business Media, LLC, part of Springer Nature.
- [39] E. Yoshida, S. Murata, A. Kamimura, K. Tomita, H. Kurokawa, and S. Kokaji, "A self-reconfigurable modular robot: Reconfiguration planning and experiments," *The International Journal of Robotics Research*, vol. 21, no. 10–11, pp. 903–915, 2002.
- [40] M. W. Mueller, S. J. Lee, and R. D'Andrea, "Design and control of drones," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, pp. 161–177, 2022. [Online]. Available: <https://doi.org/10.1146/annurev-control-042920-012045>
- [41] E. Rohmer, S. P. N. Singh, and M. Freese, "Coppeliassim (formerly v-rep): a versatile and scalable robot simulation framework," in *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2013, www.coppeliarobotics.com.