

Modeling and Adaptive Backstepping Control for Quadrotor Robots with Blade Flapping

Lih-Chang Lin and Wen-Cong Xu

Department of Mechanical Engineering, National Chung Hsing University, Taichung, Taiwan 402, R.O.C.

Abstract

Dynamics modeling and control for quadrotor aerial robots are considered in this paper. The three-dimensional dynamics model of the quadrotor is first derived using the Newton-Euler approach. When the propelling rotor rotating with respect to the moving robot, the rotating blades could be deflected with an effect on the aerial vehicles known as flapping. The modeling with blade flapping is considered. Then a nonlinear stable adaptive control with flapping parameters estimation is proposed using the backstepping technique for the position and yaw angle trajectory tracking. Finally, computer simulation is used to validate the performance of the proposed control strategy. In the simulation study, the desired trajectory is constructed using the cubic spline interpolation.

Publication History:

Received: December 15, 2016

Accepted: July 04, 2016

Published: July 06, 2016

Keywords:

Quadrotor modeling, blade flapping, adaptive control, backstepping method

Introduction

Quadrotor aerial vehicles are highly maneuverable and have enabled a number of indoor and outdoor applications, e.g., three-dimensional (3D) environment exploring, mapping, navigation, transporting, military and commercial opportunities. The ability of 3D movement brings great research interest in the field of mobile robotics.

Corke [1] provides a comprehensive introduction to the 3D dynamics model of quadrotor robots using Newton's second law and Euler equations. And a PID-based inner-loop/outer-loop scheme is proposed for the 3D translation and yaw control. Mahony et al. [2] propose a tutorial introduction to the rigid-body dynamics modeling, estimation, and control for multirotor aerial vehicles including the quadrotor case. They also give a physical discussion about the rotor blade flapping referring to [3,4], and then presented a disturbed external force model due to the blade flapping effect. Beard [5] considers the problems of dynamics, state estimation, and control of a quadrotor aerial vehicle.

Benzaid et al. [6] first present the dynamics model of a quadrotor robot and then propose an inner-loop/outer-loop based control scheme. The outer-loop comprises of a nonlinear thrust control law using the backstepping approach for the 3D position tracking control. In the derivation of control law using the backstepping technique, they introduce virtual inputs for the x-y model to derive the desired roll and pitch angles for using as the commands of the inner-loop PID control. The desired yaw-angle trajectory is directly provided exogenously, and PID is used for the yaw control. The proposed approach adopts some intuitive concept for providing the roll and pitch commands, so the stability of the overall system is not guaranteed seriously.

In Yesildirek et al. [7], the quadrotor system is decomposed into three fully actuated subsystems in pitch, roll and yaw domains. They propose a multi Lyapunov function based switching control algorithm to achieve tracking of Cartesian space motion and the heading angle of the quadrotor. Hao et al. [8] propose a change in describing the dynamics of the quadrotor and separate the dynamics into three sub-groups. Then combined with a coordinate transformation, they propose a trajectory tracking control design using the backstepping technique. Zhao et al. [9] present an asymptotic tracking controller for the quadrotor using the robust integral of the signum of the error (RISE) method and an immersion and invariance (I&I)-based adaptive control methodology. The proposed control system is composed of two parts: the inner loop for attitude control (using the RISE approach) and the outer loop for position control (using the I&I

adaptive control methodology. The proposed control system is composed of two parts: the inner loop for attitude control (using the RISE approach) and the outer loop for position control (using the I&I approach).

This paper will consider the 3D dynamics modeling of the quadrotor with blade flapping effect via the Newton-Euler approach and the position and yaw angle trajectory tracking control design using the backstepping technique. Computer simulation is conducted to illustrate the trajectory tracking performance of the 3D translation and yaw motion.

Dynamics Model of a Quadrotor Robot

Refer to Figure 1, the quadrotor robot comprised of a main frame structure and four blade rotors (numbered 1, 2, 3, and 4 clockwise) is considered as a rigid body, where the body coordinate system $\{B\}$ is defined with its origin G attached to the center of mass of the whole robot, and the inertia (world) coordinate frame $\{W\}$ with origin at O is fixed on the earth surface. The positive direction of x axis is defined along the heading direction of the vehicle. Rotors 1 and 3 rotate clockwise, and rotors 2 and 4 rotate counterclockwise. In the dynamics modeling, we consider the gravity force along the negative Z_w direction, and the four thrust forces $T_i = C_f \omega_i^2$, $i = 1, 2, 3, 4$ along Z_B axis, where C_f is the aerodynamic thrust coefficient and ω_i is the rotor (blade) rotating speed. The resisting torque about the blade axis is $\tau_i = C_q \omega_i^2$, $i = 1, 2, 3, 4$, where C_q is the aerodynamic drag coefficient. Thus, the total thrust force and the roll, pitch and yaw torques τ_r , τ_p , and τ_y are as follows [12].

$$\begin{aligned} T_f &= C_f (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ \tau_r &= \frac{l}{\sqrt{2}} C_f (\omega_1^2 - \omega_2^2 - \omega_3^2 + \omega_4^2) \\ \tau_p &= \frac{l}{\sqrt{2}} C_f (-\omega_1^2 - \omega_2^2 + \omega_3^2 + \omega_4^2) \\ \tau_y &= C_q (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{aligned} \quad (1)$$

Corresponding Author: Prof. Lih-Chang Lin, Department of Mechanical Engineering, National Chung Hsing University, Taichung, Taiwan 402, R.O.C. E-mail: lclin@mail.nchu.edu.tw

Citation: Lin LC, Xu WC (2016) Modeling and Adaptive Backstepping Control for Quadrotor Robots with Blade Flapping. Int J Mech Syst Eng 2: 117. <http://dx.doi.org/10.15344/2455-7412/2016/117>

Copyright: © 2016 Lin et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

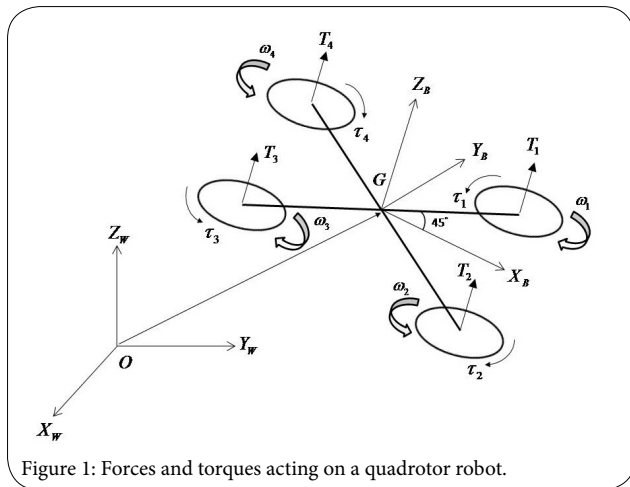


Figure 1: Forces and torques acting on a quadrotor robot.

where l is the distance from the center of mass G to each rotating axis.

