Bulgarian Academy of Sciences. Space Research and Technology Institute.
Aerospace Research in Bulgaria. 28, 2016, Sofia

QUATERNION-BASED AUTOPILOT FOR DODECACOPTERS - PART I

Svetoslav Zabunov

e-mail: SvetoslavZabunov@gmail.com

Abstract

The innovations in modern unmanned aerial vehicles do pose higher requirements against the autopilot aircraft control. Special demand is placed by the multirotor innovative helicopters for their unique control system and rotor positions.

The current article establishes the core of a quaternion based autopilot suitable for the innovative and award winning twelve rotor UAV helicopter Bulgarian Knight. Quaternions offer a number of benefits to autopilot systems, but their implementation to specialized autopilots used in innovative and unique drone models require exclusive attention and discussion. As a result, an efficient and flexible autopilot is attained, because quaternion computations are much faster and accurate than the other competing approaches. Nevertheless, a quaternion-based autopilot requires sophisticated software libraries with inherent significant complexity. The elevated accuracy and pliability of the quaternion method is a fertile means for developing a prototype, scientific and research autopilot that is suitable for customization in response to the novel UAVs specific needs.

Notation legend:

$\vec{a}$	Vectors are denoted with italic letters and an arrow above.
$\mathbf{a}$	Quaternions are denoted with bold letters.
$\mathbf{0}$	Zero quaternion.
$\mathbf{1}$	Identity quaternion.
$\mathbb{A}$	Matrices are denoted by blackboard bold letters.
$\mathbb{0}$	Zero matrix.
$\mathbb{1}$	Identity matrix.

Introduction

The modern era of unmanned aerial vehicles (UAVs) presumes highly robotized autonomous flying machines, controlled by an onboard powerful computer in their flight maneuvers and actions. The foremost control process of the aircraft is the flight control, which is carried out by the autopilot. Almost all autopilots nowadays are constructed using quaternion algebra and quaternion analysis, due to the unquestionable performance benefits of this mathematical apparatus [1, 2].

The more complex the drones become, the more sophisticated the autopilots have to be in order to respond to the higher demands of control manipulations the newer UAVs encounter [3]. In pursue of higher stability, invulnerability, reliability, efficiency, lower noise, either mechanical or electromagnetic, and higher safety, a larger number of rotors is often advised. Among the consumer multirotor drones the highest rotor number often is eight and the airframe topology used is the classic "star" topology. Neither the number of eight for the rotors, nor the "star" topology is optimal in terms of the above sought benefits. Of course, as the number of rotors increases, gradually establishes a prohibitively elevated complexity of the machine thus overthrowing the benefits of the greater number of rotors. There is a 'sweet spot' and it begins at twelve rotors, because 12 is the lowest even number of rotors a multirotor helicopter must have in order to implement an airframe structure with optimal geometric covering (figure 1). Going from twelve rotors up the optimal geometric covering may be preserved, but the complexity of the flying machine will go higher and the airframe normalized weight will increase. The airframe normalized weight is the airframe total weight divided by the number of rotors.

This article focuses on the quaternion mathematical apparatus utilized in the innovative twelve-rotor drone helicopter – *The Bulgarian Knight* (Fig. 1).

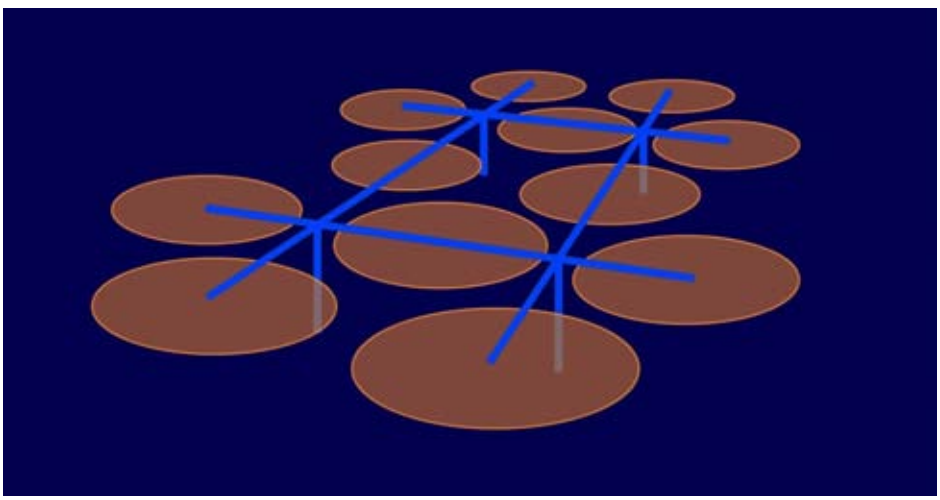

Fig. 1. Bulgarian Knight award winning dodecacopter

Quaternion analysis suitability to autopilot systems

Quaternions are used mostly to present rotations, but may be applied to the whole mathematical apparatus for computations including all mechanical models

the autopilot implements such as the 3D geometric model, 3D kinematic model, 3D dynamic model, and so on [4].

Nevertheless, the rotation matrix is the most common mathematical object used to hold spatial rotations in the three dimensional Euclidean space as described in [5]. There are other means as well, such as Euler angles and quaternions [6]. The latter have certain well pronounced benefits over the other approaches such as:

1. A smaller number of scalars used to describe the rotation, compared to a matrix form: 4 instead of 9 scalars. Thus the rotation quaternion consumes less computer memory.
2. Smaller number of mathematical operations required to calculate a rotation compared to the matrix form [7].
3. Easier to normalize than a rotation matrix.
4. Gimbal lock is not present as is the case with Euler angles.
5. Rotation quaternion exhibits slower degradation due to accumulation of numerical errors than the rotation matrix [8]. The degradation raises distortion of the rotation matrix orthogonality.
6. The transformation from rotation quaternion to another representation is fast and comfortable. This is not the case with rotation matrix and Euler angles [9].
7. When a rotation matrix is irreplaceable, the transformations between rotation quaternion and rotation matrix are convenient and reasonably fast [10].

All the above advantages gain the predilection for quaternions as the method of choice used to realize spatial rotations in autopilot systems, especially in autopilots with high level of safety, efficiency and computing accuracy. Hence, in most modern autopilot systems, from small drones to large airplanes the preferred mathematical method is quaternions.

Quaternion prerequisites

Quaternions were introduced by W. Hamilton in the year of 1843 [11]. Later, vector analysis followed as a result of quaternion analysis simplification [12]. The next paragraphs will briefly summarize the quaternion mathematical basic principles to aid the reader in comprehending all formulations about autopilot algorithms presented in the following sections of the current paper [13]. For deduction of some of the formulas and thorough examination of quaternion properties the reader may consult the following books on quaternions [5, 14–16].

