ModQuad: The Flying Modular Structure that Self-Assembles in Midair

David Saldaña[1]*, Bruno Gabrich[1]*, Guanrui Li[1], Mark Yim[1], and Vijay Kumar[1].

Abstract— **We introduce ModQuad, a novel flying modular robotic structure that is able to self-assemble in midair and cooperatively fly. The structure is composed by agile flying modules that can easily move in a three dimensional environment. The module is based on a quadrotor platform within a cuboid frame which allows it to attach to other modules by matching vertical faces. Using this mechanism, a ModQuad swarm is able to rapidly assemble flying structures in midair using the robot bodies as building units. In this paper, we focus on two important tasks for modular flying structures. First, we propose a decentralized modular attitude controller to allow a team of physically connected modules to fly cooperatively. Second, we develop a docking method that drives pairs of structures to be attached in midair. Our method precisely aligns, and corrects motion errors during the docking process. In our experiments, we tested and analyzed the performance of the cooperative flying method for multiple configurations. We also tested the docking method with successful results.**

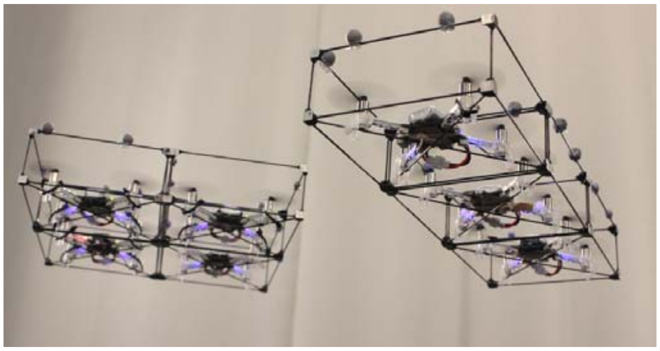


Fig. 1. Flying Modular Structures. A square shape with four modules on the left and a line shape with three modules on the right.

I. Introduction

In biological systems such as ant or bee colonies, collective effort can solve difficult problems such as exploring, transporting food and building massive structures. Some ant species are able to build living bridges by clinging to one another spanning the gaps in the foraging trail. This capability allows them to rapidly connect disjoint areas in order to transport food and resources to their colonies.

Recent works in robotics have been focusing on applying swarm behaviors to solve collective tasks such as construction and transportation. There are two main approaches to assemble structures using autonomous mobile robots. In the first approach, robots transport building units to assemble structures. In [1] is presented an algorithm to control a swarm of robotic termites that can transport, assemble and climb over building blocks. A quadrotor swarm can also be used to transport structure parts, like bricks and beams with magnets, in order to assemble three dimensional structures [2], [3]. In the second approach, robots use their own bodies as building units [4], [5], [6]. A cubic block module was introduced in [7] which is able to attach and to detach other modules forming three dimensional structures. A self-assemble and self-reconfigurable robot system was developed in [8], where robots are coordinated at the same time to dock to form planar structures. Recent algorithms have been focusing on assembling large scale structures using hundreds of robots [8], [9]. Modular boats with rectangular [10] and square shapes [6] have been proposed to self-assemble and to self-reconfigure planar structures on the water. These systems require algorithms that avoid deadlocks during the assembly process. In recent work, we speed up the assembly process by parallelizing docking actions [11].

In [12] is proposed a scalable multi-rotor aircraft which can be manually extended depending on the payload requirement. Oung et al. [13] present a reconfigurable aerial platform based on wheeled hexagonal modules. Each module contains a single propeller that is not able to fly by itself, but it can move on the ground to dock other modules and cooperatively fly. This modular system can also increase its payload capability by scaling the number of docked modules. In addition, it can self-assemble on the ground using omni-directional wheels. A cooperative multi-quadrotor system for transportation purposes is presented in [14]. The authors use multiple quadrotors forming different configurations to grasp and transport different shaped objects.

Flying structures that can self-assemble in midair has not been shown in the literature. This paper introduces ModQuad, the self-assembly structure that can cooperatively fly based on autonomous modules (See Figure 1). Using a large number of robots, it is possible to assemble structures such as bridges, rectangular platforms, among others. In contrast to related works, instead of assembling on the ground or on the water, we propose a faster way to assemble structures. Docking modules in midair offers a relevant advantage in terms of speed during the assembly process. The individual

* Equal contributors

[1] D. Saldaña, B. Gabrich, G. Li, M. Yim and V. Kumar are with the GRASP Laboratory, University of Pennsylvania, Philadelphia, PA, USA: {dsaldana, brunot, lguanrui, yim, kumar}@seas.upenn.edu.

The authors acknowledge the help and support of Michael Whitzer during the intial state of the project. We also acknowledge Justin Thomas for his introduction to quadrotor control and multi-UAV systems.

The authors gratefully acknowledge the support of the Brazilian agency CAPES. The support of DARPA grant HR00111520020, ONR grants N00014-15-1-2115 and N00014-14-1-0510, ARL grant W911NF-08-2-0004, NSF grant IIS-1426840, NSF grant 1138847 and TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

978-1-5386-3081-5/18/$31.00 ©2018 IEEE

modules are small and agile so they can rapidly move through environments with obstacles. There are scenarios where the time-response is crucial to save human lives. For example, in a burning building, the individual modules can rapidly navigate through cluttered environments from a base-station to the target building. Then, they can assemble bridges, external staircases or platforms near windows to offer alternative exits.

The contributions of this paper are threefold. *i)* We introduce ModQuad, a novel flying modular robotic structure that is able to self-assemble in midair and cooperatively fly. *ii)* ModQuad is the first modular system that is able to self-assemble in midair. We present a docking method that accurately aligns and attaches pairs of flying structures in midair. *iii)* We present a stable decentralized modular attitude controller to allow a set of attached modules to cooperatively fly. Our controller generates the required moments minimizing motor saturation.

II. ModQuad Design

Our design is focused on developing a modular robot with the ability to dock in midair. In this way, we extend the capabilities of the regular aerial flying vehicles. Our modular robot, ModQuad, is propelled by a quadrotor within a light-weight cuboid frame with a passive docking mechanism. Its main components are described as follows.

