## Title

Quaternion-based Trajectory Tracking Control of VTOL-UAVs using Command Filtered Backstepping



## Authors

Zhao, Sheng
Dong, Wenjie
Farrell, Jay A

## Publication Date

2013-06-01

Please cite the ACC 2013 version of this paper rather than this technical report:
S. Zhao, W. Dong, and J. A. Farrell, "Quaternion-based Trajectory Tracking COntrol of VTOL-UAVsusing Command Filtered Backstepping," American Control Conference, pp. 1018-1023, June 2013. DOI: 10.1109/ACC.2013.6579970

# Tech Report: Quaternion-based Trajectory Tracking Control of VTOL-UAVs using Command Filtered Backstepping

Sheng Zhao, Wenjie Dong, Jay A. Farrell

***Abstract*— This paper discusses trajectory tracking control for Vertical Take-Off and Landing (VTOL) Unmanned Aerial Vehicles (UAVs) using the command filtered backstepping technique. Quaternions are used to represent the attitude of the vehicle to ensure the global attitude tracking without singularities. Since the quaternions have their own unique algebra, they cannot be filtered by a vector-based command filter; therefore, a second-order quaternion filter is developed to filter the quaternion and automatically compute its derivative, which determines the commanded angular rate vector. A quadrotor vehicle is used as an example to show the performance of the proposed controller.**

## I. Introduction

Recently, Vertical Take-Off and Landing (VTOL) Unmanned Aerial Vehicles (UAVs) have gained tremendous interest among researchers and practitioners. In many applications, VTOL UAVs have advantages over fixed wing UAVs due to their relatively smaller size, capability to operate in cluttered environments and their hover capability. Many aerial vehicles fall into the categories of VTOL UAVs: Quadrotor, Coaxial rotorcraft and Ducted-fan UAV.

However, the control of VTOL UAV is not straightforward because of its under-actuated dynamics. The under-actuated dynamics require that translational motion of VTOL UAV be determined in part by the attitude of the vehicle. Since the translational motion of the VTOL UAV is determined by the attitude of the vehicle, the performance of the attitude tracking loop directly affects the performance of the position tracking loop. Euler-angles are commonly used in attitude control loop to represent the vehicle's attitude [4], [7]. However, the singularity problem of the Euler-angles prevents the controller to have the global tracking capability. Quaternion can represent the attitude without any singularity and thus a global attitude tracking controller can be implemented. In contrast to the rotation matrix representation, quaternion use fewer parameters (4 as opposed to 9) and quaternion is commonly used in the state estimators to represent the attitude. Hence, this paper uses quaternion to represent the attitude.

Regardless of how attitude is represented, the vehicle position trajectory tracking problem through the vehicle attitude involves strong nonlinearities. In many applications [4], [10] and commercial products [1], VTOL UAV control is implemented by linearizing the dynamics around a hover operating point and designing linear (e.g., PID-like) controllers to control the position and attitude of the vehicle. Such control design approaches have good performance near the linearized point (hover stage), but may perform poorly when the vehicle deviates away from the linearized point, which typically happens when tracking an user-specified trajectory. The trajectory commands considered in this paper can take two forms: 1) position only trajectory or 2) position and the desired yaw trajectory.

To improve the global VTOL tracking performance, various model-based nonlinear control methodologies can be considered. Among these nonlinear control methodologies, backstepping based design is widely adopted due to its systematic design and physically intuitive approach. In the backstepping design, the derivatives of virtual control signals are required in each design step. When the number of steps is greater than three, like the case of VTOL UAV, the analytic derivation of the derivatives becomes prohibitively complicated. When quaternions are involved in the control design, their special algebra and dynamics also complicates the design procedure. The paper [11] designs a quaternion-based backstepping controller to track a position trajectory and the required derivatives are computed analytically. The procedure is already very cumbersome for position tracking only. If the trajectory contains the desired yaw angle, all the derivatives computed in [11] need to be recomputed with respect to (w.r.t.) the desired yaw angle which makes the already tedious procedure even worse. There are many reasons that prevent the wide adoption of nonlinear controller in real applications, and the complexity involved in the design process is one. The command filtered implementation of the backstepping approach [3] maintains the desirable aspects of the backstepping method, has the same provable convergence properties, and simplifies the implementation process.

A command filtered backstepping trajectory tracking control approach for VTOL aircraft is presented and analyzed herein. The quaternion attitude representation requires extension of the approach represented in [3]. By exploiting the special dynamics of quaternion, this paper proposes a second-order quaternion filter to compute the commanded quaternion and its angular velocity, without differentiation. The proposed quaternion filter enables the command filtered backstepping design for the many types of vehicles utilizing quaternion-based attitude representations. Moreover, with the use of command filters, the flexibility of giving yaw commands in the trajectory is realized without further complicating the design process.

The paper is organized as follows. Section II reviews

This research is supported in part by "US ARMY grant W911NF-12-1-0094" S. Zhao and J. A. Farrell are with the Department of Electrical Engineering, University of California, Riverside, CA 92521 USA, {shzhao,farrell}@ee.ucr.edu

quaternion facts related to the modified definition. Section III presents the VTOL dynamics. Section IV presents the backstepping control design Section V presents the command filtered backstepping controller. Section VI presents the second-order quaternion filter. Section VII presents the stability analysis. Section VIII presents simulation results.

## II. Quaternion Review

In this section, we review the essential equations of quaternions for the controller design. The quaternion used in this paper follows the modified definition, instead of the Hamilton convention [13]. Derivations related to the quaternion algebra can be found in [15].

### A. Unit Quaternion

The quaternion $\bar{q} = [\boldsymbol{\epsilon}^\top \ \eta]^\top$ is a unit quaternion if it satisfies:

$$|\bar{q}| = \sqrt{\bar{q}^\top \bar{q}} = \sqrt{\|\boldsymbol{\epsilon}\|^2 + \eta^2} = 1. \tag{1}$$

The unit quaternion can be written as:

$$\bar{q} = \begin{bmatrix} \hat{\mathbf{k}} \sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix}. \tag{2}$$

In this notation, the unit vector $\hat{\mathbf{k}}$ describes the rotation axis and $\theta$ is the angle of rotation. Note that, one attitude can be represented by two quaternions: $\bar{q}$ and $-\bar{q}$. They differ by the rotating directions around the rotation axis to reach the target configuration.

### B. Quaternion Algebra

The skew-symmetric matrix $S(\mathbf{x})$ for any $\mathbf{x} \in \mathbb{R}^3$ is defined as $S(\mathbf{x}) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$. For any two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$, $\mathbf{x} \times \mathbf{y} = S(\mathbf{x})\mathbf{y}$.

