

ModQuad-Vi: A Vision-Based Self-Assembling Modular Quadrotor

Guanrui Li¹, Bruno Gabrich¹, David Saldaña¹, Jnaneshwar Das², Vijay Kumar¹, and Mark Yim¹

Abstract—Flying modular robots have the potential to rapidly form temporary structures. In the literature, docking actions rely on external systems and indoor infrastructures for relative pose estimation. In contrast to related work, we provide local estimation during the self-assembly process to avoid dependency on external systems. In this paper, we introduce ModQuad-Vi, a flying modular robot that is aimed to operate in outdoor environments. We propose a new robot design and vision-based docking method. Our design is based on a quadrotor platform with onboard computation and visual perception. Our control method is able to accurately align modules for docking actions. Additionally, we present the dynamics and a geometric controller for the aerial modular system. Experiments validate the vision-based docking method with successful results.

I. INTRODUCTION

Flying modular robots offer a suitable autonomous platform for multiple applications such as: search and rescue [1], cargo lifting [2], and object transportation [3], [4]. In addition, modular robots in a swarm can use their own bodies as building units to assemble large structures [5], [6], [7], [8], [9], [10], [11], [12], [13]. For instance, in a real disaster scenario, human beings trapped in a building surrounded by a flood might be rescued through the use of a temporary bridge or platform formed by flying modular robots. Flying modular robots have the advantage to be able to rapidly move in cluttered environments, flying as individual modules. When at a desired location, they coordinate to form temporary structures. For these applications, robotic systems require the ability to navigate in complex environments with wind, fire and confined spaces.

In literature, related work was developed on modular aerial robots. Zhao et al. [9], [14] presented aerial robots that are transformable in 3-D space. Oung et al. [15] presented a hexagonal flying modular robot with a single propeller that docks on the ground. In this case the controllable flight can only be achieved if at least four modules are docked together. In a recent work, we introduced ModQuad [7], a flying modular quadrotor that can dock in midair forming planar structures with different geometries. However, none of the systems above can perform aerial self-assembly in outdoor environments. Thus, solutions addressing completely

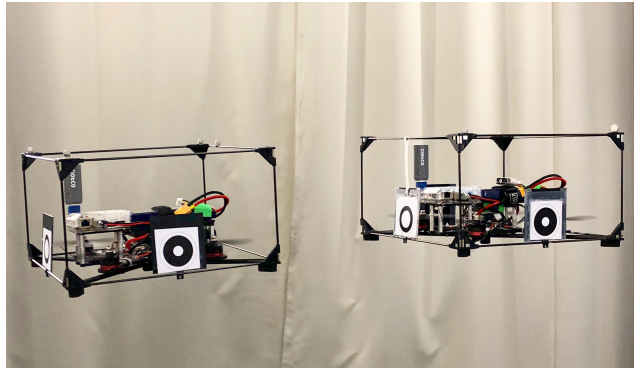


Fig. 1. ModQuad-Vi on a vision-based docking action.

or partially such issue is of great importance to enable self-assembly in outdoor environments.

Flying modules basically require two types of navigation systems: a global localization for navigating in the space and a relative localization for docking actions. The docking method in [7] relies on a motion capture system to compute the relative localization. In order to reduce dependence on motion capture system and make the robots more suitable for outdoor missions, we propose a new robotic platform with: *i*) a GPS system for global navigation and *ii*) a vision-based system for docking actions. In this paper, we focus on the vision-based docking actions, noting that global localization still depends on the motion capture system for testing purposes.

There are two major visual servoing methods: Position Based Visual Servoing (PBVS) and Image Based Visual Servoing (IBVS) [16], [17]. Both methods have been used for visual servo of docking [18], [19], [20] and perching objects [21], [22]. In [23], a geometric model is applied to recover the robot pose using conics, cylinders or other similar objects. An IBVS control law is proposed for a quadrotor that successfully perches on cylinders. In [24] an algorithm is proposed to estimate the pose of the robot based on a set of customized markers located on the docking object.

In this work, we introduce ModQuad-Vi, a new type of modular quadrotor that can dock to other modules of the same type, based on visual servoing. The robot is able to estimate its position and velocity relative to another module based on a black and white roundel. Using PBVS, we developed a control strategy to control the relative position and velocity between two flying modules.

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

² J. Das is with the School of Earth and Space Exploration, Arizona State University, Tempe, AZ, USA: djnan@asu.edu.

The authors acknowledge Bharath Nagarajan and Isabella Orr for their help and support on the robot design and experiments.

We acknowledge the support of the Brazilian agency CAPES, ARL DCIST CRA W911NF-17-2-0181, NSF grants CNS-1521617 and 1138847.

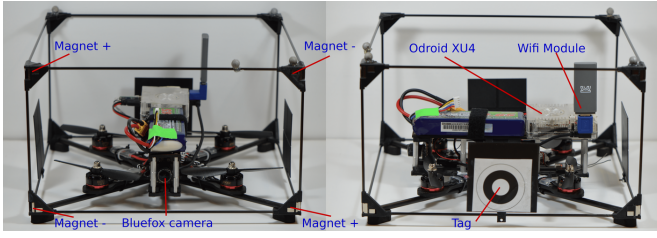


Fig. 2. A Flying Modular Robot. Our design is based in a quadrotor within a cuboid frame. It is equipped with on-board computer, a Bluefox camera and a docking mechanism.

II. MODQUAD-VI DESIGN

In this section, we present the main components of the ModQuad-Vi design. It extends the capabilities of our previous design [7] including additional sensors, computational power and more powerful actuators.

A. Modular Frame

Our design is a cuboid frame, composed of 20 mm diameter carbon fiber tubes and 3-D printed ABS connectors. The dimension of the frame is 320 mm \times 320 mm \times 112 mm and the weight is 160 g. There are three vertical tags on three sides of the frame (left, right and back). The tags are used for relative position and velocity estimation during the docking process.

