

Path-following in three dimensions using quaternions for a fixed-wing UAV

Tom Stian Andersen

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

Raymond Kristiansen

Department of Electrical Engineering
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 have been significant improvements in sensor technology and computing power. This has allowed for the development of small, cheap and powerful embedded systems with a wide variety of sensors within robotics. One particular area of robotics which have gathered a significant research interest is Unmanned Aerial Vehicles (UAVs). These flying machines can range from larger fixed-wing aircrafts to smaller multirotors. The potential applications for these autonomous vehicles are vast, but the increased autonomy means that robust and reliable algorithms need to be developed to ensure safe operation. These type of vehicles are often highly nonlinear in their dynamics which motivates the use of nonlinear control for the regulation of their dynamic behaviour. Control algorithms are usually provided with a desired state and the control system's task is to ensure that the actual state of the system converges to the desired state. The algorithms that generate these desired states are often termed guidance algorithms. Based on the actual state of the system and a certain mission the guidance algorithm should generate a desired state for the control system. The mission to be solved can vary, but can often be restated as a geometric mission of staying on some predefined path. This type of guidance is called path-following and is the topic of this paper.

A lot of work has been devoted to path-following within robotics and several algorithms have been developed. One of the most well-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 incorporating into the dynamics of the vehicles. There has also been considerable research on disturbance rejection for LOS-guidance for instance [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 assumes that there is little cross-coupling between the longitudinal and lateral motion of the UAV which in many cases is not accurate. 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. This paper aims to continue the work on defining quaternion based guidance laws in three dimensions. The algorithm proposed in this paper takes advantage of the fact that summing unit-quaternions representing rotations yields average or blended rotations. A blending function inspired by [8] is used to blend the two quaternions. One of the quaternions ensures that the UAV flies towards the closest point on the path, while the other quaternion ensures that the UAV follows the path.

This paper is organized in the following way. Section II

introduces quaternions, which is the main tool in this paper, and some useful properties. Section III details the kinematic model for the aircraft in quaternion form. Section IV derives the quaternion guidance algorithm for straight lines and circles in three dimensions. In Section V simulations are performed to show the potential of the approach and a brief conclusion is given in Section VI highlighting some possible extensions of this work.

II. PRELIMINARIES

A. Notation

Vectors are denoted by lower-case bold letters while scalars are non-bold for instance $\mathbf{x} \in \mathbb{R}^n$ is an n-dimensional vector while $a \in \mathbb{R}$ is a scalar. A positive real scalar number $a > 0$ is written as $a \in \mathbb{R}_+$. The time derivative of a vector is denoted as $\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt}$. Superscripts are used to denote the reference frame a vector is expressed in, for instance $\mathbf{x}^A \in \mathbb{R}^3$ is a three dimensional vector in frame \mathcal{F}_A . Subscripts for vectors and scalars are used to indicate the quantity they represent for instance \mathbf{v}_g^b is ground velocity while \mathbf{v}_r^b would be velocity relative to surrounding air. The inner-product between two vectors $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^n$ is written as $\langle \mathbf{x}, \mathbf{y} \rangle = (\mathbf{x})^T \mathbf{y}$. The euclidian norm is denoted as $\|\mathbf{x}\| = \langle \mathbf{x}, \mathbf{x} \rangle^{\frac{1}{2}}$.

B. Reference frames

a) *NED frame*: This coordinate reference frame denoted \mathcal{F}_n consists of a triad of unit vectors forming a right handed reference system. The x -axis points north, the y -axis points east and the z -axis points down completing the right-handed system. The origin of the NED frame is usually chosen to be at the initial position of the rigid-body we are working with. This can for instance be the take-off point in case of a fixed-wing UAV or pre-defined point in inertial space. The frame is also assumed to be inertial.

