Article

The International Journal of Robotics Research
2014, Vol. 33(3) 375–400

DOI: 10.1177/0278364913501212
ijr.sagepub.com

The Distributed Flight Array: Design, implementation, and analysis of a modular vertical take-off and landing vehicle

Raymond Oung and Raffaello D'Andrea

Abstract

We describe the design and implementation of a modular vertical take-off and landing vehicle, which can be assembled and flown in an unlimited number of arbitrary configurations. This vehicle is intended to be used as a testbed for evaluating distributed estimation and control algorithms. We present the custom hardware, dynamics model, method of state estimation, and a parameterized control strategy capable of controlling any flight-feasible configuration of the vehicle. In terms of the control strategy, we describe a method for optimizing its parameters in order to achieve the best possible performance subject to the system's physical constraints. We then show a straightforward method of mapping the configuration space of our vehicle to the control parameter space. Experimental results are included, demonstrating flight for a variety of configurations both indoors and outdoors.

Keywords

Aerial robotics, mobile and distributed robotics, cellular and modular robots, networked robots, control architectures and programming

1. Introduction

In this article we describe the design and implementation of a modular reconfigurable vertical take-off and landing (VTOL) vehicle, namely the Distributed Flight Array (DFA), with experimental results demonstrating its ability to fly in an unlimited number of arbitrary configurations (see Figure 1). The key innovations of our work include: (1) creation of the first hardware prototype to truly achieve modular reconfigurable coordinated flight and (2) a generalized scalable methodology for controlling any flight-feasible configuration of a multi-rotor VTOL vehicle.

The DFA is more than just a hardware proof of concept. It is a platform that is rich with real-world engineering problems and serves as a testbed for developing algorithms in, but not limited to, distributed estimation and control. Our aim is to develop algorithms with the expectation that the number of agents (or modules) in our system will grow and that these algorithms function robustly on the physical hardware. The motivation behind this is that it is now rapidly becoming economically viable to embed powerful computers, sensors, and communication technology into almost any physical device. Distributed algorithms that are scalable and robust will be needed to act on the vast quantities of information produced by these devices.

The technological advances made in the last decade have been the primary enabler for both modular robotics (see Gilpin and Rus (2010); Murata and Kurokawa (2007); Yim et al. (2007) and references therein) and micro aerial vehicles (see Bouabdallah et al. (2004); Hoffmann et al. (2004); How et al. (2008); Kumar and Michael (2012); Mahony et al. (2012); Meier et al. (2011); Michael et al. (2010) and references therein)—research fields that were initiated a little more than decade ago and that have seen much progress since. Our work draws inspiration from the research challenges existing in both areas. In particular, we are motivated by developing and applying distributed estimation and control algorithms to dynamic modular systems.

In the first part of this work, beginning with Section 2, we will describe in detail the mechanical, electrical, and software design of the DFA. This will be followed by a description of the vehicle's dynamic model, similar to that encountered in micro aerial vehicles (see Bouabdallah et al. (2004) and Kumar and Michael (2012)), but generalized to an N-rotor system of arbitrary configuration in Section 3. In Section 4 we describe how these dynamics scale with the size and configuration of the vehicle. It will be shown how

Institute for Dynamic Systems and Control, ETH Zurich, Zurich, Switzerland.

Corresponding author:
Raymond Oung, Institute for Dynamic Systems and Control, ETH Zurich, 8092 Zurich, Switzerland.
Email: raymond.oung@alumni.ethz.ch

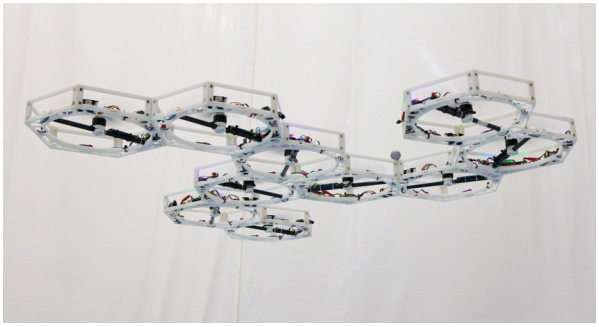


Fig. 1. The DFA is a modular reconfigurable multi-rotor VTOL vehicle composed of hexagonal-shaped *modules* that can be assembled and flown in an unlimited number of arbitrary configurations.

the configuration of a vehicle can be parameterized by only a few variables.

The second part of this work begins with Section 5, which describes the onboard estimation strategy used by the DFA. In Section 6, we characterize the exogenous disturbances that are expected to excite our system during flight. This will become important later in optimizing the parameters of our control strategy, which is described in Section 7, in order to achieve best performance.

The challenge in designing control algorithms for a system such as ours is its size. Such systems can consist of hundreds or thousands of individual elements, each of which is capable of moving and interacting dynamically, leading to very rich and often unexpected behavior (e.g. Cantoni et al. (2007); D'Andrea and Dullerud (2003); Fowler and D'Andrea (2003); Givone and Roesser (1972); Horowitz and Varaiya (2000); Jiang et al. (2006); Lynch and Law (2002); Stewart et al. (2003); Swaroop and Hedrick (1996)). This makes such systems difficult to predict and/or control using traditional methods. To address the scalability issue, we employ distributed and decentralized control methods. Research in modular robotics that has taken a step in this direction includes Varshavskaya et al. (2004) and Fitch and Butler (2008). Our control method, previously developed in Oung et al. (2009), will be summarized and built upon in Section 7. A similar control strategy is described in Mellinger et al. (2013), where the authors use multiple quadrotors to cooperatively manipulate a rigidly attached payload. We will then demonstrate in Section 8 a generalized procedure that can be used to automatically compute the parameters of our control strategy that achieves best performance subject to physical constraints.

The final part of this work starts in Section 9 by analyzing the flight performance of our vehicle for various configurations and sizes. Our analysis leads to a method, described in Section 10, that can be used to efficiently store, on board, the near-optimal control tuning parameters. We validate our strategy in Section 11 and demonstrate experimental results for up to 12 modules flying together in a variety of configurations. Concluding remarks are made in Section 12.

2. System description

This section describes the DFA as it pertains to flight. The system, however, is also capable of autonomously driving on the ground and coordinating its actions with its connected peers over an inter-module communication network (Oung and D'Andrea, 2011).

The DFA is composed of identical, hexagonal-shaped *modules* (see Figure 2). Much like a biological cell, a module is the smallest independent autonomously working unit of the system that is able to coordinate with its neighbors to perform a joint task. Each module has a characteristic length (side to side) of 250 mm and has a mass of approximately 255 g. Much of the mass resides on the outer perimeter of a module, giving it a relatively high-mass moment of inertia around its vertical axis in comparison to the horizontal axes. Its center of mass is located away from the geometrical center, in the direction of the battery. This 250 mAh, three-cell lithium-ion polymer battery (Thunder Power RC G6 Pro Lite) is capable of delivering up to 70 W of continuous power and can power the module for approximately 3 min of flight, which is sufficient for performing experiments and live demonstrations. For details on the physical parameters of a single module, see Appendix B.

In the following, we divide the description of a module into six main categories: (1) chassis, (2) actuation, (3) sensing, (4) communication, (5) computational unit, and (6) firmware architecture.

2.1. Chassis

The chassis of each module is composed of three major components (see Figure 2): (1) a base-frame, which holds the electronics, motors, battery, and so on; (2) a top-frame used for structural rigidity and protection of the internal components from impact; and (3) standoffs located on each of the six vertices of the chassis, which connect the top frame with the base frame. Each of these components, which are held together by nylon screws, is constructed from Polyamide-12 (PA-12), an engineering plastic with very good mechanical properties. Such properties enable our modules to withstand impacts from a fall of up to 2 m on hard surfaces, for example. Selective laser sintering is used as the fabrication process, which enables one to design with very little fabrication constraints. Lying across the center of the chassis is a stiff carbon fiber plain weave tube, which adds rigidity to the structure. The chassis, however, is not perfectly rigid. This is non-ideal because it violates some of our modeling assumptions, described later on.

A module has six genderless electro-mechanical interfaces located along each of its six sides. This enables it to dock with up to six modules at a time. Four square shaped

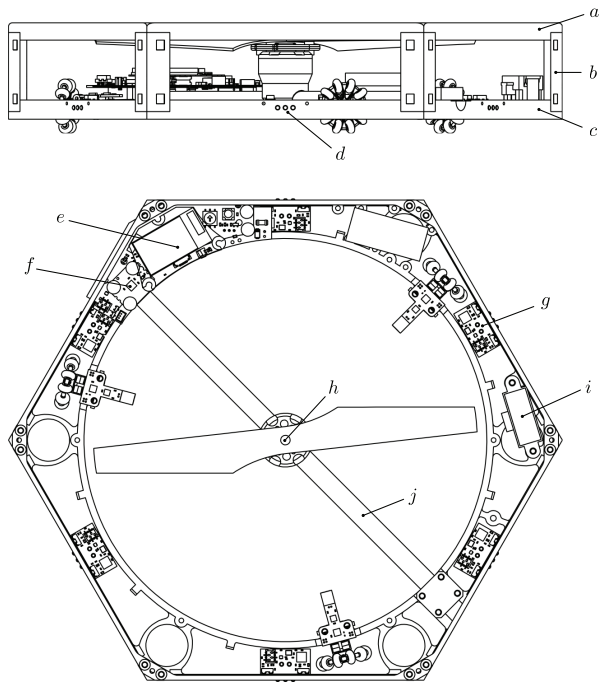


Fig. 2. The primary components of a DFA module needed for flight include: (*a*) top-frame, (*b*) standoffs, (*c*) base-frame, (*d*) communication pins, (*e*) wireless communication module and microcontroller electronics, (*f*) inertial measurement unit, (*g*) inter-module communication electronics, (*h*) brushless DC motor and 2-blade fixed-pitch propeller, (*i*) infrared distance measurement sensor, and (*j*) carbon fiber tube.

Neodymium Iron Boron magnets are used along each interface to keep adjacent modules connected. Magnets have a connection strength of 0.2 N each and are embedded in the standoffs. The connection strength along each interface has been designed to prevent a single module from disconnecting in flight due to its rotor's reaction torque (which is generated by the induced aerodynamic drag and change in angular momentum of the rotor during flight), but weak enough such that the force of gravity is sufficient to disconnect it from the vehicle when rotors are turned off. This weakness in inter-module connection is of little concern during flight as the control strategy to be described later on minimizes the inter-module shear forces and moments.

2.2. Actuation

Positioned at the geometric center of the module and sitting on top of the carbon fiber tube is a commercial off-the-shelf 14-pole outrunner brushless DC motor (Hacker X-BL 52S), which is controlled by a brushless DC motor speed controller (X-3D X-BLDC) that is capable of a 100 Hz update rate over an inter-intergrated circuit (I^2C) bus interface. The motor is directly connected to a two-blade fixed-pitch propeller (Parrot AR Drone Propellers C/A Type), which in combination can produce at least 3N of thrust at the maximum rotational velocity of the motor. Due to the inability to vary the pitch of a propeller, a combination of clockwise (CW) and counterclockwise (CCW) rotating propellers are needed in a vehicle in order to cancel out the rotor's reaction torques during flight. This is the only feature of our system that defeats its homogeneity.

2.3. Sensing

Each module has a three-axis digital-output MEMS rate-gyroscope and accelerometer (Invensense MPU-6050) rigidly mounted to its chassis on top of the carbon fiber tube. This sensor communicates its information to the onboard microcontroller over an I^2C bus, and is capable of sampling angular rates and translational accelerations at 1kHz with a resolution of 7.6×10^{-3}deg/s and 0.6×10^{-3}m/s^2 in the body-fixed frame of a module.

A module is also equipped with a general purpose infrared (IR) distance measurement sensor (Sharp GP2Y0A02YK), which is pointed downwards and perpendicular to the plane of the vehicle. This commonly used light-weight, low-cost sensor produces an output voltage that is a function of the distance being measured in the specified range of the device. This output voltage is updated by the device at approximately 24 Hz, which is an order of magnitude faster than the expected closed-loop dynamics of the vehicle. A 12-bit analog to digital converter (ADC), which is available on the onboard microcontroller, is used to measure the output voltage of this sensor resulting in a resolution of approximately 2.4 mm at 1 m distance. These sensors operate based on the principle of triangulation, which is almost immune to interference from ambient light and is relatively indifferent to varying surface reflectances of the object being detected (Siegwart and Nourbakhsh, 2004). Depending on the arrangement of sensors and the measuring distance, sensors tend to become cross-coupled when their beams overlap. For the work described here, this effect is mitigated by choosing configurations and flying height where sensors do not affect each other. In the future, this can be solved by replacing the sensor with one that has a tighter beam cone and/or modulating the sensors on and off so as to not affect neighboring sensors.

2.4. Communication

Along each interface is a set of three spring-loaded pins, which are used to make electrical contact with the adjacent pins of a neighboring module. The three electrical contacts (i.e. transmit, receive, and electrical ground) allow for full-duplex communication at 115.2 kbps between modules over a standard universal asynchronous receiver/transmitter (UART) interface. Custom designed electronic transceivers, each with their own UART interface (SC16IS75x) and 64-byte first-in, first-out (FIFO) transmit and receive buffer are connected together over an inter-intergrated circuits (I^2C)

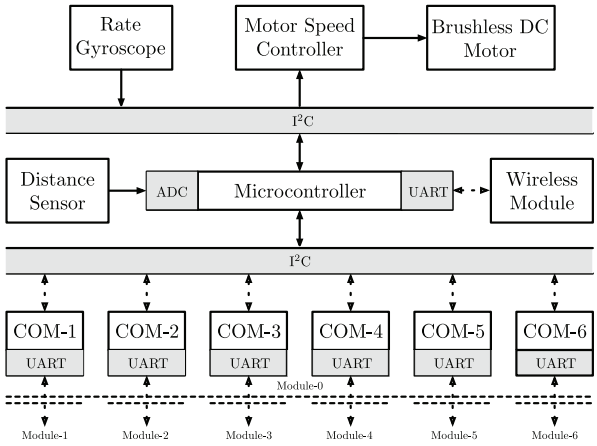


Fig. 3. The primary electrical components of a DFA module needed for flight consists of a 32-bit microcontroller connected to a three-axis rate-gyroscope, motor speed controller, and six inter-module communication interfaces over two independent I^2C buses. An IR distance measurement sensor is connected to the microcontroller's 12-bit ADC. A wireless module (interfaced through the microcontroller's UART peripheral) can be used to transmit telemetry data and/or receive commands to/from a base station.

bus, which interfaces directly with the onboard microcontroller (see Figure 3); the transceivers act as *slaves* and the microcontroller acts as a *master* on the communication bus.

We use one of two interchangeable wireless systems operating on a 2.4 GHz carrier band frequency to communicate with the modules. On the module-side, both systems use a universal asynchronous receiver/transmitter (UART) to interface with the microcontroller at a rate of 115.2 kbps. What separates the two systems is their reliability and directionality of communication. Depending on the environment and its intended use, one system is preferred over another.

2.4.1. Variable-latency, bidirectional communication A variable-latency, bidirectional multicast system (Roving Networks RN-131) can be used to communicate data (e.g. telemetry) and user-commands between modules and the base station. This system uses orthogonal frequency-division multiplexing and complies with IEEE 802.11g specifications. The advantage here is the ability to use an existing wireless infrastructure, provided that there is one. We have observed that this system does not perform reliably in Wi-Fi saturated environments, which adds additional problems if the data that is being sent over this system is needed for real-time control.

2.4.2. Low-latency, unidirectional communication A low-latency, unidirectional broadcast system (Laird Technologies LT2510[1]) is used for transmitting user-commands (e.g. take-off, land, waypoints, hold, and so on) to the modules from a base station computer. This system uses the method of frequency-hopping spread spectrum for wireless transmission (Proakis, 2000), which we have observed as operating reliably in Wi-Fi saturated environments.

All communication (i.e. inter-module and wireless) is performed in an *unreliable, but frequent* manner using the user datagram protocol (UDP) along with an addition-based modular checksum for error detection. This has the advantage of low complexity in the communication's transport layer, and for lossy communication channels the most current data always has priority. This is important in a real-time embedded system with limited memory and processing power.

2.5. Computational unit

Handling all of the computation is a single onboard 72MHz ARM-based 32-bit microcontroller (ST Microelectronics ARM M3-Cortex STM32F103x) with firmware that was developed in-house. This microcontroller resides on a custom designed printed circuit board that interfaces with all of the previously described peripherals and more (see Figure 3).

2.6. Firmware architecture

The firmware primarily employs a time-triggered architecture (Kopetz and Bauer, 2003). For modularity, the system is divided into three major concurrent *tasks*: (1) estimation, (2) control, and (3) communication. These tasks are representative of their time-scales, and each is independently handled by its own timer-based interrupt and service routine. Within each task are processes that are coordinated via a cooperative-scheduler (Pont, 2001); this minimizes the overhead needed to guarantee mutual exclusion of processes and results in very deterministic outcomes.

In a time-triggered architecture information is *pulled* at predefined periodic instances; data remains in memory until updated, and updates are performed in-place. This paradigm relegates temporal control to the highest-level functions and supports independence between processes, i.e. sending and receiving functions can operate at different rates and there are no means by which a receiver can influence a sender (Kopetz, 2011). To manage concurrency, in particular race conditions, we employ something similar to the non-blocking write protocol described in Kopetz and Reisinger (1993) for shared data. This requires an additional control variable (with atomic hardware access) for each piece of shared data. This method is more deterministic and achieves better performance compared to real-time systems that lock data, for example mutexes and semaphores, which places a waiting process in a queue (or stack) for an indeterminate period of time.

