Quaternion kinematics for the error-state KF

Joan Solà

August 2, 2016

Contents

1 Quaternions and rotation operations

1.1 Definition of quaternion

One introduction to the quaternion that I find particularly attractive is given by the Cayley-Dickson construction: If we have two complex numbers $A = a + bi$ and $C = c + di$, then constructing $Q = A + Cj$ and defining $k \triangleq ij$ yields a number in the space of quaternions $\mathbb{H}$,

$$Q = a + bi + cj + dk \in \mathbb{H} \ , \tag{1}$$

where $\{a, b, c, d\} \in \mathbb{R}$, and $\{i, j, k\}$ are three imaginary unit numbers defined so that

$$i^2 = j^2 = k^2 = ijk = -1 \ , \tag{2a}$$

from which we can derive

$$ij = -ji = k \ , \quad jk = -kj = i \ , \quad ki = -ik = j \ . \tag{2b}$$

From (1) we see that we can embed complex numbers, and thus real and imaginary numbers, in the quaternion definition, in the sense that real, imaginary and complex numbers are indeed quaternions,

$$Q = a \in \mathbb{R} \subset \mathbb{H} \ , \qquad Q = bi \in \mathbb{I} \subset \mathbb{H} \ , \qquad Q = a + bi \in \mathbb{Z} \subset \mathbb{H} \ . \tag{3}$$

Likewise, and for the sake of completeness, we may define numbers in the tri-dimensional imaginary subspace of $\mathbb{H}$ (we refer to them as *pure quaternions*),

$$Q = bi + cj + dk \in \mathbb{I}^3 \subset \mathbb{H} \ . \tag{4}$$

It is noticeable that, while regular complex numbers of unit length $\mathbf{z} = e^{i\theta}$ can encode rotations in the 2D plane (with one complex product, $\mathbf{x}' = \mathbf{z} \cdot \mathbf{x}$), "extended complex numbers" or quaternions of unit length $\mathbf{q} = e^{(u_x i + u_y j + u_z k)\theta/2}$ encode rotations in the 3D space (with a double quaternion product, $\mathbf{x}' = \mathbf{q} \otimes \mathbf{x} \otimes \mathbf{q}^*$, as we explain later in this document).

CAUTION: Not all quaternion definitions are the same. Some authors write the products as ib instead of bi, and therefore they get the property $k = ji = -ij$, which results in $ijk = 1$ and a left-handed quaternion. Also, many authors place the real part at the end position, yielding $Q = ia + jb + kc + d$. These choices have no fundamental implications but make the whole formulation different in the details. Please refer to Section 1.5 for further explanations and disambiguation.

CAUTION: There are additional conventions that also make the formulation different in details. They concern the "meaning" or "interpretation" we give to the rotation operators, either rotating vectors or rotating reference frames –which, essentially, constitute opposite operations. Refer also to Section 1.5 for further explanations and disambiguation.

1.2 Alternative representations of the quaternion

The real + imaginary notation $\{1, i, j, k\}$ is not always convenient for our purposes. Provided that the algebra (2) is used, a quaternion can be posed as a sum scalar + vector,

$$Q = q_w + q_x i + q_y j + q_z k \qquad \Leftrightarrow \qquad Q = q_w + \mathbf{q}_v \ , \tag{5}$$

where q_w is referred to as the *real* or *scalar* part, and $\mathbf{q}_v = q_x i + q_y j + q_z k = (q_x, q_y, q_z)$ as the *imaginary* or *vector* part.[1] It can be also defined as an ordered pair scalar-vector

$$Q = \langle q_w, \mathbf{q}_v \rangle \ . \tag{6}$$

We mostly represent a quaternion Q as a 4-vector $\mathbf{q}$,

$$\mathbf{q} \triangleq \begin{bmatrix} q_w \\ \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} \ , \tag{7}$$

[1] Our choice for the (w, x, y, z) subscripts notation comes from the fact that we are interested in the geometric properties of the quaternion in the 3D Cartesian space. Other texts often use alternative subscripts such as $(0, 1, 2, 3)$ (more suited for algebraic interpretations) or $(1, i, j, k)$ (for mathematical interpretations).

which allows us to use matrix algebra for operations involving quaternions. At certain occasions, we may allow ourselves to mix notations by abusing of the sign "=". Typical examples are *real quaternions* and *pure quaternions*,

$$\mathbf{q} = q_w + \mathbf{q}_v = \begin{bmatrix} q_w \\ \mathbf{q}_v \end{bmatrix} \quad , \quad \text{real: } q_w = \begin{bmatrix} q_w \\ \mathbf{0}_v \end{bmatrix} \quad , \quad \text{pure: } \mathbf{q}_v = \begin{bmatrix} 0 \\ \mathbf{q}_v \end{bmatrix} \; . \tag{8}$$

1.3 Some quaternion properties

1.3.1 Sum

The sum is straightforward,

$$\mathbf{p} \pm \mathbf{q} = \begin{bmatrix} p_w \\ \mathbf{p}_v \end{bmatrix} \pm \begin{bmatrix} q_w \\ \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} p_w \pm q_w \\ \mathbf{p}_v \pm \mathbf{q}_v \end{bmatrix} \; . \tag{9}$$

By construction, the sum is **commutative** and **associative**,

$$\mathbf{p} + \mathbf{q} = \mathbf{q} + \mathbf{p} \tag{10}$$

$$\mathbf{p} + (\mathbf{q} + \mathbf{r}) = (\mathbf{p} + \mathbf{q}) + \mathbf{r} \; . \tag{11}$$

Thus, the set of quaternions endowed with the sum operation form a commutative group, where the identity is the zero quaternion, $\mathbf{q}_0 = 0$, and the inverse is the negative $-\mathbf{q}$.

