Quaternion Path-Following in Three Dimensions for a Fixed-Wing UAV using Quaternion Blending

Tom Stian Andersen
Department of Electrical Engineering
UiT The Arctic University of Norway
NO-8514 Narvik, Norway
Email: tom.s.andersen@uit.no

Raymond Kristiansen
Department of Electrical Engineering
UiT The Arctic University of Norway
NO-8514 Narvik, Norway
Email: raymond.kristiansen@uit.no

***Abstract*—This paper presents a three dimensional guidance strategy for fixed-wing UAVs using quaternions. The algorithm is based on constructing two quaternions, one which makes the UAV fly towards the path and one that makes the UAV follow the path. These two quaternions are then blended together such that the path-following objective is reached. The guidance algorithm is applied to a simple kinematic model for a fixed-wing UAV with a simple kinematic controller. Simulations are provided to show the potential of this approach.**

I. Introduction

In the last decades there has been significant impovements in technology which has contributed to the possibility of designing small and cheap autonomous aircraft. The potential applications for these autonomous vehicles are vast, but the increased autonomy means that robust and reliable algorithms needs to be developed to ensure safe operation. The algorithms that generate desired states based on the actual state of the system and a certain mission for the control system to follow are often called guidance algorithms. In most cases the mission to be solved is stated as a geometric mission such as staying on some predefined path termed path-following.

A lot of work has been devoted to path-following within robotics and several algorithms have been developed. One of the most studied guidance methods is the Line-of-Sight (LOS) guidance approach. It has successfully been applied to surface vehicles [1], underwater vehicles [2] and UAVs [3]. In [4] path-following using LOS guidance in 2D was thoroughly analysed and solved in the ideal case of no disturbances. It was also shown how 3D path-following can be divided into a horizontal plane and a vertical plane, effectively reducing the 3D path following problem into two 2D path following problems. This approach for solving the 3D path-following problem has been widely used in literature, for instance in [5], [6] where LOS guidance is incorporated into the dynamics of the vehicles. There has also been considerable research on disturbance rejection for LOS-guidance *cf.* [7], [8] and references therein.

In recent years other guidance algorithms have also been developed. In [9] the authors derive a nonlinear path-following approach adapted from pure-pursuit methods. In [10] an LQR guidance based method is derived for straight line and circular path-following. Another path-following strategy is the vector-field type guidance algorithms [11], [12] and [13] where a vector field for the course angle is made to drive the UAV towards the desired path. A survey of two dimensional guidance algorithms applied to a kinematic model for a fixed-wing UAV can be found in [14]. The vector-field methods have also been extended to 3D in [15] and n-dimensions in [16].

A problem with previous approaches of dividing the guidance into a horizontal and vertical plane is that those methods assumes that there is little cross-coupling between the longitudinal and lateral motion of the UAV which in many cases is not accurate. Additionaly in most of the research on path-following, Euler angles are used to formulate the guidance strategies and seemingly less research has been done to design three dimensional path-following strategies using quaternions. It is desirable to develop guidance algorithms that allow for the coupling between longitudinal and lateral motion. In [17] conventional LOS guidance was converted to quaternion form and in [18] a quaternion path-following controller was derived for acrobatic maneuvers. To the authors best knowledge little work outside of this exists on three dimensional guidance of underactuated vehicles using quaternions.

The contribution of this paper is a three-dimensional quaternion based guidance approach for general paths such as splines. The work is an extension of [19] where a similar guidance approach is developed for straight lines and circles. The proposed algorithm separates the path-following mission into first converging to the path and then following the path and for each task a quaternion is constructed to ensure that the task is fulfilled. The proposed algorithm takes advantage of the fact that summing unit-quaternions representing rotations yields average or blended rotations and the two quaternions are blended together such that the overall mission of path-following is solved. A blending functions that will blend the two quaternions is presented. The proposed method is compared to results presented in [20] and it shows good performance.

This paper is organized in the following way, Section II introduces quaternions, and some useful properties. Section III details the kinematic model for the aircraft. Section IV introduces the quaternion guidance algorithm. In Section V the stability of the proposed guidance method is analysed. In Section VI a simple kinematics controller is presented. In Section VII simulations are performed to show the potential of the approach and a brief conclusion is given in Section VIII.

II. Preliminaries

A. Notation

Vectors are denoted by lower-case bold letters while scalars are non-bold for instance $\boldsymbol{x} \in \mathbb{R}^n$ is an n-dimensional real vector while $a \in \mathbb{R}$ is a real scalar. A positive real scalar number $a > 0$ is written as $a \in \mathbb{R}_+$. The time derivative of a vector is denoted as $\dot{\boldsymbol{x}} = \frac{d\boldsymbol{x}}{dt}$. Superscripts are used to denote the reference frame a vector is expressed in, for instance $\boldsymbol{x}^A \in \mathbb{R}^3$ is a three dimensional vector in frame $\mathcal{F}_A$. The inner-product between two vectors $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^n$ is written as $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^T \boldsymbol{y}$. The euclidian norm is denoted as $\|\boldsymbol{x}\| = \langle \boldsymbol{x}, \boldsymbol{x} \rangle^{\frac{1}{2}}$.