A. Flying Vehicle

The ModQuad is propelled by a quadrotor platform. In this case, we use the Crazyflie 2.0. This robot has been used in swarm applications with large numbers of robots [15]. It is open-source and open-hardware. The vehicle weighs 27g, having a maximum payload of 15g. Its dimensions are 92x92x29mm and its battery lasts around five minutes. Its low-cost and total payload gives an acceptable scenario for a large number of modules.

B. Modular Frame

The quadrotor is enclosed and attached to a cuboid frame as it can be seen in Figure 2. The dimensions of this cuboid are 116x116x48mm. Light-weight carbon fiber rods connected by eight 3-D printed ABS connectors form the frame. The frame weight is important due to tight payload constraints. Our current frame design weights 7g , about half the payload capability.

C. Docking Mechanism

To enable rigid attachments between modules, we include a docking mechanism in the modular frame. We used Neodymium Iron Boron (NdFeB) magnets as passive actuators due to their large strength-to-weight ratio. The magnets are redsquares square, 6.35x6.35x0.79mm. Two magnets are located at each of the eight 3-D printed connectors in the frame (See Figure 2). These magnets enable our modules to have connections on the four vertical faces. When two modules are connected face-to-face, the four magnets are able to provide a bonding force equivalent to a 1kg, which is a strong force in comparison to the module mass (40g) The strength-to-weight ratio is approximately 25:1. Although a module has six faces, we focus on planar structures only, where vertical faces perform horizontal dockings. Once attached, undocking is a difficult task and is left to future work.

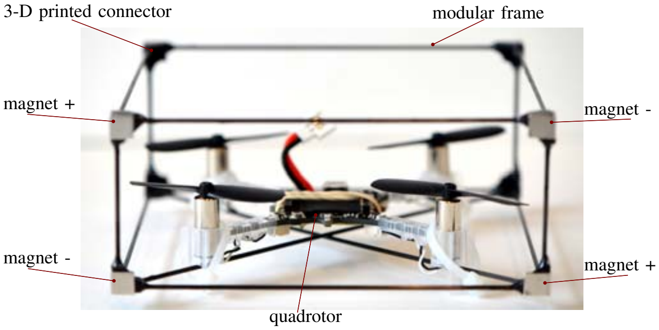


Fig. 2. A Flying Modular Robot. It is based on a quadrotor within a cuboid frame. It is equipped with a docking mechanism, made up of square permanent magnets located in the corners of the frame. This enables the modules to rigidly attach to each other.

III. ModQuad Model

In this section, we describe a general model for ModQuad and state the two main problems that we solve in this paper. This system is able to assemble structures from modules defined as follows.

Definition 1 (Module). *A* module *is a flying robot that can move by itself in a three dimensional environment and horizontally dock to other modules.*

This module is based on a quadrotor platform within a cuboid modular frame. The lower and upper faces of the cuboid have an area $w \times w$ and the frame has a height h. A docking mechanism allows the module to horizontally attach to other modules. The modular robot has a mass m, including the quadrotor, the frame and the docking mechanism.

We consider a team of N modules, which are indexed by the set $\mathcal{M} = \{1, ..., N\}$. All modules are homogeneous, including shape, mass, inertia, and actuators. We define a set of connected modules as a structure.

Definition 2 (Structure). *A* flying structure, $\mathcal{S} \subseteq \mathcal{M}$, *is a non-empty set of rigidly connected modular robots that behaves as a single rigid body. These modules are horizontally connected by docking along the sides so the resulting shape has the same height h.*

A. Coordinate frames

We set three different coordinate frames to define the module and the structure pose:

1) The world coordinate frame, W: or inertial frame is fixed and has its z-axis pointing upwards. We denote the location of the center of mass of the ith module in the world frame W by $\mathbf{x}_i \in \mathbb{R}^3$. The robot attitude is represented by the Euler angles $\mathbf{\Theta}_i = [\phi_i, \theta_i, \psi_i]^\top$ for roll ϕ_i, pitch θ_i, and

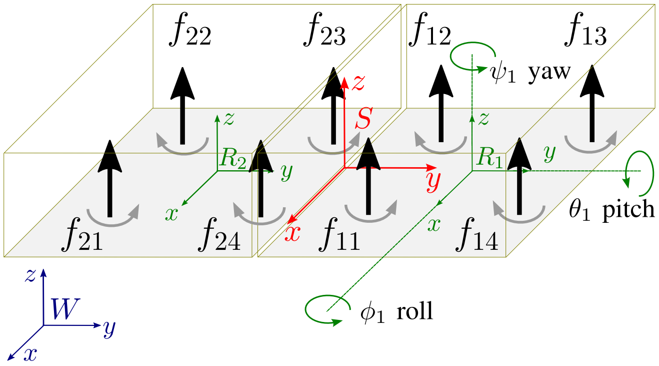


Fig. 3. Representation of an structure with two modules. The black arrows represent the rotor forces, and the curved gray arrows represent the rotor moments. The blue, green, and red axis represent the world, module and structure coordinate frames respectively.

yaw ψ_i. We can see the attitude angles with respect to the world frame for module 1 in Figure 3.

2) *The module coordinate frame, R_i:* is defined for each robot. The origin is attached to the center of mass and the x-axis is aligned to the front of the module. The angular velocities in the module frame are denoted by $\mathbf{\Omega}_i = [p_i, q_i, r_i]^\top$.

3) *The structure coordinate frame, S:* is defined for a set of attached modules $\mathcal{S}$. The origin is attached to the center of mass of the structure. We assume that all modules point in the same direction. Thus, the x-axis of the structure coordinate frame is parallel to the x-axis of all module coordinate frames in the structure. Figure 3 illustrates two modules and their associated coordinate frames.

We assume that the modules have approximately the same yaw orientation $\psi_1 = ... = \psi_N$. Without loss of generality, we assume that this orientation is $\psi_i = 0$, for all $i \in \mathcal{M}$.

B. Robot actuators

Each module i has four vertical rotors in a square configuration, indexed by $j = 1, ..., 4$, each with an angular speed ω_{ij} that generates vertical forces

$$f_{ij} = k_F\, \omega_{ij}^2,$$

and moments

$$M_{ij} = \pm k_M\, \omega_{ij}^2,$$

where k_F and k_M are motor constants that can be obtained experimentally. The sign of the moment depends on the direction that the motor spins. It is either positive for counterclockwise or negative for clockwise.

