Quaternion based attitude control of a maneuvering fixed wing UAV

1st Stephen Kimathi
Department of Control Engineering and Information Technology
Faculty of Electrical Engineering and Informatics
Budapest University of Technology and Economics
Budapest, Hungary
kimathi@iit.bme.hu

2nd Béla Lantos
Department of Control Engineering and Information Technology
Faculty of Electrical Engineering and Informatics
Budapest University of Technology and Economics
Budapest, Hungary
lantos@iit.bme.hu

2022 IEEE 26th International Conference on Intelligent Engineering Systems (INES) | 978-1-6654-9209-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/INES56734.2022.9922611

***Abstract*- This paper presents attitude control using the unit quaternion. Specifically, the orientation of a maneuvering fixed wing UAV is investigated through set point tracking. Two formulations are explored; the first, through quaternion error dynamics and the second, through quaternion logarithm. Both methods apply appropriate Lyapunov functions for the design and analysis of the closed stability of the control laws. The error dynamics are deduced from composite rotations between different frames of the UAV. Through the error dynamics, the orientation of the UAV's wind frame is aligned to a desired orientation. Using this procedure, the desired equilibrium points of the closed system are guaranteed to converge and are asymptotically stable. The formulation and implementation of these two methods is simple and intuitive, and through simulations, their successful use and effectiveness is shown.**

Keywords— attitude control, quaternion, quaternion logarithm, orientation, fixed wing UAV

I. INTRODUCTION

The dynamic model of a conventional aircraft given in different literatures use the Byran procedure to derive the corresponding equations of motion. The aim is to realize the Newton's second law of motion for the six degrees of freedom describing the motion and orientation of an aircraft at any given time [1]. A common challenge that the above method encounters with respect to representation of the attitude of an aircraft, is the gimbal lock. This effect arises due to transformation of angular velocities to Euler angles. The integration of the ensuing variables become indeterminate for $\theta = \pm\pi/2$, hence creating a singularity.

A common way to circumvent this singularity is to use a single rotation about an eigen axis coupled with a principal angle of rotation to describe the orientation of a non-inertial frame relative to an inertial frame. This method is attributed to the Euler-Rodrigues quaternion formulation where the parameters in the Euler axis are used to define four parameters [2]. These four parameters are referred to as a quaternion, $\mathbf{q}$ and can uniquely describe any orientation having three degrees of freedom. However, the description is not unique: $\mathbf{q} = -\mathbf{q}$ represent the same orientation. The formulation is simple and free from singularities. However, the interpretation of a quaternion is less intuitive than the Euler angles and can be confusing especially for composite rotations. Using quaternion formulation and the Newton's approach, the translational and rotational dynamics of any rigid body can be defined completely for a full flight envelope, that is $\alpha, \beta \cong \pm\pi/2$. For a maneuvering UAV, this nonlinear behavior would be difficult to capture using the conventional attitude kinematic representation. The unit quaternion can be used to represent such dynamics, velocities and orientation kinematics. Quaternions have since been used in satellites [3] and quadrotor UAVs [4] for attitude control. A kinematic control law formulation of rigid bodies using dual quaternions was discussed in [5] and [6], where unit quaternions were used and formulated using dual numbers. Quaternion logarithms have also been applied for nonlinear control of UAVs such as in [7].

This paper exploits the advantages of quaternions and quaternion logarithm for the solution of the orientation control problem in a maneuvering fixed wing UAV. The paper is organized as follows. Section II introduces the quaternion and the related algebra while section III provides the dynamic and kinematics of a fixed wing UAV. Section IV discusses the controller design using quaternion and quaternion logarithm. Section V gives the simulation results and discussions and the conclusion is given in section VI.

II. QUATERNION FORMULATION

Quaternions are hyper complex numbers and they are represented by one, real and three, imaginary numbers as $\mathbf{q} \in \mathbb{H} \; q_0 + q_1 i + q_2 j + q_3 k$ where $i, j, k \in I$ such that $i^2 = j^2 = k^2 = ijk = -1$. In vector form $\mathbf{q} = [q_0,\ q_1,\ q_2,\ q_3]^T$, or $\mathbf{q} = (s, \mathbf{v})^T$ in compact form with a scalar s and a vector $\mathbf{v} \in \mathbb{R}^3$. A rotation matrix denoted by $\mathbf{R}_a^b \in \mathrm{S}^3 = \{\mathbf{R} \in \mathbb{R}^{3\times 3}\}$ rotates a vector from a frame a to a frame b. Angular velocity vector denoted by $\omega_{a,c}^e$ represents the angular velocity of a frame c relative to a frame a, referenced to a frame e. Angular velocities can be added such that $\omega_{a,d}^b = \omega_{a,c}^b + \omega_{c,d}^b$. The time derivative of a rotation matrix is found as:

$$\dot{\mathbf{R}}_a^b = \mathbf{R}_a^b \mathbf{S}(\omega_{b,a}^a) \tag{1}$$

The cross product operator $\mathbf{S}(\cdot)$ is such that for two vectors $v_1 = [v_1,\ v_2,\ v_3]^T, v_2 \in \mathbb{R}^3, \mathbf{S}(v_1)v_2 = v_1 \times v_2, \mathbf{S}(v_1)v_2 = -\mathbf{S}(v_2)v_1$ and $v_1^T \mathbf{S}(v_2) v_1 = 0$ where:

$$\mathbf{S}(v_1) = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \tag{2}$$