B. Reference frames

NED frame: This frame denoted $\mathcal{F}_n$ has the x-axis pointing north, the y-axis pointing east and the z-axis pointing down completing the right-handed system and it is also assumed to be inertial.

Body frame: This frame denoted $\mathcal{F}_b$ is fixed to the UAV's centre of mass. The x-axis points through the front of the UAV, y-axis is normal to the UAVs plane of symmetry and the z-axis points downwards completing the right handed coordinate reference frame.

Wind frame: This frame denoted $\mathcal{F}_w$ differs from the UAV's body frame by the aerodynamic angles α and β denoting the angle of attack and sideslip angles respectively. The x-axis is aligned with the air-relative velocity vector of the UAV.

Path frame: This coordinate reference frame denoted $\mathcal{F}_p$ has its origin on the desired path at the point closest to the UAV and the orientation of the frame is aligned with the desired path. The path-following goal of the UAV is to align with this frame when it is on the path.

Cross-track frame: This coordinate reference frame denoted $\mathcal{F}_c$ has its origin fixed to the UAV's centre of mass. The x-axis of this frame points towards the point on the path which is closest to the position of the UAV.

C. Quaternions

In this section a brief overview of quaternions are given, for a more complete formulation consult [21] or [22]. Quaternions are well known to offer a singularity free parameterization of a rigid-body's orientation using only four parameters in contrast to nine parameters for the direction cosine matrices. Similary to direction cosine matrices they can also be used to transform vectors between reference frames. A vector $\boldsymbol{u}^b \in \mathbb{R}^3$ can be rotated from $\mathcal{F}_b$ to $\mathcal{F}_n$ by the use of the sandwich product

$$\boldsymbol{u}^n = \boldsymbol{q}_{n,b} \otimes \boldsymbol{u}^b \otimes \boldsymbol{q}^*_{n,b} \tag{1}$$

where the quaternion product for two arbitrary quaternions $\boldsymbol{q}_1 = \begin{bmatrix} q_{1,0} & \boldsymbol{q}_{1,v} \end{bmatrix}$ and $\boldsymbol{q}_2 = \begin{bmatrix} q_{2,0} & \boldsymbol{q}_{2,v} \end{bmatrix}$is defined as

$$\boldsymbol{q}_1 \otimes \boldsymbol{q}_2 = \begin{bmatrix} q_{1,0} q_{2,0} - \boldsymbol{q}_{1,v}^T \boldsymbol{q}_{2,v} \\ q_{1,0} \boldsymbol{q}_{2,v} + q_{2,0} \boldsymbol{q}_{1,v} + \boldsymbol{q}_{1,v} \times \boldsymbol{q}_{2,v} \end{bmatrix}, \tag{2}$$

and $(\cdot)^*$ is the quaternion conjugate $\boldsymbol{q}^* = \begin{bmatrix} q_0 & -\boldsymbol{q}_v \end{bmatrix}^T$. The norm of a quaternion can be defined through the quaternion product as

$$\|\boldsymbol{q}\| = \sqrt{\boldsymbol{q}^* \otimes \boldsymbol{q}}$$

and should always be equal to unity to ensure that lengths are preserved when using (1). Therefore in this paper all quaternions are assumed to belong to $\bar{\mathbb{H}} = \{\boldsymbol{q} \in \mathbb{H} : \|\boldsymbol{q}\| = 1\}$. Several quaternions can also be combined using the quaternion product to represent composite rotations as

$$\boldsymbol{q}_{a,c} = \boldsymbol{q}_{a,b} \otimes \boldsymbol{q}_{b,c}$$

and difference in rotations can be defined as

$$\boldsymbol{q}_{b,d} = \boldsymbol{q}^*_{a,b} \otimes \boldsymbol{q}_{a,d}.$$

The quaternion kinematics is defined as [21]

$$\dot{\boldsymbol{q}}_{a,b} = \frac{1}{2} \boldsymbol{\omega}^a_{a,b} \otimes \boldsymbol{q}_{a,b} = \frac{1}{2} \boldsymbol{q}_{a,b} \otimes \boldsymbol{\omega}^b_{a,b} \tag{3}$$

where $\boldsymbol{\omega}^{(\cdot)}_{a,b} \in \mathbb{R}^3$ is the angular velocity of $\mathcal{F}_b$ relative to $\mathcal{F}_a$, expressed either in $\mathcal{F}_a$ or $\mathcal{F}_b$. It is possible to interpolate between quaternions and many methods have been developed [22]. The simplest interpolation method is linear interpolation, given two quaternions $\boldsymbol{q}_a$ and $\boldsymbol{q}_b$ they can be interpolated as

$$\boldsymbol{q}_c = (1-h)\boldsymbol{q}_a + h\boldsymbol{q}_b \tag{4}$$

where $h \in [0, 1]$. When $h = 0$ the quaternion $\boldsymbol{q}_c$ is equal to $\boldsymbol{q}_a$, but as h tends towards 1 the quaternion $\boldsymbol{q}_c$ will tend towards $\boldsymbol{q}_b$. Since it is a simple linear interpolation $\boldsymbol{q}_c$ will in general not be a unit-quaternion and would have to be normalized.

