Trajectory Tracking Control of a Quadrotor Unmanned Mini-Helicopter

Zongyu Zuo*
The Seventh Research Division, Beijing University of Aeronautics and Astronautics, Beijing, China, 100191
National Key Laboratory of Science and Technology on Holistic Control

Ming Zhu†
School of Aeronautics and Engineering, Beijing University of Aeronautics and Astronautics, Beijing, China, 100191

Zewei Zheng‡
The Seventh Research Division, Beijing University of Aeronautics and Astronautics, Beijing, China, 100191
National Key Laboratory of Science and Technology on Holistic Control

In this paper, the relationship between attitude and linear acceleration of the quadrotor unmanned mini-helicopter is built and two trajectory tracking control design methodologies based on the relationship are proposed: classical inner/outer-loop control design approach and a new command filtered backstepping design strategy. Various numerical simulations demonstrate the validity of the control design. Finally, we discuss the two design methods and arrive at a conclusion via comparison.

Nomenclature

\begin{align*}
G_a & = \text{gyroscopic torques} \\
g & = \text{gravity acceleration} \\
I_f & = \text{moment of inertia matrix} \\
I_r & = \text{rotor moment of inertial} \\
k, b & = \text{rotor force parameters} \\
m & = \text{quadrotor mass} \\
p & = \text{position vector of quadrotor center of mass} \\
p, q, r & = \text{roll, pitch, yaw rate} \\
Q_i & = \text{reactive torque acts on rotor } i \\
R & = \text{rotation matrix} \\
v & = \text{linear velocity vector of the quadrotor center of mass} \\
x, y, z & = \text{position components about inertial frame} \\
Ω & = \text{angular rate vector} \\
Θ & = \text{Euler angles vector} \\
ϕ, θ, ψ & = \text{roll, pitch, yaw} \\
τ_a & = \text{control torque} \\
ω_i & = \text{rotor } i \text{ speed}
\end{align*}

This work was partially supported by Research Foundation for Key Disciplines of Beijing Municipal Commission of Education (XK100060422).

* Ph.D Candidate, Room 205, No.1 Keyan South Building, Beijing University of Aeronautics and Astronautics, 100191, Beijing. zzybobby@ss.buaa.edu.cn

† Instructor, Room 202, No.1 Keyan SouthBuilding, Beijing University of Aeronautics and Astronautics, 100191, Beijing.

‡ Ph.D Candidate, Room 205, No.1 Keyan South Building, Beijing University of Aeronautics and Astronautics, 100191, Beijing.

1 American Institute of Aeronautics and Astronautics
I. Introduction

QUADROTOR mini-helicopter, consisting of four individual rotors of “X” arrangement, is an excellent, novel vertical take-off and landing Unmanned Aerial Vehicle (UAV) for both military and civilian usages. In order to compensate the effect of the reactive torques, the four rotors are divided into two pairs of (1, 3) and (2, 4) turning in opposite direction, as shown in Fig. 1. So a quadrotor mini-helicopter is suitable for hover and pseudo-static flight.

The quadrotor mini-helicopter is a typical under-actuated, nonlinear coupled system, and the table 1 shows the quadrotor flight mechanism: vertical motion created by collectively increasing and decreasing the speed of all four rotors; pitch or roll motion is achieved by the differential speed of the front-rear set or the left-right set of rotors, coupled with lateral motion; yaw motion is realized by the different reactive torques between the (1, 3) and (2, 4) rotors. Thus the number of individual manipulating variables cannot instantaneously set the accelerations in all directions of the configuration space. In spite of the four rotors, the quadrotor is still an under-actuated and nonlinear coupled system.

| Table 1 Quadrotor Unmanned Mini-Helicopter Rotor Speed Control |
| Rotor 1 | Rotor 2 | Rotor 3 | Rotor 4 |
| Up | ↑ | ↑ | ↑ | ↑ |
| Pitch | ↑ | ↓ | ↓ | ↑ |
| Roll | ↓ | ↑ | ↓ | ↓ |
| Yaw | ↓ | ↑ | ↓ | ↑ |