1.3.2 Product

Denoted by $\otimes$, the quaternion product requires using the original form (1) and the quaternion algebra (2). Writing the result in vector form gives

$$\mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} p_w q_w - p_x q_x - p_y q_y - p_z q_z \\ p_w q_x + p_x q_w + p_y q_z - p_z q_y \\ p_w q_y - p_x q_z + p_y q_w + p_z q_x \\ p_w q_z + p_x q_y - p_y q_x + p_z q_w \end{bmatrix} \; . \tag{12}$$

This can be posed also in terms of the scalar and vector parts,

$$\mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} p_w q_w - \mathbf{p}_v^\top \mathbf{q}_v \\ p_w \mathbf{q}_v + q_w \mathbf{p}_v + \mathbf{p}_v \times \mathbf{q}_v \end{bmatrix} \; , \tag{13}$$

where the presence of the cross-product reveals that the quaternion product is **not commutative** in the general case,

$$\mathbf{p} \otimes \mathbf{q} \neq \mathbf{q} \otimes \mathbf{p} \; . \tag{14}$$

Exceptions to this general non-commutativity are limited to the cases where $\mathbf{p}_v \times \mathbf{q}_v = 0$, which happens whenever one quaternion is real, $\mathbf{p} = p_w$ or $\mathbf{q} = q_w$, or when both vector parts are parallel, $\mathbf{p}_v \| \mathbf{q}_v$. Only in these cases the quaternion product is commutative.

The quaternion product is however **associative**,

$$(\mathbf{p} \otimes \mathbf{q}) \otimes \mathbf{r} = \mathbf{p} \otimes (\mathbf{q} \otimes \mathbf{r}) \ , \tag{15}$$

and **distributive over the sum**,

$$\mathbf{p} \otimes (\mathbf{q} + \mathbf{r}) = \mathbf{p} \otimes \mathbf{q} + \mathbf{p} \otimes \mathbf{r} \qquad \text{and} \qquad (\mathbf{p} + \mathbf{q}) \otimes \mathbf{r} = \mathbf{p} \otimes \mathbf{r} + \mathbf{q} \otimes \mathbf{r} \ . \tag{16}$$

Thus, quaternions endowed with the product operation $\otimes$ form a (non-commutative) group. The group's elements identity, $\mathbf{q}_1 = 1$, and inverse, $\mathbf{q}^{-1}$, are explored below.

The product of two quaternions is bi-linear and can be expressed as two equivalent matrix products, namely

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = [\mathbf{q}_1]_L \, \mathbf{q}_2 \qquad \text{and} \qquad \mathbf{q}_1 \otimes \mathbf{q}_2 = [\mathbf{q}_2]_R \, \mathbf{q}_1 \ , \tag{17}$$

where $[\mathbf{q}_1]_L$ and $[\mathbf{q}_2]_R$ are respectively the left- and right- quaternion-product matrices, which are derived from (12) and (17) by simple inspection,

$$[\mathbf{q}]_L = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & -q_z & q_y \\ q_y & q_z & q_w & -q_x \\ q_z & -q_y & q_x & q_w \end{bmatrix} , \qquad [\mathbf{q}]_R = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & q_z & -q_y \\ q_y & -q_z & q_w & q_x \\ q_z & q_y & -q_x & q_w \end{bmatrix} , \tag{18}$$

or more concisely, from (13) and (17)

$$[\mathbf{q}]_L = q_w \, \mathbf{I} + \begin{bmatrix} 0 & -\mathbf{q}_v^\top \\ \mathbf{q}_v & [\mathbf{q}_v]_\times \end{bmatrix} , \qquad [\mathbf{q}]_R = q_w \, \mathbf{I} + \begin{bmatrix} 0 & -\mathbf{q}_v^\top \\ \mathbf{q}_v & -[\mathbf{q}_v]_\times \end{bmatrix} \tag{19}$$

Here, the *skew operator*[2] $[\bullet]_\times$ produces the cross-product matrix,

$$[\mathbf{a}]_\times \triangleq \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} , \tag{20}$$

which is a skew-symmetric matrix, $[\mathbf{a}]_\times^\top = -[\mathbf{a}]_\times$, left-hand-equivalent to the cross product, *i.e.*,

$$[\mathbf{a}]_\times \, \mathbf{b} = \mathbf{a} \times \mathbf{b} \ , \quad \forall \, \mathbf{a}, \mathbf{b} \in \mathbb{R}^3 \ . \tag{21}$$

Finally, since

$$(\mathbf{q} \otimes \mathbf{r}) \otimes \mathbf{p} = [\mathbf{p}]_R \, [\mathbf{q}]_L \, \mathbf{r} \qquad \text{and} \qquad \mathbf{q} \otimes (\mathbf{r} \otimes \mathbf{p}) = [\mathbf{q}]_L \, [\mathbf{p}]_R \, \mathbf{r} \tag{22}$$

we have the relation

$$[\mathbf{p}]_R \, [\mathbf{q}]_L = [\mathbf{q}]_L \, [\mathbf{p}]_R \ . \tag{23}$$

Further properties of these matrices are provided in Section 1.4.7.