D. Hermite splines

Given a set of n points $\boldsymbol{p}_i \in \mathbb{R}^3$ and corresponding n tangents $\boldsymbol{m}_i \in \mathbb{R}^3$, $i = 1, 2..., n$ the parametric cubic hermite spline polynomial is defined as

$$\boldsymbol{p}(\theta) = H_1(\theta)\boldsymbol{p}_i + H_2(\theta)\boldsymbol{p}_{i+1} + H_3(\theta)\boldsymbol{m}_i + H_4(\theta)\boldsymbol{m}_{i+1} \tag{5}$$

where $\theta \in [0, 1]$ is the path parameter and the coefficients H_1, H_2, H_3 and H_4 are the hermite coefficients defined as

$$H_1(\theta) = 3\left(\frac{\theta_{i+1} - \theta}{h_i}\right)^2 - 2\left(\frac{\theta_{i+1} - \theta}{h_i}\right)^3 \tag{6}$$

$$H_2(\theta) = 3\left(\frac{\theta - \theta_i}{h_i}\right)^2 - 2\left(\frac{\theta - \theta_i}{h_i}\right)^3 \tag{7}$$

$$H_3(\theta) = -h_i\left(\frac{\theta_{i+1} - \theta}{h_i}\right)^3 + h_i\left(\frac{\theta_{i+1} - \theta}{h_i}\right)^2 \tag{8}$$

$$H_4(\theta) = h_i\left(\frac{\theta - \theta_i}{h_i}\right)^3 - h_i\left(\frac{\theta - \theta_i}{h_i}\right)^2 \tag{9}$$

where $h_i = \theta_{i+1} - \theta_i$. A well known application of hermite splines is to interpolate between a set of n waypoints with $\boldsymbol{W}_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}$ for $i = 1, 2, 3....n-1, n$, such that the global curve is C^1 continuous. To construct a hermite spline interpolated path a set of θ_i needs to be defined such that when inserted into (5) they yield

$$\boldsymbol{p}(\theta_i) = \boldsymbol{W}_i$$

with $\theta_1 = 0$ and $\theta_n = 1$. The length of the path can be approximated as

$$\rho = \sum_{i=1}^{n-1} \|\boldsymbol{W}_{i+1} - \boldsymbol{W}_i\|$$

and can be used to partition the path parameter such as for any $1 \leq m < n$ the path parameter can be calculated as

$$\theta_{m+1} = \theta_m + \frac{\|\boldsymbol{W}_{m+1} - \boldsymbol{W}_m\|}{\rho}.$$

There are several ways to define the tangent vectors for instance in [23] the tangents are defined to perserve monoticity, however in this paper they are defined through the three point difference formula

$$\boldsymbol{m}_i = \frac{1}{2}\left(\frac{\boldsymbol{W}_{i+1} - \boldsymbol{W}_i}{\theta_{i+1} - \theta_i} + \frac{\boldsymbol{W}_i - \boldsymbol{W}_{i-1}}{\theta_i - \theta_{i-1}}\right)$$

with start and end tangents defined as

$$\boldsymbol{m}_1 = \frac{\boldsymbol{W}_2 - \boldsymbol{W}_1}{\theta_2 - \theta_1}$$
$$\boldsymbol{m}_n = \frac{\boldsymbol{W}_n - \boldsymbol{W}_{n-1}}{\theta_n - \theta_{n-1}}.$$

To calculate the tangent at an arbitrary point the derivative with respect to time of (5) is needed which involves the derivative of (6)-(9) with respect to the path parameter. The tangents vector can be calculate as

$$\boldsymbol{m}(\theta) = H_1'(\theta)\boldsymbol{p}_i + H_2'(\theta)\boldsymbol{p}_{i+1} + H_3'(\theta)\boldsymbol{m}_i + H_4'(\theta)\boldsymbol{m}_{i+1} \tag{10}$$

where $H_i'(\theta)$ denotes the derivative of basis function i with respect to the path parameter.