McKerrow and Hamel studied the dynamical model of the four-rotor VTOL in their literature. Tayebi et al made a slight modification of the gyroscopic torque expression on the dynamical model in Ref. 1 and 2, and designed the attitude stabilization PD controller based on unit quaternion frame that is commonly employed by the attitude control problem of a rigid body. Bouabdallah applied the classical PID and LQ algorithm respectively to quadrotor attitude stabilization. However, the quadrotor flight condition and its under-actuated and strong coupled properties render the trajectory tracking control much more challenging. In Ref. 7, Bouabdallah achieved the quadrotor trajectory tracking by combining classical inner/outer loop with backstepping and sliding-mode techniques respectively, but the two-time scale separation assumption needs large inner-loop gain to guarantee closed-loop stability. Madani and et al divided the quadrotor dynamics into three subsystems using the same methodology to track the desired trajectory via full state backstepping approach. In view of the uncertainty and the unknown dynamics, Washland and Hoffmann designed an integral sliding mode controller and a reinforcement learning control scheme respectively to make a comparison, and the results demonstrated good robustness of the two controllers.

It is well known that the motion equations of a rigid body contain translation and rotation of two components, and satisfy the strict feedback form. That is why backstepping technique is so popular in flight control systems. However, analytic derivative expressions of pseudo control variables are usually overly complicated or unknown especially for the high order systems and the uncertain systems, which limit the backstepping technique in practical applications. This work adopts command filtered backstepping technique to stabilize the quadrotor attitude without calculating pseudo control signal derivative, and to decrease the dependent degree on analytic model. The trajectory tracking controller employs PD linear feedback control methodology to construct the attitude command signals. In order to avoid introducing inner- and outer-loop under the time scale separation assumption, this work presents a linear tracking-differentiator to extract the attitude command derivatives without tedious computation. This method can avoid the complication to obtain the analytic expressions of the attitude commanded signals.

This work is organized as follows. Section II presents the detailed modeling process of the quadrotor. Section III and IV give two trajectory tracking control design methodologies. In section V, numerical simulations of the two controllers are discussed. In section VI, we present our conclusions.

II. Dynamic Modeling of the Quadrotor Unmanned Mini-Helicopter

A. Kinematic Equations

Let $\mathcal{I} = \{O, x, y, z_r\}$ denote an earth-fixed inertial frame and $\mathcal{A} = \{Oxyz\}$ a body-fixed frame whose origin $O$ is at the center of mass of the quadrotor, as shown in Figure 1. The absolute position of the quadrotor is defined by $p = (x, y, z)^T$ and its attitude by the three Euler angles $\Theta = (\phi, \theta, \psi)^T$. $R \in SO(3)$ is the orthogonal rotation matrix to orient the quadrotor.
The quadrotor is a six-degree-of-freedom rigid described by three translations \( \mathbf{v} = (v_x, v_y, v_z)^T \) and three rotations \( \mathbf{Ω} = (p, q, r)^T \). Then the quadrotor mini-helicopter kinematic equations can be expressed as

\[
\mathbf{p} = \mathbf{v} \\
\mathbf{R} = \mathbf{R} \mathbf{S}(\mathbf{Ω})
\]

where \( \mathbf{S}(\mathbf{Ω}) \) is a skew-symmetric matrix and is defined as follows:

\[
\mathbf{S}(\mathbf{Ω}) = \begin{bmatrix}
0 & -r & q \\
r & 0 & -p \\
-q & p & 0
\end{bmatrix}
\]

By simply calculating Eq. (2), we obtain

\[
\dot{\mathbf{Θ}} = \mathbf{W}\mathbf{Ω}
\]

where

\[
\mathbf{W} = \begin{bmatrix}
1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\
0 & \cos \phi & -\sin \phi \\
0 & \sin \phi \sec \theta & \cos \phi \sec \theta
\end{bmatrix}
\]

and \( \det(\mathbf{W}) = \sec \theta \). So when the pitch angle satisfies \( \theta \neq (2k-1)\pi / 2, (k = 1, 2, 3, \ldots) \), the matrix \( \mathbf{W} \) is invertible.

**B. Dynamic Equations**

It is well known that any rigid motion can be described by Newton-Euler formula. As for modeling of the quadrotor, several reasonable assumptions are made:

A1 Quadrotor is a rigid body;
A2 Aerodynamic effect can be ignored at low speed;
A3 Quadrotor is symmetric with respect to axis \( \mathbf{Ox}, \mathbf{Oy} \) and \( \mathbf{Oz} \).