The rotation matrix can be retrieved from the $\bar{q}$ as

$$\mathbf{R}(\bar{q}) = \mathbf{I}_{3\times 3} - 2\eta S(\boldsymbol{\epsilon}) + 2S(\boldsymbol{\epsilon})^2. \tag{3}$$

The multiplication of two quaternions is defined as

$$\bar{q} \otimes \bar{p} = \begin{bmatrix} \eta_q \boldsymbol{\epsilon}_p + \eta_p \boldsymbol{\epsilon}_q - \boldsymbol{\epsilon}_q \times \boldsymbol{\epsilon}_p \\ \eta_q \eta_p - \boldsymbol{\epsilon}_q^\top \boldsymbol{\epsilon}_p \end{bmatrix}. \tag{4}$$

Quaternion multiplication is distributive and associative, but not commutative. The multiplication of quaternions is analogous to multiplication of rotation matrix, in the same order:

$$\mathbf{R}(\bar{q})\mathbf{R}(\bar{p}) = \mathbf{R}(\bar{q} \otimes \bar{p}). \tag{5}$$

The inverse of a quaternion (analogous to the transpose of rotation matrix) is defined as $\bar{q}^{-1} = [-\boldsymbol{\epsilon}^\top \ \eta]^\top$. For any quaternion $\bar{q}$, we have

$$\bar{q} \otimes \bar{q}^{-1} = \bar{q}^{-1} \otimes \bar{q} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}. \tag{6}$$

Given a vector $\mathbf{p}$, we can form the quaternion as $\bar{q}_{\mathbf{p}} = [\mathbf{p}^\top \ 0]^\top$. The transformation of the representation of vector $\mathbf{p}$ from frame $a$ to frame $b$ using quaternions is

$${}^b\bar{q}_{\mathbf{p}} = {}^b_a\bar{q} \otimes {}^a\bar{q}_{\mathbf{p}} \otimes {}^b_a\bar{q}^{-1} \tag{7}$$

$$= \begin{bmatrix} {}^b_a\mathbf{R}\,{}^a\mathbf{p} \\ 0 \end{bmatrix}. \tag{8}$$

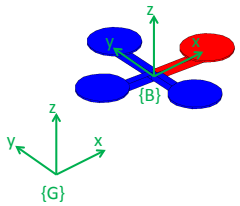


Fig. 1. The definitions of global and body frames

### C. Quaternion Dynamics

Let $\{G\}$ represent the inertial frame and $\{B\}$ represent the body frame. Let $\boldsymbol{\omega}^B_{GB}$ denote the angular velocity of $\{B\}$ w.r.t. $\{G\}$ expressed in $\{B\}$.

The dynamics of ${}^B_G\mathbf{R}$ and ${}^B_G\bar{q}$ are:

$${}^B_G\dot{\mathbf{R}} = -S(\boldsymbol{\omega}^B_{GB}){}^B_G\mathbf{R} \tag{9}$$

$${}^B_G\dot{\bar{q}} = \frac{1}{2}\bar{q}_{\boldsymbol{\omega}^B_{GB}} \otimes {}^B_G\bar{q} \tag{10}$$

$$= \frac{1}{2}\Phi({}^B_G\bar{q})\boldsymbol{\omega}^B_{GB} \tag{11}$$

where $\Phi = \begin{bmatrix} S(\boldsymbol{\epsilon}) + \eta\mathbf{I} \\ -\boldsymbol{\epsilon}^\top \end{bmatrix}$.

The dynamics of ${}^G_B\mathbf{R}$ and ${}^G_B\bar{q}$ are:

$${}^G_B\dot{\mathbf{R}} = {}^G_B\mathbf{R}S(\boldsymbol{\omega}^B_{GB}) \tag{12}$$

$${}^G_B\dot{\bar{q}} = \frac{1}{2}{}^G_B\bar{q} \otimes \bar{q}_{-\boldsymbol{\omega}^B_{GB}} \tag{13}$$

$$= \frac{1}{2}\begin{bmatrix} S(\boldsymbol{\epsilon}) - \eta\mathbf{I} \\ \boldsymbol{\epsilon}^\top \end{bmatrix}\boldsymbol{\omega}^B_{GB}. \tag{14}$$

The derivations of the rotation matrix dynamics can be found in Section 2.6.1 of [2]. The derivations of the dynamics of quaternion are in Section 2.4 of [15].

## III. System Dynamics

The rigid body dynamics of a VTOL UAV are:

$$\dot{\mathbf{p}} = \mathbf{v}, \qquad \dot{\mathbf{v}} = \tfrac{F}{m}{}^G_B\mathbf{R}\mathbf{e}_3 - g\mathbf{e}_3$$
$${}^B_G\dot{\bar{q}} = \tfrac{1}{2}\bar{q}_{\boldsymbol{\omega}^B_{GB}} \otimes {}^B_G\bar{q}, \quad \mathbf{J}\dot{\boldsymbol{\omega}}^B_{GB} = -\boldsymbol{\omega}^B_{GB} \times \mathbf{J}\boldsymbol{\omega}^B_{GB} + \boldsymbol{\tau}$$

where $\mathbf{e}_3 = [0 \ 0 \ 1]^\top$, $F$ is the collective force, $\boldsymbol{\tau}$ is the torque w.r.t. the center of gravity of the vehicle, $\mathbf{J}$ is the body-referenced inertia matrix, $m$ is the mass of the vehicle and $g$ the gravity constant. The definition of $\{G\}$-frame and $\{B\}$-frame is illustrated by using the example of a quadrotor in Fig. 1.

In this paper, the standard backstepping controller is derived first, including the attitude and force command extraction steps. Then command filters are designed to generate the required virtual control signals and their derivatives required at each step for implementation. This is done in a manner that maintains the desirable stability properties of the command filter implementation, relative to the standard backstepping approach, as discussed in [3].

## IV. Backstepping Control

This section derives the quaternion-based backstepping control laws for the position trajectory tracking task. This derivation is similar to the control laws presented in [11].

The subscript $c$ denotes the virtual control variable. The gains $\mathbf{K}_i \in \Re^{3\times 3}$, $i = 1...4$ are positive definite matrices.

***Step 1 (Position Control)***

Assume that the continuous signals $\mathbf{p}_d(t)$ and $\dot{\mathbf{p}}_d(t)$ are given. The goal in this step is to define a signal $\mathbf{v}_c(t)$ that causes $\mathbf{p}(t)$ to track the desired position $\mathbf{p}_d(t)$. The position tracking error dynamics are

$$\dot{\tilde{\mathbf{p}}} = \mathbf{v}_c + \tilde{\mathbf{v}} - \dot{\mathbf{p}}_d \tag{15}$$

where $\tilde{\mathbf{v}} = \mathbf{v} - \mathbf{v}_c$ and $\tilde{\mathbf{p}} = \mathbf{p} - \mathbf{p}_d$. Let the Lyapunov function be

$$V_1 = \frac{1}{2}\tilde{\mathbf{p}}^\top \tilde{\mathbf{p}}. \tag{16}$$

The time derivative of $V_1$ along solutions of eqn. (15) is

$$\dot{V}_1 = \tilde{\mathbf{p}}^\top (\mathbf{v}_c + \tilde{\mathbf{v}} - \dot{\mathbf{p}}_d). \tag{17}$$

Choosing $\mathbf{v}_c = -\mathbf{K}_1\tilde{\mathbf{p}} + \dot{\mathbf{p}}_d$, yields $\dot{V}_1 = -\tilde{\mathbf{p}}^\top \mathbf{K}_1 \tilde{\mathbf{p}} + \tilde{\mathbf{p}}^\top \tilde{\mathbf{v}}$.