The coordinates of the rotors 1 to 4 in the module frame R_i are $(d, -d, 0)$, $(-d, -d, 0)$, $(-d, d, 0)$, and $(d, d, 0)$ respectively, where d is the distance from the rotor to the respective x- or y-axis of the module coordinate frame.

C. Robot sensors

Each module i is able to measure its pose, its location in the world coordinate frame $\mathbf{x}_i$, its angles for roll ϕ_i, pitch θ_i, and yaw ψ_i, and its respective angular velocities p_i, q_i, and r_i. We identify if a module i is connected to another module j using the location sensor. We check if $\|\mathbf{x}_i - \mathbf{x}_j\| = w$ is satisfied, where $\|\cdot\|$ denotes the Euclidean norm.

D. Dynamics of the modular rigid body

A set of connected modules – structure $\mathcal{S}$ – forms a single rigid body. We denote the number of robots in the structure $\mathcal{S}$ by $n = |\mathcal{S}|$ ($|\mathcal{S}|$ is the cardinality of $\mathcal{S}$). All modules in the structure share the same plane and have the same orientation. We denote the location of the ith module in the structure coordinate frame by (x_i, y_i, z_i).

The thrust and attitude of the flying structure depends on the forces and moments produced by each rotor. The total thrust F and the roll, pitch and yaw moments, denoted by M_x, M_y, and M_z respectively, are computed as the result of all the rotor forces in the structure

$$\begin{bmatrix} F \\ M_x \\ M_y \\ M_z \end{bmatrix} = \sum_i \begin{bmatrix} 1 & 1 & 1 & 1 \\ y_{i1} & y_{i2} & y_{i3} & y_{i4} \\ -x_{i1} & -x_{i2} & -x_{i3} & -x_{i4} \\ \frac{k_M}{k_F} & -\frac{k_M}{k_F} & \frac{k_M}{k_F} & -\frac{k_M}{k_F} \end{bmatrix} \begin{bmatrix} f_{i1} \\ f_{i2} \\ f_{i3} \\ f_{i4} \end{bmatrix} \quad (1)$$

where (x_{ij}, y_{ij}) denotes the location of the rotor $j = 1, ..., 4$ that belongs to the ith module with respect to S. In this dynamical system, we can control the force of each individual actuator by the input vector

$$[f_{i1}, f_{i2}, f_{i3}, f_{i4}]^\top = \mathbf{u}_i. \quad (2)$$

The resultant force and moments generate translational and rotational accelerations.

$$nm\, \ddot{\mathbf{x}}_{\mathcal{S}} = \begin{bmatrix} 0 \\ 0 \\ -nm\, g \end{bmatrix} + \mathbf{R}_S^W \begin{bmatrix} 0 \\ 0 \\ \sum_{ij} f_{ij} \end{bmatrix},$$

where g is the gravity constant, $\mathbf{R}_S^W \in \mathbb{R}^{3\times 3}$ is the rotation matrix that transforms from the structure coordinate frame S to the world coordinate frame W.

We assume that the each module is symmetric and its inertia tensor is a diagonal matrix, denoted by $\mathbf{I} = \mathrm{Diag}(I_x, I_y, I_z)$. We can describe the rotational accelerations using the linearized model

$$\mathbf{I}_{\mathcal{S}} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix}, \quad (3)$$

where $\mathbf{I}_{\mathcal{S}}$ is the mass moment of inertia of the structure $\mathcal{S}$.

We can compute the structure inertia, denoted by $\mathbf{I}_{\mathcal{S}}$, based on the inertia matrix of a single module $\mathbf{I}$, using the parallel axis theorem

$$\mathbf{I}_{\mathcal{S}} = n\mathbf{I} + m \begin{bmatrix} \sum_i y_i^2 & 0 & 0 \\ 0 & \sum_i x_i^2 & 0 \\ 0 & 0 & \sum_i x_i^2 + y_i^2 \end{bmatrix}. \quad (4)$$

E. Objective

In this paper, we want to control two of the main actions that the team of modular robots can perform. These actions are cooperative flying and docking in midair.

Problem 1. *Given a desired attitude $\mathbf{\Theta}^*$ and thrust F for a flying structure $\mathcal{S}$, the problem is to find a control input $\mathbf{u}_i$, for all $i \in \mathcal{S}$, such that the modular flying structure is driven to the attitude $\mathbf{\Theta}^*$ while generating the thrust F.*

Problem 2. *Given a pair of structures $\mathcal{S}_1$ and $\mathcal{S}_2$, find the control inputs $\mathbf{u}_i$, for all $i \in \mathcal{S}_1$ and $\mathbf{u}_j$ for all $j \in \mathcal{S}_2$, such that the structure $\mathcal{S}_1$ docks to the structure $\mathcal{S}_2$ in the three dimensional space.*

We approach Problem 1 and 2 in Section IV and V respectively.

IV. Control of the Flying Modular Structure

In order to allow the flying structure to navigate in a three dimensional environment, we control thrust and attitude to generate vertical and horizontal translations, and rotation in the yaw angle. We assume that the robots know the shape of the structure as well as their location $\mathbf{x}_i^S = [x_i, y_i, z_i]^\top$ in the structure coordinate frame S.

The total thrust can be computed by dividing the total thrust F among all rotors. The contribution for each rotor is given by

$$f_{ij} = \frac{F}{4n}.$$

The controller first generates moments for the structure, then we formulate a modular attitude controller to satisfy those moments.

