

Development of a Quadrotor Test Bed — Modelling, Parameter Identification, Controller Design and Trajectory Generation

Regular Paper

Wei Dong¹, Guo-Ying Gu^{1*}, Xiangyang Zhu¹ and Han Ding¹

¹ State Key Laboratory of Mechanical System and Vibration, Institute of Robotics, School of Mechanical Engineering, Shanghai Jiaotong University, Shanghai, China

* Corresponding author(s) E-mail: guguoying@sjtu.edu.cn

Received 4 December 2013; Accepted 16 October 2014

DOI: 10.5772/59618

© 2015 The Author(s). Licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

In this paper, a quadrotor test bed is developed. The technical approach for this test bed is firstly proposed by utilizing a commercial quadrotor, a Vicon motion capture system and a ground station. Then, the mathematical model of the quadrotor is formulated considering aerodynamic effects, and the parameter identification approaches for this model are provided accordingly. Based on the developed model and identified parameters, a simulation environment that is consistent with the real system is developed. Subsequently, a flight control strategy and a trajectory generation method, both of which are conceptually and computationally lightweight, are developed and tested in the simulation environment. The developed algorithms are then directly transplanted to the real system, and the experimental results show that their responses in the real-time flights match well with those from the simulations. This indicates that the control algorithms developed for the quadrotor can be preliminarily verified and refined through simulations, and then directly implemented to the real system, which could significantly reduce the experimental risks and costs. Meanwhile, real-time

experiments show that the developed flight controller can efficiently stabilize the quadrotor when external disturbances exist, and the trajectory generation approach can provide safe guidance for the quadrotor to fly smoothly through cluttered environments with obstacle rings. All of these features are valuable for real applications, thus demonstrating the feasibility of further development.

Keywords Quadrotor, Parameter Identification, Flight Control, Trajectory Generation, Test Bed Development

1. Introduction

The development of small and low-cost quadrotors has become popular in recent years owing to the progress of sensors, embedded controllers and communication technologies [1]. The quadrotors can work without skilled pilots in both military and civilian environments that are dangerous or where space is limited for human beings, such as traffic monitoring, reconnaissance, search and rescue, photography, etc. [2].

Several works have demonstrated the capabilities of the quadrotors for those missions, thus illustrating the significant opportunities and challenges that exist in this field. For example, in [1, 3–5], the quadrotors were adopted to transport, construct cubic structures and fly across obstacles. With cameras, the quadrotors could also do the jobs of simultaneous localization and mapping [6]. A quadrotor from the Federal Institute of Technology Zurich was controlled to support an invert pendulum [7]. In addition, two quadrotors have been designed for ball juggling [8]. A large-scale quadrotor (weighing 4 kg with a 1 kg payload) was proposed in [9], and the quadrotor was reported to be stabilized in [10]. Some other examples are aerobatics [11], cooperative manipulation [12], avian-inspired grasping [13], etc. As successful fulfilment of these real-time tasks is closely tied with the flight control of the quadrotors, several different flight controllers have been developed over the last decade. In the early years, the controllers were mainly developed to meet the requirements for high-performance trajectory tracking and way-point navigation. For this purpose, a number of conventional control algorithms, such as proportional-integral-derivative (PID), linear quadratic (LQ) [14], backstepping, sliding-mode [15], H_∞ with feedback linearization [16], hierarchy [17], fuzzy PID [18], model-based predictive [19] and the Lyapunov function design method [20], were introduced in the flight control, and they have established a solid foundation for subsequent research. Recently, as the demand for applying the quadrotors to the real world keeps increasing, flight controllers are required to be capable of coping with complex real flight scenarios where external disturbances exist such as wind gusts [2]. Therefore, high-performance disturbance rejection control and fault-tolerant control are attracting more and more attention [21, 22]. With well-constructed flight controllers, properly designed trajectory generation approaches are also essential for the quadrotors to cope with cluttered environments and fulfill specific missions. Some representative works of this field may be referred to [3, 23–26], where the desired trajectories were typically generated based on optimal control theory.

To assist this research and development, quadrotor test beds are undoubtedly important [1, 27]. In constructing such test beds, two approaches are mainly adopted. One approach is to build prototypes of the quadrotors as a ground test bench, such as the OS4 aircraft prototype [14, 15, 28, 29], the STARMAC quadrotor [30] and the 6-DOF test bench [27]. Such quadrotor prototypes are useful for studying the dynamics of the quadrotors at the early stage, and in this way the safety for parameter tuning and experimental tests can always be guaranteed. The other approach is to purchase commercial quadrotors to build the test beds, such as the GRASP test bed [1], QBall X4 [19] and Flying Machine Arena [31]. In these test beds, as the controllers for the quadrotors are directly implemented to the real systems, the real-time flight experiments can provide a convincing way to verify the effectiveness of the

developed controllers. In view of the state-of-the-art, it would be valuable if the test bed could inherit the merits from both of these two approaches, that is, providing real-time experiments and also getting rid of the potential risks at the early stage.

This work is therefore motivated to build a quadrotor test bed that can safely verify the developed algorithms in a ground station prior to real-time flights. This is achieved by simultaneously developing a real-time experimental environment and constructing a simulation environment that is consistent with the real system. For this purpose, a commercial quadrotor, a Vicon motion capture system and a control station are firstly integrated within an experimental field. Then the mathematical model of this quadrotor is derived, and the approaches for parameter identification are proposed. Utilizing this model, a simulation environment that is consistent with the real system is developed. In this way, the control algorithms can be developed, tested and refined in this simulation environment, and then directly converted into C-codes and implemented to the real system. As the control algorithms have been preliminarily verified by the simulations, in the development cycle one can significantly reduce and/or remove most of the experimental risks. Comparing the results of simulations and experiments, one can estimate the differences and further improve the parameters of the mathematical model.

To verify the effectiveness of the developed test bed, a conventional proportional-integral-derivative (PID) controller is firstly developed for the quadrotor. The performances of this controller in simulations and experiments are compared, and the feasibilities of the simulations can be thereby verified. Considering external disturbances always exist in real-world environments, a disturbance observer (DOB) is then implemented to enhance the performance of the flight control. The capabilities of this DOB-based controller are verified with a quadrotor suffering from the measurement zero-drift. To further verify the performance of the flight control under complex real-time flight scenarios, three rings are set up in this test bed to demonstrate a cluttered environment. In view of these obstacle rings, a trajectory generation method that is conceptually and computationally lightweight is proposed. The generated trajectory is sent as a reference to the position controller of the quadrotor. With such an implementation, the quadrotor can smoothly pass through the obstacle rings with only small deviations from its desired trajectory.

The contributions of this work are as follows. Firstly, the dynamic model of the quadrotor considering aerodynamic effects is developed, and the parameter identification approaches for the developed model are presented. Secondly, a quadrotor test bed that can effectively fill the gaps between simulations and experiments is developed. Within this test bed, control algorithms can be preliminarily developed and verified through simulations, and then directly implemented to the real system.

In this way, experimental risks and experimental costs can be significantly reduced. Finally, to demonstrate the functionalities of the developed test bed, a DOB-based PID controller and a novel trajectory generation approach, both of which are conceptually and computationally lightweight, are implemented in real-time applications for verification. It is observed that the real-time experimental results match well with the simulation results and the implemented control algorithms can cope well with the cluttered environment, all of which demonstrate the effectiveness of this development.

The remainder of this paper is organized as follows. Section 2 describes the technical approach. The dynamic model is derived in Section 3, and the controller is designed in Section 4. Section 5 presents the trajectory generation approach. Then the results from the simulations and experiments are shown in Section 6, and Section 7 concludes this work.

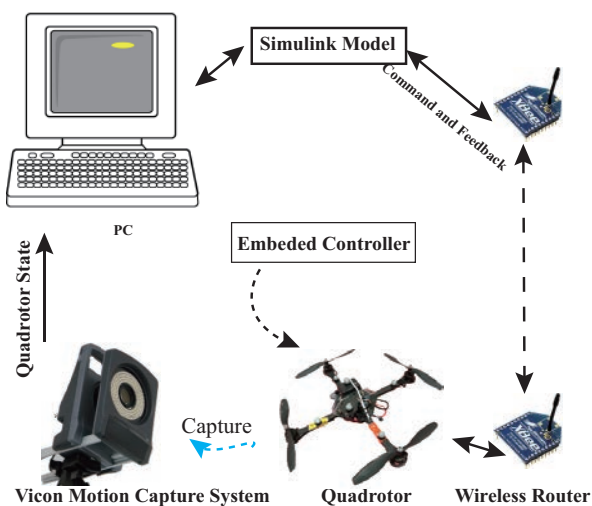


Figure 1. A Hummingbird quadrotor works with a Vicon motion capture system in the quadrotor test bed

2. Technical approach

Fig. 1 illustrates the structure of the quadrotor test bed developed in this work. This test bed comprises a quadrotor, a pair of wireless routers, a control station, a Vicon motion capture system and a simulation environment. The simulation environment is developed according to the dynamics of the quadrotor. In such an environment, control algorithms for the quadrotor can be developed, tested and optimized, and then directly implemented to the quadrotor. The quadrotor communicates with a ground control station via a couple of wireless routers at a frequency of 50 Hz. To guarantee the real-time performances, the control station is constructed in the Ubuntu operating system. The control station fetches the status data of the quadrotor from

the data server of the Vicon motion capture system and sends them to the embedded controller of the quadrotor. The motion capture system runs at 200 Hz and estimates the position of the quadrotor by kinematic fitting the attached markers to it. The block diagram of this test bed is shown in Fig. 2.

