International Journal of Control, Automation and Systems 15(X) (2017) 1-12
http://dx.doi.org/10.1007/s12555-015-0289-3
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Trajectory Tracking Control of Multirotors from Modelling to Experiments: A Survey

Hyeonbeom Lee and H. Jin Kim*

Abstract: Multirotors have received a great attention from researchers and the general public, as a platform on which various ideas can be easily demonstrated. This paper aims to provide background materials by categorizing various representations of multirotor dynamics and existing control approaches for multirotor control. First, many ways of expressing the translation and the attitude dynamics of a quadrotor UAV are described. Second, linear and nonlinear control laws are reviewed considerably. Finally, we show various types of flight test-beds configured for validating the controller. In experiments, the performance of linear and nonlinear controller are described.

Keywords: Autonomous control, multirotor, quadrotor, review, trajectory tracking.

1. INTRODUCTION

Multirotor research has been accelerated during the last decade, although the first quadrotor concept, the Breguet-Richet quadrotor helicopter Gyroplane No.1, was built in 1907 [1]. Affordability and simple hardware structure combined with inexpensive, simple interaction sensors and processors have promoted the rapidly growing interests in multirotors as an easy-to-work-with platform on which various ideas can be easily implemented. Many researchers demonstrated various applications of multirotors, such as surveillance, aerial photograph, grasping and moving an object and acrobatic maneuvers for entertaining, etc., as shown in Fig. 1 and Table. 1. Still, discussion of the existing controllers will facilitate the utilization of the full capability of multirotors in wider range of applications.

Although multirotor is a nonlinear and under-actuated system with highly coupled states, basic linear controllers such as a PID or LQR have been successful for autonomous multirotor flight to some degree. Due to their simplicity in design and implementation, they have been widely used in many cases [2–8]. To improve robustness against disturbances or modeling error, linear robust controllers including H_2 or H_∞ were proposed in [9–14]. However, they were still derived based on linearized dynamics around some operation points. Their demonstrations were mostly limited to non-agile movements.

As control techniques to handle nonlinearity more efficiently, nonlinear controllers are proposed using feedback linearization, backstepping, geometric control techniques, etc. Feedback linearization approaches such as input-output linearization and state-space linearization [19] were proposed for multirotors in [20, 21]. Backstepping control, which stabilizes dynamics naturally rather than cancelling the nonlinear dynamics as in feedback linearization [22], was applied to a multirotor in [23–27] using sliding mode techniques and Lyapunov stability theory. A dynamic inversion control law with feedback linearization and backstepping control was addressed in [28]

Fig. 1. Various application of multirotors (clockwise) : aerial manipulation [15], entertainment [16], military purpose [17] and autonomous exploration [18].

Manuscript received July 27, 2015; revised December 13, 2015; accepted January 13, 2016. Recommended by Associate Editor DaeEun Kim under the direction of Editor Euntai Kim. This work was supported by the Technology Innovation Program (10051673) funded by the Ministry of Trade, industry and Energy(MI, Korea) and the program of Development of Space Core Technology through the National Research Foundation of Korea funded by the Ministry of Science, ICT and Future Planning (NRF-2015M1A3A3A05027630).

Hyeonbeom Lee and H. Jin Kim are with the School of Mechanical and Aerospace Engineering, Seoul National University, 1 Gwanack-ro, Gwanack-gu, Seoul, Korea (e-mails: {koreaner33, hjinkim}@snu.ac.kr).
* Corresponding author.

resulting in the controller that only requires the second derivative of the desired output and first derivative of theerror. A geometric tracking controller was derived based on the special Euclidean group $SE(3)$ to avoid singularities and complexity which can happen in other control laws [29–35]. Using the concept of a geometric controller for the multirotor, [36] proposed a backstepping controller based on $SE(3)$ for stable trajectory tracking and showed satisfactory experimental results.

Besides linear and nonlinear control laws, learning strategies for the multirotor are proposed in [27, 37–39]. An online learning algorithm was implemented for acrobatic motions such as flip, perching as shown in [38, 39]. In [27, 37], they used the approximation ability of the neural networks as same with [40] for learning the dynamics of the multirotor in combination with PD or backstepping control. However, in the end, the baseline controllers are almost the same as PD, LQR or backstepping control.

In order to review these variety of control laws for multirotor, research summaries were given in [41–44]. In [41–43], they provided an introduction to modeling, estimation and control of unmanned rotorcraft systems including a multirotor. Feedback control design for a family of vertical take-off and landing (VTOL) capabilities such as multirotors, ducted-fan and tail-sitters were described in [44]. However, those papers did not provide a variety of representations for the multirotor dynamics and practical issues that arise in the trajectory tracking problem including flight test-beds.

In this study, we describe several control laws for the trajectory tracking problem of a multirotor. Unlike aforementioned research papers, we focus on discussing the principle of many control techniques for the multirotor. The contribution of this paper can be summarized as follows: First, we describe various representations of the multirotor dynamics. Second, from the trajectory tracking point of view, the problem of computing desired roll and pitch angles is considered, which are virtual signals derived to achieve the desired position. Then, we review the control laws in two parts: 1) linear controllers, and 2) nonlinear controllers. PID, LQR and H_∞ control laws are dealt with in the linear control section. Feedback linearization, backstepping, geometric control, dynamic inversion and backstepping on $SE(3)$ are shown in the nonlinear control section. In addition, we present various types of flight test-beds to validate the controller. Through the discussions on the dynamics, control and experimental test-beds, this paper aims to promote understanding on the process of multirotor control design and experiments.

This paper is structured as follows: in Section 2, we describe a variety of multirotor dynamic models. Linear and nonlinear controllers are explained in Section 3 and Section 4, respectively. Section 5 describes hardware experiments, including flight test-beds. Concluding remarks are in Section 6.

Table 1. Application of a multirotor and controller type they used

Application	Controller	Literature
Moving an object	PID	[45]
	Geometric control	[35]
	Adaptive control	[46, 47]
Aerial Manipulation	PID	[48]
	Backstepping with sliding mode	[49, 50]
Entertainment	PID	[2, 51]
	PD	[38, 39, 52]
Surveillance	PID	[53, 54]
	LQR	[6]
	Geometric control	[30, 31, 55]
Searching and Mapping	PID	[56, 57]
	Inversion	[58–60]

2. MODELLING OF MULTIROTORS

In this section, we present the equations of motion for a multirotor. As shown in Fig. 2, the equations of motion with mass m and inertia tensor J are written as

$$m\ddot{\mathbf{x}} = -mge_3 + fRe_3, \tag{1}$$
$$\dot{R} = R\hat{\Omega},$$
$$J\dot{\Omega} = -\Omega \times J\Omega + \tau, \tag{2}$$

where f and $\tau = [\tau_x, \tau_y, \tau_z]^T$ are applied force and torque input, $\mathbf{x}$=$[x,\ y,\ z]^T$ is the multirotor position in the inertial frame, Ω=$[p_B,\ q_B,\ r_B]^T$ is the angular velocity in the body frame, g is gravitational force, $e_3 = [0;0;1]^T$ and R is the coordinate transformation matrix from the body frame to the inertial frame. $^\wedge : \mathbb{R}^3 \to so(3)$ is the hat map which transforms a vector in $\mathbb{R}^3$ to a 3×3 skew-symmetric matrix. In general, we will use bold letters (e.g., $\mathbf{x}$) to indicate vector quantities.