***Step 2 (Velocity Control)***

Assume that the continuous signals $\mathbf{v}_c(t)$ and $\dot{\mathbf{v}}_c(t)$ are given. The goal in this step is to define a signal $\boldsymbol{\mu}_c(t)$ that causes $\mathbf{v}(t)$ to track the desired velocity $\mathbf{v}_c(t)$. The velocity tracking error dynamic is

$$\dot{\tilde{\mathbf{v}}} = \frac{F}{m}\,{}^G_B\mathbf{R}\,\mathbf{e}_3 - g\mathbf{e}_3 - \dot{\mathbf{v}}_c \tag{18}$$

$$= \boldsymbol{\mu}_c + \tilde{\boldsymbol{\mu}} - g\mathbf{e}_3 - \dot{\mathbf{v}}_c \tag{19}$$

where $\boldsymbol{\mu} = \frac{F}{m}\,{}^G_B\mathbf{R}\,\mathbf{e}_3$, $\boldsymbol{\mu}_c$ is the desired value of $\boldsymbol{\mu}$, and $\tilde{\boldsymbol{\mu}} = \boldsymbol{\mu} - \boldsymbol{\mu}_c$. The desired force $F_c$ and the desired attitude ${}^G_{B^c}\bar{q}$ will be determined from $\boldsymbol{\mu}_c$ at Step 3. Let the Lyapunov function be

$$V_2 = V_1 + \frac{1}{2}\tilde{\mathbf{v}}^\top \tilde{\mathbf{v}}. \tag{20}$$

Then the time derivative of $V_2$ is

$$\dot{V}_2 = \dot{V}_1 + \tilde{\mathbf{v}}^\top (\boldsymbol{\mu}_c + \tilde{\boldsymbol{\mu}} - g\mathbf{e}_3 - \dot{\mathbf{v}}_c). \tag{21}$$

Choosing

$$\boldsymbol{\mu}_c = -\mathbf{K}_2\tilde{\mathbf{v}} + g\mathbf{e}_3 + \dot{\mathbf{v}}_c - \tilde{\mathbf{p}}, \tag{22}$$

yields $\dot{V}_2 = -\tilde{\mathbf{p}}^\top \mathbf{K}_1 \tilde{\mathbf{p}} - \tilde{\mathbf{v}}^\top \mathbf{K}_2 \tilde{\mathbf{v}} + \tilde{\mathbf{v}}^\top \tilde{\boldsymbol{\mu}}$. In this paper, we assume $\|\boldsymbol{\mu}_c\| > 0$. This is typically the case, because if $\boldsymbol{\mu}_c = \boldsymbol{\mu} = \mathbf{0}$, then the system is in freefall.

***Step 3 (Attitude & Force Extraction)***

The total applied thrust vector $\boldsymbol{\mu}$ is determined by the force $F$ and the attitude ${}^B_G\bar{q}$: $\boldsymbol{\mu}(F, {}^B_G\bar{q}) = \frac{F}{m}\,{}^G_B\mathbf{R}\,\mathbf{e}_3$. The desired value $\boldsymbol{\mu}_c$ defined in eqn. (22) is implemented through selection of a desired force $F_c$ and desired attitude ${}^B_G\bar{q}_c$ such that $\boldsymbol{\mu}(F_c, {}^B_G\bar{q}_c) = \boldsymbol{\mu}_c$. The desired force is $F_c = m\,\|\boldsymbol{\mu}_c\|$. Let $\bar{\boldsymbol{\mu}}_c = \boldsymbol{\mu}_c / \|\boldsymbol{\mu}_c\|$.

Attitude extraction is discussed below separately depending on whether the yaw angle trajectory is or is not specified.

***Case 1:*** When the yaw trajectory is specified, let the unit vector $\mathbf{p}_y$ be the desired yaw direction at $t$ and assume that $\mathbf{p}_y$ is not parallel with $\boldsymbol{\mu}_c$. The desired direction cosine matrix is ${}^G_B\mathbf{R}_c = [\mathbf{y} \times \bar{\boldsymbol{\mu}}_c,\ \mathbf{y},\ \bar{\boldsymbol{\mu}}_c]$, where $\mathbf{y} = \bar{\boldsymbol{\mu}}_c \times \mathbf{p}_y$ [5]. Therefore, ${}^B_G\mathbf{R}_c = {}^G_B\mathbf{R}_c^\top$, and ${}^B_G\bar{q}_c$ can be retrieved from ${}^B_G\mathbf{R}_c$ by using eqn. (D.15) in page 504 of [2] (see the faq).

***Case 2:*** When the yaw trajectory is not specified, the desired attitude is not unique, due to the freedom to choose the yaw angle. The attitude extraction in this case is denoted as ${}^B_G\bar{q}_c = \Xi(\boldsymbol{\mu}_c, \bar{q}_r)$ where we have introduced a reference $\{r\}$-frame and a rotation $\bar{q}_r \triangleq {}^r_G\bar{q}$ to accommodate this free variable. Define $\bar{q}_m \triangleq {}^r_B\bar{q}$, which rotates the $\{B\}$ frame to the $\{r\}$ frame. Lemma 1 is used to obtain $\bar{q}_m$ by setting $u = \mathbf{e}_3$ and $v = \mathbf{R}(\bar{q}_r)\bar{\boldsymbol{\mu}}_c$. Then the desired quaternion is computed by ${}^B_G\bar{q}_c = \bar{q}_m^{-1} \otimes \bar{q}_r$.

**Lemma 1.** *[11] Given two unit vectors $u$ and $v$ with $u \neq -v$, the unit quaternion $\bar{q}$ with minimal rotation angle $\theta$ that satisfies $\mathbf{R}(\bar{q}_m)u = v$ is given by*

$$\boldsymbol{\epsilon} = \sqrt{\frac{1}{2(1+u^\top v)}}\,S(v)u, \quad \eta = \sqrt{\frac{1+u^\top v}{2}} \tag{23}$$

**Note 1.** *The reference attitude ($\bar{q}_r$) could be chosen as the global frame (${}^G_G\bar{q}$), the current vehicle attitude (${}^B_G\bar{q}$) or the current desired attitude (${}^B_G\bar{q}_c$). If a design goal is to choose the desired attitude such that the yaw rotation is minimized, then choosing the global frame as the reference attitude could result in unnecessary yaw rotation at start-up. Through simulations, we find that using the current desired attitude (${}^B_G\bar{q}_c$) as the reference attitude results in smaller yaw rotation during the trajectory tracking than using the current vehicle attitude.*

***Step 4 (Attitude Control)***

Assume that the continuous signals $F_c$, ${}^B_G\bar{q}_c$ and $\bar{\boldsymbol{\omega}}$ are available. To simplify notation, we define $\bar{q} \triangleq {}^B_G\bar{q}$ and $\bar{q}_c \triangleq {}^B_G\bar{q}_c$, and let $\bar{\boldsymbol{\omega}} \triangleq \bar{\boldsymbol{\omega}}^{B^c}_{GB^c}$ represent the rotational velocity vector of $\bar{q}_c$ whose dynamic is defined in eqn. (10). Computation of $\bar{\boldsymbol{\omega}}$ is discussed in Section VI. The goal in this step is to choose $\boldsymbol{\omega}_c$ to ensure the attitude of vehicle ${}^B_G\bar{q}(t)$ tracks the desired attitude $\bar{q}_c(t)$.

The attitude tracking error $\tilde{q}$ is defined as

$$\tilde{q} = {}^B_G\bar{q} \otimes \bar{q}_c^{-1}. \tag{24}$$

Thus, using (5) we have

$$\tilde{q} = \bar{q} \otimes \bar{q}_c^{-1} \text{ and } \mathbf{R}(\tilde{q})\mathbf{R}(\bar{q}_c) = \mathbf{R}(\bar{q}). \tag{25}$$

With this definition of attitude tracking error, the dynamic of $\tilde{q}$ is derived in Appendix I to be

$$\dot{\tilde{q}} = \frac{1}{2}\Phi(\tilde{q})\left(\boldsymbol{\omega} - \mathbf{R}(\tilde{q})\bar{\boldsymbol{\omega}}\right), \tag{26}$$

$$= \frac{1}{2}\Phi(\tilde{q})(\boldsymbol{\omega}_c + \tilde{\boldsymbol{\omega}} - \mathbf{R}(\tilde{q})\bar{\boldsymbol{\omega}}), \tag{27}$$

where $\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} - \boldsymbol{\omega}_c$ and $\boldsymbol{\omega} \triangleq \boldsymbol{\omega}^B_{GB}$ to simplify the notation.