All of those components are integrated in an experimental field which is an indoor space with a volume of approximately $5 \times 5 \times 3.5 \text{ m}^3$. This field is protected by a nylon net with holes of size $12 \times 12 \text{ cm}^2$. A ground mat with thickness 2.5 cm protects the quadrotor from damage during crashes.

2.1 Quadrotor platform

The Hummingbird quadrotor from Ascending technologies is chosen as our quadrotor platform. Its two level processor structure and safety landing mode can help the researchers to pull back the quadrotor when it is out of control. One can develop control algorithms via a Matlab Simulink model, then translate them into C-codes, which can be compiled and directly flashed to the quadrotor's high-level processor. Another Simulink model is provided to send real-time commands to the quadrotor and collect debugging data. All of these features can significantly reduce the development time of the test bed.

Force/moment	Positive actuation	Negative actuation
Roll moment	F_4	F_2
Pitch moment	F_3	F_1
Yaw moment	$M_1 + M_3$	$M_2 + M_4$
Vertical thrust	$F_1 + F_2 + F_3 + F_4$	Gravity

Table 1. One control force and three control moments for a quadrotor

The Hummingbird quadrotor used in this work is shown in Fig. 3. With approximately 50 cm of tip-to-tip wingspan and 8 cm of height, it weighs 547 g with an additional 200 g payload capacity. When equipped with a 12 V/2100 mAh battery, it can undertake a continuous flight for approximate 20.

2.2 Vicon motion capture system

A Vicon motion capture system with eight cameras (two mega-pixels) is built to estimate the states of the quadrotor [32]. This system provides three benefits. Firstly, it is fast and accurate: running at 200 Hz, the system can measure the position of the quadrotor with a deviation of less than 0.5 mm, which is well beyond the requirements of the experiments. Secondly, it provides a convenient software interface: it works as a server and one can fetch the real-time data from it in a Matlab/Simulink model. Lastly, the system is robust: theoretically, the states of the quadrotor can be estimated when the attached markers are captured by at least three cameras.

3. Modelling and parameter identification

In this section, the dynamic model for the quadrotor is developed.

As shown in Fig. 4, there are mainly three kinds of force/moment acting on the quadrotor: gravity (mg), aerodynamic drag force (F_D) and thrust/moment (F_i / M_i) generated by the rotor. In this configuration, only the thrust/moment can be directly manipulated by steering the rotor speed, therefore they are chosen as the control inputs for the quadrotor system. As illustrated in Table 1, when the positive actuation is larger than the nega-

tive one, a positive control input for the corresponding motion is obtained, and vice versa. Therefore, four control inputs are defined as

$$\begin{cases} U_1 = F_1 + F_2 + F_3 + F_4 \\ U_2 = (F_3 - F_1)L \\ U_3 = (F_4 - F_2)L \\ U_4 = M_1 + M_3 - M_2 - M_4 \end{cases} \quad (1)$$

where L is the length of the rotor boom.

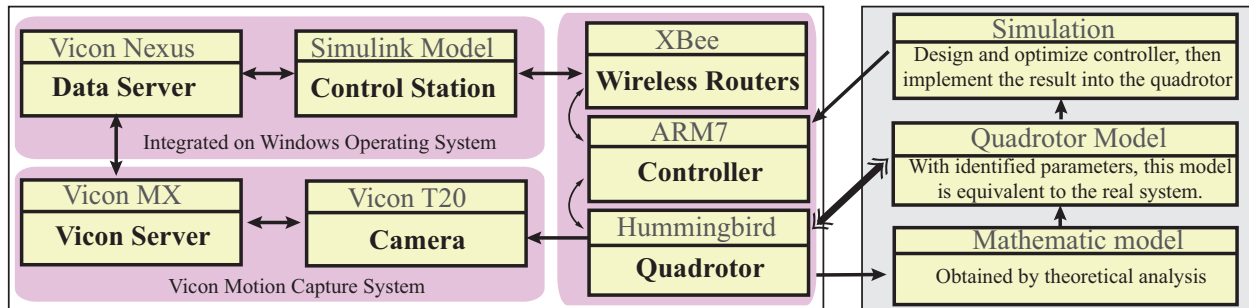


Figure 2. The block diagram of the quadrotor test bed

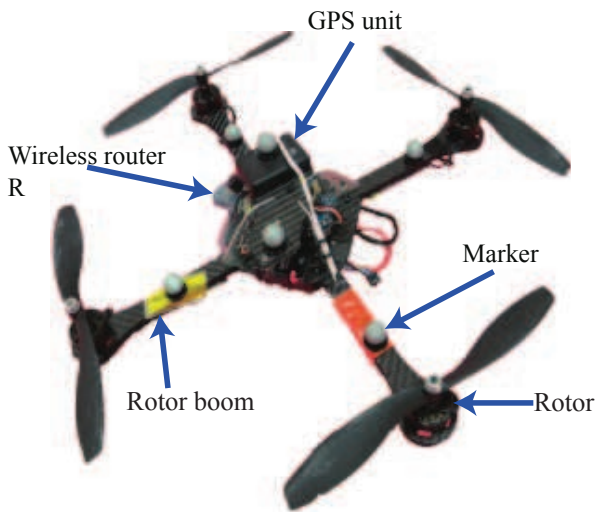


Figure 3. Hummingbird quadrotor

As shown in Fig. 4, these four control inputs are defined in a body fixed frame (X_B, Y_B and Z_B), whereas the position (\mathbf{r}) and speed ($\dot{\mathbf{r}}$) of the quadrotor are measured in an inertial frame (X_I, Y_I and Z_I). Therefore, the following matrix is required to describe the transformation from the body fixed frame to the inertial frame

$$R(\phi, \theta, \psi) = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - s\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (2)$$

where s stands for sine, c stands for cosine, and ϕ , θ and ψ represent attitude angles of roll, pitch and yaw, respectively.

According to the Newton-Euler formalism, the rigid body dynamics are governed by

$$\begin{cases} m\ddot{\mathbf{r}} = R \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + F_D \\ I\ddot{\mathbf{q}} = \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} - \dot{\mathbf{q}} \times I\dot{\mathbf{q}} \end{cases} \quad (3)$$

where m is the mass of the quadrotor, g is the local gravity constant, \mathbf{q} is attitude in body frame, and \mathbf{I} is the rotary inertia.

Item	Quantity
m (kg)	0.547
I_{xx} (kgm^2)	0.0033
I_{yy} (kgm^2)	0.0033
I_{zz} (kgm^2)	0.0058

Table 2. The measured mass and inertia

The mass m is measured by a digital scale. To measure the inertia of the quadrotor, experiments using a three-wire pendulum [33, 34] are conducted. For each axis, five tests

are carried out, and the average value is taken as the final result. The measured mass and inertia are shown in Table 2.

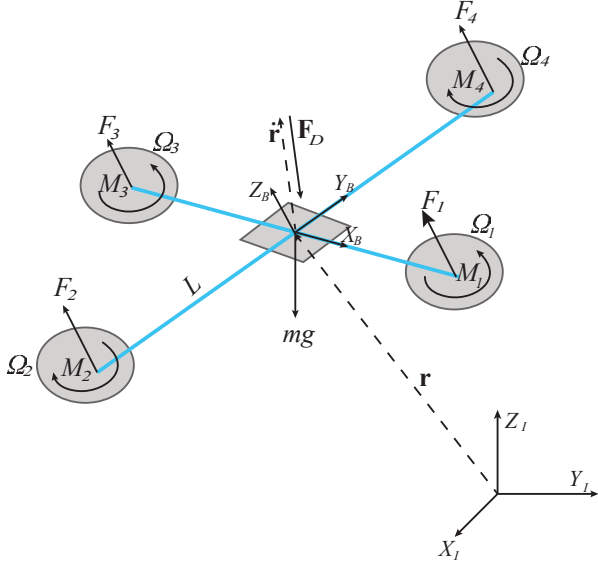


Figure 4. Force diagram of the quadrotor

3.1 Rotor model

The thrust F_i produced by each rotor is commonly calculated as [1, 35]

$$F_i = \rho C_T A \Omega_i^2 R^2 = k_T \Omega_i^2 \quad (4)$$

where C_T is the thrust coefficient, ρ is the air density, A is the rotor disk area, Ω_i is the speed of the rotor and R is the blade radius. Since the collective thrust equals to the gravity sensed by the quadrotor in the hovering state ($\phi \approx 0$ and $\theta \approx 0$), the parameter k_T can be estimated by $\frac{mg}{4\bar{\Omega}_h^2}$, where $\bar{\Omega}_h$ is the average speed of the rotors in the hovering state. In this work, k_T is determined to be $1.5 \times 10^{-7} \text{ N}/(\text{r}/\text{min})^2$.