A quaternion is defined through three objects called fundamental quaternion units: $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$. There is a strong resemblance to complex numbers, but a quaternion has four elements instead of two. Hamilton defined quaternions as follows [11]:

(1) $\mathbf{a} = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$

In (1) $\mathbf{a}$ is a quaternion with elements w, x, y, and z. Other notations are: $(w, x, y, z) \Leftrightarrow [w, \vec{v}] \Leftrightarrow w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$, where $\vec{v}$ is a three-dimensional vector (x, y, z).

Note that lowercase bold is used to denote quaternions in this article. The basis elements themselves are quaternions. The basis element scalar 1 is the identity element and is represented by the identity quaternion:

(2) $\mathbf{1} = (1, 0, 0, 0) = [1, 0] = 1$

The zero quaternion is a quaternion with all elements zeros:

(3) $\mathbf{0} = (0, 0, 0, 0) = [0, 0] = 0$

A quaternion is invariant under multiplication by 1. Quaternion multiplication is defined through the products of the basis elements:

(4) $\mathbf{ii} = \mathbf{jj} = \mathbf{kk} = \mathbf{ijk} = -1$

This equation was carved by Hamilton on a stone on Brougham Bridge over the Royal Canal in Dublin on October 16th, 1843. It holds all the essence of quaternions. From (2) it is easy to derive all combinations of basis products:

(5) $\mathbf{ij} = \mathbf{k}$, $\mathbf{jk} = \mathbf{i}$, $\mathbf{ki} = \mathbf{j}$

(6) $\mathbf{ji} = -\mathbf{k}$, $\mathbf{kj} = -\mathbf{i}$, $\mathbf{ik} = -\mathbf{j}$

From (4), (5), and (6) the general multiplication of two quaternions is derived:

(7) $$\mathbf{qq}' = (w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k})(w' + x'\mathbf{i} + y'\mathbf{j} + z'\mathbf{k}) = [ww' - \vec{v}\vec{v}', w\vec{v}' + w'\vec{v} + \vec{v} \times \vec{v}']$$

Quaternions addition and multiplication operations obey the associative law:

(8) $(\mathbf{q} + \mathbf{q}') + \mathbf{q}'' = \mathbf{q} + (\mathbf{q}' + \mathbf{q}'')$

(9) $(\mathbf{qq}')\mathbf{q}'' = \mathbf{q}(\mathbf{q}'\mathbf{q}'')$

These two operations obey also the distributive law:

(10) $\mathbf{q}(\mathbf{q}' + \mathbf{q}'') = \mathbf{q}\mathbf{q}' + \mathbf{q}\mathbf{q}''$

Further, the addition law is commutative:

(11) $\mathbf{q} + \mathbf{q}' = \mathbf{q}' + \mathbf{q}$

For the product of a quaternion with itself one obtains:

(12) $\mathbf{q}^2 = [w^2 - \vec{v}^2, 2w\vec{v}]$

One should notice that the quaternion product is non-commutative:

(13) $\mathbf{q}\mathbf{q}' \neq \mathbf{q}'\mathbf{q}$

The quaternion conjugate is defined as:

(14) $\tilde{\mathbf{q}} = w - x\mathbf{i} - y\mathbf{j} - z\mathbf{k} = [w, -\vec{v}]$

The product of a quaternion with its conjugate has similar properties to complex numbers:

(15) $\mathbf{q}\tilde{\mathbf{q}} = \tilde{\mathbf{q}}\mathbf{q} = [ww + \vec{v}\vec{v}, -w\vec{v} + w\vec{v}] = [w^2 + \vec{v}^2, 0] = w^2 + x^2 + y^2 + z^2$

And also the following rule holds:

(16) $(\mathbf{q}\mathbf{p})^{\sim} = \tilde{\mathbf{p}}\tilde{\mathbf{q}}$

Quaternion norm is used to keep a unit rotation quaternion stable:

(17) $|\mathbf{q}| = \sqrt{\mathbf{q}\tilde{\mathbf{q}}} = \sqrt{\tilde{\mathbf{q}}\mathbf{q}} = \sqrt{w^2 + x^2 + y^2 + z^2} = |\tilde{\mathbf{q}}|$

A multiplicative law is applicable to the norm:

(18) $|\mathbf{p}\mathbf{q}| = |\mathbf{p}||\mathbf{q}| = |\mathbf{q}||\mathbf{p}| = |\mathbf{q}\mathbf{p}|$

The unit quaternion is defined as the quaternion divided by its norm. It is also called a versor:

$$\mathbf{n_q} = \frac{\mathbf{q}}{|\mathbf{q}|} \tag{19}$$

The unit quaternion has a unit norm:

$$|\mathbf{n_q}| = \mathbf{1} \tag{20}$$

With finite rotations (they shall be called just rotations from now on), the order they are applied to a vector matters. Rotations are naturally expressed by algebraic systems with non-commutative products. At first glance, the non-commutability would affect the quaternion reciprocal:

$\mathbf{q}\mathbf{q}_r^{-1} = \mathbf{q}_l^{-1}\mathbf{q} = \mathbf{1}$ and it is expected that $\mathbf{q}_r^{-1} \neq \mathbf{q}_l^{-1}$

But taking into view (15) it is observed that:

$$\frac{\mathbf{q}\tilde{\mathbf{q}}}{|\mathbf{q}|^2} = \frac{\tilde{\mathbf{q}}\mathbf{q}}{|\mathbf{q}|^2} = \mathbf{1} \Rightarrow \mathbf{q}^{-1} = \mathbf{q}_l^{-1} = \mathbf{q}_r^{-1} = \frac{\tilde{\mathbf{q}}}{|\mathbf{q}|^2} \tag{21}$$

An important property of the versor is that its reciprocal is equal to its conjugate. This formula follows from (17) and (21):

$$\mathbf{n}_\mathbf{q}^{-1} = \left(\frac{\mathbf{q}}{|\mathbf{q}|}\right)^{-1} = |\mathbf{q}|\mathbf{q}^{-1} = |\mathbf{q}|\frac{\tilde{\mathbf{q}}}{|\mathbf{q}|^2} = \frac{\tilde{\mathbf{q}}}{|\mathbf{q}|} = \frac{\tilde{\mathbf{q}}}{|\tilde{\mathbf{q}}|} = \tilde{\mathbf{n}}_\mathbf{q} \tag{22}$$

According to the above properties of quaternions and taking into account that for any nonzero quaternion $\mathbf{q}$, there is a quaternion $-\mathbf{q}$ such that $\mathbf{q} + (-\mathbf{q}) = \mathbf{0}$ and the non-commutative law it follows that quaternions are a non-commutative division ring. If quaternions had a commutative product they would have been a field.