B. Quadrotor Platform

The platform uses a custom quadrotor. Each rotor can generate a maximum force of 1024 g, while the platform weighs 971 g. The mounted flight controller is the PIXFALCON autopilot with the open source PX4 autopilot firmware. The on-board computer communicates with a ground control station using a Wi-Fi module. The ROS platform [25] runs on the Odroid for controls and image processing. A Matrix Vision mvBlueFOX-MLC200wG camera with a wide angle lens is mounted on-board to transmit the captured images through USB 2.0 at 30 – 40 Hz. The camera provides monocular images with 752 \times 480 resolution.

C. Docking Mechanism

Our docking mechanism is based on square Neodymium (NdFeN) magnets with dimensions of 12.7 mm \times 12.7 mm \times 3.175 mm. The ABS connectors at each vertex contains two magnets. When two modules are docked with 4 pairs, the normal force supports 16.48 kg. This magnetic field corrects misalignment errors up to 1 cm during the docking procedure.

III. MODQUAD DYNAMICS AND CONTROL

In this section, we present the main concepts of the modular aerial system, its dynamics, and our proposed controller.

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.

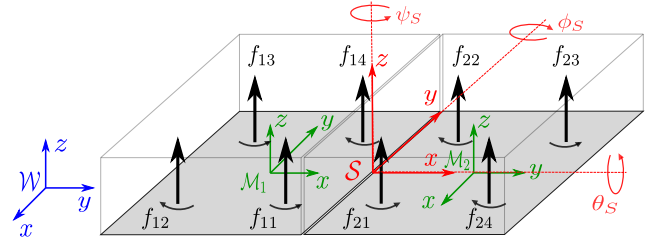


Fig. 3. A pair of modules forming a structure. The arrows represent the forces of the propellers. Note that the robots and the structure do not have to share the same orientation in the z -axis.

All modules are homogeneous with mass m and dimension $w \times w \times h$. A module can rigidly attach to other modules; we define a set of connected modules as follows.

Definition 2 (Structure). A flying structure, \mathcal{R} , 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 .

We define three types of coordinate frames:

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

2) *The module coordinate frame, \mathcal{M}_i :* is defined as the coordinate frame of the module i , which is a right-hand Cartesian frame. The origin of \mathcal{M}_i is attached to the center of mass of i . The angular velocities in the module frame are denoted by $\Omega_i = [p_i, q_i, r_i]^\top$.

3) *The structure coordinate frame, \mathcal{S} :* is defined for a set of attached modules. The origin of \mathcal{S} is located at the center of the mass of the structure. The position of the module i in the structure frame \mathcal{S} is denoted by $[x_i, y_i, z_i]^\top$.

Figure 3 illustrates two attached modules, their associated coordinate frames $\mathcal{M}_1, \mathcal{M}_2$ and the corresponding structure frame \mathcal{S} . The position vector of the structure in \mathcal{W} is denoted by $\mathbf{r}_S \in \mathbb{R}^3$. The linear velocity and acceleration are represented by $\dot{\mathbf{r}}_S$ and $\ddot{\mathbf{r}}_S$. The attitude of the structure is represented by $\Theta_S = [\theta_S, \phi_S, \psi_S]^\top$. The angular velocity and angular acceleration of the structure are $\Omega_S = [p_S, q_S, r_S]^\top$ and $\dot{\Omega}_S = [\dot{p}_S, \dot{q}_S, \dot{r}_S]^\top$. The total thrust, roll, pitch and yaw moments of the structure are denoted as F_S, M_{xS}, M_{yS} and M_{zS} respectively. We assume there are n modules in the structure. Each module has four rotors, indexed by $j = 1, \dots, 4$. The location of the rotor j of the module i in \mathcal{S} is denoted as $[x_{ij}, y_{ij}, z_{ij}]$. The rotor j of module i produces an angular speed ω_{ij} that generates vertical forces and moments

$$f_{ij} = k_f \omega_{ij}^2, \quad M_{ij} = k_m \omega_{ij}^2,$$

where k_m and k_f are motor constants obtained experimentally. The thrust and moments of the structure are results of all rotor forces in the structure.

A. Dynamics

The dynamics of the structure can be written as follows

$$nm\ddot{\mathbf{r}}_S = \begin{bmatrix} 0 \\ 0 \\ -nmg \end{bmatrix} + \mathbf{R}_S^{\mathcal{W}} \begin{bmatrix} 0 \\ 0 \\ F_S \end{bmatrix},$$

$$\mathbf{I}_S \dot{\boldsymbol{\Omega}}_S = \mathbf{M}_S - \boldsymbol{\Omega}_S \times \mathbf{I}_S \boldsymbol{\Omega}_S,$$

where $\mathbf{R}_S^{\mathcal{W}}$ is the orientation of \mathcal{S} with respect to \mathcal{W} , \mathbf{I}_S represents the inertia tensor of the structure and $\mathbf{M}_S = [M_{zS}, M_{yS}, M_{xS}]^\top$. Once the structure is assembled by n modules, its inertia can be computed utilizing the parallel axis theorem

$$\mathbf{I}_S = n\mathbf{I} + m \begin{bmatrix} \sum_i x_i^2 & 0 & 0 \\ 0 & \sum_i y_i^2 & 0 \\ 0 & 0 & \sum_i x_i^2 + y_i^2 \end{bmatrix},$$

where \mathbf{I} is the inertia tensor of a single module.

B. Control

The control consists of a centralized trajectory control, which is based on the geometric control in [26], and a decentralized attitude control [4].