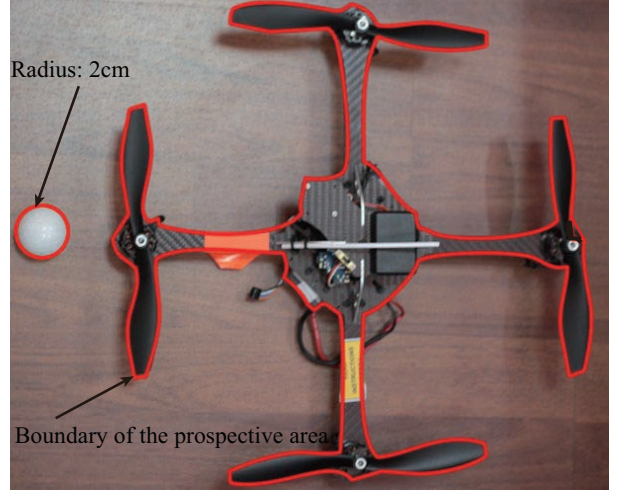
The moment M_i produced by each rotor is commonly calculated as [1, 35]

$$M_i = \rho C_Q A \Omega_i^2 R^3 = k_M \Omega_i^2 \quad (5)$$

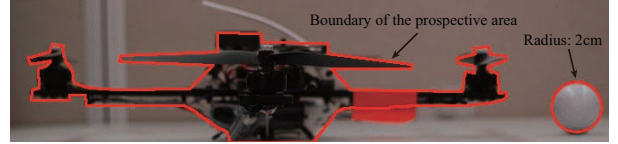
where C_Q is the drag moment coefficient. Based on Equation (3) and the measured inertia, the constant k_M can be determined by analysing the angular acceleration and the rotor speed from a step response of the yaw angle. In this work, k_M is determined to be $3.75 \times 10^{-9} \text{ N} \cdot \text{m}/(\text{r}/\text{min})^2$.

The rotor itself is generally described by a first order system [1, 29]

$$\dot{\Omega}_i = k_m (\Omega_i^{des} - \Omega_i) \quad (6)$$



(a) Estimate the area S_i in the top view



(b) Estimate the area S_s in the side view

Figure 5. The orthogonal prospective areas of the quadrotor is estimated using the imaging processing technique

where Ω_{des} is the desired speed of the rotor. k_m represents the speed of response [1, 29]. By analysing the flight data, k_m is as high as 1000 s^{-1} when $\Omega_{des} - \Omega_i$ is less than $1000 \text{ r}/\text{min}$. In practice, the rotor dynamics are relative fast for the controller development, so one can assume they are instantaneously achieved [3, 25].

3.2 Aerodynamic drag force

In general, the drag force is expressed as [35, 36]

$$F_D = \frac{1}{2} C_D \rho S \circ \text{or} |\dot{\mathbf{x}}| \quad (7)$$

where the operator \circ denotes the element wise product, C_D is the drag force coefficient and is approximate 1.0 for planar shape objects [36, 37], $\dot{\mathbf{x}}$ is the velocity of the quadrotor, and S is the reference area, which is the projected frontal area of the quadrotor along its flight direction.

The area S can be estimated based on the projected area S_i in the top view and the projected area S_s in the side view of the quadrotor, and this is handled as follows. As shown in Fig. 5, the pictures of the side view and top view of the quadrotor are taken with a ball of known radius b_r . In each of these figures, denoting the number of pixels within the profile of the quadrotor as n_q and the number of pixels within the profile of the ball as n_b , the quadrotor's projection area can be determined as $\pi b_r^2 n_q / n_b$. With this procedure, S_i is estimated as 0.022 m^2 and S_s is estimated as 0.011 m^2 . With known S_i and S_s in the near hovering state, the

reference area S can then be estimated along three orthogonal directions

$$S_x \approx S_s, S_y \approx S_s, S_z \approx S_t \quad (8)$$

where S_x, S_y , and S_z are the projected areas of S along X_I, Y_I and Z_I directions.

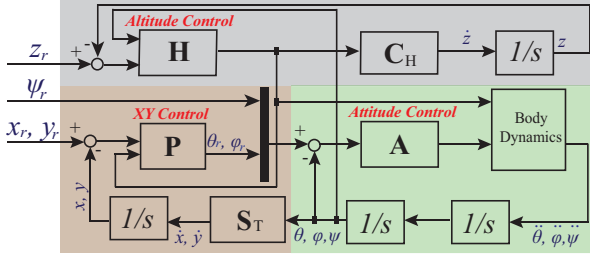


Figure 6. A cascade structure for control system

4. Controller design

The controller is constructed with a cascaded structure as shown in Fig. 6, where the attitude θ and ϕ are taken as the pseudo control inputs for the translational motion x and y . Based on this structure, one can develop algorithms to fully control a set of four states of the quadrotor. That is, either x, y, z, ψ or θ, ϕ, z, ψ can be fully controlled simultaneously.

4.1 Attitude controller

According to Eq. (3), as $I_{xx} = I_{yy}$ and ψ equals to zero in most cases, the angular acceleration can be expressed as [1, 19, 28, 38]

$$\begin{cases} \ddot{\phi} = \dot{\theta}\dot{\psi} \left(\frac{I_{yy} - I_{zz}}{I_{xx}} \right) + \frac{U_2}{I_{xx}} \approx \frac{U_2}{I_{xx}} \\ \ddot{\theta} = \dot{\phi}\dot{\psi} \left(\frac{I_{zz} - I_{xx}}{I_{yy}} \right) + \frac{U_3}{I_{yy}} \approx \frac{U_3}{I_{yy}} \\ \ddot{\psi} = \dot{\theta}\dot{\phi} \left(\frac{I_{xx} - I_{yy}}{I_{zz}} \right) + \frac{U_4}{I_{zz}} \approx \frac{U_4}{I_{zz}} \end{cases} \quad (9)$$

which means that the angular acceleration is proportional to the corresponding control input U_i ($i \in \{2, 3, 4\}$). As U_i is constructed with the thrust F_i or moment M_i , its dynamics should also be investigated to facilitate the controller design. To this end, the dynamics of attitude along with the corresponding U_i are reformulated as follows (since ψ is less important than ϕ and θ in practice and U_4 is constructed with M_i instead of F_i , this formulation only focuses on ϕ and θ , and one can use the same procedure for ψ).

Firstly, substituting Eq. (4) into Eq. (1) and taking the linearization of it via a Taylor series expansion in the near hovering state, one can obtain

$$U_i = k_T(\Omega_{i+1}^2 - \Omega_{i-1}^2) = 4k_T\bar{\Omega}_i\delta\Omega_i \approx 4k_T\bar{\Omega}_h\delta\Omega_i \quad (10)$$

where $i \in \{2, 3\}$, $\bar{\Omega}_h = (\Omega_{i-1} + \Omega_{i+1})/2$ and $\delta\Omega_i = (\Omega_{i-1} - \Omega_{i+1})/2 \approx \Omega_{i-1} - \bar{\Omega}_h = \bar{\Omega}_h - \Omega_{i+1}$.

Then, rewriting Eq. (6) as $\delta\dot{\Omega}_i = k_m(\delta\Omega_{des} - \delta\Omega_i)$, where $\delta\Omega_{des} = \Omega_{des} - \Omega^- h$ and $\delta\Omega_i = \Omega_i - \Omega^- h$, and taking Laplace transformation, one can obtain [19, 39]

$$\frac{\delta\Omega_i}{\delta\Omega_i^{des}} = \frac{k_m}{s + k_m} \quad (11)$$

In view of Eqs. (9), (10), and (11), one can obtain

$$\ddot{q} \approx \frac{k_a}{\tau_m s + 1} \delta\Omega_i^{des} \stackrel{\Delta}{=} \frac{k_a}{\tau_m s + 1} \delta u_{RPM} \quad (12)$$

That is

$$q / \delta u_{RPM} \approx \frac{k_a}{(\tau_m s + 1)s^2} \quad (13)$$

where $q \in \{\theta, \phi\}$, k_a is the gain of the process, τ_m represents the time constant, and δu_{RPM} is the variation of the revolution per minute (RPM) command for the rotor.