Thus the dynamic equations can be expressed as:

\[
\mathbf{m}\ddot{\mathbf{v}} = -\mathbf{mgz} + \mathbf{T}\mathbf{z}_e
\]

\[
\mathbf{I}_f\ddot{\mathbf{Ω}} = -\mathbf{Ω} \times \mathbf{I}_f \mathbf{Ω} - \mathbf{G}_e + \mathbf{τ}_e
\]

where \( \mathbf{m} \) denotes the quadrotor mass, \( \mathbf{g} \) the gravity acceleration, \( \mathbf{z}_e = (0,0,1)^T \) the unit vector in the frame \( \mathbf{T} \), and \( \mathbf{T} \) the total lift produced by four rotors:

\[
\mathbf{T} = \sum_{i=1}^{4} \mathbf{f}_i = b \sum_{i=1}^{4} \mathbf{ω}_i^2
\]

The rotor dynamics can be express as
\[ I_i \omega_i = \tau_i - Q_i, \quad (i = 1, 2, 3, 4) \]  

(7)

where \( I_i \) and \( \omega_i \) denote the moment of inertia and the speed of the rotor \( i \) respectively, \( \tau_i \) is the electrical torques of DC motor, and \( Q_i \) is the reactive torque caused by air drag and given by

\[ Q_i = k \omega_i^q \]  

(8)

where the parameters \( b \) and \( k \) are some positive constant relative to air density and the shape of the blade, et al. Under assumption A3, the total inertial matrix \( I \in \mathbb{R}^{3 \times 3} \) is a symmetric positive definite constant matrix expressed in the frame \( A \). The vector \( G_a \) denotes the gyroscopic torque vector and is given by

\[ G_a = \sum_{i=1}^{4} I_i (\Omega \times \omega_i) (-1)^i \omega_i \]  

(9)

If the distance from the rotors to the center of mass is denoted by \( l \), the control torque generated by the four rotors is

\[ \tau_a = \begin{pmatrix} \tau_{ar}^1 \\ \tau_{ar}^2 \\ \tau_{ar}^3 \\ \tau_{ar}^4 \end{pmatrix} = \begin{pmatrix} bl \left( \omega_i^2 - \omega_j^2 \right) \\ bl \left( \omega_i^2 - \omega_j^2 \right) \\ k \left( \omega_i^2 + \omega_j^2 - \omega_k^2 - \omega_l^2 \right) \end{pmatrix} \]  

(10)

In order to facilitate the computation for real control input, we put Eq. (6) and Eq. (10) together:

\[ \begin{bmatrix} \tau_{ar}^1 \\ \tau_{ar}^2 \\ \tau_{ar}^3 \\ \tau_{ar}^4 \end{bmatrix} = \begin{bmatrix} b & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & b & 0 \\ 0 & 0 & 0 & k \end{bmatrix} \begin{bmatrix} \omega_i^2 + \omega_j^2 + \omega_k^2 + \omega_l^2 \\ \omega_i^2 - \omega_j^2 \\ \omega_i^2 - \omega_j^2 \\ \omega_i^2 + \omega_j^2 - \omega_k^2 - \omega_l^2 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ -bl & 0 & bl & 0 \\ -bl & 0 & bl & 0 \\ -k & -k & -k & k \end{bmatrix} \begin{bmatrix} \omega_i^2 \\ \omega_j^2 \\ \omega_k^2 \\ \omega_l^2 \end{bmatrix} \]  

(11)

C. Equilibrium Points of the Quadrotor Mini-Helicopter

According to Eq. (1)-(2) and Eq. (4)-(5), the equilibrium state of the quadrotor satisfies

\[ \begin{cases} \dot{p} = v = 0 \\ \dot{R} = R\tilde{S}(\Omega) = 0 \\ m \dot{v} = -mgz + TRz = 0 \\ I_i \dot{\Omega} = -\Omega \times I_i \Omega - G_a + \tau_a = 0 \end{cases} \]  

(12)

From the first and second equation, we easily have

\[ \begin{align*} v &= 0 \\ \Omega &= 0 \end{align*} \]

Substituting \( v = \Omega = 0 \) into the third and fourth equation, we obtain

\[ \begin{cases} \Theta = (0, 0, \varphi)^T \\ T = mg \\ \tau_a = 0 \end{cases} \]

where \( \varphi \) are any constants. Therefore the quadrotor equilibrium state is

\[ \begin{cases} \Theta_e = (0, 0, \varphi)^T \\ p_e = (\bar{x}, \bar{y}, \bar{z})^T \end{cases} \]

Besides, combining previous equation

\[ \begin{cases} T = mg \\ \tau_a = 0 \end{cases} \]

with Eq. (11), we have
III. Trajectory Tracking Control Design Methodology

In consideration of under-actuated and strong coupled properties of a quadrotor, this paper proposes two control design methodologies respectively, as shown in Figure 2: The scheme a) consists of inner attitude stabilization loop and outer trajectory tracking loop controller under two-time scale separation assumption; b) employs command filtered backstepping technique to track the attitude command signal produced by trajectory tracking controller and uses linear tracking-differentiator to extract the attitude command derivative signals required by the backstepping control law without inner/outer loop structure. The main difference between these two schemes lies in the introduction of a low pass filter to compensate the dynamics of commanded derivative signal in the attitude stabilization control law calculation, while the trajectory tracking controller design method is the same.