A rotation matrix can be parameterized using quaternions, where a quaternion can represent a rotation from a frame a to a frame b ; $\mathbf{R}_a^b := \mathbf{q}_{b,a} \in \mathrm{S}^3 = \{\mathbf{q} \in \mathbb{R}^4 : \mathbf{q}^T\mathbf{q} = 1\}$ such that $\mathbf{q}_{b,a} = \left[\eta_{b,a},\ \varepsilon_{b,a}^T\right]^T = \left[\cos\left(\frac{\vartheta_{b,a}}{2}\right),\ \mathbf{k}_{b,a}^T \sin\left(\frac{\vartheta_{b,a}}{2}\right)\right]^T$. $\eta_{b,a}$ is a scalar, $\varepsilon_{b,a} \in \mathbb{R}^3$ a vector, $\mathbf{k}_{c,a}$ is the axis of rotation and $\vartheta_{c,a}$ is the principal angle of rotation. From this organization, a rotation matrix can be reconstructed as:

$$\mathbf{R}_a^b = \mathbf{I} + 2\eta_{b,a}\mathbf{S}(\varepsilon_{b,a}) + 2\mathbf{S}^2(\varepsilon_{b,a}) \tag{3}$$

The product of two quaternions $\mathbf{q}_1, \mathbf{q}_2$ is expressed as $\mathbf{q}_1 \otimes \mathbf{q}_2 := \left[\eta_1\eta_2 - \varepsilon_1^T\varepsilon_2,\ \eta_1\varepsilon_2 + \eta_2\varepsilon_1 + \mathbf{S}(\varepsilon_1)\varepsilon_2\right]^T$. Composite rotations using the quaternion product are given as:

$$\mathbf{q}_{b,e} = \mathbf{q}_{b,a} \otimes \mathbf{q}_{a,e} = \mathbf{T}(\mathbf{q}_{b,a})\mathbf{q}_{a,e} \tag{4}$$

where $\mathbf{T}(\mathbf{q}_{b,a}) = \left[-\varepsilon_{b,a}^T,\ \eta_{c,a}\mathbf{I} + \mathbf{S}(\varepsilon_{b,a})\right]^T$ ensures the resultant quaternion is of unit length. The quaternion kinematics of a composite rotation is written as:

$$\dot{\mathbf{q}}_{b,a} = \frac{1}{2}\mathbf{q}_{b,a} \otimes \left[0,\ \omega_{b,a}^a\right]^T = \frac{1}{2}\mathbf{T}(\mathbf{q}_{b,a})\left[\omega_{b,a}^a\right] \tag{5}$$

The logarithm of a quaternion in compact form is $\log(\mathbf{q}) = \left[\log|\mathbf{q}|,\ \frac{\varepsilon}{|\varepsilon|}\arccos\left(\frac{\eta}{|\mathbf{q}|}\right)\right] = \left[0,\ \frac{\vartheta}{2}\mathbf{k}^T\right]^T$. For a rotation from a frame a to a frame b the logarithm is given as $\log(\mathbf{q}_{b,a}) = \left[0,\ \frac{\vartheta_{b,a}}{2}\mathbf{k}^T{}_{b,a}\right]^T$ and the time derivative as in:

$$\frac{d}{dt}\log(\mathbf{q}_{b,a}) = \left[0,\ \frac{\dot{\vartheta}_{b,a}}{2}\mathbf{k}^T{}_{b,a}\right]^T \tag{6}$$

The orientation of a body frame with respect to an inertia frame in a rigid body can be described using a unit quaternion according to $R_b^n = I + 2\eta[\varepsilon\times] + 2[\varepsilon\times][\varepsilon\times]$ where $\eta = \cos(\vartheta/2)$ and $\varepsilon = \sin(\vartheta/2)\mathbf{k}^T$. This relation is easily converted to a form using quaternions, q_0, q_1, q_2, q_3 [8] as

$$R_b^n = \begin{bmatrix} q_0^2+q_1^2-q_3^2-q_2^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2+q_2^2-q_3^2-q_1^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2+q_3^2-q_2^2-q_1^2 \end{bmatrix}.$$

III. DYNAMICS AND KINEMATICS

UAV dynamics can be categorized into translational dynamics and rotational dynamics. Superscript denotes reference frame of a vector, where b is body frame, w is wind frame, n is inertial frame and d is a desired frame.

A. Translational Dynamics

The translational dynamics of a fixed wing UAV are defined using (7) and (8),

$$\dot{\mathbf{p}}^n = \mathbf{R}_b^n v^b \tag{7}$$

$$v_r^b = v^b - \mathbf{R}_n^b v_{wind}^n \tag{8}$$

$\mathbf{p}^n := \left[x,\ y,\ z\right]^T$ is the position of the UAV in the NED frame, $\mathbf{R}_b^n$ is the rotation matrix from body to the NED frame, v^b is the velocity of the UAV relative to the Earth and $v_r^b := \left[u,\ v,\ w\right]^T$ is the velocity relative to the surrounding air, v_{wind}^n is the velocity vector of the wind. The total velocity, V_a is related to v_r^b through:

$$V_a := \left\|v_r^b\right\| \Rightarrow v_r^b = \mathbf{R}_w^b\left[V_a,\ 0,\ 0\right]^T \tag{9}$$