ZYX Euler angles (ψ, θ, ϕ) are used for representing the orientation of the quadrotor vehicle's body frame with respect to the world frame wR_B . The first rotation is the yaw angle ψ about Z_E axis, then the second rotation is the pitch angle θ about Y_E axis, and the third rotation is the roll angle ϕ about X_E axis. Thus wR_B can be represented as [13]

$${}^wR_B = R_Z(\psi)R_Y(\theta)R_X(\phi)$$

$$= \begin{bmatrix} c\psi c\theta & -s\psi c\theta & c\psi s\theta s\phi & s\psi s\theta s\phi + c\psi s\theta c\phi \\ s\psi c\theta & c\psi c\theta & s\psi s\theta s\phi & -c\psi s\theta s\phi + s\psi s\theta c\phi \\ -s\theta & c\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}$$

where, $s\phi = \sin \phi$, $c\phi = \cos \phi$ etc.

Define $t = [x, y, z]^T$ as the position vector of the center of mass G of the quadrotor robot in terms of the world frame, and assume that the robot structure is symmetrical about the coordinate system. Consider the quadrotor robot being under normal flight conditions, i.e., the roll and pitch angles are very small, and the flight drag force is negligible. By the Newton's second law and Euler equations of motion, we can derive the dynamics model of the quadrotor robot as follows: [1,14]

$$\begin{cases} \ddot{x} = (s\psi s\phi + c\psi s\theta c\phi) \frac{T_f}{m} \\ \ddot{y} = (-c\psi s\phi + s\psi s\theta c\phi) \frac{T_f}{m} \\ \ddot{z} = -g + c\theta c\phi \frac{T_f}{m} \end{cases} \quad (2a)$$

$$\begin{cases} \ddot{\phi} = \begin{pmatrix} I_{yy} - I_{zz} \\ I_{xx} \end{pmatrix} \dot{\theta} \dot{\psi} + \frac{\tau_x}{I_{xx}} \\ \ddot{\theta} = \begin{pmatrix} I_{zz} - I_{xx} \\ I_{yy} \end{pmatrix} \dot{\psi} \dot{\phi} + \frac{\tau_y}{I_{yy}} \\ \ddot{\psi} = \begin{pmatrix} I_{xx} - I_{yy} \\ I_{zz} \end{pmatrix} \dot{\phi} \dot{\theta} + \frac{\tau_z}{I_{zz}} \end{cases} \quad (2b)$$

where m is the mass of the quadrotor robot, g is the gravitation acceleration constant, $I = \text{diag}[I_{xx}, I_{yy}, I_{zz}]$ is the mass inertia matrix about the body frame $\{B\}$, I_{xx} , I_{yy} , and I_{zz} are the moments of inertia about Z_B axes, respectively.

Mahony et al. [2] considered the blade flapping effect and proposed a simplified flapping model. Neglecting the effect of the vertical velocity component, we consider the quadrotor robot being with a flight velocity vector $v^B = [u, v, 0]^T$. In order to simplify the flapping model, the four rotors are assumed with same conditions. The aerodynamic disturbance forces resulting from the deflection of the rotor blades in the $X_B Y_B$ plane can be derived as follows. As shown in Figure 2, define the flapping angles β and β^\perp as [2]

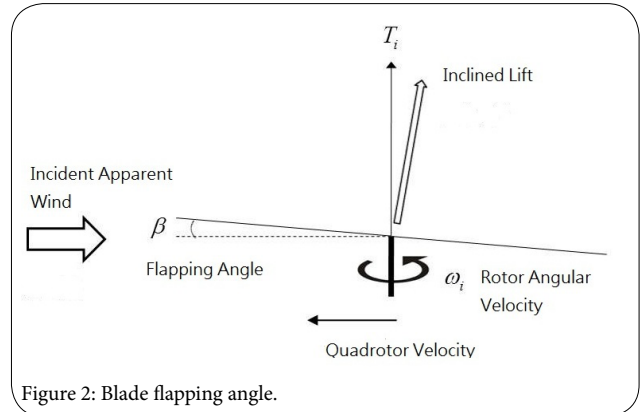


Figure 2: Blade flapping angle.

$$\beta = -\frac{\mu A_{lc}}{(1 - \frac{1}{2}\mu^2)}; \beta^\perp = -\frac{\mu A_{ls}}{(1 + \frac{1}{2}\mu^2)} \quad (3)$$

where β is the tilt angle of the rotor away from the incoming apparent wind, and β^\perp is the tilt orthogonal to the incident wind; A_{lc} and A_{ls} are positive constants; $\mu = \sqrt{u^2 + v^2} / (\omega r)$ is the advance ratio (ratio of the robot translational speed to the blade's tip tangential speed relative to the robot). Consider the case of small advance ratio μ , thus $\mu^2 \approx 0$, and the flapping angles are small and become as follows:

$$A_{flap} = \frac{1}{\omega r} \begin{bmatrix} A_{lc} & A_{ls} & 0 \\ -A_{ls} & A_{lc} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4)$$

then the total force acting on the quadrotor robot can be represented as:

$$F^B = \begin{bmatrix} 0 \\ 0 \\ T_f \end{bmatrix} - T_f A_{flap} v^B \quad (5)$$