[2] The skew-operator can be found in the literature in a number of different names and notations, either related to the cross operator $\times$, or to the 'hat' operator $^\wedge$, so that all these forms are equivalent,

$$[\mathbf{a}]_\times \equiv [\mathbf{a}_\times] \equiv \mathbf{a}\times \equiv \mathbf{a}_\times \equiv [\mathbf{a}] \equiv \hat{\mathbf{a}} \equiv \mathbf{a}^\wedge \ ,$$

1.3.3 Identity

The identity quaternion $\mathbf{q}_{\mathbf{1}}$ with respect to the product is such that $\mathbf{q}_{\mathbf{1}} \otimes \mathbf{q} = \mathbf{q} \otimes \mathbf{q}_{\mathbf{1}} = \mathbf{q}$. It corresponds to the real product identity '1' expressed as a quaternion,

$$\mathbf{q}_{\mathbf{1}} = 1 = \begin{bmatrix} 1 \\ \mathbf{0}_v \end{bmatrix} .$$

1.3.4 Conjugate

The conjugate of a quaternion is defined by

$$\mathbf{q}^* \triangleq q_w - \mathbf{q}_v = \begin{bmatrix} q_w \\ -\mathbf{q}_v \end{bmatrix} . \tag{24}$$

This has the properties

$$\mathbf{q} \otimes \mathbf{q}^* = \mathbf{q}^* \otimes \mathbf{q} = q_w^2 + q_x^2 + q_y^2 + q_z^2 = \begin{bmatrix} q_w^2 + q_x^2 + q_y^2 + q_z^2 \\ \mathbf{0}_v \end{bmatrix} , \tag{25}$$

and

$$(\mathbf{p} \otimes \mathbf{q})^* = \mathbf{q}^* \otimes \mathbf{p}^* . \tag{26}$$

1.3.5 Norm

The norm of a quaternion is defined by

$$\|\mathbf{q}\| \triangleq \sqrt{\mathbf{q} \otimes \mathbf{q}^*} = \sqrt{\mathbf{q}^* \otimes \mathbf{q}} = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2} . \tag{27}$$

It has the property

$$\|\mathbf{p} \otimes \mathbf{q}\| = \|\mathbf{p}\|\|\mathbf{q}\| . \tag{28}$$

1.3.6 Inverse

The inverse quaternion $\mathbf{q}^{-1}$ is such that

$$\mathbf{q} \otimes \mathbf{q}^{-1} = \mathbf{q}^{-1} \otimes \mathbf{q} = \mathbf{q}_{\mathbf{1}} . \tag{29}$$

It can be computed with

$$\mathbf{q}^{-1} = \mathbf{q}^* / \|\mathbf{q}\|^2 . \tag{30}$$

1.3.7 Unit or normalized quaternion

For unit quaternions, $\|\mathbf{q}\| = 1$, and therefore

$$\mathbf{q}^{-1} = \mathbf{q}^* \ . \tag{31}$$

From (28), unit quaternions endowed with the product operation $\otimes$ form a (non-commutative) group.

When interpreting the unit quaternion as an orientation specification, or as a rotation operator, this property implies that the inverse rotation can be accomplished with the conjugate quaternion. Unit quaternions can always be written in the form,

$$\mathbf{q} = \begin{bmatrix} \cos\theta \\ \mathbf{u}\sin\theta \end{bmatrix} \ , \tag{32}$$

where $\mathbf{u} = u_x i + u_y j + u_z k$ is a unit vector and θ is a scalar.

1.3.8 Product of pure quaternions

Pure quaternions are those with null real or scalar part, $Q = \mathbf{q}_v$ or $\mathbf{q} = [0, \mathbf{q}_v]$. From (13) we have

$$\mathbf{p}_v \otimes \mathbf{q}_v = -\mathbf{p}_v^\top \mathbf{q}_v + \mathbf{p}_v \times \mathbf{q}_v = \begin{bmatrix} -\mathbf{p}_v^\top \mathbf{q}_v \\ \mathbf{p}_v \times \mathbf{q}_v \end{bmatrix} \ . \tag{33}$$

This implies

$$\mathbf{p}_v \otimes \mathbf{q}_v - \mathbf{q}_v \otimes \mathbf{p}_v = 2\, \mathbf{p}_v \times \mathbf{q}_v \ . \tag{34}$$

1.3.9 Powers of pure quaternions

Let us define $\mathbf{q}^n$ as the n-th power of $\mathbf{q}$ using the quaternion product $\otimes$. Then, if $\mathbf{v}$ is a pure quaternion and we let $\mathbf{v} = \mathbf{u}\,\theta$, with $\theta = \|\mathbf{v}\| \in \mathbb{R}$ and $\mathbf{u}$ unitary, we get from (33) the cyclic pattern

$$\mathbf{v}^2 = -\theta^2 \quad , \quad \mathbf{v}^3 = -\mathbf{u}\,\theta^3 \quad , \quad \mathbf{v}^4 = \theta^4 \quad , \quad \mathbf{v}^5 = \mathbf{u}\,\theta^5 \quad , \quad \mathbf{v}^6 = -\theta^6 \quad , \quad \cdots \tag{35}$$

1.3.10 Exponential of pure quaternions

The exponential of a general quaternion is defined as the absolutely convergent series

$$e^{\mathbf{q}} \triangleq \sum_{k=0}^{\infty} \frac{\mathbf{q}^n}{k!} \quad \in \mathbb{H} \ . \tag{36}$$

Then, the exponential of a pure quaternion $\mathbf{v} = v_x i + v_y j + v_z k$ is a new quaternion defined by the absolutely convergent series,

$$e^{\mathbf{v}} = \sum_{k=0}^{\infty} \frac{\mathbf{v}^n}{k!} \quad \in \mathbb{H} \ . \tag{37}$$

Letting $\mathbf{v} = \mathbf{u}\,\theta$, with $\theta = \|\mathbf{v}\| \in \mathbb{R}$ and $\mathbf{u}$ unitary, and considering (35), we group the scalar and vector terms in the series, and recognize in them, respectively, the series of $\cos\theta$ and $\sin\theta$.[3] This results in

$$e^{\mathbf{v}} = e^{\mathbf{u}\,\theta} = \cos\theta + \mathbf{u}\sin\theta = \begin{bmatrix} \cos\theta \\ \mathbf{u}\sin\theta \end{bmatrix} , \tag{38}$$

which constitutes a beautiful extension of the Euler formula, $e^{i\theta} = \cos\theta + i\sin\theta$. Notice that since $\|e^{\mathbf{v}}\|^2 = \cos^2\theta + \sin^2\theta = 1$, the exponential of a pure quaternion is a unit quaternion.