As mentioned above, the three dimensional rotation is an important procedure in autopilot systems. The rotation is defined as follows:

(23) $$\mathbf{q}_R = \left[\cos\frac{\theta}{2}, \vec{n}\sin\frac{\theta}{2}\right]$$

In (23) $\vec{n}$ is a unit vector specifying the axis of rotation and θ is the angle of rotation. It should be noted that the rotation quaternion is a versor [17]:

(24) $$\left|\mathbf{q}_R\right| = \sqrt{\mathbf{q}_R\tilde{\mathbf{q}}_R} = \sqrt{\cos^2\frac{\theta}{2} + \sin^2\frac{\theta}{2}\left(x^2 + y^2 + z^2\right)} = \\ = \sqrt{\cos^2\frac{\theta}{2} + \sin^2\frac{\theta}{2}} = \sqrt{1} = 1$$

Immediately follows that the reciprocal of the rotation quaternion is its conjugate:

(25) $$\mathbf{q}_R^{-1} = \tilde{\mathbf{q}}_R$$

The rotation operator on a vector using a rotation quaternion is as follows:

(26) $$\mathbf{q}_R\mathbf{v}\tilde{\mathbf{q}}_R = \mathbf{q}_R\mathbf{v}\mathbf{q}_R^{-1}$$

Here quaternion $\mathbf{v} = \left[0, \vec{v}\right]$ represents the vector that is to be rotated. Changing the sign of the rotation quaternion gives the same rotation:

(27) $$-\mathbf{q}_R = \left[-\cos\frac{\theta}{2}, -\vec{n}\sin\frac{\theta}{2}\right] = \left[\cos\left(\pi + \frac{\theta}{2}\right), \vec{n}\sin\left(\pi + \frac{\theta}{2}\right)\right] = \\ = \left[\cos\left(\frac{2\pi + \theta}{2}\right), \vec{n}\sin\left(\frac{2\pi + \theta}{2}\right)\right]$$

Obviously, quaternion $-\mathbf{q}_R$ rotates by an angle of $2\pi + \theta$, but this is the same rotation as rotating by θ, because $2\pi = 360^o$ adds full turn to the rotation and does not act as a transformation. To perform an inverse rotation to an angle of $-\theta$ one should use the conjugate of the rotation quaternion:

(28) $$\tilde{\mathbf{q}}_R\mathbf{v}\mathbf{q}_R = \mathbf{q}_R^{-1}\mathbf{v}\mathbf{q}_R$$

Applying two consecutive rotations $\mathbf{p}_R$ and $\mathbf{q}_R$ yields:

(29) $$\mathbf{p}_R\mathbf{q}_R\mathbf{v}\tilde{\mathbf{q}}_R\tilde{\mathbf{p}}_R = \mathbf{p}_R\mathbf{q}_R\mathbf{v}\left(\mathbf{p}_R\mathbf{q}_R\right)^{\tilde{}} = \mathbf{t}_R\mathbf{v}\tilde{\mathbf{t}}_R$$

The new composite rotation is $\mathbf{t}_R = \mathbf{p}_R\mathbf{q}_R$. The rotation of a product of two vectors, represented as quaternions, is equal to the product of the rotations of these two vectors, represented as quaternions:

(30) $$\mathbf{q}_R\mathbf{ab}\tilde{\mathbf{q}}_R = \mathbf{q}_R\mathbf{a}\tilde{\mathbf{q}}_R\mathbf{q}_R\mathbf{b}\tilde{\mathbf{q}}_R$$

Autopilots perform numerical integration according to their kinematic and dynamic models [18]. For this purpose differentiation of the rotation quaternion is required. The derivative of a rotation quaternion, in respect to time, is as follows [19]:

(31) $$\frac{d\mathbf{q}_R}{dt} = \dot{\mathbf{q}}_R = \frac{1}{2}\boldsymbol{\omega}\mathbf{q}_R$$

Quaternion $\boldsymbol{\omega} = \left[0, \vec{\omega}\right]$. Vector $\vec{\omega}$ is the vector of the angular velocity. By differentiating (31), the second derivative may be obtained as follows:

(32) $$\ddot{\mathbf{q}}_R = \frac{1}{2}\dot{\boldsymbol{\omega}}\mathbf{q}_R + \frac{1}{2}\boldsymbol{\omega}\dot{\mathbf{q}}_R = \frac{1}{2}\boldsymbol{\varepsilon}\mathbf{q}_R + \frac{1}{4}\boldsymbol{\omega}^2\mathbf{q}_R$$

Quaternion $\boldsymbol{\varepsilon} = \left[0, \vec{\varepsilon}\right]$ represents the vector of the angular acceleration $\vec{\varepsilon}$. The derivative product rule in regard to time holds for quaternions. One should notice that in the general case of product of two quaternion functions with quaternion arguments the derivative product rule in regard to a quaternion variable does not hold [20].

Finally, the autopilot control calculations may require at certain places transformation from rotation matrix to rotation quaternion and vice versa. The transformation from rotation quaternion to rotation matrix is performed according to the next formula [21]:

(33) $$\mathbb{R} = \begin{bmatrix} w^2 + x^2 - y^2 - z^2 & 2xy - 2wz & 2xz + 2wy \\ 2xy + 2wz & w^2 - x^2 + y^2 - z^2 & 2yz - 2wx \\ 2xz - 2wy & 2yz + 2wx & w^2 - x^2 - y^2 + z^2 \end{bmatrix} = \begin{bmatrix} R_{00} & R_{01} & R_{02} \\ R_{10} & R_{11} & R_{12} \\ R_{20} & R_{21} & R_{22} \end{bmatrix}$$