The i-th rotor of the multirotor has angular velocity ω_i and produces force F_i :

$$F_i = k_f \omega_i^2, \tag{3}$$

where $i = 1,\ldots,4$ denotes the number of rotors and k_f is thrust coefficient. The roll, pitch and yaw angles are adjusted by increasing or decreasing the angular velocity of motors. The motor control command aed on the configuration of the multirotor as shown in Fig. 2 can be obtained as

$$\begin{bmatrix} f \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & d & 0 & -d \\ -d & 0 & d & 0 \\ c_m & -c_m & c_m & -c_m \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix}, \tag{4}$$

where d is the arm length, i.e. the distance from the axis of rotation of the rotors to the center of the quadrotor, and c_m is k_m/k_f and k_m is drag coefficient.

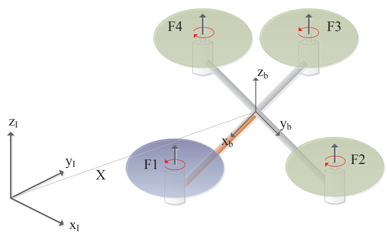


Fig. 2. Common configuration of a quadrotor (The rotor 1 and 3 rotate clockwise and the rotors 2 and 4 rotate counter clockwise).

The aerodynamic effects for a multirotor (discussed in [4,5,61]), including blade flapping, inertia of rotor or drag force, etc., are neglected for retaining our focus on the derivation of control techniques.

2.1. Translational dynamics

The translational dynamics of the multirotor are shown in (1) and also can be represented as

$$\ddot{x} = \frac{f}{m}u_x,\ \ddot{y} = \frac{f}{m}u_y,\ \ddot{z} = \frac{f}{m}(\mathrm{c}_\theta \mathrm{c}_\phi) - g, \tag{5}$$

where $u_x = (\mathrm{c}_\psi \mathrm{s}_\theta \mathrm{c}_\phi + \mathrm{s}_\psi \mathrm{s}_\phi)$, $u_y = (\mathrm{s}_\psi \mathrm{s}_\theta \mathrm{c}_\phi - \mathrm{c}_\psi \mathrm{s}_\phi)$, $s*$ is $\sin(*)$ and $c*$ is $\cos(*)$.

If yaw (ψ) are small enough, i.e., $|\psi| \ll 1$, we can simplify the translational dynamics using small angle approximation as,

$$\ddot{x} = \frac{f}{m}\mathrm{s}_\theta \mathrm{c}_\phi,\ \ddot{y} = -\frac{f}{m}\mathrm{s}_\phi,\ \ddot{z} = \frac{f}{m}\mathrm{c}_\theta \mathrm{c}_\phi - g. \tag{6}$$

If roll (ϕ) and pitch (θ) are also small, we can obtain the linearized form, $\dot{x} = Ax + Bu$, (6) as

$$\ddot{x} = g\theta,\ \ddot{y} = -g\phi,\ \ddot{z} = \frac{1}{m}\delta f, \tag{7}$$

where δf=f-mg [7].

2.2. Attitude dynamics

The attitude dynamics shown in (2) have the states consisting of the Euler angle in the inertial frame and the body angular velocity in the body frame. Equation (2) can be replaced with dynamics which have all the states in the inertial frame. This dynamics model can be obtained by the Lagrangian method and has been shown in [10,27,62]. The detailed proof is in [63].

Recalling that the Lagrangian is the difference between kinetic energy and potential energy i.e. $L = T - V$, the equation of motion is expressed as

$$\Gamma_i = \frac{d}{dt}\left(\frac{\delta L}{\delta \dot{\mathbf{q}}_i}\right) - \frac{\delta L}{\delta \mathbf{q}_i}, \tag{8}$$

where $\Gamma_i = [fRe_3{}^T\,;\,\tau_\eta{}^T]$ is the generalized force and $\mathbf{q}_i = [x\ y\ z\ \phi\ \ \theta\ \ \psi]^T$ are generalized coordinates.

We define the rotational inertia tensor J_η as

$$J_\eta = T_\eta{}^T J T_\eta, \tag{9}$$

where T_η is the jacobian matrix to convert Ω to $\dot{\eta}$. The rotational kinetic energy is expressed as

$$E_{Rot} = \frac{1}{2}\Omega^T J \Omega = \frac{1}{2}\dot{\eta}^T J_\eta \dot{\eta}, \tag{10}$$

where η=$[\phi,\ \theta,\ \psi]^T$. The Lagrange rotational equations in terms of η can be written as

$$J_\eta \ddot{\eta} + \frac{d}{dt}\{J_\eta\}\dot{\eta} - \frac{1}{2}\left(\frac{\delta}{\delta \eta}\dot{\eta}^T J_\eta \dot{\eta}\right) = \tau, \tag{11}$$

or simply represented as

$$J_\eta \ddot{\eta} + C(\eta, \dot{\eta})\dot{\eta} = \tau. \tag{12}$$

The reader can find more detail on J_η and $C(\eta, \dot{\eta})$ in [10].

If the multirotor is near hover, or roll(ϕ) and pitch (θ) are small, we can represent the multirotor dynamics simpler. These simplified version of dynamics can help derive a controller more easily. The simplified dynamics are represented as

$$\ddot{\phi} \approx I_{yzx}\dot{\psi}\dot{\phi} + \frac{\tau_\phi}{I_{xx}},\ \ddot{\theta} \approx I_{zxy}\dot{\phi}\dot{\psi} + \frac{\tau_\theta}{I_{yy}},\ \ddot{\psi} \approx I_{xyz}\dot{\phi}\dot{\theta} + \frac{\tau_\psi}{I_{zz}}, \tag{13}$$

where $I_{xyz} = (I_{xx} - I_{yy})/(I_{zz})$.

If the Euler angular rates are small, further simplified equations are

$$\ddot{\phi} \approx \frac{\tau_\phi}{I_{xx}},\ \ddot{\theta} \approx \frac{\tau_\theta}{I_{yy}},\ \ddot{\psi} \approx \frac{\tau_\psi}{I_{zz}}. \tag{14}$$

3. LINEAR CONTROLLER DESIGN

In this section, we address how to design the controller for the multirotors. The controller can be divided into two categories based on the modelling method: linear and nonlinear controllers. The common trajectory control structure is shown in Fig. 3. Using the force input f and torque input M, the desired motor speeds are calculated based on motor matrix for the multirotor.

We present the first approach, linear controllers for the multirotor such as PID, PID for aggressive maneuvers, LQR and robust control. These types of controllers are known as intuitive and easy to implement.

3.1. Classic PID control

A PID controller is one of the most well-known control laws and it is still widely used for many multirotor projects. Linearized multirotor dynamics and PID controllers appeared in [3–5]. Complicated linearized multirotor dynamics considering rotor flapping effects appeared in [5].

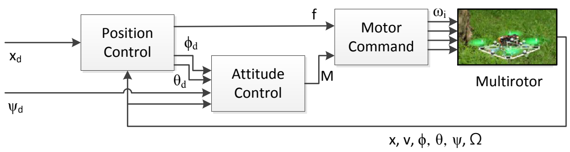


Fig. 3. Common structure of trajectory tracking control for a multirotor.

Designing a PID controller is based on the transfer function between the control signal and Euler angles. To obtain the simple transfer function, we linearize the multirotor dynamics. Assuming that the multirotor is near the hover states, or at small roll and pitch angles, the attitude dynamics appear as in (14).