1.3.11 Exponential of general quaternions

From the non-commutativity property of the quaternion product, we cannot write for general quaternions $\mathbf{p}$ and $\mathbf{q}$ that $e^{\mathbf{p}+\mathbf{q}} = e^{\mathbf{p}}e^{\mathbf{q}}$. However, commutativity holds when any of the product members is a scalar, therefore,

$$e^{\mathbf{q}} = e^{q_w + \mathbf{q}_v} = e^{q_w}\, e^{\mathbf{q}_v} . \tag{39}$$

Then, using (38) with $\mathbf{v} = \mathbf{u}\,\theta = \mathbf{q}_v$ we get

$$e^{\mathbf{q}} = e^{q_w} \begin{bmatrix} \cos\|\mathbf{q}_v\| \\ \frac{\mathbf{q}_v}{\|\mathbf{q}_v\|}\sin\|\mathbf{q}_v\| \end{bmatrix} . \tag{40}$$

1.3.12 Logarithm of unit quaternions

It is immediate to see that, if $\|\mathbf{q}\| = 1$,

$$\log\mathbf{q} = \log(\cos\theta + \mathbf{u}\sin\theta) = \log(e^{\mathbf{u}\,\theta}) = \mathbf{u}\,\theta = \begin{bmatrix} 0 \\ \mathbf{u}\,\theta \end{bmatrix} , \tag{41}$$

that is, the logarithm of a unit quaternion is a pure quaternion.

1.3.13 Logarithm of general quaternions

By extension, if $\mathbf{q}$ is a general quaternion,

$$\log\mathbf{q} = \log(\|\mathbf{q}\|\frac{\mathbf{q}}{\|\mathbf{q}\|}) = \log\|\mathbf{q}\| + \log\frac{\mathbf{q}}{\|\mathbf{q}\|} = \log\|\mathbf{q}\| + \mathbf{u}\,\theta = \begin{bmatrix} \log\|\mathbf{q}\| \\ \mathbf{u}\,\theta \end{bmatrix} . \tag{42}$$

1.4 Rotations and cross-relations

1.4.1 The 3D vector rotation formula

We illustrate in Fig. 1 the rotation, following the right-hand rule, of a general 3D vector $\mathbf{x}$, by an angle ϕ, around the axis defined by the unit vector $\mathbf{u}$. This is accomplished by

[3] We remind that $\cos\theta = 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \cdots$ and $\sin\theta = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \cdots$.

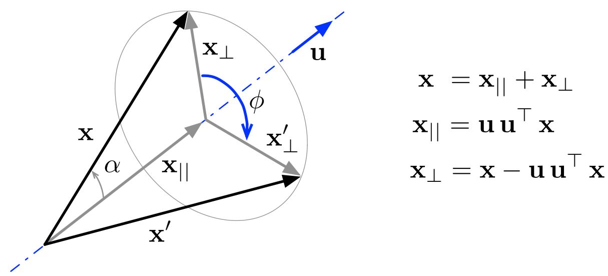

Figure 1: Rotation of a vector $\mathbf{x}$, by an angle ϕ, around the axis $\mathbf{u}$. See text for details.

decomposing the vector $\mathbf{x}$ into a part $\mathbf{x}_{||}$ parallel to $\mathbf{u}$, and a part $\mathbf{x}_\perp$ orthogonal to $\mathbf{u}$, so that

$$\mathbf{x} = \mathbf{x}_{||} + \mathbf{x}_\perp \ . \tag{43}$$

These parts can be computed easily,

$$\mathbf{x}_{||} = \mathbf{u}\,(\|\mathbf{x}\|\cos\alpha) = \mathbf{u}\,\mathbf{u}^\top\,\mathbf{x} \tag{44}$$

$$\mathbf{x}_\perp = \mathbf{x} - \mathbf{x}_{||} = \mathbf{x} - \mathbf{u}\,\mathbf{u}^\top\,\mathbf{x} \ . \tag{45}$$

Upon rotation, the parallel part does not rotate,

$$\mathbf{x}'_{||} = \mathbf{x}_{||} \ , \tag{46}$$

and the orthogonal part experiences a planar rotation in the plane normal to $\mathbf{u}$. That is, if we create an orthogonal base $\{\mathbf{e}_1, \mathbf{e}_2\}$ of this plane with

$$\mathbf{e}_1 = \mathbf{x}_\perp \tag{47}$$

$$\mathbf{e}_2 = \mathbf{u}\times\mathbf{x}_\perp = \mathbf{u}\times\mathbf{x} \ , \tag{48}$$

satisfying $\|\mathbf{e}_1\| = \|\mathbf{e}_2\|$, then $\mathbf{x}_\perp = \mathbf{e}_1\cdot 1 + \mathbf{e}_2\cdot 0$. A rotation of ϕ rad on this plane produces,

$$\mathbf{x}'_\perp = \mathbf{e}_1\cos\phi + \mathbf{e}_2\sin\phi \ , \tag{49}$$

which develops as,

$$\mathbf{x}'_\perp = \mathbf{x}_\perp\cos\phi + (\mathbf{u}\times\mathbf{x})\sin\phi \ , \tag{50}$$

yielding the expression of the rotated vector, $\mathbf{x}' = \mathbf{x}'_{||} + \mathbf{x}'_\perp$, which is known as the *vector rotation formula*,

$$\boxed{\mathbf{x}' = \mathbf{x}_{||} + \mathbf{x}_\perp\cos\phi + (\mathbf{u}\times\mathbf{x})\sin\phi} \ . \tag{51}$$

1.4.2 The $SO(3)$ group and the rotation matrix

Rotations are rigid transformations that leave one point unchanged: the origin. They can be defined from the metrics of Euclidean space, constituted by the dot and cross products.