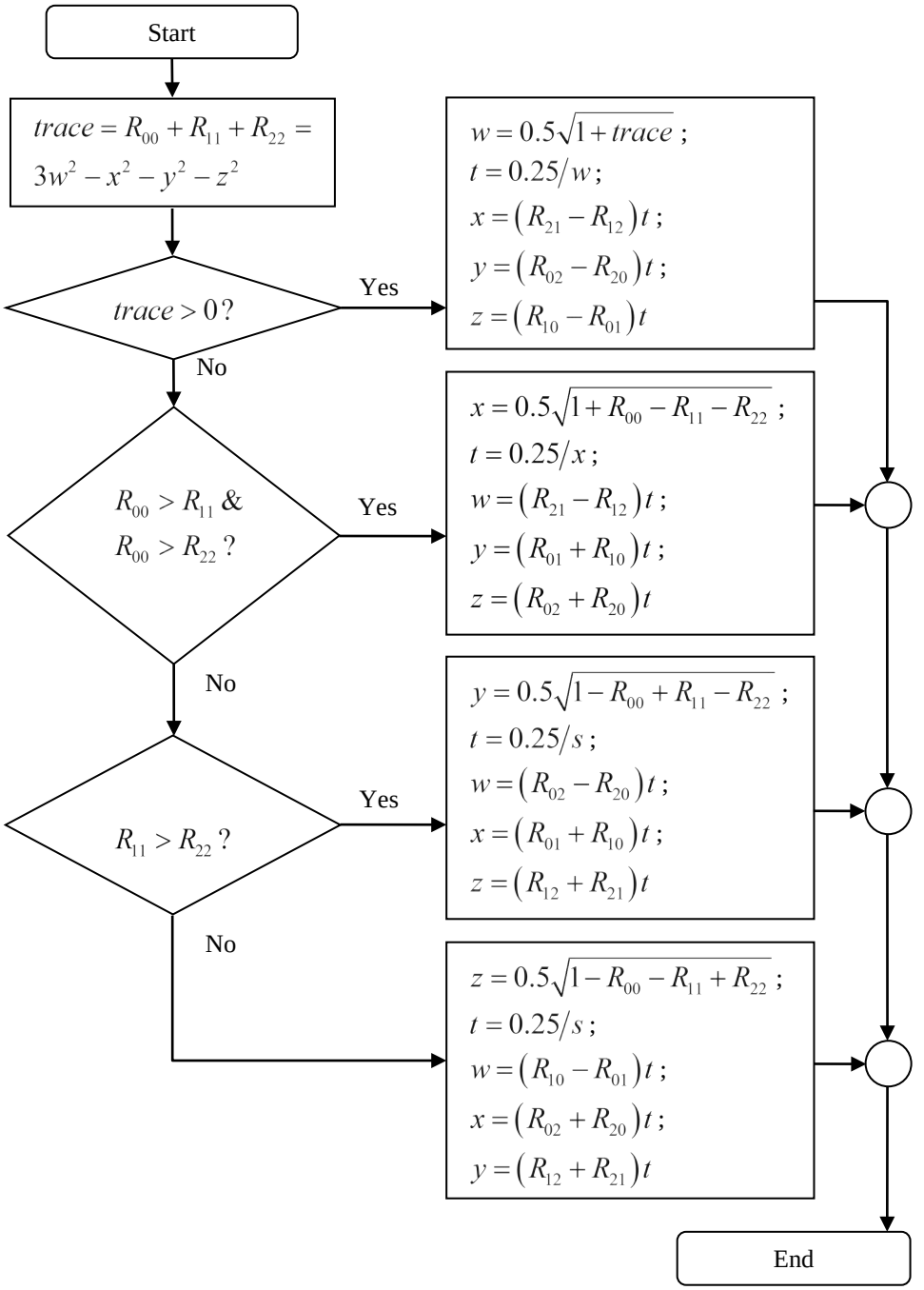

Fig. 2. Rotation matrix to rotation quaternion transformation algorithm

This rotation matrix may be used to rotate a row-matrix presented vector $\mathbb{k}$ using the operator $\mathbb{kR}$. The transformation from rotation matrix to rotation quaternion is more complex, due to a concern about the numerical stability. Division by small numbers needs to be avoided [21]. The idea is illustrated with the algorithm presented on figure 2. The algorithm first checks if the trace of the matrix is positive. In this case $|w| > 0.5$. If this condition is not met, the largest diagonal element is chosen, because it corresponds to the largest of the other three quaternion components absolute values $|x|$, $|y|$, or $|z|$. One of the latter three must be larger than $|w| > 0.5$.

3D geometric parallel between quaternions, vectors and matrices

There are many 3D geometric operations that are required to be implemented in an autopilot. Most of these operations are well known from vectors and matrices, but not so evident when utilizing quaternions. A discussion of such operations follows.

A vector may be presented in matrix form as a one-column or one-row matrix. This paper uses the one-row notation. As shown in (26) a quaternion may represent a vector:

$$\vec{k} \Leftrightarrow \mathbb{k} = \begin{bmatrix} x & y & z \end{bmatrix} \Leftrightarrow \mathbf{k} = \begin{bmatrix} 0, \vec{k} \end{bmatrix} \tag{34}$$

Using (7) the dot and cross products of two vectors represented as quaternions may be easily calculated:

$$\mathbf{k}\mathbf{k}' = (0 + x\mathbf{i} + y\mathbf{j} + z\mathbf{k})(0 + x'\mathbf{i} + y'\mathbf{j} + z'\mathbf{k}) = [-\vec{k}\vec{k}', \vec{k} \times \vec{k}'] \tag{35}$$

From (35) it follows that the dot product of two vectors represented as quaternions is:

$$\vec{k}\vec{k}' \Leftrightarrow -\mathrm{Re}(\mathbf{k}\mathbf{k}') \tag{36}$$

In (36) function $\mathrm{Re}(\ )$ returns the real part of a quaternion w. The cross product of two vectors is computed in a similar way:

$$\vec{k} \times \vec{k}' \Leftrightarrow \mathrm{Im}(\mathbf{k}\mathbf{k}') \tag{37}$$

Again, in (37) function $\mathrm{Im}(\ )$ returns the pure imaginary part $x\mathbf{i}+y\mathbf{j}+z\mathbf{k}$.

A general kinematic model based on quaternions used in autopilots

In this section the equations describing particle kinematics in a non-inertial reference frame are examined using quaternion formalism instead of vector formalism or matrix formalism. Let $Oxyz$ be the inertial and not moving reference frame and $O'x'y'z'$ be a non-inertial, moving and rotating reference frame (Fig. 3). All variables in regard to the inertial reference frame are non-primed and all variables in regard to the non-inertial reference frame are primed.

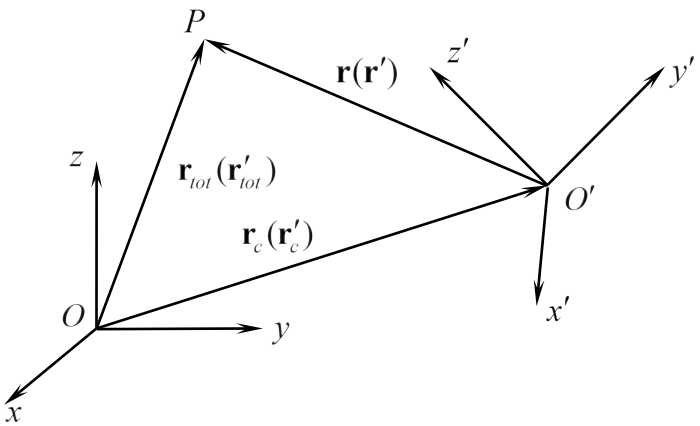

Fig. 3. Kinematics of a point particle P in regard to an inertial reference frame $Oxyz$ and non-inertial reference frame $O'x'y'z'$ described by quaternions

The non-inertial reference frame $O'x'y'z'$ has position $\mathbf{r}_C$ and orientation $\mathbf{q}$ in regard to the inertial reference frame $Oxyz$. The latter two quantities are quaternions. Quaternion $\mathbf{r}_C$ represents a vector and has zero real part. Quaternion $\mathbf{q}$ is a rotational quaternion. The position of the particle P in reference frame $Oxyz$ is $\mathbf{r}_{tot}$ and in reference frame $O'x'y'z'$ is $\mathbf{r}'$. Quaternion $\mathbf{r}'$ in regard to reference frame $Oxyz$ is $\mathbf{r}=\mathbf{q}\mathbf{r}'\tilde{\mathbf{q}}$:

(38) $$\mathbf{r}_{tot}=\mathbf{r}_C+\mathbf{r}=\mathbf{r}_C+\mathbf{q}\mathbf{r}'\tilde{\mathbf{q}}$$

To proceed further with the derivatives of (38) the reader must consider the following equations that follow from (31):

(39) $$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\omega}\mathbf{q} = \frac{1}{2}\mathbf{q}\tilde{\mathbf{q}}\boldsymbol{\omega}\mathbf{q} = \frac{1}{2}\mathbf{q}\boldsymbol{\omega}'$$

(40) $$\boldsymbol{\omega} = 2\dot{\mathbf{q}}\tilde{\mathbf{q}}$$

(41) $$\boldsymbol{\omega}' = 2\tilde{\mathbf{q}}\dot{\mathbf{q}}$$

The derivative of the rotation quaternion conjugate will also be utilized. Hence it is deduced in the following equation:

(42) $$\mathbf{q}\tilde{\mathbf{q}} = 1 \Rightarrow \frac{d}{dt}\left(\mathbf{q}\tilde{\mathbf{q}}\right) = 0 \Rightarrow \dot{\mathbf{q}}\tilde{\mathbf{q}} + \mathbf{q}\dot{\tilde{\mathbf{q}}} = 0 \Rightarrow \dot{\mathbf{q}}\tilde{\mathbf{q}} = -\mathbf{q}\dot{\tilde{\mathbf{q}}} \Rightarrow \dot{\tilde{\mathbf{q}}} = -\tilde{\mathbf{q}}\dot{\mathbf{q}}\tilde{\mathbf{q}}$$

By elaborating further on (42) one obtains:

(43) $$\dot{\tilde{\mathbf{q}}} = -\tilde{\mathbf{q}}\dot{\mathbf{q}}\tilde{\mathbf{q}} = -\frac{1}{2}\tilde{\mathbf{q}}\boldsymbol{\omega}\mathbf{q}\tilde{\mathbf{q}} = -\frac{1}{2}\tilde{\mathbf{q}}\boldsymbol{\omega}$$

And also:

(44) $$\dot{\tilde{\mathbf{q}}} = -\tilde{\mathbf{q}}\dot{\mathbf{q}}\tilde{\mathbf{q}} = -\frac{1}{2}\tilde{\mathbf{q}}\boldsymbol{\omega}\mathbf{q}\tilde{\mathbf{q}} = -\frac{1}{2}\boldsymbol{\omega}'\tilde{\mathbf{q}}$$

Deriving $\boldsymbol{\omega}$ from (43) results in:

(45) $$\boldsymbol{\omega} = -2\mathbf{q}\dot{\tilde{\mathbf{q}}}$$

When applying the rotation operator with quaternions we should mark off that its derivative has the form.

(46) $$\frac{d}{dt}\left(\mathbf{q}\mathbf{a}'\tilde{\mathbf{q}}\right) = \dot{\mathbf{q}}\mathbf{a}'\tilde{\mathbf{q}} + \mathbf{q}\dot{\mathbf{a}}'\tilde{\mathbf{q}} + \mathbf{q}\mathbf{a}'\dot{\tilde{\mathbf{q}}} =$$
$$\frac{1}{2}\mathbf{q}\boldsymbol{\omega}'\mathbf{a}'\tilde{\mathbf{q}} + \mathbf{q}\dot{\mathbf{a}}'\tilde{\mathbf{q}} - \frac{1}{2}\mathbf{q}\mathbf{a}'\boldsymbol{\omega}'\tilde{\mathbf{q}} = \mathbf{q}\boldsymbol{\omega}'\mathbf{a}'\tilde{\mathbf{q}} + \mathbf{q}\dot{\mathbf{a}}'\tilde{\mathbf{q}}$$

The above equation relies on the property of the quaternion product to be anti-commutative when the two quaternions have zero real part (represent vectors).

Now (38) may be differentiated to obtain the equation of speed:

$$(47) \qquad \mathbf{v}_{tot} = \dot{\mathbf{r}}_{tot} = \dot{\mathbf{r}}_C + \dot{\mathbf{r}} = \dot{\mathbf{r}}_C + \mathbf{q}\boldsymbol{\omega}'\mathbf{r}'\tilde{\mathbf{q}} + \mathbf{q}\dot{\mathbf{r}}'\tilde{\mathbf{q}} = \mathbf{v}_C + \mathbf{q}\mathbf{v}'\tilde{\mathbf{q}} + \mathbf{q}\boldsymbol{\omega}'\mathbf{r}'\tilde{\mathbf{q}}$$

And differentiating again the equation of acceleration is deduced:

$$(48) \qquad \mathbf{a}_{tot} = \ddot{\mathbf{r}}_{tot} = \ddot{\mathbf{r}}_C + \ddot{\mathbf{r}} = \mathbf{a}_C + \mathbf{q}\mathbf{a}'\tilde{\mathbf{q}} + 2\mathbf{q}\boldsymbol{\omega}'\mathbf{v}'\tilde{\mathbf{q}} + \mathbf{q}\boldsymbol{\omega}'\,\mathrm{Im}\left(\boldsymbol{\omega}'\mathbf{r}'\right)\tilde{\mathbf{q}} + \mathbf{q}\boldsymbol{\varepsilon}'\mathbf{r}'\tilde{\mathbf{q}}$$

Equation (48) may be written entirely in the inertial reference frame:

$$(49) \qquad \mathbf{a}_{tot} = \mathbf{a}_C + \mathbf{a} + 2\boldsymbol{\omega}\mathbf{v} + \boldsymbol{\omega}\,\mathrm{Im}\left(\boldsymbol{\omega}\mathbf{r}\right)\mathbf{r} + \boldsymbol{\varepsilon}\mathbf{r}$$

or in the non-inertial reference frame:

$$(50) \qquad \mathbf{a}' = \mathbf{a}'_{tot} - \mathbf{a}'_C - 2\boldsymbol{\omega}'\mathbf{v}' - \boldsymbol{\omega}'\,\mathrm{Im}\left(\boldsymbol{\omega}'\mathbf{r}'\right) - \boldsymbol{\varepsilon}'\mathbf{r}'$$

The terms in (50) are the different accelerations that appear when a particle is moving in a non-inertial reference frame:

$$(51) \qquad \mathbf{a}'_{net} = \mathbf{a}'_{tot} - \mathbf{a}'_C$$ - net inertial acceleration in quaternion form

$$(52) \qquad \mathbf{a}'_{cor} = -2\boldsymbol{\omega}'\mathbf{v}'$$ - Coriolis acceleration in quaternion form

$$(53) \qquad \mathbf{a}'_{cen} = -\boldsymbol{\omega}'\,\mathrm{Im}\left(\boldsymbol{\omega}'\mathbf{r}'\right)$$ - centrifugal acceleration in quaternion form

$$(54) \qquad \mathbf{a}'_{ang} = -\boldsymbol{\varepsilon}'\mathbf{r}'$$ - Euler acceleration in quaternion form

In the above four equations only the imaginary part of the result is of significance, as it holds a vector. The real part should be ignored.