A. Controlling moments

We want to generate a desired moment $\mathbf{M} = [M_x,\, M_y,\, M_z]^\top$ on the structure $\mathcal{S}$ using the rotor forces f_{ij}. To achieve this, each rotor ij can contribute to the total moment of the structure with $\mathbf{M}_{ij} = [M_{x_{ij}}, M_{y_{ij}}, M_{z_{ij}}]^\top$. The local moments $\mathbf{M}_{ij}$ must satisfy

$$\mathbf{M} = \sum_{ij} \mathbf{M}_{ij}.$$

Since the system is redundant, this equation has infinite solutions. We might distribute the total moment evenly among all actuators, but it would overload the robots that are closer to the center of mass because a high force would be required given the small moment arm. In [14], the authors compute the pseudo inverse of the dynamics. Although this approach minimizes the squared sum of the local moments and forces, it overloads some of the actuators. This is especially problematic when there are many small rotors that can easily saturate. In a similar way, the authors in [13] distribute the forces linearly with respect to the center of mass, overloading and saturating the rotors that are far from the center of rotation. Additionally, the batteries of those modules drain faster than the rotors in the middle. The main problem in these cooperative systems is that if one module fails, the whole structure fails. For these reasons, we want to develop a method that given thrust and moments $[F, \mathbf{M}]^\top$, it minimizes the maximum rotor force as:

$$\mathbf{u}_{\mathcal{S}}^* = \underset{\mathbf{u}_{\mathcal{S}}}{\operatorname{argmin}} \|\mathbf{u}_{\mathcal{S}}\|_\infty, \tag{5}$$

subject to (1), where $\mathbf{u}_{\mathcal{S}}$ is the vector of all control inputs in the structure and $\|\cdot\|_\infty$ denotes the infinity norm, also known as Chebyshev norm, which returns the maximum absolute value in the input vector. In this way, it is possible to generate the maximum performance of the system without saturating the motors. A solution that satisfies the structures dynamics (1) is

$$f_{ij} = \frac{F}{4n} + \frac{M_{x_{ij}}}{y_{ij}} + \frac{M_{y_{ij}}}{x_{ij}} + (-1)^{j+1}\frac{k_F}{k_M} M_{z_{ij}}. \tag{6}$$

The solution for (5) evenly distributes the forces among all rotors. To achieve this, we proportionally distribute the total moments M_x and M_y based on the distance of the actuator to the respective axis. Then, we have

$$M_{x_{ij}} = \frac{|y_{ij}|}{\sum_{ij}|y_{ij}|} M_x, \tag{7}$$

$$M_{y_{ij}} = \frac{|x_{ij}|}{\sum_{ij}|x_{ij}|} M_y. \tag{8}$$

In the case of M_z, all the rotors are in the same plane. Then we can evenly distribute the yaw moment among all rotors as

$$M_{z_{ij}} = \frac{M_z}{4n}. \tag{9}$$

Substituting (7), (8), and (9), in (6), we obtain

$$f_{ij} = \frac{F}{4n} + \frac{\chi(y_{ij})M_x}{\sum_{ij}|y_{ij}|} + \frac{\chi(x_{ij})M_y}{\sum_{ij}|x_{ij}|} + (-1)^{j+1}\frac{k_F M_z}{n\, k_M},$$

where χ denotes the signum function $\chi(x) := x/|x|$. Using the location of the rotor in the robot coordinate frame, we can define the constants of the structure

$$\begin{aligned} C_x &= \sum_{ij}|y_{ij}| = 2\sum_i |y_i + d| + |y_i - d| \\ C_y &= \sum_{ij}|x_{ij}| = 2\sum_i |x_i + d| + |x_i - d| \\ C_z &= \frac{4n\, k_M}{k_F}. \end{aligned}$$

Then, we can write the control input for robot i, in a compact form

$$\mathbf{u}_i = \mathbf{P}_i\, \mathbf{C} \begin{bmatrix} F \\ \mathbf{M} \end{bmatrix}, \tag{10}$$

where the matrix

$$\mathbf{P}_i = \begin{bmatrix} 1 & \chi(y_i - d) & \chi(x_i + d) & 1 \\ 1 & \chi(y_i - d) & \chi(x_i - d) & -1 \\ 1 & \chi(y_i + d) & \chi(x_i - d) & 1 \\ 1 & \chi(y_i + d) & \chi(x_i + d) & -1 \end{bmatrix}$$

has a form $\{1, -1\}^{4\times 4}$ that depends on the location of the ith module (x_i, y_i) in the structure coordinate frame and the distance d. The structure matrix $\mathbf{C} =$

Diag($[1/4n, C_x^{-1}, C_y^{-1}, C_z^{-1}]^\top$) contains the constants of the structure, which is the same for all robots.

We can see in (10) that the rotor forces only depend on the quadrant in the structure coordinate frame. In approaches like [13] and [14], the rotors with higher $|x_{ij}|$ and $|y_{ij}|$ saturate first without maximizing the use of the other rotors. In our approach, the rotors in the same quadrant generate the same force. Therefore, instead of saturating a few rotors, all rotors in the quadrant saturate at the same time. This approach is very efficient for low-cost quadrotors where the force interval $[f_{min}, f_{max}]$ is narrow. However, this approach is not energy efficient and drain the batteries faster than [13] and [14].

B. Decentralized Modular Attitude control

We want to drive a structure $\mathcal{S}$ to a desired attitude $\mathbf{\Theta}^* = [\phi^*, \theta^*, \psi^*]^\top$ and thrust F. In our decentralized setup, the central trajectory planner sends $\mathbf{\Theta}^*$ and F to the structure. Then, each module $i \in \mathcal{S}$ has to independently compute its control input $\mathbf{u}_i$ based on its location in the structure coordinate frame and its local inertial sensor. We use a proportional-derivative controller for a single quadrotor:

$$\begin{aligned} \ddot{\phi}^* &= K_{p,\phi}\,(\phi^* - \phi) + K_{d,\phi}\,(p^* - p), \\ \ddot{\theta}^* &= K_{p,\theta}\,(\theta^* - \theta) + K_{d,\theta}\,(q^* - q), \\ \ddot{\psi}^* &= K_{p,\psi}\,(\psi^* - \psi) + K_{d,\psi}\,(r^* - r), \end{aligned}$$

where $K_{p,\cdot} K_{d,\cdot} > 0$ are gain constants for a single robot. We can write this in a compact form,

$$\dot{\mathbf{\Omega}}^* = \mathbf{K}_p(\mathbf{\Theta}^* - \mathbf{\Theta}) + \mathbf{K}_d(\mathbf{\Omega}^* - \mathbf{\Omega}) \tag{11}$$