with $\mathbf{R}_b^w = \begin{bmatrix} \cos(\alpha)\cos(\beta) & \sin(\beta) & \sin(\alpha)\cos(\beta) \\ -\cos(\alpha)\sin(\beta) & \cos(\beta) & -\sin(\alpha)\sin(\beta) \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix}$ being the rotation matrix from the body frame to the wind frame. $\alpha = \tan^{-1}(w/u)$ and $\beta = \sin^{-1}(v/V_a)$. The acceleration in a UAV is expressed as:

$$\dot{\mathbf{v}}_r^b = \frac{1}{m}\left(f_{thrust}^b + \mathbf{R}_w^b f_{aero}^w\right) + \mathbf{R}_n^b f_g^n - \mathbf{S}(\omega_{n,b}^b)v_r^b \tag{10}$$

where m is the mass of the vehicle, $f_{thrust}^b = \left[T,\ 0,\ 0\right]^T$ is the thrust vector which is assumed to be aligned with the x^b axis, $f_g^n = \left[0,\ 0,\ g\right]^T$ is gravitational acceleration vector. The aerodynamic relations relating to forces are given by the expressions in (11) and (12).

$$f_{aero}^w = \frac{1}{2}\rho S V_a^2\left[-C_D,\ C_Y,\ -C_L\right]^T \tag{11}$$

$$\begin{aligned} C_D &= C_{D_0} + C_{D_\alpha}\alpha + \frac{\bar{c}}{2V_a}C_{D_q}q + C_{D_{\delta_e}}\delta_e \\ C_Y &= C_{Y_0} + C_{Y_\beta}\beta + \tilde{b}C_{Y_p}p + \tilde{b}C_{Y_r}r + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r \\ C_L &= C_{L_0} + C_{L_\alpha}\alpha + \frac{\bar{c}}{2V_a}C_{L_q}q + C_{L_{\delta_e}}\delta_e \end{aligned} \tag{12}$$

ρ is the air density, S the wing area, b the wing span, $\bar{c}$ the mean aerodynamic chord, $C(\cdot)$ represents the control and stability derivatives, and $\tilde{b} = b/2V_a$. The aerodynamics are linear in the angle of attack and sideslip angle. This model is valid for small angles of attack.

B. Rotational Dynamics

Using quaternions, the rotational kinematics and dynamics can be expressed as in (13) and (14).

$$\dot{\mathbf{q}}_{n,b} = \frac{1}{2}\mathbf{q}_{n,b} \otimes \left[0,\ \omega_{n,b}^b\right]^T = \frac{1}{2}\mathbf{T}(\mathbf{q}_{n,b})\omega_{n,b}^b \tag{13}$$

$$I^n\dot{\omega}_{n,b}^b = -\mathbf{S}(\omega_{n,b}^b)I^n\omega_{n,b}^b + f(x) + D(x)\omega_{n,b}^b + G(x)u \tag{14}$$

$\mathbf{q}_{n,b}$ represents the orientation of the body frame relative to the NED frame, $\omega_{n,b}^{b} = [p, q, r]^T$ is the angular velocity, $I^n \in \mathbb{R}^{3\times3}$ is the inertia matrix, $x = [V_a,\ \alpha,\ \beta]^T$ and $u = [\delta_a,\ \delta_e,\ \delta_r]^T$ form the UAV control deflections. The breakdown of components listed in (14) is in:

$$f(x) = \frac{1}{2}\rho S V_a^{\,2} \begin{bmatrix} b(C_{l_0} + C_{l_\beta}\beta) \\ \bar{c}(C_{m_0} + C_{m_\alpha}\alpha) \\ b(C_{n_0} + C_{n_\beta}\beta) \end{bmatrix} \tag{15}$$

$$D(x) = \frac{1}{2}\rho S V_a^{\,2} \begin{bmatrix} \frac{b^2}{2V_a}C_{l_p} & 0 & \frac{b^2}{2V_a}C_{l_r} \\ 0 & \frac{\bar{c}^2}{2V_a}C_{m_q} & 0 \\ \frac{b^2}{2V_a}C_{n_p} & 0 & \frac{b^2}{2V_a}C_{n_r} \end{bmatrix} \tag{16}$$

$$G(x) = \frac{1}{2}\rho S V_a^{\,2} \begin{bmatrix} bC_{l_{\delta_a}} & 0 & bC_{l_{\delta_r}} \\ 0 & \bar{c}C_{m_{\delta_e}} & 0 \\ bC_{n_{\delta_a}} & 0 & bC_{n_{\delta_r}} \end{bmatrix}. \tag{17}$$

Equations (15), (16) and (17) represent partitions of the aerodynamic moments. With the aid these dynamics, a set of control deflections *u* can be designed to achieve a desired orientation and trajectory.

IV. CONTROLLER DESIGN

A. Quaternion Control

The control problem is to design a control law to point the UAV wind frame to a desired orientation. Let $\mathbf{q}_{n,d}, \omega_{n,d}^{d}, \dot{\omega}_{n,d}^{d}$ be the desired orientation, angular velocity and acceleration respectively. The attitude error of the wind frame relative to the desired frame using composite quaternion rotations is given as in:

$$\mathbf{q}_{d,w} = \mathbf{q}_{d,n} \otimes \mathbf{q}_{n,b} \otimes \mathbf{q}_{b,w} \tag{18}$$