III. Modeling

A kinematic model of a fixed-wing UAV can be expressed using quaternions as

$$\dot{\boldsymbol{p}}^n = \boldsymbol{q}_{n,w} \otimes \boldsymbol{v}^w \otimes \boldsymbol{q}_{n,w}^* \tag{11}$$
$$\dot{\boldsymbol{q}}_{n,w} = \frac{1}{2}\boldsymbol{q}_{n,w} \otimes \boldsymbol{\omega}_{n,w}^w \tag{12}$$
$$\dot{\boldsymbol{\omega}}_{n,w}^w = k_\omega\left(\boldsymbol{\omega}_c^w - \boldsymbol{\omega}_{n,w}^w\right) \tag{13}$$

where $\boldsymbol{p}^n \in \mathbb{R}^3$ is the UAV's inertial position, $\boldsymbol{v}^w \in \mathbb{R}^3$ is the UAV's velocity in the wind frame, $\boldsymbol{q}_{n,w} \in \bar{\mathbb{H}}$ is the quaternion expressing the orientation of the wind frame with respect to $\mathcal{F}_n$, $\boldsymbol{\omega}_{n,w}^w \in \mathbb{R}^3$ is the angular velocity of $\mathcal{F}_w$ relative to $\mathcal{F}_n$ expressed in $\mathcal{F}_w$, $\boldsymbol{\omega}_c^w \in \mathbb{R}^3$ is the commanded angular velocity to be defined later and $k_\omega \in \mathbb{R}_+$ is some constant gain used in (13) to mimic the behaviour of a low-level auto-pilot. It is assumed that there is no wind and the air-relative velocity in the wind frame is defined as

$$\boldsymbol{v}^w = \begin{bmatrix} V & 0 & 0 \end{bmatrix}^T.$$

It is further assumed that the velocity of the UAV is held constant while in flight, $0 < \beta_V \leq V$. The wind frame angular velocity is defined as

$$\boldsymbol{\omega}_{n,w}^w = \begin{bmatrix} p_w & q_w & r_w \end{bmatrix}^T.$$

This kinematic model describes a fixed-wing UAV in 6-DOF motion with four actuators making it underactuated. This implies that there are constraints on the motion of the UAV and it cannot be made to follow arbitrary paths. However, although the UAV is underactuated when viewed in 6-DOF it can be seen as fully actuated if only the rotational motion is considered which is part of the motivation for using quaternions.

IV. Guidance

The guidance objective can be separated into two tasks where the first task is to converge to the path by minimizing the cross-track error. The second task is to follow the path when the UAV is on it. The desired path which a UAV should follow is often a combination of straight lines which are smoothed using piecewise polynomial functions such as hermite splines. Therefore in this section we derive the guidance algorithm to solve the problem of following a spline smoothed path.

A. Converging to path

To ensure that the UAV converges to the path a quaternion representing the cross-track error needs to be defined. The difference between the closest point to the UAV on the path and the UAV position defines a direction which always points towards the path. This direction can be expressed as a quaternion $\boldsymbol{q}_{n,c}$ such that when $\boldsymbol{q}_{n,w}^* \otimes \boldsymbol{q}_{n,c} = \boldsymbol{q}_I$ the UAV is flying towards the path. The closest point on a path to the UAV can be found through the optimization problem

$$\min_{\theta \in [0,1]} \|\boldsymbol{p}^n(\theta) - \boldsymbol{p}^n\|^2 \tag{14}$$

where $\boldsymbol{p}^n(\theta)$ is the path and $\boldsymbol{p}^n$ is the position of the UAV in $\mathcal{F}_n$. The θ_0 which minimizes (14) can be found by using optimization methods such as Newton's method. The initial θ_0 for the optimization algorithm is found by sampling the path and choosing the θ that yields the smallest distance. The cross-track is then defined as the difference between the closest point and the UAV position, i.e.

$$\boldsymbol{p}_e^w = \boldsymbol{q}_{n,w}^* \otimes (\boldsymbol{p}^n(\theta_0) - \boldsymbol{p}^n) \otimes \boldsymbol{q}_{n,w}.$$

The cross-track quaternion should be defined such that

$$\boldsymbol{p}_e^w = \boldsymbol{q}_{n,c} \otimes \boldsymbol{p}_c^c \otimes \boldsymbol{q}_{n,c}^* \tag{15}$$

where $\boldsymbol{p}_c^c = \begin{bmatrix} \|\boldsymbol{p}_e^w\| & 0 & 0 \end{bmatrix}$. This is done by defining the rotation angle

$$\theta_c = \arccos \frac{\boldsymbol{p}_c^{cT}\boldsymbol{p}_e^w}{\|\boldsymbol{p}_e^w\|\|\boldsymbol{p}_c^c\|} \tag{16}$$

and rotation axis

$$\boldsymbol{k}_c = \frac{\boldsymbol{p}_c^c \times \boldsymbol{p}_e^w}{\|\boldsymbol{p}_c^c \times \boldsymbol{p}_e^w\|}. \tag{17}$$

The cross-track quaternion $\boldsymbol{q}_{w,c}$ can then be calculated as

$$\boldsymbol{q}_{w,c} = \begin{bmatrix}\cos\frac{\theta_c}{2} & \boldsymbol{k}_c \sin\frac{\theta_c}{2}\end{bmatrix} \tag{18}$$

where the cross-track quaternion represents the relation between $\mathcal{F}_c$ and $\mathcal{F}_w$. The quaternion $\boldsymbol{q}_{n,c}$ is then found by

$$\boldsymbol{q}_{n,c} = \boldsymbol{q}_{n,w} \otimes \boldsymbol{q}_{w,c}. \tag{19}$$