1) *Centralized Trajectory Control*: A desired trajectory, includes position, velocity and acceleration of the center of mass of the structure. We compute the desired acceleration as follows

$$\ddot{\mathbf{r}}_S^* = \mathbf{K}_{pS}(\mathbf{r}_{S,T} - \mathbf{r}_S) + \mathbf{K}_{dS}(\dot{\mathbf{r}}_{S,T} - \dot{\mathbf{r}}_S) + \ddot{\mathbf{r}}_{S,T},$$

where $\mathbf{r}_{S,T}$, $\dot{\mathbf{r}}_{S,T}$, $\ddot{\mathbf{r}}_{S,T}$ are the desired position, velocity and acceleration in the desired trajectory, while $\mathbf{K}_{pS} = \text{Diag}([K_{pxS}, K_{pyS}, K_{pzS}])$ is a diagonal matrix of proportional gains and $\mathbf{K}_{dS} = \text{Diag}[K_{dxS}, K_{dyS}, K_{dzS}]$ is a diagonal matrix of derivative gains. Hence, the desired force of the structure can be computed as

$$\mathbf{f}_S^* = nm(\mathbf{g} + \ddot{\mathbf{r}}_S^*). \quad (1)$$

Then, let the desired thrust F_S be

$$F_S = \mathbf{f}_S^* \cdot \mathbf{R}_S^{\mathcal{W}} \mathbf{e}_3, \quad (2)$$

where $\mathbf{e}_3 = [0, 0, 1]^\top$. The desired rotation matrix of the structure is denoted as follows

$$\mathbf{R}_S^* = [\mathbf{b}_{1S}^* \quad \mathbf{b}_{2S}^* \quad \mathbf{b}_{3S}^*], \quad (3)$$

where \mathbf{b}_{3S}^* should be oriented to the desired thrust direction:

$$\mathbf{b}_{3S}^* = \mathbf{f}_S^* / \|\mathbf{f}_S^*\|. \quad (4)$$

We denote the yaw orientation of the structure along the trajectory as ψ_{S_T} . Based on ψ_{S_T} , we can define a unit vector \mathbf{b}^{yaw} as follows

$$\mathbf{b}^{yaw} = [\cos \psi_{S_T} \quad \sin \psi_{S_T} \quad 0]^\top. \quad (5)$$

Based on \mathbf{b}^{yaw} and \mathbf{b}_{3S}^* , we can then compute \mathbf{b}_{2S}^* and \mathbf{b}_{1S}^* as

$$\mathbf{b}_{2S}^* = \frac{\mathbf{b}_{3S}^* \times \mathbf{b}^{yaw}}{\|\mathbf{b}_{3S}^* \times \mathbf{b}^{yaw}\|}, \quad \mathbf{b}_{1S}^* = \mathbf{b}_{2S}^* \times \mathbf{b}_{3S}^*. \quad (6)$$

2) *Decentralized Attitude Control*: From (3), we obtain the desired rotation matrix of the structure. The modules do not necessarily share the same attitude as the structure. Hence, we derive a desired rotation matrix of the module, which can be computed as

$$\mathbf{R}_i^* = \mathbf{R}_S^* \mathbf{R}_{\mathcal{M}_i}^S, \quad (7)$$

where $\mathbf{R}_{\mathcal{M}_i}^S$ is the rotation matrix from \mathcal{S} to \mathcal{M}_i . After that, we can then obtain the desired angular acceleration of module i as follows

$$\dot{\boldsymbol{\Omega}}_i^{\mathcal{M}_i} = [\dot{\Omega}_{xi}^{\mathcal{M}_i}, \dot{\Omega}_{yi}^{\mathcal{M}_i}, \dot{\Omega}_{zi}^{\mathcal{M}_i}]^\top = -\mathbf{K}_{Ri} \mathbf{e}_{Ri} - \mathbf{K}_{\Omega i} \mathbf{e}_{\Omega i}, \quad (8)$$

where \mathbf{e}_{Ri} and $\mathbf{e}_{\Omega i}$ are error vectors of the orientation and angular velocity respectively. The errors can be computed as

$$\mathbf{e}_{Ri} = \frac{1}{2}(\mathbf{R}_i^{*\top} \mathbf{R}_i - \mathbf{R}_i^\top \mathbf{R}_i^*)^\vee, \quad \mathbf{e}_{\Omega i} = \boldsymbol{\Omega}_i - \boldsymbol{\Omega}_i^*,$$

where \vee is the ‘vee’ map defined in [26] and the desired angular velocity $\boldsymbol{\Omega}_i^*$ can either be generated by the trajectory or set to zero. Since $\dot{\boldsymbol{\Omega}}_i^{\mathcal{M}_i}$ is obtained in \mathcal{M}_i , we then need to generate the angular acceleration for module i in \mathcal{S} . Based on that, we apply the following transformation

$$\dot{\boldsymbol{\Omega}}_i^S = \mathbf{R}_{\mathcal{M}_i}^S \dot{\boldsymbol{\Omega}}_i^{\mathcal{M}_i}.$$

where $\dot{\boldsymbol{\Omega}}_i^S$ is the angular acceleration of module i with respect to the frame \mathcal{S} . Using the computed $\dot{\boldsymbol{\Omega}}_i^S$, and considering all modules are rigidly attached to each other, we obtain the moment for the structure,

$$\mathbf{M}_S = \mathbf{I}_S \dot{\boldsymbol{\Omega}}_i^S \quad (9)$$

With this approach, we highlight that given a desired trajectory for the structure, $\mathbf{r}_{S,T}$, $\dot{\mathbf{r}}_{S,T}$, $\ddot{\mathbf{r}}_{S,T}$, we can generate the total control moment locally in each module in the structure in a decentralized manner.