For example, the resulting attitude transfer function for roll axis appears as

$$\frac{\Phi(s)}{T_\phi(s)} = \frac{d}{s^2(\mu s+1)I_{xx}}, \tag{15}$$

where μ is the thrust time delay and T_* is the Laplace transform of the control input. Using (15), the control input τ_* is computed based on proportional, derivative and integral error signal. Control gains are chosen so that closed-loop dynamics have negative poles.

To apply this control law to trajectory tracking, the properties of an under-actuated system must be addressed. In other words, the desired roll and pitch angles should be computed from the $x-y$ position error. We deal with this problem in Section 3.2, since the approach for converting altitude and attitude control to trajectory control is the same. Another method to compute the desired roll angle(ϕ_d) and the desired pitch angle(θ_d) is shown in [3] for a way point tracking problem.

3.2. PID control for aggressive maneuvers

A PID controller for aggressive maneuvers is different from aforementioned classic PID control, because it has to be derived based on nonlinear dynamics in (2). This means that the transfer function approach in (15) cannot be employed.

Control for agile maneuvers of the multirotor was proposed in [2, 38, 51, 64]. Aggressive maneuvers using classic PID control techniques with analysis of blade flapping and thrust variation appeared in [51]. Aggressive maneuvers based on a 2D multirotor model and optimization techniques were in [38, 64] and trajectory generation and tracking control for aggressive maneuvers appeared in [2].

In this control law, for trajectory tracking, desired roll and pitch angle should be obtained, assuming that the multirotor is near the hover state. ϕ_{des} and θ_{des} can be calculated as

$$\phi_{des} = \frac{1}{g}(\ddot{x}_{com}s_{\psi_{des}} - \ddot{y}_{com}c_{\psi_{des}}),$$

$$\theta_{des} = \frac{1}{g}(\ddot{x}_{com}c_{\psi_{des}} + \ddot{y}_{com}s_{\psi_{des}}), \tag{16}$$

here $\ddot{x}_{com}$ and $\ddot{y}_{com}$ are the acceleration command for the multi-copters in the x and y direction, respectively. Desired body angular rates are computed by numerical differentiation of desired Euler angles. In order to improve the performance, the desired roll and pitch angles are modified by numerical optimization such as that shown in [2].

3.3. LQR control

There are several attempts to use LQR (Linear Quadratic Regulator) for a multirotor [6, 7], despite the limitation that this controller offers an optimal solution only when the multirotor is near the hover state or in some specific region.

Before designing the LQR controller, the state space model for the multirotors should be obtained. The simplified linearized multirotor model with states as:

$$\mathbf{x}_l = [x, \dot{x}, y, \dot{y}, z, \dot{z}, \phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}]^T \tag{17}$$

and control inputs $\mathbf{u}_l = [\delta f, \tau_\phi, \tau_\theta, \tau_\psi]^T$ are obtained by (7) and (14). It can be written in matrix form as

$$\dot{\mathbf{x}}_l = A\mathbf{x}_l + B\mathbf{u}_l, \tag{18}$$

where the matrices $A \in \mathbb{R}^{12\times12}$ and $B \in \mathbb{R}^{12\times4}$ are easily obtained by (7) and (14). The control gain K_l is designed to minimize the user-defined cost function. There is an augmented LQR formulation which considers the integration of error between reference and output [8]. It shows more robust performance than conventional LQR controller.

The LQR controller can be also applied to trajectory tracking. If time dependent desired trajectory $x_{des}(t)$ is given, the LQR controller can be applied to minimize the error between the fully measured state $\mathbf{x}$ and reference state $\mathbf{x}_{des}(t)$. The control input is designed based on the nominal control input to follow the desired trajectory $\mathbf{x}_{des}(t)$.

Note that when the multirotor is in the trajectory following mode, the stability of the closed-loop matrix $A+BK_l$ can not be guaranteed. The stability region can be found by a grid search algorithm for the condition

$$Re(\lambda(A(\phi,\theta,f) - K_{lqr}B(\phi,\theta))) < 0, f_{\min} < f < f_{\max}, \tag{19}$$

where $\lambda(\cdot)$ means eigenvalues and $Re(\cdot)$ means their real parts. The stability region for trajectory tracking appears as $\phi^2+\theta^2 < 48^\circ$ (see [6]) and the generated trajectory should also satisfy this condition to ensure stability.

3.4. Robust control

Robust control is a technique to ensure that the performance objective is satisfied for all possible operations in the uncertainty set [65]. When robust control is applied to a multirotor, usually linearization techniques are needed as same with other linear control laws. One common type of linearization techniques is feedback linearization which is covered in Section 4.1 in more detail.

Using these linearizaton techniques, many researchers proposed robust controllers for multirotors. A robust attitude tracking controller is designed in [66] based on nonlinear DOB (disturbance observer) developed in [67]. The DOB-based attitude controller ensures that the trajectory of the uncertain plant stays arbitrarily close to that of the linear nominal system. In [9], they designed a $2-$DOF linear H_∞ controller using loop shaping for stabilization, speed, throttle and yaw control based on a multirotor model at the hover equilibrium point. Robust control with a linearized model is proposed in [11–14]. In [11], they presented a robust feedback linearization controller and generalized H_∞ controller as an outer-loop controller. This control structure is designed to make the system internally asymptotically stable and its output to follow a desired trajectory in the presence of external disturbances. In [13, 14], they used multiple linearized models at operation points of Euler angle rate and used LMIs (Linear Matrix Inequalities) to design separate controllers subject to stability and pole placement constraints. Feedback linearization as an inner-loop controller makes linear multirotor model around an operating point whereas robust control stabilizes the multirotor based on predefined operation points. An approximate feedback linearization controller using first order derivatives was presented in [12] and it showed satisfactory experimental results. Controller gains are synthesized giving suboptimal H_2 and H_∞ robust performances guarantees.

One of benefits of the robust controller with LMIs is that the controller is not limited to the near hover condition, unlike other linear controllers such as PID or LQR. The controller with LMIs can be designed using linearization technique and linear models is obtained at each operation point of multirotors. In this case, an LMI method can be used to synthesize gains with guaranteed performance over the specified operating envelope for the multirotors. The detailed procedure is in [12], to which we direct the reader for details. In addition, robust control with DOB can provide satisfactory tracking performance in noisy environments and time delay caused by actuator dynamics [66]. In windy sites such as sea coast, robust control can be an appropriate solution for stable flight.

4. NONLINEAR CONTROLLER DESIGN

We present nonlinear controllers for multirotors such as feedback linearization, backstepping, geometric control and blended controllers including dynamic inversion and backstepping on $SE(3)$.

4.1. Feedback linearization control

Feedback linearization (FL) approaches such as input-output linearization and state-space linearization [19] were proposed for multirotors in [20, 21]. In [20], feedback linearization was compared with an adaptive sliding mode controller. A feedback linearization controller combined with Luenberger observer was presented in [21]. In general, these controllers involve high-order derivative terms which make the controller less reliable against sensor noise.

Although the noises and the disturbances including model uncertainty can cause performance degradation of the controller because of high-order derivatives, feedback linearization is a good method to obtain a linearized model of multirotors. A linear relationship between control input and system state can be obtained by differentiating (5) and ψ equation in (14) until the control input terms appear. However, in this case, FL controller is designed involving complex computation and high-order derivatives. To make computation easy and reduce the occurrences of high-order derivatives, simplified dynamics as shown in (6) and (14) can be used [20]. Assuming that $|\psi| \ll 1$, the dynamics can be written as

$$\begin{aligned} &\ddot{x} = u_1 s_\theta c_\phi, \quad \ddot{y} = -u_1 s_\phi, \quad \ddot{z} = u_1, c_\theta c_\phi - g, \\ &\ddot{\phi} = u_2 d, \quad \ddot{\theta} = u_3 d, \quad \ddot{\psi} = u_4, \end{aligned} \tag{20}$$

where $u_1 = f/m$, $u_2 = \tau_\phi/(I_{xx}d)$, $u_3 = \tau_\theta/(I_{yy}d)$ and $u_4 = \tau_\psi/(I_{zz})$.