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

c) *Path frame*: This coordinate reference frame denoted \mathcal{F}_p has its origin on the desired path 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.

d) *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 [19] or [20]. 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. Similarly to direction cosine matrices they can also be used to transform

vectors between reference frames. A vector $\mathbf{u}^b \in \mathbb{R}^3$ can be rotated from \mathcal{F}_b to \mathcal{F}_n by the use of the sandwich product

$$\mathbf{u}^n = \mathbf{q}_{n,b} \otimes \mathbf{u}^b \otimes \mathbf{q}_{n,b}^* \quad (1)$$

where the quaternion product is defined as

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

and $(\cdot)^*$ is the quaternion conjugate defined as

$$\mathbf{q}^* = [q_0 \quad -q_1 \quad -q_2 \quad -q_3]^T.$$

The norm of a quaternion can be defined through the quaternion product as

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

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 $\bar{\mathbb{H}} = \{\mathbf{q} \in \mathbb{H} : \|\mathbf{q}\| = 1\}$. Several quaternions can also be combined using the quaternion product to represent composite rotations as

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

and difference in rotations can be defined as

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

The quaternion kinematics is defined as [19]

$$\dot{\mathbf{q}}_{a,b} = \frac{1}{2} \boldsymbol{\omega}_{a,b}^a \otimes \mathbf{q}_{a,b} = \frac{1}{2} \mathbf{q}_{a,b} \otimes \boldsymbol{\omega}_{a,b}^b \quad (3)$$

where $\boldsymbol{\omega}_{a,b}^{(\cdot)} \in \mathbb{R}^3$ 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 [20]. The simplest interpolation method is linear interpolation, given two quaternions \mathbf{q}_a and \mathbf{q}_b they can be interpolated in the following way

$$\mathbf{q}_c = (1-h)\mathbf{q}_a + h\mathbf{q}_b$$

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

III. MODELING

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

$$\dot{\mathbf{p}}^n = \mathbf{q}_{n,b} \otimes \mathbf{v}^b \otimes \mathbf{q}_{n,b}^* \quad (4)$$

$$\dot{\mathbf{q}}_{n,b} = \frac{1}{2} \mathbf{q}_{n,b} \otimes \boldsymbol{\omega}_{n,b}^b \quad (5)$$

$$\dot{\boldsymbol{\omega}}_{n,b}^b = k_\omega (\boldsymbol{\omega}_c^b - \boldsymbol{\omega}_{n,b}^b) \quad (6)$$

where $\mathbf{p}^n \in \mathbb{R}^3$ is the UAV's inertial position, $\mathbf{v}^b \in \mathbb{R}^3$ is the UAV's body frame velocity, $\mathbf{q}_{n,b} \in \bar{\mathbb{H}}$ is the quaternion expressing the attitude of the UAV, $\boldsymbol{\omega}_{n,b}^b \in \mathbb{R}^3$ is the UAV's angular velocity relative to \mathcal{F}_n expressed in \mathcal{F}_b , $\boldsymbol{\omega}_c^b \in \mathbb{R}^3$ is the commanded angular velocity to be defined later and $k_\omega \in \mathbb{R}_+$ is some constant gain used in (6) to mimic the behaviour of a

low-level auto-pilot. The UAV is assumed to have an actuator that generates thrust in the body's x -axis only and that the body frame velocity lies entirely along the body's x -axis. This is essentially the same as assuming no wind conditions and the body frame velocity is therefore defined as

$$\mathbf{v}^b = [V \ 0 \ 0]^T.$$

It is further assumed that the velocity of the UAV is held constant and lower bounded while in flight, $0 < \beta_V \leq V$. The UAV is also assumed to have ailerons, a rudder and an elevator for attitude control and the body frame angular velocity is defined as

$$\boldsymbol{\omega}_{n,b}^b = [p \ q \ r]^T.$$

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 desired path which a UAV should follow is often a combination of straight lines and arcs which either interpolates or approximates between some waypoints. An example of an interpolating path that is widely used would be a Dubin's paths, see for instance [21]. Therefore in this section we derive the guidance algorithm to solve these two cases. In later publications more general paths will be studied.