The parameters k_a can be estimated according to Eqs. (4) and (9), and τ_m is identified by an experiment as follows. With a preliminary designed controller in which the RPM input is scaled to 0-2, the quadrotor is conducted to fly with a random trajectory, and the data of this flight are recorded and analysed by the System Identification Toolbox in Matlab. With this procedure, the parameters are identified as $k_a = 32.2$ and $\tau_m = 32.2$.

With the identified result of Eq. (13), a PD controller is firstly chosen to stabilize the attitude [1, 40]. The closed loop transfer function of attitude with PD control is thereby obtain as

$$A_C(s) = \frac{32.2k_{ad}s + 32.2k_{ap}}{0.001s^3 + s^2 + 32.2k_{ad}s + 32.2k_{ap}k_{ad}} \quad (14)$$

where k_{ap} and k_{ad} are the proportional and derivative control gains, respectively.

In Eq. (14), there is a closed loop zero at $s = -\frac{k_{ap}}{k_{ad}}$, which will cause a large overshoot when a step disturbance arises unless the plant poles are heavily damped [40]. To elimi-

nate this effect, the configuration of this controller is modified as a P-D controller, as shown in Fig. 7. In this way, the closed loop transfer function is changed as follows

$$A_c(S) = \frac{32.2k_{ad}k_{ap}}{0.001s^3 + s^2 + 32.2k_{ad}s + 32.2k_{ad}k_{ap}} \quad (15)$$

According to the Routh's stability criterion, the stability of Eq. (15) can be guaranteed if both k_{ap} and k_{ad} are positive and $k_{ap} < 1000$ [41].

4.2 Position controller

Considering Equations (3) and (7), the dynamics of the position of the quadrotor is expressed as

$$\begin{cases} \ddot{x} = \frac{1}{m}(-k_x \dot{x}|\dot{x}| + U_1(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)) \\ \ddot{y} = \frac{1}{m}(-k_y \dot{y}|\dot{y}| + U_1(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)) \\ \ddot{z} = \frac{1}{m}(-k_z \dot{z}|\dot{z}| + U_1(\cos\phi\cos\theta)) - g \end{cases} \quad (16)$$

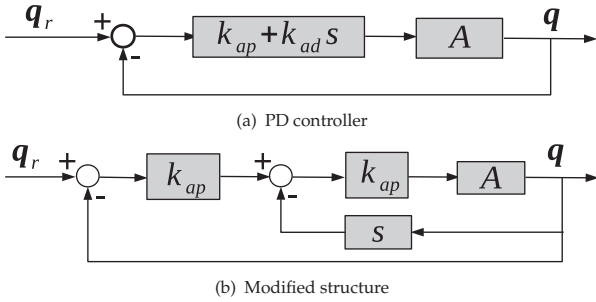


Figure 7. Attitude control configuration with inner loop

where $k_x = 1/2C_D\rho S_x k_y = 1/2C_D\rho S_y$ and $k_z = 1/2C_D\rho S_z$.

When the quadrotor flies with a steady speed, the second and higher order derivatives of position are zero. Taking the linearization of Eq. (16) at the steady state, $\ddot{x} = \ddot{y} = \ddot{z} = 0$, $k_x = k_{x0}$, $k_y = k_{y0}$, $k_z = k_{z0}$, $\dot{x} = \dot{x}_0$, $\dot{y} = \dot{y}_0$, $\dot{z} = \dot{z}_0$, $\phi = \phi_0$, $\theta = \theta_0$, and $U_1 = U_{10}$, one can obtain where small terms are neglected.

$$\begin{cases} \delta\dot{x} \approx \frac{U_{10}}{2k_{x0}|\dot{x}_0|}(\cos\phi_0\cos\theta_0\cos\psi\delta\theta + \cos\phi_0\sin\psi\delta\phi) \\ \delta\dot{y} \approx \frac{U_{10}}{2k_{y0}|\dot{y}_0|}(\cos\phi_0\cos\theta_0\cos\psi\delta\theta + \cos\phi_0\sin\psi\delta\phi) \\ \delta\dot{z} \approx \frac{1}{2k_{z0}|\dot{z}_0|}(\cos\phi_0\cos\theta_0)\delta U_1 \end{cases} \quad (17)$$

Eq. (17) can be rewritten in a state space form as

$$\delta\dot{r} \approx G\delta u \quad (18)$$

where $\delta\dot{r} = [\delta\dot{x}, \delta\dot{y}, \delta\dot{z}]^T$, $\delta u = [\delta u_1, \delta\theta, \delta\phi]^T$, and $G =$

$$\begin{bmatrix} 0 & \frac{U_{10}\cos\phi_0\cos\theta_0\cos\psi}{2k_{x0}|\dot{x}_0|} & \frac{U_{10}\cos\phi_0\sin\psi}{2k_{x0}|\dot{x}_0|} \\ 0 & \frac{U_{10}\cos\phi_0\cos\theta_0\cos\psi}{2k_{y0}|\dot{y}_0|} & \frac{U_{10}\cos\phi_0\sin\psi}{2k_{y0}|\dot{y}_0|} \\ \frac{\cos\phi_0\cos\theta_0}{2k_{z0}|\dot{z}_0|} & 0 & 0 \end{bmatrix}$$

As the matrix G is nontrivial when $\phi_0 < \frac{\pi}{2}$ and $\theta_0 < \frac{\pi}{2}$, the tracking error $\delta\dot{r}$ can be calculated as $\delta u = G^{-1}\delta\dot{r}$. Therefore, \dot{x} , \dot{y} can be controlled by ϕ , θ , and \dot{z} can be controlled by U_1 respectively. A PID controller is then constructed to stabilize the speed of the quadrotor as

$$\delta u^- = k_p\delta\dot{r} + k_i \int_0^t \delta\dot{r} dt + k_d \frac{d(\delta\dot{r})}{dt} \quad (19)$$

where k_p, k_i and k_d are control gains. As the translational motion (x and y) possesses different dynamics with the altitude (z), their control gains might also be different. In the following development, the control gains k_{pv}, k_{iv} and k_{dv} are selected to stabilize the translational motion, and the control gains k_{pv}, k_{iv} and k_{dv} are selected to stabilize the altitude.

Since external disturbances always exist in real applications, to eliminate these effects, a disturbance observer is developed as

$$\hat{W} = \text{sat}(\dot{r} - \int_0^t \mathbf{f}(u_0 + \delta u) dt, M_w) \quad (20)$$

where $\mathbf{f}(\cdot)$ represents the dynamic model of the quadrotor described in (16), $u_0 = [U_{10}, \theta_0, \phi_0]^T$, $\text{sat}(\epsilon) \triangleq \epsilon$ when $|\epsilon| < M_w$, and $\text{sat}(\epsilon) \triangleq M_w \epsilon / |\epsilon|$ when $|\epsilon| \geq M_w$.

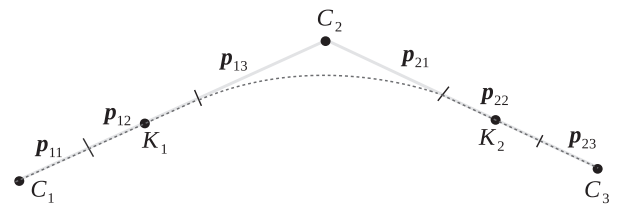


Figure 8. Trajectory generation of the quadrotor with key points and control points

With the observed disturbance \hat{w} in Eq. (20), the control input δu with DOB can be constructed with the observed disturbance \hat{w} as

$$\delta u = \delta u^- - G^{-1} \hat{w} \quad (21)$$

In this way, the effects of the external disturbances are compensated by \hat{w} , and the control input δu^- can simply treat the system as its nominal model Eq. (17).

Remark 1. The boundary M_w is assigned to prevent a unlimited increasing of \hat{w} , which might occur when the quadrotor stay on the ground. The details formulation of the observer Eq. (20) and its stability analysis can be found in our previous research [22].

With the stabilized speed, a proportional controller is developed to stabilize the position of the quadrotor as

$$\dot{r}_d = k_{pp}(r_d - r) \quad (22)$$

where \dot{r}_d is a virtual input which provides the reference for the developed speed controller Eq. (19), r_d is the desired position, and r is the actual position.

In the following discussions, Eqs.(19) and (22) are referred to as the PID controller, and Eqs. (21) and Eqs.(22) are referred to as the DOB-based PID controller.

5. Trajectory generation

In order to generate reference trajectories for the developed position controller to cope with cluttered environments, a trajectory generation approach is developed in this section.