The firmware is divided into three major layers; starting from the bottom and working our way up these are: (1) hardware abstraction, (2) generic middleware, and (3) application layer. The hardware abstraction layer is specific to the hardware and it accounts for low-level communication

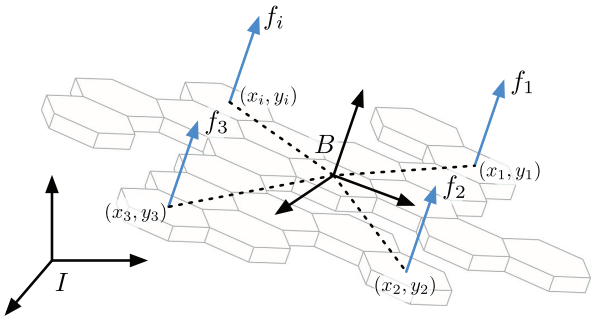


Fig. 4. The DFA's body-fixed coordinate frame B is located at its center of mass and is aligned with the principal axes of rotation. The orientation of the vehicle with respect to an inertial frame I can be described by a set of ZYX-Euler angles. Control of the vehicle is achieved by varying the thrust f_i produced by module i located at (x_i, y_i).

between peripheral devices. The generic middleware tracks and coordinates data traveling to and/or from the various peripherals; it generally handles all of the timing requirements. The application layer contains high-level algorithms and various libraries needed, for example, in state estimation and control of the vehicle. This layer contains code that can be wrapped in a MEX file and simulated in MATLAB.

3. Dynamics model

The DFA is modeled as a rigid body that is capable of producing multiple independently controlled positive thrust vectors directed upwards and perpendicular to the plane of the vehicle. These thrust vectors of magnitude f_i are positioned at (x_i, y_i) with respect to a body-fixed coordinate frame B, where the subscript $i \in \{1 \ldots \mathrm{N}\}$ is used to uniquely identify a module (see Figure 4). It is assumed here that each module is identical in mass and size and that each is capable of producing enough thrust to lift its own weight mg, where m is the mass of a single module and g is the acceleration due to gravity. The rigid body assumption greatly simplifies our system by discarding all inter-module dynamic behavior, which in reality exists due to non-ideal elastic behavior of the chassis, but is assumed to be of little consequence here.

The body frame B is positioned at the vehicle's center of mass and is aligned in a right-hand coordinate system with the principal axes of rotation, where the z-axis points upward. The orientation of B with respect to an inertial frame I is described by ZYX-Euler angles, which is commonly used in the aeronautics field to represent yaw α, pitch β, and roll γ acting first around the z-axis, followed by the y-axis, and lastly around the x-axis, respectively (Craig, 2004). The orientation of B with respect to I can therefore be expressed by the rotation matrix

$$
\begin{aligned}
{}^{I}_{B}\mathbf{R}_{\mathrm{zyx}} &= \mathbf{R}_{\mathrm{z}}(\alpha)\,\mathbf{R}_{\mathrm{y}}(\beta)\,\mathbf{R}_{\mathrm{x}}(\gamma), \qquad (1)\\
&= \begin{bmatrix} \mathrm{c}_\alpha \mathrm{c}_\beta & \mathrm{c}_\alpha \mathrm{s}_\beta \mathrm{s}_\gamma - \mathrm{s}_\alpha \mathrm{c}_\gamma & \mathrm{c}_\alpha \mathrm{s}_\beta \mathrm{c}_\gamma + \mathrm{s}_\alpha \mathrm{s}_\gamma \\ \mathrm{s}_\alpha \mathrm{c}_\beta & \mathrm{s}_\alpha \mathrm{s}_\beta \mathrm{s}_\gamma + \mathrm{c}_\alpha \mathrm{c}_\gamma & \mathrm{s}_\alpha \mathrm{s}_\beta \mathrm{c}_\gamma - \mathrm{c}_\alpha \mathrm{s}_\gamma \\ -\mathrm{s}_\beta & \mathrm{c}_\beta \mathrm{s}_\gamma & \mathrm{c}_\beta \mathrm{c}_\gamma \end{bmatrix},
\end{aligned}
$$

where c and s are shorthand forms of cosine and sine, respectively, and

$$
\mathbf{R}_{\mathrm{z}}(\alpha) := \begin{bmatrix} \mathrm{c}_\alpha & -\mathrm{s}_\alpha & 0 \\ \mathrm{s}_\alpha & \mathrm{c}_\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad (2)
$$

$$
\mathbf{R}_{\mathrm{y}}(\beta) := \begin{bmatrix} \mathrm{c}_\beta & 0 & \mathrm{s}_\beta \\ 0 & 1 & 0 \\ -\mathrm{s}_\beta & 0 & \mathrm{c}_\beta \end{bmatrix}, \qquad (3)
$$

$$
\mathbf{R}_{\mathrm{x}}(\gamma) := \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{c}_\gamma & -\mathrm{s}_\gamma \\ 0 & \mathrm{s}_\gamma & \mathrm{c}_\gamma \end{bmatrix}. \qquad (4)
$$

3.1. Force and torque generation

A module can use its rotor to generate thrust and simultaneously a reaction torque. The latter is generated mainly by the induced aerodynamic drag and, to some extent, the change in angular momentum of the rotor during flight. We assume that the thrust of each rotor is oriented in the z-axis of the vehicle, although we note that this assumption does not hold once the rotor begins to rotate and translate through the air, an effect that is known as *rotor flapping* (Leishman, 2006).

Each rotor is directly connected to a brushless DC motor, which is controlled by a motor speed controller. The motor's speed controller effectively enables us to control the rotor's angular velocity ω_i. Experimental results have shown the steady-state thrust f_i, which is generated by a rotor i that is not translating horizontally or vertically in free air, to behave as the lumped parameter model

$$
f_i = \mathrm{k}_f \omega_i^2, \qquad (5)
$$

where k_f is a constant that is a function of the rotor's disk area, radius, geometry, and profile, as well as air density. This parameter can easily be determined from static thrust experiments, which has the advantage that it will also naturally incorporate the effect of aerodynamic drag induced by the flow of air around the chassis.

In a similar experiment, the reaction torque τ_i acting on the chassis that is generated by a rotor i was shown to exhibit the following behavior

$$
\tau_i = \mathrm{k}_\tau \omega_i^2, \qquad (6)
$$

where the coefficient k_τ is also a constant that depends on the same parameters as k_f and can be determined by static thrust experiments. The torque τ_i can therefore be approximated as a linear function of thrust f_i,

$$
\tau_i = c_i f_i, \qquad (7)
$$

where c_i is a constant of magnitude $|c_i| = \mathrm{c}$ for all i and its sign depends on the handedness of the propeller.

3.2. Rigid body dynamics

From first principles, the equations of motion for a rigid body of mass Nm and mass moment of inertia $\mathbf{J}$ that is

subjected to nonconservative forces $\mathbf{f} = (0, 0, \sum_i^{\mathrm{N}} f_i)$ and torques $\mathbf{t} = (0, 0, \sum_i^{\mathrm{N}} \tau_i)$ are given by

$$\dot{\mathbf{p}} = \mathbf{v}, \tag{8}$$

$$\mathrm{Nm}\dot{\mathbf{v}} = {}^{I}_{B}\mathbf{R}_{\mathrm{zyx}}\mathbf{f}, \tag{9}$$

$${}^{I}_{B}\dot{\mathbf{R}}_{\mathrm{zyx}} = {}^{I}_{B}\mathbf{R}_{\mathrm{zyx}}\omega^{\times}, \tag{10}$$

$$\mathbf{J}\dot{\omega} = -\omega \times \mathbf{J}\omega + \mathbf{t}. \tag{11}$$

The vectors $\mathbf{p} = (x, y, z)$ and $\mathbf{v} = (\dot{x}, \dot{y}, \dot{z})$ denote, respectively, the position and velocity of the rigid body with respect to inertial frame I; the vector $\omega = (p, q, r)$ denotes the angular velocities in the body frame, and the superscript $\times$ denotes the skew symmetric matrix of that vector, i.e.

$$\omega^{\times} = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}. \tag{12}$$

Four of the six degrees of freedom (DOF) of the vehicle—altitude (distance to the ground) z, roll γ, pitch β, and yaw α—can be directly controlled by appropriately varying the thrust f_i of each rotor i around the equilibrium. The altitude (or vertical) dynamics are a function of the total control effort produced by N rotors, $\sum_{i=1}^{\mathrm{N}} f_i$. The rolling and pitching torques of the vehicle are achieved using differential thrust; more precisely, they are a function of the control efforts acting along the y and x moment arms, $\sum_{i=1}^{\mathrm{N}} y_i f_i$ and $-\sum_{i=1}^{\mathrm{N}} x_i f_i$, respectively. The yawing torque around the vertical axis is the sum of all reaction torques, $\sum_{i=1}^{\mathrm{N}} c_i f_i$. The equations of motion can be summarized as the following:

$$\mathrm{Nm}\ddot{x} = (\mathrm{c}_{\alpha}\mathrm{s}_{\beta}\mathrm{c}_{\gamma} + \mathrm{s}_{\alpha}\mathrm{s}_{\gamma}) \sum_{i=1}^{\mathrm{N}} f_i, \tag{13}$$

$$\mathrm{Nm}\ddot{y} = (\mathrm{s}_{\alpha}\mathrm{s}_{\beta}\mathrm{c}_{\gamma} - \mathrm{c}_{\alpha}\mathrm{s}_{\gamma}) \sum_{i=1}^{\mathrm{N}} f_i, \tag{14}$$

$$\mathrm{Nm}\ddot{z} = \mathrm{c}_{\beta}\mathrm{c}_{\gamma} \sum_{i=1}^{\mathrm{N}} f_i - \mathrm{Nmg}, \tag{15}$$

$$I_{\mathrm{x}}\dot{p} = qr(I_{\mathrm{y}} - I_{\mathrm{z}}) + \sum_{i=1}^{\mathrm{N}} y_i f_i, \tag{16}$$

$$I_{\mathrm{y}}\dot{q} = pr(I_{\mathrm{z}} - I_{\mathrm{x}}) - \sum_{i=1}^{\mathrm{N}} x_i f_i, \tag{17}$$

$$I_{\mathrm{z}}\dot{r} = pq(I_{\mathrm{x}} - I_{\mathrm{y}}) + \sum_{i=1}^{\mathrm{N}} c_i f_i, \tag{18}$$

where $(I_{\mathrm{x}}, I_{\mathrm{y}}, I_{\mathrm{z}})$ each represents the principal mass moment of inertia around the axis denoted by its subscript. Appendix C shows how this physical parameter is computed.

Hover is considered to be the nominal state of operation and the point of equilibrium for the vehicle; it is defined here as the process of maintaining a constant position (x, y, z) in the air and a constant rotation around the vertical axis (or yaw angle α) with respect to the inertial frame I.

At equilibrium, the total thrust of the vehicle perfectly counteracts the acceleration due to gravity g. In the case where there are an equal number of CW and CCW rotors, the thrust required of each rotor is simply mg. In general, however, this may not be the case and one should use some secondary criteria, such as minimizing the maximum thrust produced. In this case, the nominal thrust of each rotor can be solved via least squares, see Appendix D. To simplify the analysis, the vehicle will be assumed to have an equal number of CW and CCW rotors. As such, each rotor i around the nominal operating point produces a control effort $u_i = f_i - \mathrm{mg}$, in units of force. Assuming small angles, the linearized dynamics can be reduced to the following:

$$\mathrm{Nm}\ddot{x} = \beta\mathrm{Nmg}, \tag{19}$$

$$\mathrm{Nm}\ddot{y} = -\gamma\mathrm{Nmg}, \tag{20}$$

$$\mathrm{Nm}\ddot{z} = \sum_{i=1}^{\mathrm{N}} u_i, \tag{21}$$

$$I_{\mathrm{x}}\ddot{\gamma} = \sum_{i=1}^{\mathrm{N}} y_i u_i, \tag{22}$$

$$I_{\mathrm{y}}\ddot{\beta} = -\sum_{i=1}^{\mathrm{N}} x_i u_i, \tag{23}$$

$$I_{\mathrm{z}}\ddot{\alpha} = \sum_{i=1}^{\mathrm{N}} c_i u_i, \tag{24}$$

where the angular accelerations $\dot{\omega} = (\dot{p}, \dot{q}, \dot{r})$ have been converted to second derivative Euler angles $\ddot{\psi} = (\ddot{\gamma}, \ddot{\beta}, \ddot{\alpha})$ through the following kinematic relations (Craig, 2004) such that all equations of motion are represented with respect to the inertial frame:

$$\dot{\psi} = \mathbf{T}(\psi)\,\omega, \tag{25}$$

$$\ddot{\psi} = \mathbf{T}(\psi)\,(\dot{\omega} - \dot{\mathbf{T}}^{-1}(\psi)\,\dot{\psi}), \tag{26}$$

where $\mathbf{T}(\psi)$ is a nonlinear transformation of the form

$$\mathbf{T}(\psi) := \begin{bmatrix} 1 & \sin\gamma\tan\beta & \cos\gamma\tan\beta \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma\sec\beta & \cos\gamma\sec\beta \end{bmatrix}. \tag{27}$$

Since the DFA operates around the equilibrium, the transformation (27) can be reduced to the following linear transform $L\{\cdot\}$ assuming small angle approximations:

$$L\{\mathbf{T}(\psi)\} := \begin{bmatrix} 1 & 0 & \beta \\ 0 & 1 & -\gamma \\ 0 & \gamma & 1 \end{bmatrix}. \tag{28}$$

In the rest of this work we will express the equations of motion (19)–(24) more compactly as

$$\ddot{x} = \beta g, \tag{29}$$

$$\ddot{y} = -\gamma g, \tag{30}$$

$$\mathbf{M}\begin{bmatrix} \ddot{z} \\ \ddot{\gamma} \\ \ddot{\beta} \\ \ddot{\alpha} \end{bmatrix} = \mathbf{P}^{\mathrm{T}}\mathbf{u}, \tag{31}$$

where

$$\mathbf{M} = \mathrm{diag}(\mathrm{Nm}, I_{\mathrm{x}}, I_{\mathrm{y}}, I_{\mathrm{z}}), \tag{32}$$

$$\mathbf{u} = (u_1, \ldots, u_{\mathrm{N}}), \tag{33}$$

and the matrix $\mathbf{P} \in \mathbb{R}^{\mathrm{N}\times 4}$ contains information pertaining to the configuration of the vehicle,

$$\mathbf{P} = \begin{bmatrix} 1 & y_1 & -x_1 & c_1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & y_{\mathrm{N}} & -x_{\mathrm{N}} & c_{\mathrm{N}} \end{bmatrix}. \tag{34}$$

4. Configuration parameterization and scaling

Let ℓ denote the characteristic length of a module and assume that the vehicle's configuration is relatively circular, then $\frac{\ell\sqrt{\mathrm{N}}}{2}$ is comparable to the radius of the vehicle. Assuming a disk-like configuration, the vehicle's principal mass moment of inertia around the radial axis approximates that of a full disk,

$$I_d = \frac{\mathrm{Nm}}{4}\left(\frac{\ell\sqrt{\mathrm{N}}}{2}\right)^2. \tag{35}$$

In reality, however, the vehicle may take on much sparser and eccentric configurations. To capture this, two scaling parameter, ϵ_{x} and ϵ_{y}, are employed to capture the mass distribution of the vehicle along the x- and y-axis, respectively. The principal mass moment of inertia of the vehicle can thus be summarized as the following:

$$I_{\mathrm{x}} = \epsilon_{\mathrm{x}} I_d, \tag{36}$$

$$I_{\mathrm{y}} = \epsilon_{\mathrm{y}} I_d, \tag{37}$$

and for planar objects such as the DFA, the mass moment of inertia around the z-axis follows the Perpendicular Axis Theorem (Riley et al., 2006)

$$I_{\mathrm{z}} = I_{\mathrm{x}} + I_{\mathrm{y}} = (\epsilon_{\mathrm{x}} + \epsilon_{\mathrm{y}})\, I_d. \tag{38}$$

The mass distribution parameters $(\epsilon_{\mathrm{x}}, \epsilon_{\mathrm{y}})$ can be used to parameterize the configuration space of the vehicle. They are expected to be close to 1 for a disk-like configuration, and they are expected to get larger or smaller depending on the eccentricity of the vehicle's configuration. The upper and lower bounds of these parameters are characterized by a straight line configuration along the x- and y-axis.

By substituting (36)–(38) into (21)–(24), then normalizing the various physical parameters and control efforts as the following:

$$\hat{x}_i = \frac{x_i}{\frac{\ell\sqrt{\mathrm{N}}}{2}}, \quad \hat{y}_i = \frac{y_i}{\frac{\ell\sqrt{\mathrm{N}}}{2}}, \quad \hat{c}_i = \frac{c_i}{\ell}, \quad \hat{u}_i = \frac{u_i}{\mathrm{m}}, \tag{39}$$

we can gain some intuition about how the flight dynamics, and maneuverability, scale with the size of the vehicle N, and how they are affected by the vehicle's configuration. Here the normalized position coordinates $\hat{x}_i$ and $\hat{y}_i$ are at most on the order of 1 for a disk-like configuration, $\hat{c}_i$ is the normalized force to torque conversion constant and is expected to be much less than 1, and $\hat{u}_i$ is the normalized control effort in units of acceleration.