A. Trajectory Tracking Control Design

Define the position error

\[ \mathbf{e}_p = \mathbf{p}_c - \mathbf{p} \]

where the vector \( \mathbf{p}_c = (x_c, y_c, z_c)^T \) is the position command value. Then construct the position closed-loop equation as

\[ \dot{\mathbf{e}}_p + \mathbf{K}_1 \mathbf{e}_p + \mathbf{K}_2 \mathbf{p}_c = 0 \]

\[ (14) \]

where \( \mathbf{K}_1 \) and \( \mathbf{K}_2 \) are both positive definite matrix. According to Routh-Hurwitz criterion, position error \( \mathbf{p}_c \) converges to zero exponentially. Continuously, we can rewrite Eq. (14) as follows

\[ \dot{\mathbf{p}} = \mathbf{p}_c + \mathbf{K}_1 ( \dot{\mathbf{p}}_c - \dot{\mathbf{p}} ) + \mathbf{K}_2 ( \mathbf{p}_c - \mathbf{p} ) \]

\[ (15) \]

Define virtual control signal \( \mathbf{U} = \mathbf{p}_c - \mathbf{p} \) and from Eq. (4) we have

\[ \mathbf{U} = (U_1, U_2, U_3)^T \]
$$U = -g_z + \frac{1}{m} TRz$$

Move the gravity acceleration item $-g_z$ to the left-hand side and left multiply rotation matrix $R^T$ on both sides of the previous equation. Then we obtain

$$R^T(U + g_z) = \frac{1}{m} Tz$$

That is

$$
\begin{bmatrix}
\cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\
\sin \theta \cos \psi - \sin \psi \cos \phi & \sin \theta \sin \psi + \cos \psi \cos \phi & \cos \theta \sin \phi \\
\sin \theta \cos \psi + \sin \psi \sin \phi & \sin \theta \sin \psi \cos \phi - \cos \psi \sin \phi & \cos \theta \cos \phi
\end{bmatrix}
\begin{bmatrix}
U_1 \\
U_2 \\
U_3 + g
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
\frac{1}{m} T
\end{bmatrix}
$$

After simple algebraic computation, we obtain

$$U_1 \cos \theta \cos \psi + U_2 \cos \theta \sin \psi - (U_3 + g) \sin \theta = 0 \quad (17)$$

$$U_1 (\sin \theta \cos \psi \sin \phi - \sin \psi \cos \phi) + U_2 (\sin \theta \sin \psi \sin \phi + \cos \psi \cos \phi) + (U_3 + g) \cos \theta \sin \phi = 0 \quad (18)$$

$$U_1 (\sin \theta \cos \psi \cos \phi + \sin \psi \sin \phi) + U_2 (\sin \theta \sin \psi \cos \phi - \cos \psi \sin \phi) + (U_3 + g) \cos \theta \cos \phi = \frac{1}{m} T \quad (19)$$

Using the fact $\cos \theta \neq 0$ (considering the quadrotor physical property), divide both sides of Eq. (17) by $\cos \theta$ and we will get

$$\theta = \arctan\left(\frac{U_1 \cos \psi + U_2 \sin \psi}{U_1 + g}\right) \quad (20)$$

Eq. (19) $\sin \phi$ - Eq. (18) $\cos \phi$ yields

$$\frac{1}{m} T \sin \phi = U_1 \sin \psi - U_2 \cos \psi \quad (21)$$

In view of Eq. (16), following relationship can be obtained

$$\left(\frac{1}{m} Tz\right)^T \left(\frac{1}{m} Tz\right) = (U + g_z)^T (U + g_z)$$

which is equivalent to

$$U_1^2 + U_2^2 + (U_3 + g)^2 = \left(\frac{1}{m} T\right)^2 \quad (22)$$

Combining Eq. (22) with Eq. (21), we get

$$\phi = \arcsin\left(\frac{U_1 \sin \psi - U_2 \cos \psi}{\sqrt{U_1^2 + U_2^2 + (U_3 + g)^2}}\right) \quad (23)$$