To facilitate the development, the concepts of key point, control point and trajectory segment are firstly introduced as follows. The key point is a way point which the quadrotor must pass through in its specific missions. The control point is the intermediate point introduced to adjust the profile of the trajectory, and the quadrotor will choose whether to pass it or not according to the trajectory generation approach. The trajectory segment is the route between two adjacent control points. In this way, all of the key points can be included in the trajectory segments by a set of properly assigned control points.

As an illustration, a trajectory constructed with three control points (C_1, C_2, C_3) and two key points (K_1, K_2) is shown in Fig. 8. The dashed line indicates the desired trajectory, and it will be generated as follows.

Firstly, the trajectory segment from C_1 to C_2 is denoted as P_1 , and the trajectory segment from C_2 to C_3 is denoted as P_2 . Then, P_i ($i \in \{1, 2\}$) is further divided into three parts: p_{i1} , p_{i2} and p_{i3} . Denoting w_i as a ratio number, these three parts are expressed as $p_{i1} = p_{i3} = P_i w_i$ and $p_{i2} = P_i(1-2w_i)$. The quadrotor is designed to accelerate in p_{i1} , decelerate in p_{i3} and fly with constant speed in p_{i2} .

To meet these requirements, a time-dependent function that describes the position of the trajectory along the direction of P_i is expressed as follows

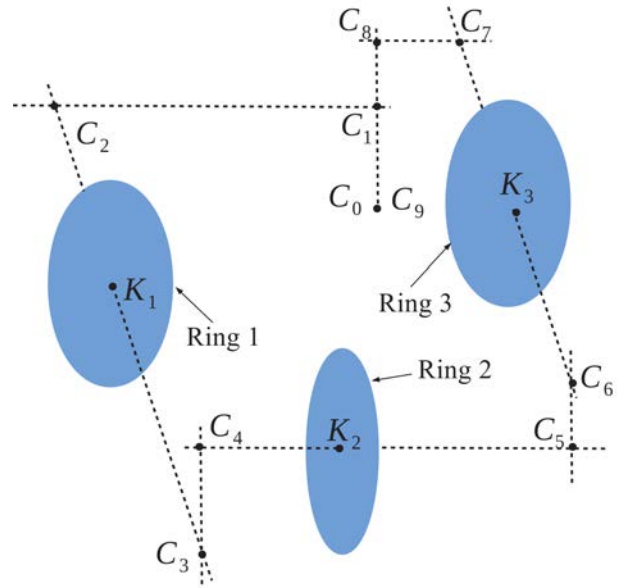


Figure 9. Key points and control points for the trajectory generation

$$T_i(P_i, t) = \begin{cases} 0 & t < t_{i0} \\ P_i w_i (1 - \cos(\lambda_i(t - t_{i0}))) & t_{i0} \leq t < t_{i1} \\ P_i w_i + P_i w_i \lambda_i (t - t_{i1}) & t_{i1} \leq t < t_{i2} \\ P_i (1 - w_i) + P_i w_i \sin(\lambda_i(t - t_{i2})) & t_{i2} \leq t < t_{i3} \\ P_i & t \geq t_{i3} \end{cases} \quad (23)$$

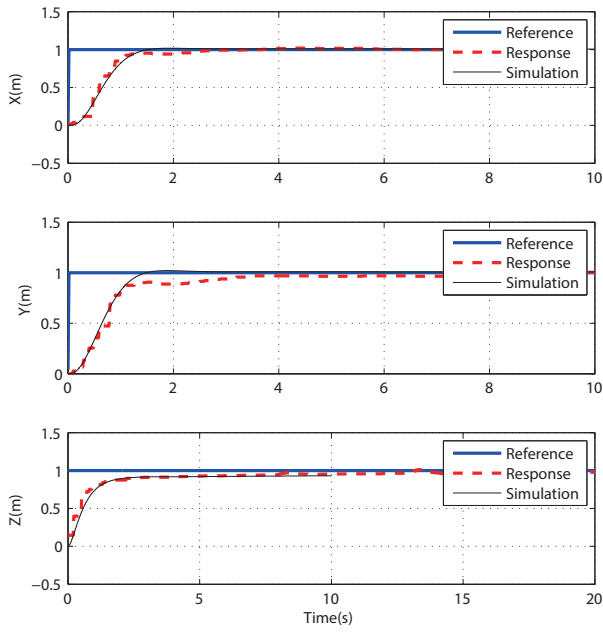
where $T_i(P_i, t)$ is the time-dependent function of the generated trajectory, t_{i0} is the start time, t_{i3} is the finish time, $t_{i1} = \frac{\pi}{2\lambda_i} + t_{i0}$, $t_{i2} = t_{i1} + \frac{P_i(1-2w_i)}{P_i w_i \lambda_i}$, $t_{i3} = t_{i1} + t_{i2} - t_{i0}$ and λ_i is a variable designed to modify the tracking speed of each trajectory segment.

With $T_i(P_i, t)$ generated with Eq. (23), the entire trajectory is constructed as

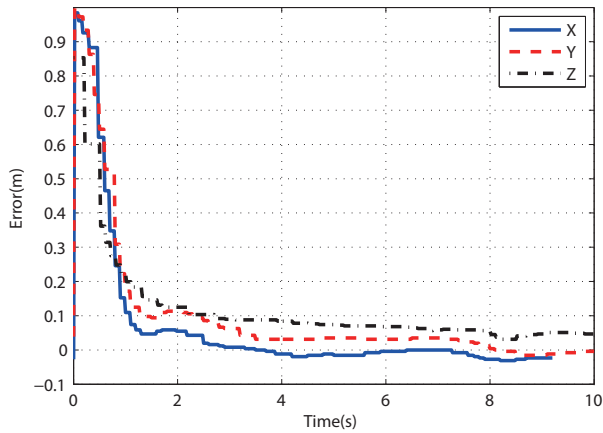
$$\Gamma(t) = T_i \otimes T_j \quad (24)$$

where $T_i(P_i, t)$ and $T_j(P_j, t)$ are abbreviated as T_i and T_j , $\Gamma(t)$ is the entire trajectory, and the operator \otimes is defined to smoothly connect T_i and T_j by enforcing $t_{j0} < t_{i3}$. In this way, the arc segment similar to that in Fig. 8 is established in the profile of the generated trajectory. To ensure the quadrotor safely passes through the key points, λ_i, λ_j are required to satisfy the inequalities $t_{j0} > t_{i2}$, $t_{j1} > t_{i3}$.

Apparently, the second order derivative of $\Gamma(t)$ defined in Eq. (24) is finite, and in this way, the smoothness of the flight can be guaranteed. Therefore, with known key points and control points, a trajectory can be directly generated according to Eqs. (23) and (24).



(a) The step response

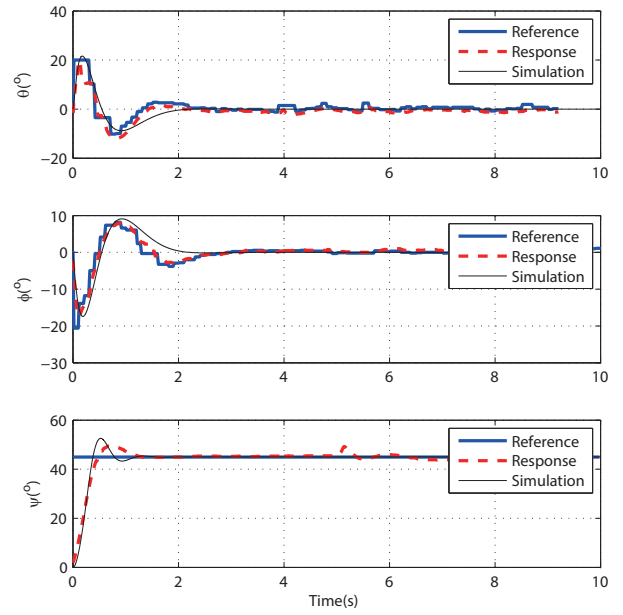


(b) Error plot

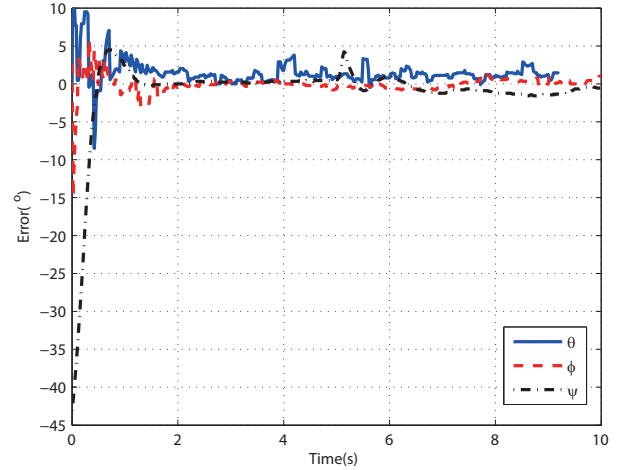
Figure 10. The step response and error plot for the position controller of the quadrotor