B. Following path

When the UAV is on the path a quaternion needs to be defined that enables the UAV to follow the path. Since space curves inherently have the Serret-Frenet frame defined at each point on the curve it is a natural choice to use this as a basis for the path quaternion. There are however possible other choices such as rotation minimizing frames [24] and parallel transport frames [25], but in this work Serret-Frenet frames are used because of their simplicity. Only the tangent vector will be used to construct the quaternion since the UAV should be allowed to freely rotate around the tangent vector to generate necessary accelerations. The rotation angle is defined as

$$\theta_p = \arccos\left(\boldsymbol{i}^T \boldsymbol{t}^w\right) \tag{20}$$

and the rotation axis is defined as

$$\boldsymbol{k}_p = \frac{\boldsymbol{i} \times \boldsymbol{t}^w}{\|\boldsymbol{i} \times \boldsymbol{t}^w\|}. \tag{21}$$

where $\boldsymbol{i} = \begin{bmatrix}1 & 0 & 0\end{bmatrix}$ and $\boldsymbol{t}^w$ is the tangent vector of the path at point $\boldsymbol{p}^n(\theta_0)$ expressed in the wind frame. The path quaternion $\boldsymbol{q}_{n,p}$ can then be calculated as

$$\boldsymbol{q}_{w,p} = \begin{bmatrix}\cos\frac{\theta_p}{2} & \boldsymbol{k}_p \sin\frac{\theta_p}{2}\end{bmatrix} \tag{22}$$

which represents the relation between $\mathcal{F}_p$ and $\mathcal{F}_w$. The quaternion $\boldsymbol{q}_{n,c}$ is then found by

$$\boldsymbol{q}_{n,p} = \boldsymbol{q}_{n,w} \otimes \boldsymbol{q}_{w,p}. \tag{23}$$

C. Combining goals

The two quaternions $\boldsymbol{q}_{n,p}$ and $\boldsymbol{q}_{n,c}$ should be blended in such a way that they ensure that the UAV converges to path and then follows it. To blend the quaternions a blending function as specified in (4) of the form

$$\boldsymbol{q}_{n,d} = (\boldsymbol{q}_{n,p} - \boldsymbol{q}_{n,c})(1 - \tanh(k_1\|\boldsymbol{p}_e^n\|)) + \boldsymbol{q}_{n,c} \tag{24}$$

is used, where $k_1 \in \mathbb{R}_+$. The function behaves in the way that when $\|\boldsymbol{p}_e^n\| \to \infty$ then $\tanh(k_1\|\boldsymbol{p}_e^n\|) \to 1$ which implies that $\boldsymbol{q}_{n,d} \to \boldsymbol{q}_{n,c}$ which will guide the UAV towards the path. When the UAV approaches the path $\|\boldsymbol{p}_e^n\| \to 0$ such that $\tanh(k_1\|\boldsymbol{p}_e^n\|) \to 0$ which implies that $\boldsymbol{q}_{n,d} \to \boldsymbol{q}_{n,p}$ so that the UAV follows the path. The gain k_1 can be tuned to give a satisfactory blending between the two quaternions. The error quaternion which represents the rotation from the desired orientation to the UAV orientation is defined as

$$\boldsymbol{q}_e = \boldsymbol{q}_{n,w}^* \otimes \boldsymbol{q}_{n,d}. \tag{25}$$

V. Stability Analysis

Let the error be defined as

$$\boldsymbol{e}^n = \boldsymbol{p}_d^n - \boldsymbol{p}^n \tag{26}$$

with corresponding error kinematics

$$\dot{\boldsymbol{e}}^n = \dot{\boldsymbol{p}}_d^n - \dot{\boldsymbol{p}}^n \tag{27}$$

where the desired velocity is the projection of $\dot{\boldsymbol{p}}^n$ along the path tangent at point $\boldsymbol{p}^n(\theta_0)$. Consider the Lyapunov function candidate

$$V(\boldsymbol{e}^n) = \frac{1}{2}(\boldsymbol{e}^n)^T \boldsymbol{e}^n \tag{28}$$

which is positive definite and radially unbounded. The derivative of (28) along the system trajectories is

$$\dot{V} = (\boldsymbol{e}^n)^T(\dot{\boldsymbol{p}}_d^n - \dot{\boldsymbol{p}}^n). \tag{29}$$