Substituting A_{flap} and Equation (4) into Equation (5), the translational equations of motion of the quadrotor robot can be derived as follows using the Newton's second law:

$$\begin{cases} \ddot{x} = [c\psi c\theta(\beta \cos \alpha + \beta^\perp \sin \alpha) \\ + (-s\psi c\theta + c\psi s\theta s\phi)(-\beta^\perp \cos \alpha + \beta \sin \alpha) \\ + (s\psi s\phi + c\psi s\theta c\phi)] T_f / m \\ \ddot{y} = [s\psi c\theta(\beta \cos \alpha + \beta^\perp \sin \alpha) \\ + (c\psi c\theta + s\psi s\theta s\phi)(-\beta^\perp \cos \alpha + \beta \sin \alpha) \\ + (-c\psi s\phi + s\psi s\theta c\phi)] T_f / m \\ \ddot{z} = [-s\theta(\beta \cos \alpha + \beta^\perp \sin \alpha) \\ + c\theta s\phi(-\beta^\perp \cos \alpha + \beta \sin \alpha) + c\theta c\phi] T_f / m - g \end{cases} \quad (7)$$

where $\alpha = \text{atan2}(v, u)$ is the angle between the velocity vector v^B and X_B axis. And the rotational equations of the robot are still expressed by Equations (2b).

Adaptive Control via Backstepping for Quadrotors with Blade Flapping

Since a quadrotor aerial vehicle has only four rotor actuators, it is an underactuated system and difficult to control it to have arbitrary six degrees-of-freedom motion in the free Cartesian space. Due to the limit of only four rotors, it is natural to choose the four important motions including the three translational components and the heading rotation (i.e., the yaw angle) as the control objectives. That is, we can set four arbitrary desired trajectories: $x_d(t)$, $y_d(t)$, $z_d(t)$ and $\psi_d(t)$, and propose a suitable stable control law to obtain the trajectory tracking control objective.

In this paper, we will propose a nonlinear stable adaptive control design using the backstepping approach [6,15] for quadrotor robots considering the blade flapping effect. Based on the dynamics model: Equations (6) and (2b), and after substituting (1) into the model, we can first define the following four control input variables:

$$\begin{cases} U_1 = C_f(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ U_2 = C_f(\omega_1^2 - \omega_2^2 - \omega_3^2 + \omega_4^2) \\ U_3 = C_f(-\omega_1^2 - \omega_2^2 + \omega_3^2 + \omega_4^2) \\ U_4 = C_g(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{cases} \quad (7)$$

and choose the state vector $\mathbf{x} = [\mathbf{x}_1^T \ \mathbf{x}_2^T]^T$
 $\mathbf{x}_1 = [x \ y \ z \ \phi \ \theta \ \psi]^T$, $\mathbf{x}_2 = [\dot{x} \ \dot{y} \ \dot{z} \ \dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$

After defining the regression vectors as:

$$\begin{cases} \boldsymbol{\varphi}_z = [-s\theta c\alpha + c\theta s\phi s\alpha \quad -s\theta s\alpha - c\theta s\phi c\alpha]^T \\ \boldsymbol{\varphi}_x = [c\psi c\theta c\alpha + (-s\psi c\phi + c\psi s\theta s\phi) s\alpha \\ \quad c\psi c\theta s\alpha + (s\psi c\phi - c\psi s\theta s\phi) c\alpha]^T \\ \boldsymbol{\varphi}_y = [s\psi c\theta c\alpha + (c\psi c\phi + s\psi s\theta s\phi) s\alpha \\ \quad s\psi c\theta s\alpha - (c\psi c\phi + s\psi s\theta s\phi) c\alpha]^T \end{cases} \quad (8)$$

and the unknown flapping angles vector as:

$$\hat{\mathbf{i}} = [\boldsymbol{\beta} \ \boldsymbol{\beta}^\perp]^T \quad (9)$$

Equations (6) can be expressed as the regression model:

$$\begin{cases} \ddot{x} = \frac{1}{m}(\boldsymbol{\varphi}_x^T \boldsymbol{\xi} + u_x) U_1 \\ \ddot{y} = \frac{1}{m}(\boldsymbol{\varphi}_y^T \boldsymbol{\xi} + u_y) U_1 \\ \ddot{z} = -g + \frac{1}{m}(\boldsymbol{\varphi}_z^T \boldsymbol{\xi} + c\theta c\phi) U_1 \end{cases} \quad (10a)$$

where

$$u_x = s\psi s\phi + c\psi s\theta c\phi, \quad u_y = -c\psi s\phi + s\psi s\theta c\phi.$$

And $d_x = \boldsymbol{\varphi}_x^T \boldsymbol{\xi}$, and $d_y = \boldsymbol{\varphi}_y^T \boldsymbol{\xi}$ are $d_z = \boldsymbol{\varphi}_z^T \boldsymbol{\xi}$ the flapping disturbance terms along the axes, respectively. Equations (2b) can be expressed as

$$\begin{cases} \ddot{\phi} = \frac{I_{yy} - I_{zz}}{I_{xx}} \dot{\theta} \dot{\psi} + \frac{l}{\sqrt{2} I_{xx}} U_2 \\ \ddot{\theta} = \frac{I_{zz} - I_{xx}}{I_{yy}} \dot{\psi} \dot{\phi} + \frac{l}{\sqrt{2} I_{yy}} U_3 \\ \ddot{\psi} = \frac{I_{xx} - I_{yy}}{I_{zz}} \dot{\phi} \dot{\theta} + \frac{l}{I_{zz}} U_4 \end{cases} \quad (10b)$$

In the derivation of stable adaptive control law, the estimate of the unknown flapping angles is defined as $\hat{\boldsymbol{\xi}} = [\boldsymbol{\beta} \ \boldsymbol{\beta}^\perp]^T$, and the estimation error is defined as $\tilde{\boldsymbol{\xi}} = \boldsymbol{\xi} - \hat{\boldsymbol{\xi}}$. The estimates of the flapping disturbances are defined as: $\hat{d}_x = \boldsymbol{\varphi}_x^T \hat{\boldsymbol{\xi}}$, $\hat{d}_y = \boldsymbol{\varphi}_y^T \hat{\boldsymbol{\xi}}$, $\hat{d}_z = \boldsymbol{\varphi}_z^T \hat{\boldsymbol{\xi}}$.