To derive the input of the position controller, (20) is differentiated until the input terms appear. The control input terms appear in the 4-th order derivatives, $x^{(4)}$, $y^{(4)}$ and $z^{(4)}$. Attitude controller in roll and pitch angles can be designed the same as the position controller. In addition, the PD controller can handle the linear relationship between control input and yaw, which is expressed as

$$u_4 = \ddot{\psi}_d - k_{\psi 1}(\dot{\psi} - \dot{\psi}_d) - k_\psi(\psi - \psi_d). \tag{21}$$

4.2. Backstepping control with sliding mode technique

A backstepping controller is a one of powerful nonlinear controller. It has different features from feedback linearization controllers. First, feedback linearization controllers need high order derivative terms while a backstepping controller does not. Second is about global stability. The nonlinearity of the system cannot be completely cancelled when modelling error exists in feedback linearization. Thus, global stability cannot always be guaranteed using feedback linearization [22]. However, the backstepping approach does not suffer from these problems when the controller is designed properly.

A backstepping controller is designed to stabilize the whole system using Lyapunov stability theory in the recursive structure. For controlling the multirotors, a backstepping controller with sliding mode techniques is used based on nonlinear translational equation (1) and simplified attitude equation (13) in [24–26]. The backstepping controller can also be designed based on Lagrangian formulation (12) in [23, 27].

The difference between (13) and (12) is the assumption that ϕ and θ are small. Due to this difference, the backstepping controller is more suitable than the conventional backstepping control [24–26] when roll and pitch angle are high, while the complexity of controller increases.

In this control law, to handle under-actuation properties, ϕ_d and θ_d is computed from $u_x^d(= c_{\psi_d}s_{\theta_d}c_{\phi_d} + s_{\psi_d}s_{\phi_d})$ and $u_y^d(= s_{\psi_d}s_{\theta}c_{\phi_d} - c_{\psi_d}s_{\phi_d})$, which can be written as:

$$u_x^d = \frac{m}{f}(\ddot{x}_{com}), u_y^d = \frac{m}{f}(\ddot{y}_{com}), \tag{22}$$

where $\ddot{x}_{com}$ and $\ddot{y}_{com}$ are the acceleration command which is computed by backstepping controller. Using (22), the desired roll (ϕ_d) and pitch (θ_d) angles can be extracted by computing u_x and u_y. If the multirotor follows the linear motion, a simpler way exists for obtaining the desired roll (ϕ_d) and pitch (θ_d) [25]. Another method is extracting ϕ_d and θ_d from the desired rotation matrix R_d. This will be described in detail in Section 4.3.

Recently, the backstepping control with sliding-mode technique is applied to aerial manipulation with multi-DOF robotic arm also [49, 68–70]. It is mainly because an additional torque due to the robotic arm can be compensated by the controller using Lypunov stability analysis in recursive structure.

4.3. Geometric control

A geometric controller for the multirotor is designed to satisfy exponential convergence in $SO(3)$ and uses a measure of the error in rotations, given the desired attitude R_d [42]. The controller offers not only a solution for avoiding singularities in trajectory tracking but also simplicity of the controller form, whereas the quaternion-based controller in [71] provides a solution for the singularity problem only. The simplicity of the controller and the absence of singularity can be a benefit in real experiments. There were many papers which showed experimental results using the geometric controller [30, 31, 35, 36, 55, 72]. A robust attitude controller for the multirotor using rotation matrix was proposed in [32]. They considered an attitude controller with an anti-windup solution and semi-global asymptotic stability is proven.

The position controller in geometric control is designed by using position and velocity errors. The position controller form is almost same with backstepping approach. However, attitude controller is different with backstepping controller.

In advance to the attitude controller, the desired rotation matrix R_d should be defined. Since the direction of the third body-fixed axis, b_{3d}, is defined by the desired force vector, the normalized $b_{3d} = F_d/\| F_d \|$ is obtained, where the denominator is assumed to be nonzero.

The input for trajectory tracking as shown in Fig. 3 contains the desired yaw angle (ψ_d), so $b_{1d} = [\cos(\psi_d), \sin(\psi_d)\ , 0]^T$ is defined. The direction of the second body-fixed axis is $b_{2d} = b_{3d} \times b_{1d}/\|b_{3d} \times b_{1d}\|$. From b_{2d} and b_{3d}, b_{1d} is redefined as $b_{2d} \times b_{3d}$ to satisfy orthogonality with b_{3d} and b_{2d}. The desired rotation matrix R_d is calculated as $R_d = \begin{bmatrix} b_{2d} \times b_{3d} & b_{2d} & b_{3d} \end{bmatrix}$.

Now, attitude control is considered. The attitude error is defined as

$$e_R = \frac{1}{2}(\tilde{R} - \tilde{R}^T)^\vee,\ e_\Omega = \Omega - \tilde{R}^T\Omega_d, \tag{23}$$

where $\tilde{R} = {R_d}^T R$ and $^\vee : so(3) \to \mathbb{R}^3$ is the vee map which transforms a 3×3 skew-symmetric matrix to a vector in $\mathbb{R}^3$. In (23), $\tilde{R}^T$ pre-multiplies Ω_d to make $\dot{R} = R\hat{\Omega}$ and $\dot{R}_d = R_d\hat{\Omega}_d$ on the same tangential vector $T_R SO(3)$, i.e. $T_{R_d}SO(3) \to T_R SO(3)$. In addition, Ω_d in (23) is obtained from the rotation dynamics in (2). Using these errors (23), the attitude control input is defined such as PD feedback controller. Under the proper control law, e_R and e_Ω can converge exponentially to zero (See [29, 32]). Recently, due to the simplicity and exponential stability of the geometric controller, researchers develop the controller for carrying a cable-suspended load [35, 73].

4.4. Blended controller form I : Dynamic inversion with FL and backstepping

In the previous section, nonlinear control techniques, such as feedback linearization, backstepping, geometric control, for multirotor trajectory tracking are considered. Each controller has its own advantages, but has disadvantages as well, such as the existence of high-order terms or the singularity problem in the attitude. In order to overcome the shortcomings by combining the advantages of each controller, blended controller forms are proposed in many papers [12, 28, 36, 60]. The control law is designed as a mixture of two or more types of controllers to achieve some specific effects, e.g., for making the controller simple or for the robustness, etc.

The dynamic inversion with feedback linearization and backstepping approach for a multirotor was designed in [28]. The controller consists of two parts: 1) a dynamic inversion based on feedback linearization as the inner-loop, and 2) an internal dynamics stabilizer based on backstepping as the outer-loop. The internal stabilizer loop is derived to obtain ϕ_d and θ_d. This method for obtaining ϕ_d and θ_d is similar to (22) based on the backstepping approach. The detailed process is presented in [28].

4.5. Blended controller form II : Backstepping control based on $SE(3)$

A backstepping controller based on $SE(3)$ is designed in [36] to overcome the singularity problem in backstepping control and give robustness to geometric control. Therein, the position controller is used to track the desired Cartesian coordinates using position and velocity errors. The attitude controller uses the rotation matrix error and body angular velocity error to stabilize attitude dynamics expressed on $SO(3)$ similar to geometric control. The controller including the integral term makes it more robust than geometric control.