Lateral dynamics are unaffected by size and/or configuration, as can be seen from (29) and (30). However, the remaining DOF are affected, which can be seen in the following normalized and linearized equations of motion:

$$\ddot{z} = \frac{1}{\mathrm{N}}\sum_{i=1}^{\mathrm{N}} \hat{u}_i, \tag{40}$$

$$\hat{I}_{\mathrm{x}}\ddot{\gamma} = \frac{1}{\mathrm{N}}\sum_{i=1}^{\mathrm{N}} \hat{y}_i\hat{u}_i, \tag{41}$$

$$\hat{I}_{\mathrm{y}}\ddot{\beta} = -\frac{1}{\mathrm{N}}\sum_{i=1}^{\mathrm{N}} \hat{x}_i\hat{u}_i, \tag{42}$$

$$\hat{I}_{\mathrm{z}}\ddot{\alpha} = \frac{1}{\mathrm{N}}\sum_{i=1}^{\mathrm{N}} \hat{c}_i\hat{u}_i, \tag{43}$$

where

$$\hat{I}_{\mathrm{x}} = \frac{\epsilon_{\mathrm{x}}\ell\sqrt{\mathrm{N}}}{8}, \quad \hat{I}_{\mathrm{y}} = \frac{\epsilon_{\mathrm{y}}\ell\sqrt{\mathrm{N}}}{8}, \quad \hat{I}_{\mathrm{z}} = \frac{(\epsilon_{\mathrm{x}} + \epsilon_{\mathrm{y}})\,\ell\mathrm{N}}{8}. \tag{44}$$

This shows that the maximum vertical acceleration is independent of N. The maximum angular accelerations in roll and pitch, however, decrease by a factor of $\sqrt{\mathrm{N}}$, while the maximum angular acceleration in yaw decreases by a factor of N. Angular accelerations are also affected by the vehicle's configuration; angular accelerations in roll and pitch decrease by a factor of ϵ_{x} and ϵ_{y}, respectively, while yaw decreases by a combination of both mass distribution parameters.

5. State estimation

To control the DFA we require an estimate of the vehicle's attitude, angular velocities, position, and translational velocities. Each module is capable of estimating a subset of these states using its onboard sensors. The remaining states are measured using offboard sensors. Figure 5 shows a block diagram of the various sensors that are used to obtain a full-state estimate of our vehicle.

5.1. Onboard sensors

Each module is equipped with a three-axis rate-gyroscope and an IR distance measurement sensor, see Section 2 for details. The rate-gyroscope measures body angular velocities of the vehicle (p, q, r), which can be converted to

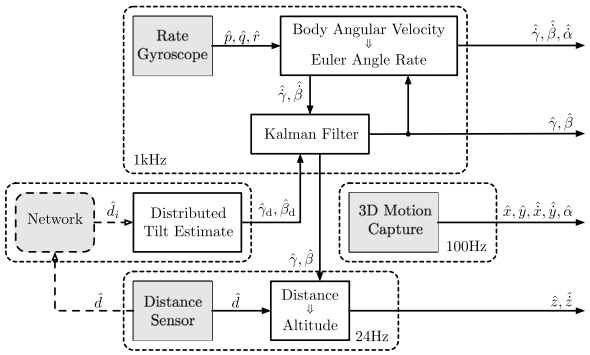


Fig. 5. The onboard sensors, consisting of a three-axis rate-gyroscope and IR distance measurement sensor, can be used to estimate Euler angle rates $(\hat{\dot{\gamma}}, \hat{\dot{\beta}}, \hat{\dot{\alpha}})$, tilt $(\hat{\gamma}, \hat{\beta})$, altitude $\hat{z}$, and vertical velocity $\hat{\dot{z}}$. Distance measurements can be shared with other modules over the inter-module communication network, which can be used to estimate tilt $(\hat{\gamma}_d, \hat{\beta}_d)$ (Kriegleder et al., 2012). An offboard 3D motion capture system is used to measure XY-position/-velocity $(\hat{x}, \hat{y}, \hat{\dot{x}}, \hat{\dot{y}})$ and yaw-angle $\hat{\alpha}$.

Euler angle rates $(\dot{\gamma}, \dot{\beta}, \dot{\alpha})$ through the transformation in (25). Euler angle rates $(\dot{\gamma}, \dot{\beta}, \dot{\alpha})$ are therefore a function of the Euler angles (γ, β) and the body angular velocities (p, q, r). Euler angles are approximated using the most recent estimate from the previous time step. This approximation works well in practice because the update rate of our Euler angle estimate is relatively fast (i.e. 1 kHz).

Static calibration parameters of the rate-gyroscope such as scale factor, misalignment, and non-orthogonality (Titterton et al., 2004) are assumed to have been calibrated out in an earlier sensor calibration step and that the sensor's sensitivity to linear acceleration is negligible. Upon start-up, while the vehicle remains on the ground and the motors are turned off, each module estimates its sensor bias by performing a running average of its measurements over a few seconds.

By integrating the roll and pitch Euler angle rates, we can obtain a local tilt estimate that has good short-term characteristics. However, a bias in the rate measurement (due to, e.g., temperature effects and quantization error) will result in unbounded drift in the tilt estimate, which will be particularly noticeable in the translation dynamics of the vehicle, see (29) and (30). It is possible to correct for this error using the existing onboard three-axis accelerometer (Martin and Salaun, 2010). Another solution, which exploits the distributed nature of the system, is a method that was developed in Kriegleder et al. (2012, 2013) to use the IR distance measurement sensors to estimate the vehicle's tilt (i.e. roll γ and pitch β) by exchanging information between neighboring modules over the vehicle's inter-module communication network. This method assumes that the vehicle flies over a flat horizontal surface. Assuming that the sensors are calibrated, this estimate is non-biased, drift-free, and relatively precise; however, the update rate is relatively slow and depends on the algebraic connectivity of the network (Olfati-Saber and Murray, 2004). A Kalman filter can be used to optimally combine both the tilt estimate with the local rate-gyroscope measurements.

In addition to estimating tilt, the IR distance measurement sensors can be used to estimate altitude z, again assuming that the surface over which it flies is flat and horizontal. These sensors point downwards, measuring the distance from the module to a point on the ground. Altitude is a function of this distance d, the sensor's position (r_x, r_y) with respect to the vehicle's body frame B, and the tilt of the vehicle. More precisely, altitude z of the vehicle is the scalar projection of a vector ${}_B\mathbf{k} = (r_x, r_y, -d)$ in the vehicle's body frame, pointing from the center of rotation (or origin of the body frame B) of the vehicle to the point of measurement, onto the unit normal ${}_B\hat{\mathbf{n}}$ of the vehicle's plane (also known as the Hesse Normal form),

$$-{}_B\hat{\mathbf{n}} \cdot {}_B\mathbf{k} = z. \tag{45}$$

The unit normal ${}_B\hat{\mathbf{n}}$ is obtained by rotating the z-axis unit vector in the inertial frame ${}_I\hat{\mathbf{z}} = (0, 0, 1)$ to the body-frame B via the rotation matrix (1),

$${}_B\hat{\mathbf{n}} = {}^I_B\mathbf{R}_{zyx} \cdot {}_I\hat{\mathbf{z}}, \tag{46}$$

$$= \begin{bmatrix} \sin\beta \\ -\cos\beta\sin\gamma \\ -\cos\beta\cos\gamma \end{bmatrix}. \tag{47}$$

Making the appropriate substitutions into (45) and solving for altitude z yields

$$z = d\cos\beta\cos\gamma + r_x\sin\beta - r_y\cos\beta\sin\gamma. \tag{48}$$

It is clear from (48) that the altitude estimate has a non-linear dependency on the roll γ and pitch β angles of the vehicle. At equilibrium, however, this can be linearized to the following:

$$L\{z\} = d + r_x\beta - r_y\gamma. \tag{49}$$

An estimate of the vehicle's altitude $\hat{z}$ can be computed using (49), using the latest IR distance sensor measurement $\hat{d}$, the known position of the sensors (r_x, r_y), and the latest tilt estimate $(\hat{\gamma}, \hat{\beta})$. The discrete-time derivative of the altitude estimate can be used to estimate the vertical velocity of the vehicle $\hat{\dot{z}}$.

5.2. External sensors

With its existing set of sensors, the DFA is incapable of accurately estimating its lateral position and velocity in flight, nor its yaw-angle. To measure these states, and to simultaneously evaluate our control and estimation algorithms, the DFA is flown in the workspace of a 3D motion capture system (Vicon MX), which is capable of measuring the 3D pose (i.e. position and orientation) of an object with respect to an intertial coordinate frame in real-time.

The system functions by illuminating this workspace with IR light[2] and visually tracks uniquely placed IR-reflective markers on the object; the baseline of these markers determine the precision of the system's pose estimate. From these markers the pose of the object, in this case the DFA, is computed by dedicated hardware at a rate of 200 Hz, and this information is sent directly over Ethernet to a computer (base station), which then forwards this information to each module over a wireless interface at a rate of 100 Hz. In future revisions of the vehicle, however, one can eliminate the need for a 3D motion capture system with additional onboard sensors, such as a global positioning system (Kim et al., 2006), machine-vision (Bristeau et al., 2011; Meier et al., 2011), amongst others.

Wireless transmission of information is inherently prone to transmission losses. To account for this, we use a model-based predictor to estimate the states at any given time and we use measurements only when they become available. More precisely, let $\hat{\mathbf{x}}[n]$ represent the state estimate of a vehicle at time step n consisting of the position, velocity, Euler angles, and Euler angle rates; let the elements of the vector $\mathbf{u}[n]$ consist of the command efforts u_i from all modules. A prediction of the state at time step $n+1$ using an estimate of the state up to time step n is given by

$$\hat{\mathbf{x}}[n+1|n] = \mathbf{f}(\,\hat{\mathbf{x}}[n|n], \mathbf{u}[n]\,), \tag{50}$$

where $\mathbf{f}$ is a function describing the linear dynamics (29)–(31). The state is updated whenever a measurement $\tilde{\mathbf{x}}$ becomes available,

$$\hat{\mathbf{x}}[n+1|n+1] = \tilde{\mathbf{x}}[n+1|n+1]. \tag{51}$$

However, in the event of temporary measurement failure, such as a transmission loss in a wireless link, the estimator simply pushes forward the previous estimate,

$$\hat{\mathbf{x}}[n+1|n+1] = \hat{\mathbf{x}}[n+1|n]. \tag{52}$$

The use of a predictor enables the system to operate in situations where measurements arrive at unpredictable instances in time and thus increases the robustness of the system.

6. Expected exogenous disturbances

The expected exogenous disturbances can be divided into two categories: (1) process noise and (2) measurement noise. We experimentally characterized the power spectral density (PSD) of each disturbance for our system and demonstrate that they can all be modeled as zero-mean white noise.

The PSD $S_{\nu\nu}$ for a discrete ergodic signal $\nu[m]$ of sample size K, which is assumed to be periodic (i.e. $\nu[m + l\mathrm{K}] = \nu[m]\ \forall l \in \mathbb{Z}$), is the discrete Fourier transform of the signal's autocorrelation function (Stoica and Moses, 1997):

$$S_{\nu\nu}(\,\Omega_k) = \sum_{n=0}^{\mathrm{K}-1} R_{\nu\nu}[n] e^{-j\Omega_k n} \text{ for } k = \{0, 1, \ldots, \mathrm{K}-1\}, \tag{53}$$

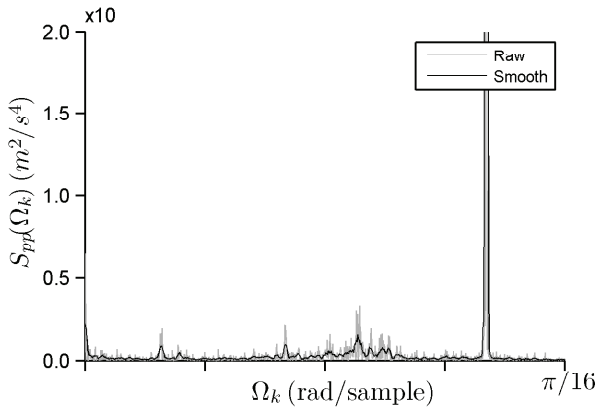


Fig. 6. The PSD of the rotor-generated disturbance at equilibrium thrust shows that the main portion of power is contained at 0.16 rad/sample (or 1144 rad/s), which is twice the angular velocity of the module's two-blade rotor. Measurements in this experiment were sampled at 7KHz.

where $\Omega_k = 2\pi k/\mathrm{K}$ is the normalized frequency and $R_{\nu\nu}[n]$ is the autocorrelation function of an ergodic signal $\nu[m]$ computed with wrap-around, which has units of the signal squared, and is defined as

$$R_{\nu\nu}[n] = \frac{1}{\mathrm{K}} \sum_{m=0}^{\mathrm{K}-1} \nu[m]\nu[m-n]. \tag{54}$$

6.1. Process noise

The primary source of process noise comes from the turbulent flow of air that is generated by the rotor and aerodynamic properties of the chassis. We model this type of disturbance as being spatially *uncorrelated* across the vehicle. In an outdoor environment, one could also take into account spatially *correlated* disturbances, such as a wind gusts using a Dryden wind turbulence model (Hoblit, 1988). To characterize the spatially uncorrelated disturbance, a single module was rigidly attached to a six-axis force-torque sensor (ATI Industrial Automation Mini40) with a sampling frequency of 7 KHz to measure the normalized thrust (in units of acceleration) produced at equilibrium; thrust is normalized by the mass m of a single module.

As expected, the PSD (see Figure 6) of this disturbance is relatively constant across a large frequency band, except for a significant amount of power contained at 0.16 radians per sample (or 1144 rad/s), which is approximately twice the angular velocity of our two-bladed rotor (5460 RPM). Since the PSD is not completely constant, it may not be appropriate to assume that the process noise due to the rotor is white. However, most of this power lies in a frequency band substantially higher than the expected closed-loop dynamics of the system. It is therefore reasonable to assume a zero-mean white noise spectrum with a standard deviation of $\sigma_p = 1.16\ \mathrm{m/s^2}$ in the frequency range of interest.

This disturbance, which is independently generated by each rotor, manifests itself as process noise on the state of the vehicle. This can be seen by considering (31); the variances in process noise for those states is given by

$$\begin{bmatrix} \sigma_{\ddot{z}}^2 \\ \sigma_{\ddot{\gamma}}^2 \\ \sigma_{\ddot{\beta}}^2 \\ \sigma_{\ddot{\alpha}}^2 \end{bmatrix} = \mathbf{M}^{-1}\mathbf{P}^{\mathrm{T}}\mathrm{Var}[\mathbf{u}]\mathbf{P}(\mathbf{M}^{-1})^{\mathrm{T}}, \tag{55}$$

where $\mathrm{Var}[\mathbf{u}] = \sigma_p^2$. The standard deviations resulting from this expression are given by the following:

$$\sigma_{\ddot{z}} = \frac{\sigma_p}{\mathrm{m}\sqrt{\mathrm{N}}}, \tag{56}$$

$$\sigma_{\ddot{\gamma}} = \frac{\sigma_p}{\mathrm{m}\sqrt{\sum_{i=1}^{\mathrm{N}} y_i^2}}, \tag{57}$$

$$\sigma_{\ddot{\beta}} = \frac{\sigma_p}{\mathrm{m}\sqrt{\sum_{i=1}^{\mathrm{N}} x_i^2}}, \tag{58}$$

$$\sigma_{\ddot{\alpha}} = \frac{\mathrm{N}|c_i|\sigma_p}{\mathrm{m}\sqrt{\sum_{i=1}^{\mathrm{N}} (x_i^2 + y_i^2)}}, \tag{59}$$

As expected, process noise decreases as the size of the vehicle gets larger.

It is worth mentioning that one type of disturbance that is neglected in this model due to modeling difficulties is the transverse airflow caused by neighboring rotors (Leishman, 2010). These effects may not be neglected, especially when flying near the ground. However, our model and experiments occur well above ground level, so we assume these effects to be negligible.

6.2. Measurement noise

The PSD for each sensor (i.e. rate-gyroscope, IR distance measurement sensor, and 3D motion capture system) was computed using data obtained from experiments.

6.2.1. Rate-gyroscope The PSD of the measurements obtained from the rate-gyroscope is shown in Figure 7. Measurements were taken at 700 Hz while sitting at rest. The PSD shows a reasonably flat spectrum, with a very gradual decrease above 0.18 radians per sample (or 125.6 rad/s). This can be attributed to the sensor's built-in low-pass filter, which has a cut-off frequency that was manually set to 20 Hz (or 125.6 rad/s)—an order of magnitude larger than the expected closed-loop dynamics of the system.

The variance of the Euler angle rate estimate is approximated by sending $\mathrm{Var}[\hat{\omega}] = (\sigma_p^2, \sigma_q^2, \sigma_r^2)$ through the transformation in (25) around the equilibrium, which simply results in $\mathrm{Var}[\hat{\dot{\psi}}] = \mathrm{Var}[\hat{\omega}]$. The Euler angle rate estimates are modeled as having a zero-mean white noise spectrum with a standard deviation of $\sigma_{\dot{\gamma}} = \sigma_{\dot{\beta}} = 5.4 \times 10^{-4}$ rad/s around the x- and y-axis, respectively, and $\sigma_{\dot{\alpha}} = 5.7 \times 10^{-4}$ rad/s around the z-axis.

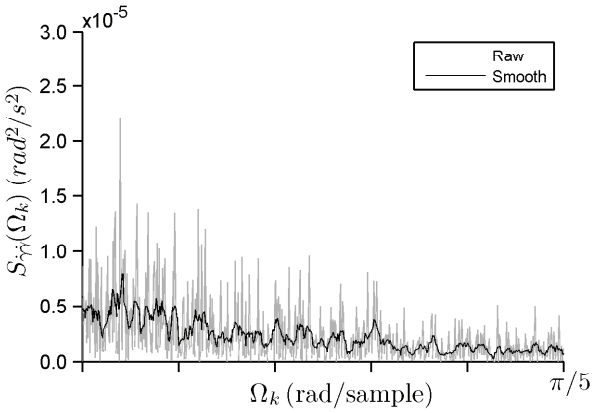


Fig. 7. The PSD of the measurements obtained from the three-axis rate-gyroscope is relatively flat, with a very gradual decrease above 0.18 rad/sample, which is the cut-off frequency of the sensor's internal low-pass filter. Measurements were taken at 700Hz. The response shown here is for angular rate measurements in roll $\dot{\gamma}$; angular rate measurements taken for the other two DOF show a similar response.