A general dynamic model based on quaternions used in autopilots

The most common dynamic model with variances used in autopilots is the rigid body motion model of the aircraft. Further in this section the rigid body motion dynamic model shall be discussed in terms of quaternion implementation. On Fig. 4 a numerical 3D simulation is used to visualize the examined

phenomenon. The body reference frame $O'x'y'z'$ (non-inertial) is drawn with cyan colored vector arrows. The space reference frame $Oxyz$ (inertial) is visualized using white arrows. Angular momentum vector $\vec{L}$ is shown in magenta color. The external force arm coincides with $\overrightarrow{O'z'}$ vector. The external force vector $\vec{F}$ is in red color and the moment of external force vector $\vec{M}$ is drawn in green color. The orange vector is the angular velocity vector $\vec{\omega}$.

The equations of rigid body motion also known as Euler equations in matrix form are as follows:

(55) $\mathbb{L}' = \boldsymbol{\omega}'\mathbb{I}' \Rightarrow \mathbb{L}'\mathbb{R} = \boldsymbol{\omega}'\mathbb{R}\mathbb{R}^{\sim}\mathbb{I}'\mathbb{R} \Rightarrow \mathbb{L} = \boldsymbol{\omega}\mathbb{I}$

(56) $$\mathbb{M}_{ext} = \frac{d}{dt}(\mathbb{L}) = \frac{d}{dt}(\mathbb{L}'\mathbb{R}) =$$
$$\frac{d}{dt}(\mathbb{L}')\mathbb{R} + \mathbb{L}'\frac{d}{dt}(\mathbb{R}) = \frac{d}{dt}(\boldsymbol{\omega}')\mathbb{I}'\mathbb{R} + \boldsymbol{\omega}'\mathbb{I}'\frac{d}{dt}(\mathbb{R})$$

The (55) inference uses the well-known similarity transformation that rotates an arbitrary 3×3 matrix (in this case $\mathbb{I}'$) using a rotation matrix $\mathbb{R}$ and thus transforms the 3×3 matrix from body reference frame $O'x'y'z'$ to space reference frame $Oxyz$:

(57) $\mathbb{R}^{\sim}\mathbb{I}'\mathbb{R} = \mathbb{I}$

The matrix $\mathbb{R}^{\sim}$ is the transposed of $\mathbb{R}$. In (55) and (56) $\mathbb{L}$ is the angular momentum vector in matrix form, $\boldsymbol{\omega}$ is the angular velocity vector in matrix form and $\mathbb{M}_{ext}$ is the moment of external force vector in matrix form (one-row matrices). The 3×3 matrix $\mathbb{I}$ is the moment of inertia tensor, which is invariant in the body reference frame for a given non-changing rigid body. $\mathbb{I}'$ is defined as:

(58) $$\mathbb{I}' = \begin{bmatrix} I'_{xx} & I'_{xy} & I'_{xz} \\ I'_{xy} & I'_{yy} & I'_{yz} \\ I'_{xz} & I'_{yz} & I'_{zz} \end{bmatrix} =$$

$$\iiint_{V'} \begin{bmatrix} r_z'^2 + r_y'^2 & -r_x' r_y' & -r_x' r_z' \\ -r_x' r_y' & r_x'^2 + r_z'^2 & -r_y' r_z' \\ -r_x' r_z' & -r_y' r_z' & r_x'^2 + r_y'^2 \end{bmatrix} \rho(x', y', z')\, dx' dy' dz'$$

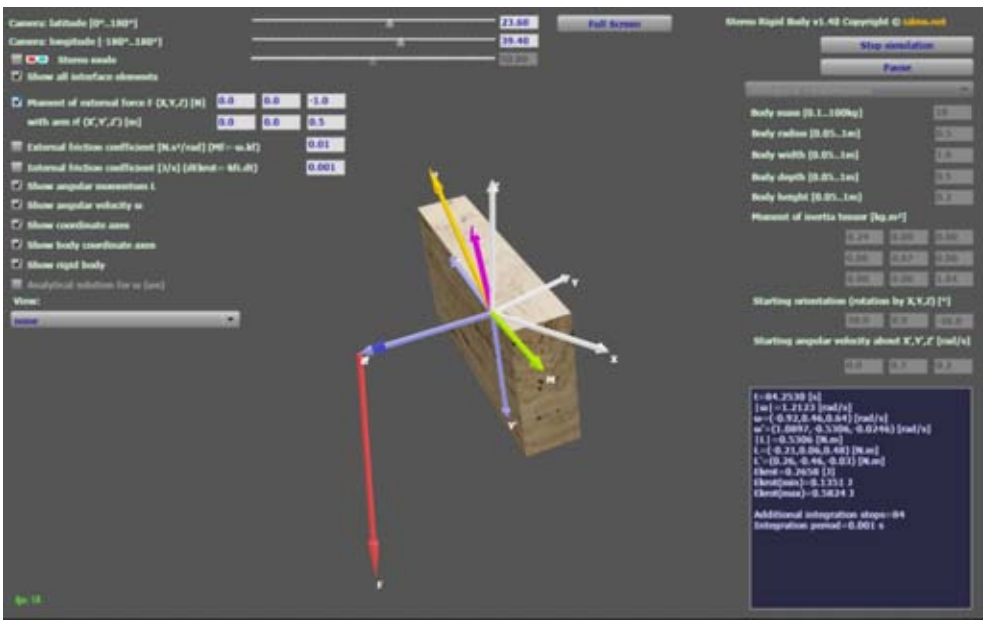

Fig. 4. Rigid body motion in a numerical 3D simulation

In (58) we observe that tensor $\mathbb{I}'$ is a symmetric matrix. Vector $\begin{bmatrix} r_x' & r_y' & r_z' \end{bmatrix}$ is the radius-vector of the current infinitesimal point in the rigid body that is being integrated. Scalar function $\rho(x', y', z')$ gives the density of the rigid body at the current point of integration. The integration is performed over the volume V' of the rigid body. For an elaborate derivation of (58) please consult [22]. Further, $\mathbb{I}'$ is a diagonal matrix if the body reference frame is chosen along the principal axis of inertia:

(59) $$\mathbb{I}' = \begin{bmatrix} I_{xx}' & 0 & 0 \\ 0 & I_{yy}' & 0 \\ 0 & 0 & I_{zz}' \end{bmatrix} = const$$

Its inverse is also a diagonal matrix and has the simple form of

$$(60)\qquad \mathbb{I}'^{-1} = \begin{bmatrix} I'^{-1}_{xx} & 0 & 0 \\ 0 & I'^{-1}_{yy} & 0 \\ 0 & 0 & I'^{-1}_{zz} \end{bmatrix} = \begin{bmatrix} 1/I'_{xx} & 0 & 0 \\ 0 & 1/I'_{yy} & 0 \\ 0 & 0 & 1/I'_{zz} \end{bmatrix} = const$$

