Full attitude tracking control of a tiltrotor UAV on SO(3)

Shimeng Dong¹ and Tao Zhang²

¹ Xi’an Modern Control Technology Research Institute. Xi’an, 710065, China.
² Xi’an United Aircraft Intelligent Equipment Research Institute. Xi’an, 710086, China.

E-mail: 775180265@qq.com

Abstract. This paper presents novel results for the attitude tracking control of a tiltrotor unmanned aerial vehicle (UAV). This kind of UAV has two tiltable rotors while both thrust force and thrust direction can be controlled, and thus has four input degrees of freedom. A nonlinear dynamic model defined on SO(3) is proposed and analysed. An attitude tracking controller is developed using Lie Group and Lie Algebra theory, achieving smooth attitude tracking and avoiding the singularity problem of Euler Angles. Numerical simulation shows the controller has desirable closed loop performance and global asymptotic stability.

1. Introduction

In the last decade, considerable contribution from robotic and aeronautic research community has been made on geometric control of multirotor Unmanned Aerial Vehicles (UAV). Conventional control algorithm do not treat UAV attitude control as a geometric tracking problem because Euler Angles are used to parameterize the rotation, when yaw/pitch/roll axis could only be controlled separately with each reference angle commands [1-3]. Geometric Control takes the manifold structure of rotation into account, so a smooth movement trajectory on SO(3) can be realized and, at the same time, solving gimbal lock problem of Euler Angle control method [4-6].

However, despite the substantial interest in multirotor UAV control, little has been mentioned about constructing nonlinear controllers for tiltrotor ones, which occupy both vertical take-off and landing (VTOL) capability and high speed capability. Classical linear methods are used to design control law for tiltrotor aircraft, where nonlinear dynamics are only used in the final verifying simulation [7,8].

In this paper, a geometric attitude control algorithm is designed for a tiltrotor UAV on SO(3). The rotation dynamic of UAV is analyzed and a globally defined dynamic model is established on the manifold of the special orthogonal group SO(3). A nonlinear controller is constructed to track predefined rotation command. It is shown that the designed controller guarantees global exponential attractiveness to the zero equilibrium of tracking errors.

This paper is organized as follows. First a nonlinear tiltrotor UAV rotation dynamics model is defined and analysed in Section 2. The whole control law design is discussed in Section 3. Numerical simulations are conducted and results are shown in Section 4. Finally we conclude this paper and discuss possible direction of future work in Section 5.

2. Tiltrotor Dynamics Model

Following frames are presumed: an inertial reference frame \{e_x, e_y, e_z\} and a body-fixed frame \{b_x, b_y, b_z\}. The tiltrotor UAV configuration discussed in this paper is illustrated in figure 1. This
configuration includes two tiltable rotors, with separately controllable tilt angles ($a_1^t, a_2^t$) around the body lateral axis $b_y$ and separately controllable thrust force along the rotor axis ($f_1, f_2$). The two thrust-applied points both lie on the axis $b_y$ with a distance $R_b$ from the body-fixed axis origin. The center of mass of the UAV is located at point $(0, 0, D_{com})$ in body-fixed frame, while rotor tilt angles are assumed to have no effect on the center-of-mass location.

The basic idea of this configuration is illustrated in figure 2 and it is shown that the moments of three axis in body-fixed frame can be directly controlled by four available control inputs ($a_1, a_2, f_1, f_2$). Choosing down force in body-fix axis ($f_2$) as an augmented control goal, following maps are defined: $(f_2, M_x, M_y, M_z) \triangleq \text{ForwardMap}(a_1, a_2, f_1, f_2)$ and $(a_1, a_2, f_1, f_2) \triangleq \text{ReverseMap}(f_2, M_x, M_y, M_z)$. The ForwardMap is calculated using equation (2) and (4), while ReverseMap is its inverse function.