Assume that there is an ideal rotational autopilot that ensures that $\boldsymbol{q}_{n,w} = \boldsymbol{q}_{n,d}$ where the desired quaternion is defined by (24). Inserting (11) into (29) and using $\boldsymbol{q}_{n,w} = \boldsymbol{q}_{n,d}$ the derivative becomes

$$\begin{aligned}\dot{V} =& \boldsymbol{e}^{nT}\big(V\cos\theta \boldsymbol{u}_p^n - (h\boldsymbol{q}_{n,p} + \\ & (1-h)\boldsymbol{q}_{n,c}) \otimes \boldsymbol{v}^d \otimes (h\boldsymbol{q}_{n,p} + (1-h)\boldsymbol{q}_{n,c})^{-1}\big)\end{aligned} \tag{30}$$

with $h = 1 - \tanh(k_1\|\boldsymbol{p}_e^n\|)$. Since $\boldsymbol{e}^n$ represents the vector from the UAV to the closest point on the path it will always be perpendicular to the tangent $\boldsymbol{u}_p^n$ at that poiny and it is therefore obtained

$$\begin{aligned}\dot{V} =& -\boldsymbol{e}^{nT}(h\boldsymbol{q}_{n,p} + (1-h)\boldsymbol{q}_{n,c}) \otimes \boldsymbol{v}^d \otimes (h\boldsymbol{q}_{n,p} + \\ & (1-h)\boldsymbol{q}_{n,c})^{-1}.\end{aligned} \tag{31}$$

When $h \to 0$ it implies that $\|\boldsymbol{e}^n\| \to \infty$ and subsequently the blending yields $\boldsymbol{q}_{n,c} \otimes \boldsymbol{v}^c \otimes \boldsymbol{q}_{n,c}^{-1}$ which is anti-parallel to $\boldsymbol{e}^n$ which implies that $\dot{V}$ is negative definite. When $h \to 1$ it implies that $\|\boldsymbol{e}^n\| \to 0$ and the blending yields $\boldsymbol{q}_{n,p} \otimes \boldsymbol{v}^p \otimes \boldsymbol{q}_{n,p}^{-1}$ which in perpendicular to $\boldsymbol{e}^n$ and the scalar product would give 0, but that implies $\boldsymbol{e}^n = \begin{bmatrix}0 & 0 & 0\end{bmatrix}$ and the UAV is therefore on the path. Whenever $h \in (0,1)$ the quaternion $\boldsymbol{q}_{n,d}$ lies between $\boldsymbol{q}_{n,c}$ and $\boldsymbol{q}_{n,p}$ so that $\boldsymbol{v}^n$ can always be decomposed into an anti-parallel part and an perpendicular part to $\boldsymbol{e}^n$ which implies that $\dot{V}$ is negative.

VI. Kinematic control

To ensure that the UAV aligns itself with the desired orientation an appropriate $\boldsymbol{\omega}_c^w$ should be defined. As this is not the main topic of the paper the following commanded angular velocity is chosen

$$\boldsymbol{\omega}_c^w = k_c \mathrm{sign}(q_0)\boldsymbol{q}_v \tag{32}$$

where $\boldsymbol{q}_e = \begin{bmatrix}q_0 & \boldsymbol{q}_v\end{bmatrix}^T$ is the error quaternion defined in (25) while $k_c \in \mathbb{R}_+$. The control law avoids quaternion unwinding, but it is discontinuous at $q_0 = 0$.

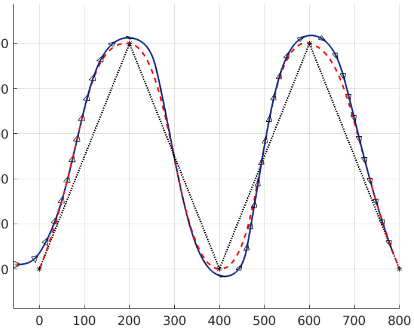

Fig. 1. Top view of the UAV following an hermite spline interpolated path.

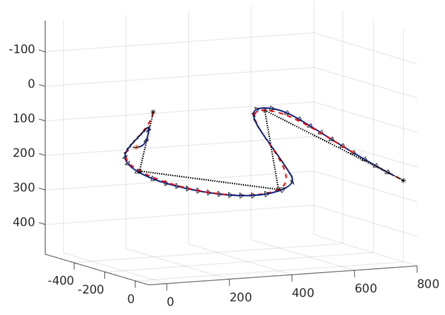

Fig. 2. The waypoints differ in height up to 250 meters.

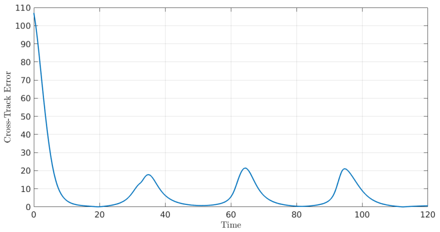

Fig. 3. Cross-track error between the UAV and the interpolated path.

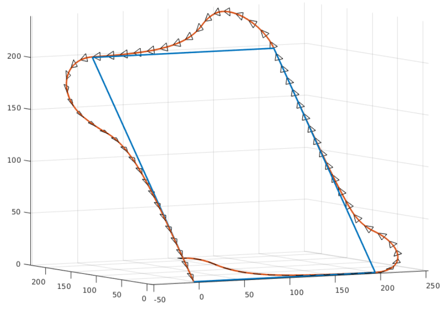

Fig. 4. UAV following path with $k_1 = 0.02$.

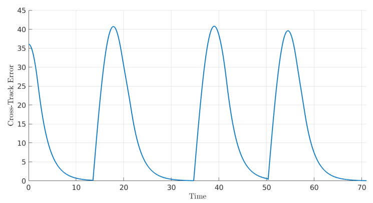

Fig. 5. Cross-track error between the UAV and the path.