A. Straight-line paths

A straight-line path can be defined using two waypoints $\mathbf{W}_1^n, \mathbf{W}_2^n \in \mathbb{R}^3$ as a line starting at one waypoint and ending in the other. A unit vector containing the direction of the straight-line path can be defined as

$$\mathbf{u}_p^n = \frac{\mathbf{W}_2^n - \mathbf{W}_1^n}{\|\mathbf{W}_2^n - \mathbf{W}_1^n\|} = [u_{px} \ u_{py} \ u_{pz}]^T.$$

Using this direction the course and elevation of the path can be defined as

$$\begin{aligned} \chi_p &= \arctan 2(u_{px}, u_{py}) \\ \gamma_p &= -\arctan 2\left(u_{pz}, \sqrt{(u_{px})^2 + (u_{py})^2}\right). \end{aligned}$$

These two angles can be used to define two quaternions. The first quaternions $\mathbf{q}_{n,\chi}$ defines the orientation needed to follow the path in the xy -plane and is defined as

$$\mathbf{q}_{n,\chi} = \left[\cos \frac{\chi_p}{2} \ 0 \ 0 \ \sin \frac{\chi_p}{2}\right]^T.$$

The second quaternion $\mathbf{q}_{n,\gamma}$ defines the necessary orientation to follow the path along the xz -plane and is defined as

$$\mathbf{q}_{n,\gamma} = \left[\cos \frac{\gamma_p}{2} \ 0 \ \sin \frac{\gamma_p}{2} \ 0\right]^T.$$

These two orientations can then be combined through the quaternion product as

$$\mathbf{q}_{n,p} = \mathbf{q}_{n,\chi} \otimes \mathbf{q}_{n,\gamma}$$

where $\mathbf{q}_{n,p}$ is the path quaternion representing the relation between \mathcal{F}_p and \mathcal{F}_n . If the UAV is on the path and $\mathbf{q}_{n,p}^* \otimes \mathbf{q}_{n,b} = \mathbf{q}_I$ then the UAV is following the path, however if the UAV is not on the path it only implies that the UAV is flying parallel to the 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 position of the closest point on the path to the UAV and the UAV position defines a direction which always points towards the path. This direction can be expressed as a quaternion $\mathbf{q}_{n,c}$ such that when $\mathbf{q}_{n,c}^* \otimes \mathbf{q}_{n,b} = \mathbf{q}_I$ the UAV is flying towards the path. The closest point on a path to the UAV when the path is a straight-line is the projection of the UAV position onto the path. The cross-track is then defined as the difference between the closest point and the UAV position

$$\mathbf{p}_e^n = \left((\mathbf{p}^n - \mathbf{W}_1^n)^T \mathbf{u}_p^n\right) \mathbf{u}_p^n + \mathbf{W}_1^n - \mathbf{p}^n.$$

Using the cross-track error two new angles can be defined as

$$\begin{aligned} \chi_c &= \arctan 2(\mathbf{p}_{ey}^n, \mathbf{p}_{ex}^n) \\ \gamma_c &= -\arctan 2\left(\mathbf{p}_{ez}^n, \sqrt{(\mathbf{p}_{ex}^n)^2 + (\mathbf{p}_{ey}^n)^2}\right) \end{aligned}$$

which allows us to define two new quaternions

$$\mathbf{q}_{n,\chi} = \left[\cos \frac{\chi_c}{2} \ 0 \ 0 \ \sin \frac{\chi_c}{2}\right]^T$$

and

$$\mathbf{q}_{n,\gamma} = \left[\cos \frac{\gamma_c}{2} \ 0 \ \sin \frac{\gamma_c}{2} \ 0\right]^T.$$

These two quaternions can be combined as

$$\mathbf{q}_{n,c} = \mathbf{q}_{n,\chi} \otimes \mathbf{q}_{n,\gamma}$$

where $\mathbf{q}_{n,c}$ is the cross-track quaternion representing the relation between \mathcal{F}_c and \mathcal{F}_n . The two quaternions $\mathbf{q}_{n,p}$ and $\mathbf{q}_{n,c}$ 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 inspired from [8] of the form

$$\mathbf{q}_{n,d} = (\mathbf{q}_{n,p} - \mathbf{q}_{n,c}) e^{-k_1 \|\mathbf{p}_e^n\|} + \mathbf{q}_{n,c} \quad (7)$$