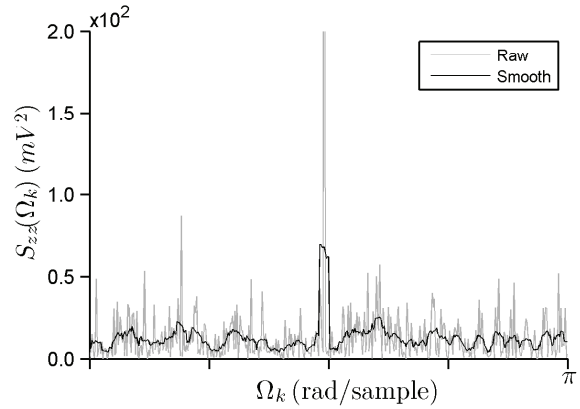


Fig. 8. The PSD of the measurements obtained from the IR distance measurement sensor is relatively flat, except for the peak at $\Omega_k = \pi/2$, which corresponds to the refresh rate of the sensor (24Hz). Measurements were taken at 100 Hz.

6.2.2. IR distance measurement sensor The PSD of the measurements obtained from the IR distance measurement sensor is shown in Figure 8. Static measurements were taken at 1 m distance (i.e. the hovering height of the vehicle) with a sampling frequency of 100 Hz. The PSD shows a relatively flat response, except for the concentration of power contained at approximately $\Omega_k = \pi/2$ radians per sample, which corresponds to the refresh rate of the sensor (24 Hz) and is again larger than the expected closed-loop dynamics of the system. The altitude estimate is therefore modeled as having a zero-mean white noise spectrum with a standard deviation of $\sigma_z = 5.8. \times 10^{-3}$m.

As mentioned in Section 5, in addition to estimating altitude, IR distance measurements are used to estimate tilt (γ, β). Due to its linear relationship (49), the PSD of

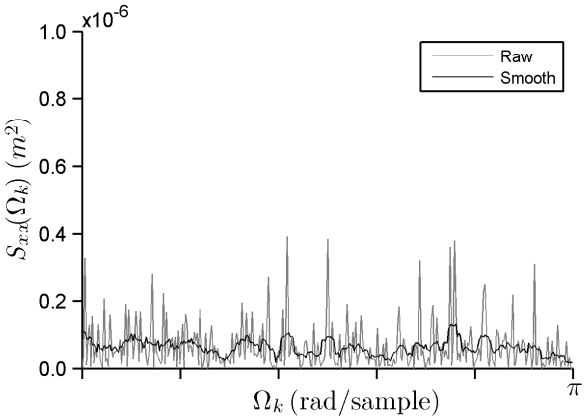


Fig. 9. The PSD of the measurements taken by the 3D motion capture system is relatively flat over the entire spectrum. The response shown here is for the position x; the responses for the other DOF are similiar. Measurements were taken at 200 Hz.

Table 1. Standard deviations of all state estimates.

	Standard deviation
σ_x	3.0×10^{-4} m
σ_y	3.0×10^{-4} m
σ_z	5.8×10^{-3} m
$\sigma_{\dot{x}}$	3.0×10^{-2} m/s
$\sigma_{\dot{y}}$	3.0×10^{-2} m/s
$\sigma_{\dot{z}}$	5.8×10^{-3} m/s
$\sigma_\gamma, \sigma_\beta$	see Kriegleder et al. (2012)
σ_α	1.6×10^{-3} rad
$\sigma_{\dot{\gamma}}$	5.4×10^{-4} rad/s
$\sigma_{\dot{\beta}}$	5.4×10^{-4} rad/s
$\sigma_{\dot{\alpha}}$	5.7×10^{-4} rad/s
$\sigma_{\ddot{x}}, \sigma_{\ddot{y}}$	0 m/s^2
$\sigma_{\ddot{z}}, \sigma_{\ddot{\gamma}}, \sigma_{\ddot{\beta}}, \sigma_{\ddot{\alpha}}$	see (56)–(59)

these estimates also show a flat response and can therefore be modeled as a zero-mean white noise spectrum. The standard deviations of these measurements ($\sigma_\gamma, \sigma_\beta, \sigma_z$), however, will depend on the physical configuration of the vehicle (Kriegleder et al., 2012).

6.2.3. 3D motion capture system The PSD of the measurements obtained from the 3D motion capture system, with the tracked object at rest, is shown in Figure 9. Measurements were taken at 200 Hz. The PSD is shown to have a relatively flat response across the entire spectrum, for all DOF. Position, velocity, and yaw angle measurements of the vehicle are therefore modeled as having a zero-mean white noise spectrum with a standard deviation of $\sigma_x = \sigma_y = 3.0 \times 10^{-4}$ m, $\sigma_{\dot{x}} = \sigma_{\dot{y}} = 3.0 \times 10^{-2}$ m/s, and $\sigma_\alpha = 1.6 \times 10^{-3}$ rad, respectively.

To summarize, all exogenous disturbances expected to act on our system can in practice be modeled as zero-mean white noise. The standard deviations for the estimate or measurement of each state is listed in Table 1.

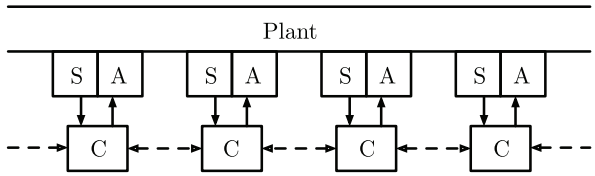


Fig. 10. The system can be modeled as an ad hoc mesh of interconnected controllers; shown here is a one-dimensional representation. Each control unit (or module) C uses its onboard sensors S to independently control its actuator A in order to maintain overall stability of the vehicle. Only adjacent modules may communicate with one another.

7. Control strategy

The system can be thought of as a rigid body of varying shape and size with multiple inter-communicating control units, each of which is capable of independently controlling its own actuator (see Figure 10). A relatively straightforward, parameterized controller is presented here to control the system around the equilibrium (or hover). Although this approach may seem to limit the vehicle's maneuverability, VTOL vehicles with linear controllers designed around equilibrium have been demonstrated to be quite agile and capable of performing aggressive maneuvers (Hehn and D'Andrea, 2011; Huang et al., 2009; Mellinger et al., 2012; Purwin and D'Andrea, 2009). Nevertheless, nonlinear controllers that are exponentially stable almost everywhere in SO(3) exist (Lee et al., 2010) and can in principal be applied here as well.

With full-state feedback it is possible to design, for example, a linear quadratic regulator (Anderson and Moore, 1990) for a given configuration. This method, however, has some drawbacks: (1) the controller itself is essentially a black box to the designer and provides little intuition on the closed-loop dynamics of the system; and (2) the size and computational complexity of the controller increases with the size of the vehicle and is therefore not scalable. Due to these limitations, we have devised a scalable control strategy based on loop-shaping (Skogestad and Postlethwaite, 2007) for a modular VTOL vehicle using a finite number of tuning parameters. This was previously presented in Oung and D'Andrea (2011), the results of which are summarized below.

We employ a strongly cascaded control architecture (see Figure 11). The lowest level control loop is the motor speed controller. Next is the altitude and attitude controller, which are the four DOF that can be directly controlled. The highest level control loop is the XY-position controller.

7.1. Actuator dynamics

The transfer function that takes the desired control effort $u_{d,i}$ to the actual control effort u_i, around the nominal operating point, can be modeled as a linear time-invariant (LTI) system. We use a non-parametric frequency response identification method to characterize this system (Rake, 1980).

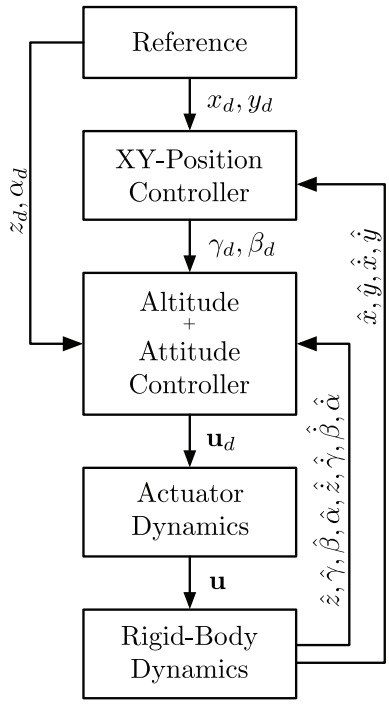


Fig. 11. A strongly cascaded control architecture is employed, consisting of three layers: (1) low-level actuator dynamics; (2) mid-level altitude and attitude control; and (3) high-level XY-position control.

A sinusoidally varying desired control effort, converted to angular velocity via (5), was sent to the motor speed controller with an offset equivalent to the equilibrium thrust and an amplitude representative of the operating region. The thrust resulting from the commanded input was measured using a six-axis force-torque strain gauge sensor with a resolution of 0.01 N at a sampling frequency an order of magnitude higher than the maximum input frequency. This procedure was repeated for inputs with frequencies ranging from 0.1 to 100 rad/s. As shown in Figure 12, the thrust response can be approximated as a first-order LTI system up to 45 rad/s,

$$\frac{u_i(s)}{u_{d,i}(s)} = \frac{\omega_r}{s + \omega_r}, \tag{60}$$

where the rotor's bandwidth ω_r is approximately 30 rad/s.

7.2. Altitude and attitude controller

Consider a control strategy of the form:

$$\mathbf{u}_d = \mathbf{Q} \begin{bmatrix} f_z \\ f_\gamma \\ f_\beta \\ f_\alpha \end{bmatrix}, \tag{61}$$

where the elements of the vector $\mathbf{u}_d \in \mathbb{R}^{N\times 1}$ are the desired control efforts imparted by each module. The matrix

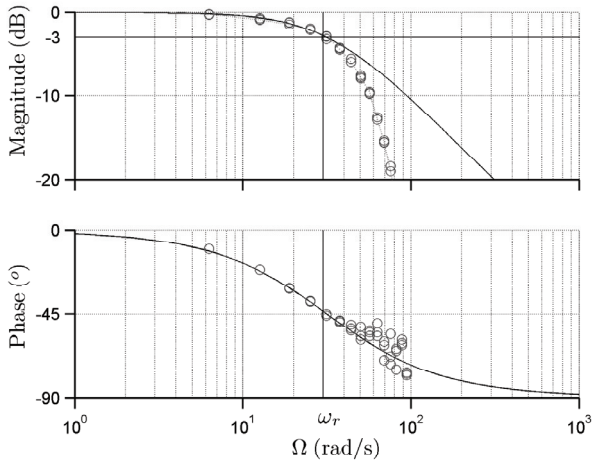


Fig. 12. The frequency response of the motor for commanded input control effort $u_{d,i}$ to actual output control effort u_i (see (60)) approximates a first-order transfer function up to approximately 45 rad/s. The system's bandwidth (or –3dB frequency) is approximately 30 rad/s. At higher frequencies, the measured response diverges from the approximation due to higher order effects such as time delay.

$\mathbf{Q} \in \mathbb{R}^{N\times 4}$ allocates control effort, a method widely adopted in aerospace literature for handling actuator redundancy (see Härkegård and Glad (2005) and references therein), and is a function of the vehicle's configuration. In this case, $\mathbf{Q}$ can be designed such that $\mathbf{P}^T\mathbf{Q} = \mathbf{I}_{4\times 4}$; recall that $\mathbf{P}$ captures the vehicle's configuration (34). In doing so, the DOF of the closed-loop system become decoupled. The functions $(f_z, f_\gamma, f_\beta, f_\alpha)$ essentially describe the closed-loop dynamics of the system. Note that this is only possible if $\mathbf{P}$ has full column rank, which physically means that the configuration is flight-feasible; this can be used as an algorithmic check to determine whether or not a configuration can be flown.

Assuming that an estimator is used to obtain full-state estimate of the system, one can force the closed-loop response of each DOF to behave as a second-order mass-spring-damper.

$$f_z = -\mathrm{Nm}(2\omega_z\zeta_z\dot{\hat{z}} + \omega_z^2(\hat{z} - z_d)), \tag{62}$$

$$f_\gamma = -I_x(2\omega_\gamma\zeta_\gamma\dot{\hat{\gamma}} + \omega_\gamma^2(\hat{\gamma} - \gamma_d)), \tag{63}$$

$$f_\beta = -I_y(2\omega_\beta\zeta_\beta\dot{\hat{\beta}} + \omega_\beta^2(\hat{\beta} - \beta_d)), \tag{64}$$

$$f_\alpha = -I_z(2\omega_\alpha\zeta_\alpha\dot{\hat{\alpha}} + \omega_\alpha^2(\hat{\alpha} - \alpha_d)). \tag{65}$$

The hat notation is used to denote the estimate of that state and the subscript d denotes the desired state. This strategy contains two design parameters per DOF: (1) a closed-loop natural frequency $\omega_s > 0$, and (2) a damping ratio $\zeta_s > 0$, where the subscript $s \in \{z, \gamma, \beta, \alpha\}$ denotes the DOF. One of the features of this control strategy is that it minimizes the shear forces and moments between modules (Oung et al., 2009).

7.3. XY-position controller

As previously mentioned in Section 3, the lateral DOF (x, y) are indirectly controlled through the roll γ and pitch β angles of the vehicle, see (29) and (30). Since these two DOF are also decoupled, they can be designed to exhibit a second-order response. Let the functions f_x and f_y represent the closed-loop accelerations in x and y, respectively,

$$f_x = -2\omega_x\zeta_x\hat{\dot{x}} - \omega_x^2(\hat{x} - x_d), \tag{66}$$

$$f_y = -2\omega_y\zeta_y\hat{\dot{y}} - \omega_y^2(\hat{y} - y_d). \tag{67}$$

Making use of the tilt to lateral acceleration relationships in (29) and (30), this can be converted to the two desired angles in roll γ_d and in pitch β_d of (63) and (64),

$$\gamma_d = -\frac{f_y}{g} = \frac{2\omega_y\zeta_y\hat{\dot{y}} + \omega_y^2(\hat{y} - y_d)}{g}, \tag{68}$$

$$\beta_d = \frac{f_x}{g} = -\frac{2\omega_x\zeta_x\hat{\dot{x}} + \omega_x^2(\hat{x} - x_d)}{g}. \tag{69}$$

7.4. Implementation

Assuming full-state feedback of the vehicle, which we denote as $\tilde{\mathbf{y}}$, our controller can be written as

$$\mathbf{u}_d = -\mathbf{Q}\mathbf{M}\mathbf{K}_\mathrm{o}\tilde{\mathbf{y}}, \tag{70}$$

where the vector $\tilde{\mathbf{y}}$ consists of position, Euler angles, translational velocities, and Euler angle rates,

$$\tilde{\mathbf{y}} = (x, y, z, \gamma, \beta, \alpha, \dot{x}, \dot{y}, \dot{z}, \dot{\gamma}, \dot{\beta}, \dot{\alpha}). \tag{71}$$

This is scaled by the control gain matrix, which contains 12 unique physical parameters,

$$\mathbf{K}_\mathrm{o} = \begin{bmatrix} 0 & 0 & \omega_z^2 & 0 & 0 & 0 & 0 & 0 & 2\omega_z\zeta_z & 0 & 0 & 0 \\ 0 & -\omega_\gamma^2\omega_y^2 & 0 & \omega_\gamma^2 & 0 & 0 & 0 & -\omega_\gamma^2 2\omega_y\zeta_y & 0 & 2\omega_\gamma\zeta_\gamma & 0 & 0 \\ \omega_\beta^2\omega_x^2 & 0 & 0 & 0 & \omega_\beta^2 & 0 & \omega_\beta^2 2\omega_x\zeta_x & 0 & 0 & 0 & 2\omega_\beta\zeta_\beta & 0 \\ 0 & 0 & 0 & 0 & 0 & \omega_\alpha^2 & 0 & 0 & 0 & 0 & 0 & 2\omega_\alpha\zeta_\alpha \end{bmatrix}. \tag{72}$$

The result is scaled by the vehicle's total mass and principal mass moment of inertia, contained in the matrix $\mathbf{M}$, which results in the closed-loop dynamics of the vehicle. The matrix $\mathbf{Q}$ maps these dynamics to the desired control effort of each module, see (61).

In the case of the DFA, each module i has a full-state estimate of the vehicle denoted by $\mathbf{y}_i$. It is therefore possible to implement our control strategy in a *decentralized* manner, i.e. modules do not need to communicate during flight. This provides us with a lower benchmark in terms of performance for comparing distributed algorithms in future work.

Thus each module i commands its motor with a desired control effort of the form

$$u_{d,i} = -\mathbf{K}_i\mathbf{y}_i, \tag{73}$$

where

$$\mathbf{K}_i = \mathbf{q}_i\mathbf{M}\mathbf{K}_\mathrm{o}, \tag{74}$$

and $\mathbf{q}_i$ denotes row i of matrix $\mathbf{Q}$. The vector of desired control efforts is thus given by

$$\mathbf{u}_d = -\mathbf{K}\mathbf{y}, \tag{75}$$

where

$$\mathbf{K} = [\mathbf{K}_1, \mathbf{K}_2, \ldots, \mathbf{K}_\mathrm{N}]^\mathrm{T}, \tag{76}$$

$$\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_\mathrm{N}]^\mathrm{T}. \tag{77}$$

Assuming that the state estimates of each module are identical, the resulting control effort would be equivalent to (70) and there would be no shear force or moment acting along the interface of adjacent modules caused by the control effort. However, since state estimates will naturally differ between modules due to uncorrelated sensor noise, there will be an apparent shear force and moment acting along the interface of adjacent modules caused by the control effort alone.