In this work, the quadrotor is designed to fly through the obstacle rings, so the key points are defined at the centre of each ring. To properly include these key points by trajectory segments, the control points are defined as follows. As indicated in Fig. 9, the location for taking off and landing is firstly selected to be a coincident point (C_0 and C_9). Then five lines are presented: the vertical line for taking off, central lines of three rings and the vertical line for landing. For each two adjacent lines, the perpendicular line between them can be determined, and the corresponding perpendicular feet ($C_1 \sim C_8$) are selected as the intermediate control points. With P_i defined as the trajectory segment between C_i and C_{i+1} , and T_i abbreviated for $T_i(P_i, t)$, the entire trajectory can be designed as

$$\Gamma(t) = T_0 \otimes T_1 \otimes T_2 \otimes T_3 \otimes T_4 \otimes T_5 \otimes T_6 \otimes T_7 \otimes T_8 \quad (25)$$



(a) Attitude response



(b) Attitude error

Figure 11. The attitude response of the quadrotor

6. Simulations and experiments

In this section, the merits of the development are demonstrated through simulations and experiments.

The structure of the simulation environment is illustrated in the right of Fig. 2. In this environment, the quadrotor model solving the dynamic equations of the quadrotor is firstly developed according to Eqs. (9) and (16). The inputs for this subsystem are RPM commands and the outputs are attitude and position of the quadrotor. The controllers are firstly implemented in the simulation environment for testing and parameter tuning. Then, to experimentally verify their performances, these developed controllers are directly converted into C-codes and flushed to the real system to perform real-time experiments.

All of the following discussions are based on the experimental results. It should be noted that one of the objectives of

this work is to develop a simulation environment that is consistent with the real system. Therefore, to demonstrate the feasibilities of the modelling and simulation environment, the simulation results and the experimental results of the same test are plotted together for comparison. To ensure the controllers are implemented in exactly the same way in both simulations and experiments, the parameters of the controllers in the simulations and experiments are selected with the same values, which are provided in Table 3.

Item	k_{ap}	k_{ad}	k_{pt}	k_{dt}	k_{it}	k_{ph}	k_{dh}	k_{ih}	k_{pp}	M_w
Value	10	10	0.6	0.15	0.4	0.3	0.45	0.1	1	1

Table 3. Parameters of controllers

6.1 Step responses of the position controller

The step responses of the position controller are shown in Fig. 10, and the corresponding pseudo control inputs θ and ϕ are provided in Fig. 11. As the yawing angle is independent from other angles, its step response is also shown in Fig. 11.

The rising time of the translational motion controller is approximately 0.9 seconds. Apart from this, the responses are not exactly the same for the X and Y axes. For the X axis, the settling time with a steady state error of 5% is 2.5 seconds, but is 3.5 seconds for the Y axis. Besides, the position along the X axis can be stabilized with an error of 2.0%, which is less than that of the Y axis. This is caused by mechanic tolerances on the Y axis, since the rotor booms along the Y axis are a bit looser than the other pair. In order to avoid rapid taking off and landing, the gain of altitude controller is smaller. Therefore, its steady state error is larger and the response is also slower. The rising time in the Z direction is 2 seconds, and the settling time with a steady state error of 5% is 8.0 seconds.

The attitude control shows a rapid and smooth response. The maximum value of θ and ϕ in the whole process is 20 degrees, which satisfies the small angles assumption. The controller closely tracks the reference, with a time delay of less than 0.1 seconds. For the yawing angle, the rising time is less than 0.3 seconds. With an overshoot of 10%, the quadrotor is finally stabilized with a deviation of less than 1% within 1.2 seconds. At the 5th second, when a step input for the X axis is set for the position controller, a perturbation, which might be caused by unmodelled coupling dynamics of the real system and the input saturation of the rotors [8], is found in the yawing angle. In this case, the attitude controller can stabilize the yawing angle within 0.5 seconds.

These results show the quadrotor can be controlled well within the test bed, which verifies the feasibilities of the modelling, parameter identification and controller development. The fact that the results from the experiments match well with those from the simulations implies that the controllers can be firstly verified within the simulation

environment, and then directly implemented to the real system.

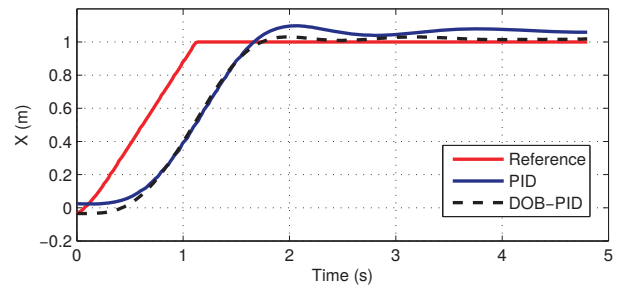


Figure 12. Comparative experiments between PID controller and DOB-based PID controller

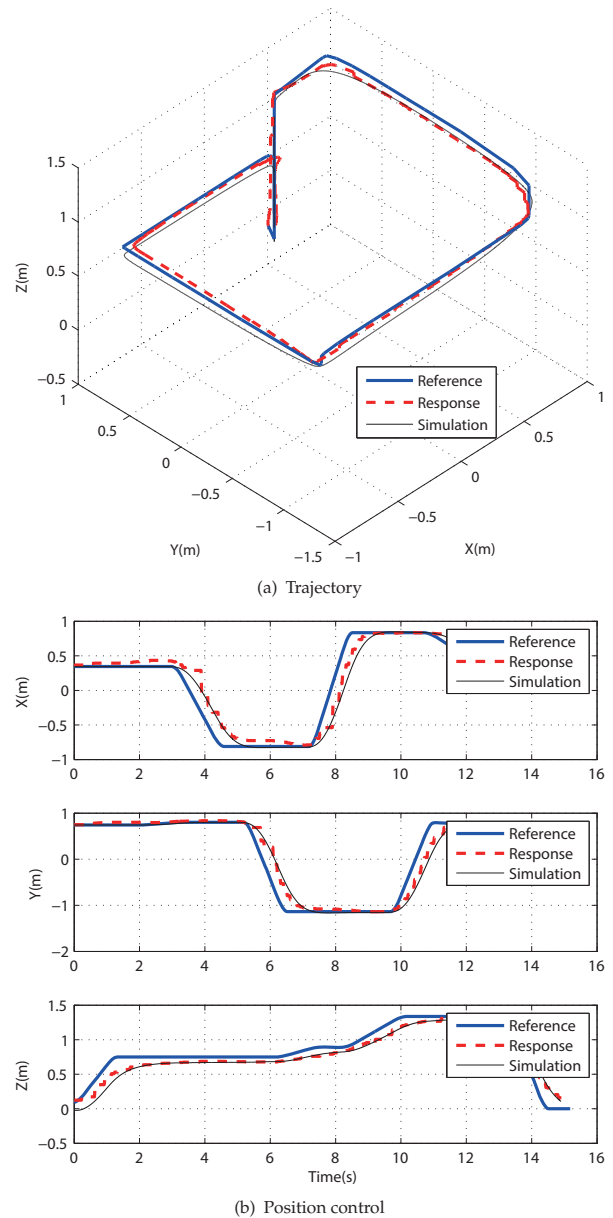
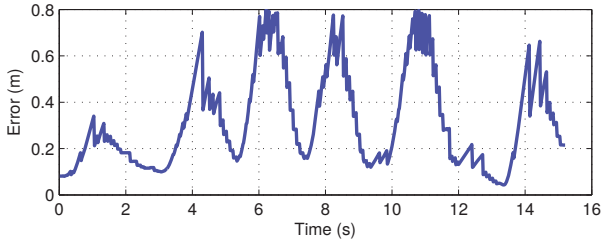
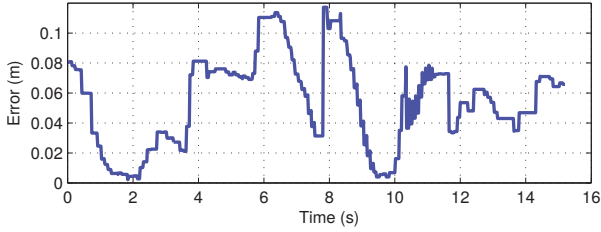