is used, where $k_1 \in \mathbb{R}_+$. This function behaves in the way that when $\|\mathbf{p}_e^n\| \rightarrow \infty$ then $e^{-k_1 \|\mathbf{p}_e^n\|} \rightarrow 0$ which implies that $\mathbf{q}_{n,d} \rightarrow \mathbf{q}_{n,c}$ which will guide the UAV towards the path. When the UAV approaches the path $\|\mathbf{p}_e^n\| \rightarrow 0$ such that $e^{-k_1 \|\mathbf{p}_e^n\|} \rightarrow 1$ which implies that $\mathbf{q}_{n,d} \rightarrow \mathbf{q}_{n,p}$ so that the UAV follows the path. The gain k_1 can be tuned to give a satisfactory blending between the two quaternions. A high k_1 will result in a faster convergence towards the path, but will cause overshoots. While a smaller k_1 results in a slower convergence towards the path, but a smoother transition with less overshoot. Using the desired orientation and the UAV orientation we can define the error as

$$\mathbf{q}_e = \mathbf{q}_{n,d}^* \otimes \mathbf{q}_{n,b}. \quad (8)$$

Remark: After the blending step in (7) the desired quaternion $\mathbf{q}_{n,d}$ should be normalized to ensure unit length.

B. Orbit following

A point on a circle is given as

$$\mathbf{p}_o^n(\theta) = \mathbf{c}^n + [R_c \cos \theta \quad R_c \sin \theta \quad 0]^T. \quad (9)$$

where $\mathbf{c}^n \in \mathbb{R}^3$ is the position of the circle's center and $R_c \in \mathbb{R}_+$ is the circle radius. The angle that gives the closest point to the UAV can be found as the angle between the north direction and the line from the circle center to the UAV

$$\chi_c = \arctan 2(p_y - c_y, p_x - c_x). \quad (10)$$

Inserting (10) into (9) gives us the closest point on the circle to the UAV, from which we obtain the cross-track error expressed in \mathcal{F}_n as

$$\mathbf{p}_e^n = \mathbf{p}_c^n(\chi_c) - \mathbf{p}^n.$$

From the cross-track error the quaternion that will let the UAV approach the circle with angles

$$\begin{aligned} \chi_c &= \arctan 2(\mathbf{p}_{ey}^n, \mathbf{p}_{ex}^n) \\ \gamma_c &= -\arctan 2\left(\mathbf{p}_{ez}^n, \sqrt{(\mathbf{p}_{ex}^n)^2 + (\mathbf{p}_{ey}^n)^2}\right). \end{aligned}$$

is obtained from the combination of the two quaternions

$$\begin{aligned} \mathbf{q}_{n,\chi} &= [\cos \frac{\chi_c}{2} \quad 0 \quad 0 \quad \sin \frac{\chi_c}{2}]^T \\ \mathbf{q}_{n,\gamma} &= [\cos \frac{\gamma_c}{2} \quad 0 \quad \sin \frac{\gamma_c}{2} \quad 0]^T \end{aligned}$$

using the quaternion product

$$\mathbf{q}_{n,c} = \mathbf{q}_{n,\chi} \otimes \mathbf{q}_{n,\gamma}.$$

When the UAV is on the circle it should follow the orbit tangent at the point $\mathbf{p}_o^n(\chi_c)$ defined as

$$\mathbf{p}_t^n(\chi_c) = [-\sin \chi_c \quad \cos \chi_c \quad 0]^T$$

since the circle does not have any z-component there is no need for an elevation quaternion, also since only the direction is of interest the tangent vector is normalized. The path course then becomes

$$\chi_p = \arctan 2(p_{ty}^n, p_{tx}^n)$$

from which the path quaternion is found as

$$\mathbf{q}_{n,p} = [\cos \frac{\chi_p}{2} \quad 0 \quad 0 \quad \sin \frac{\chi_p}{2}]^T.$$

The desired quaternion can then be defined in the same way as (7)

$$\mathbf{q}_{n,d} = (\mathbf{q}_{n,p} - \mathbf{q}_{n,c}) e^{-k_1 \|\mathbf{p}_e^n\|} + \mathbf{q}_{n,c} \quad (11)$$

which is used to find the error quaternion

$$\mathbf{q}_e = \mathbf{q}_{n,d}^* \otimes \mathbf{q}_{n,b}. \quad (12)$$

Remark: The course angles should be calculated such that it takes into consideration the UAV's orientation. The course angles should be calculated as detailed in [21]

$$\chi(\cdot) = \arctan 2((\cdot), (\cdot)) + 2\pi m$$

where $m \in \mathbb{N}$ ensures that $-\pi \leq \chi(\cdot) - \chi \leq \pi$ and χ is the UAV's course angle.