where $\mathbf{K}_p$ and $\mathbf{K}_d$ are diagonal matrices that include the gain constants. In this controller, we can set the desired angular velocities to zero $\mathbf{\Omega}^* = \mathbf{0}$. Without re-tuning the gain constants, we want to be able to fly the structure for any configuration and any number of robots. Since all modules in a structure are on the same plane and pointing towards the same direction, each robot is able to use its local sensor to estimate the attitude $\mathbf{\Theta}$ and angular velocities $\mathbf{\Omega}$ of the whole structure $\mathcal{S}$. In this way, we can say that $\mathbf{\Theta}_1 = ... = \mathbf{\Theta}_n = \mathbf{\Theta}$ and $\mathbf{\Omega}_1 = ... = \mathbf{\Omega}_n = \mathbf{\Omega}$. Thus, each module $i \in \mathcal{S}$ is able to independently compute the desired angular accelerations $\dot{\mathbf{\Omega}}^*$ using its local measurements $\mathbf{\Theta}_i, \mathbf{\Omega}_i$ in (11). Maintaining the same angular desired accelerations of a single robot for the multi-robot structure is hard to achieve because inertia grows rapidly with the number of robots (from (4)). Additionally, agility is reduced with the size of the rigid body as it is studied in [16], [17]. Therefore, we can define the desired angular acceleration of the structure $\dot{\mathbf{\Omega}}_{\mathcal{S}}^*$ as a function of the desired acceleration of a single module $\dot{\mathbf{\Omega}}^*$. We propose a function that increases the angular accelerations proportionally to the location of the rotors and inversely proportional to the inertia of the structure,

$$\begin{aligned} \dot{p}_{\mathcal{S}} &= \frac{I_x}{I_x^{\mathcal{S}}}\frac{C_x}{4d}\dot{p} \\ \dot{q}_{\mathcal{S}} &= \frac{I_y}{I_y^{\mathcal{S}}}\frac{C_y}{4d}\dot{q} \\ \dot{r}_{\mathcal{S}} &= \frac{I_z}{I_z^{\mathcal{S}}}C_z\dot{r}. \end{aligned}$$

In this equation, C_x and C_y are related to the locations of all robots in the structure, while $4d$ is related to the distance of the four rotors in a single module. We can rewrite the new desired acceleration of the structure $\mathcal{S}$ in a compact form

$$\dot{\mathbf{\Omega}}_{\mathcal{S}}^* = \mathbf{I}_{\mathcal{S}}^{-1}\mathbf{I}\mathbf{D}\dot{\mathbf{\Omega}}^* \tag{12}$$

where the matrix $\mathbf{D} = \text{Diag}([C_x/4d,\ C_y/4d,\ C_z]^\top)$ contains the constants of the structure. From (10), the control input that satisfies the new desired angular accelerations is described by

$$\mathbf{u}_i = \mathbf{P}_i\,\mathbf{C}\begin{bmatrix} F \\ \mathbf{I}_{\mathcal{S}}\dot{\mathbf{\Omega}}_{\mathcal{S}}^* \end{bmatrix}.$$

Using (12), we can write the control input as a function of the desired angular acceleration of a single module $\dot{\mathbf{\Omega}}^*$:

$$\mathbf{u}_i = \mathbf{P}_i\,\mathbf{E}\begin{bmatrix} F \\ \mathbf{I}\dot{\mathbf{\Omega}}^* \end{bmatrix}, \tag{13}$$

where the matrix $\mathbf{E}$ reformulates $\mathbf{C}$ to satisfy the new angular accelerations as $\mathbf{E} = \text{Diag}([1/4n, 4d, 4d, 4]^\top)$. We highlight that in our modular attitude controller (from (13)), it is not necessary to tune the attitude gains for each configuration of the structure. As a summary, given a control input $[F, \mathbf{\Theta}^*]^\top$, we use (11) to obtain the desired acceleration of a single robot $\dot{\mathbf{\Omega}}^*$. We use the computed $\dot{\mathbf{\Omega}}^*$ and the desired input thrust F to obtain the desired input $\mathbf{u}_i$ based on (13). When a new structure is assembled, each module only needs to know the number of robots in the structure n and its location in the structure to compute $\mathbf{P}_i$.

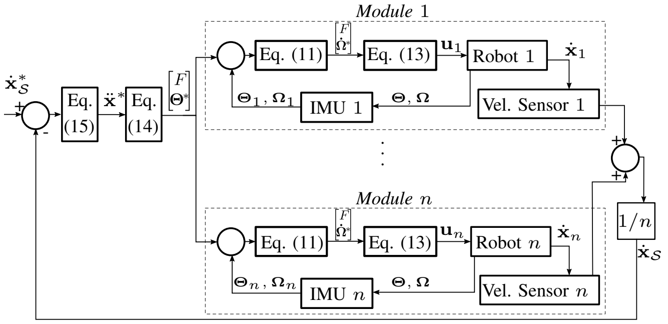
V. Hovering and Docking in midair

In this section, we describe our method to dock a structure $\mathcal{S}_1$ to a structure $\mathcal{S}_2$, resulting a new structure $\mathcal{S}_3 = \mathcal{S}_1 \cup \mathcal{S}_2$. In order to achieve this goal, we initially describe how to control the linear velocity of the structure using the modular attitude controller from previous section. Then, we use the velocity controller to hover in a desired waypoint and to dock pair of structures in midair.

A. Controlling linear velocity

We can control a structure $\mathcal{S}$ as a single quadrotor [14], [18]. All robots have the same desired yaw angle and without loss of generality, we assume that this angle is zero. The desired linear acceleration $\ddot{\mathbf{x}}_{\mathcal{S}}^* = [\ddot{x}_{\mathcal{S}}^*, \ddot{y}_{\mathcal{S}}^*, \ddot{z}_{\mathcal{S}}^*]^\top$ is transformed to the attitude controller $[F, \mathbf{\Theta}^*]^\top$ using

$$\begin{bmatrix} F \\ \mathbf{\Theta}^* \end{bmatrix} = \begin{bmatrix} mg + m\,\ddot{z}_{\mathcal{S}}^* \\ -\ddot{y}_{\mathcal{S}}^*/g \\ \ddot{x}_{\mathcal{S}}^*/g \\ 0 \end{bmatrix}. \tag{14}$$