Converting the rigid body motion equations into quaternion form requires the introduction of the Hadamard product of two quaternions:

$$(61)\qquad \mathbf{a} \circ \mathbf{b} = \left(w_a w_b, x_a x_b, y_a y_b, z_a z_b \right)$$

The diagonal tensor $\mathbb{I}'$ will be presented as a quaternion $\mathbf{I}' = \left(0, I'_{xx}, I'_{yy}, I'_{zz}\right)$. Analogously, $\mathbb{I}'^{-1}$ will be presented as $\mathbf{I}'^{-1} = \left(0, I'^{-1}_{xx}, I'^{-1}_{yy}, I'^{-1}_{zz}\right)$. Now by transforming (55) and (56) to quaternion form we get:

$$(62)\qquad \mathbb{L}'=\boldsymbol{\omega}'\mathbb{I}' \Rightarrow \mathbf{L}' = \boldsymbol{\omega}' \circ \mathbf{I}' \Rightarrow \boldsymbol{\omega}' = \mathbf{L}' \circ \mathbf{I}'^{-1}$$

$$(63)\qquad \mathbb{M}_{ext} = \frac{d}{dt}(\mathbb{L}) \Rightarrow$$
$$\mathbf{M}_{ext} = \dot{\mathbf{L}} = \mathbf{q}\boldsymbol{\omega}'\mathbf{L}'\tilde{\mathbf{q}} + \mathbf{q}\dot{\mathbf{L}}'\tilde{\mathbf{q}} = \mathbf{q}\boldsymbol{\omega}'\left(\boldsymbol{\omega}' \circ \mathbf{I}'\right)\tilde{\mathbf{q}} + \mathbf{q}\left(\dot{\boldsymbol{\omega}}' \circ \mathbf{I}'\right)\tilde{\mathbf{q}}$$

$$(64)\qquad \mathbf{M}'_{ext} = \tilde{\mathbf{q}}\mathbf{M}_{ext}\mathbf{q} = \boldsymbol{\omega}'\left(\boldsymbol{\omega}' \circ \mathbf{I}'\right) + \dot{\boldsymbol{\omega}}' \circ \mathbf{I}'$$

A special case of rigid body motion is the free rigid body motion when the moment of external force is zero:

$$(65)\qquad \mathbf{M}_{ext} = \dot{\mathbf{L}} = 0 \Rightarrow \mathbf{L} = \mathbf{q}\mathbf{L}'\tilde{\mathbf{q}} = \mathbf{q}\left(\boldsymbol{\omega}' \circ \mathbf{I}'\right)\tilde{\mathbf{q}} = const$$

The second constant of free rigid body motion is the kinetic energy of rotation E_{Krot}:

$$(66)\qquad E_{Krot} = -\frac{\mathrm{Re}\left(\mathbf{L}\boldsymbol{\omega}\right)}{2} = -\frac{\mathrm{Re}\left(\mathbf{L}'\boldsymbol{\omega}'\right)}{2} = -\frac{\mathrm{Re}\left(\left(\boldsymbol{\omega}' \circ \mathbf{I}'\right)\boldsymbol{\omega}'\right)}{2} = const$$

Quaternion numerical integration used in autopilots

The current orientation of the body reference frame (i.e. the aircraft airframe) in respect to the space reference frame is obtained by integrating the angular velocity vector. The latter is derived from the onboard gyroscope. The onboard gyroscope measures the angular velocity $\boldsymbol{\omega}'$ in regard to the body reference frame. Equation (39) comes in hand and it should be now numerically integrated:

$$\mathbf{q} = \mathbf{q}_0 + \int_0^{\Delta t} \dot{\mathbf{q}} dt \tag{67}$$

Different integration schemes may be implemented such as Euler, Runge-Kutta, etc. A symplectic method of integration is preferred as it preserves the rotational energy under conservative torque. Below an example is given with the symplectic semi-implicit Euler–Cromer integration method:

$$\boldsymbol{\omega}'_{n+1} = \boldsymbol{\omega}'_n + \left(\tilde{\mathbf{q}}_n \mathbf{M}_{(ext)n} \mathbf{q}_n - \boldsymbol{\omega}'_n \left(\boldsymbol{\omega}'_n \circ \mathbf{I}'\right)\right) \circ \mathbf{I}'^{-1} \Delta t \tag{68a}$$

$$\boldsymbol{\omega}'_{n+1} = gyroscope\_omega\left(t_{n+1}\right) \tag{68b}$$

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \frac{1}{2}\mathbf{q}_n \boldsymbol{\omega}'_{n+1} \Delta t = \mathbf{q}_n \left(\mathbf{1} + \frac{\boldsymbol{\omega}'_{n+1} \Delta t}{2}\right) \tag{68c}$$

Equation (68a) demonstrates the predicted angular velocity. The gyroscope read angular velocity is presented in (68b). Finally, the new orientation is calculated in (68c). The autopilot utilized algorithm selects between the predicted angular velocity according to the dynamic model and the gyroscope read angular velocity. Further corrections on the current position are carried out according to other autopilot sensors and procedures, which are not disclosed in (68).

In the above example the so called naive quaternion integration was implemented. A better approach is to integrate the quaternion over the hypersphere surface in the four dimensional quaternion space [23]. This four dimensional sphere has radius of 1 (see figure 5). The rotation quaternions $\mathbf{q}_n$ and $\mathbf{q}_{n+1}$ are unit quaternions and lie on the unit hypersphere. The natural method of integrating the rotation quaternion is to interpolate it over the hypersphere surface. The time change for one integration step of the rotation quaternion $\mathbf{q}_n$ is $\dot{\mathbf{q}}\Delta t$. The latter quaternion is perpendicular to $\mathbf{q}_n$, but having non-infinitesimal length goes out of the hypersphere.