Rotations are the set of operators r that keep the norms of the vectors joining the origin with any point in the solid,

$$\|r(\mathbf{v})\| = \sqrt{r(\mathbf{v})^\top r(\mathbf{v})} = \sqrt{\mathbf{v}^\top \mathbf{v}} \triangleq \|\mathbf{v}\| \;, \quad \forall \mathbf{v} \in \mathbb{R}^3 \;, \tag{52}$$

as well as their relative angles,

$$r(\mathbf{v})^\top r(\mathbf{w}) = \mathbf{v}^\top \mathbf{w} = \|\mathbf{v}\|\|\mathbf{w}\| \cos\alpha \;, \quad \forall \mathbf{v}, \mathbf{w} \in \mathbb{R}^3 \;, \tag{53}$$

and their relative orientations,

$$r(\mathbf{u}) \times r(\mathbf{v}) = r(\mathbf{w}) \iff \mathbf{u} \times \mathbf{v} = \mathbf{w} \;. \tag{54}$$

It is easily proved that the first two conditions are equivalent. We can thus define the set of rotations as,

$$SO(3) : \{r : \mathbb{R}^3 \to \mathbb{R}^3 \,/\, \forall\, \mathbf{v}, \mathbf{w} \in \mathbb{R}^3 \;,\; \|r(\mathbf{v})\| = \|\mathbf{v}\| \;,\; r(\mathbf{v}) \times r(\mathbf{w}) = r(\mathbf{v} \times \mathbf{w})\} \tag{55}$$

where its name $SO(3)$ is explained by the fact that r can be represented by the set of proper or special orthogonal matrices, as we explore now.

The operator r is linear, and can therefore be represented by a matrix $\mathbf{R} \in \mathbb{R}^{3\times 3}$,

$$r(\mathbf{v}) = \mathbf{R}\,\mathbf{v} \tag{56}$$

Injecting it in (52) and developing we have that for all $\mathbf{v}$,

$$(\mathbf{R}\mathbf{v})^\top \mathbf{R}\mathbf{v} = \mathbf{v}^\top \mathbf{R}^\top \mathbf{R}\mathbf{v} = \mathbf{v}^\top \mathbf{v} \tag{57}$$

yielding the orthogonality condition,

$$\boxed{\mathbf{R}^\top \mathbf{R} = \mathbf{I} = \mathbf{R}\,\mathbf{R}^\top} \;, \tag{58}$$

that is, the column vectors $\mathbf{r}_i$ of $\mathbf{R}$, with $i \in \{1, 2, 3\}$, are of unit length and orthogonal to each other,

$$\langle \mathbf{r}_i, \mathbf{r}_j \rangle = \mathbf{r}_i^\top \mathbf{r}_j = 1 \;, \quad \text{if } i = j \tag{59}$$

$$\langle \mathbf{r}_i, \mathbf{r}_j \rangle = \mathbf{r}_i^\top \mathbf{r}_j = 0 \;, \quad \text{if } i \neq j \;. \tag{60}$$

The set of transformations keeping vector norms and angles is for this reason called the *orthogonal group*, denoted $O(3)$. The notion of group here means essentially that the product of two orthogonal matrices is always an orthogonal matrix.[4] The orthogonality condition (58) also implies that the inverse rotation is achieved with the transposed matrix,

$$\mathbf{R}^{-1} = \mathbf{R}^\top \;. \tag{61}$$

Adding the orientation condition (54) renders the matrix $\mathbf{R}$ *proper*, or *special*, and results in one additional constraint on $\mathbf{R}$,

$$\boxed{\det(\mathbf{R}) = 1} \;, \tag{62}$$

(notice that the case of reflections satisfies $\det(\mathbf{R}) = -1$). The set of rotation matrices forms a subgroup of $O(3)$, called the *special orthogonal* group $SO(3)$. The product of two rotation matrices is always a rotation matrix.

[4] Let $\mathbf{Q}$ and $\mathbf{R}$ be two orthogonal matrices, and build $\mathbf{P} = \mathbf{Q}\,\mathbf{R}$. Then $\mathbf{P}^\top \mathbf{P} = \mathbf{R}^\top \mathbf{Q}^\top \mathbf{Q}\mathbf{R} = \mathbf{R}^\top \mathbf{I}\mathbf{R} = \mathbf{I}$.

1.4.3 The exponential map $\mathfrak{so}(3) \to SO(3)$

Rotations constitute rigid motions. This means that it is possible to define a continuous trajectory or *path* $r(t)$ in $SO(3)$ that continuously rotates the rigid body from its initial $r(0)$ to its final $r(t)$ orientations. Being continuous, it is legitimate to investigate the time-derivatives of such transformations, which we do for the rotation matrix representation $\mathbf{R}(t)$ as follows. First of all, we notice that it is impossible to continuously escape the unit determinant condition, because this would imply a jump from $+1$ to -1. Therefore we only need to investigate the time-derivative of the orthogonality condition (58),

$$(\dot{\mathbf{R}^\top \mathbf{R}}) = \dot{\mathbf{R}}^\top \mathbf{R} + \mathbf{R}^\top \dot{\mathbf{R}} = 0 \ , \tag{63}$$

which results in

$$\dot{\mathbf{R}}^\top \mathbf{R} = -(\dot{\mathbf{R}}^\top \mathbf{R})^\top \ , \tag{64}$$

meaning that the matrix $\dot{\mathbf{R}}^\top \mathbf{R}$ is skew-symmetric. The set of skew-symmetric 3×3 matrices is denoted $\mathfrak{so}(3)$. These matrices have 3 DOF and can be always written as in (20), thus establishing a one-to-one mapping $\boldsymbol{\omega} \in \mathbb{R}^3 \to [\boldsymbol{\omega}]_\times \in \mathfrak{so}(3)$, and constitute the *tangent space* to $SO(3)$, also known as its *Lie algebra*. Let us then take a vector $\boldsymbol{\omega} \in \mathbf{R}^3$ and write