$\mathbf{R}_w^d = I^{3\times3} \Leftrightarrow \mathbf{q}_{d,w}^{\infty} = [\pm1, \bar{\mathbf{0}}]$ is the equilibria, and has two solutions states which represent the same orientation, but algebraically different. The error in orientation becomes:

$$\mathbf{e}_{q\pm} \coloneqq \mathbf{q}_{d,w}^{\infty} - \mathbf{q}_{d,w} = [1 \mp \eta_{d,w}, \varepsilon_{d,w}^T]^T \tag{19}$$

where $\|\mathbf{e}_{q\pm}\| \neq 1$. The error kinematics are expressed as in:

$$\dot{\mathbf{e}}_{q\pm} = \mathbf{T}_{eq}(\mathbf{e}_{q\pm})\mathbf{R}_b^w \omega_{d,w}^b \tag{20}$$

where $\mathbf{T}_{eq} = \left[\varepsilon_{d,w}^T,\ \eta_{d,w}\mathbf{I} + \mathbf{S}(\varepsilon_{d,w})\right]^T$.

Lemma 1 The following inequality holds for this purpose $\mathbf{e}_{q\pm}{}^T \mathbf{T}_{eq}(\mathbf{e}_{q\pm}) \mathbf{T}_{eq}{}^T(\mathbf{e}_{q\pm}) \mathbf{e}_{q\pm} \geq \frac{1}{8}\mathbf{e}_{q\pm}{}^T \mathbf{e}_{q\pm}$. The proof of this relation can be found in [10].

The angular velocity between the relative frames is given in (21). $\omega_{d,w}^b$ is the angular velocity in *w* frame relative to *d* frame, referenced to the *b* frame. The angular velocity of the *w* frame relative to the *b* frame is given as $\omega_{b,w}^w = \left[-\dot{\alpha}\sin(\beta),\ -\dot{\alpha}\cos(\beta),\ \dot{\beta}\right]^T$ [9].

$$\omega_{d,w}^b = \omega_{n,b}^b - \mathbf{R}_d^b \omega_{n,d}^d + \mathbf{R}_w^b \omega_{b,w}^w \tag{21}$$

The control goal is to make $(\mathbf{e}_{q\pm}, \omega_{d,w}^b) \to (\bar{\mathbf{0}}, \bar{\mathbf{0}})$ such that the UAV aligns with the desired orientation and angular velocity. A sliding surface controller is designed using the above formulation and following the procedure given in [11] to design the control deflections as in:

$$\begin{aligned} u = \mathbf{G}^{-1}(x)(I^n \dot{\omega}_{n,r}^b - D(x)\omega_{n,r}^b + \mathbf{S}(\omega_{n,b}^b) I^n \omega_{n,b}^b \\ -\mathbf{f}(x) - k_s s - k_q \mathbf{R}_w^b \mathbf{T}_{eq}^T \mathbf{e}_q) \end{aligned} \tag{22}$$

From (22), $s = \omega_{n,b}^b - \omega_{n,r}^b$ is the designed sliding surface and $\omega_{n,r}^b = \mathbf{R}_d^b \omega_{n,d}^d - \mathbf{R}_w^b \omega_{b,w}^w - \Lambda \mathbf{R}_w^b \mathbf{T}_{eq}{}^T \mathbf{e}_q$ is an angular velocity relative to *n* frame and whose derivative is expressed as:

$$\begin{aligned} \dot{\omega}_{n,r}^b = \mathbf{R}_d^b \dot{\omega}_{n,d}^d - \mathbf{S}(\omega_{n,b}^b)\mathbf{R}_d^b \omega_{n,d}^d - \mathbf{R}_w^b \dot{\omega}_{b,w}^w \\ -\Lambda \mathbf{R}_w^b (\omega_{b,w}^w) \mathbf{T}_{eq}{}^T \mathbf{e}_q - \frac{1}{2}\Lambda \mathbf{R}_w^b \dot{\varepsilon}_{d,w} \end{aligned} \tag{23}$$

$k_q, k_s > 0 \in \mathrm{R}^1$ and $\Lambda = \Lambda^T > 0$ assures that the closed loop system is uniformly exponentially stable. As $(\mathbf{e}_{q\pm}, s) \to (\bar{\mathbf{0}}, 0)$ then from (21) $\omega_{d,w}^b \to 0$ asymptotically as $t \to \infty$.

The two equilibrium states from (18) arise due to the fact that $\mathbf{q} \Leftrightarrow -\mathbf{q}$ in quaternions represent the same orientation matrix however, one is rotated by 2π relative to the other. Therefore, under the control law (22), $\mathbf{q}_{n,b}$ in the closed loop system will converge to $(-1,\ \bar{\mathbf{0}}^T)^T$ if the initial state is near $(-1,\ \bar{\mathbf{0}}^T)^T$ or vice versa, therefore taking the 'shortest' path. Reference [6] proves that both orientations are stable equilibrium states.