Let the Lyapunov function be

$$V_3 = V_2 + 2(1 - \tilde{h}\tilde{\eta}) \tag{28}$$

where $\tilde{\eta}$ is the scalar part of $\tilde{q}$, and $\tilde{h} \in \{-1, 1\}$ is a hybrid variable introduced in [8] that determines the convergence point of $\tilde{q}$ to either $[\mathbf{0}\ 1]^\top$ or $[\mathbf{0}\ -1]^\top$. The dynamics of $\tilde{h}$ contain both continuous and discontinuous parts:

$$\begin{cases} \dot{\tilde{h}} = 0, & \text{if} \quad \mathbf{x} \in C \\ \tilde{h}^+ = -\tilde{h}, & \text{if} \quad \mathbf{x} \in D \end{cases} \tag{29}$$

where $\tilde{h}^+$ denotes the discrete update of the variable, $\mathbf{x}$ is the controller state vector, $C$ is the flow set and $D$ is the jump set [9]. The flow set and the jump set will be defined in the next control step.

The time derivative of $V_3$ is

$$\dot{V}_3 = \dot{V}_3^s + \tilde{\mathbf{v}}^\top \tilde{\boldsymbol{\mu}} + \tilde{h}\tilde{\boldsymbol{\epsilon}}^\top(\boldsymbol{\omega}_c + \tilde{\boldsymbol{\omega}} - \mathbf{R}(\tilde{q})\bar{\boldsymbol{\omega}}) \tag{30}$$

where $\dot{V}_3^s = -\tilde{\mathbf{p}}^\top \mathbf{K}_1 \tilde{\mathbf{p}} - \tilde{\mathbf{v}}^\top \mathbf{K}_2 \tilde{\mathbf{v}}$. To proceed, we rewrite $\tilde{\boldsymbol{\mu}}$ as $\tilde{\boldsymbol{\mu}} = \mathbf{W}(\bar{q}_c, \tilde{q}, F_c)\tilde{\boldsymbol{\epsilon}}$ as derived in the Appendix II. This yields

$$\dot{V}_3 = \dot{V}_3^s + \tilde{\mathbf{v}}^\top \mathbf{W} \tilde{\boldsymbol{\epsilon}} + \tilde{h}\tilde{\boldsymbol{\epsilon}}^\top(\boldsymbol{\omega}_c + \tilde{\boldsymbol{\omega}} - \mathbf{R}(\tilde{q})\bar{\boldsymbol{\omega}}). \tag{31}$$

Choosing

$$\boldsymbol{\omega}_c = -\mathbf{K}_3 \tilde{h} \tilde{\boldsymbol{\epsilon}} - \mathbf{W}^\top \tilde{\mathbf{v}} + \mathbf{R}(\tilde{q})\bar{\boldsymbol{\omega}} \tag{32}$$

and noticing that $\tilde{h}^2 = 1$, we have

$$\dot{V}_3 = \dot{V}_3^s - \tilde{\boldsymbol{\epsilon}}^\top \mathbf{K}_3 \tilde{\boldsymbol{\epsilon}} + \tilde{h}\tilde{\boldsymbol{\epsilon}}^\top \tilde{\boldsymbol{\omega}}. \tag{33}$$

***Step 5 (Angular Velocity Control)***

In this step, the goal is choose $\boldsymbol{\tau}$ to cause the actual angular rate $\boldsymbol{\omega}(t)$ track the desired angular velocity $\boldsymbol{\omega}_c(t)$. We assume $\boldsymbol{\omega}_c$ and $\dot{\boldsymbol{\omega}}_c$ are available, continuous, and known.

The dynamic of the angular velocity tracking error $\tilde{\boldsymbol{\omega}}$ is

$$\begin{aligned} \mathbf{J}(\dot{\tilde{\boldsymbol{\omega}}} + \dot{\boldsymbol{\omega}}_c) &= -(\tilde{\boldsymbol{\omega}} + \boldsymbol{\omega}_c) \times \mathbf{J}(\tilde{\boldsymbol{\omega}} + \boldsymbol{\omega}_c) + \boldsymbol{\tau} \\ \mathbf{J}\dot{\tilde{\boldsymbol{\omega}}} &= -(\tilde{\boldsymbol{\omega}} + \boldsymbol{\omega}_c) \times \mathbf{J}(\tilde{\boldsymbol{\omega}} + \boldsymbol{\omega}_c) - \mathbf{J}\dot{\boldsymbol{\omega}}_c + \boldsymbol{\tau} \\ \mathbf{J}\dot{\tilde{\boldsymbol{\omega}}} &= \boldsymbol{\Sigma}(\tilde{\boldsymbol{\omega}}, \boldsymbol{\omega}_c)\tilde{\boldsymbol{\omega}} - S(\boldsymbol{\omega}_c)\mathbf{J}\tilde{\boldsymbol{\omega}} + \boldsymbol{\tau}_f(\boldsymbol{\omega}_c, \dot{\boldsymbol{\omega}}_c) + \boldsymbol{\tau} \end{aligned} \tag{34}$$

where

$$\boldsymbol{\Sigma}(\tilde{\boldsymbol{\omega}}, \boldsymbol{\omega}_c) = S(\mathbf{J}\tilde{\boldsymbol{\omega}}) + S(\mathbf{J}\boldsymbol{\omega}_c) \tag{35}$$

$$\boldsymbol{\tau}_f(\boldsymbol{\omega}_c, \dot{\boldsymbol{\omega}}_c) = S(\mathbf{J}\boldsymbol{\omega}_c)\boldsymbol{\omega}_c - \mathbf{J}\dot{\boldsymbol{\omega}}_c. \tag{36}$$

Note that $\boldsymbol{\Sigma}$ is a skew-symmetric matrix for any $\tilde{\boldsymbol{\omega}}$ and $\boldsymbol{\omega}_c$.

Choosing the Lyapunov function as

$$V_4 = V_3 + \tilde{\boldsymbol{\omega}}^\top \mathbf{J} \tilde{\boldsymbol{\omega}}, \tag{37}$$

yields the time derivative of $V_4$ as

$$\dot{V}_4 = \dot{V}_4^s + \tilde{h}\tilde{\boldsymbol{\epsilon}}^\top \tilde{\boldsymbol{\omega}} + \tilde{\boldsymbol{\omega}}^\top(\boldsymbol{\Sigma}\tilde{\boldsymbol{\omega}} - S(\boldsymbol{\omega}_c)\mathbf{J}\tilde{\boldsymbol{\omega}} + \boldsymbol{\tau}_f + \boldsymbol{\tau}) \tag{38}$$

where $\dot{V}_4^s = \dot{V}_3^s - \tilde{\boldsymbol{\epsilon}}^\top \mathbf{K}_3 \tilde{\boldsymbol{\epsilon}}$. Choosing

$$\boldsymbol{\tau} = -\mathbf{K}_4 \tilde{\boldsymbol{\omega}} - \tilde{h}\tilde{\boldsymbol{\epsilon}} + S(\boldsymbol{\omega}_c)\mathbf{J}\tilde{\boldsymbol{\omega}} - \boldsymbol{\tau}_f \tag{39}$$

and noticing that $\tilde{\boldsymbol{\omega}}^\top \boldsymbol{\Sigma} \tilde{\boldsymbol{\omega}} = 0$ yields

$$\dot{V}_4 = \dot{V}_4^s - \tilde{\boldsymbol{\omega}}^\top \mathbf{K}_4 \tilde{\boldsymbol{\omega}} \tag{40}$$

which is a negative definite function of the error state. The change in $V_4$ following a jump in $\tilde{h}$ can be shown, using eqns. (28), (32) and (37) and the procedure on page 2526 in [9], to be:

$$V_4(\tilde{h}^+) - V_4(\tilde{h}) = 2(1 - (-\tilde{h})\tilde{\eta}) \tag{41}$$

$$+(\boldsymbol{\omega} - \boldsymbol{\omega}_c(\tilde{h}^+))^\top \mathbf{J}(\boldsymbol{\omega} - \boldsymbol{\omega}_c(\tilde{h}^+)) - 2(1 - \tilde{h}\tilde{\eta}) \tag{42}$$

$$-(\boldsymbol{\omega} - \boldsymbol{\omega}_c(\tilde{h}))^\top \mathbf{J}(\boldsymbol{\omega} - \boldsymbol{\omega}_c(\tilde{h})) \tag{43}$$

$$= 4\tilde{h}(\tilde{\eta} - \tilde{\boldsymbol{\epsilon}}^\top \mathbf{K}_3^\top \mathbf{G}) \tag{44}$$

where $\mathbf{G} = \boldsymbol{\omega} + \mathbf{W}^\top \tilde{\mathbf{v}} - \mathbf{R}(\tilde{q})\bar{\boldsymbol{\omega}}$. Letting $\boldsymbol{\Lambda} = \tilde{\eta} - \tilde{\boldsymbol{\epsilon}}^\top \mathbf{K}_3^\top \mathbf{G}$, then the flow and jump sets can be defined as

$$C_1 = \{\tilde{h}\boldsymbol{\Lambda} \geq -\delta\} \tag{45}$$

$$D_1 = \{\tilde{h}\boldsymbol{\Lambda} \leq -\delta\} \tag{46}$$

where $\delta \in (0, 1)$ is a design variable that determines at what point to switch the convergence point. Implementation of this control law requires computation of the various command signals and their derivatives.

## V. Command Filtered Backstepping

The backstepping controller an signal requirements at each step are summarized below.

| Step | Rqrd. sig. | Control Law |
|---|---|---|
| 1 | $\mathbf{p}_d$, $\dot{\mathbf{p}}_d$ | $\mathbf{v}_c = -\mathbf{K}_1\tilde{\mathbf{p}} + \dot{\mathbf{p}}_d$ |
| 2 | $\mathbf{v}_c$, $\dot{\mathbf{v}}_c$ | $\boldsymbol{\mu}_c = -\mathbf{K}_2\tilde{\mathbf{v}} + g\mathbf{e}_3 + \dot{\mathbf{v}}_c - \tilde{\mathbf{p}}$ |
| 3 | $\boldsymbol{\mu}_c$ | $F = m\left\|\boldsymbol{\mu}_c\right\|$, $\bar{q}_c = \Xi(\boldsymbol{\mu}_c, \bar{q}_c)$ |
| 4 | $\bar{q}_c$, $\bar{\boldsymbol{\omega}}$ | $\boldsymbol{\omega}_c = -\mathbf{K}_3\tilde{h}\tilde{\boldsymbol{\epsilon}} - \mathbf{W}^\top\tilde{\mathbf{v}} + \mathbf{R}(\tilde{q})\bar{\boldsymbol{\omega}}$ |
| 5 | $\boldsymbol{\omega}_c$, $\dot{\boldsymbol{\omega}}_c$ | $\boldsymbol{\tau} = -\mathbf{K}_4\tilde{\boldsymbol{\omega}} - \tilde{h}\tilde{\boldsymbol{\epsilon}} + S(\boldsymbol{\omega}_c)\mathbf{J}\tilde{\boldsymbol{\omega}} - \boldsymbol{\tau}_f$ |

In the standard backstepping approach, the variables $\dot{\mathbf{v}}_c$, $\bar{\boldsymbol{\omega}}$ and $\dot{\boldsymbol{\omega}}_c$ are derived analytically. This is cumbersome as shown in [11], which only considered position tracking. For the position and yaw tracking, the analytic derivation would become even more cumbersome as all derivatives in [11] must re-computed to include the commanded yaw direction. Command filters are employed herein to automate the computation.

The command filtered backstepping controller is summarized below:

1) $\mathbf{v}_c^o = -\mathbf{K}_1\tilde{\mathbf{p}} + \dot{\mathbf{p}}_d$; $\quad [\mathbf{v}_c, \dot{\mathbf{v}}_c] = CF_1(\mathbf{v}_c^o)$.
2) $\boldsymbol{\mu}_c^o = -\mathbf{K}_2\tilde{\mathbf{v}} + g\mathbf{e}_3 + \dot{\mathbf{v}}_c - \tilde{\mathbf{p}}$.
3) $F_c = m\left\|\boldsymbol{\mu}_c^o\right\|$, $\bar{q}_c^o = \Xi(\boldsymbol{\mu}_c^o, \bar{q}_c)$; $\quad [\bar{q}_c, \bar{\boldsymbol{\omega}}] = QF(\bar{q}_c^o)$.
4) $\boldsymbol{\omega}_c^o = -\mathbf{K}_3\tilde{h}\tilde{\boldsymbol{\epsilon}} - \mathbf{W}^\top\tilde{\mathbf{v}} + \mathbf{R}(\tilde{q})\bar{\boldsymbol{\omega}}$; $[\boldsymbol{\omega}_c, \dot{\boldsymbol{\omega}}_c] = CF_2(\boldsymbol{\omega}_c^o)$.
5) $\boldsymbol{\tau} = -\mathbf{K}_4\tilde{\boldsymbol{\omega}} - \tilde{h}\tilde{\boldsymbol{\epsilon}} + S(\boldsymbol{\omega}_c)\mathbf{J}\tilde{\boldsymbol{\omega}} - \boldsymbol{\tau}_f$.

where $QF$ is the quaternion command filter discussed in Section VI, and $CF_1$ and $CF_2$ are command filters defined as

$$\dot{\mathbf{x}}_1 = \mathbf{x}_2 \tag{47}$$

$$\dot{\mathbf{x}}_2 = -\omega_n^2(\mathbf{x}_1 - \mathbf{u}) - 2\xi\omega_n\mathbf{x}_2 \tag{48}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are in $\Re^3$ , and $\mathbf{u} = \mathbf{v}_c^o$ for $CF_1$ and $\mathbf{u} = \boldsymbol{\omega}_c^o$ for $CF_2$. The outputs of the filters are $\mathbf{x}_1$ and $\mathbf{x}_2$. For example, in the case of $CF_1$, $\mathbf{v}_c(t) = \mathbf{x}_1(t)$, $\dot{\mathbf{v}}_c(t) = \mathbf{x}_2(t)$.

The analysis in [3] ensures that as $\omega_n$ is increased, the tracking error performance of the command filtered implementation approaches the performance of the backstepping controller using analytically derived command derivatives. To complete the derivation for this application, we must

present a quaternion command filter and show that it fits within the framework of [3].

## VI. Second-order Quaternion Filter

This section develops a second-order quaternion filter that takes the quaternion $\bar{q}_c^o(t)$ as the input and produces the filtered quaternion $\bar{q}_c(t)$ and the corresponding angular velocity $\bar{\boldsymbol{\omega}}(t)$ as outputs. The purpose of the filter is to ensure producing $\bar{\boldsymbol{\omega}}(t)$ without differentiation and to ensure that the error between $\bar{q}_c^o(t)$ and $\bar{q}_c(t)$, defined as

$$\tilde{\bar{q}} = \bar{q}_c \otimes (\bar{q}_c^o)^{-1}, \tag{49}$$

is small. The symbol $\tilde{\boldsymbol{\epsilon}}$ represents the vector part of $\overset{\hat{B}}{B}\tilde{\bar{q}}$. Both $\bar{q}_c^o(t)$ and $\bar{q}_c(t)$ are known and available at every time instant; hence, $\tilde{\bar{q}}$ and $\tilde{\boldsymbol{\epsilon}}$ can be computed at every time instant.