To this end, Eq. (20) and Eq. (23) can be used to compute the required attitude command input to the inner loop, and the virtual control signal can be computed from Eq. (15). The closed-form expressions for pitch and roll attitudes are

$$\theta_\phi = \tan^{-1}\left(\frac{U_1 \cos \psi + U_2 \sin \psi}{U_1 + g}\right)$$

$$\phi_\psi = \sin^{-1}\left(\frac{U_1 \sin \psi - U_2 \cos \psi}{\sqrt{U_1^2 + U_2^2 + (U_3 + g)^2}}\right)$$

where $\psi_\phi$ is commanded yaw attitude. Besides, the required total lift generated by the four rotors can be calculated from Eq. (19):

$$T = m[U_1 (\sin \theta \cos \psi \cos \phi + \sin \psi \sin \phi) + U_2 (\sin \theta \sin \psi \cos \phi - \cos \psi \sin \phi) + (U_3 + g) \cos \theta \cos \phi]$$

B. Attitude Command Control Design

B1. Inner Loop Attitude Linear Control Scheme

Similar to trajectory tracking control design, first construct the attitude error closed-form equation:
where the vector \( \Theta_e := \Theta_e - \Theta \) denotes the attitude error and \( \Theta_e = (\phi_e, \theta_e, \psi_e) \) denotes the commanded attitudes. Matrix \( K_d \) and \( K_p \) are both positive, so the attitude error vector \( \Theta_e \) exponentially converges to zero. Substituting Eq. (3) and Eq. (5) into Eq. (24) yields the control torque:

\[
\tau_v = \Omega \times I, \Omega + G_a - I, W^{-1} W \Omega - I, K_d \Omega + I, W^{-1} K_p (\Theta_e - \Theta) + I, W^{-1} (\dot{\Theta}_e + K_d \Theta_e)
\]

(25)

where

\[
W = \begin{bmatrix}
0 & \phi \cos \phi \tan \theta + \theta \sin \phi \sec^2 \theta & -\dot{\phi} \sin \phi \tan \theta + \theta \cos \phi \sec^2 \theta \\
0 & -\phi \sin \phi & -\phi \cos \phi \\
0 & \phi \cos \phi \sec \theta + \dot{\theta} \sin \phi \tan \theta \sec \theta & -\dot{\phi} \sin \phi \sec \theta + \dot{\theta} \cos \phi \tan \theta \sec \theta
\end{bmatrix}
\]

(26)

B2 Command Filtered Backstepping Attitude Control Scheme

The inner loop control scheme requires the fast response of actuator and large inner-loop gain to guarantee stability. However the high inner-loop gain may saturate the control inputs or excite the unmodeled dynamics and therefore induce robustness problem. To avoid this disadvantage, command filtered backstepping approach is applied to the attitude command control design. Define attitude and angular rate following errors:

\[
Z_1 = \Theta - \Theta_c
\]

(27)

\[
Z_2 = \Omega - \Omega_c
\]

(28)

where \( \Theta_c \) and \( \Omega_c \) are commanded attitude and angular rate respectively. Let us consider the Lyapunov function candidate

\[
V_1 = \frac{1}{2} Z_1^T Z_1
\]

(29)

The derivative of the Lyapunov function \( V_1 \) along the trajectories of Eq. (3) is given by

\[
\dot{V}_1 = Z_1^T \dot{Z}_1 = Z_1^T (W \Omega - \Theta_e)
\]

where \( \det(W) = \sec \theta \neq 0 \) if \( \theta \neq (2k - 1)\pi / 2 \), \((k = 1, 2, 3, \ldots)\). Angular rate \( \Omega \) can be seen as virtual control signal and then extracted satisfying \( \dot{V}_1 < 0 \):

\[
\Omega_d = W^{-1} (\dot{\Theta}_e - \Gamma_1 Z_1)
\]

(30)

where \( \Omega_d \) is the desired angular rate and \( \Gamma_1 \) is a positive definite matrix. In order to avoid tediously taking time derivative of \( \Omega_d \), we employ a first-order command filter

\[
\dot{\Theta}_e = -\bar{\Gamma} (\Theta_e - \Omega_d)
\]