Position control is designed as same as the backstepping control. The attitude tracking error, $\mathbf{e}_R$, and angular velocity tracking error, e_Ω, are defined the same as Section 4.3. The geometric controller uses the same error definition, but uses the backstepping control approach for improved robustness. We define the sliding mode error $\mathbf{r}_a$ as

$$\mathbf{r}_a = \dot{e}_R + \Lambda_2 e_R, \tag{24}$$

where Λ_2 is a gain matrix. The control input can be obtained to make the derivative of Lypapunov candidate function negative semi-definite. The stability proof and experimental results were presented in our previous study [36].

5. HARDWARE EXPERIMENTS

In previous section, we designed the various controllers for the multirotors. These controllers can be validated in the flight test-beds. In this section, we address the various multirotors platform and flight setups. We also show the flight experiment using the PID and backstepping controllers and the flight setups.

5.1. Multirotor platforms

Among many multirotors, we mention several types of multirotors frequently utilized in research. More details about sensors and filters used for multirotors can be found in [74].

Crazyflie 2.0 developed by Bitcraze is one of the smallest multirotors among commercially available platforms. It only weighs 27 grams and 10 grams payload. Using the small size of the platforms, it is easy to operate in tightly constrained environments and closer swarm formations [75].

Arducopter and Mikrokopter are well-known multirotors for hobby. Their source codes are publicly available because they are based on OSPs (Open-Source Projects). Since it is difficult to employ user-defined codes on onboard processor in those multirotors, they are equipped with an extra computer to handle higher computation loads as shown in [12, 68, 76]. In addition, one of the famous drones for hobby, AR drone, is also utilized in many studies [77–79]. AR drone is applicable for visual servoing or navigation because it has two cameras at front and down sides.

Research platforms provided by Ascending technologies (also known as Asctec) or DJI are also used in many researches. Although research platforms of Asctec is more expensive than AR drone or commercial OSP multirotors, users can take advantage of many electronics and accessories such as vison camera or on-board computer. Quadrotor-type UAV (Unmanned Aerial Vehicle), Hummingbird is a light-weight platform and usually used for algorithm verification as shown in [2, 38, 39, 52, 72]. Larger platforms such as Pelican or Firefly are often used for SLAM (Simultaneous Localization And Mapping) [58, 60] or aerial manipulation [69, 70]. DJI recently released the research platform for multirotor, Matrice 100, which have integrated visual cameras and ultrasonic sensors [80]. This platform is commercially available and source codes are open, same as the Asctec multirotors.

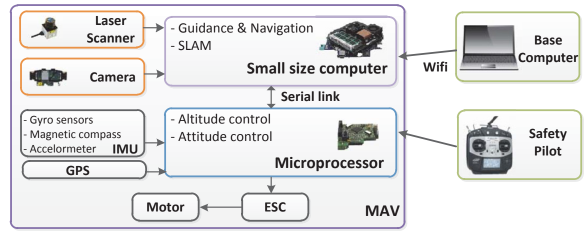


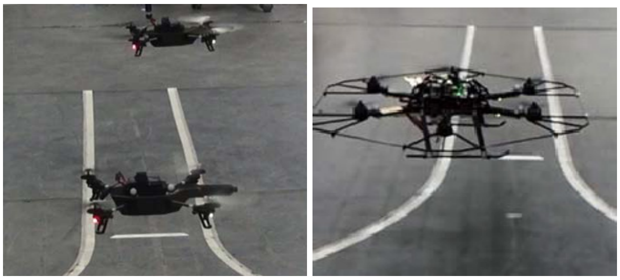
Fig. 4. Common structure for SLAM using autonmous multirotors.

5.2. Experimental setup

In this subsection, we address various types of the experimental setup for flying multirotors. In this paper, we review the experimental setups in two parts: 1) Position estimation by On-board Camera including SLAM, 2) Position updates by external sources such as Vicon [81] or GPS (Global Positioning System).

The first approach is to obtain the position data by vision information. For flying in unknown, GPS-denied environments, multirotors are required to solve SLAM problem as well as precise state estimation using laser range scanners [82] or vision cameras [58, 60], etc. With SLAM and state estimation algorithm, multirotors can fly even in an earthquake-damaged building [58]. However, for solving SLAM, a high performance computer is necessary. The common structure for autonomous multirotors for SLAM is shown in Fig. 4. The external computer computes pose estimation using SLAM and then transmits the estimated pose date to the microprocessor. The microprocessor computes the control input and transmit the motor control input to ESC (Electronic Speed Control).

Another configuration for flying multirotors is based on external positioning system such as Vicon or GPS. Here, since the GPS-based flight system is almost same with Fig. 4 except camera, we focus on the configuration using Vicon system. There are three ways to establish the test-beds in Vicon environments: 1) PCTx, 2) using open source in



(a) AR drones. (b) Firefly.

Fig. 5. Multirotors used in the experiments.

ROS and 3) custom-made communication by Zigbee or Wi-Fi (Wireless Fidelity).

The first approach is using PCTx which sits in between your PC and transmitter. It allows a PC side application to control a radio control vehicle or robot [83]. Users can design the controller on the ground computer and send a command to the R/C (Radio controlled) transmitter using PCTx. The transmitter gives R/C signal to the multirotor, and then on-board attitude controller runs on the multirotor based on received R/C signal. The flight configuration and experiments can be found in [7, 8, 36, 54]. Although this configuration can be applied to any commercial multirotors such as Arducopter or Mikrokopter, frequency of transmission is limited to frequency of the transmitter (almost 50Hz).

The second approach is using ROS (Robot Operating System). It consists of a collection of tools, libraries, and conventions for a variety of robotic platforms [84] and is open for everyone. AR drone is also supported by ROS driver [85]. User can send a control command including desired translational and angular velocities by Wi-Fi communication. By using ROS, users can run test-bed for flight of multiple multirotors. Multirotors made by Asctec are also supported by ROS driver [86]. In this platform, a small size of computer is recommended to run ROS for transmitting and receiving the enough size of data.

Third approach is using custom-made Zigbee or Wi-Fi communication. Using custom-made communication, users can make their own flight setups as shown in [12,68]. In [12], they made custom-made quadrotor with commercial cell phone and Wi-Fi communication. In [68], they made the flight setup using Mikrocopter and Odroid computer [87]. The data and vision information are transmitted and received by Wi-Fi communication on Odroid computer. Although Zigbee has slower transmission rate than Wi-Fi, serial communication by Zigbee can be a good solution for establishing the custom-made communication system. The range of the Zigbee communication is over 1 mile, while Wi-Fi is less than a few hundred meters even in outdoor environments. In our previous study [70], we used the Zigbee communication system developed by Asctec and showed satisfactory flight performance.

5.3. Flight test

In this section, we present two flight experiments to illustrate the difference between the linear controller (i.e., the classic PID controller) and the nonlinear controller (i.e., the backstepping controller with sliding mode technique) based on the aforementioned flight setups. For qualitative evaluation, we used the Vicon system. There are two types of multirotors used in the experiments as shown in Fig. 5, i.e., AR drone with ROS and Firefly with Zigbee communication. The flight experiment using PCTx can be found in our previous study [36].