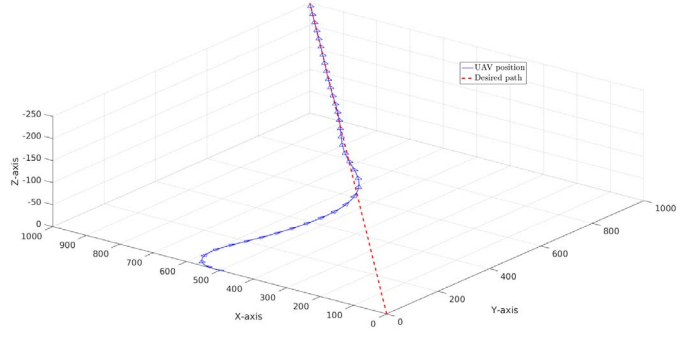


Fig. 1. UAV converging to and following a straight line.



Fig. 2. Distance error for straight line following.

C. Kinematic control

To ensure that the UAV aligns itself with the desired orientation an appropriate ω_c^b should be defined. As this is not the main topic of the paper and because of space constraints the following commanded angular velocity is chosen without further justification

$$\omega_c^b = -k_c q_0 \mathbf{q}_v \quad (13)$$

where $\mathbf{q}_e = [q_0 \quad \mathbf{q}_v]^T$ is the error quaternion defined in (8) and (12) while $k_c \in \mathbb{R}_+$. Eventhough the controller will ensure convergence to the path in most cases, a complete analysis of the kinematic control of UAVs using quaternions should be performed and it will be the topic of a later publication.

V. SIMULATION

This section shows the potential of the guidance approach using three scenarios. In the first scenario the UAV should converge and follow a straight-line path and in the second and third scenario the mission is to converge to and follow an orbit path. The kinematics of (4)-(6) together with the kinematic controller defined in (13) are simulated with the parameters $V = 25$, $k_\omega = 1$, $k_1 = 0.01$, $k_c = 1$ and $R_c = 600$.

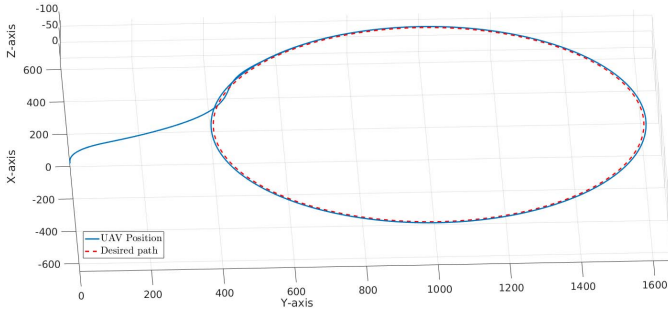


Fig. 3. UAV converging to and following a circle.

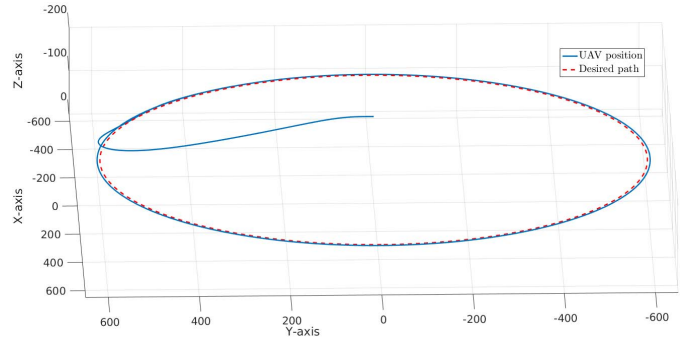


Fig. 5. UAV converging to and following a circle.

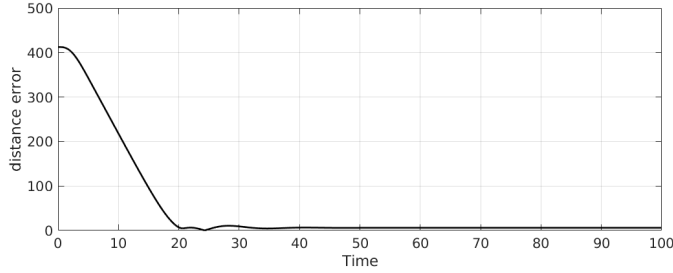


Fig. 4. Distance error for loiter path following.