3) *Force Distribution*: We use the method shown in [7] to distribute the forces among the rotors. This approach evenly distributes battery usage for all modules in the structure, and at the same time minimizes the maximum force needed to achieve a desired moment. The equal force distribution can be written as follows,

$$f_{ij} = \frac{F_S}{4n} + \frac{\chi(y_{ij})}{\sum_{ij} |y_{ij}|} M_{xS} + \frac{\chi(x_{ij})}{\sum_{ij} |x_{ij}|} M_{yS} + (-1)^{j+1} \frac{k_f}{k_m} \frac{M_{zS}}{4n},$$

where $\chi(x) = x/|x|$. We can write the rotor forces for the i th module in a matrix form as follow

$$\begin{bmatrix} f_{i1} \\ f_{i2} \\ f_{i3} \\ f_{i4} \end{bmatrix} = \begin{bmatrix} \frac{1}{4n} & c_{xi1} & c_{yi1} & \frac{k_f}{4nk_m} \\ \frac{1}{4n} & c_{xi2} & c_{yi2} & -\frac{k_f}{4nk_m} \\ \frac{1}{4n} & c_{xi3} & c_{yi3} & \frac{k_f}{4nk_m} \\ \frac{1}{4n} & c_{xi4} & c_{yi4} & -\frac{k_f}{4nk_m} \end{bmatrix} \begin{bmatrix} F_S \\ M_{xS} \\ M_{yS} \\ M_{zS} \end{bmatrix},$$

where

$$c_{xij} = \frac{\chi(y_{ij})}{\sum_{ij} |y_{ij}|}, \quad c_{yij} = \frac{\chi(x_{ij})}{\sum_{ij} |x_{ij}|}. \quad (10)$$

During the assembling process, the center of mass of the structure shifts whenever new modules dock to it. This

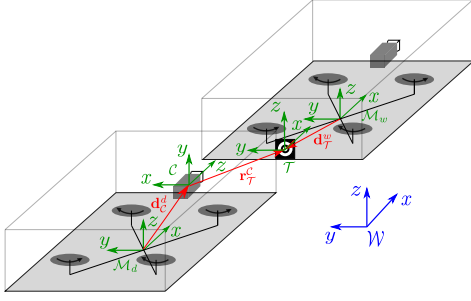


Fig. 4. The z -axis of the camera frame \mathcal{C} points to the front of the module aligning with the x -axis of \mathcal{M}_d . The y -axis of the camera frame \mathcal{C} points up and the x -axis is the cross product of the y and z axis.

process leads to a change in x_{ij}, y_{ij} . Hence, c_{xij} and c_{yij} in (10) are dynamic variables in the assembling process. In this work, we name them as mixer coefficients. Each module in the structure has its mixer coefficients updated whenever docking takes place. The procedure to update these variables will be explained in Section V.

IV. VISUAL DOCKING

Based on the definitions presented in Section III, we can further define Docking module and Waiting module:

Definition 3 (Waiting module). A Waiting module is a module that hovers at a fixed position in space.

Definition 4 (Docking module). A Docking module is a module that is moving towards the Waiting module horizontally in order to dock.

We denote the Docking module as d and the Waiting module as w . The coordinate frame of d is denoted as \mathcal{M}_d and the coordinate frame of w as \mathcal{M}_w . Since there are three tag panels in each module, we define the p th tag in the module as t_p and the coordinate frame of t_p as \mathcal{T}_p , where $p = \{1, 2, 3\}$. As shown in Figure 4, the camera frame \mathcal{C} is defined as a coordinate frame fixed at the Docking module. The origin of \mathcal{C} is located at the center of the camera.

A. Visual Estimation

The position of the Docking module in \mathcal{M}_w is denoted by $\mathbf{r}_d^{\mathcal{M}_w} = [x_d^{\mathcal{M}_w}, y_d^{\mathcal{M}_w}, z_d^{\mathcal{M}_w}]^\top$. The velocity and acceleration of the Docking module in \mathcal{M}_w are $\dot{\mathbf{r}}_d^{\mathcal{M}_w} = [\dot{x}_d^{\mathcal{M}_w}, \dot{y}_d^{\mathcal{M}_w}, \dot{z}_d^{\mathcal{M}_w}]^\top$, $\ddot{\mathbf{r}}_d^{\mathcal{M}_w} = [\ddot{x}_d^{\mathcal{M}_w}, \ddot{y}_d^{\mathcal{M}_w}, \ddot{z}_d^{\mathcal{M}_w}]^\top$ correspondingly.

1) *Position Estimation:* The position $\mathbf{r}_d^{\mathcal{M}_w}$ of the Docking module in \mathcal{M}_w can be expressed by

$$\mathbf{r}_d^{\mathcal{M}_w} = \mathbf{d}_{\mathcal{T}_p}^{\mathcal{M}_w} + \mathbf{R}_{\mathcal{T}_p}^{\mathcal{M}_w} \mathbf{r}_d^{\mathcal{T}_p}, \quad (11)$$

where $\mathbf{d}_{\mathcal{T}_p}^{\mathcal{M}_w}$ is the translation vector of \mathcal{T}_p with respect to \mathcal{M}_w , $\mathbf{R}_{\mathcal{T}_p}^{\mathcal{M}_w}$ is the rotation matrix from \mathcal{M}_w to \mathcal{T}_p and $\mathbf{r}_d^{\mathcal{T}_p}$ is the position vector of the Docking module with respect to \mathcal{T}_p . Both $\mathbf{d}_{\mathcal{T}_p}^{\mathcal{M}_w}$ and $\mathbf{R}_{\mathcal{T}_p}^{\mathcal{M}_w}$ are invariant, since the tag panel is fixed relative to the module. The $\mathbf{r}_d^{\mathcal{T}_p}$ in (11) can be expressed by

$$\mathbf{r}_d^{\mathcal{T}_p} = -\mathbf{R}_{\mathcal{M}_d}^{\mathcal{T}_p} \mathbf{r}_{t_p}^{\mathcal{M}_d}, \quad (12)$$