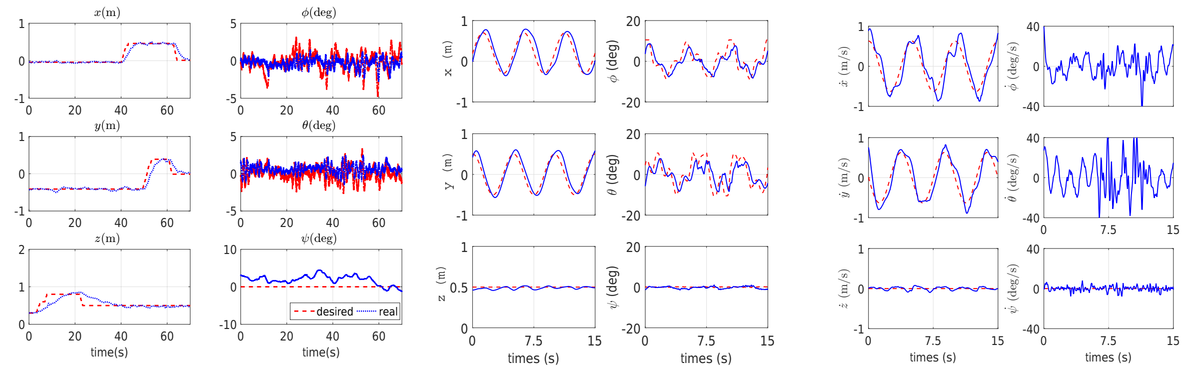
First, we performed the experiment with AR drone and ROS. We integrated the classic PID controller in Section 3.1. The position controller runs at 20Hz and the attitude controller runs at 200Hz. Fig. 6(a) shows the flight performance at way-point tracking experiment and Figs. 6(b) and 6(c) shows the performance at fast trajectory tracking experiments.

Second experiment is performed with Firefly and Zigbee communication. We integrated the backstepping controller with the sliding mode technique in Section 4.2. The control inputs are transmitted and on-board data is received at 20Hz, simultaneously. Fig. 6(d) shows the flight performance at way-point tracking and Figs. 6(e) and 6(f) shows the performance at fast trajectory tracking. The desired trajectory at Fig. 6(b) and Fig. 6(e) is set to be same.

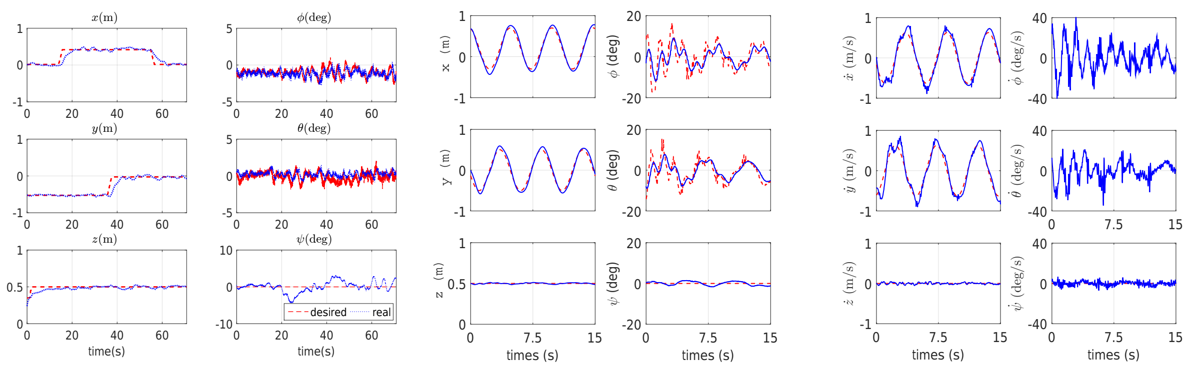
In way-point tracking experiment, both controllers show satisfactory tracking performances. However, in the fast trajectory tracking experiment shown in Fig. 6, the classic PID controller with AR drone shows worse performance than the backstepping controller with Firefly. This is mainly because the classic PID controller is designed based on the linearized dynamics and suitable for low-speed tracking, not fast tracking. In addition, blade flapping also affects the flight performance. Since AR drone has more flexible propellers, nonlinearity becomes significant enough that the classic PID controller cannot handle the nonlinearity, effectively.

6. CONCLUSION

In this paper, we have reviewed control laws for trajectory tracking of multirotors. Various forms of multirotor dynamics were explained. Linear controllers such as PID, LQR and H_∞ control law for a multirotor were presented. PID control is the most well-known control law and easy to implement, but it needs modifications for the aggressive maneuvers. LQR controller can provide optimality but it only holds near the hover states. H_∞ control is more robust than other linear controllers, but it is still derived based on



(a) Waypoint tracking of AR drone. (b) Fast position tracking of AR drone. (c) Fast velocity tracking of AR drone.



(d) Waypoint tracking of Firefly. (e) Fast position tracking of Firefly. (f) Fast velocity tracking of Firefly.

Fig. 6. Performance comparison between lineaer PID controller with AR drone (a)-(c) and nonlinear backstepping controller with Firefly (d)-(f).

linearized dynamics or some operation points, which has limited its applications to non-agile maneuvers.

Nonlinear control laws for the multirotor such as feedback linearization, backstepping, geometric control, dynamic inversion and backstepping on $SE(3)$ were explained. Feedback linearization is easy to implement, but it will be a problem when model error or sensor noise exists because of high-order derivative terms. Dynamic inversion control with feedback linearization and backstepping control can make the controller only require the second derivative of the desired output and first derivative of the error. Backstepping control can give us global stability while feedback linearization does not, but control input terms are complicated in comparison with other linear control laws. Geometric control can avoid singularities and is also very intuitive, but the specific initial condition should be satisfied and it lacks robustness. Backstepping control on $SE(3)$ can give a simple integral solution to the geometric controller.

In hardware experiments, we summarized various types of multirotors frequently utilized in research. The flight setups for validating the controller were explained. The experimental results using the flight test-beds show the performance difference between the linear and nonlinear controllers.

REFERENCES

[1] J. Leishman, “The breguet-richet quad-rotor helicopter of 1907,” *Vertiflite*, vol. 47, no. 3, pp. 58-60, 2002.

[2] D. Mellinger, N. Michael, and V. Kumar, “Trajectory generation and control for precise aggressive maneuvers with quadrotors,” *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664-674, 2012.

[3] G. Hoffmann, S. Waslander, and C. Tomlin, “Quadrotor helicopter trajectory tracking control,” in *Proc. of AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, 2008, pp. 1-14.

[4] S. Bouabdallah, A. Noth, and R. Siegwart, “PID vs LQ control techniques applied to an indoor micro quadrotor,” *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 2451-2456, Sept.-2 Oct. 2004.

[5] P. Pounds, R. Mahony, and P. Corke, “Modelling and control of a quad-rotor robot,” *Proceedings Australasian Conference on Robotics and Automation*, 2006.

[6] I. Cowling, O. Yakimenko, J. Whidborne, and A. Cooke, "Direct method based control system for an autonomous quadrotor," *Journal of Intelligent & Robotic Systems*, vol. 60, no. 2, pp. 285-316, 2010.

[7] M. Valenti, B. Bethke, G. Fiore, J. How, and E. Feron, "Indoor multi-vehicle flight testbed for fault detection, isolation, and recovery," *Proc. of the AIAA Guidance, Navigation, and Control Conference*, vol. 63, 2006.

[8] Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky, "Adaptive control of quadrotor uavs: A design trade study with flight evaluations," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1400-1406, 2013.

[9] M. Chen and M. Huzmezan, "A Combined MBPC/2 DOF H_∞ Controller for a Quad Rotor UAV," *Proc. of AIAA Guidance, Navigation, and Control Conference and Exhibit, AAS Astrodynamics Specialist Conference*, 2003.