(31)

to track the desired angular rate \( \Omega_d \). The filter time constant matrix \( \bar{\Gamma} = \text{diag}\{t_1, t_2, t_3\} > 0 \) should be as large as possible to promise the fast tracking. In view of this influence caused by the command filter, a new vector \( \varepsilon \) is introduced to compensate the tracking error in the rigorous stability analysis:

\[
\dot{\varepsilon} = -\bar{\Gamma}_1 \varepsilon + W (\Theta_e - \Omega_d)
\]

(32)

So we can re-define the attitude tracking error as

\[
\dot{Z}_1 = Z_1 - \varepsilon - \Theta_e - \Theta_c
\]

(33)

Choose compound Lyapunov function candidate as

\[
V_2 = \frac{1}{2} Z_1^T Z_1 + \frac{1}{2} Z_2^T Z_2
\]

Taking time derivative of \( V_2 \) along vector field Eq. (3) and Eq. (5), we get
\begin{align}
\dot{V}_2 &= \dot{Z}_2^T \dot{Z}_2 + Z_2^T \dot{Z}_2 \\
&= \dot{Z}_2^T \left[ \dot{\Theta} - \dot{\Theta}_e - \dot{\hat{e}} \right] + Z_2^T \left[ -I_{ij}^T (\Omega \times I_j, \Theta) - I_{ij}^T \dot{G}_a + I_{ij}^T \tau_a - \dot{\Omega}_e \right] \\
&= Z_2^T \left[ W \dot{\Theta} - \Theta_e + \Gamma_e \dot{e} - W (\Omega - \dot{\Theta}_e) \right] + Z_2^T \left[ -I_{ij}^T (\Omega \times I_j, \Theta) - I_{ij}^T \dot{G}_a + I_{ij}^T \tau_a - \dot{\Theta}_e \right] \\
&= -Z_2^T \dot{\Gamma}_2 \dot{Z}_2 + Z_2^T \left[ -I_{ij}^T (\Omega \times I_j, \Theta) - I_{ij}^T \dot{G}_a + I_{ij}^T \tau_a - \dot{\Theta}_e + W^T \dot{Z}_1 \right].
\end{align}

If the control torque \( \tau_a \) is extracted as
\begin{align}
\tau_a &= (\Omega \times I_j, \Theta) + G_a + I_j \dot{\Theta}_e - I_j W^T \dot{Z}_1 - I_j \dot{\Theta}_e \dot{Z}_2 \\
&= (\Omega \times I_j, \Theta) + G_a - I_j \tilde{\Theta}_e + I_j \tilde{\Theta}^{-1} \dot{\Theta}_e - I_j \tilde{\Theta}^{-1} \dot{\Theta}_e - I_j W^T (\Theta - \Theta_e) - I_j \Gamma_1 (\Theta - \Theta_e) - I_j \Gamma_2 (\Omega - \Omega_e), \quad (34)
\end{align}

the time derivative of Lyapunov function candidate \( V_2 \) satisfies
\begin{align}
\dot{V}_2 < 0
\end{align}

Thus tracking errors \( \dot{Z}_1 \) and \( \dot{Z}_2 \) converge to zero exponentially. Combining Eq. (31) and Eq. (32) with singular perturbation theory\(^2\), we know that \( \Theta \) converges to \( \Theta_e \) ultimately. However, there exists commanded attitude derivative signal in the explicit expression of \( \tau_a \). Although the commanded attitude \( \Theta_e \) is known, analytically computing its derivative expression is quite boring and intricate. Here we adopt a linear tracking-differentiator to extract the derivative signal \( \dot{\Theta}_e \):
\begin{align}
\begin{align}
\dot{X}_1 &= X_2 \\
\dot{X}_2 &= -2AX_1 - A^2 (X_1 - \Theta_e) \quad (35)
\end{align}
\end{align}

where \( A = \text{diag} \{ \gamma_1, \gamma_2, \gamma_3 \} > 0 \). Similar to previous first-order filter, \( \gamma_i (i=1,2,3) \) ought to be larger enough to guarantee the fast tracking to commanded attitude derivative signal. Thus we can replace \( \dot{\Theta}_e \) with \( X_2 \):
\begin{align}
\tau_a &= (\Omega \times I_j, \Theta) + G_a - I_j \tilde{\Theta}_e + I_j \tilde{\Theta}^{-1} \dot{\Theta}_e - I_j \tilde{\Theta}^{-1} \dot{\Theta}_e - I_j W^T (\Theta - \Theta_e) - I_j \Gamma_1 (\Theta - \Theta_e) - I_j \Gamma_2 (\Omega - \Omega_e) \quad (36)
\end{align}

Command filtered backstepping attitude stabilization controller diagram is depicted as Figure 3.