A. Scenario 1: Straight-line path

In this scenario the initial conditions are set to

$$\begin{aligned} \mathbf{p}^n &= [500 \ 0 \ 0]^T \\ \mathbf{q}_{n,b} &= [1 \ 0 \ 0 \ 0]^T \\ \mathbf{w}_{n,b}^b &= [0 \ 0 \ 0]^T \\ \mathbf{v}^b &= [V \ 0 \ 0]^T \\ \mathbf{W}_1^n &= [0 \ 0 \ 0]^T \\ \mathbf{W}_2^n &= [1000 \ 1000 \ -250]^T. \end{aligned}$$

As can be seen in Figure 1 the UAV reaches the straight line path and follows it. In the simulation there is a significant overshoot which in some cases is undesirable, the gains in the simulations could be adjusted to try and minimize the overshoot as well as a different kinematic controller which includes a dampening term. In Figure 2 the distance error between the path and the UAV is plotted. It can be seen that it converges towards zero but has a long settling time. It can also be seen that the distance error increases in the beginning which is because the UAV is initially pointing away from the path and it needs to initiate a turn to fly towards the path.

B. Scenario 2: Orbit path

In this scenario the initial conditions are set to

$$\begin{aligned} \mathbf{p}^n &= [0 \ 0 \ 0]^T \\ \mathbf{q}_{n,b} &= [1 \ 0 \ 0 \ 0]^T \\ \mathbf{w}_{n,b}^b &= [0 \ 0 \ 0]^T \\ \mathbf{v}^b &= [V \ 0 \ 0]^T \\ \mathbf{c}^n &= [0 \ 1000 \ -100]^T. \end{aligned}$$

As can be seen in Figure 3 the UAV converges towards the circle and follows it. From Figure 4 it can be seen that the distance error never reaches zero but a constant value (7.8 meters in this case) which is because the path quaternion is defined from the tangent of the closest point on the circle to the UAV. The UAV needs some time to converge to this path quaternion and by the time this has happened the UAV has moved slightly away from the circle. The distance error could be reduced for instance by choosing a tangent vector that lies a bit ahead of the UAV so that it has time to converge to the path quaternion.

C. Scenario 3: Orbit path

In this scenario the initial conditions are set to

$$\begin{aligned} \mathbf{p}^n &= [0 \ 0 \ -200]^T \\ \mathbf{q}_{n,b} &= [1/2 \ 1/2 \ 1/2 \ 1/2]^T \\ \mathbf{w}_{n,b}^b &= [0 \ 0 \ 0]^T \\ \mathbf{v}^b &= [V \ 0 \ 0]^T \\ \mathbf{c}^n &= [0 \ 0 \ -100]^T. \end{aligned}$$

Here the UAV starts at the center of the circle such that there is no defined closest point. Since the closest point is calculated based on angle the line from the center of the circle to the UAV the angle becomes zero when the UAV is at the circle center. Therefore the UAV will calculate the closest point to be at $\mathbf{p}_c^n(0)$. This forces the UAV to leave the circle's center and when that happens the closest point is again defined. In this scenario there is also a constant distance error as explained in the previous scenario.

VI. CONCLUSION

A guidance algorithm for three dimensional path-following for fixed-wing UAVs using quaternions has been derived for straight line paths and orbits. The algorithm works in three stages, first the tangent vector to the path is found and a path quaternion is defined from it, then the closest point is found from the UAV to the desired path which defines the cross-track error and a cross-track quaternion is defined from it. Lastly the two quaternions are blended together such that they complete the goal of converging to and following a pre-defined path. The approach should be generalizable to arbitrary parametric paths and part of the future work will be to detail this generalization. One of the key assumptions in this paper was the absence of wind which is an unrealistic assumption and future work should include these type of 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 vehicles," *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] O. Egeland and J. T. Gravdahl, *Modeling and Simulation for Automatic Control*. Marine Cybernetics, 2002.
- [20] E. B. Dam, M. Koch, and M. Lillholm, "Quaternion, interpolation and animation," University of Copenhagen, Tech. Rep., 1998.
- [21] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012.