The controller described here has been designed based on continuous-time system dynamics, with no consideration for the sampling rate of the system. Performance losses due to this approximation are considered to be negligible since the sampling rate of all the sensors is significantly faster than the closed-loop system dynamics. The controller itself operates at a rate of 100 Hz, limited only by the maximum command rate of the actuator.

The following three sections deal specifically with the control parameters, and the reader may skip to Section 11 without any loss of continuity. More concretely, we will describe in Section 8 how one can compute the optimal control tuning parameters and examine how these optimal parameters are affected by the size and configuration of the vehicle in Section 9. In Section 10, we demonstrate how these parameters can be efficiently stored on a system with limited computational resource such as the DFA.

8. Optimizing performance

The control tuning parameters in (72) can be optimized for flight performance of the system, while satisfying its physical constraints. What follows is a description of a general procedure with implementation details pertaining to the DFA.

8.1. Performance objective

The objective is to find control tuning parameters that minimize our closed-loop system's output (or error) $\mathbf{z}$ power due to an exogenous disturbance $\mathbf{w}$. This can be quantified by computing the $\mathcal{H}_2$-norm $||\cdot||_2$ of our closed-loop system,

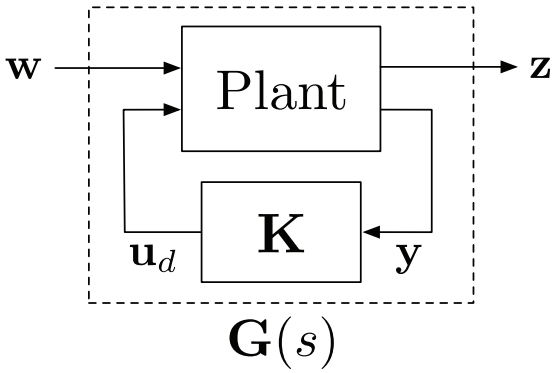


Fig. 13. General block diagram for computing the $\mathcal{H}_2$- norm of a system, where **w** denotes the exogenous inputs to the system and **z** respresents the weighted response resulting from these inputs; the signal $\mathbf{u}_d$ and **y** denote the desired control effort and measured states, respectively.

which we denote as $\mathbf{G}(s)$. In continuous-time, the $\mathcal{H}_2$ norm is defined as

$$||\mathbf{G}(s)||_2^2 = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \mathrm{tr}(\,\mathbf{G}^*(j\omega)\,\mathbf{G}(j\omega)\,)\,d\omega, \tag{78}$$

where the term $\mathrm{tr}(\,\mathbf{G}^*(j\omega)\,\mathbf{G}(j\omega)\,)$ is the sum of the squared magnitudes of all the elements of $\mathbf{G}(j\omega)$ (Anderson and Moore, 1990). The $\mathcal{H}_2$ norm can be interpreted as an average gain of the system, performed over all the elements of the matrix transfer function $\mathbf{G}(s)$ over all frequencies, or roughly speaking it is the root-mean square value of the measured response due to white noise input (Skogestad and Postlethwaite, 2007). High performance in this context means minimizing the $\mathcal{H}_2$ norm of our system,

$$\arg\min_{\mathbf{K}_o} ||\mathbf{G}(s)||_2 \quad \text{s.t. physical constraints.} \tag{79}$$

Searching for the optimal control tuning parameters $\mathbf{K}_o$ means finding the minimum of this constrained multivariable function, which can be implemented with standard software tools like MATLAB's Optimization Toolbox or CVX (Grant and Boyd, 2012).

Our linear time invariant (LTI) closed-loop feedback system $\mathbf{G}(s)$ takes the form shown in Figure 13, where **w** denotes a vector of exogenous input disturbances with a fixed or bounded power spectrum; **z** represents the weighted system output (or error), which we want to minimize; the signal $\mathbf{u}_d$ denotes the desired control effort; **y** denotes the measured states, see (77); and **K** is given by (76). The state-space description of the generalized plant is given by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}_1\mathbf{w} + \mathbf{B}_2\mathbf{u}_d, \tag{80}$$

$$\mathbf{z} = \mathbf{C}_1\mathbf{x}, \tag{81}$$

$$\mathbf{y}_i = \mathbf{C}_2\mathbf{x} + \mathbf{D}_i\mathbf{w}, \tag{82}$$

where all matrices are real matrices of compatible dimensions and the state consists of the vehicle's position, Euler angles, translational velocities, Euler angle rates, and the control effort produced by each rotor,

$$\mathbf{x} = (x, y, z, \gamma, \beta, \alpha, \dot{x}, \dot{y}, \dot{z}, \dot{\gamma}, \dot{\beta}, \dot{\alpha}, u_1, \ldots, u_{\mathrm{N}}). \tag{83}$$

The state transition matrix **A** for nominal flight dynamics of the DFA, including motor dynamics as described in Section 7, is given by

$$\mathbf{A} = \begin{bmatrix} 0_{6\times6} & \mathbf{I}_{6\times6} & 0_{6\times\mathrm{N}} \\ \mathcal{A}_{2\times6} & 0_{2\times6} & 0_{2\times\mathrm{N}} \\ 0_{4\times6} & 0_{4\times6} & \mathbf{M}^{-1}\mathbf{P}^{\mathrm{T}} \\ 0_{\mathrm{N}\times6} & 0_{\mathrm{N}\times6} & -\omega_r\mathbf{I}_{\mathrm{N}\times\mathrm{N}} \end{bmatrix}, \tag{84}$$

where

$$\mathcal{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & \mathrm{g} & 0 \\ 0 & 0 & 0 & -\mathrm{g} & 0 & 0 \end{bmatrix}. \tag{85}$$

The desired control effort $\mathbf{u}_d$ is fed into the matrix $\mathbf{B}_2$, which scales these values by the rotor's bandwidth ω_r,

$$\mathbf{B}_2 = \begin{bmatrix} 0_{12\times\mathrm{N}} \\ \omega_r\mathbf{I}_{\mathrm{N}\times\mathrm{N}} \end{bmatrix}. \tag{86}$$

The feedback vector $\mathbf{y}_i$ of module i is a function of the system's states, which does not include control effort, therefore

$$\mathbf{C}_2 = \begin{bmatrix} \mathbf{I}_{12\times12} & 0_{12\times\mathrm{N}} \end{bmatrix}. \tag{87}$$

The expected exogenous disturbances **w**, which are described in Section 6, consist of unit variance process noise $\mathbf{w}_p$ and measurement noise $\mathbf{w}_m$,

$$\mathbf{w} = (\,\mathbf{w}_p, \mathbf{w}_m). \tag{88}$$

Process noise $\mathbf{w}_p$ may consist of correlated and uncorrelated disturbances. Here we only consider the effect that rotor disturbances have on the states of our system, since correlated disturbances such as wind gusts are not a factor when flying indoors in a laboratory environment. This is given by

$$\mathbf{w}_p = (\,w_{\ddot{x}}, w_{\ddot{y}}, w_{\ddot{z}}, w_{\dot{\gamma}}, w_{\ddot{\beta}}, w_{\ddot{\alpha}}), \tag{89}$$

and is scaled by the matrix $\mathbf{B}_1$ using the appropriate standard deviations listed in Table 1,

$$\mathbf{B}_1 = \left[\begin{array}{c} 0_{6\times6} \\ \mathrm{diag}(\,\sigma_{\ddot{x}}, \sigma_{\ddot{y}}, \sigma_{\ddot{z}}, \sigma_{\dot{\gamma}}, \sigma_{\ddot{\beta}}, \sigma_{\ddot{\alpha}}) \\ 0_{\mathrm{N}\times6} \end{array}\middle|\; 0_{(12+\mathrm{N})\times12\mathrm{N}} \right]. \tag{90}$$

Measurement noise $\mathbf{w}_m$ consists of an individual sensor noise vector for each module i,

$$\mathbf{w}_m = (\,\mathbf{w}_{m,1}, \ldots, \mathbf{w}_{m,\mathrm{N}}), \tag{91}$$

where

$$\begin{aligned}\mathbf{w}_{m,i} = (&w_{x,i}, w_{y,i}, w_{z,i}, w_{\gamma,i}, w_{\beta,i}, w_{\alpha,i}, \ldots \\ &w_{\dot{x},i}, w_{\dot{y},i}, w_{\dot{z},i}, w_{\dot{\gamma},i}, w_{\dot{\beta},i}, w_{\dot{\alpha},i}),\end{aligned} \tag{92}$$

and is scaled by the matrix $\mathbf{D}_i$ using the appropriate standard deviation listed in Table 1,

$$\mathbf{D}_i = \left[\begin{array}{cccc} 0_{12\times 6} & \mathcal{D}_1 & \ldots & \mathcal{D}_{\mathrm{N}} \end{array}\right], \tag{93}$$

where,

$$\mathcal{D}_j = \begin{cases} 0_{12\times 12} & \text{if } j \neq i, \\ \mathrm{diag}(\sigma_x, \sigma_y, \sigma_z, \sigma_\gamma, \sigma_\beta, \sigma_\alpha, \ldots & \\ \quad \sigma_{\dot{x}}, \sigma_{\dot{y}}, \sigma_{\dot{z}}, \sigma_{\dot{\gamma}}, \sigma_{\dot{\beta}}, \sigma_{\dot{\alpha}}) & \text{otherwise.} \end{cases}$$

The output response **z**, which is to be minimized, is a function of the states weighted by

$$\mathbf{C}_1 = \left[\begin{array}{cccccc|c} \mathrm{w_x} & 0 & 0 & 0 & 0 & 0 & \\ 0 & \mathrm{w_y} & 0 & 0 & 0 & 0 & \\ 0 & 0 & \mathrm{w_z} & 0 & 0 & 0 & 0_{4\times(6+\mathrm{N})} \\ 0 & 0 & 0 & 0 & 0 & \mathrm{w}_\alpha & \end{array}\right]. \tag{94}$$

Since the objective of the vehicle is to hover, only position and yaw-angle is considered. It is sensible to set the position weights $(\mathrm{w_x}, \mathrm{w_y}, \mathrm{w_z})$ to the same value; this value is normalized to a constant value of 1. There exists then just one free design parameter: the weight w_α on yaw-angle in units of meters per radian. It is difficult to find a suitable expression for this weight in terms of physical parameters. Experiments have shown $\frac{1}{35}$ m/rad demonstrates reasonable performance, with the deviation in yaw-angle α reasonable in comparison to the position error.

Frequency dependent weights, for both the exogenous input and regulated output, could also be used. For example, had our exogenous disturbances not been zero-mean white noise, a filter could have been designed to exhibit the same characteristic response given a zero-mean white noise input. This would then be absorbed by the generalized plant. One could also consider penalizing high frequency actuation to reduce wear and tear of the actuator or because it may not make sense to ask for performance past the point where the model integrity begins to degrade.

8.2. Physical constraints

Constraints are added to ensure that the parameters obtained from the optimization work within the physical limits of the system. They are also used to reduce the search space of the optimization routine; for example, the parameter search space is reduced to $\omega_s > 0$ and $\zeta_s > 0$.

A physical limit that we want to avoid is saturation of the control effort. The maximum available control effort can be found through experiment; this is approximately $\mathrm{u_{sat}} = 3.5$ N of thrust for our system. We also want to ensure that the shear forces and moments resulting from differences in control effort do not disconnect modules from the vehicle. Modules are held together by a set of magnets. Experiments with two modules demonstrate that the connection strength is weakest around an axis running horizontally along the edge of an interface. Assuming one module is fixed, in order to remain connected the connection strength of the magnets f_m should counteract the moment that is created by the combination of a module's control effort f_i and gravitational acceleration. Figure 14 shows a free body diagram of a single module connected on one side to a fixed module. Although modules are not necessarily fixed, this provides a conservative estimate of the control effort needed to disconnect a module from the vehicle. For simplicity, assume that the center of mass is at the geometric center of a module. The following condition must be satisfied in order for a module to remain connected:

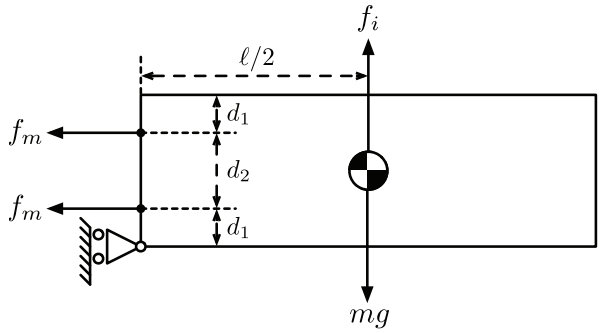


Fig. 14. Consider a conservative scenario where one module is attached to one other module, which is fixed (not shown). The weakest link is subject to a moment created around the edge-axis of a module (pointing out of the page). The connection strength of the magnets f_m should counteract the moment created by the combination of module's control effort f_i and gravitational acceleration g.

$$f_m \cdot (2\mathrm{d_1} + \mathrm{d_2}) > \left| (f_i - \mathrm{mg}) \frac{\ell}{2} \right|, \tag{95}$$

where $\mathrm{d_1}$ is the vertical distance from magnet to the top/base of the module and $\mathrm{d_2}$ is the relative distance between both magnets. This expression can be re-arranged to obtain the control effort $\mathrm{u_{mag}} = f_i - \mathrm{mg}$ effort that is necessary to decouple two connected modules.

Depending on the strength of the magnets, the magnetic constraint may be more restrictive than the maximum available control effort. Let $\mathrm{U_{max}}$ denote the maximum control effort that satisfies both constraints,

$$\mathrm{U_{max}} = \min(\mathrm{u_{sat}}, \mathrm{u_{mag}}). \tag{96}$$

In the case of the DFA, it is sufficient if the control effort lies below $\mathrm{U_{max}}$ *most of the time*. This can be formulated as N inequality constraints, where some multiple of the output control effort's standard deviation σ_{u_i} must lie below $\mathrm{U_{max}}$. While this does not guarantee that all control signals will remain below $\mathrm{U_{max}}$, it does quantify the probability of saturation.

Once again, the $\mathcal{H}_2$ system norm can be employed and this time used to obtain the standard deviation σ_{u_i} of the output control effort u_i. It can be shown that the $\mathcal{H}_2$ norm of a single-output system is equivalent to the root-mean-squared (RMS) response of that system due to white noise

input. Since this is taken at equilibrium, the RMS response is the standard deviation of the control effort.

Let us denote the system used to compute the control effort's standard deviation of module i as $\mathbf{G}_{c,i}(s)$. This system is identical to $\mathbf{G}(s)$ (see (80)–(82)), except that the weighted output response is the control effort of module i,

$$\mathbf{z}_{c,i} = \mathbf{C}_{1_{c,i}}\mathbf{x}, \tag{97}$$

where

$$\mathbf{C}_{1_{c,i}} = \left[\ 0_{1\times 12} \mid \mathcal{C}_{1\times \mathrm{N}}\ \right], \tag{98}$$

and the elements of the vector $\mathcal{C}$ are

$$c_j = \begin{cases} 1 & \text{if } j = i, \\ 0 & \text{otherwise.} \end{cases} \tag{99}$$

The standard deviation of the control effort produced by module i is therefore

$$\sigma_{u_i} = ||\mathbf{G}_{c,i}(s)\,||_2. \tag{100}$$

Our analysis until now has been performed in continuous-time; our system, however, is implemented in discrete-time. Therefore σ_{u_i} will need to be converted to its discrete-time equivalent in order to properly compare it with the constraint $\mathrm{U}_{\max}$ on the control effort. A relationship between the discrete and continous time $\mathcal{H}_2$ norms is given by Gawronski (2004):

$$||\mathbf{S}_{\mathrm{c}}||_2 = \frac{1}{\sqrt{\mathrm{T}_s}}||\mathbf{S}_{\mathrm{d}}||_2, \tag{101}$$

where $\mathbf{S}_{\mathrm{c}}$ is a continous-time system and $\mathbf{S}_{\mathrm{d}}$ is the discretization of this system using zero-order hold with a sampling frequency of T_s. Therefore, the physical constraint of the system is given by

$$||\mathbf{G}_{c,i}(s)\,||_2 < \frac{\mathrm{U}_{\max}}{\sqrt{\mathrm{T}_s}} \quad \text{for } i \in \{1,\ldots,\mathrm{N}\}. \tag{102}$$

9. Control parameter analysis

Using the optimization procedure described in the previous section, Section 8, the tuning parameters for our parameterized controller were computed using MATLAB's optimization toolbox for various configurations ranging from 4 to 20 modules in size. To provide some statistics, the procedure was implemented on 100 randomly chosen configurations for each vehicle size.

9.1. Performance versus size

We compared the performance of the DFA using our parameterized controller with optimized parameters, implemented in a decentralized architecture, to that of two other controllers in simulation:

1. **Parameterized controller in a centralized architecture**: This is the parameterized controller implemented in a centralized architecture; a controller aggregates and averages sensor measurements from all of the modules, then commands each module with the appropriate control effort. Although not scalable, this is considered the best case scenario when using our paramterized controller.
2. $\mathcal{H}_2$ **optimal**: This is the $\mathcal{H}_2$ optimal controller Doyle et al. (1989), which takes into account physical constraints, implemented for our system. It assumes full-state feedback of the vehicle and a centralized control architecture, see Appendix E for implementation details. This is considered here to be the best performing controller for our system.