$$\dot{\mathbf{R}}^\top \mathbf{R} = [\boldsymbol{\omega}]_\times \ , \tag{65}$$

which leads to

$$\dot{\mathbf{R}} = \mathbf{R}\, [\boldsymbol{\omega}]_\times \ , \tag{66}$$

and let us call $\boldsymbol{\omega}$ the vector of instantaneous angular velocities. If $\boldsymbol{\omega}$ is constant, this differential equation can be integrated as

$$\mathbf{R}(t) = \mathbf{R}(0)\, e^{[\boldsymbol{\omega}]_\times t} = \mathbf{R}(0)\, e^{[\boldsymbol{\omega} t]_\times} \tag{67}$$

where $e^{[x]_\times}$ is defined by its Taylor series, as we see in the following section. Assuming $\mathbf{R}(0)$ is a rotation matrix, then clearly $\mathbf{R} \triangleq e^{[\boldsymbol{\omega} t]_\times}$ is a rotation matrix. Defining the vector $\mathbf{v} \triangleq \boldsymbol{\omega}\Delta t$, we have

$$\boxed{\mathbf{R} = e^{[\mathbf{v}]_\times}} \ . \tag{68}$$

This is known as the exponential map, an application from $\mathfrak{so}(3)$ to $SO(3)$,

$$\exp : [\mathbf{v}]_\times \in \mathfrak{so}(3) \to e^{[\mathbf{v}]_\times} \in SO(3) \ , \tag{69a}$$

which we often express with some abuse of notation, *i.e.*, confounding $\mathbf{v} \in \mathbb{R}^3$ with $[\mathbf{v}]_\times \in \mathfrak{so}(3)$, as in

$$\exp : \mathbf{v} \in \mathfrak{so}(3) \to e^{[\mathbf{v}]_\times} \in SO(3) \ . \tag{69b}$$

In the following sections we'll see that the vector $\mathbf{v}$, called the rotation vector, encodes through $\mathbf{v} = \boldsymbol{\omega}\Delta t = \phi\mathbf{u}$ the angle ϕ and axis $\mathbf{u}$ of rotation.

1.4.4 Rotation matrix and rotation vector

The rotation matrix is defined from the rotation vector $\mathbf{v} = \phi\mathbf{u}$ through the exponential map (68), with the cross-product matrix $[\mathbf{v}]_\times$ as defined in (20). When applied to unit vectors, $\mathbf{u}$, the matrix $[\mathbf{u}]_\times$ satisfies

$$[\mathbf{u}]_\times^2 = \mathbf{u}\mathbf{u}^\top - \mathbf{I} \tag{70}$$

$$[\mathbf{u}]_\times^3 = -[\mathbf{u}]_\times \ , \tag{71}$$

and thus all powers of $[\mathbf{u}]_\times$ can be expressed in terms of $[\mathbf{u}]_\times$ and $[\mathbf{u}]_\times^2$ in a cyclic pattern,

$$[\mathbf{u}]_\times^4 = -[\mathbf{u}]_\times^2 \qquad [\mathbf{u}]_\times^5 = [\mathbf{u}]_\times \qquad [\mathbf{u}]_\times^6 = [\mathbf{u}]_\times^2 \qquad [\mathbf{u}]_\times^7 = -[\mathbf{u}]_\times \quad \cdots \ . \tag{72}$$

Now, writing the Taylor expansion of (68) with $\mathbf{v} = \phi\mathbf{u}$,

$$e^{\phi[\mathbf{u}]_\times} = \mathbf{I} + \phi\,[\mathbf{u}]_\times + \frac{1}{2}\phi^2\,[\mathbf{u}]_\times^2 + \frac{1}{3}\phi^3\,[\mathbf{u}]_\times^3 + \dots \tag{73}$$

grouping in terms of $[\mathbf{u}]_\times$ and $[\mathbf{u}]_\times^2$, and identifying in them, respectively, the series of $\sin\phi$ and $\cos\phi$, leads to a closed form to obtain the rotation matrix from the rotation vector, the so called *Rodrigues rotation formula*,

$$\mathbf{R} = e^{[\mathbf{v}]_\times} = \mathbf{I} + \sin\phi\,[\mathbf{u}]_\times + (1-\cos\phi)\,[\mathbf{u}]_\times^2 \ , \tag{74}$$

which we denote $\mathbf{R} = \mathbf{R}\{\mathbf{v}\}$.

Rotating a vector $\mathbf{x}$ by ϕ rad around the unit axis $\mathbf{u}$ following the right-hand rule is performed with the linear product

$$\mathbf{x}' = \mathbf{R}\,\mathbf{x} \ , \tag{75}$$

as can be shown by developing (75), using (74), (21) and (70),

$$\begin{aligned}
\mathbf{x}' &= \mathbf{R}\,\mathbf{x} \\
&= (\mathbf{I} + \sin\phi\,[\mathbf{u}]_\times + (1-\cos\phi)\,[\mathbf{u}]_\times^2)\,\mathbf{x} \\
&= \mathbf{x} + \sin\phi\,[\mathbf{u}]_\times\,\mathbf{x} + (1-\cos\phi)\,[\mathbf{u}]_\times^2\,\mathbf{x} \\
&= \mathbf{x} + \sin\phi(\mathbf{u}\times\mathbf{x}) + (1-\cos\phi)(\mathbf{u}\mathbf{u}^\top - \mathbf{I})\,\mathbf{x} \\
&= \mathbf{x}_\parallel + \mathbf{x}_\perp + \sin\phi(\mathbf{u}\times\mathbf{x}) - (1-\cos\phi)\,\mathbf{x}_\perp \\
&= \mathbf{x}_\parallel + (\mathbf{u}\times\mathbf{x})\sin\phi + \mathbf{x}_\perp\cos\phi \ ,
\end{aligned} \tag{76}$$

which is precisely the vector rotation formula (51).