where $\mathbf{R}_{\mathcal{M}_d}^{\mathcal{T}_p}$ is the rotation matrix from \mathcal{T}_p to \mathcal{M}_d and $\mathbf{r}_{t_p}^{\mathcal{M}_d}$ is the position vector of the tag panel with respect to \mathcal{M}_d . In order to compute $\mathbf{r}_{t_p}^{\mathcal{M}_d}$ in (12), we first need to obtain $\mathbf{r}_{t_p}^{\mathcal{C}}$, which is the position of the tag in \mathcal{C} . Here, we use the ‘‘Whycon’’ algorithm [27] to obtain $\mathbf{r}_{t_p}^{\mathcal{C}}$, which is based on a fast and precise detection of a black and white roundel. Comparing with other existed vision-based localization methods that use planar printable patterns, like ARTag [28], ARToolkit [29], ‘‘Whycon’’ is faster without losing the precision, which is preferred in scenarios requiring fast and accurate odometry feedback. Detailed comparisons are available in [27].

Knowing the transformation from \mathcal{C} to \mathcal{M}_d based on the design of the robot, the $\mathbf{r}_{t_p}^{\mathcal{M}_d}$ can be obtained as follows

$$\mathbf{r}_{t_p}^{\mathcal{M}_d} = \mathbf{d}_{\mathcal{C}}^{\mathcal{M}_d} + \mathbf{R}_{\mathcal{C}}^{\mathcal{M}_d} \mathbf{r}_{t_p}^{\mathcal{C}}, \quad (13)$$

where $\mathbf{d}_{\mathcal{C}}^{\mathcal{M}_d}$ is the translation vector of the camera frame \mathcal{C} in \mathcal{M}_d and $\mathbf{R}_{\mathcal{C}}^{\mathcal{M}_d}$ is the rotation matrix from \mathcal{M}_d to \mathcal{C} . Applying (12) and (13) in (11), we can obtain

$$\mathbf{r}_d^{\mathcal{M}_w} = \mathbf{d}_{\mathcal{T}_p}^{\mathcal{M}_w} - \mathbf{R}_{\mathcal{T}_p}^{\mathcal{M}_w} \mathbf{R}_{\mathcal{M}_d}^{\mathcal{T}_p} (\mathbf{d}_{\mathcal{C}}^{\mathcal{M}_d} + \mathbf{R}_{\mathcal{C}}^{\mathcal{M}_d} \mathbf{r}_{t_p}^{\mathcal{C}}). \quad (14)$$

Due to symmetry of the tag, its yaw orientation cannot be obtained through the detector. Hence it is difficult to fully obtain $\mathbf{R}_{\mathcal{M}_d}^{\mathcal{T}_p}$ in (14) directly. To solve this issue we apply the following procedure

$$\mathbf{R}_{\mathcal{T}_p}^{\mathcal{M}_w} \mathbf{R}_{\mathcal{M}_d}^{\mathcal{T}_p} = \mathbf{R}_{\mathcal{M}_d}^{\mathcal{M}_w} = (\mathbf{R}_{\mathcal{M}_w}^{\mathcal{W}})^\top \mathbf{R}_{\mathcal{M}_d}^{\mathcal{W}}, \quad (15)$$

where $\mathbf{R}_{\mathcal{M}_d}^{\mathcal{M}_w}$ is a rotation matrix transforming from \mathcal{M}_w to \mathcal{M}_d , $\mathbf{R}_{\mathcal{M}_w}^{\mathcal{W}}$ and $\mathbf{R}_{\mathcal{M}_d}^{\mathcal{W}}$ are the rotation matrices representing the attitude of the Waiting module and the Docking module with respect to \mathcal{W} . Both rotations can be obtained from the local orientation estimator on both modules. Therefore, we can rewrite (14) as follows

$$\mathbf{r}_d^{\mathcal{M}_w} = \mathbf{d}_{\mathcal{T}_p}^{\mathcal{M}_w} - (\mathbf{R}_{\mathcal{M}_w}^{\mathcal{W}})^\top \mathbf{R}_{\mathcal{M}_d}^{\mathcal{W}} (\mathbf{d}_{\mathcal{C}}^{\mathcal{M}_d} + \mathbf{R}_{\mathcal{C}}^{\mathcal{M}_d} \mathbf{r}_{t_p}^{\mathcal{C}}). \quad (16)$$

All the variables in (16) can be obtained from the design, local attitude estimator and the ‘‘Whycon’’.

2) *Kalman Filter:* In order to estimate the linear velocity, we propose a Kalman Filter that uses the position estimation in (16) as well as the accelerometer data from the IMUs. With this Kalman Filter, the position estimation rate is also increased.

The estimated state vector is defined as follows

$$\mathbf{x} = [\mathbf{x}_{1t} \quad \mathbf{x}_{2t} \quad \mathbf{x}_{3t}]^\top = [\mathbf{r}_d^{\mathcal{M}_w} \quad \dot{\mathbf{r}}_d^{\mathcal{M}_w} \quad \mathbf{b}_a]^\top, \quad (17)$$

where \mathbf{b}_a is the bias vector of the accelerometer.