Z-axis motion control design

First, consider the $\dot{x}_3 = x_3$ subsystem. Introducing the virtual input β_1 , we have:

$$\dot{x}_3 = \beta_1$$

Define the tracking error

$$e_1 = Z_d - x_3$$

where $Z_d(t)$ is the desired trajectory for z . Taking the time derivative, we have:

$$\dot{e}_1 = \dot{Z}_d - \dot{x}_3 = \dot{Z}_d - \beta_1$$

Choosing Lyapunov function candidate as:

$$V_1 = \frac{1}{2} \left[e_1^2 + \lambda_1 \left(\int_0^t e_1(\tau) d\tau \right)^2 \right], \quad \lambda_1 > 0 \quad (11)$$

we can obtain the time derivative as follows:

$$\dot{V}_1 = e_1 \left(\dot{Z}_d - \beta_1 + \lambda_1 \int_0^t e_1(\tau) d\tau \right)$$

By choosing the virtual input β_1 as:

$$\beta_1 = \dot{Z}_d + \lambda_1 \int_0^t e_1(\tau) d\tau + c_1 e_1, \quad c_1 > 0 \quad (12)$$

we have

$$\dot{V}_1 = -c_1 e_1^2 \leq 0 \quad (13)$$

Consider the subsystem:

$$\begin{cases} \dot{x}_3 = x_3 \\ \dot{x}_9 = -g + \frac{1}{m}(\boldsymbol{\varphi}_z^T \boldsymbol{\xi} + c\theta c\phi) U_1 \end{cases} \quad (14)$$

And define

$$e_2 = \beta_1 - x_3 = \dot{Z}_d + \lambda_1 \int_0^t e_1(\tau) d\tau + c_1 e_1 - x_3$$

we can obtain:

$$\dot{e}_2 = \ddot{Z}_d + \lambda_1 e_1 + c_1 \dot{e}_1 - \dot{x}_3$$

Choosing Lyapunov function candidate:

$$V_2 = V_1 + \frac{1}{2} e_2^2 = \frac{1}{2} \left[e_1^2 + \lambda_1 \left(\int_0^t e_1(\tau) d\tau \right)^2 + e_2^2 \right]$$

and taking the time derivative, we have:

$$\begin{aligned} \dot{V}_2 = & -c_1 e_1^2 + e_2 \left[(1 + \lambda_1 - c_1^2) e_1 + c_1 e_2 \right. \\ & \left. + \ddot{Z}_d - c_1 \lambda_1 \int_0^t e_1(\tau) d\tau + g \right. \\ & \left. - \frac{1}{m} (\boldsymbol{\varphi}_z^T \tilde{\boldsymbol{\xi}} + \boldsymbol{\varphi}_z^T \hat{\boldsymbol{\xi}} + c\theta c\phi) U_1 \right] \end{aligned} \quad (15)$$

Hence, we can choose the control law:

$$\begin{aligned} U_1 = & \frac{m}{\boldsymbol{\varphi}_z^T \hat{\boldsymbol{\xi}} + c\theta c\phi} \left[g + (1 + \lambda_1 - c_1^2) e_1 + c_1 e_2 \right. \\ & \left. + \ddot{Z}_d - c_1 \lambda_1 \int_0^t e_1(\tau) d\tau + c_2 e_2 \right], \quad c_2 > 0 \end{aligned} \quad (16)$$

and obtain:

$$\dot{V}_2 = -c_1 e_1^2 - c_2 e_2^2 - \frac{e_2 U_1}{m} \boldsymbol{\varphi}_z^T \tilde{\boldsymbol{\xi}}$$

x-axis subsystem control design

Similarly as the z-axis design procedure, the x-axis subsystem can be considered by introducing the virtual inputs β_2, β_{ux} for x_7 and u_x , respectively, and defining the errors:

$$\begin{cases} e_3 = x_d - x_1 \\ e_4 = \beta_2 - x_7 \end{cases} \quad (17)$$

where $x_d(t)$ is the desired trajectory for x. Choose Lyapunov function candidates:

$$V_3 = \frac{1}{2} \left[e_3^2 + \lambda_2 \left(\int_0^t e_3(\tau) d\tau \right)^2 \right], \quad \lambda_2 > 0 \quad (18)$$

$$V_4 = \frac{1}{2} \left[e_3^2 + \lambda_2 \left(\int_0^t e_3(\tau) d\tau \right)^2 + e_4^2 \right]$$

After taking their time derivatives and choosing β_2 and β_{ux} as:

$$\beta_2 = \dot{x}_d + \lambda_2 \int_0^t e_3(\tau) d\tau + c_3 e_3, \quad c_3 > 0$$

$$\beta_{ux} = \frac{m}{U_1} \left[(1 + \lambda_2 - c_3^2) e_3 + c_3 e_4 \right.$$

$$\left. + \ddot{x}_d - c_3 \lambda_2 \int_0^t e_3(\tau) d\tau + c_4 e_4 \right] - \boldsymbol{\varphi}_x^T \tilde{\boldsymbol{\xi}}, \quad c_4 > 0 \quad (19)$$

we can obtain:

$$\begin{aligned} \dot{V}_3 &= -c_3 e_3^2 \leq 0 \\ \dot{V}_4 &= -c_3 e_3^2 - c_4 e_4^2 - \frac{e_4 U_1}{m} \boldsymbol{\varphi}_x^T \tilde{\boldsymbol{\xi}} \end{aligned} \quad (20)$$

y-axis subsystem control design

Similarly, the y-axis subsystem can be considered by introducing the virtual inputs β_3, β_{wy} for x_8 and u_y respectively, and defining the errors:

$$\begin{cases} e_5 = y_d - x_2 \\ e_6 = \beta_3 - x_8 \end{cases} \quad (21)$$

where $Y_d(t)$ is the desired trajectory for y. Choose Lyapunov function candidates:

$$\begin{aligned} V_5 &= \frac{1}{2} \left[e_5^2 + \lambda_3 \left(\int_0^t e_5(\tau) d\tau \right)^2 \right], \quad \lambda_3 > 0 \\ V_6 &= \frac{1}{2} \left[e_5^2 + \lambda_3 \left(\int_0^t e_5(\tau) d\tau \right)^2 + e_6^2 \right] \end{aligned} \quad (22)$$