B. Quaternion Logarithm Control

The goal is to make the quaternion $\mathbf{q}_{d,w} \to \mathbf{q}_{n,d}$ and $\omega_{d,w}^b \to 0$. From (6), we can use the property $\omega_{d,w}^b = \dot{\vartheta}_{d,w} k_{d,w}$ such that:

$$\frac{d}{dt}\log(\mathbf{q}_{d,w}) = \left[0,\ \frac{\dot{\vartheta}_{d,w}}{2}\mathbf{k}^T{}_{d,w}\right]^T = \left[0,\ \omega_{d,w}^b\right]^T \tag{24}$$

The right hand side of (24) is only an approximation [12]. We select a Lyapunov function V_1 as:

$$V_1 = k_q \log(\mathbf{q}_{d,w})^T \log(\mathbf{q}_{d,w}) + \frac{1}{2}(\omega_{d,w}^b)^T I^n \omega_{d,w}^b \tag{25}$$

Taking the time derivative of the Lyapunov function, V_1 as:

$$\dot{V}_1 = k_q \log(\mathbf{q}_{d,w})^T \left[0,\ \left(R_b^w \omega_{d,w}^b\right)^T\right]^T + (\omega_{d,w}^b)^T I^n \dot{\omega}_{d,w}^b \quad (26)$$

Taking the time derivative of (21), we get the relation:

$$\dot{\omega}_{d,w}^b = \dot{\omega}_{n,b}^b - \mathbf{R}_d^b \dot{\omega}_{n,d}^d + \mathbf{R}_w^b \dot{\omega}_{b,w}^w \quad (27)$$

We make substitutions from (14) and (27), such that:

$$\begin{aligned} I^n \dot{\omega}_{d,w}^b = &-\mathbf{S}(\omega_{n,b}^b) I^n \omega_{n,b}^b + f(x) + D(x)\omega_{n,b}^b + G(x)u \\ &- I^n \mathbf{R}_d^b \dot{\omega}_{n,d}^d + I^n \mathbf{R}_w^b \dot{\omega}_{b,w}^w \end{aligned} \quad (28)$$

Moreover, the first term in (26) can be manipulated as:

$$\begin{aligned} \log(\mathbf{q}_{d,w})^T \left[0,\ \left(\mathbf{R}_b^w \omega_{d,w}^b\right)^T\right]^T = &\left(\left[0,\ \frac{\vartheta_{d,w}}{2}\mathbf{k}^T{}_{d,w}\right]\right. \\ &\left.\left[0,\ \left(\mathbf{R}_b^w \omega_{d,w}^q\right)^T\right]^T\right) = \frac{\vartheta_{d,w}}{2}\left(\omega_{d,w}^b\right)^T \mathbf{R}_b^w \mathbf{k}^T{}_{d,w} \end{aligned} \quad (29)$$

Inserting (28) and (29) into (26), a quaternion logarithm control is chosen as:

$$\begin{aligned} u = \mathbf{G}^{-1}\Big(&\mathbf{S}(\square) I^n \omega_{n,b}^b + I^n \mathbf{R}_d^b \dot{\omega}_{n,d}^d - I^n \mathbf{S}(\square)\mathbf{R}_d^b \omega_{n,d}^d \\ &-\mathbf{f} - D\omega_{n,b}^b + I^n \dot{\omega}_{b,w}^w - k_q \mathbf{R}_w^b \mathbf{k}_{d,w}^T \frac{\vartheta_{d,w}}{2} - k_w \omega_{d,w}^b\Big) \end{aligned} \quad (30)$$

with $k_q, k_w > 0 \in R^1$, $\mathbf{S}(\square) = \mathbf{S}(\omega_{n,b}^b)$. With this choice, all the terms in $\dot{V}_1$ cancel out, yielding $\dot{V}_1 = -k_w \left\|\omega_{d,w}^b\right\|^2$ which is negative semi-definite. This implies uniform asymptotic stability, that is $(\mathbf{q}_{d,w}, \omega_{d,w}^b) \to (\mathbf{q}_{n,d}, \bar{0})$. Proof of this claim can be realized using Matrosov theorem, see [13].

V. SIMULATIONS

The model used in these simulations is a Yak 54 UAV with the aerodynamic properties listed in Table 1. Other physical properties are as given in [14].

The initial values for the model are chosen as $\omega_{n,b}^b = [0.001, 0.001, 0.001]^T$, $\mathbf{q}_{n,b} = [0,0,0,1]^T$ for the angular velocity and quaternion (initial orientation) respectively. A velocity, V_a of $36 m/s$ was used. The gains for the quaternion based control were chosen as $k_q = k_s = 40$ and $\Lambda = 5\mathbf{I}^{\in 3\times 3}$. The quaternion logarithm control gains were chosen as $k_q = 40, k_w = 40$ and the desired angular velocity $\omega_{n,d}^d = [0.01, 0, 0]^T$. For both the formulations, the desired orientation is $\mathbf{q}_{n,d} = [1,0,0,0]^T$ such that the UAV performs a rotation of π radians in the $\mathbf{x}^n$ axis. The control equations for the two methods are given by (22) and (30).

TABLE 1: AERODYNAMIC PROPERTIES

Longitudinal derivatives (1/rad)		*Lateral derivatives (1/rad)*	
C_{D_u}	0.0011	C_{y_β}	-0.3462
C_{D_α}	-0.0863	C_{y_p}	0.0073
C_{L_u}	0.0017	C_{y_r}	0.2372
C_{L_α}	4.5363	C_{l_β}	-0.0255
$C_{L_{\dot{\alpha}}}$	1.9314	C_{l_p}	-0.3817
C_{L_q}	5.1515	C_{l_r}	0.0504
C_{m_u}	0.0004	C_{n_β}	0.0954
C_{m_α}	-0.3701	C_{n_p}	-0.0156
$C_{m_{\dot{\alpha}}}$	-4.4705	C_{n_r}	-0.1161
C_{m_q}	-8.5026	$C_{y_{\delta r}}$	0.1928
$C_{L_{\delta e}}$	0.3762	$C_{l_{\delta\alpha}}$	0.3490
$C_{m_{\delta e}}$	-0.8778	$C_{l_{\delta r}}$	0.0154
C_{D_q}	0	$C_{n_{\delta\alpha}}$	-0.0088
$C_{D_{\delta e}}$	0	$C_{n_{\delta r}}$	-0.0996

A. Simulation results using quaternion control

Fig. 1 show the evolution of attitude error using a quaternion, e_q alongside the evolution of quaternion in w frame relative to d frame, $\mathbf{q}_{d,w}$. In this case, the shortest path was towards $(-1, \bar{\mathbf{0}}^T)^T$ as the quaternion heads to $[-1, 0, 0, 0]^T$ as opposed to $[1, 0, 0, 0]^T$.