![Figure 1. Tiltrotor UAV configuration.](image1)

![Figure 2. Basic principle to generate moments.](image2)

Following assumptions are made about the dynamic model. Thrust of each propeller is controlled by thrust command and a transfer function $\text{THR}(s)$ is assumed to be the actuation dynamic. The left propeller rotate clockwise and right propeller rotate counterclockwise and the torque generated by each propeller is directly proportional to its thrust. Rotor tilting is realized by two rotatory actuator with a transfer function $\text{TILT}(s)$ to be the actuation dynamic. Rotor tilting actuation is assumed to generate no torque to the UAV body.

$$\text{THR}(s) = \frac{1}{0.02s + 1}, \quad \text{TILT}(s) = \frac{1}{0.04s + 1}$$

(1)

The total control force in body-fixed frame is calculated as following:

$$F = F_1 + F_r = \left[ f_1 \cos(a_1) + f_r \cos(a_r) \quad 0 \quad -f_1 \sin(a_1) - f_r \sin(a_r) \right]^T$$

(2)

The total moment in body-fixed frame is calculated as following:

$$M_1 = r_i \times F_1 - k_m F_1, \quad r_i = \begin{bmatrix} 0 & -R_b & -D_{com} \end{bmatrix}^T$$

$$M_r = r_i \times F_r + k_m F_r, \quad r_i = \begin{bmatrix} 0 & R_b & -D_{com} \end{bmatrix}^T$$

$$M = M_1 + M_r = \begin{bmatrix} R_b f_1 \sin(a_1) - R_b f_r \sin(a_r) + f_r k_m \cos(a_1) - f_1 k_m \cos(a_r) \\ -D_{com} f_1 \cos(a_1) - D_{com} f_r \cos(a_r) \\ f_r k_m \sin(a_1) - f_r k_m \sin(a_r) + R_b f_1 \cos(a_1) - R_b f_r \cos(a_r) \end{bmatrix}$$

(3)

(4)

The equations of rotation motion can be written as:

$$\begin{cases} \dot{R} = R \hat{\omega}_b \\ J \hat{\omega}_b = \omega_b \times J \omega_b = M \end{cases}$$

(5)
3. Attitude Tracking Control Law on SO(3)

3.1. Lie Group Theory

A Lie group is a smooth manifold whose elements satisfy the group axioms. Lie groups possess the local properties of smooth manifolds, which help us to define some concepts, such as error, distance, direction etc., for doing calculation on the manifold. SO(3) (special orthogonal group) is the group of rotation matrices as a common example of Lie group.

Assume two coordinate frames: frame1, frame2 and corresponding rotation matrices \( R_1, R_2 \in \text{SO}(3) \). Vector \( V \) expressed in two frames are \( V_1, V_2 \in \mathbb{R}^3 \) and so \( V_2 = R_2^T R_1 V_1 \). Then \( \delta R = R_2^T R_1 \) can be defined as the displacement rotation matrix and \( \delta R \) actually describe the ‘rotation error’ between frame1 and frame2, as shown in figure 3. However, this ‘error’ still lies on SO(3) and so it cannot be directly used.

Each Lie group has its own associated Lie algebra, which is defined as the tangent space of Lie group’s manifold at the identity. Lie algebra, from an intuitive perspective, describe the local defined velocity of a point moving on a Lie group’s manifold. The Lie algebra is isomorphic to vector space and so it’s simple to be converted to vector space using hat map and vee map: (see appendix A) and then used for error calculation. For SO(3), the associated Lie algebra is \( \mathfrak{so}(3) = \{ S \in \mathbb{R}^{3 \times 3}; S^T = -S \} \).

\( \mathfrak{so}(3) \) is the set of skew-symmetric matrices and isomorphic to \( \mathbb{R}^3 \). For example, the Lie algebra of \( \delta R \) actually stands for a rotation vector \( \delta \phi \) locally defined in frame2. If seeing from frame2, we need an incremental rotation \( \delta \phi \) to rotate to frame1 (just rotate the frame itself, but not any vector). The relationship between SO(3) and \( \mathfrak{so}(3) \) is defined by the exponential map and logarithmic map (see appendix B).