After taking their time derivatives and choosing β_3 and β_{wy} as:

$$\begin{aligned} \beta_3 &= \dot{y}_d + \lambda_3 \int_0^t e_5(\tau) d\tau + c_5 e_5, \quad c_5 > 0 \\ \beta_{wy} &= \frac{m}{U_1} \left[(1 + \lambda_3 - c_5^2) e_5 + c_5 e_6 \right. \\ &\quad \left. + \ddot{y}_d - c_5 \lambda_3 \int_0^t e_5(\tau) d\tau + c_6 e_6 \right] - \boldsymbol{\varphi}_y^T \hat{\boldsymbol{\xi}}, \quad c_6 > 0 \end{aligned} \quad (23)$$

we can obtain:

$$\dot{V}_6 = -c_5 e_5^2 - c_6 e_6^2 - \frac{e_6 U_1}{m} \boldsymbol{\varphi}_y^T \tilde{\boldsymbol{\xi}} \quad (24)$$

Flapping Angles Estimator Design

Choose Lyapunov function candidate:

$$V_t = V_2 + V_4 + V_6 + \frac{1}{2} \tilde{\boldsymbol{\xi}}^T \boldsymbol{\Gamma}^{-1} \tilde{\boldsymbol{\xi}} \quad (25)$$

where $\boldsymbol{\Gamma}$ is a 2x2 positive-definite weight matrix. Taking the time derivative and substituting into U_1, β_{ux} , and $\tilde{\boldsymbol{\xi}} \approx 0$ and using the assumption of $\dot{\tilde{\boldsymbol{\xi}}} = -\tilde{\boldsymbol{\xi}}$ we can obtain:

$$\begin{aligned} \dot{V}_t &= -c_1 e_1^2 - c_2 e_2^2 - c_3 e_3^2 - c_4 e_4^2 - c_5 e_5^2 - c_6 e_6^2 \\ &\quad - \left[\frac{U_1}{m} (e_2 \boldsymbol{\varphi}_z^T + e_4 \boldsymbol{\varphi}_x^T + e_6 \boldsymbol{\varphi}_y^T) + \dot{\tilde{\boldsymbol{\xi}}}^T \boldsymbol{\Gamma}^{-1} \tilde{\boldsymbol{\xi}} \right] \tilde{\boldsymbol{\xi}} \end{aligned} \quad (26)$$

Thus, we can choose the unknown flapping angles estimator as:

$$\dot{\tilde{\boldsymbol{\xi}}} = -\frac{U_1}{m} \boldsymbol{\Gamma} (e_2 \boldsymbol{\varphi}_z + e_4 \boldsymbol{\varphi}_x + e_6 \boldsymbol{\varphi}_y) \quad (27)$$

and obtain

$$\dot{V}_t = -\sum_{i=1}^6 c_i e_i^2 \leq 0 \quad (28)$$

Since \dot{V}_t is negative definite, the adaptive translation subsystem is asymptotically stable [16].

Pitch Subsystem Control Design

Consider the pitch subsystem and introduce virtual input β for x_{11} . Define the error

$$e_{ux} = \beta_{ux} - u_x = \beta_{ux} - (s\psi s\phi + c\psi s\theta c\phi) \quad (29)$$

and choose Lyapunov function candidate:

$$V_7 = V_t + \frac{1}{2} K_{ux} e_{ux}^2, \quad K_{ux} > 0 \quad (30)$$

In order to make the time derivative \dot{V}_7 negative definite, we can select β_e as:

$$\begin{aligned} \beta_\theta &= \frac{1}{c\psi c\theta c\phi} \left[\dot{\beta}_{ux} + (s\psi s\theta c\phi - c\psi s\phi) x_{12} \right. \\ &\quad \left. + (c\psi s\theta s\phi - s\psi c\phi) x_{10} + \frac{1}{K_{ux}} \left(\frac{1}{m} e_4 U_1 + c_{ux} e_{ux} \right) \right] \end{aligned} \quad (31)$$

where $c_{ux} > 0$

Define the error $e_{\beta\theta} = \beta_\theta - x_{11}$, and choose Lyapunov function candidate as:

$$V_8 = V_7 + \frac{1}{2} K_{\beta\theta} e_{\beta\theta}^2, \quad K_{\beta\theta} > 0 \quad (32)$$

In order to make the time derivative \dot{V}_8 negative definite, we can select the control law U_3 as follows:

$$\begin{aligned} U_3 &= \frac{\sqrt{2} I_{yy}}{l} \left[\dot{\beta}_\theta - \left(\frac{I_{zz} - I_{xx}}{I_{yy}} x_{12} x_{10} \right) \right. \\ &\quad \left. + \frac{1}{K_{\beta\theta}} (K_{ux} e_{ux} c\psi c\theta c\phi + c_{\beta\theta} e_{\beta\theta}) \right], \quad c_{\beta\theta} > 0 \end{aligned} \quad (33)$$

Roll Subsystem Control Design

Consider the roll subsystem and introduce virtual input input β_ϕ for x_{10} . Define the error

$$e_{uy} = \beta_{uy} - u_y = \beta_{uy} - (-c\psi s\phi + s\psi s\theta c\phi) \quad (34)$$

and choose Lyapunov function candidate:

$$V_9 = V_t + \frac{1}{2} K_{uy} e_{uy}^2, \quad K_{uy} > 0 \quad (35)$$

To make \dot{V}_9 negative definite, we can select β_ϕ as:

$$\begin{aligned} \beta_\phi &= \frac{1}{c\psi c\phi + s\psi s\theta s\phi} \left[-\dot{\beta}_{wy} \right. \\ &\quad \left. + (s\psi s\phi + c\psi s\theta c\phi) x_{12} + s\psi c\theta c\phi x_{11} \right. \\ &\quad \left. + \frac{1}{K_{uy}} \left(-\frac{1}{m} e_6 U_1 - c_{uy} e_{uy} \right) \right], \quad c_{uy} > 0 \end{aligned} \quad (36)$$

Define the error $e_{\beta\phi} = \beta_\phi - x_{10}$, and choose Lyapunov function candidate as:

$$V_{10} = V_9 + \frac{1}{2} K_{\beta\phi} e_{\beta\phi}^2, \quad K_{\beta\phi} > 0 \quad (37)$$