Process Model: The accelerometer measurement \mathbf{a}_m is defined as

$$\mathbf{a}_m = \mathbf{R}_{\mathcal{W},t}^{\mathcal{M}_d} (\ddot{\mathbf{r}}_d^{\mathcal{W}} + \mathbf{g}) + \mathbf{b}_a + \mathbf{n}_a \quad (18)$$

Rearranging (18), we compute the linear acceleration of the docking module in \mathcal{W} as

$$\ddot{\mathbf{r}}_d^{\mathcal{W}} = -\mathbf{g} + \mathbf{R}_{\mathcal{M}_d,t}^{\mathcal{W}} (\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a),$$

where $\mathbf{g} = [0, 0, -9.8]^\top$ is the gravity acceleration vector in \mathcal{W} , $\mathbf{R}_{\mathcal{M}_d,t}^{\mathcal{W}}$ is the orientation of \mathcal{M}_d with respect to \mathcal{W}

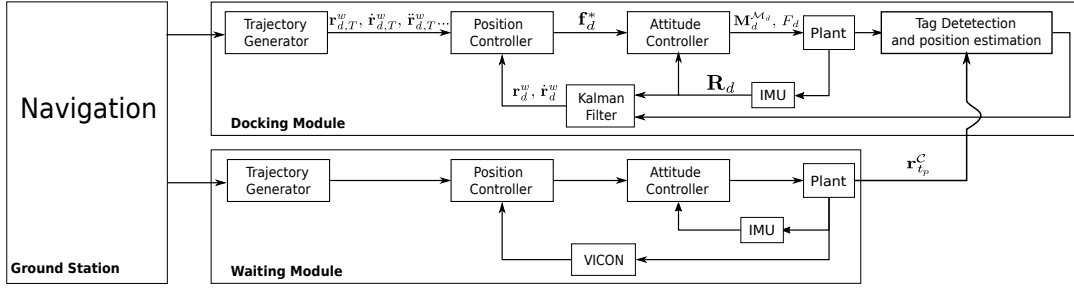


Fig. 5. Controller diagram for the Docking module and Waiting module in a docking procedure.

at time t and $\mathbf{n}_a \sim N(0, Q_{\mathbf{n}_a})$ is an additive Gaussian white noise. In the model, we assumed that the drift in the accelerometer bias vector is a Gaussian white noise defined by $\dot{\mathbf{b}}_a = \mathbf{n}_{ba}(t)$, where $\mathbf{n}_{ba} \sim N(0, Q_{\mathbf{n}_{ba}})$.

The Waiting module holds its position and orientation in the inertial frame. Hence, we could assume that $\mathbf{R}_{\mathcal{W}}^{\mathcal{M}_w}$ is a invariant rotation that is used in the filter. Thus, we can express $\ddot{\mathbf{r}}_d^{\mathcal{M}_w}$ as follows

$$\ddot{\mathbf{r}}_d^{\mathcal{M}_w} = \mathbf{R}_{\mathcal{W}}^{\mathcal{M}_w}(-\mathbf{g} + \mathbf{R}_{\mathcal{M}_d, t}^{\mathcal{W}}(\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a))$$

and the first derivative of (17) as

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{x}_{2t} \\ \mathbf{R}_{\mathcal{W}}^{\mathcal{M}_w}(-\mathbf{g} + \mathbf{R}_{\mathcal{M}_d, t}^{\mathcal{W}}(\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a)) \\ \mathbf{n}_{ba} \end{bmatrix}.$$

Measurement Model: As shown in Section IV-A.1, we can measure $\mathbf{r}_d^{\mathcal{M}_w}$ with the camera. Hence, our measurement is linear as shown below

$$\mathbf{r}_d^{\mathcal{M}_w} = \mathbf{z} = \mathbf{C}\mathbf{x} + \mathbf{v},$$

where $\mathbf{C} = [\mathbf{I}_{3 \times 3} \quad \mathbf{0} \quad \mathbf{0}]$, and \mathbf{v} is an additive Gaussian white noise defined as $\mathbf{v} \sim N(0, Q_{\mathbf{v}})$.

B. Control

The control method for the docking action of the Docking module is very similar to the one presented in III-B.2, with the difference that our new reference is \mathcal{M}_w instead of \mathcal{W} . Therefore, we can write

$$\ddot{\mathbf{r}}_d^{\mathcal{M}_w^*} = \mathbf{K}_p(\mathbf{r}_{d, T}^{\mathcal{M}_w} - \mathbf{r}_d^{\mathcal{M}_w}) + \mathbf{K}_d(\dot{\mathbf{r}}_{d, T}^{\mathcal{M}_w} - \dot{\mathbf{r}}_d^{\mathcal{M}_w}) + \ddot{\mathbf{r}}_{d, T}^{\mathcal{M}_w},$$

where $\mathbf{r}_{d, T}^{\mathcal{M}_w}$, $\dot{\mathbf{r}}_{d, T}^{\mathcal{M}_w}$, $\ddot{\mathbf{r}}_{d, T}^{\mathcal{M}_w}$ are the relative position, velocity and acceleration along the desired trajectory. To obtain $\ddot{\mathbf{r}}_d^{\mathcal{M}_w^*}$ we can premultiply $\ddot{\mathbf{r}}_d^{\mathcal{M}_w^*}$ by $\mathbf{R}_{\mathcal{M}_w}^{\mathcal{W}}$. We then use (1), (2), (3), (5) (8) and (9) along with $\psi_{d, T}$ and \mathbf{I} , to compute F_d and $\mathbf{M}_d^{\mathcal{M}_d}$.

V. EXPERIMENT AND ANALYSIS

We performed experiments to evaluate our new docking method with the ROS using MAVROS nodes. These experiments mainly focused on docking and cooperative flight.

In the docking experiment, a waiting module hovers at a fixed position with desired attitude $\Theta_w = [0, 0, 0]^T$. Then, the docking module docks to the back side of the Waiting module. This experiment consists of three main actions:

1. Hover: Both modules take off and hover at a position based on the odometry feedback from the MOCAP system. Once the “Whycon” algorithm in the Docking module has a consistent tag detection, the ground control station sends the “Track” command to the docking module.