The $\mathcal{H}_2$ optimal controller is expected to perform better than our parameterized control strategy because it is a dynamic controller that also grows with the size of the vehicle N. In contrast, our parameterized controller is a static gain matrix (72) containing just 12 parameters (two for each DOF) for all vehicle sizes. Traditional control techniques such as $\mathcal{H}_2$ optimal control are impractical for handling the incredible size and computational complexity of such systems. In this case, the system matrices describing the input–output state behavior of N interconnected modules will be approximately $\mathrm{N}\times\mathrm{N}$, and most matrix arithmetic operations will require $\mathcal{O}(\mathrm{N}^3)$ floating point operations (Golub and Van Loan, 1996). This makes traditional robust and optimal controller design prohibitively expensive (slow) for a large number of modules, never mind the difficulties of implementing such a controller on a resource-constrained embedded system such as the DFA. We demonstrate here that there is a trade-off between performance and scalability. In addition, thanks to its structure, our controller provides physical intuition about the closed-loop system. The same cannot be said about the $\mathcal{H}_2$ optimal controller, which is essentially a black box to the designer.

The results in Figure 15 show that flight performance improves (i.e. decreasing $||\mathbf{G}(s)\,||_2$) as the size of the vehicle increases. This is to be expected for a system with no correlated process noise such as wind gusts, and we demonstrated previously in (56)–(59) that the effect of spatially uncorrelated process noise decreases as the system grows. With decreasing exogenous disturbances acting on our system as the size of the vehicle gets larger, we would expect to see a decrease in the system's output response. Although this is a positive result, it is important to keep in mind that, in general, the maneuverability of the vehicle decreases with its size, see Section 4.

The error bars in the figure denote the variability in performance for similar sized vehicles, which suggests that performance is also a function of vehicle configuration. We showed previously in (40)–(43) that the vehicle's ability to accelerate is indeed a function of the vehicle's

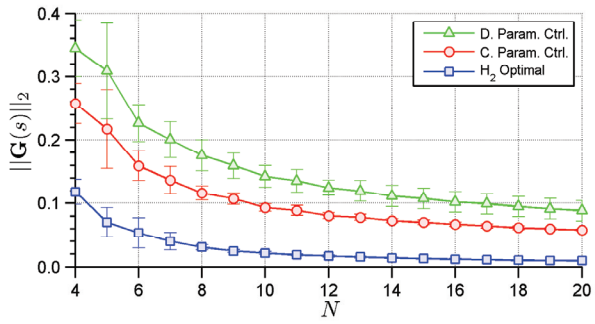


Fig. 15. The performance of the DFA improves (i.e. decreasing $||\mathbf{G}(s)||_2$) as the size of the vehicle N increases. This is to be expected because of the effective decrease in uncorrelated process noise acting on our system as the size of the vehicle gets larger. The $\mathcal{H}_2$ optimal controller is shown to perform better in comparison to the parameterized controller, for both the centralized and decentralized architecture.

configuration,

$$\ddot{z} \propto 1, \quad \ddot{\gamma} \propto \frac{1}{\epsilon_x}, \quad \ddot{\beta} \propto \frac{1}{\epsilon_y}, \quad \ddot{\alpha} \propto \frac{1}{\epsilon_x + \epsilon_y}. \tag{103}$$

The result of this and the control effort constraint imposed on our optimization procedure has the effect of varying the performance for similar sized vehicles.

In the limit, as N goes to infinity, the effect that uncorrelated process noise has on the states of our system goes to zero and the resulting performance is equivalent to a closed-loop system with measurement noise acting as the only exogenous disturbance.

9.2. Control parameters versus size and configuration

In Figure 16 we see that the optimal control tuning parameters are a function of the vehicle's size and configuration. This is to be expected since our minimization procedure is a function of the expected exogenous disturbances, which are in turn functions of vehicle's size and configuration.

As previously discussed, increasing vehicle size N has the effect of decreasing the expected exogenous disturbance, namely uncorrelated process noise, see (56)–(59). This has two immediate effects on our closed-loop system: (1) the $\mathcal{H}_2$-norm of the performance objective decreases, which is clearly illustrated in Figure 15; and (2) the expected variance in control effort decreases. These two effects coupled with the physical constraints has the effect of varying optimal control parameters with respect to vehicle size N, as well as configuration for similar sized vehicles. In general, with a reduction in disturbances, one can expect the optimization to choose a more aggressive controller.

In Figure 16(b), we see that the change in optimal control parameters with respect to the vehicle's configuration is most pronounced in the DOF of roll γ and pitch β. This is to be expected since the ability for a vehicle to roll γ and pitch β, as well as the effect in which uncorrelated process noise has on these DOF is a function of its configuration, see (103) and (57)–(58), respectively.

10. Onboard control parameter storage

Due to the limited computational resources available on board the DFA, it would be impractical to compute or store the optimal parameters for our parameterized controller using the generalized optimization procedure described in Section 8. Instead, we make use of our analysis in Section 9 and show how these parameters can be compressed and efficiently stored on our embedded system.

10.1. Mapping configuration to control parameter space

One solution is to design a function that maps the size N and configuration space (ϵ_x, ϵ_y) of the DFA to its control parameter space (ω_s, ζ_s), where $s \in \{x, y, z, \gamma, \beta, \alpha\}$. Due to symmetry of the problem, one can reduce the configuration space to just one of the two parameters. One way to obtain such a map is to surface-fit a function via least-squares to a dataset of optimal parameters—one function for each parameter (see Figure 16(b)). In general, the control parameters can be modeled as an exponential function with respect to size N and configuration ϵ_x. Some parameters (e.g. ω_x, ω_y, ζ_z, and ζ_α) can simply be assigned constant values (see Figure 16). For the range of data that has been generated (i.e. up to 100 unique configurations for each vehicle size), it is sufficient to use a first- and second-order polynomial function to model the values of the optimal parameters. For example, the parameters $p = \{\omega_z, \omega_\alpha\}$ can be modeled by a first-order polynomial of the form:

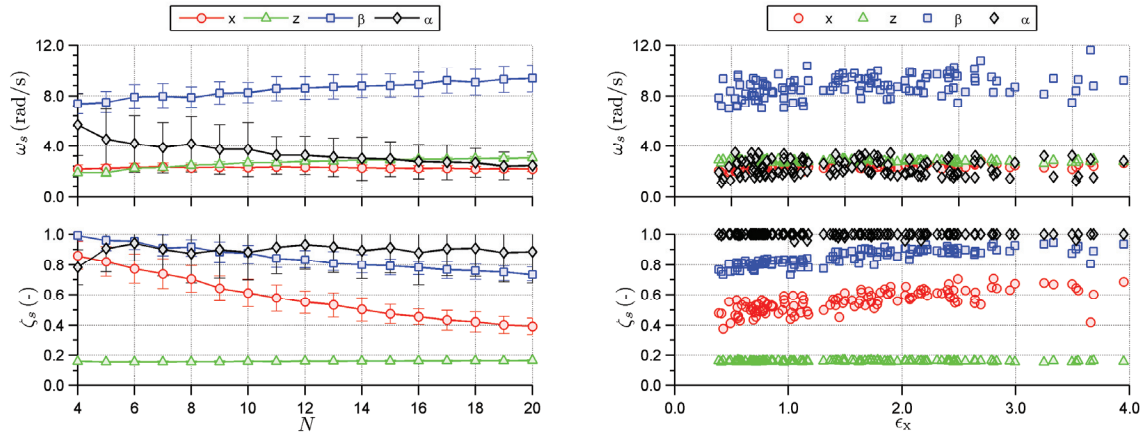
$$p = \mathrm{aN} + \mathrm{b}, \tag{104}$$

where a and b are constants. The remaining parameters, $p = \{\omega_\gamma, \omega_\beta, \zeta_x, \zeta_y, \zeta_\gamma, \zeta_\beta\}$, can be modeled as a first-order polynomial in N and second-order polynomial in ϵ_x,

$$p = \mathrm{aN} + \mathrm{b}\epsilon_x^2 + \mathrm{c}\epsilon_x + \mathrm{d}\epsilon_x \mathrm{N} + \mathrm{e}, \tag{105}$$

where a, b, c, d, e are all constants. The set of surface-fitted functions can then be tested against an independent dataset of optimal parameters. We can assess the fit by comparing the $\mathcal{H}_2$-norm of a system that uses the *fitted parameters* with one that uses the *optimal parameters*.

In summary, the only information that is needed in order to obtain reasonable control parameters for flying an arbitrary configuration of the vehicle is the size and configuration (N, ϵ_x) of the vehicle. This information can be computed before taking flight, see Appendix C. The drawback to this method, however, is that there is no guarantee that the fitted parameters will not violate the vehicle's physical constraints, see Section 8. For practical purposes, however, this may be of little concern since these constraints are conservative.



(a) As the size of the vehicle increases, uncorrelated process noise effectively decreases, and the control tuning parameters adjust to further reduce the $\mathcal{H}_2$ norm of the system. With a reduction in the expected exogenous input disturbance, the optimization may choose a more aggressive controller. The DOF in y and γ are not shown because they follow the same trend as x and β, respectively.

(b) The closed-loop optimal parameters for a set of 250 unique DFA configurations of a vehicle containing 12 modules. Most parameters maintain a relatively constant value except for the DOF in x and β. The DOF in y and γ are not shown, but follow a similar trend in ϵ_y—the results are symmetrical.

Fig. 16. The control tuning parameters resulting from our optimization procedure are a function of the vehicle's size (a) and configuration (b).

10.2. Single set of parameters

An even simpler solution is to use a single set of parameters for all sizes and configurations. This is motivated by the fact that, practically speaking, control parameters change relatively little with respect to size and configuration (see Figure 16). For the majority of configurations, the performance will be far from optimal. However, from a practical point of view the performance may be sufficient. Based on our analysis in Section 9, the optimal parameters for large vehicles, in comparison to smaller ones, are relatively aggressive. Using such parameters on smaller sized vehicles will likely violate physical constraints. This, in fact, can be verified in simulation. On the other hand, choosing optimal parameters for small-size vehicles will in general not violate physical constraints for larger vehicles (see Figure 17(b)); however the performance will be relatively sluggish.

In the case of the DFA, the performance using a single set of parameters improves with size (see Figure 17(a)), and simulation results show that the physical constraints are not violated. In this case one can choose, for example, the mean optimal parameters for a four-module vehicle to be used for all vehicle sizes and configurations.

11. Experiments

We have successfully demonstrated modular flight for up to 12 modules, both indoors and outdoors, in a variety of configurations, see *Indoor* and *Outdoor Flight Experiments* (Extensions 1 and 2), listed in Appendix A. In doing so, we validated the the linear model proposed in Section 3, the control strategy described in Section 7, and made use of the procedure for computing the optimal parameters described in Section 8.

11.1. Procedure

For the experiments performed indoors, the vehicle was operated in the workspace of a 3D motion capture system, see Section 5. This system is used for XY-position/velocity and yaw-angle feedback. It is also used as our source of ground truth for evaluating flight performance.

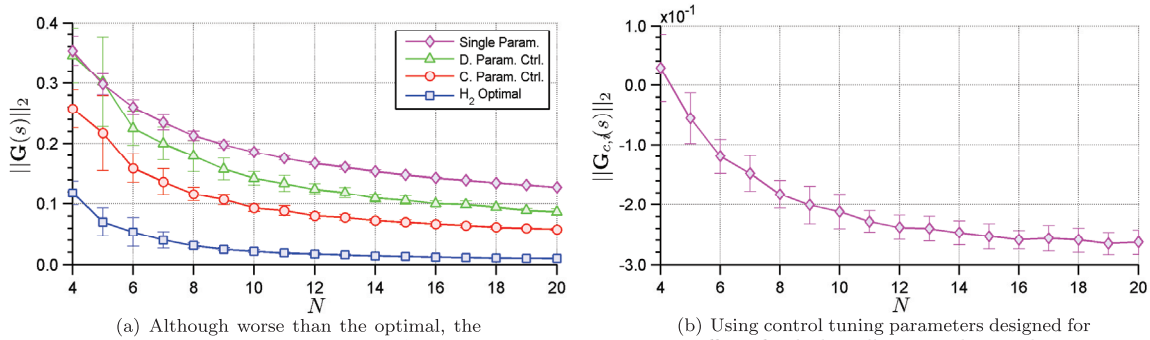
A variety of configurations were tested, some randomly assembled, others hand-picked in order to test the limits of the system (see Figure 18). After manually assembling the vehicle, commands such as *take-off* and *land* were transmitted to the vehicle over a wireless joystick interface. All control functions took place on board the vehicle. A single set of control parameters, as explained in Section 10.2, was used to fly all of the configurations tested in this experiment.

Immediately before take-off, each module executes a 3 s static calibration routine. During this period, a running average of the measurements obtained from its three-axis rate-gyroscope is taken as its bias, since the mean angular velocity is known to be zero. Upon completing its calibration routine, all modules simultaneously start its rotor and increases its angular velocity to equilibrium thrust.

Feedback from the 3D motion camera system is sent to the modules at a rate of 100 Hz. This is also the rate of the control loop. Each configuration was flown for approximately 30 s. The vehicle was then manually landed by joystick.

11.2. Nominal performance evaluation

Due to the lack of an integrator in the position controller, the vehicle does not necessarily hover at the commanded reference position. Rather, it has been observed to hover at a repeatable offset position. The cause of this offset may



(a) Although worse than the optimal, the performance using a single set of parameters still improves with increasing vehicle size.

(b) Using control tuning parameters designed for small-sized vehicles will in general not violate physical constraints (below zero) for larger vehicles.

Fig. 17. Performance (a) and constraint (b) statistics using a single set of parameters for various vehicle sizes and configurations. The single set of parameters used in this dataset are the mean optimal parameters for a four-module vehicle. The dataset used here consists of 100 randomly chosen configurations for each vehicle size.

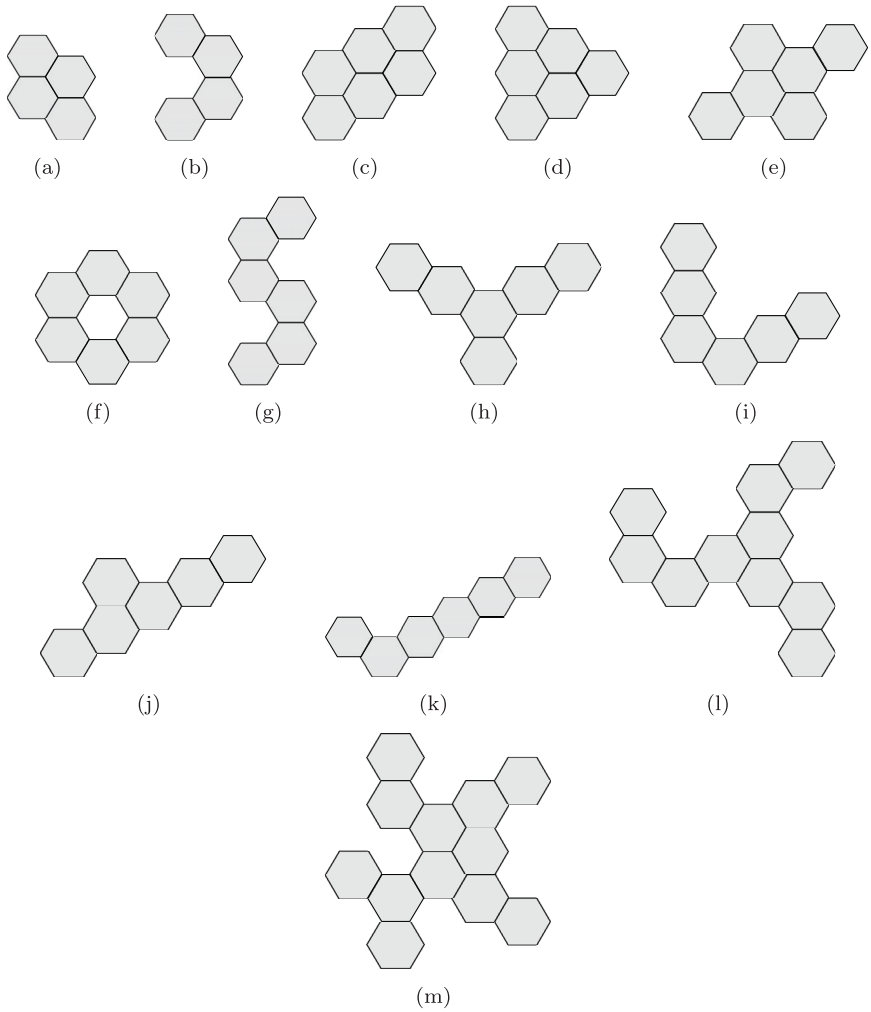


Fig. 18. A variety of configurations, ranging from 4 to 12 modules, were used to evaluate the flight performance (listed in Table 2) of the vehicle indoors.

be due to propeller quality, deviations in manufacturing and assembly of the modules, and/or 3D motion capture system frame-misalignment. Since the effect is repeatable, they can be calibrated out. However, for the purpose of these experiments, this is not important, and this calibration is left out to simplify experimental procedure.

Table 2. Standard deviation of the regulated states for various configurations in flight.

Config.	x (m)	y (m)	z (m)	α (rad)
(a)	0.025	0.022	0.028	0.012
(b)	0.052	0.042	0.036	0.035
(c)	0.011	0.011	0.010	0.002
(d)	0.024	0.026	0.018	0.004
(e)	0.054	0.044	0.027	0.023
(f)	0.020	0.029	0.018	0.004
(g)	0.014	0.013	0.013	0.023
(h)	0.019	0.035	0.014	0.012
(i)	0.011	0.027	0.014	0.006
(j)	0.021	0.029	0.030	0.031
(k)	0.030	0.033	0.031	0.027
(l)	0.031	0.026	0.017	0.027
(m)	0.015	0.031	0.017	0.004

The standard deviation σ_s of the regulated states $s \in \{x, y, z, \alpha\}$ is used as a metric for evaluating the flight performance of the vehicle,

$$\sigma_s := \sqrt{\frac{1}{\mathrm{K}} \sum_{k=1}^{\mathrm{K}} (\hat{s}[k] - \bar{s})^2}, \tag{106}$$

where K is the total number of samples k of the measured state $\hat{s}$, and $\bar{s}$ is the state average taken over all samples. Measurements were made at a rate of 200 Hz.