In order to make the time derivative \dot{V}_{10} negative definite, we can select the control law U_2 as follows:

$$\begin{aligned} U_2 &= \frac{\sqrt{2} I_{xx}}{l} \left[\dot{\beta}_\phi - \left(\frac{I_{yy} - I_{zz}}{I_{xx}} x_{11} x_{12} \right) \right. \\ &\quad \left. + \frac{1}{K_{\beta\phi}} (-K_{uy} e_{uy} (c\psi c\phi + s\psi s\theta s\phi) + c_{\beta\phi} e_{\beta\phi}) \right], \quad c_{\beta\phi} > 0 \end{aligned} \quad (38)$$

Yaw Subsystem Control Design

Finally, consider the control of the yaw subsystem:

$$\begin{cases} \dot{x}_6 = x_{12} \\ \dot{x}_{12} = \frac{I_{xx} - I_{yy}}{I_{zz}} x_{10} x_{11} + \frac{U_4}{I_{zz}} \end{cases} \quad (39)$$

First, introduce the virtual input β_4 for x_{12} , and consider:

$$\dot{x}_6 = \beta_4 \quad (40)$$

Define the yaw tracking error:

$$e_7 = \psi_d - x_6 \quad (41)$$

where ψ_d is the desired trajectory for ψ . Consider Lyapunov function candidate:

$$V_{11} = \frac{1}{2} \left[e_7^2 + \lambda_4 \left(\int_0^t e_7(\tau) d\tau \right)^2 \right], \lambda_4 > 0 \tag{42}$$

After taking the time derivative \dot{V}_{11} and selecting

$$\beta_4 = \dot{\psi}_d + \lambda_4 \int_0^t e_7(\tau) d\tau + c_7 e_7, \quad c_7 > 0 \tag{43}$$

we can obtain:

$$\dot{V}_{11} = -c_7 e_7^2 \leq 0 \tag{44}$$

Then define the error

$$e_8 = \beta_4 - x_{12} \tag{45}$$

and choose Lyapunov function candidate:

$$V_{12} = \frac{1}{2} \left[e_7^2 + \lambda_4 \left(\int_0^t e_7(\tau) d\tau \right)^2 + e_8^2 \right] \tag{46}$$

After taking the time derivative \dot{V}_{12} and selecting the control law U_4 as follows:

$$U_4 = I_{zz} \left[-\frac{I_{xx} - I_{yy}}{I_{zz}} x_{10} x_{11} + (1 + \lambda_4 - c_7^2) e_7 + c_7 e_8 + \ddot{\psi}_d - c_7 \lambda_4 \int_0^t e_7(\tau) d\tau + c_8 e_8 \right], \quad c_8 > 0 \tag{47}$$

we can obtain:

$$\dot{V}_{12} = -c_7 e_7^2 - c_8 e_8^2 \leq 0 \tag{48}$$

Thus, the closed-loop yaw subsystem is asymptotically stable.

The proposed overall control system for a quadrotor robot can be summarized as shown in Figure 3.

same height in 50 seconds. Then the quadrotor is required to maintain a steady hover in the final 10 seconds. The yaw angle is kept constant at ψ in the whole flight.

In the simulation study, the parameters of the quadrotor robot are selected as follows $m=0.8$ kg, $r=0.1$ m, $g=9.81$ m/s², $l=0.5$ m, $A_{1c} = A_{1s} = 1.5$,

$$I = \text{diag} \left[1.5 \times 10^{-3} \quad 1.5 \times 10^{-3} \quad 1.5 \times 10^{-3} \right] \text{ kg} \cdot \text{m}^2,$$

$$C_f = 1.2459 \times 10^{-6} \text{ N / rpm}^2, \quad C_q = 3.29 \times 10^{-11} \text{ Nm / rpm}^2.$$

The parameters of the controller used are as follows:

$$\lambda_1 = 5 \times 10^{-5}, \lambda_2 = 4 \times 10^{-5}, \lambda_3 = 5 \times 10^{-5}, \lambda_4 = 2 \times 10^{-3}, c_1 = 2.7, c_2 = 2.5, c_3 = 6, c_4 = 5.5, c_5 = 6, c_6 = 5, c_7 = 3, c_8 = 2,$$

$$K_{ux} = 0.6, K_{uy} = 0.9, K_{\beta\theta} = 0.5, K_{\beta\phi} = 0.8, c_{ux} = 16, c_{uy} = 15, c_{\beta\theta} = 16, c_{\beta\phi} = 22, \text{ and } \Gamma = \text{diag} \left[7 \times 10^{-5}, 7 \times 10^{-5} \right].$$

The three-dimensional tracking result of the whole trajectory is shown in Figure 4. The trajectory tracking in the x , y , and z axes and the yaw angle tracking are shown in Figure 4.1. The tracking errors are shown in Figure 4.2. The induced roll and pitch angles during the flight are shown in Figure 4.3. Equation (7) describes the relation between the control variables U_i and the squared rotor speeds. Using the inverse transform we can find the required squared rotor speeds corresponding to the control signals. In order to get all positive the parameters of the controller must be suitably chosen. In this work, the response in the y -axis is thus lagged the desired trajectory approximately from $t=10$ s and has larger tracking error and the largest tracking error is approximately 0.23 m. From Figure 4.3, we know that