2. Track: The docking module receives $[-1.0, 0.0, 0.2]^T$ as a desired position in \mathcal{M}_w and a desired yaw angle of 0 with respect to the Waiting module. The desired velocity and acceleration are both $\mathbf{0}$. A quintic trajectory is then generated from the current state to the desired state mentioned above. The orientation estimation is obtained from the PX4 orientation estimator which fuses the yaw measurements from the MOCAP system. Although, we remark that the yaw measurement can be easily replaced by a GPS-based or a vision-based navigation system. If the Docking module holds a stable position relative to the waiting module, the ground control station sends a “Dock” command to the Docking module.

3. Dock: In this action, the Docking module generates a quintic trajectory between the current relative position and the final position of $[-0.32, 0, 0]^T$ in \mathcal{M}_w . The 0.32 m is the designed side length of the module. The desired yaw angle remains 0 with respect to \mathcal{M}_w . Based on this command, the docking module approaches to the Waiting module in order to dock. The magnets on both modules help to correct the position error and attract them facilitating the docking. A docked state detector runs on-board to compute the distance d between the two modules, which is the L_2 -norm of \mathbf{x}_{1t} in the state vector of the filter. If the distance between the modules satisfies $d \leq 0.32 m$, then the two modules are considered docked.

Once the modules are docked, we reassign the mixer coefficients in (10), based on the rotor position in \mathcal{S} . The coefficients are computed in ROS and sent to the on-board PIXFALCON through serial communication. We customized the PX4 firmware to modify the coefficients in the registers along with the registers where it stores thrust and moments. The mixer in the flight controller obtains the coefficients on the registers and generates the individual motor commands. The two modules fly cooperatively together based on the control developed in Section III with the odometry feedback from the MOCAP system. During the **Track** and **Dock** stages, the Docking module might lose sight of the tag if it moves out of the camera field of view. Another cause of lost tag

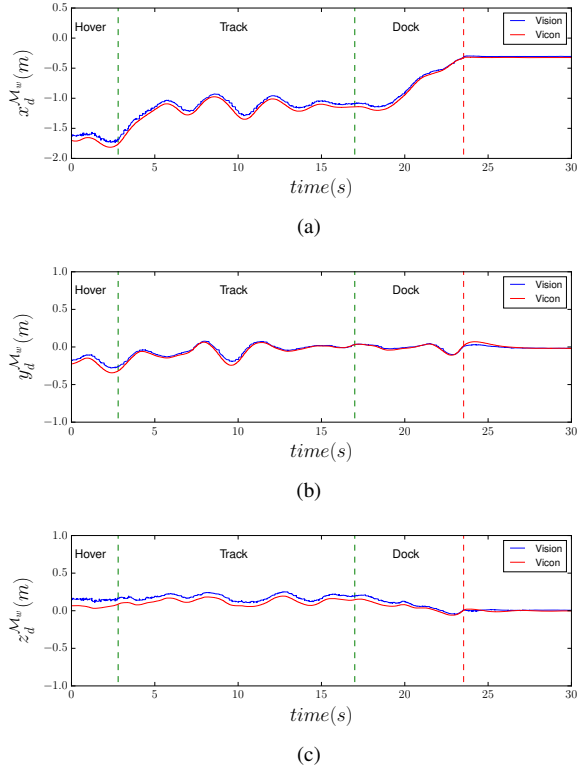


Fig. 6. Relative position of the Docking module d with respect to the \mathcal{M}_w .

vision image is blur caused by fast motions. If the Docking module loses sight of a tag during the steps mentioned above, the estimation results from the Kalman filter drifts badly. To protect the robot, the Docking module skips these stages and hovers at the position where it lost the tag using the MOCAP system feedback.

Figures 6, 7, and 8 show successful docking data, estimation of the relative position and relative velocity as well as acceleration measurements respectively. There are three main time stages in the figures, separated by dashed lines. The right hand side of the red dash line is the portion where the modules have docked and the new mixer coefficients have been assigned to both modules. In Figure 6(a) - 6(c), it is possible to observe that the visual estimation becomes constant, confirming that both modules are docked. Another evidence of docking success is the spikes observed in the acceleration measurement illustrated as a red dash line in Figure 8. The spikes indicate the existence of collision between the two modules. In Figure 7(c), we notice oscillations in the visual velocity estimation around ground truth measurement, which is caused by the fluctuation (see Figure 8) in the accelerometer measurement along the z -axis. This fluctuation in the accelerometer is caused by a descending motion of the structure after the mixer coefficients have been reassigned. A similar behaviour is also observed in our previous work [7], which can be explained by the abrupt change in the rotors behavior due to the update of the mixer coefficients.

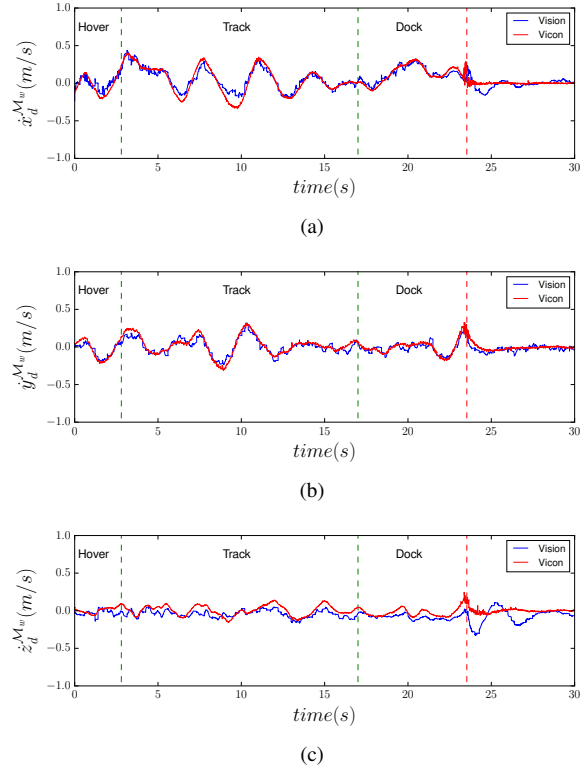


Fig. 7. Relative velocity of the Docking module d with respect to the \mathcal{M}_w .