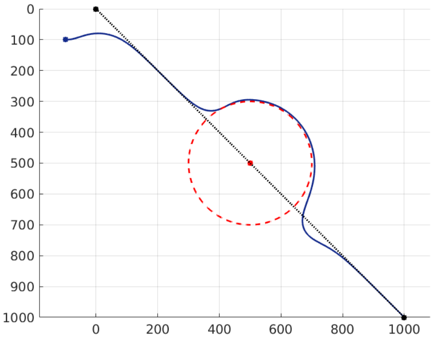

Fig. 6. The guidance method is being used to also provide collision avoidance functionality.

VII. Simulation

A. Scenario 1: Hermite spline interpolated waypoints

In this scenario the UAV is set to follow a hermite intepolated path. The initial conditions of the UAV are set to

$$\boldsymbol{p}^n = \begin{bmatrix}-50 & -10 & 100\end{bmatrix}^T, \quad \boldsymbol{v}^w = \begin{bmatrix}20 & 0 & 0\end{bmatrix}^T$$
$$\boldsymbol{q}_{n,w} = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T, \quad \boldsymbol{w}^w_{n,w} = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T.$$

while the autopilot gain is set to $k_\omega = 2$, the controller gain is set to $k_c = 2$ and the blending factor is set to $k_1 = 0.01$. As can be seen in Figure 1 and Figure 2, the UAV converges to the path and then follows it. In Figure 3 the norm of the position error is shown and it can be seen that the error increases near the waypoints of the path which can also be observed in Figure 1. This is mainly because the UAV is only acting on information provided at the current closest point on the path and to reduce the error some sort of feedforward term is needed in the kinematic controller. There are several ways to add a feedforward term to the kinematic controller for instance by including information about the path in from of the UAV or by formulating an angular velocity of the path that the UAV follows when there is no orientation error.

B. Scenario 2: Comparison with [20]

In [20] several guidance methods such as Lookahead, NLGL, PLOS and Vector Field Guidance were extended to three dimensions and compared. The guidance method in this paper is applied to the comparison scenario of [20] with $k_c = 20$. The kinematics in (13) is omitted and $\boldsymbol{\omega}^w_{n,w}$ is set to be equal to $\boldsymbol{\omega}^w_c$. The angular velocity is also saturated with $|q_w| < 0.19$ rad/s and $|r_w| < 0.33$ rad/sec. The result is seen in Figure 4 with the cross-track error seen in Figure 5. When $k_1 = 0.02$ the root mean squared value is 18.98 while standard deviation is 14.04 which is lower than the methods presented in [20] except for the standard deviation of the Vector Field guidance method being lower. However by lowering $k_1 = 0.01$ the proposed method outperforms Vector Field guidance with an root mean squared value of 20.06 and standard deviation of 12.41. The results however should be taken with some care since the kinematic model used in [20] might not be equivalent to 11-12 in all cases and further studies should be done.

C. Scenario 3: Collision avoidance

In this scenario the UAV is set to follow a desired path in the form of a line which cuts through an obstacle where a keep away circle of radius R_o is centered. To avoid the obstacle an avoidance quaternion $\boldsymbol{q}_{col}$ is constructed and blended with the quaternion from (24)

$$\boldsymbol{q}_{n,d} = (\boldsymbol{q}_{col} - \boldsymbol{q}_{n,d})(1 - \tanh{(k_2 \|\boldsymbol{p}^n - \boldsymbol{p}_o^n\|)}) + \boldsymbol{q}_{n,d} \quad (33)$$

where $\boldsymbol{p}_o^n$ is the closest point on the keep away circle to the UAV. The result is shown in Figure 6 and it is seen that the UAV overshoots the path when it has avoided the obstacle. This is because the UAV is still being influenced by the collision avoidance quaternion as long as it's close to the keep away circle. There are several ways to avoid such behaviours by designing the collosion avoidance quaternion to take this into consideration.

VIII. Conclusion

A guidance algorithm for three dimensional path-following for fixed-wing UAVs using quaternions has been derived for parametric paths. The path-following problem is solved using two quaternions that are blended together such that the UAV converges to and follows a pre-defined path. As seen in VII-C the method can be used for more than just path following and can be classified as a behavioural control method similar to methods such as Null-Spaced Based Behavioral Control [26] and subsumption architecture [27] and as a future work comparisons could be made of the methods. The assumption of no wind in this paper is an unrealistic assumption and future work should include windy conditions.

Acknowledgment

This research has been funded by the Norwegian Research Council and is part of the Arctic Earth Observation and Surveillance Technologies project 195142/I60.

References

[1] K. Y. Pettersen and E. Lefeber, "Way-point tracking control of ships," in *Proceedings of IEEE Conference on Decision and Control*, vol. 1, 2001, pp. 940–945.

[2] M. Breivik and T. I. Fossen, "Guidance-based path following for autonomous underwater vehicles," in *Proceedings of IEEE OCEANS*, 2005, pp. 2807–2814 Vol. 3.