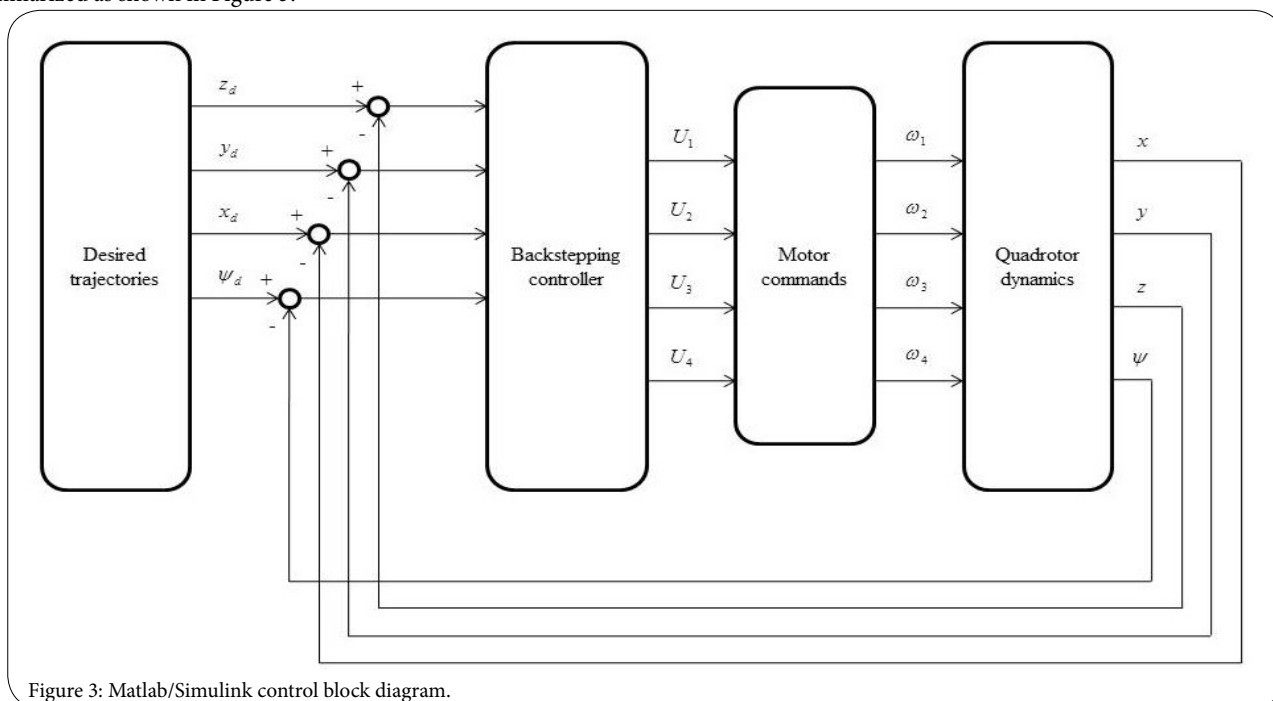


Figure 3: Matlab/Simulink control block diagram.

Computer Simulation and Discussion

The tracking performance of the proposed adaptive control system with flapping compensation is tested via computer simulation using Matlab/Simulink as shown in Figure 3. Consider the desired trajectory created using the cubic spline method as shown in Figure 4. The quadrotor is requested to flight straightly up to a height of 4 m from the origin of the world coordinate system in 10 seconds, and then move counterclockwise in a circle with radius 5 m keeping the

the roll and pitch angles are small and reasonable, thus the flight is stable. The largest tracking errors in the x and z axes are approximately 0.05 m and 0.02 m, respectively. The yaw angle can be kept at ψ as shown in Figures 4.1 and 4.2. Figure 4.3 shows that in order to keep steady hovering, the required roll and pitch angles are larger during the hovering period. The estimates of the flapping angles are shown in Figure 4.4, and the estimate errors are larger in the hovering period.

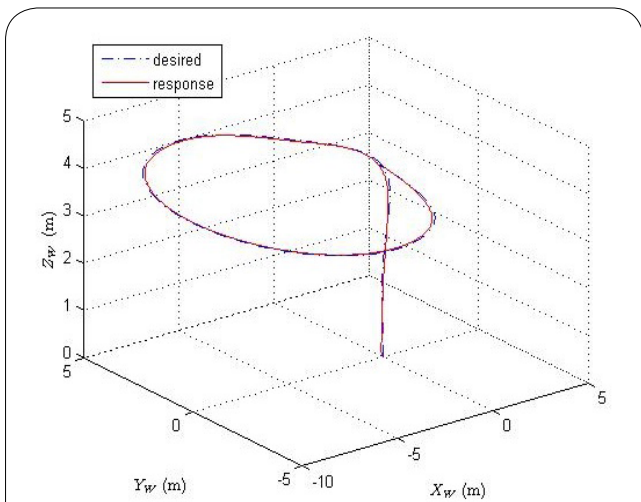


Figure 4: Desired and response trajectories.

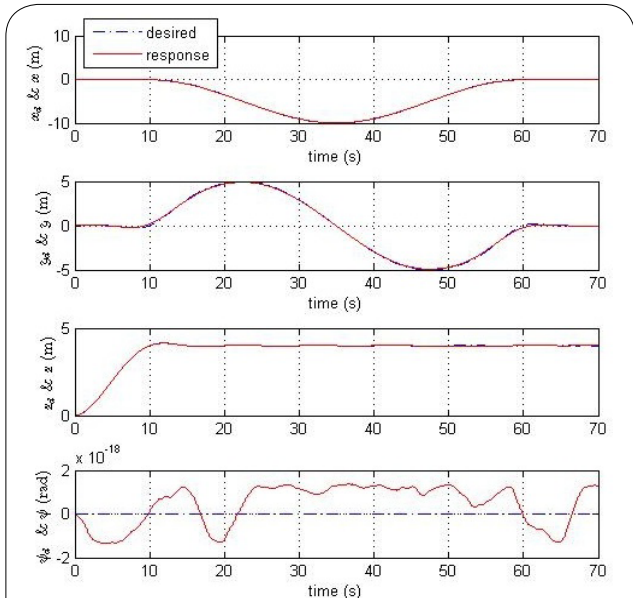


Figure 4.1: Tracking results of the position and yaw.

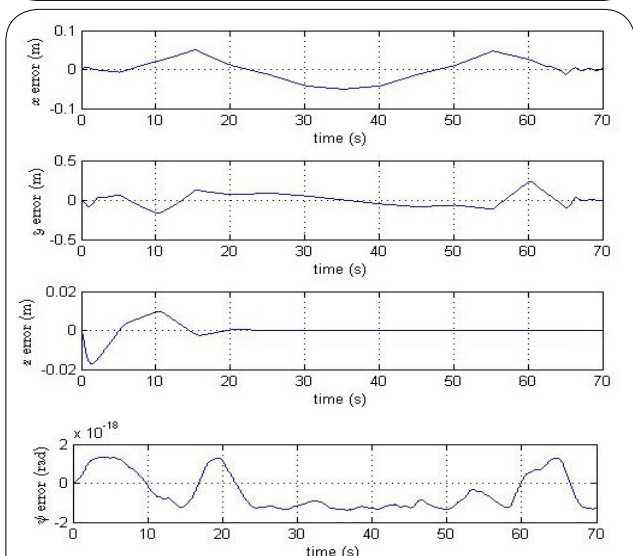


Figure 4.2: Tracking errors of the position and yaw.