Each configuration was flown over an undisturbed interval of 10 seconds. The standard deviation of the regulated states for each configuration (see Figure 18) is listed in Table 2.

For some configurations, namely the ones shown in Figure 18(a) and Figure 18(c), modules tend to *jitter*. This is clearly demonstrated in Extension 1. The cause of this has mainly to do with relatively high control gains (or tuning parameter values) used in flying the vehicle. This jitter, or high velocity content, is not necessarily minimized in the optimization procedure. Moreover, the single set of parameters used for flying all of the configurations does not necessarily imply that all vehicles should behave in the same way.

11.3. Disturbance rejection

In a different experiment, the vehicle was flown in the configuration shown in Figure 18(d) and was disturbed by pushing it in the lateral direction. Figure 19 shows this 25 s dataset, with the disturbance applied at 5 s and 18 s. The results demonstrate the tracking capability of the controller.

11.4. Outdoor flight

Although performance evaluation was made indoors in a controlled environment, the system performs well outdoors even with weakly structured configurations (see Figure 20)

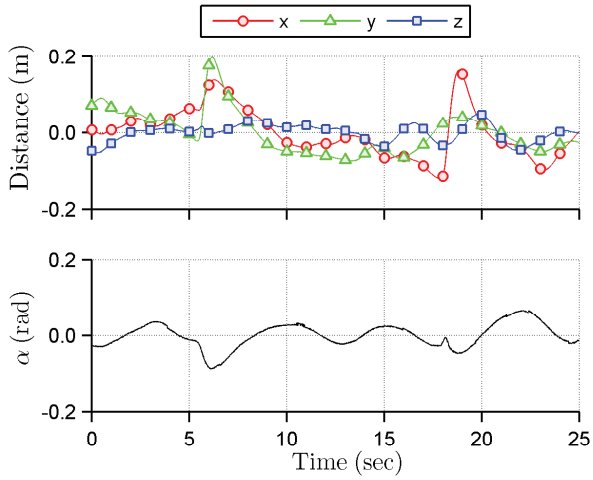


Fig. 19. The vehicle is shown to recover from a disturbance applied to it by pushing it along the y-axis and then the x-axis, shown at 5 s and 18 s, respectively.

in a light breeze (Huler, 2007) (0.3 to 1.5 m/s). In this case, joystick was used as feedback to control the lateral DOF (through roll and pitch angles) and yaw-angle, and each configuration was flown for at least 30 s.

12. Conclusions

This work documents the complete solution, spanning hardware design, modeling, estimation, control, and experimental results of a modular vertical take-off and landing vehicle, namely the DFA. With this hardware testbed, we have successfully demonstrated coordinated flight for an assortment of vehicle configurations and sizes. The results from our experiments validate the models used and proposed scalable control strategy. It has also given us insight into how the system can be improved.

One of the main contributions of this work is a parameterized method for controlling a modular vertical take-off and landing vehicle of arbitrary size and configuration, as well as an approach to automatically computing the parameters of this controller that maximizes flight performance subject to the system's physical constraints. The control strategy we describe is scalable, using a finite set of parameters for any vehicle size and configuration. Our approach to computing the control parameters is general enough such that it can easily be extended to include different disturbance models apart from those already considered and it is straightforward to consider performance metrics apart from the one we used (i.e. $\mathcal{H}_2$ system norm).

In this work we compared our control strategy to other traditional centralized control methods in simulation, and although it demonstrates slightly lower performance, it gives us much more intuition into the behavior of our system. More importantly, however, is that it lends its simplicity in enabling us to create a function that maps the configuration space of our vehicle to its control parameter space.

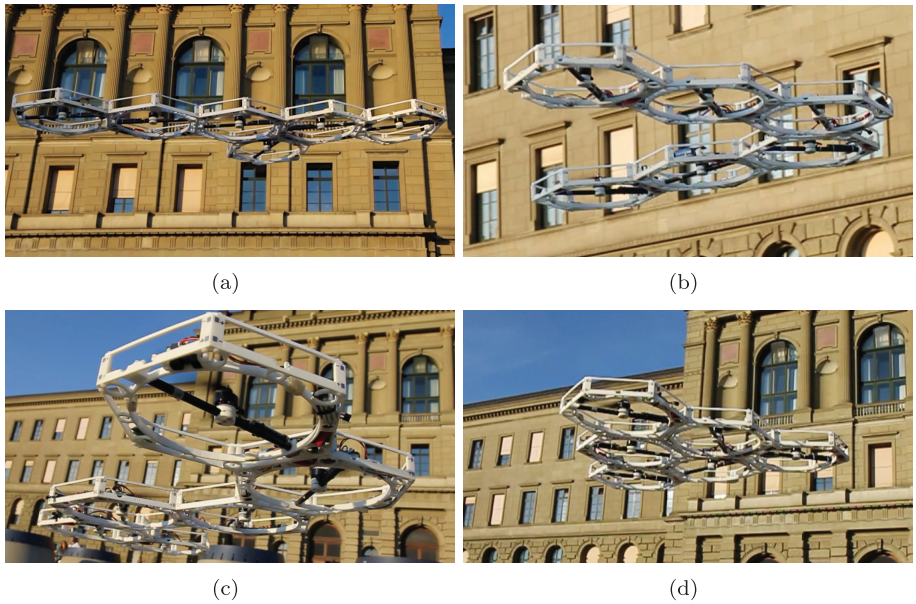


Fig. 20. A variety of six-module vehicle configurations flying outdoors using a joystick as position and yaw-angle feedback.

We show that this map can be designed as a polynomial function. Thus, computing the control tuning parameters for *any* flight-feasible configuration is now nothing more than solving a few straightforward equations.

The work described here provides a lower benchmark in terms of flight performance for this type of vehicle. Apart from sharing distance measurements between neighbors in order to estimate the vehicle's tilt, no other information is shared during flight. It will be interesting to quantify the benefits we obtain by sharing information in such a system and to evaluate the trade-offs, for example, performance versus cost of communication. We would also like to explore coupled distributed estimation and control strategies, i.e. strategies that do not a priori split the task into first estimation and then control, which is what is currently implemented in the existing system. This scheme is supported by the results in Langbort et al. (2004); they will have to be extended, however, to impose various physical constraints such as limited communication bandwidth.

Apart from improving the estimator and controller, flight performance can be improved with a stiffer chassis and a more rigid mechanical interface between modules, or otherwise a more detailed physical model. One of the major assumptions in our model is that the system is thought to behave as a single rigid body. Experiments show this to be clearly false. However, the design of the system was not without any consideration. Apart from the requirements of a stiff chassis and one that is relatively robust such that it can be used reliably as an experimental platform, a major challenge was to keep the weight of the system to a minimum in order to maximize flight time and control authority. Such constraints result in a tradeoff in design requirements. The choice of both mechanical and electrical design only came about after various design iterations and improvements.

One of the major drawbacks to our control strategy, however, is the need to know the vehicle's configuration a priori. This poses an issue for robustness in case, for example, a module becomes disconnected or that the vehicle's configuration is poorly estimated. This leads to another area of interest, which is to develop fast and scalable methods for estimating the configuration of a vehicle. Although the existing method (which floods the network with topology information) works well for relatively small networks, it does not scale well in terms of communication bandwidth. Methods to solve this problem and/or eliminate the need for knowing the physical configuration of a vehicle altogether would lead to very interesting results, such as in-flight vehicle reconfiguration.

Acknowledgments

The authors would like to acknowledge a handful of people that made this work possible. Thank you to Marc André Corzillius for assisting with the development and assembly of the electronics; Igor Thommen and Hans Ulrich Honegger for assisting with the mechanical design and assembly of the testbed; Miguel Picallo Cruz for his initial work in developing a MATLAB framework used to compute the optimal parameters for the parameterized controller; Gajamohan Mohanarajah and Maximilian Kriegleder for their suggestions and feedback; and finally to all of the anonymous reviewers.

Funding

This work was supported by the Swiss National Science Foundation (SNSF) (grant number 127020).

Notes

1. Although the LT2510 can also be used bidirectionally, implementation is not straightforward with multiple transceivers and is therefore not used.
2. The 3D motion capture system that is used in our experiments uses an IR wavelength that is different from the IR light emitted a module's IR distance measurement sensor, resulting in little or no interference.

References

Anderson BDO and Moore JB (1990) *Optimal Control: Linear Quadratic Methods*, Vol. 1. Englewood Cliffs, NJ: Prentice Hall.

Bouabdallah S, Murrieri P and Siegwart R (2004) Design and control of an indoor micro quadrotor. In: *Proceedings of the IEEE International conference on robotics and automation*, New Orleans, LA, 26 April–1 May 2004, Vol. 5, pp. 4393–4398.

Bristeau PJ, Callou F, Vissière D and Petit N (2011) The navigation and control technology inside the AR. Drone micro UAV. In: *Proceedings of the International federation of automatic control world congress*, Milano, Italy, 28 August–2 September 2011, Vol. 18, pp.1477–1484.

Cantoni M, Weyer E, Li Y, Ooi SK, Mareels I and Ryan M (2007) Control of large-scale irrigation networks. *Proceedings of the IEEE* 95(1): 75–91.

Craig JJ (2004) *Introduction to Robotics: Mechanics and Control*, 3rd edition. Englewood Cliffs, NJ: Prentice Hall.

D'Andrea R and Dullerud GE (2003) Distributed control design for spatially interconnected systems. *IEEE Transactions on Automatic Control* 48(9): 1478–1495.

Doyle JC, Glover K, Khargonekar PP and Francis BA (1989) State-space solutions to standard H2 and H∞ control problems. *IEEE Transactions on Automatic Control* 34(8): 831–847.

Fitch R and Butler Z (2008) Million module march: Scalable locomotion for large self-reconfiguring robots. *The International Journal of Robotics Research* 27(3–4): 331–343.

Fowler JM and D'Andrea R (2003) A formation flight experiment. *IEEE Control Systems* 23(5): 35–43.

Fowles G (1977) *Analytical Mechanics* (*Holt-Saunders international editions: Physics*). New York: Holt, Rinehart, and Winston.

Gawronski W (2004) *Advanced Structural Dynamics and Active Control of Structures*. New York: Springer Verlag.

Gilpin K, Kotay K, Rus D and Vasilescu I (2008) Miche: Modular shape formation by self-disassembly. *The International Journal of Robotics Research* 27(3–4): 345–372.

Gilpin K and Rus D (2010) Modular robot systems. *IEEE Robotics & Automation Magazine* 17(3): 38–55.

Givone DD and Roesser RP (1972) Multidimensional linear iterative circuits—general properties. *IEEE Transactions on Computers* 100(10): 1067–1073.

Golub GH and Van Loan CF (1996) *Matrix Computations*, Vol. 3. Baltimore, MD: Johns Hopkins University Press.

Grant M and Boyd S (2012) CVX: Matlab software for disciplined convex programming. Available at: http://cvxr.com/cvx.

Härkegård O and Glad ST (2005) Resolving actuator redundancy optimal control vs. control allocation. *Automatica* 41(1): 137–144.

Hehn M and D'Andrea R (2011) A flying inverted pendulum. In: *Proceedings of the IEEE International conference on robotics and automation*, Shanghai, 9–13 May 2011, pp.763–770.

Hoblit F (1988) *Gust Loads on Aircraft: Concepts and Applications*. Washington DC: American Institute of Aeronautics and Astronautics.

Hoffmann G, Rajnarayan DG, Waslander SL, Dostal D, Jang JS and Tomlin CJ (2004) The Stanford testbed of autonomous rotorcraft for multi agent control (STARMAC). In: *Proceedings of digital avionics systems conference*, 24–28 October 2004, Vol. 2, pp.12.E.4–121-10.

Horowitz R and Varaiya P (2000) Control design of an automated highway system. *Proceedings of the IEEE* 88(7): 913–925.

How JP, Bethke B, Frank A, Dale D and Vian J (2008) Real-time indoor autonomous vehicle test environment. *IEEE Control Systems Magazine* 28(2): 51–64. doi 10.1109/MCS.2007.914691.

Huang H, Hoffmann GM, Waslander SL and Tomlin CJ (2009) Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In: *Proceedings of the IEEE International conference on robotics and automation*, Kobe, Japan, 12–17 May 2009, pp.3277–3282.

Huler S (2007) *Defining the Wind: The Beaufort Scale and How a 19th-Century Admiral Turned Science into Poetry*. Random House Digital.

Jiang S, Voulgaris PG, Holloway LE and Thompson LA (2006) Distributed control of large segmented telescopes. In: *American Control Conference*, Minneapolis, MN, 14–16 June 2006, p.6.

Kane TR and Levinson DA (1985) *Dynamics, Theory and Applications*. New York: McGraw Hill.

Kim JH, Sukkarieh S and Wishart S (2006) Real-time navigation, guidance, and control of a UAV using low-cost sensors. In: *Field and Service Robotics*, (*Springer Tracts in Advanced Robotics*, Vol. 24). Berlin, Heidelberg: Springer, pp.299–309.

Kopetz H (2011) *Real-time Systems: Design Principles for Distributed Embedded Applications*, 2nd edition (*Real-Time Systems Series*, Vol. 25). New York: Springer.

Kopetz H and Bauer G (2003) The time-triggered architecture. *Proceedings of the IEEE* 91(1): 112–126.

Kopetz H and Reisinger J (1993) The non-blocking write protocol NBW: A solution to a real-time synchronization problem. In: *Real-time systems symposium*, Raleigh Durham, NC, 1–3 December 1993, pp.131–137.

Kriegleder M, Oung R and D'Andrea R (2012) Distributed altitude and attitude estimation from multiple distance measurements. In: *Proceedings of the IEEE/RSJ International conference of intelligent robots and systems*, Algarve, Portugal, 7–12 October 2012, pp. 3626–3632.

Kriegleder M, Oung R and D'Andrea R (2013) Asynchronous implementation of a distributed average consensus algorithm. In: *IEEE/RSJ International conference of intelligent robots and systems*. Tokyo, Japan, 3-8 November 2013.

Kumar V and Michael N (2012) Opportunities and challenges with autonomous micro aerial vehicles. *The International Journal of Robotics Research* 31(11): 1279–1291.

Langbort C, Chandra R and D'Andrea R (2004) Distributed control of heterogeneous systems interconnected over an arbitrary graph. *IEEE Transactions on Automatic Control* 9(49): 1502–1519.

Lee T, Leoky M and McClamroch NH (2010) Geometric tracking control of a quadrotor UAV on SE(3). In: *IEEE conference on decision and control*, Atlanta, GA, 15–17 December 2010, pp.5420–5425. DOI: 10.1109/CDC.2010.5717652.

Leishman JG (2006) *Principles of Helicopter Aerodynamics*, 2nd edition. Cambridge, New York: Cambridge University Press.

Leishman JG (2010) Rotorcraft aerodynamics. *Encyclopedia of Aerospace Engineering*. Chichester: Wiley-Blackwell, pp. 8–11.

Lynch JP and Law KH (2002) Decentralized control techniques for large-scale civil structural systems. In: *Proceedings of the International modal analysis conference*, Los Angeles, CA, 4–7 February 2002, pp.4–7.

Mahony R, Kumar V and Corke P (2012) Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics & Automation Magazine* 19(3): 20–32.

Martin P and Salaun E (2010) The true role of accelerometer feedback in quadrotor control. In: *Proceedings of the IEEE International conference on robotics and automation*, Anchorage, AK, 3–8 May 2010, pp.1623–1629.

Meier L, Tanskanen P, Fraundorfer F and Pollefeys M (2011) PIXHAWK: A system for autonomous flight using onboard computer vision. In: *Proceedings of the IEEE International conference on robotics and automation*, Shanghai, 9–13 May 2011, pp.2992–2997.

Mellinger D, Michael N and Kumar V (2012) Trajectory generation and control for precise aggressive maneuvers with quadrotors. *The International Journal of Robotics Research* 31(5): 664–674.

Mellinger D, Shomin M, Michael N and Kumar V (2013) Cooperative grasping and transport using multiple quadrotors. In: *Distributed Autonomous Robotic Systems*. Berlin, Heidelberg: Springer, pp. 545–558.

Merris R (1994) Laplacian matrices of graphs: A survey. *Linear Algebra and its Applications* 197: 143–176.

Michael N, Mellinger D, Lindsey Q and Kumar V (2010) The GRASP multiple micro-UAV testbed. *IEEE Robotics & Automation Magazine* 17(3): 56–65.

Murata S and Kurokawa H (2007) Self-reconfigurable robots. *IEEE Robotics & Automation Magazine* 14(1): 71–78.

Olfati-Saber R and Murray RM (2004) Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control* 49(9): 1520–1533.

Oung R and D'Andrea R (2011) The Distributed Flight Array. *Mechatronics* 21(6): 908–917.

Oung R, Ramezani A and D'Andrea R (2009) Feasibility of a Distributed Flight Array. In: *Proceedings of the IEEE conference on Decision and Control*, Shanghai, 16–18 December 2009, pp.3038–3044.

Pont M (2001) *Patterns for Time-Triggered Embedded Systems: Building Reliable Applications with the 8501 Family of Microcontrollers*. Harlow, UK: Addison Wesley Longman.

Proakis J (2000) *Digital Communications*. (*McGraw-Hill Series in Electrical and Computer Engineering*). New York: McGraw-Hill.

Purwin O and D'Andrea R (2009) Performing aggressive maneuvers using iterative learning control. In: *Proceedings of the IEEE International conference on robotics and automation*, Kobe, Japan, 12–17 May 2009, pp.1731–1736.

Rake H (1980) Step response and frequency response methods. *Automatica* 16(5): 519–526.

Riley KF, Hobson P and Bence SJ (2006) *Mathematical Methods for Physics and Engineering: A Comprehensive Guide*. Cambridge: Cambridge University Press.