![Figure 3. Rotation error \( \delta R \) on SO(3) manifold and its associated Lie algebra \( \delta \phi \).](image)

3.2. The Whole Control Loop

Now for the control law design in this paper, the following are defined:

- \( R_c \in \text{SO}(3) \), the reference rotation matrix; \( f_c \in \mathbb{R} \), the reference force in down direction;
- \( R_f \in \text{SO}(3) \), the feedback rotation matrix; \( \omega_f^b \in \mathbb{R}^3 \), the feedback angular rate in body-fixed axis.
As shown in figure 4, a control law is designed to let UAV attitude track predefined reference attitude command $R_r(t)$ and reference thrust command $f_s(t)$, which often comes from trajectory planning algorithm in real application (specific trajectory planning algorithm is out of scope of this paper). The control law output control commands to the actuators to generate expected forces and moments to let UAV attitude $R_r(t)$ track the reference attitude command.

The control law structure is illustrated in figure 5. A cascade PI control structure is used, including attitude control as the outer loop and angular rate control as the inner loop.

![Control Law Diagram](image)

**Figure 5.** Cascade control law structure illustration.

3.3. **Attitude Controller Design**

First calculate the displacement rotation $e_R \in SO(3)$ between reference and feedback: $e_R = R_f^{-1} R_r$.

The Lie-Algebra of displacement rotation $e_{LA} \in so(3)$ is derived by applying logarithmic map on $e_R$: $e_{LA} = \log(e_R)$ and then converted to the local defined attitude error $e_\phi = R^{1/2}_L e_{LA}$ using vee map.

Three PI controllers (one for each axis) use attitude error to calculate compensation angular rate to fix attitude error: $e_{PI} = PI e_\phi$. The PI controller is gain scheduled as:

- **If** $|e_\phi| > 0.2 \text{ rad} : K_p = K_{PI1}$ (larger), $K_i = 0$ and integrator is reset to 0;
- **Else** : $K_p = K_{PI2}$ (smaller), $K_i = K_{IR}$ and integrator is activated.

The angular rate of reference rotation $\omega_r$ can be estimated from $R_r$ using equation (6), possible filters could be used here to for noise suppression according to signal quality of $R_r$.

$$\omega_r(k) = \frac{\log(R^{-1}_r(k) R_r(k + 1))}{T_{sample}}$$  \hspace{1cm} (6)

Transform $\omega_r$ to body-fixed axis via $e_R$ and take it as a feedforward compensation: $\omega_{ff} = e_R \omega_r$. The outer loop outputs a final reference angular rate $\omega_{r}^{b} = \omega_{ff} + \omega_{PI}^{b}$.

3.4. **Angular Rate Controller Design**

The inner loop possess a similar feedforward-feedback combined structure. The angular rate reference $\omega_{r}^{b}$ is directly used to calculate angular rate error: $e_{ob} = \omega_{r}^{b} - \omega_{b}^{b}$. Three PI controllers ($K_{pos}, K_{io}$) are used to calculate $M_{PI}$ to compensate for angular rate error as a feedback fix term. The angular rate reference $\omega_{r}^{b}$ and its estimated derivative $\dot{\omega}_{r}^{b}$ is used to generate a feedforward moment reference command $M_{ff} = J \omega_{r}^{b} + \omega_{r}^{b} \times J \omega_{b}^{b}$. A low-pass filter is recommended to calculate the derivative:

$$\dot{\omega}_{r}^{b} = \frac{s}{0.02s + 1} \omega_{r}^{b}$$  \hspace{1cm} (7)

Finally the inner loop outputs a moment reference $M_r$. 
4. Numerical Simulation Result

Physical parameters of UAV assumed: \( m = 1\, \text{kg}, J = \text{diag}(0.01, 0.01, 0.01)\, \text{kgm}^2, k_m = 0.015, R_0 = 0.2\, \text{m}, D_{\text{com}} = 0.1\, \text{m}. \) Control parameters tuned: \( K_{pR1} = 3.2, K_{pR2} = 1.68, K_{iR} = 0.34, K_{p\omega} = 0.064, K_{i\omega} = 0.051. \) Noise (\( 1\sigma = 0.01\, \text{rad/s} \)) is applied at sampled angular rate feedback. Two test cases are set:

4.1. Recover From Upside down Initial Attitude.

4.2. Track Sinusoid Reference Attitude Under Disturbance Moments Applied Condition.
5. Conclusions and Future Work
In this paper a novel nonlinear control algorithm designed for tiltrotor UAV attitude tracking on SO(3) is presented. We combine the geometric characteristic of SO(3) manifold with a cascade control loop structure and the simulation results show that it gains almost global asymptotic stability, smooth tracking behavior and disturbance rejection capability.