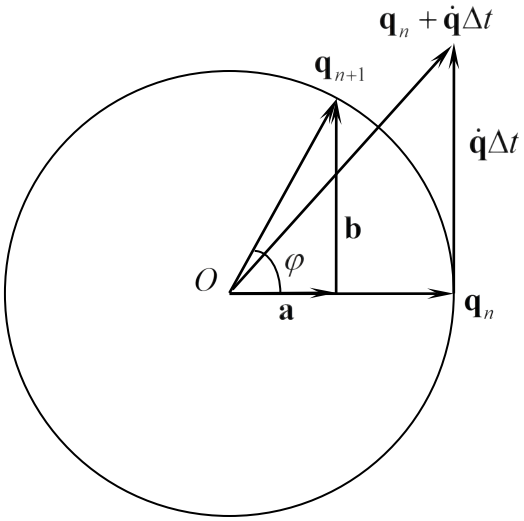

Fig. 5. Integrating the quaternion over the hypersphere surface

The new rotation quaternion $\mathbf{q}_{n+1} = \mathbf{q}_n + \dot{\mathbf{q}}\Delta t$ has magnitude > 1. On the other hand the correct value for $\mathbf{q}_{n+1}$ is a quaternion positioned on the hypersphere. It also lies on the two dimensional unit circle defined by $\mathbf{q}_n$ and $\dot{\mathbf{q}}\Delta t$ (Fig. 5). The arc of the hypersphere between $\mathbf{q}_n$ and $\mathbf{q}_{n+1}$ has length $\varphi = |\dot{\mathbf{q}}|\,\Delta t$. It follows that quaternion $\mathbf{b} = \dfrac{\dot{\mathbf{q}}}{|\dot{\mathbf{q}}|}\sin\varphi$ and quaternion $\mathbf{a} = \mathbf{q}_n \cos\varphi$. For $\mathbf{q}_{n+1}$ we obtain:

(69) $$\mathbf{q}_{n+1} = \mathbf{a} + \mathbf{b} = \mathbf{q}_n \cos\varphi + \frac{\dot{\mathbf{q}}}{|\dot{\mathbf{q}}|}\sin\varphi$$

Conclusion

The utilization of quaternions in autopilots has been a privilege to large and expensive aircraft till recently. The quaternion arithmetic computations are much faster and more accurate than other approaches, but require well written software libraries of significant complexity. The increased accuracy and flexibility of the quaternion method is a fruitful avenue for developing prototype, scientific and research autopilot systems. Furthermore, the modern unmanned helicopters do require custom made autopilots that can respond to their high specific needs. Such machine is the award winning *The Bulgarian Knight* Dodecacopter

that has an unique rotor arrangement and benefits from a custom and specialized accurate and fast autopilot, realized using the quaternion approach.

References

1. Pervin, E., J.A. Webb. Quaternions in Computer Vision and Robotics. Carnegie-Mellon University, 1992.
2. Foreman, D., C. Tournes, and Y. Shtessel. Integrated missile flight control using quaternions and Third-order sliding mode control. *IEEE* 11th International Workshop on Variable Structure Systems (VSS), 2010, 370–375.
3. Mahony, R., V. Kumar, and P. Corke. Multirotor aerial vehicles: Modeling, estimation and control of quadrotor. IEEE Robotics and Automation Magazine, 2012, 19(3), 20–32.
4. Dam, E.B., M. Koch, and M. Lillholm. Quaternions, interpolation, and animation. Technical Report DIKU-TR-98/5, Department of Computer Science, University of Copenhagen, Denmark, 1998.
5. Goldman, R. Rethinking Quaternions: Theory and Computation. Morgan & Claypool Publishers, 2010. ISBN 978-1-60845-420-4.
6. Faugeras, O.D. and M. Hebert. The representation, recognition, and locating of 3-D objects. International Journal of Robotics Research, 1986, 5(3), 27–52.
7. Shoemake, K. Animating rotation with quaternion curves. Computer Graphics, 1985, 19(3), 245–254.
8. Zhao, F. and B.G.M. van Wachem. A novel quaternion integration approach for describing the behaviour of non-spherical particles. Acta Mechanica, 2013, 224, 3091–3109.
9. Horn, B.K.P. Closed-form solution of absolute orientation using unit quaternions. Journal of Optical Society of America A, 1987, 4(4), 629–642.
10. Shoemake, K. Quaternion calculus and fast animation. SIGGRAPH Course Notes, 1987, 10, 101–121.
11. Hamilton, W.R. On quaternions, or on a new system of imaginaries in algebra. Philosophical Magazine. 1844, 25(3), 489–495.
12. Gutmann-Madsen, T. Course notes for Mathematics 1MA (calculus). Matematisk Notetryk, Institute of Mathematics, University of Copenhagen, Denmark, Copenhagen, 1991.
13. Hamilton, W.R. Lectures on Quaternions. Hodges Smith & Co., Dublin, 1853.
14. Hamilton, W.R. Elements of Quaternions, Vol. 1–2. Longmans, Green and Co., 1899.
15. Kuipers, J.B. Quaternions and rotation Sequences: a Primer with Applications to Orbits, Aerospace, and Virtual Reality, Princeton University Press, 1999. ISBN 978-0-691-10298-6
16. Chris, D., A.N. Lasenby. Geometric Algebra for Physicists. Cambridge University Press, 2003. ISBN 978-0-521-48022-2.
17. Maillot, P.-G. Using quaternions for coding 3D transformations. In: Andrew Glassner, editor, Graphics Gems 1, chapter 10, 498–515. Academic Press, Inc., 1990.
18. Omelyan, I.P. Algorithm for numerical integration of the rigid-body equations of motion, Phys. Rev. E., 1998, 58(1), 1169.

19. Kim, M.-J., M.-S., Kim, and S.Y. Shin. A compact differential formula for the first derivative of a unit quaternion curve. Journal of Visualization and Computer Animation, 1996, 7(1), 43–57.
20. Xu, D., C. Jahanchahi, C.C. Took, and D.P. Mandic. Quaternion Derivatives: The GHR Calculus. Royal Society Open Science, 2015, 2(8), 150255.
21. van Waveren, J.M.P. From Quaternion to Matrix and Back, 27 February 2005, Id Software, Inc.
22. Zabunov, S., P. Getsov, and M. Gaydarova. Stabilization of Free Rigid Body Motion Stereo 3D Simulation through Invariants, International Journal of Advanced Research in Computer Science, 2014, 5(6), 9–16.
23. Jia, Y.-B. Quaternion and Rotation. Com S, 2013, 477(577), 15.

АВТОПИЛОТ ЗА ДВАНАДЕСЕТОКОПТЕРИ, БАЗИРАН ВЪРХУ КВАТЕРНИОНИ – ЧАСТ I

С. Забунов

Резюме

Иновациите в областта на модерните безпилотни летателни апарати поставят по-високи изисквания към автопилотите. Иновативните мулти-роторни хеликоптери, поради своята уникална система за контрол и разположение на роторите, изискват от автопилотите специални условия.

Настоящата статия представя ядрото на автопилот, който е базиран върху кватерниони. Този автопилот е подходящ за иновативния и спечелил международни награди дванадесет роторен безпилотен хеликоптер “Българският Рицар”. Кватернионите предлагат на автопилотните системи редица предимства. Тяхното приложение в специализираните автопилоти изисква изключително внимание и обсъждане. Като резултат се получава ефективен и гъвкав автопилот, защото кватернионните изчисления са много по-бързи и по-точни от другите конкурентни подходи, но такъв автопилот изисква сложни софтуерни библиотеки. Повишената точност и адаптивност на кватернионния метод го правят обещаващо средство за разработка на прототипни, научни и изследователски автопилотни системи, подходящи за специфичните нужди на иновативните дронове.