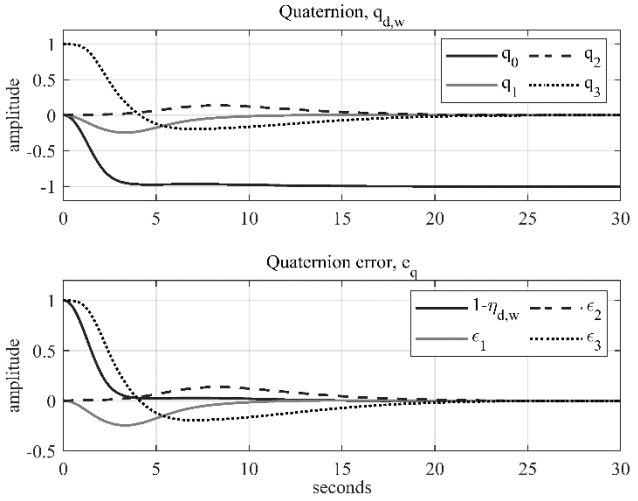

Fig. 1: Evolution of $q_{d,w}$ quaternion and the quaternion error

The angular velocity between the relative frames from (21) is shown in Fig. 2, and it converges to zero as $t \to \infty$, according to the design objective.

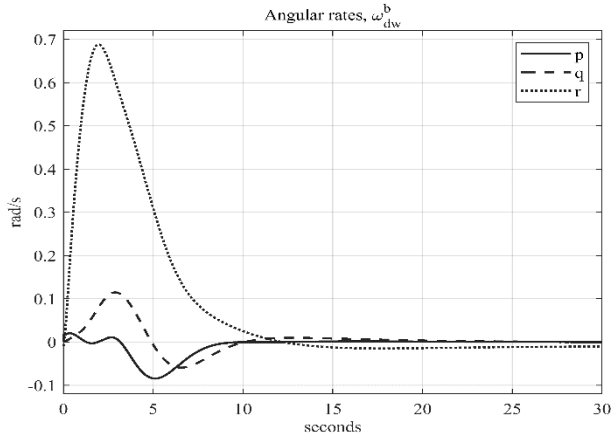

Fig. 2: Angular rates

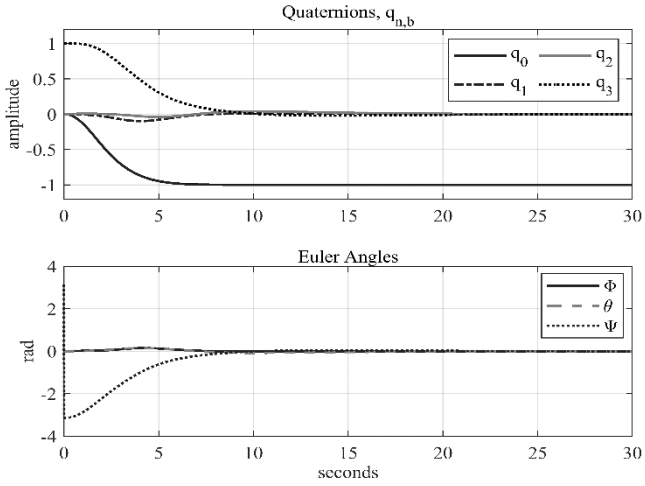

Fig. 3: Quaternion and the Euler angles

Fig. 3 show the orientation of the b frame with respect to the n frame. We note that the evolution of ψ, shifts from π to $-\pi$ at the start of the simulation. Both angles represent the same orientation i.e. $\pi \Leftrightarrow -\pi$ rad. The control deflection angles, angle of attack and the sideslip angle are shown in Fig. 4. The velocity components during the simulation are shown in Fig. 5.