Siegwart R and Nourbakhsh IR (2004) *Introduction to Autonomous Mobile Robots*. Cambridge, MA: MIT press.

Skogestad S and Postlethwaite I (2007) *Multivariable Feedback Control: Analysis and Design*, 2nd edition. Chichester, UK, New York: Wiley.

Stewart GE, Gorinevsky DM and Dumont GA (2003) Feedback controller design for a spatially distributed system: The paper machine problem. *IEEE Transactions on Control Systems Technology* 11(5): 612–628.

Stoica P and Moses RL (1997) *Introduction to Spectral Analysis*. Upper Saddle River, NJ: Prentice Hall .

Swaroop D and Hedrick JK (1996) String stability of interconnected systems. *IEEE Transactions on Automatic Control* 41(3): 349–357.

Titterton D, Weston J, Titterton DH and Weston JL (2004) *Strapdown Inertial Navigation Technology*, 2nd edition. Stevenage, UK: IET.

Varshavskaya P, Kaelbling LP and Rus D (2004) Learning distributed control for modular robots. In: *Proceedings of the IEEE/RSJ International conference on intelligent robots and systems*, Sendai, Japan, 28 September–1 October 2004, Vol. 3, pp.2648–2653.

Yim M, Shen WM, Salemi B, Rus D, Moll M, Lipson H, Klavins E and Chirikjian GS (2007) Modular self-reconfigurable robot systems (Grand challenges of robotics). *IEEE Robotics & Automation Magazine* 14(1): 43–52. doi10.1109/MRA.2007.339623.

Appendix A: Index to Multimedia Extensions

Extension	Media type	Description
1	Video	Indoor flight experiments
2	Video	Outdoor flight experiments

Appendix B: Physical parameters of a module

The physical parameters of a DFA module are listed in Table 3. All distance measurements are listed with respect to the geometric center of a module (see Figure 21), where the origin along the vertical axis sits at the base of the module.

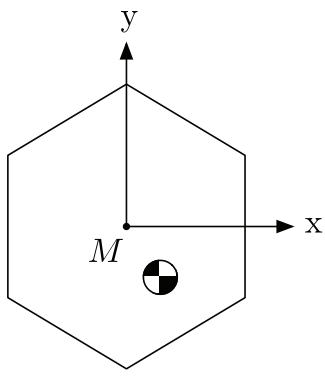


Fig. 21. The module's coordinate frame M is located at the geometric center of the chassis. The origin along the vertical axis (pointing out of the page) sits at the base of the module. The center of mass is located off center (see Table 3 for details).

Table 3. Physical parameters of a module.

Symbol	Description	Value
ℓ	Characteristic length	0.250 m
-	Position of the center of mass along the *x*-axis	0.009 m
-	Position of the center of mass along the *y*-axis	–0.0196 m
-	Position of the center of mass along the *z*-axis	0.0179 m
-	Position of the distance measurement sensor along the *x*-axis	–0.1086m
-	Position of the distance measurement sensor along the *y*-axis	–0.0652m
-	Position of the distance measurement sensor along the *z*-axis	0 m
m	Mass	0.255 kg
I_x	Mass moment of inertia around the *x*-axis	0.00179 kg.m^2
I_y	Mass moment of inertia around the *y*-axis	0.00185 kg.m^2
I_z	Mass moment of inertia around the *z*-axis	0.00322 kg.m^2
|c|	Force to torque conversion constant	0.0164 m

The module's center of mass was measured simply by hanging a module from three orthogonal sides off a gimbal. The intersection of all three lines drawn along the gravitational axis through the gimbal is the module's center of mass.

The mass moment of inertia I_a for $a \in \{x, y, z\}$ was computed by swinging a module around the axis a of a gimbal connected to a rotary encoder and measuring the period of oscillation. The period of oscillation T for a physical pendulum is given by Fowles (1977):

$$T = 2\pi \sqrt{\frac{I_a}{mgh}}, \tag{107}$$

where m is the mass of a module, g is the acceleration due to gravity, and h is the distance of the center of mass from the axis of rotation. It is straightforward to solve this expression for I_a. The Parallel Axis (Huygens–Steiner) Theorem (Kane and Levinson, 1985) was used transform this value to the module's coordinate frame M. This method is unable to measure off-diagonal terms in the mass moment of inertia tensor, which we assume to be negligible.

Appendix C: Physical parameters of the vehicle

Given the size and configuration of the vehicle, as well as the known physical parameters of a module (see Appendix B), the mass and principal mass moment of inertia of the vehicle can be computed.

Appendix C.1. Configuration database

Each module is capable of determinining the network topology and, in turn, the physical configuration of the vehicle that it may be connected to. This is accomplished by having each module flood the network with its own module identifier (ID), neighbor ID(s), and communication interface ID(s) (see Figure 22); the latter is crucial in determining the orientation and physical position of a module with respect to another. Upon receiving such information from another module, each module stores this data in a database, similar in structure to a Laplacian matrix (Merris, 1994). The Laplacian matrix can be used to find many properties of the network, such as connectivity and the number of spanning trees. The size of this matrix N × N denotes the size of the network (or the number of modules in the vehicle).

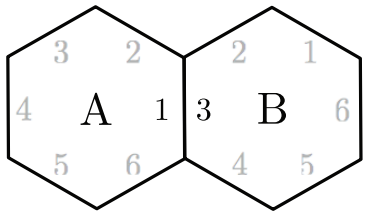


Fig. 22. For any module-to-module connection, there are two pieces of information needed in order to determine the position and orientation of a module with respect to another: (a) module IDs, shown here as A and B; and (b) communication interface ID, shown here as 1 on module A and 3 on module B.

This approach is by no means efficient; it is nevertheless simple to implement and reliable for the number of modules that we work with (up to 12). More efficient techniques similar to the breadth-first *Localization Algorithm* proposed in Gilpin et al. (2008) could, however, be implemented.

Appendix C.2. Centre of mass

Let the position vector $\mathbf{r}_k = (x_k, y_k, z_k)$ denote the center of mass of a module k with respect to a module's coordinate frame M (see Figure 23). This information is straightforward to obtain from the configuration database introduced previously. The vehicle's center of mass (which is also the location of the vehicle's body coordinate frame B) with respect to a module's coordinate frame M is the average of all these positions $\frac{1}{N}\sum_{k=1}^{N} \mathbf{r}_k$.

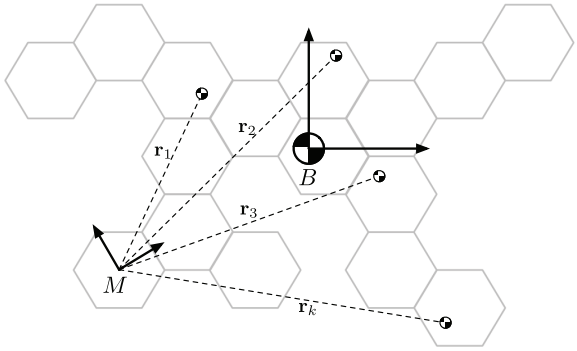


Fig. 23. The vehicle's center of mass B with respect to a module M can be computed by taking the average of all position vectors $\mathbf{r}_k$, which point from M to a module k's center of mass.

Appendix C.3. Mass

The total mass of the vehicle is simply the product of the total number of modules N and the mass of a single module m,

$$\sum_{k=1}^{\mathrm{N}} \mathrm{m} = \mathrm{Nm}. \tag{108}$$

Appendix C.4. Principal mass moment of inertia

Each module computes the principal mass moment of inertia with respect to an agreed upon module M, for example the module with the lowest ID. In this way, we can ensure that all modules compute the same principal mass moment of inertia, since the x and y axes are ambiguous.

The inertia tensor ${}_M\mathbf{J}$ of the vehicle with respect to the coordinate frame of module M is computed by repeated use of the Parallel Axis Theorem (Kane and Levinson, 1985) over the total number of modules N. The elements ${}_M\mathbf{J}_{i,j}$ of this inertia tensor for row i and column j is given by:

$${}_M\mathbf{J}_{i,j} = \sum_{k=1}^{\mathrm{N}} \left[\mathrm{I}_{i,j} + \mathrm{m}\left(||\mathbf{r}_k||^2 \delta_{i,j} - r_{k,i} r_{k,j} \right) \right], \tag{109}$$

where $\delta_{i,j}$ is the Kronecker delta, $\mathrm{I}_{i,j}$ are the elements of a module's inertia tensor, which is given by diag($\mathrm{I_x}, \mathrm{I_y}, \mathrm{I_z}$) (off diagonal terms are assumed to be negligible), and $r_{k,i}$ and $r_{k,j}$ are respectively the ith and jth element of vector $\mathbf{r}_k$.

The inertia tensor ${}_M\mathbf{J}$ is then transformed to the vehicle's center of mass (also the location of the vehicle's body coordinate frame B) by once again applying the Parallel Axis Theorem. We denote this new inertia tensor as ${}_B\mathbf{J}$. The vehicle's *principal* mass moment of inertia ($I_\mathrm{x}, I_\mathrm{y}, I_\mathrm{z}$) can then be computed from the eigendecomposition of the inertia tensor ${}_B\mathbf{J}$, which is given by

$$\begin{bmatrix} I_\mathrm{x} \\ I_\mathrm{y} \\ I_\mathrm{z} \end{bmatrix} = \mathbf{V}^{-1} \cdot {}_B\mathbf{J} \cdot \mathbf{V}, \tag{110}$$

where the matrix $\mathbf{V}$ is a rotation matrix that rotates the coordinate frame of the inertia tensor ${}_B\mathbf{J}$ to the principal axes of rotation.

Appendix D: Nominal thrust for an over-actuated system

In order to hover, the vehicle must satisfy four conditions: (1) the total thrust must counteract the acceleration due to gravity g, and (2)–(4). The net moment of the vehicle in roll, pitch, and yaw must be zero. In the case of the DFA, this implies having at least four modules with a pair of rotors that rotate CW and another pair that rotates CCW.

In general, however, the system may be over-actuated. In which case one should use a secondary criteria, such as minimizing the maximum amount of thrust produced. This can be formulated as a least squares problem, where the solution is in fact optimal (all thrusts are equal) when N is an even number. The system of equations can be written as

$$\begin{bmatrix} \mathrm{Nmg} \\ 0 \\ 0 \\ 0 \end{bmatrix} = \mathbf{P}^{\mathrm{T}} \begin{bmatrix} f_1 \\ \vdots \\ f_\mathrm{N} \end{bmatrix}, \tag{111}$$

where $\mathbf{P}$ is the configuration matrix given by (34). The solution to this requires taking the Moore–Penrose (or pseudoinverse) as follows:

$$\begin{bmatrix} f_1 \\ \vdots \\ f_\mathrm{N} \end{bmatrix} = \underbrace{\mathbf{P}(\mathbf{P}^{\mathrm{T}}\mathbf{P})^{-1}}_{\mathbf{Q}} \begin{bmatrix} \mathrm{Nmg} \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{112}$$

Appendix E: $\mathcal{H}_2$ -optimal controller

The $\mathcal{H}_2$-optimal controller used in Section 9 for performance comparison was computed using MATLAB's Robust Control Toolbox. The state-space description of the generalized plant in this case is given by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}_1\mathbf{w} + \mathbf{B}_2\mathbf{u}_d, \tag{113}$$

$$\mathbf{z} = \mathbf{C}_1\mathbf{x} + \mathbf{D}_{11}\mathbf{w} + \mathbf{D}_{12}\mathbf{u}_d, \tag{114}$$

$$\mathbf{y} = \mathbf{C}_2\mathbf{x} + \mathbf{D}_{21}\mathbf{w} + \mathbf{D}_{22}\mathbf{u}_d. \tag{115}$$

where all matrices are real matrices of compatible dimensions, and the state consists of position, Euler angles, translational velocities, Euler angle rates, and the control effort produced by each rotor,

$$\mathbf{x} = (x, y, z, \gamma, \beta, \alpha, \dot{x}, \dot{y}, \dot{z}, \dot{\gamma}, \dot{\beta}, \dot{\alpha}, u_1, \ldots, u_\mathrm{N}) \in \mathbb{R}^{12+\mathrm{N}}. \tag{116}$$

The state transition matrix $\mathbf{A}$ for nominal flight dynamics of the DFA, including motor dynamics as described in Section 3, is given by

$$\mathbf{A} = \begin{bmatrix} 0_{6\times6} & \mathbf{I}_{6\times6} & 0_{6\times\mathrm{N}} \\ \mathcal{A}_{2\times6} & 0_{2\times6} & 0_{2\times\mathrm{N}} \\ 0_{4\times6} & 0_{4\times6} & \mathbf{M}^{-1}\mathbf{P}^{\mathrm{T}} \\ 0_{\mathrm{N}\times6} & 0_{\mathrm{N}\times6} & -\omega_r \mathbf{I}_{\mathrm{N}\times\mathrm{N}} \end{bmatrix}, \tag{117}$$

where

$$\mathcal{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & \mathrm{g} & 0 \\ 0 & 0 & 0 & -\mathrm{g} & 0 & 0 \end{bmatrix}. \tag{118}$$

The desired control efforts contained in vector $\mathbf{u}_d \in \mathbb{R}^{\mathrm{N}\times1}$ is fed into the matrix $\mathbf{B}_2$, which scales these values by the rotor's bandwidth ω_r,

$$\mathbf{B}_2 = \begin{bmatrix} 0_{12\times\mathrm{N}} \\ \omega_r \mathbf{I}_{\mathrm{N}\times\mathrm{N}} \end{bmatrix}. \tag{119}$$

The state feedback vector $\mathbf{y} \in \mathbb{R}^{12\times1}$ of the vehicle is a function of the system's state $\mathbf{x}$, not including the control efforts, therefore

$$\mathbf{C}_2 = \begin{bmatrix} \mathbf{I}_{12\times12} & 0_{12\times\mathrm{N}} \end{bmatrix}. \tag{120}$$

The expected exogenous disturbances contained in vector $\mathbf{w} \in \mathbb{R}^{18\times1}$, which are described in Section 6, consist of unit variance process noise $\mathbf{w}_p$ and measurement noise $\mathbf{w}_m$,

$$\mathbf{w} = (\mathbf{w}_p, \mathbf{w}_m), \tag{121}$$

where

$$\mathbf{w}_p = (w_{\ddot{x}}, w_{\ddot{y}}, w_{\ddot{z}}, w_{\ddot{\gamma}}, w_{\ddot{\beta}}, w_{\ddot{\alpha}}), \tag{122}$$

$$\mathbf{w}_m = (w_x, w_y, w_z, w_\gamma, w_\beta, w_\alpha, \ldots \tag{123}$$

$$w_{\dot{x}}, w_{\dot{y}}, w_{\dot{z}}, w_{\dot{\gamma}}, w_{\dot{\beta}}, w_{\dot{\alpha}}). \tag{124}$$

The exogenous disturbance vector is scaled by the matrix $\mathbf{B}_1$ and the matrix $\mathbf{D}_{21}$ using the appropriate standard deviations listed in Table 1 of Section 6,

$$\mathbf{B}_1 = \left[\begin{array}{c|c} 0_{6\times6} & \\ \mathrm{diag}(\sigma_{\ddot{x}}, \sigma_{\ddot{y}}, \sigma_{\ddot{z}}, \sigma_{\ddot{\gamma}}, \sigma_{\ddot{\beta}}, \sigma_{\ddot{\alpha}}) & 0_{(12+\mathrm{N})\times12} \\ 0_{\mathrm{N}\times6} & \end{array}\right], \tag{125}$$

$$\mathbf{D}_{21} = \begin{bmatrix} 0_{12\times6} & \mathcal{D} \end{bmatrix}, \tag{126}$$

where,

$$\mathcal{D} = \mathrm{diag}(\sigma_x, \sigma_y, \sigma_z, \sigma_\gamma, \sigma_\beta, \sigma_\alpha, \ldots \tag{127}$$

$$\sigma_{\dot{x}}, \sigma_{\dot{y}}, \sigma_{\dot{z}}, \sigma_{\dot{\gamma}}, \sigma_{\dot{\beta}}, \sigma_{\dot{\alpha}}). \tag{128}$$

The output response $\mathbf{z}$, which is to be minimized through the synthesis of the $\mathcal{H}_2$-optimal controller, consists of the system's states and control effort weighted by the following two matrices, respectively:

$$\mathbf{C}_1 = \left[\begin{array}{cccccc|c} \mathrm{w_x} & 0 & 0 & 0 & 0 & 0 & \\ 0 & \mathrm{w_y} & 0 & 0 & 0 & 0 & \\ 0 & 0 & \mathrm{w_z} & 0 & 0 & 0 & 0_{4\times(6+\mathrm{N})} \\ 0 & 0 & 0 & 0 & 0 & \mathrm{w}_\alpha & \end{array}\right], \tag{129}$$

$$\mathbf{D}_{12} = \begin{bmatrix} 0_{4\times\mathrm{N}} \\ \mathrm{w}_u \mathbf{I}_{\mathrm{N}\times\mathrm{N}} \end{bmatrix}. \tag{130}$$

To compare the performance of this control method with the parameterized control strategy described in Section 8, the states are weighted similarly; the position weights $(\mathrm{w_x}, \mathrm{w_y}, \mathrm{w_z})$ are all set to 1 and the weight on the yaw-angle w_α is set to $\frac{1}{35}$ m/rad. Weight on the control effort, denoted w_u, is adjusted such that the control effort does not exceed the physical constraints, which can be formulated in the same way as described in Section 8.2.

There are no feed-through terms for the exogenous input $\mathbf{w}$ to the weighted output response $\mathbf{z}$ in (114), nor are there any for the control effort in (115). Hence, $\mathbf{D}_{11}$ and $\mathbf{D}_{22}$ are zero.