1.4.5 Quaternion and rotation vector

Given the rotation vector $\mathbf{v} = \phi\mathbf{u}$, representing a right-handed rotation of ϕ rad around the axis given by the unit vector $\mathbf{u} = u_x i + u_y j + u_z k$, we build the unit quaternion through another exponential map, $\mathbf{v} \in \mathbb{R}^3 \to e^{\mathbf{v}/2} \in \mathbb{H}$, this time based on the quaternion product,

$$\mathbf{q} = e^{\mathbf{v}/2} \ , \tag{77}$$

which can be developed using an extension of the *Euler formula* (see (35–38)),

$$\mathbf{q} = e^{\mathbf{v}/2} = \cos\frac{\phi}{2} + (u_x i + u_y j + u_z k)\sin\frac{\phi}{2} = \begin{bmatrix}\cos(\phi/2)\\ \mathbf{u}\sin(\phi/2)\end{bmatrix} . \tag{78}$$

We call this the *rotation vector to quaternion conversion*, denoted by $\mathbf{q} = \mathbf{q}\{\mathbf{v}\}$. Conversely, using the 4-quadrant version of $\arctan(y, x)$ we have

$$\phi = 2\arctan(\|\mathbf{q}_v\|, q_w) \tag{79}$$

$$\mathbf{u} = \mathbf{q}_v/\|\mathbf{q}_v\| . \tag{80}$$

Rotating a vector $\mathbf{x}$ by ϕ around $\mathbf{u}$ is performed with the double quaternion product, also known as the sandwich product,

$$\mathbf{x}' = \mathbf{q} \otimes \mathbf{x} \otimes \mathbf{q}^* , \tag{81}$$

where the vector $\mathbf{x}$ has been written in quaternion form, that is,

$$\mathbf{x} = xi + yj + zk = \begin{bmatrix}0\\ \mathbf{x}\end{bmatrix} . \tag{82}$$

To see it, we use (13), (78), and basic vector and trigonometric identities, to develop the product (81) as follows,

$$\begin{aligned}
\mathbf{x}' &= \Big(\cos\frac{\phi}{2} + \mathbf{u}\sin\frac{\phi}{2}\Big) \otimes \mathbf{x} \otimes \Big(\cos\frac{\phi}{2} - \mathbf{u}\sin\frac{\phi}{2}\Big)\\
&= \mathbf{x}\cos^2\frac{\phi}{2} + (\mathbf{u}\otimes\mathbf{x} - \mathbf{x}\otimes\mathbf{u})\sin\frac{\phi}{2}\cos\frac{\phi}{2} - \mathbf{u}\otimes\mathbf{x}\otimes\mathbf{u}\sin^2\frac{\phi}{2}\\
&= \mathbf{x}\cos^2\frac{\phi}{2} + 2(\mathbf{u}\times\mathbf{x})\sin\frac{\phi}{2}\cos\frac{\phi}{2} - (\mathbf{x}(\mathbf{u}^\top\mathbf{u}) - 2\mathbf{u}(\mathbf{u}^\top\mathbf{x}))\sin^2\frac{\phi}{2}\\
&= \mathbf{x}(\cos^2\frac{\phi}{2} - \sin^2\frac{\phi}{2}) + (\mathbf{u}\times\mathbf{x})(2\sin\frac{\phi}{2}\cos\frac{\phi}{2}) + \mathbf{u}(\mathbf{u}^\top\mathbf{x})(2\sin^2\frac{\phi}{2})\\
&= \mathbf{x}\cos\phi + (\mathbf{u}\times\mathbf{x})\sin\phi + \mathbf{u}(\mathbf{u}^\top\mathbf{x})(1-\cos\phi)\\
&= (\mathbf{x} - \mathbf{u}\,\mathbf{u}^\top\mathbf{x})\cos\phi + (\mathbf{u}\times\mathbf{x})\sin\phi + \mathbf{u}\,\mathbf{u}^\top\mathbf{x}\\
&= \mathbf{x}_\perp\cos\phi + (\mathbf{u}\times\mathbf{x})\sin\phi + \mathbf{x}_{||} ,
\end{aligned} \tag{83}$$

which is precisely the vector rotation formula (51).

1.4.6 Rotation matrix and quaternion

Unit quaternions act as rotation operators in a way somewhat similar to rotation matrices,

$$\overline{\mathbf{x}}' = \mathbf{q} \otimes \overline{\mathbf{x}} \otimes \mathbf{q}^* , \qquad\qquad \mathbf{x}' = \mathbf{R}\,\mathbf{x} .$$

Here, we used the bar notation $\overline{\mathbf{x}}$ to indicate that the vector $\mathbf{x}$ in the left expression is expressed in quaternion form (82), thus differentiating it from that on the right. However, this circumstance is mostly unambiguous and can be derived from the context, and especially by the presence of the quaternion product $\otimes$. In what is to follow, we are omitting this bar and writing simply, $\mathbf{x}' = \mathbf{q} \otimes \mathbf{x} \otimes \mathbf{q}^*$. This allows us to abuse of the notation and write[5],

$$\mathbf{q} \otimes \mathbf{x} \otimes \mathbf{q}^* = \mathbf{R}\,\mathbf{x} \,. \tag{84}$$

As both sides of this identity are linear in $\mathbf{x}$, an expression of the rotation matrix equivalent to the quaternion is found by developing the left hand side and identifying terms on the right, yielding

$$\mathbf{R} = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_x q_y + q_w q_z) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} , \tag{85}$$

denoted throughout this document by $\mathbf{R} = \mathbf{R}\{\mathbf{q}\}$. The rotation matrix $\mathbf{R}$ has the following properties with respect to the quaternion,

$$\mathbf{R}\{[1,0,0,0]^\top\} = \mathbf{I} \tag{86}$$

$$\mathbf{R}\{-\mathbf{q}\} = \mathbf{R}\{\mathbf{q}\} \tag{87}$$

$$\mathbf{R}\{\mathbf{q}^*\} = \mathbf{R}\{\mathbf{q}\}^\top \tag{88}$$

$$\mathbf{R}\{\mathbf{q}_1 \otimes \mathbf{q}_2\} = \mathbf{R}\{\mathbf{q}_1\}\mathbf{R}\{\mathbf{q}_2\} \,, \tag{89}$$

where we observe that: (86) the identity quaternion encodes the null rotation; (87) a quaternion and its negative encode the same rotation; (88) the conjugate quaternion encodes the inverse rotation; and (89) the quaternion product composes consecutive rotations in the same order as rotation matrices do.