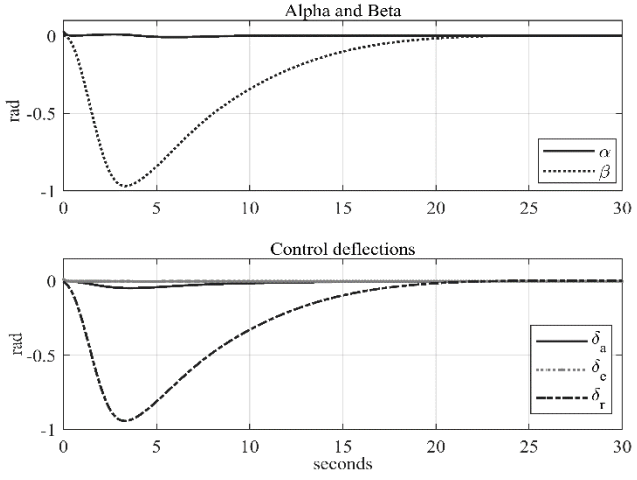

Fig. 4: The Control deflections, angle of attack and sideslip angle

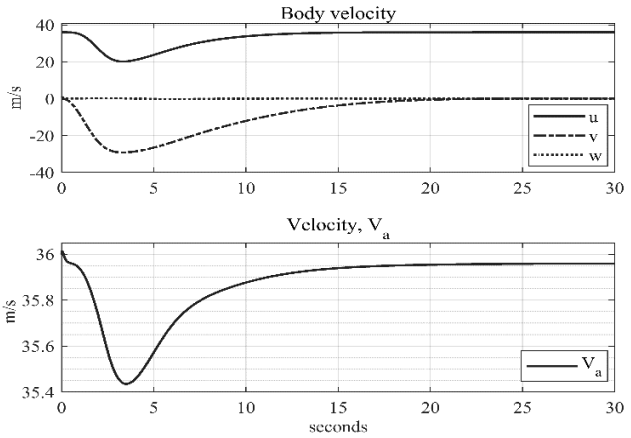

Fig. 5: Evolution of velocity and velocity components

B. *Simulation results using quaternion logarithm control*

Fig. 6 shows evolutions of $\mathbf{q}_{d,w}$ with respect to time. The error in orientation, $\mathbf{q}_e$ is strictly the difference between two orientations $\mathbf{q}_{d,w}$ and $\mathbf{q}_{n,d}$, and it is not similar to an attitude orientation error. This dissimilarity asymptotically heads to zero as the UAV aligns with the desired orientation.

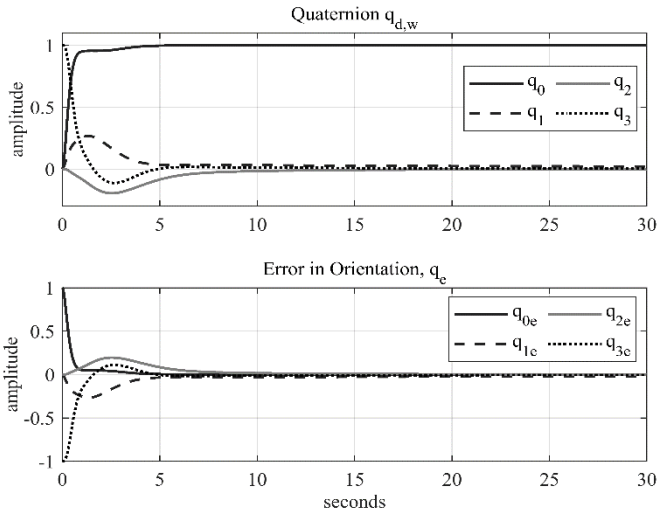

Fig. 6: Desired orientation and the error in orientation

The angular velocity components are shown in Fig. 7, and they converge to zero.

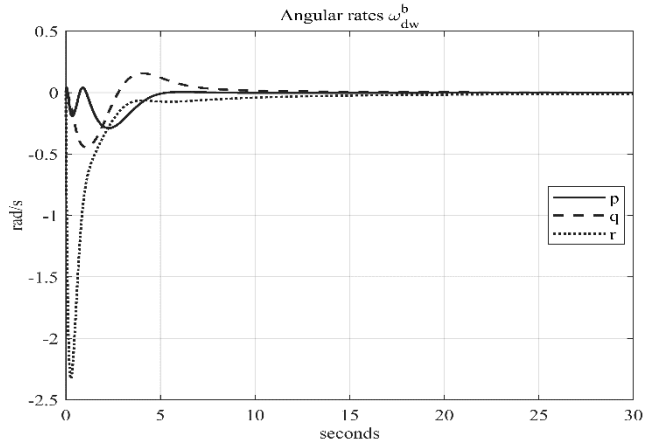

Fig. 7: Angular velocity

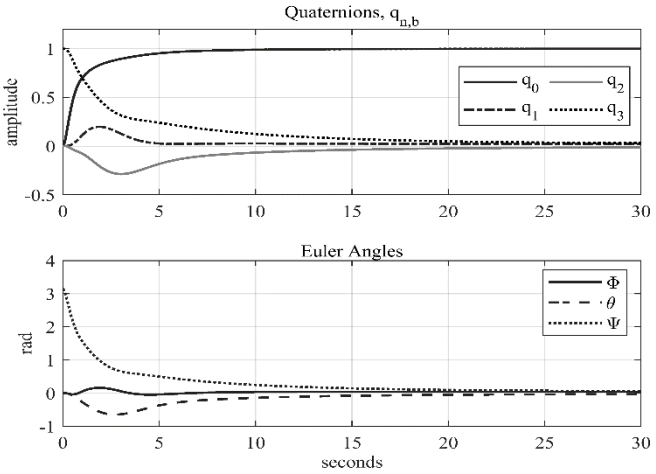

Fig. 8: Quaternion and the Euler angles

Fig. 8 show the orientation of the b frame relative to n frame from an initial orientation of $\mathbf{q}_{n,b} = [0,0,0,1]^T$ to the desired orientation $\mathbf{q}_{n,d} = [1,0,0,0]^T$. The Euler angles show the evolution of the three angles during the orientation alignment. Fig. 9 shows the angle of attack, the sideslip angle and the control deflections during the simulation.