Module 1
$\dot{\mathbf{x}}_{\mathcal{S}}^*$
+
-
Eq. (15)
$\ddot{\mathbf{x}}^*$
Eq. (14)
$\begin{bmatrix} F \\ \mathbf{\Theta}^* \end{bmatrix}$
Eq. (11)
$\begin{bmatrix} F \\ \dot{\mathbf{\Omega}} \end{bmatrix}$
Eq. (13)
$\mathbf{u}_1$
Robot 1
$\dot{\mathbf{x}}_1$
$\mathbf{\Theta}_1, \mathbf{\Omega}_1$
IMU 1
$\mathbf{\Theta}, \mathbf{\Omega}$
Vel. Sensor 1
+
+
$1/n$
$\dot{\mathbf{x}}_{\mathcal{S}}$
Module n
Eq. (11)
$\begin{bmatrix} F \\ \dot{\mathbf{\Omega}} \end{bmatrix}$
Eq. (13)
$\mathbf{u}_n$
Robot n
$\dot{\mathbf{x}}_n$
$\mathbf{\Theta}_n, \mathbf{\Omega}_n$
IMU n
$\mathbf{\Theta}, \mathbf{\Omega}$
Vel. Sensor n

Fig. 4. Block diagram that illustrates the interaction between the velocity controller and the distributed modular attitude controller for the flying structure.

We can move the robot with a desired velocity $\dot{\mathbf{x}}_{\mathcal{S}}^*$ using

$$\ddot{\mathbf{x}}_{\mathcal{S}}^* = K_a(\dot{\mathbf{x}}_{\mathcal{S}}^* - \dot{\mathbf{x}}_{\mathcal{S}}). \tag{15}$$

Then, we can control the linear velocities by considering the desired velocity as the control input

$$\dot{\mathbf{x}}_{\mathcal{S}}^* = \mathbf{w}_{\mathcal{S}}. \tag{16}$$

The velocity controller and its interaction with the modular attitude controller is illustrated in Figure 4. It starts with the velocity controller, from (16). Its output $\ddot{\mathbf{x}}_{\mathcal{S}}^*$ is transformed into a desired attitude $[F, \mathbf{\Theta}^*]^\top$ for the whole structure using (14). In a decentralized manner, each module i receives the desired attitude and combines it with feedback $\mathbf{\Theta}_i, \mathbf{\Omega}_i$ from its inertial measurement unit (IMU) based on (11). Using the desired angular acceleration of the structure and the desired thrust, $[F, \dot{\mathbf{\Omega}}]^\top$, each module computes its own control input $\mathbf{u}_i$ using (13). In order to close the loop for the velocity controller, we compute the velocity of the structure by averaging the velocity of all modules.

Now, we proceed to dock pairs of flying structures in midair. In our approach, one flying structure waits in a hovering action and the other structure performs the docking action. It reduces the motion errors and disturbances from the hovering structure, so the other structure can precisely align and dock. Both the hovering and the docking actions use the velocity controller as described in the following subsections.

B. Hovering

In the hovering state, we want to maintain the robotic structure in a desired point $\mathbf{x}^*$. We can drive it by using the velocity controller, from (15), as

$$\mathbf{w}_{\mathcal{S}} = K_v(\mathbf{x}_{\mathcal{S}}^* - \mathbf{x}_{\mathcal{S}}). \tag{17}$$

C. Docking action

In order to dock structure $\mathcal{S}_1$ to $\mathcal{S}_2$, we take as reference a pair of modules (i, j), such that $i \in \mathcal{S}_1$ and $j \in \mathcal{S}_2$. Assuming that module i docks to module j, and it docks through the x-axis of R_j. The location of module i in the coordinate frame of module j, R_j, is denoted by $(x_i^{(j)}, y_i^{(j)}, z_i^{(j)})$.

Our docking method is based on a gradient function $\nabla \mathbf{f}$, which is followed by the velocity controller as

$$\mathbf{w}_{\mathcal{S}} = \nabla \mathbf{f}(x_i^{(j)}, y_i^{(j)}, z_i^{(j)}).$$

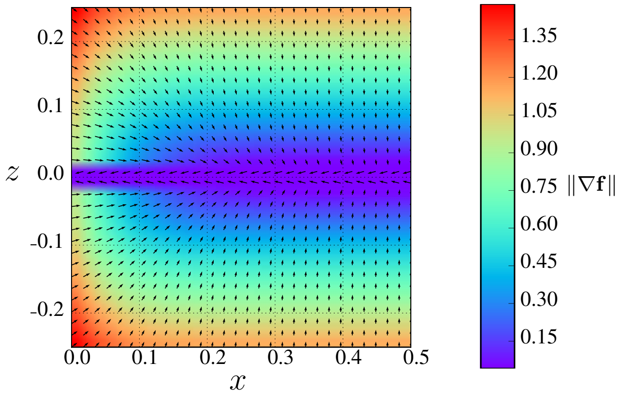


Fig. 5. Gradient function in the $xz-$plane. It is symetric and depicts the same for the $yz-$plane.The arrows represent the direction of the gradient on a given point and the background color represents the magnitude of the gradient.

Our docking function defines what we call *approaching tunnel*, which is a cylinder with radius r aligned with the $x-$axis of R_j. The radius r is related to the motion error that is tolerable during docking, e.g. a small misalignment error ($< 25mm$) is fixed by the attraction force of the magnetic docking mechanism. If the module i is inside the tunnel, it will move towards module j with constant velocity v. Otherwise, meaning that module i is not aligned to module j, the module will rapidly be attracted towards the tunnel before approaching to the other module. If the module is outside the tunnel and close to the yz-plane, the gradient will push it backwards to correct the alignment before approaching.

Our gradient function for docking is given by

$$\nabla \mathbf{f}(x, y, z) = - \begin{bmatrix} g(x)\, x/|x| \\ k_1\, y \\ k_2\, z \end{bmatrix},$$

where the rejection/approaching tunnel function is

$$g(x, y, z) = \begin{cases} -k_3\, e^{-k_4|x|} & \text{if } y^2 + z^2 \geq r, \\ v & \text{otherwise.} \end{cases} \tag{18}$$

Using this function, the module in any arbitrary location will be driven towards the x-axis of R_j (by the proportional controller in the $y-$ and $z-$axis). When the module is close to the $x-$axis, with a distance less than r, within the tunnel, it will be approaching the origin with constant speed v. When the module is outside the tunnel, it is rejected of the $yz-$plane. Using the method, we guarantee that the module is aligned for the docking action, since the module can only approach to the origin if it is within the tunnel. We depict the gradient function $\nabla \mathbf{f}(x, y, z)$ for the $xz-$plane in Figure 5. Since the function is symmetric, the vector field in the $xy-$plane is the same as the vector field in the $xz-$plane. We can see that the only way to approach to the zero is by the purple region, where the module slowly moves and maintains the alignment during the docking action. Theoretically, this gradient-following approach guarantees a successful docking action, since the function $\mathbf{f}$ has a single minimum and the only way to reach it is by a proper alignment.