The proposed quaternion filter is

$$\dot{\bar{q}}_c = \frac{1}{2}\Phi(\bar{q}_c)\bar{\boldsymbol{\omega}} \tag{50}$$

$$\dot{\bar{\boldsymbol{\omega}}} = \alpha(-\tilde{k}\tilde{h}_f\tilde{\boldsymbol{\epsilon}} - \bar{\boldsymbol{\omega}}) \tag{51}$$

where $\tilde{h}_f$ is a hybrid variable as discussed in Section IV, and $\alpha,\ \tilde{k} \in \mathbb{R}_+$. The form of eqn. (50) is designed to maintain the unit norm property for $\bar{q}_c$. The form of eqn. (51) is designed to cause $\bar{q}_c(t)$ to track $\bar{q}_c^o(t)$. In this filter, the parameter $\alpha$ determines how fast $\bar{\boldsymbol{\omega}}$ tracks $-\tilde{k}\tilde{h}_f\tilde{\boldsymbol{\epsilon}}$ and $\tilde{k}$ determines how fast $\bar{q}_c$ tracks the input attitude $\bar{q}_c^o$.

The dynamic of $\tilde{\bar{q}}$ is (see Appendix I)

$$\dot{\tilde{\bar{q}}} = \frac{1}{2}\Phi(\tilde{\bar{q}})\tilde{\boldsymbol{\omega}} \tag{52}$$

where $\tilde{\boldsymbol{\omega}} = \bar{\boldsymbol{\omega}} - \mathbf{R}(\tilde{\bar{q}})\bar{\boldsymbol{\omega}}^o$, and $\bar{\boldsymbol{\omega}}^o$ is the angular rate of the $\bar{q}_c^o$ which is not available (i.e., unknown).

The stability of the filter is shown by considering the zero input case. Define the Lyapunov function $V = 2\alpha\tilde{k}(1 - \tilde{h}_f\tilde{\eta}) + \tilde{\boldsymbol{\omega}}^\top\tilde{\boldsymbol{\omega}}$, where $\tilde{\eta}$ is the scalar part of $\tilde{\bar{q}}$. Then the time derivative of $V$ can be computed as

$$\dot{V} = \alpha\tilde{k}\tilde{h}_f\tilde{\boldsymbol{\epsilon}}^\top\tilde{\boldsymbol{\omega}} + \tilde{\boldsymbol{\omega}}^\top(-\alpha\tilde{k}\tilde{h}_f\tilde{\boldsymbol{\epsilon}} - \alpha\bar{\boldsymbol{\omega}}) \tag{53}$$

$$= -\alpha\tilde{\boldsymbol{\omega}}^\top\tilde{\boldsymbol{\omega}} \leq 0. \tag{54}$$

The change in $V$ following the jump of $\tilde{h}_f$ is $V(\tilde{h}_f^+) - V(\tilde{h}_f) = 4\alpha\tilde{k}\tilde{h}_f\tilde{\eta}$. By defining the flow and jump sets of the filter to be

$$C_2 = \{\tilde{h}_f\tilde{\eta} \geq -\delta\} \quad \text{and} \quad D_2 = \{\tilde{h}_f\tilde{\eta} \leq -\delta\}, \tag{55}$$

it follows that $V(\tilde{h}_f^+) - V(\tilde{h}_f) < 0$. Since $\dot{V} = 0$ if only if $\tilde{\boldsymbol{\omega}} = 0$ and $\{\mathbf{x} \in D_2 : V(\tilde{h}_f^+) - V(\tilde{h}_f) = 0\} = \emptyset$, by Theorem 4.7 in [12] the filter is global asymptotically stable.

**Note 2.** *The vector command filter of eqns. (47-48) could also be used to produce $\bar{q}_c(t)$ and $\bar{\boldsymbol{\omega}}(t)$ from $\bar{q}_c^o(t)$. However, that filter would not maintain the quaternion having unit norm and would not necessarily track the input quaternion through the path of minimal rotation. In contrast, the hybrid variable $\tilde{h}_f$ in the proposed quaternion filter ensures the input quaternion is tracked through the path of minimal rotation. Moreover, the state vector in the proposed quaternion filter has smaller size than the one in the vector-based filter (7 as opposed to 8).*

**Note 3.** *Either a normalization step or the Lie group variational integrator, see e.g., [6], can be used to ensure the unit norm property of the quaternion in practice.*

## VII. Stability Analysis

The only difference between the proposed controller in this paper and the controller in [3] is the involvement of the quaternion filter. Therefore, this section shows that the quaternion filter can be written into the form of $\dot{\mathbf{z}} = \bar{\boldsymbol{\epsilon}}F(\cdot)$, which is required to apply the singular perturbation analysis as presented in [3]. Once this is shown, the stability of the proposed controller in this paper can be proved in an identical manner as was done in [3].

Eqn. (51) can be rewritten as

$$\dot{\bar{\boldsymbol{\omega}}} = -\alpha\tilde{k}(\tilde{h}_f\tilde{\boldsymbol{\epsilon}}) - \alpha\bar{\boldsymbol{\omega}}. \tag{56}$$

Comparing eqns. (48) and (56), we note that $\tilde{h}_f\tilde{\boldsymbol{\epsilon}}$ is equivalent to $(\mathbf{x}_1 - \mathbf{u})$ and $\bar{\boldsymbol{\omega}}$ is equivalent to $\mathbf{x}_2$. Hence we can let $\alpha\tilde{k} = \omega_n^2$ and $\alpha = 2\xi\omega_n$ to obtain

$$\omega_n = \sqrt{\alpha\tilde{k}}, \quad \xi = \frac{1}{2}\sqrt{\alpha/\tilde{k}}. \tag{57}$$

Thus, eqn. (56) can be written as

$$\dot{\bar{\boldsymbol{\omega}}} = -\omega_n^2(\tilde{h}_f\tilde{\boldsymbol{\epsilon}}) - 2\xi\omega_n\bar{\boldsymbol{\omega}}. \tag{58}$$

Letting $\mathbf{z}_1 = \bar{q}$ and $\mathbf{z}_2 = \bar{\boldsymbol{\omega}}/\omega_n$, we can rewrite the quaternion filter (eq. (50) and (51)) into the form

$$\begin{bmatrix}\dot{\mathbf{z}}_1\\ \dot{\mathbf{z}}_2\end{bmatrix} = \omega_n\begin{bmatrix}\frac{1}{2}\Phi(\mathbf{z}_1)\mathbf{z}_2\\ -\tilde{h}_f\tilde{\boldsymbol{\epsilon}} - 2\xi\mathbf{z}_2\end{bmatrix}$$

which has the desired form $\dot{\mathbf{z}} = \bar{\boldsymbol{\epsilon}}F(\mathbf{z}, \bar{q}_c)$, where $\bar{\boldsymbol{\epsilon}} = \omega_n$. Then, the singular perturbation theorem can be applied as in [3] to prove the stability of the proposed controller.

## VIII. Simulation Results

First, we demonstrate in Fig. 2 that the vector-based filter (red) may take a non-minimum angle path while tracking the input quaternion when the attitude difference is greater than 180 degrees. The vector-based filter uses eqns. (47-48) with $\mathbf{x}_1$ and $\mathbf{x}_2$ in $\Re^4$, and $\mathbf{u} = \bar{q}_c^o(t)$ is the input quaternion. The angular velocity is $\boldsymbol{\omega} = 2\Phi(\mathbf{x}_1)^\top\mathbf{x}_2$ based on eqn. (11) using the fact that $\Phi^\top\Phi = \mathbf{I}$. The initial quaternion is set to $[\mathbf{0}\ 1]^\top$ and the initial angular velocity is set to $\mathbf{0}$ for both the vector-based filter and the quaternion filter. The input quaternion is a constant value created by letting $\hat{\mathbf{k}} = [0\ 0\ 1]^\top$ and $\theta = 240^\circ$ in eqn. (2). Fig. 2 shows that both filters successfully track the input quaternion, but the vector-based filter (red) rotates counter-clockwise to reach the desired attitude while the proposed quaternion (blue) filter rotates clockwise which is a shorter route to reach the desired attitude.