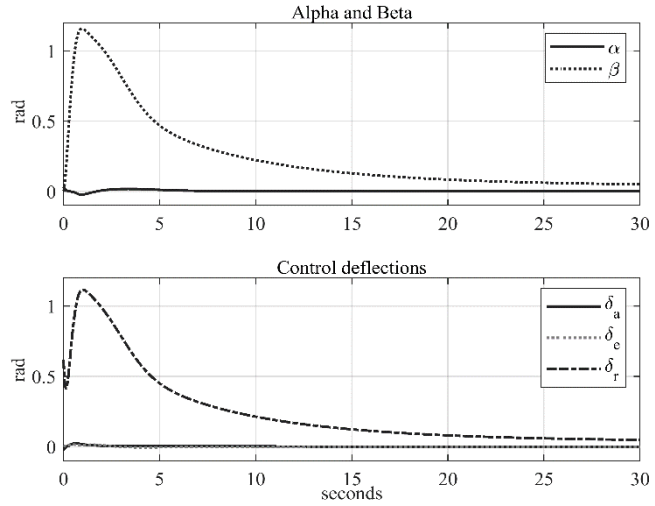

Fig. 9: The Control deflections, angle of attack and sideslip angle

The components of velocity are finally shown in Fig. 10.

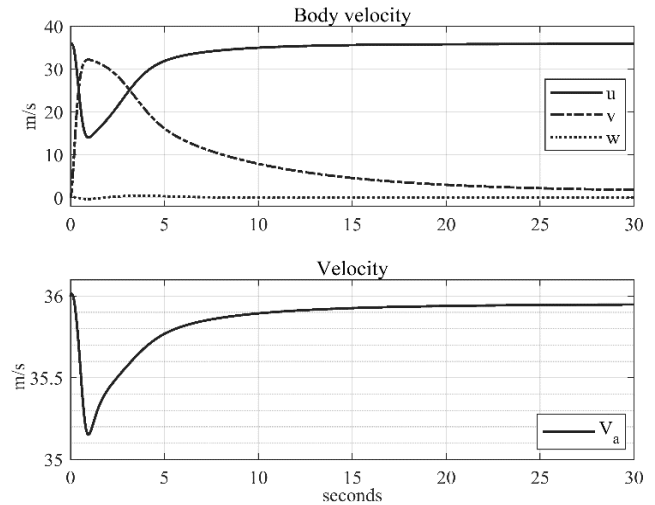

Fig. 10: Evolution of velocity and velocity components

VI. CONCLUSION

Two methods utilizing quaternions in attitude control have been presented. The first method derives a quaternion error from two rotations represented as quaternions such that as the error approaches zero, the UAV inertial orientation is aligned with a desired orientation. The second method uses the quaternion logarithm to steer the wind frame to a desired orientation as the angular velocity between the relative frames heads to zero. Through simulations, the two methods were able to reach the desired orientation thereby demonstrating the effectiveness of the designed control laws for attitude control in a maneuvering fixed wing UAV.

REFERENCES

[1] R. Beard and T. McLain, *Small Unmanned Aircraft; Theory and Practice*. Princeton University Press, 2011.

[2] S. Altmann. "Hamilton, Rodrigues and the Quaternion scandal," Mathematics Magazine, vol. 62(5), pp. 291-308, 1989.

[3] H. Dapeng, W. Qing and L. Zexiang. "Attitude Control Based on the Lie-group Structure of Unit Quaternions," in Proceedings of the 26th Chinese Control Conference, July 2007.

[4] E. Fresk and G. Nikolakopoulos. "Full Quaternion Based Attitude Control for a Quadrotor," European Control Conference, pp. 3864-3869, July 2013.

[5] D. Han, Q. Wei and Z. Li. "Kinematic Control of Free Rigid Bodies Using Dual Quaternions," International Journal of Automation and Computing, vol. 5(3), pp. 319-324, July 2008.

[6] X. Wang and C. Yu. "Unit Dual quaternion-based feedback linearization tracking problem for attitude and position dynamics," Systems & Control Letters, vol. 62, pp. 225–233, 2013.

[7] Z. Bodó and B. Lantos. "Nonlinear control of maneuvering fixed wing UAVs using quaternion logarithm," in 23rd International Symposium on Measurement and Control in Robotics, October 2020.

[8] W. Phillips and C. Hailey "Review of Attitude Representations used for Aircraft Kinematics," Journal of Aircraft, vol. 38(4), pp 718-737, 2001.

[9] E. Oland. "Quaternion-based Control of Fixed-Wing UAVs Using Logarithmic Mapping," in 9th International Conference on Mechanical and Aerospace Engineering, July 2018.

[10] R. Kristiansen, A. Loria, A. Chaillet, and P. Nicklasson. "Spacecraft relative rotation tracking without angular velocity measurements," Automatica, vol. 45(3), pp. 750-756, 2009.

[11] J. Slotine and W. Li. "On the Adaptive Control of Robot Manipulators," International Journal of Robotics Research, vol. 6(3), pp. 49-59, 1987.

[12] B. Lantos and L. Marton, *Nonlinear Control of Vehicles and Robots*. Springer, 2011.

[13] V. Matrosov. "On the Stability of Motion," Journal of Applied Mathematics and Mechanics, vol. 26(5), pp. 1337–1353, 1962.

[14] S. Keshmiri, H. Leong, R. Jager and R. Hale. "Modeling and Simulation of the Yak-54 Scaled Unmanned Aerial Vehicle Using Parameter and System Identification," in AIAA Atmospheric Flight Mechanics Conference and Exhibit, pp. 1-15, August 2008.