VI. Experiments

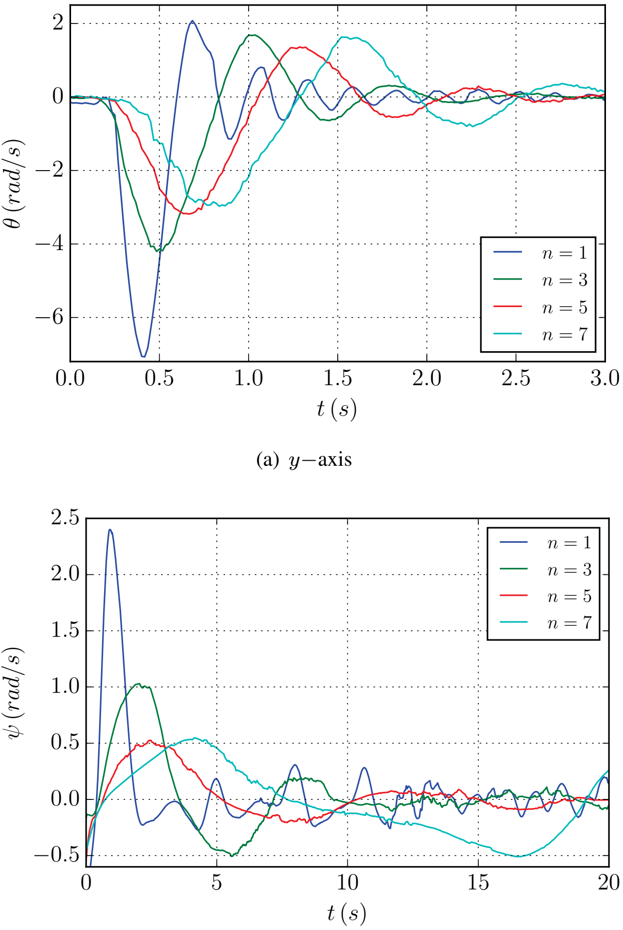
We performed three different experiments to study: the dynamics performance of the modular system; the behavior of the docking action; and the interaction of the flying controller and the docking method for multiple configurations.

In our experimental testbed, we used the Crazyflie-ROS node [19] to control the robots, and to determine the robot pose in space as well as relative locations for docking. We are using a motion capture system (VICON) operating at 100 Hz. We are able to measure the angular velocities of the robots using their IMU sensors. On the computer, we run the velocity (15) and attitude (14) controllers as ROS nodes. The same attitude command is broadcasted to all robots in the structure via 2.4GHz radio. The original firmware of the Crazyflie robot was modified to implement the force distribution from (13). The PD controller from (11) is already implemented in the firmware.

A. Dynamics performance

In this experiment, we evaluate the dynamics performance for multiple modules in a line configuration. We used our modular attitude controller sharing same gains for all modules. The modules are distributed along the x-axis of the structure in such a way that the $x-$axis of the structure and $x-$axis of the modules are aligned. In our setup, we attached the robotic structure to a mechanical system that constrains the system motion allowing only rotation in the pitch angle. We tested the angular velocity in pitch θ for $n = \{1, 3, 5, 7\}$ modules using the same gains. We set the desired pitch angle to zero, $\theta^* = 0$, and an initial angle $\theta = \pi/4$. From the initial angle we released the structure allowing it to achieve its desired angle. Figure 6(a) shows the angular velocities of the structure meanwhile it drives its error to zero. It is possible to observe how the time response is reduced with respect to the number of modules. Using the obtained data, we estimated the angular accelerations of the structure. Based on (12), we can obtain that the angular acceleration in the line configuration is reduced with order $\mathcal{O}(1/n)$, since the inertia grows cubically and C_y grows quadratically in the line configuration. It matches the results of our experiment, since the maximum estimated angular accelerations were $\max(\dot{q}) = \{38.93, 16.52, 9.61, 8.55\}$ (rad/s^2) for $n = \{1, 3, 5, 7\}$ respectively. A similar test were conducted for the yaw angle case (see Figure 6(b)). In comparison to the pitch case, we can see that the increment of modules generates a higher impact in the reduction of the angular velocity. The angular acceleration in yaw is reduced with order $\mathcal{O}(1/n^2)$ (see (12)) because the constant C_z for the yaw angle grows slower than C_x. For this reason, the yaw angle is the most affected when the number of modules in the structure is increased.

We can see that all robots try to converge to the steady state, but the angular acceleration is reduced when we increase the number of robots. A small number of modules are very agile and converge to zero in a faster way. In contrast, a large number of robots present a smoother behavior.



(a) $y-$axis

(b) $z-$axis

Fig. 6. Angular velocities in the $y-$ and $z-$axis for $n = \{1, 3, 5, 7\}$ robots. The modules are arranged in a line configuration. In this experiment the robot structure is released from an angle $\phi = \pi/4$ and (13) drives it to $\phi = 0$.

B. Docking action

In our second experiment, we want to test the behavior of the docking action from multiple initial points. We set a static and a moving module. The module, that performs the docking action, follows the gradient function, from (18), with constants $k_1 = 3.0$, $k_2 = 3.0$, $k_3 = 1.0$, $k_4 = 10.0$ and $v = 0.1$. We present multiple trajectories that the module performs during the docking action from different initial locations (see Figure 7). The docking action is successfully completed even in the presence of motion errors. We can see that the module follows the gradient and always executes the approaching procedure satisfying the alignment requirement. We can also see that the robots are overshooting on the desired z and have to perform an additional curve. However, this behavior and motion errors are always compensated by the docking function. In the experiments we observed that the magnetic field created by the docking mechanism was able to compensate and correct misalignments between two structures generating successful docking even with motion errors.