[3] R. Rysdyk, "UAV path following for constant line-of-sight," in *Infotech@Aerospace Conferences*. American Institute of Aeronautics and Astronautics, Sep. 2003, pp. –.

[4] M. Breivik and T. I. Fossen, "Principles of guidance-based path following in 2D and 3D," in *Proceedings of IEEE Conference on Decision and Control*, Dec 2005, pp. 627–634.

[5] Z. Chu and D. Zhu, "3D path-following control for autonomous underwater vehicle based on adaptive backstepping sliding mode," in *Proceedings of IEEE International Conference on Information and Automation*, Aug 2015, pp. 1143–1147.

[6] X. Guo-Qing, Y. Ying, and Z. Wei-Guang, "Path-following in 3D for underactuated AUV in the presence of ocean current," in *Proceedings of International Conference on Measuring Technology and Mechatronics Automation*, Jan 2013, pp. 788–791.

[7] S. Moe, K. Y. Pettersen, T. I. Fossen, and J. T. Gravdahl, "Line-of-sight curved path following for underactuated USVs and AUVs in the horizontal plane under the influence of ocean currents," in *Proceedings of Mediterranean Conference on Control and Automation*, June 2016, pp. 38–45.

[8] A. M. Lekkas and T. I. Fossen, "Integral LOS path following for curved paths based on a monotone cubic hermite spline parametrization," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 6, pp. 2287–2301, Nov 2014.

[9] S. Park, J. Deyst, and J. P. How, "Performance and lyapunov stability of a nonlinear path following guidance method," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 6, pp. 1718–1728, Nov. 2007.

[10] A. Ratnoo, P. Sujit, and M. Kothari, "Adaptive optimal path following for high wind flights," *IFAC*, vol. 44, no. 1, pp. 12 985–12 990, Jan. 2011.

[11] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, "Vector field path following for miniature air vehicles," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 519–529, June 2007.

[12] ——, "Vector field path following for small unmanned air vehicles," in *Proceedings of American Control Conference*, June 2006, pp. 7 pp.–.

[13] E. W. Frew, D. A. Lawrence, C. Dixon, J. Elston, and W. J. Pisano, "Lyapunov guidance vector fields for unmanned aircraft applications," in *Proceedings of American Control Conference*, July 2007, pp. 371–376.

[14] P. B. Sujit, S. Saripalli, and J. B. Sousa, "Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless," *IEEE Control Systems*, vol. 34, no. 1, pp. 42–59, Feb 2014.

[15] Y. Liang, Y. Jia, J. Du, and J. Zhang, "Vector field guidance for three-dimensional curved path following with fixed-wing UAVs," in *Proceedings of American Control Conference*, July 2015, pp. 1187–1192.

[16] V. M. Goncalves, L. C. A. Pimenta, C. A. Maia, B. C. O. Dutra, and G. A. S. Pereira, "Vector fields for robot navigation along time-varying curves in n-dimensions," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 647–659, Aug 2010.

[17] A. M. Lekkas and T. I. Fossen, "A quaternion-based LOS guidance scheme for path following of AUVs," *IFAC*, vol. 46, no. 33, pp. 245–250, 2013.

[18] J. Hall, R. Beard, and T. McClain, "Quaternion control for autonomous path following maneuvers," in *Infotech@Aerospace Conferences*. American Institute of Aeronautics and Astronautics, 2012.

[19] T. S. Andersen and R. Kristiansen, "Path-following in three dimensions using quaternions for a fixed-wing uav," in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, June 2017, pp. 1117–1122.

[20] G. V. Pelizer, N. B. F. da Silva, and K. R. L. J. Branco, "Comparison of 3d path-following algorithms for unmanned aerial vehicles," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2017, pp. 498–505.

[21] O. Egeland and J. T. Gravdahl, *Modeling and Simulation for Automatic Control*. Marine Cybernetics, 2002.

[22] E. B. Dam, M. Koch, and M. Lillholm, "Quaternion, interpolation and animation," University of Copenhagen, Tech. Rep., 1998.

[23] F. N. Fritsch and R. E. Carlson, "Monotone piecwise cubic interpolation," *SIAM*, 1980.

[24] R. T. Farouki, C. Giannelli, C. Manni, and A. Sestini, "Design of rational rotation-minimizing rigid body motions by hermite interpolation," *Mathematics of Computing*, 2012.

[25] A. J. Hanson and H. Ma, "Parallel transport approach to curve framing," Indiana University, Tech. Rep., 1995.

[26] G. Antonelli, F. Arrichiello, and S. Chiaverini, "The null-space-based behavioral control for autonomous robotic systems," *Intelligent Service Robotics*, vol. 1, no. 1, pp. 27–39, Jan 2008.

[27] E. Oland, T. S. Andersen, and R. Kristiansen, "Subsumption architecture applied to flight control using composite rotations," *Automatica*, vol. 69, pp. 195 – 200, 2016.