[10] G. Raffo, M. Ortega, and F. Rubio, "An integral predictive/nonlinear control structure for a quadrotor helicopter," *Automatica*, vol. 46, no. 1, pp. 29-39, 2010.

[11] A. Mokhtari, A. Benallegue, and B. Daachi, "Robust feedback linearization and GH_∞ controller for a quadrotor unmanned aerial vehicle," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1198-1203, Aug. 2005.

[12] T. Ryan and H. J. Kim, "Approximate feedback linearization and LMI based gain synthesis for robust VTOL UAV control," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 4, pp. 1173-1178, 2013.

[13] F. Yacef, O. Bouhali, H. Khebbache, and F. Boudjema, "Takagi-sugeno model for quadrotor modelling and control using nonlinear state feedback controller," *International Journal of Control Theory and Computer Modelling*, vol. 2, no. 3, pp. 9-24, 2012.

[14] H. Lee and H. J. Kim, "Robust control of a quadrotor using takagi-sugeno fuzzy model and an lmi approach," *Proc. of International Conference on Control, Automation and Systems*, pp. 370-374, 2014.

[15] M. Sanchez, J. Acosta, and A. Ollero, "Integral action in first-order closed-loop inverse kinematics. application to aerial manipulators," *Proc. of IEEE International Conference on Robotics and Automation*, 2015., pp. 5297-5302

[16] "Ar drone by parrot," http://ardrone2.parrot.com/.

[17] "Indago VTOL quadrotor by lockheed martin," http://www.lockheedmartin.com/.

[18] D. Scaramuzza *et al.*, "Vision-controlled micro flying robots: From system design to autonomous navigation and mapping in gps-denied environments," *IEEE Robotics Automation Magazine*, vol. 21, no. 3, pp. 26-40, Sept 2014.

[19] M. Henson and D. Seborg, "Feedback linearizing control," *Nonlinear Process Control*, pp. 149-231, 1997

[20] D. Lee., H. J. Kim, and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *International Journal of Control, Automation and Systems*, vol. 7, no. 3, pp. 419-428, 2009.

[21] A. Mokhtari, N. M'Sirdi, K. Meghriche, and A. Belaidi, "Feedback linearization and linear observer for a quadrotor unmanned aerial vehicle," *Advanced Robotics*, vol. 20, no. 1, pp. 71-91, 2006.

[22] O. Härkegård and S. Glad, *Flight Control Design Using Backstepping*, Linköping University Electronic Press, 2001.

[23] T. Madani and A. Benallegue, "Control of a quadrotor mini-helicopter via full state backstepping technique," *Proc. of IEEE Conference on Decision and Control*, pp. 1515-1520, 2006.

[24] P. Castillo, P. Albertos, P. Garcia, and R. Lozano, "Simple real-time attitude stabilization of a quad-rotor aircraft with bounded signals," *Proc. of IEEE Conference on Decision and Control*, pp. 1533-1538, 2006.

[25] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," *Proc. of IEEE International Conference on Robotics and Automation*, pp. 2247-2252, April 2005.

[26] P. Adigbli, C. Gr, J. Mouret, and S. Doncieux, "Nonlinear attitude and position control of a micro quadrotor using sliding mode and backstepping techniques," *European Micro Air Vehicle Conference*, pp. 1-9, 2007.

[27] A. Das, F. Lewis, and K. Subbarao, "Backstepping approach for controlling a quadrotor using lagrange form dynamics," *Journal of Intelligent & Robotic Systems*, vol. 56, no. 1, pp. 127-151, 2009.

[28] A. Das, K. Subbarao, and F. Lewis, "Dynamic inversion with zero-dynamics stabilisation for quadrotor control," *IET Control Theory & Applications*, vol. 3, no. 3, pp. 303-314, 2009.

[29] T. Lee, M. Leoky, and N. McClamroch, "Geometric tracking control of a quadrotor uav on SE(3)," *Proc. of IEEE Conference on Decision and Control*, pp. 5420-5425, Dec. 2010.

[30] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," *Proc. of IEEE International Conference on Robotics and Automation*, pp. 2520-2525, May 2011.

[31] A. Kushleyev, D. Mellinger, and V. Kumar, "Towards a swarm of agile micro quadrotors," in *Robotics: Science and Systems*, 2012.

[32] K. Rudin, M. Hua, G. Ducard, and S. Bouabdallah, "A robust attitude controller and its application to quadrotor helicopters," *Proc. of 18th IFAC World Congress*, pp. 10379-10384, 2011.

[33] T. Fernando, J. Chandiramani, T. Lee, and H. Gutierrez, "Robust adaptive geometric tracking controls on SO(3) with an application to the attitude dynamics of a quadrotor uav," *Proc. of IEEE Conference on Decision and Control and European Control Conference*, pp. 7380-7385, 2011.

[34] T. Lee, M. Leok, and N. McClamroch, "Nonlinear robust tracking control of a quadrotor UAV on SE(3)," *Asian Journal of Control*, vol. 15, no. 3, pp. 391-498, 2013.

[35] F. Goodarzi, D. Lee, and T. Lee, "Geometric control of a quadrotor UAV transporting a payload connected via flexible cable," *International Journal of Control, Automation, and Systems*, vol. 13, no. 6, pp. 1486-1498, 2016.

[36] H. Lee, S. Kim, T. Ryan, and H. J. Kim, "Backstepping control based on SE(3) of a micro quadrotor for stable trajectory tracking," *Proc. of IEEE International Conference on Systems, Man, and Cybernetics*, pp. 4522-4527, 2013.

[37] T. Dierks and S. Jagannathan, "Output feedback control of a quadrotor UAV using neural networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 50-66, 2010.

[38] S. Lupashin, A. Schoellig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," in *IEEE International Conference on Robotics and Automation*, pp. 1642-1648, 2010.

[39] T. Tomic, M. Maier and S. Haddadin, "Learning quadrotor maneuvers from optimal control and generalizing in real-time," *Proc. of IEEE International Conference on Robotics and Automation*, pp. 1747-1754, 2014.

[40] X. Zhao, P. Shi, X. Zheng, and J. Zhang, "Intelligent tracking control for a class of uncertain high-order nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 9, pp. 1976-1982, 2016.

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

[42] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 20-32, sept. 2012.

[43] F. Kendoul, "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," *Journal of Field Robotics*, vol. 29, no. 2, pp. 315-378, 2012.

[44] M.-D. Hua, T. Hamel, P. Morin, and C. Samson, "Feedback control of underactuated VTOL vehicles," *IEEE Control Systems*, vol. 33, no. 1, pp. 61-75, 2013.

[45] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative grasping and transport using multiple quadrotors," *Distributed Autonomous Robotic Systems*, pp. 545-558, 2010.

[46] I. Palunko, P. Cruz, and R. Fierro, "Agile load transportation: Safe and efficient load manipulation with aerial robots," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 69-79, 2012.

[47] G. Antonelli, E. Cataldi, P. Giordano, S. Chiaverini, and A. Franchi, "Experimental validation of a new adaptive control scheme for quadrotors MAVs," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2439-2444, 2013.

[48] M. Orsag, C. Korpela, and P. Oh, "Modeling and control of MM-UAV: Mobile manipulating unmanned aerial vehicle," *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1, pp. 227-240, 2013.