This paper mainly concentrates on attitude control, while in future work path planning and position/velocity control of tiltrotor UAVs on SE(3) will be discussed. Despite numerical simulations, experiments on real hardware are also on the future work schedule.

Appendix A
Hat map $\hat{\omega} : \omega \in \mathbb{R}^3 \rightarrow \Omega \in \mathfrak{so}(3)$. $\Omega=\omega \hat{\omega} \triangleq \left[ \begin{array}{ccc} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{array} \right]$. 

Vee map $\omega^\wedge : \mathfrak{so}(3) \rightarrow \mathbb{R}^3$. Inverse map of hat map. $\omega= (\omega^\wedge)^\wedge$.

Appendix B
Exponential map $\exp(\bullet) : \Phi \in \mathfrak{so}(3) \rightarrow R \in \text{SO}(3), \ R=I+\sin \phi u^\wedge + (1-\cos \phi)(u^\wedge)^\wedge$. $\Phi=(\phi u)^\wedge$ and $u$ is an unit vector.

Logarithmic map $\log(\bullet) : R \in \text{SO}(3) \rightarrow \Phi \in \mathfrak{so}(3), \ \Phi=(\phi u)^\wedge, \ \phi = \arccos \left( \frac{\text{trace}(R)-1}{2} \right), \ u = \left( R - R^T \right)^\wedge / 2 \sin \phi$. 

References
[1] Erdinc A, James P O and Robert M. 2002 Proc. Int. Conf. on Robotics and Automation (Washington) 1 72-7
[2] Samir B, Pierpaolo M and Roland S. 2004 Proc. Int. Conf. on Robotics and Automation (New Orleans) 5 4393-8
[3] Samir B and Roland S 2007 Proc. Int. Conf. on Intelligent robots and systems (San Diego) 153-8
[4] Francesco B and Richard M 1995 In European Control Conf. 2 1091-7
[5] Taeyoung L, Melvin L and N. Harris M 2010 Proc. Int. Conf. on Decision and Control 5420-5
[6] Yun Y, Shuo Y, Mingxi W, Cheng L and Zexiang L 2015 Proc. Int. Conf. on Robotics and Automation (Seattle) pp 1698-703
[7] Jing Z, Liguos S, Xiangju Q and Liuping W 2018 Chinese Journal of Aeronautics 31 632-42
[8] Zhen H, Chengzhi C, Jingkai Z 2018 Proc. Chinese Control Conf. (Wuhan) pp 9962-7
[9] David C, Rita C and Carlos S 2008 Proc. Conf. on Decision and Control(Cancun) pp 3583-8
[10] Samir B, Roland S 2005 Proc. Int. Conf. on Robotics and Automation (Barcelona) pp 2259-64
[11] RongHua L and Li C 2019 Defence Technology 15 964
[12] Yao L, Chaoyang D and Qing W 2017 Defence Technology 13 1
[13] Jiguang L, Xin C, Yajuan L and Rong Z 2017 Defence Technology 13 397
[14] Yu W, Xiaoming W and Jiyan Y 2018 Defence Technology 14 709
[15] Leonardo A 2017 Defence Technology 13 360
[16] Jiao M, Zhufeng S, Liwen G, Fugui X and Xiaqiang T 2017 Robotics and Computer-Integrated Manufacturing 46 48-57
[17] ZhuFeng S, Xiaqiang T and Liping W 2015 International Journal of Advanced Robotic Systems IJARS12(2015) p 144