**Figure 3.** Backstepping attitude command control block diagram.

IV. Control Algorithm

A. Algorithm One: Inner/Outer Loop Control Methodology

Step 1 Trajectory planning: Give the desired trajectory \( (x_i, y, z) = (x_i (\alpha(t)), y, z, (\alpha(t))) \) and desired yaw attitude \( \psi_e (\alpha(t)) \), where \( \alpha(t) \) is a function of time, and usually \( \alpha(t) = \tau \).

Step 2 Calculation of virtual control signal \( U \):
\begin{align}
\begin{bmatrix}
U_1 \\
U_2 \\
U_3
\end{bmatrix} &= \begin{bmatrix}
\dot{x}_c \\
\dot{y}_c \\
\dot{z}_c
\end{bmatrix} + \begin{bmatrix}
k_{d1} & 0 & 0 \\
0 & k_{d2} & 0 \\
0 & 0 & k_{d3}
\end{bmatrix} \begin{bmatrix}
\dot{x}_c - \dot{x} \\
\dot{y}_c - \dot{y} \\
\dot{z}_c - \dot{z}
\end{bmatrix} + \begin{bmatrix}
k_{p1} & 0 & 0 \\
0 & k_{p2} & 0 \\
0 & 0 & k_{p3}
\end{bmatrix} \begin{bmatrix}
x_c - x \\
y_c - y \\
z_c - z
\end{bmatrix}
\end{align}

Step 3 Calculation of commanded pitch and roll attitude:
\begin{align*}
\theta_c &= \tan^{-1}\left(\frac{U_1 \cos \psi_c + U_z \sin \psi_c}{U_1 + g}\right) \\
\phi_c &= \sin^{-1}\left(\frac{U_1 \sin \psi_c - U_z \cos \psi_c}{\sqrt{U_1^2 + U_z^2 + (U_1 + g)^2}}\right)
\end{align*}

Step 4 Calculation of real control input $T$ and $\tau_a$:
\[ T = m\left[U_1 (\sin \theta \cos \psi \cos \phi + \sin \psi \sin \phi) + U_z (\sin \theta \sin \psi \cos \phi - \cos \psi \sin \phi) + (U_1 + g) \cos \theta \cos \phi\right] \]
\[ \tau_a = \Omega \times I + G_a - I, W^{-1} W \Omega - I, K_e \Omega + I, W^{-1} K_e (\Theta - \Theta) \]

B. Algorithm Two: Command Filtered Backstepping Control Methodology
Step 4 Calculation of commanded angular rate $\Omega_c$ and filter error $\epsilon$:
\[ \Omega_c = -J \left(\Omega_c - \Omega_d\right) \]
\[ \epsilon = -K_\Omega \left[\epsilon + W \left(\Omega_c - \Omega_d\right)\right] \]

Step 5 Extract commanded attitude derivative signal $\dot{\Theta}_c$:
\[ \dot{X}_1 = X_2 \]
\[ \dot{X}_2 = -2 \lambda X_1 - A^T \left( X_1 - \Theta_c \right) \]
\[ \dot{\Theta}_c = X_3 \]

Step 6 Calculation of real control input $T$ and $\tau_a$:
\[ T = m\left[U_1 (\sin \theta \cos \psi \cos \phi + \sin \psi \sin \phi) + U_z (\sin \theta \sin \psi \cos \phi - \cos \psi \sin \phi) + (U_1 + g) \cos \theta \cos \phi\right] \]
\[ \tau_a = (\Omega \times I, \Omega) + G_a - I, \bar{T} \Omega - I, \tilde{W}^{-1} X_2 - I, \bar{T} \tilde{W}^{-1} \Gamma_1 (\Theta - \Theta_c) - I, W^T (\Theta - \Theta_c - \epsilon) - I, \Gamma_2 (\Omega - \Omega_c) \]