[49] A. Jimenez-Cano, J. Martin, G. Heredia, A. Ollero, and R. Cano, "Control of an aerial robot with multi-link arm for assembly tasks," *Proc. of IEEE International Conference on Robotics and Automation*, pp. 4916-4921, 2013.

[50] S. Kim, S. Choi, and H. J. Kim, "Aerial manipulation using a quadrotor with a two dof robotic arm," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4990-4995, 2013.

[51] H. Huang, G. Hoffmann, S. Waslander, and C. Tomlin, "Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering," *Proc. of IEEE International Conference on Robotics and Automation*, pp. 3277-3282, 2009.

[52] A. Schöllig and R. D'Andrea, "Optimization-based iterative learning control for trajectory tracking," *Proceedings of the European Control Conference*, pp. 1505-1510, 2009.

[53] F. Kendoul, Z. Yu, and K. Nonami, "Guidance and nonlinear control system for autonomous flight of minirotorcraft unmanned aerial vehicles," *Journal of Field Robotics*, vol. 27, no. 3, pp. 311-334, 2010.

[54] J. How, B. Bethke, A. Frank, D. Dale, and J. Vian, "Real-time indoor autonomous vehicle test environment," *IEEE Control Systems*, vol. 28, no. 2, pp. 51-64, 2008.

[55] D. Zhou and M. Schwager, "Vector field following for quadrotors using differential flatness," *Proc. of IEEE International Conference on Robotics and Automation*, pp. 6567-6572, 2014.

[56] J. Stowers, M. Hayes, and A. Bainbridge-Smith, "Altitude control of a quadrotor helicopter using depth map from microsoft kinect sensor," *Proc. of IEEE International Conference on Mechatronics*, pp. 358-362, 2011.

[57] F. Fraundorfer, L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, and M. Pollefeys, "Vision-based autonomous mapping and exploration using a quadrotor MAV," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4557-4564, 2012.

[58] N. Michael, *et al.*, "Collaborative mapping of an earthquake-damaged building via ground and aerial robots," *Journal of Field Robotics*, vol. 29, no. 5, pp. 832-841, 2012.

[59] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained MAV," *Proc. of IEEE international conference on Robotics and automation*, pp. 20-25, 2011.

[60] M. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Inversion based direct position control and trajectory following for micro aerial vehicles," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2933-2939, 2013.

[61] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," *Proc. of the AIAA Guidance, Navigation, and Control Conference*, pp. 1-20, 2007.

[62] P. Castillo, R. Lozano, and A. Dzul, "Stabilization of a mini-rotorcraft having four rotors," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 2693-2698, 2005.

[63] S. Bouabdallah, *Design and Control of Quadrotors with Application to Autonomous Flying*, Lausanne Polytechnic University, 2007.

[64] J. Gillula, H. Huang, M. Vitus, and C. Tomlin, "Design of guaranteed safe maneuvers using reachable sets: Autonomous quadrotor aerobatics in theory and practice," *IEEE International Conference on Robotics and Automation*, pp. 1649-1654, 2010.

[65] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, vol. 2, Wiley, 2007.

[66] K. Lee, J. Back, and I. Choy, "Nonlinear disturbance observer based robust attitude tracking controller for quadrotor UAVs," *International Journal of Control, Automation and Systems*, vol. 12, no. 6, pp. 1266-1275, 2014.

[67] J. Back and H. Shim, "An inner-loop controller guaranteeing robust transient performance for uncertain mimo nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 54, no. 7, pp. 1601-1607, 2009.

[68] S. Kim, H. Seo, and H. J. Kim, "Operating an unknown drawer using an aerial manipulator," *Proc. of IEEE International Conference on Robotics and Automation*, pp. 5503-5508, 2015.

[69] H. Lee, S. Kim, and H. J. Kim, "Control of an aerial manipulator using on-line parameter estimator for an unknown payload," *Proc. of IEEE International Conference on Automation Science and Engineering*, pp. 316-321, 2015.

[70] H. Lee, H. Kim, and H. J. Kim, "Path planning and control of multiple aerial manipulators for a cooperative transportation," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2386-2391, 2015.

[71] A. Tayebi and S. McGilvray, "Attitude stabilization of a VTOL quadrotor aircraft," *IEEE Transactions on Control System Technology*, vol. 14, no. 3, pp. 562-571, 2006.

[72] M. Turpin, N. Michael, and V. Kumar, "Trajectory design and control for aggressive formation flight with quadrotors," *Autonomous Robots*, vol. 33, no. 1, pp. 143-156, 2012.

[73] K. Sreenath, T. Lee, and V. Kumar, "Geometric control and differential flatness of a quadrotor UAV with a cablesuspended load," *Proc. of IEEE 52nd Annual Conference on Decision and Control*, pp. 2269-2274, 2013.

[74] 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*, vol. 19, no. 3, pp. 33-45, 2012.

[75] C. Lehnert and P. Corke, "μAV-design and implementation of an open source micro quadrotor," *AC on Robotics and Automation*, Eds, 2013.

[76] M. Asadpour, B. Van den Bergh, D. Giustiniano, K. A. Hummel, S. Pollin, and B. Plattner, "Micro aerial vehicle networks: An experimental analysis of challenges and opportunities," *IEEE Communications Magazine*, vol. 52, no. 7, pp. 141-149, 2014.

[77] J. Engel, J. Sturm, and D. Cremers, "Camera-based navigation of a low-cost quadrocopter," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2815-2821, 2012.

[78] C. Bills, J. Chen, and A. Saxena, "Autonomous MAV flight in indoor environments using single image perspective cues," *Proc. of IEEE international Conference on Robotics and Automation*, pp. 5776-5783, 2011.

[79] P.-J. Bristeau, F. Callou, D. Vissiere, N. Petit, *et al.*, "The navigation and ontrol technology inside the ar. drone micro UAV," *Proc of 18th IFAC World Congress*, vol. 18, no. 1, pp. 1477-1484, 2011.

[80] "Matrice 100, DJI," http://dev.dji.com.

[81] "Vicon system," http://www.vicon.com.

[82] S. Grzonka, G. Grisetti, and W. Burgard, "Towards a navigation system for autonomous indoor flying," *Proc. of IEEE International Conference on Robotics and Automation*, pp. 2878-2883. , 2009

[83] "Endurance R/C," http://www.endurance-rc.com.

[84] "Robot operating system," http://www.ros.org.

[85] "Ardrone autonomy," http://wiki.ros.org/ardrone_autonomy.

[86] "Asctec mav framework," http://wiki.ros.org/asctec_mav_framework.

[87] "Odroid," http://www.hardkernel.com.

Hyeonbeom Lee received the B.S. degree in Mechanical and Control Engineering from Handong Global University in 2011, and the M.S. degree in Mechanical and Aerospace Engineering from Seoul National University in 2013. He is currently pursuing the Ph.D. degree in the Department of Mechanical and Ae-rospace Engineering at Seoul National University. His research interests include aerial manipulation and motion planning of aerial robots.

H. Jin Kim received the B.S. degree from Korea Advanced Institute of Technology (KAIST) in 1995, and the M.S. and Ph.D. degrees in Mechanical Engineering from University of California, Berkeley (UC Berkeley), in 1999 and 2001, respectively. From 2002 to 2004, she was a Postdoctoral Researcher in Electrical Engineering and Computer Science (EECS), UC Berkeley. In September 2004 she joined the Department of Mechanical and Aerospace Engineering at Seoul National University, Seoul, Korea, as an Assistant Professor where she is currently a Professor. Her research interests include intelligent control of robotic systems and motion planning.