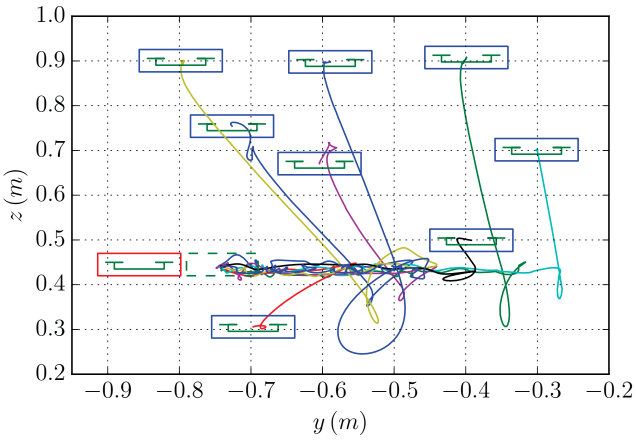


Fig. 7. Robot trajectories during the docking action. The initial locations of the the docking robot are represented by the blue rectangle. This robot moves towards a static module (red rectangle) and docks where the dashed green rectangle is located.

C. Assembling Structures

In our third experiment, we want to test the integration of the flying modular attitude controller and the docking method. We performed the docking experiment for multiple robots and multiple configurations. We implemented a ROS-node that automatically detects when two or multiple robots are attached by using the robot locations from the VICON System. For each detected structure, the matrix $\mathbf{P}_i$ in (13) is uploaded to every module in the structure using radio communication. Once all modules receive the updated parameters, the structure recovers its stability.

In general, we tested our controller for structures with multiple configurations. The angular acceleration in roll and pitch angles is reduced when the number of robots is increased but they are still stable. However, in the case of the yaw angle, due to the low-powerful motors of the Crazyflie, we experimented motor saturations. For this reason, the gains in yaw should be maintained as low as possible. Although the controller maintains the modules stable, the scalability of the system is mainly constrained to the yaw moment that the rotors can offer without saturation. We also observed that the shear stress between connected modules was not an issue due to the high strength-to-weight ratio the docking mechanism provides.

VII. Conclusions and Future Work

This paper introduces ModQuad, a novel flying modular robotic structure that is able to self-assemble in midair and cooperatively fly. One of the main challenges was developing a modular attitude controller in which each module has to use its own inertia sensor to compute its local control input. We implemented this controller in low-cost robots. Although our approach is not energy efficient, it allows us to use low-cost motors with a low maximum thrust. In our approach, the gain constants in our controller do not need to be re-tuned as the configurations change, but the angular acceleration is reduced with the addition of modules.

Using a velocity controller, we are able to dock multiple robots in midair in order to assemble arbitrary structures. The docking system and control has been validated through multiple experiments.

Future work includes examining how the system scales as more modules are attached and rigorously determining the effects of motors saturation on the performance and stability of the structure. In addition, methods and mechanisms for undocking will be explored.

References

[1] J. Werfel and R. Nagpal, “Three-dimensional construction with mobile robots and modular blocks,” *The International Journal of Robotics Research*, vol. 27, no. 3-4, pp. 463–479, 2008.

[2] Q. Lindsey, D. Mellinger, and V. Kumar, “Construction of Cubic Structures with Quadrotor Teams,” *Mechanical Engineering*, 2011.

[3] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, F. Gramazio, M. Kohler, and R. D’Andrea, “The flight assembled architecture installation: Cooperative contruction with flying machines,” *IEEE Control Systems*, vol. 34, no. 4, pp. 46–64, 2014.

[4] S. Murata, H. Kurokawa, and S. Kokaji, “Self-assembling machine,” in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, May 1994, pp. 441–448 vol.1.

[5] B. T. Kirby, B. Aksak, J. D. Campbell, J. F. Hoburg, T. C. Mowry, P. Pillai, and S. C. Goldstein, “A modular robotic system using magnetic force effectors,” in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2007, pp. 2787–2793.

[6] M. J. Doyle, X. Xu, Y. Gu, F. Perez-Diaz, C. Parrott, and R. Gro, “Modular hydraulic propulsion: A robot that moves by routing fluid through itself,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 5189–5196.

[7] J. W. Romanishin, K. Gilpin, and D. Rus, “M-Blocks : Momentum-driven , Magnetic Modular Robots,” pp. 4288–4295, 2013.

[8] L. Murray, J. Timmis, and A. Tyrrell, “Modular self-assembling and self-reconfiguring e-pucks,” *Swarm Intelligence*, vol. 7, no. 2, pp. 83–113, 2013.

[9] M. Rubenstein, A. Cornejo, and R. Nagpal, “Programmable self-assembly in a thousand-robot swarm,” vol. 345, no. 6198, pp. 795–799, 2014.

[10] J. Seo, M. Yim, and V. Kumar, “Assembly sequence planning for constructing planar structures with rectangular modules,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 5477–5482.

[11] D. Saldaña, B. Gabrich, M. Whitzer, A. Prorok, M. F. Campos, M. Yim, and V. Kumar, “A decentralized algorithm for assembling structures with modular robots,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[12] 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.

[13] R. Oung and R. D’Andrea, “The distributed flight array,” *Mechatronics*, vol. 21, no. 6, pp. 908–917, 2011.

[14] 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.

[15] J. A. Preiss, W. Hönig, G. S. Sukhatme, and N. Ayanian, “Crazyswarm: A large nano-quadcopter swarm,” in *Proc. IEEE International Conference on Robotics and Automation*, 2017.

[16] Y. Mulgaonkar, M. Whitzer, B. Morgan, C. M. Kroninger, A. M. Harrington, and V. Kumar, “Power and weight considerations in small, agile quadrotors,” pp. 9083 – 9083 – 16, 2014.

[17] V. Kumar and N. Michael, “Opportunities and challenges with autonomous micro aerial vehicles,” *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1279–1291, 2012.

[18] 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, Sept 2010.

[19] W. Hoenig, C. Milanes, L. Scaria, T. Phan, M. Bolas, and N. Ayanian, “Mixed reality for robotics,” in *IEEE/RSJ Intl Conf. Intelligent Robots and Systems*, Hamburg, Germany, Sept 2015, pp. 5382 – 5387.