V. Numerical Simulation

In order to verify the effectiveness of the controllers proposed in this paper, several simulations are performed on Simulink using data taken from Ref. 3, as given in Table 2. Here we will ignore the dynamics of DC motor temporarily and only make comparison of the two design methods. The controller parameters of algorithm A and B in simulations are fixed at $K_1 = \text{diag}\{2, 2, 2\}$, $K_2 = \text{diag}\{1, 1, 1\}$, $K_\phi = \text{diag}\{20, 20, 20\}$, $K_d = \text{diag}\{100, 100, 100\}$, $\Gamma_1 = \text{diag}\{0.4, 0.4, 0.4\}$, $\Gamma_2 = \text{diag}\{0.1, 0.1, 0.1\}$, and the filter time constant matrix is chosen as $\bar{T} = A = \text{diag}\{20, 20, 20\}$ in Algorithm B. In consideration of the limitations of real measuring device, the sampling time is fixed to $\Delta t = 0.01\text{sec}$. The initial positions and Euler angles are $p_0 = (0, 0, 0)^T$, $\Theta_0 = (0, 0, 0)^T$ respectively, so are linear velocities and angular rates respectively. The considered desired trajectories are a vertical and a horizontal rectangle and given by
\[ x_c = t \text{fsg}(t, 0, 20) + 20 \text{fsg}(t, 20, 40) + (60 - t) \text{fsg}(t, 40, 60) \]
\[ y_c = 0 \]
\[ z_c = 5 \text{fsg}(t, 20, 60) \]
and
\[ x_c = t \text{fsg}(t, 0, 20) + 20 \text{fsg}(t, 20, 40) + (60 - t) \text{fsg}(t, 40, 60) \]
\[ y_c = (t - 20) \text{fsg}(t, 20, 40) + 20 \text{fsg}(t, 40, 60) \]
\[ z_c = 0 \]
where $\text{fsg}$ denotes an interval function and is expressed as

American Institute of Aeronautics and Astronautics
\[ f_{sg}(x, a, b) = \frac{\text{sign}(x - a) - \text{sign}(x - b)}{2} \]

Simulation results are shown in Figure 4 - Figure 7: Figure 4 and Figure 6 depict the time histories of Euler angles, control inputs and trajectories using classical inner/outer-loop algorithm A, while Figure 5 and Figure 7 show the results using command filter backstepping algorithm B. As expected, both design methodologies make the quadrotor track the desired trajectories in a satisfactory way and even the transient response is almost the same. However, the control gain matrices of the inner loop are much larger than those of backstepping approach without time scale separation.

Table 2 Quadrotor Unmanned Mini-Helicopter Model Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>( m )</th>
<th>( g )</th>
<th>( l )</th>
<th>( I_x )</th>
<th>( I_y )</th>
<th>( I_z )</th>
<th>( b )</th>
<th>( k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>0.468</td>
<td>9.81</td>
<td>0.225</td>
<td>( 3.357 \times 10^{-5} )</td>
<td>( 4.856 \times 10^{-3} )</td>
<td>( 8.801 \times 10^{-3} )</td>
<td>( 2.98 \times 10^{-6} )</td>
<td>( 1.14 \times 10^{-7} )</td>
</tr>
<tr>
<td>Units</td>
<td>kg</td>
<td>m/s²</td>
<td>m</td>
<td>kg·m²</td>
<td>kg·m²</td>
<td>kg·m²</td>
<td>kg·m²</td>
<td>kg·m²</td>
</tr>
</tbody>
</table>

Figure 4. Vertical Rectangle Flight Using Algorithm A
Figure 5. Vertical Rectangle Flight Using Algorithm B

Figure 6. Horizontal Rectangle Flight Using Algorithm A
VI. Conclusions

This paper has presented two trajectory tracking design methodologies for a quadrotor unmanned mini-helicopter: classical inner/outer-loop PD control technique vs. command filtered backstepping technique. A main motivation is driven by the requirement of eliminating the two-time scale separation assumption and simplifying the computation of derivative signals, including the virtual control signal and the given commanded signal. Although the simulation results cannot tell the better one, by a comparison of these controller gain matrices we will easily come to the conclusion. In the first approach, we ignore the dynamics of $\dot{\Theta}$ and $\ddot{\Theta}$ by choosing large inner-loop gains, which may deteriorate the system performance, but the second one employs low-pass filter to track commanded signal derivatives $\dot{\Theta}$ and $\ddot{\Omega}$, and compensates these dynamics in the control law design and eliminates time scale separation assumption. Finally, numerical simulations of several typical trajectories of a quadrotor are performed to demonstrate the performance of the proposed controllers.

Acknowledgments

The authors sincerely acknowledge the motivating help from Prof. Huo Wei of the Seventh Research Division, Beihang University (formerly Beijing University of Aeronautics and Astronautics).

References