Figure 13. The quadrotor tracks a generated trajectory

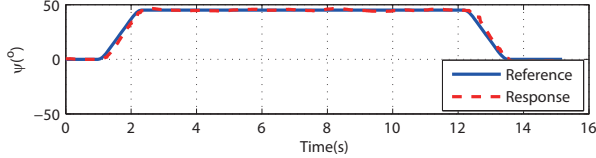
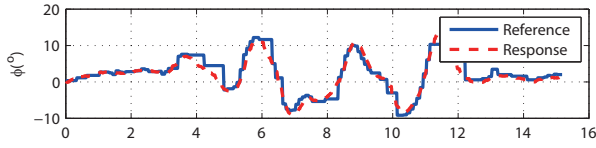
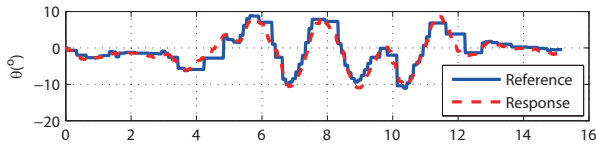


(a) The time history of the error along the trajectory

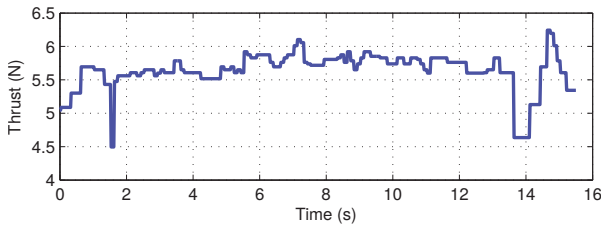


(b) Cross-tracking error

Figure 14. The time history of tracking error along the trajectory and the cross-tracking error



(a) Attitude control



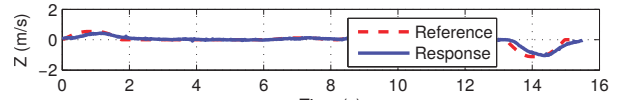
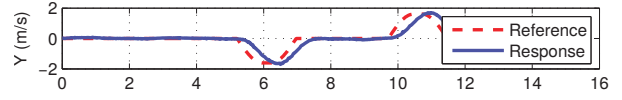
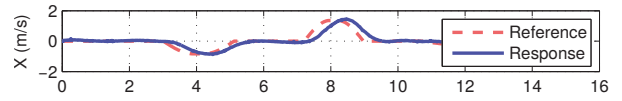
(b) Thrust control

Figure 15. The time histories of the control inputs

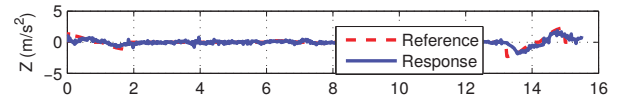
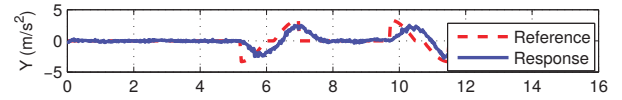
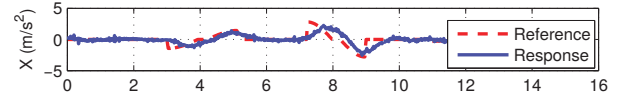
6.2 Disturbance rejection

The capabilities of disturbance rejection of the developed DOB-based PID controllers are verified with a quadrotor suffering from measurement zero-drift. After serving over six months in the test bed, structural degradations, e.g., the rigidity of rotor booms decreases, are found in the quadrotor. This causes the gyroscope to stray in the horizontal plane when the quadrotor is in the hovering state. As a

result, extra deviations are introduced into the feedback system, or equivalently, nonzero inputs are required to maintain the quadrotor in the hovering state. This will negatively affect the performance of the conventional PID controller. As shown in Fig. 12, the steady state error of the PID controller is as high as 8%. By contrast, as the DOB can estimate these disturbances and thereby effectively eliminate them in the control strategy, the steady state error of the DOB-based PID controller is less than 2.5%. This indicates that the developed DOB-based PID controller can still guarantee the performance of the quadrotor even if external disturbances arise.



(a) Speed history



(b) Acceleration history

Figure 16. The speed and acceleration plot when the quadrotor tracks a generated trajectory

6.3 Trajectory generation

To verify the effectiveness of the trajectory generation approach and the consistency between simulations and experiments, the quadrotor is conducted to track the generated trajectory to fly across three obstacle rings. Each ring's inner diameter is 0.80 m, and their coordinates are estimated by the Vicon motion capture system with $K_1 = (-0.81, -0.36, 0.75)$, $K_2 = (0.20, -1.14, 0.89)$ and $K_3 = (0.83, -0.12, 1.34)$, measured in metres. The two control points selected as the taking off and landing locations are $C_0 = (0.8, 0.8, 0)$ and $C_9 = (0.8, 0.8, 0)$, which can be modified for each experiment. According to the approach proposed in Section 5, all intermediate control points are

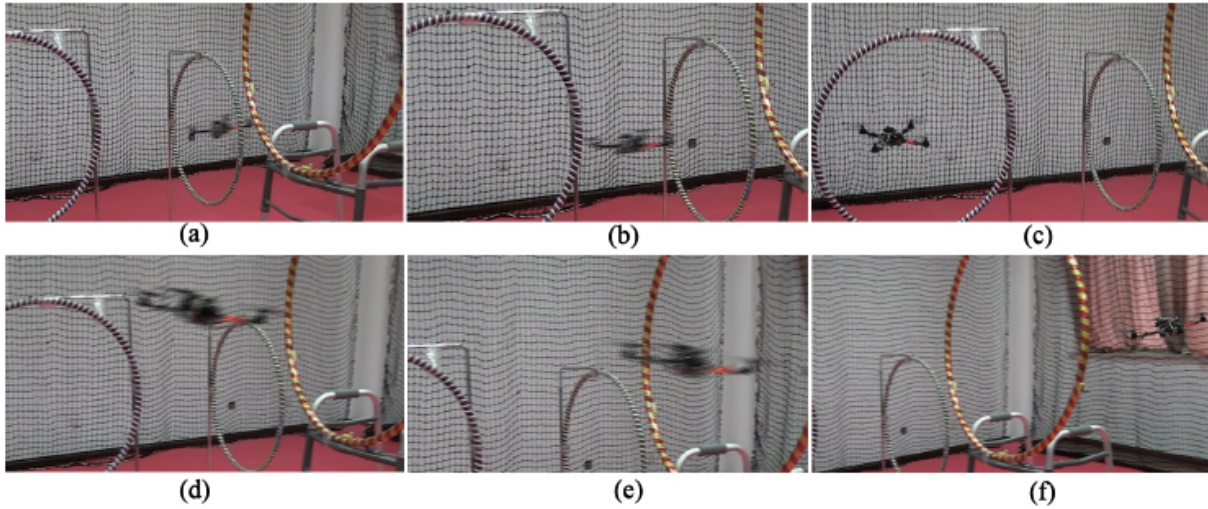


Figure 17. The scenario shows the quadrotor flies across three obstacle rings from (a) to (f)

$$\begin{aligned}
 C_1 &= (c_{0x}, c_{0y}, k_{1z}), & C_2 &= (k_{1x}, c_{0y}, k_{1z}) \\
 C_3 &= (k_{1x}, k_{2y}, k_{1z}), & C_4 &= (k_{1x}, k_{2y}, k_{2z}) \\
 C_5 &= (k_{3x}, k_{2y}, k_{2z}), & C_6 &= (k_{3x}, k_{2y}, k_{3z}) \\
 C_7 &= (k_{3x}, c_{9y}, k_{3z}), & C_8 &= (c_{9x}, c_{9y}, k_{3z})
 \end{aligned}$$

With $w_i = 0.2$, the trajectory is generated according to Equations (23) and (25). With an Intel Core i5-3470 processor, the entire trajectory can be generated in Matlab within 0.05 seconds, which is fast enough for real-time applications. The corresponding curves are shown in Fig. 13(a). It can be seen that smooth transitions are implemented between adjacent trajectory segments, and the reference trajectory offers a safe guidance for the quadrotor to fly through these obstacle rings. The trajectory is sent to the quadrotor according to the time schedule, and Fig. 13(a) shows the quadrotor tracking the desired trajectory very well. The responses of the corresponding position control are also shown in Fig. 13(b), which shows the details of the trajectory tracking process. It can be seen that the position of the quadrotor falls behind the moving reference point. Therefore, the dynamic error in this process is as high as 0.8 m, as shown in Fig. 14(a). However, the feasibility of the development is not affected by this shortage. As indicated in Fig. 14(b), the cross-track error is less than 0.12 m, which shows that the reference trajectory is closely tracked and therefore enables the quadrotor to safely pass through the settled rings in the test bed.

The corresponding control inputs are shown in Fig. 15. The attitude θ and ϕ are the pseudo control inputs for the translational movement, and the thrust steers the altitude movement. It should be noted that in the trajectory tracking process, the yawing angle is $\pi/4$, therefore, θ and ϕ cooperatively steer the translational motion in the way of Eq. (16).