In the second simulation, we use quadrotor as an example of a VTOL UAV to demonstrate the performance of the proposed tracking controller. The desired trajectory in this simulation is a position trajectory which is shown in blue in Fig. 3 with the actual trajectory of the vehicle in red.

The position tracking errors $\mathbf{p} - \mathbf{p}_d$ are plotted in Fig. 4. In this figure, we compare the performance of controller by setting a different $\alpha$ in the quaternion filter. We also compare the yaw rotation of the vehicle in the trajectory tracking by running the simulation separately with using the current desired attitude and the current vehicle attitude as the reference attitude. The comparison result given in Fig. 5 shows that using the current desired attitude results in lesser yaw rotation.

The parameters used in the simulation are: $m = 0.5$kg, $\mathbf{J} = diag([0.0820; 0.0845; 0.1377])$, $\mathbf{K}_1 = 2\mathbf{I}$, $\mathbf{K}_2 = \mathbf{I}$, $\mathbf{K}_3 = 8\mathbf{I}$, $\mathbf{K}_4 = \mathbf{I}$, $\alpha = 200$, $k = 50$, $\delta = 10^{-2}$, and $\xi = 1$, $\omega_n = 100$ for both $CF_1$ and $CF_2$.

## IX. Conclusions & Future Work

This paper presents trajectory tracking control for VTOL UAVs using the command filtered backstepping technique. The commanded trajectory may include just the position or position and desired yaw. Quaternions are used for attitude control, which ensures the global attitude tracking performance. The quaternion used in this paper follows the modified definition, which is well recognized by the navigation and robotics communities, to facilitate the adoption of various quaternion-based nonlinear controllers by practitioners outside the control community in their applications. More importantly, in the backstepping control design, command filters are used to avoid the often prohibitively difficult analytic computation of the required command derivatives in each step. For the quaternion filtering, a standard vector-based command filter does not exploit the special dynamics of the quaternion, which could result in a longer route being taken by the filter to track the desired quaternion. To address this issue, a second-order quaternion filter is introduced that automatically computes the derivative of quaternion (namely the angular velocity) without differentiation, always follows the smallest angular path, and maintains the unit norm property of the quaternion. As a benefit of using command filters, the flexibility of giving yaw commands in the trajectory is realized without adding extra efforts in the design process. In the future, model error, actuator allocation, and an adaptive version of the controller can be considered.

## Appendix I
## Attitude Error Dynamics

The purpose of this appendix is to show that the dynamic of the attitude error $\tilde{q}$, defined in eqn. (24), is

$$\dot{\tilde{q}} = \frac{1}{2}\bar{q}_{\tilde{\boldsymbol{\omega}}^B} \otimes \tilde{q} \tag{59}$$

where $\tilde{\boldsymbol{\omega}}^B \triangleq \tilde{\boldsymbol{\omega}}^B_{B^cB} = \boldsymbol{\omega}^B_{GB} - \mathbf{R}(\tilde{q})\boldsymbol{\omega}^{B^c}_{GB^c}$. The proof follows the derivation in [14].

*Proof.* From eqn. (24), we have

$$^B_G\bar{q} = \tilde{q} \otimes \bar{q}_c \tag{60}$$

Taking the time derivative on both sides of (60) yields

$$^B_G\dot{\bar{q}} = \dot{\tilde{q}} \otimes \bar{q}_c + \tilde{q} \otimes \dot{\bar{q}}_c \tag{61}$$

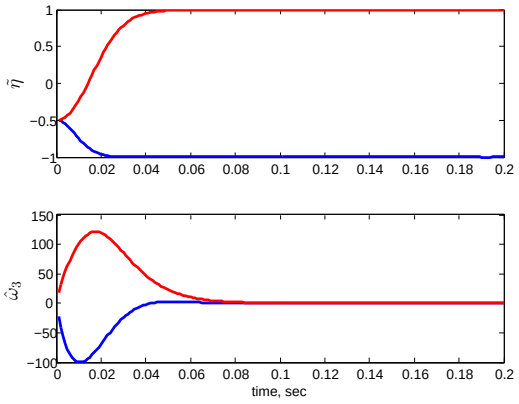


Fig. 2. The quaternion tracking error ($\tilde{\eta}$) and the last component of the output angular velocity ($\hat{\boldsymbol{\omega}}_3$) are shown for the vector-based filter (red) and the proposed quaternion filter (blue).

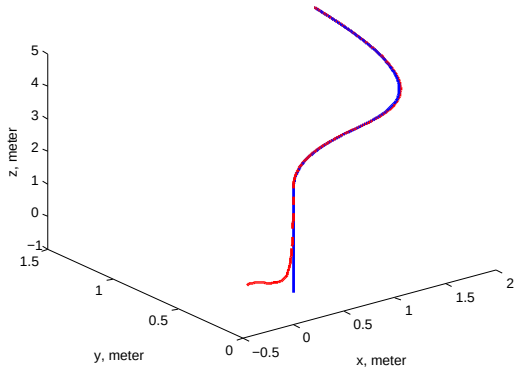


Fig. 3. The desired trajectory (blue), and the actual trajectory of the vehicle (red). The initial position of the vehicle is $[-0.2, 0.2, 0]^\top$.

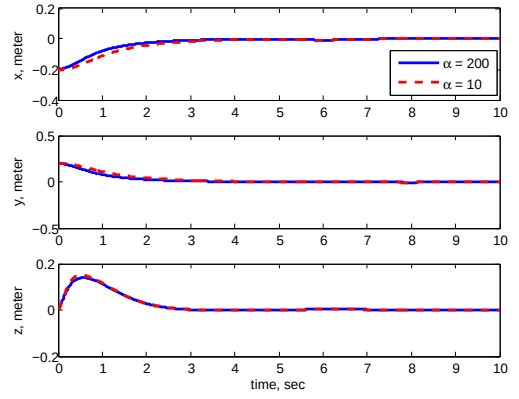


Fig. 4. The position tracking errors ($\mathbf{p} - \mathbf{p}_d$). The blue line is using $\alpha = 200$ and the red dash line is using $\alpha = 10$.

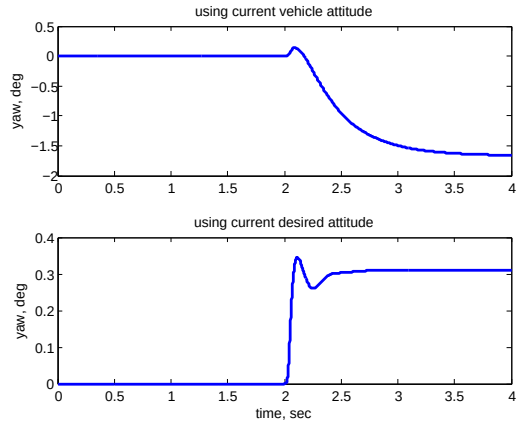


Fig. 5. The comparison of yaw rotation by choosing the reference attitude to be the current desired attitude or the current vehicle attitude.