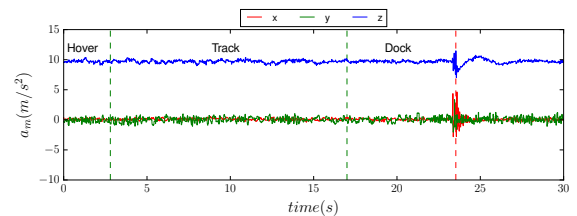


Fig. 8. Accelerometer measurements in the Docking module.

VI. CONCLUSIONS AND FUTURE WORK

In conclusion, we present ModQuad-Vi, a team of vision-based flying modular robots that are able to self-assemble in midair. This work focuses on the estimation and visual servoing method for the docking procedure which struggles with the constraints of coupling air frame attitude and camera field of view. We first briefly introduce the dynamics and geometric control of the flying structure. We then develop the estimation of the relative position and velocity between modules based on vision and IMU feedback. Finally we develop a visual servoing method to control the relative position and velocity of the robot with respect to one another.

In the future, we plan to continue developing ModQuad-Vi to work in outdoor environments. Our proposed robot design includes the necessary elements to include additional sensors. For instance, integrating monocular VIO or GPS for the global position and yaw measurement. In addition, we want to study how the system scales.

REFERENCES

- [1] 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.
- [2] 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.
- [3] 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.
- [4] B. Gabrich, D. Saldaña, V. Kumar, and M. Yim, "A flying gripper based on cuboid modular robots," in *IEEE International Conference on Robotics and Automation 2018*, Brisbane, Australia, 2018.
- [5] I. O'Hara, J. Paulos, J. Davey, N. Eckenstein, N. Doshi, T. Tosun, J. Greco, J. Seo, M. Turpin, V. Kumar, and M. Yim, "Self-assembly of a swarm of autonomous boats into floating structures," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1234–1240, 2014.
- [6] 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.
- [7] D. Saldaña, B. Gabrich, G. Li, M. Yim, and V. Kumar, "Modquad: The flying modular structure that self-assembles in midair," in *IEEE International Conference on Robotics and Automation 2018*, May 2018.
- [8] 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. [Online]. Available: <http://dx.doi.org/10.1177/0278364907084984>
- [9] 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, April 2018.
- [10] 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.
- [11] B. T. Kirby, B. Aksak, J. D. Campbell, J. F. Hoberg, 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.
- [12] 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.
- [13] 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.
- [14] M. Zhao, K. Kawasaki, X. Chen, S. Noda, K. Okada, and M. Inaba, "Whole-body aerial manipulation by transformable multirotor with two-dimensional multilinks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 5175–5182.
- [15] R. Oung and R. D'Andrea, "The distributed flight array," *Mechatronics*, vol. 21, no. 6, pp. 908–917, 2011.
- [16] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics Automation Magazine*, vol. 13, no. 4, pp. 82–90, Dec 2006.
- [17] —, "Visual servo control. ii. advanced approaches," *IEEE Robotics Automation Magazine*, vol. 14, no. 1, pp. 109–118, March 2007.
- [18] M. Myint, K. Yonemori, A. Yanou, M. Minami, and S. Ishiyama, "Visual-servo-based autonomous docking system for underwater vehicle using dual-eyes camera 3d-pose tracking," in *2015 IEEE/SICE International Symposium on System Integration (SII)*, Dec 2015, pp. 989–994.
- [19] M. F. Yahya and M. R. Arshad, "Position-based visual servoing for underwater docking of an autonomous underwater vehicle," in *2016 IEEE International Conference on Underwater System Technology: Theory and Applications (USYS)*, Dec 2016, pp. 121–126.
- [20] P.-M. Lee, B.-H. Jeon, and S.-M. Kim, "Visual servoing for underwater docking of an autonomous underwater vehicle with one camera," in *Oceans 2003. Celebrating the Past ... Teaming Toward the Future (IEEE Cat. No.03CH37492)*, vol. 2, Sept 2003, pp. 677–682 Vol.2.
- [21] J. Thomas, M. Pope, G. Loianno, E. W. Hawkes, M. A. Estrada, H. Jiang, M. R. Cutkosky, and V. Kumar, "Aggressive Flight for Perching on Inclined Surfaces," *Journal of Mechanisms and Robotics*, vol. 8, no. October, pp. 1–10, 2015.
- [22] N. INABA, M. ODA, and M. HAYASHI, "Visual servoing of space robot for autonomous satellite capture," *TRANSACTIONS OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES*, vol. 46, no. 153, pp. 173–179, 2003.
- [23] J. Thomas, G. Loianno, K. Daniilidis, and V. Kumar, "Visual servoing of quadrotors for perching by hanging from cylindrical objects," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 57–64, Jan 2016.
- [24] S. Cho, S. Huh, and D. Shim, "Visual detection and servoing for automated docking of unmanned spacecraft," vol. 12, pp. a107–a116, 01 2014.
- [25] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," vol. 3, 01 2009.
- [26] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se(3)," in *49th IEEE Conference on Decision and Control (CDC)*, Dec 2010, pp. 5420–5425.
- [27] T. Krajník, M. Nitsche, J. Faigl, P. Vaněk, M. Saska, L. Přeučil, T. Duckett, and M. Mejail, "A practical multirobot localization system," *Journal of Intelligent & Robotic Systems*, 2014.
- [28] M. Fiala, "Artag, a fiducial marker system using digital techniques," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, June 2005, pp. 590–596 vol. 2.
- [29] D. Wagner and D. Schmalstieg, "Artoolkitplus for pose tracking on mobile devices," 01 2007.