The time history of the speed and the acceleration of the quadrotor are shown in Fig. 16. The maximum speed during the flight is approximately 1.5 m/s, and the maximum acceleration is 2.5 m/s² through the entire flight. All of this indicates that the quadrotor can smoothly pass through cluttered environments at moderate speed.

The scenarios of the real-time experiments are shown in Fig. 17.

From all of the plots, it can be seen that the provided trajectory generation approach can provide a feasible guidance for the quadrotor to fly through the cluttered environment, and the developed PID controller can track the generated trajectory well. In addition, the simulations provide good predictions for the controller development. One can therefore verify the performances of the control algorithms through simulations prior to implementing them to the real systems, which would significantly reduce time and cost of development.

7. Conclusion

This work has developed a quadrotor test bed, which consists of a commercial quadrotor platform, a Vicon motion capture system, a control station and a simulation environment. The simulation environment is developed according to the quadrotor's dynamics, thus the behaviours of the quadrotor can be preliminarily verified through this environment. To illustrate such functionality, simulations and experiments are performed comparatively for the flight controllers and trajectory generation. The results show that the results from the experiments match well with those from the simulations, and the quadrotor can be well controlled to smoothly fly through the cluttered environment. This indicates that one can easily develop and refine these control algorithms in the simulation environment,

and transplant the developed algorithms to the real system with no gaps in the controller implementation process. Therefore, the risks during experiments are minimized, and the development time can be reduced. All of these facts demonstrate the merits of this development.

In future, more research will be carried out to verify and improve the dynamic model of the quadrotor, and the trajectory generation approach will be improved for high-speed flight and obstacle avoidance.

8. Acknowledgements

This work was funded by the Open Foundation of the State Key Laboratory of Fluid Power Transmission and Control.

9. References

- [1] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar. The grasp multiple micro-UAV testbed. *IEEE Robotics & Automation Magazine*, 17(3):56–65, 2010.
- [2] V. Kumar and N. Michael. Opportunities and challenges with autonomous micro aerial vehicles. *The International Journal of Robotics Research*, 31(11):1279–1291, 2012.
- [3] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2520–2525, 2011.
- [4] D. Mellinger, M. Shomin, N. Michael, and V. Kumar. Cooperative grasping and transport using multiple quadrotors. In *Proceedings of the International Symposium on Distributed Autonomous Robotic Systems*, pages 545–558, 2010.
- [5] Q. Lindsey, D. Mellinger, and V. Kumar. Construction with quadrotor teams. *Autonomous Robots*, pages 1–14, 2012.
- [6] S. Shen, N. Michael, and V. Kumar. Autonomous multi-floor indoor navigation with a computationally constrained MAV. In *Proceedings of the IEEE International Conference on Robotics and automation*, pages 20–25, 2011.
- [7] M. Hehnand and R. D’Andrea. A flying inverted pendulum. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 763–770, 2011.
- [8] M. Muller, S. Lupashin, and R. D’Andrea. Quadrocopter ball juggling. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5113–5120, 2011.
- [9] P. Pounds and R. Mahony. Design principles of large quadrotors for practical applications. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3265–3270, 2009.
- [10] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar. Stability of small-scale UAV helicopters and quadrotors with added payload mass under PID control. *Autonomous Robots*, pages 1–14, 2012.
- [11] S. Lupashin and R. D’Andrea. Adaptive fast open-loop maneuvers for quadrocopters. *Autonomous Robots*, 33(1-2):89–102, 2012.
- [12] R. Ritz, MW Muller, M. Hehn, and R. D’Andrea. Cooperative quadrocopter ball throwing and catching. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4972–4978, 2012.
- [13] J. Thomas, J. Polin, K. Sreenath, and V. Kumar. Avian-inspired grasping for quadrotor micro UAVs. In *Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V06AT07A014–V06AT07A014. American Society of Mechanical Engineers, 2013.
- [14] S. Bouabdallah, A. Noth, and R. Siegwart. PID vs LQ control techniques applied to an indoor micro quadrotor. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2451–2456, 2004.
- [15] S. Bouabdallah. *Design and control of quadrotors with application to autonomous flying*. PhD thesis, Lausanne Polytechnic University, Switzerland, 2007.
- [16] A. Mokhtari, A. Benallegue, and B. Daachi. Robust feedback linearization and GH controller for a quadrotor unmanned aerial vehicle. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1198–1203, 2005.
- [17] Y. Zhang, K. Kondak, T. Lu, J. Du, M. Bernard, and G. Wang. Development of the model and hierarchy controller of the quad-copter. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 222(1):1–12, 2008.
- [18] D. Gautam and C. Ha. Control of a quadrotor using a smart self-tuning fuzzy PID Controller. *International Journal of Advanced Robotic Systems*, 10:1–14, 2013.
- [19] M. Abdolhosseini, Y. M. Zhang, and C. A. Rabbath. An efficient model predictive control scheme for an unmanned quadrotor helicopter. *Journal of Intelligent & Robotic Systems*, pages 1–12, 2012.
- [20] A. Tayebi and S. McGilvray. Attitude stabilization of a VTOL quadrotor aircraft. *IEEE Transactions on Control Systems Technology*, 14(3):562–571, 2006.
- [21] Z. Dydek, A. Annaswamy, and E. Lavretsky. Adaptive control of quadrotor UAVs: A design trade study with flight evaluations. *IEEE Transactions on Control Systems Technology*, 21(4):1400–1406, 2013.
- [22] W. Dong, G. Gu, X. Zhu, and H. Ding. High-performance trajectory tracking control of a quadrotor with disturbance observer. *Sensors and Actuators A: Physical*, 211:67–77, 2014.

- [23] I. Cowling. *Towards autonomy of a quadrotor UAV*. PhD thesis, Cranfield University, Bedfordshire, 2008.
- [24] I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, and A. K. Cooke. A prototype of an autonomous controller for a quadrotor UAV. In *European Control Conference*, pages 1–8, 2007.
- [25] D. Mellinger, N. Michael, and V. Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. *The International Journal of Robotics Research*, 31(5):664–674, 2012.
- [26] M. W. Mueller, M. Hehn, and R. D’Andrea. A computationally efficient algorithm for state-to-state quadcopter trajectory generation and feasibility verification. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3480–3486. IEEE, 2013.
- [27] Y. Yu and X. Ding. A quadrotor test bench for six degree of freedom flight. *Journal of Intelligent & Robotic Systems*, pages 1–16, 2012.
- [28] S. Bouabdallah, P. Murrieri, and R. Siegwart. Design and control of an indoor micro quadrotor. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 5, pages 4393–4398, 2004.
- [29] S. Bouabdallah and R. Siegwart. Full control of a quadrotor. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 153–158, 2007.
- [30] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pages 1–20, 2007.
- [31] S. Lupashin, M. Hehn, M. W. Mueller, A. P. Schoellig, M. Sherback, and R. D’Andrea. A platform for aerial robotics research and demonstration: The flying machine arena. *Mechatronics*, 24(1):41–54, 2014.
- [32] Vicon. Vicon mx systems. <http://www.vicon.com/>, 2012. Accessed on 10 Oct 2012.
- [33] Wikipedia. Moment of inertia. http://en.wikipedia.org/wiki/Moment_of_inertia, 2014. Accessed on 10 Jun 2014.
- [34] W. Gracey. The experimental determination of the moments of inertia of airplanes by a simplified compound-pendulum method. NACA Technical Note, (1629), 1948.
- [35] J. G. Leishman. *Principles of helicopter aerodynamics*. Cambridge University Press, 2006.
- [36] Wikipedia. Drag equation. http://en.wikipedia.org/wiki/drag_equation, 2013. Accessed on 11 Nov 2013.
- [37] Wikipedia. Drag coefficient. http://en.wikipedia.org/wiki/drag_coefficient, 2013. Accessed on 11 Nov 2013.
- [38] W. Dong, G. Y. Gu, X. Y. Zhu, and H. Ding. Modeling and control of a quadrotor UAV with aerodynamic concepts. In *ICIUS 2013 : International Conference on Intelligent Unmanned Systems*, pages 377–382, 2013.
- [39] D. Bawek. Design and implementation of a control system for a quadrotor MAV. Master’s thesis, The University of Maryland, Maryland, 2012.
- [40] H. Lim, J. Park, D. Lee, and H. J. Kim. Build your own quadrotor: Open-source projects on unmanned aerial vehicles. *IEEE Robotics & Automation Magazine*, 19(3):33–45, 2012.
- [41] K. Ogata. *Modern control engineering*. Pearson Education Asia Limited and Tsinghua University Press, Beijing, 4th edition, 2005.