Rearrange the above equation

$$\dot{\tilde{q}} \otimes \bar{q}_c = {}^B_G\dot{\bar{q}} - \tilde{q} \otimes \dot{\bar{q}}_c \tag{62}$$

$$\dot{\tilde{q}} = \left({}^B_G\dot{\bar{q}} - \tilde{q} \otimes \dot{\bar{q}}_c\right) \otimes \bar{q}_c^{-1} \tag{63}$$

Applying eqn. (13) twice yields

$$\dot{\tilde{q}} = \frac{1}{2}\left(\bar{q}_{\boldsymbol{\omega}^B_{GB}} \otimes {}^B_G\bar{q} - \tilde{q} \otimes \bar{q}_{\boldsymbol{\omega}^{B^c}_{GB^c}} \otimes \bar{q}_c\right) \otimes \bar{q}_c^{-1}$$

Distributing $\bar{q}_c^{-1}$ into the bracket and using eqn. (24) yields:

$$\dot{\tilde{q}} = \frac{1}{2}\left(\bar{q}_{\boldsymbol{\omega}^B_{GB}} \otimes \tilde{q} - \tilde{q} \otimes \bar{q}_{\boldsymbol{\omega}^{B^c}_{GB^c}}\right) \tag{64}$$

Multiplying $\tilde{q}^{-1} \otimes \tilde{q}$ to the right of the second term on the right-hand side yields

$$\dot{\tilde{q}} = \frac{1}{2}\left(\bar{q}_{\boldsymbol{\omega}^B_{GB}} \otimes \tilde{q} - \tilde{q} \otimes \bar{q}_{\boldsymbol{\omega}^{B^c}_{GB^c}} \otimes \tilde{q}^{-1} \otimes \tilde{q}\right)$$

Then eqn. (7) is used to obtain:

$$\dot{\tilde{q}} = \frac{1}{2}\left(\bar{q}_{\boldsymbol{\omega}^B_{GB}} \otimes \tilde{q} - \bar{q}_{\boldsymbol{\omega}^B_{GB^c}} \otimes \tilde{q}\right) \tag{65}$$

$$= \frac{1}{2}\bar{q}_{\tilde{\boldsymbol{\omega}}^B} \otimes \tilde{q} \tag{66}$$

where $\tilde{\boldsymbol{\omega}}^B \triangleq \tilde{\boldsymbol{\omega}}^B_{B^cB} = \boldsymbol{\omega}^B_{GB} - \boldsymbol{\omega}^B_{GB^c} = \boldsymbol{\omega}^B_{GB} - \mathbf{R}(\tilde{q})\boldsymbol{\omega}^{B^c}_{GB^c}$. □

## APPENDIX II

In this appendix, we rewrite the acceleration tracking error $\tilde{\boldsymbol{\mu}}$ into a function of the vector portion of the quaternion error $\tilde{\boldsymbol{\epsilon}}$. Let $A_c = \frac{F_c}{m}$ to simplify the notation. From the definition of $\tilde{\boldsymbol{\mu}}$, we have

$$\tilde{\boldsymbol{\mu}} = A_c {}^G_B\mathbf{R}\mathbf{e}_3 - \boldsymbol{\mu}_c \tag{67}$$

$$= A_c\left({}^G_B\mathbf{R} - {}^G_{B^c}\mathbf{R}\right)\mathbf{e}_3 \tag{68}$$

$$= A_c\left(\mathbf{R}(\bar{q}) - \mathbf{R}(\bar{q}_c)\right)\mathbf{e}_3 \tag{69}$$

where we use the assumption of perfect actuators (i.e., $F_c(t) = F(t)$). Applying eqn. (25) yields

$$\tilde{\boldsymbol{\mu}} = A_c\left(\mathbf{R}(\tilde{q})\mathbf{R}(\bar{q}_c) - \mathbf{R}(\bar{q}_c)\right)\mathbf{e}_3 \tag{70}$$

$$= A_c\left(\mathbf{R}(\tilde{q}) - \mathbf{I}\right)\mathbf{R}(\bar{q}_c)\mathbf{e}_3 \tag{71}$$

Using eqn. (3) gives

$$\tilde{\boldsymbol{\mu}} = 2A_c\left(-\tilde{\eta}S(\tilde{\boldsymbol{\epsilon}}) + S^2(\tilde{\boldsymbol{\epsilon}})\right)\mathbf{R}(\bar{q}_c)\mathbf{e}_3 \tag{72}$$

$$= 2A_c\left(-\tilde{\eta}\mathbf{I} + S(\tilde{\boldsymbol{\epsilon}})\right)S(\tilde{\boldsymbol{\epsilon}})\mathbf{R}(\bar{q}_c)\mathbf{e}_3 \tag{73}$$

$$= 2A_c\left(\tilde{\eta}\mathbf{I} - S(\tilde{\boldsymbol{\epsilon}})\right)S(\mathbf{R}(\bar{q}_c)\mathbf{e}_3)\tilde{\boldsymbol{\epsilon}} \tag{74}$$

$$= \mathbf{W}(\bar{q}_c, \tilde{q}, F_c)\tilde{\boldsymbol{\epsilon}} \tag{75}$$

where

$$\mathbf{W}(\bar{q}_c, \tilde{q}, F_c) = 2\frac{F_c}{m}\left(\tilde{\eta}\mathbf{I} - S(\tilde{\boldsymbol{\epsilon}})\right)S(\mathbf{R}(\bar{q}_c)\mathbf{e}_3) \tag{76}$$

## REFERENCES


[1] P. Bristeau, F. Callou, D. Vissière, N. Petit *et al.*, "The navigation and control technology inside the AR. Drone micro UAV," in *Preprints of the 18th IFAC World Congress*, 2011, pp. 1477–1484.

[2] J. A. Farrell, *Aided navigation: GPS with high rate sensors*. McGraw-Hill New York, NY, USA, 2008.

[3] J. A. Farrell, M. Polycarpou, M. Sharma, and W. Dong, "Command filtered backstepping," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1391–1395, 2009.

[4] G. Hoffmann, S. Waslander, and C. Tomlin, "Quadrotor helicopter trajectory tracking control," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008.

[5] T. Lee, M. Leoky, and N. McClamroch, "Geometric tracking control of a quadrotor UAV on SE (3)," in *IEEE Conference on Decision and Control*, 2010, pp. 5420–5425.

[6] T. Lee, N. H. McClamroch, and M. Leok, "Optimal control of a rigid body using geometrically exact computations on SE (3)," in *IEEE Conference on Decision and Control*, 2006, pp. 2710–2715.

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

[8] C. Mayhew, R. Sanfelice, and A. Teel, "Quaternion-based hybrid control for robust global attitude tracking," *IEEE Transactions on Automatic Control*, no. 99, pp. 1–1, 2010.

[9] C. G. Mayhew, R. G. Sanfelice, and A. R. Teel, "Robust global asymptotic attitude stabilization of a rigid body by quaternion-based hybrid feedback," in *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference (CDC/CCC)*, 2009, pp. 2522–2527.

[10] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, 2012.

[11] A. Roberts and A. Tayebi, "Adaptive position tracking of VTOL UAVs," *IEEE Transactions on Robotics*, no. 99, pp. 1–14, 2011.

[12] R. G. Sanfelice, R. Goebel, and A. R. Teel, "Invariance principles for hybrid systems with connections to detectability and asymptotic stability," *IEEE Transactions on Automatic Control*, vol. 52, no. 12, pp. 2282–2297, 2007.

[13] M. Shuster, "The nature of the quaternion," *Journal of the Astronautical Sciences*, vol. 56, no. 3, p. 359, 2010.

[14] A. Tayebi, "Unit quaternion-based output feedback for the attitude tracking problem," *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1516–1520, 2008.

[15] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep. 2005-002, Mar. 2005.