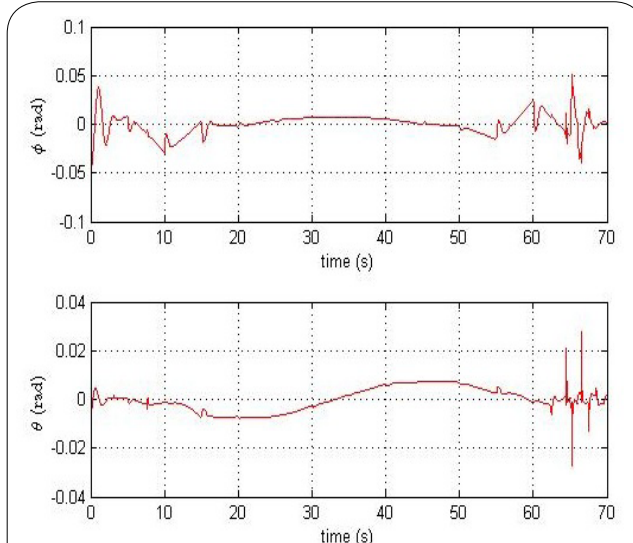


Figure 4.3: Roll and pitch angle responses.

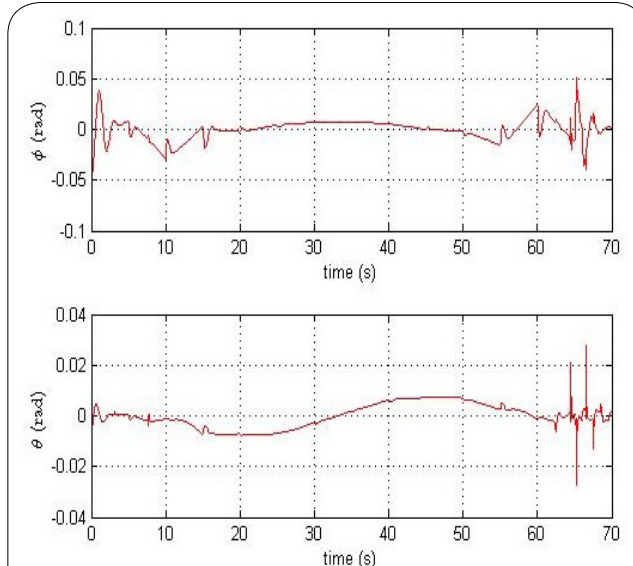


Figure 4.4: Actual and estimated flapping angles.

Conclusion

This article considers the dynamics modeling and nonlinear stable adaptive control design with blade flapping compensation for the quadrotor robots. The quadrotor robot is under-actuated since it has only four rotor actuators for three-dimensional flight maneuvering. It is rather difficult to construct an arbitrary six-dimensional flight control, so the usual approach to the control design adopts only the three-dimensional translation and yaw angle tracking control. The dynamic x-axis motion of a quadrotor could be induced by suitable small pitch rotation, and the y-axis motion could be affected by the roll motion. Based on these characteristics, this paper proposes a backstepping-based stable adaptive control design considering the blade flapping effect. Computer simulation is conducted to illustrate the trajectory tracking performance of the 3D position and heading control. The roll and pitch angles generated by the controller are small enough, so the flight is stable and reasonable. In the simulation, the choice of the controller parameters is compromised for getting positive squared rotor speeds to compute practical rotor speed commands.

Competing Interests

The authors have no competing interests with the work presented in this manuscript.

Author Contributions

All the authors substantially contributed to the study conception and design as well as the acquisition and interpretation of the data and drafting of the manuscript.

Funding

This research was supported by the R.O.C. Ministry of Science and Technology Grant MOST 104-2221-E- 005-030.

References

1. Corke P (2013) *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer.
2. Mahony RE, Kumar V, Corke P (2012) Multirotor aerial vehicles: modeling, estimation, and control of quadrotor. *IEEE Robot Automat Mag* 19: 20-32.
3. Huang H, Hoffmann GM, Waslander SL, Tomlin CJ (2009) Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. *Proc IEEE Int Conf Robot Automat*, 3277-3282.
4. Leishman J (2002) *Principles of Helicopter Aerodynamics*. Cambridge University Press.
5. Beard R (2008) Quadrotor dynamics and control Rev 0.1. All Faculty Publications. Paper 1325.
6. Benzaid K, Mansouri N, Labbani-Igbida, O (2013) Robust trajectory tracking control of a quadrotor UAV. *Proc Int Conf third Syst Contr (ICSC)*, 206-211.
7. Yesildirek A, Imran B (2014) Nonlinear control of quadrotor using multi Lyapunov functions. *American Control Conference*, 3844-3849.
8. Hao ND, Mohamed B, Rafaralahy H (2014) Trajectory-tracking control design for an under- actuated quadrotor. *European Control Conference (ECC)*, 1765-1770.
9. Zhao B, Xian B, Zhang Y, Zhang X (2015) Nonlinear robust adaptive tracking control of a quadrotor UAV via immersion and invariance methodology. *IEEE Trans Industrial Electronics* 62: 2891-2902.
10. Sadr S, Moosavian SAA, Zarafshan P (2014) Dynamics modeling and control of a quadrotor with swing load. *J Robotics* 265897.
11. Rinaldi F, Chiesa S, Quagliotti F (2013) Linear quadratic control for quadrotors UAVs dynamics and formation flight. *J Intell Robot Syst* 70: 203-220.
12. Sun Y (2012) *Modeling, Identification and Control of a Quad-rotor Drone Using Low-Resolution Sensing*. Master's Thesis, University of Illinois at Urbana- Champaign.
13. Craig JJ (1989) *Introduction to Robotics: Mechanics and Control*, (2nd Edn), Addison-Wesley.
14. Spong MW, Hutchinson S, Vidyasagar M (2006) *Robot Modeling and Control*, Wiley.
15. Spooner JT, Maggiore M, Ordóñez R, Passino KM (2002) *Adaptive Control and Estimation for Nonlinear Systems: Neural and Fuzzy Approximator Techniques*. Wiley.
16. Slotine J-JE, Li W (2005) *Applied Nonlinear Control*. Pearson Education Taiwan Ltd.