The matrix form of the quaternion product (17–19) provides us with an alternative formula for the rotation matrix, since

$$\mathbf{q} \otimes \overline{\mathbf{x}} \otimes \mathbf{q}^* = [\mathbf{q}^*]_R \, [\mathbf{q}]_L \begin{bmatrix} 0 \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{R}\,\mathbf{x} \end{bmatrix} \,, \tag{90}$$

which leads after some easy developments to

$$\boxed{\mathbf{R} = (q_w^2 - \mathbf{q}_v^\top \mathbf{q}_v)\,\mathbf{I} + 2\,\mathbf{q}_v \mathbf{q}_v^\top + 2\,q_w\,[\mathbf{q}_v]_\times} \,. \tag{91}$$

1.4.7 Quaternion and isoclinic rotations

We note that, for unit quaternions $\mathbf{q}$, the quaternion product matrices $[\mathbf{q}]_L$ and $[\mathbf{q}]_R$ defined in (19) satisfy,

$$[\mathbf{q}]\,[\mathbf{q}]^\top = \mathbf{I}_4 \tag{92}$$

$$\det([\mathbf{q}]) = +1 \tag{93}$$

$$[\mathbf{q}]^\top = [\mathbf{q}]^{-1} \,, \tag{94}$$

[5]The proper expressions would be, $\mathbf{q} \otimes \overline{\mathbf{x}} \otimes \mathbf{q}^* = \overline{\mathbf{R}\mathbf{x}}$, or $\mathbf{q} \otimes \begin{bmatrix} 0 \\ \mathbf{x} \end{bmatrix} \otimes \mathbf{q}^* = \begin{bmatrix} 0 \\ \mathbf{R}\mathbf{x} \end{bmatrix}$. See Section 1.2.

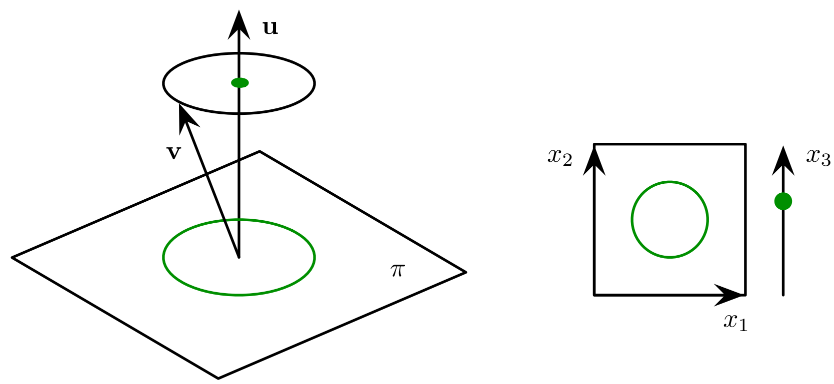

Figure 2: Rotation in $\mathbb{R}^3$. A rotation of a vector $\mathbf{v}$ around an axis $\mathbf{u}$ describes a circumference in a plane orthogonal to the axis. We represent this plane at the base of the vector with π, and project the circumference on it (green circle). The two components of $\mathbf{v}$ orthogonal to the axis rotate along the green circumference. The component of $\mathbf{v}$ parallel to the axis does not move, and is represented by the small green dot on the axis. The sketch on the right illustrates the radically different behaviors of the rotating point on the plane and axis dimensions.

and are therefore proper rotation matrices in the $\mathbb{R}^4$ space. To be more specific, they represent isoclinic rotations.

In order to understand isoclinic rotations in $\mathbb{R}^4$, we first need to understand general rotations in $\mathbb{R}^4$. And to understand rotations in $\mathbb{R}^4$, we first need to go back to $\mathbb{R}^3$.

In $\mathbb{R}^3$, let us consider the rotations around an arbitrary axis represented by the vector $\mathbf{u}$ —see Fig. 2. Upon rotation, vectors parallel to the axis of rotation $\mathbf{u}$ do not move, and vectors perpendicular to the axis rotate equally in a plane perpendicular to this axis. For general vectors $\mathbf{v}$, we have the two components in the plane rotating in this plane, while the axial component remains static.

In $\mathbb{R}^4$, see Fig. 3, due to the extra dimension, the one-dimensional axis of rotation of $\mathbb{R}^3$ becomes a new two-dimensional plane. Therefore, in this new plane an independent rotation can take place. Indeed, rotations in $\mathbb{R}^4$ encompass two independent rotations in two orthogonal 2-planes of the 4-space. This means that every 4-vector of each of these 2-planes rotates in its own plane, and that rotations of general 4-vectors with respect to one plane leave unaffected the vector components in the other plane. These 2-planes are for this reason called *'invariant'*.

Isoclinic rotations are those rotations in $\mathbb{R}^4$ where the angles of rotation in the two invariant planes have the same magnitude. Then, when the two angles have also the same sign, we speak of left-isoclinic rotations. And when they have opposite signs, we speak of right-isoclinic rotations. A remarkable property of isoclinic rotations, that we had already seen in (23), is that left- and right- isoclinic rotations commute,

$$[\mathbf{p}]_R\,[\mathbf{q}]_L = [\mathbf{q}]_L\,[\mathbf{p}]_R\ . \tag{95}$$

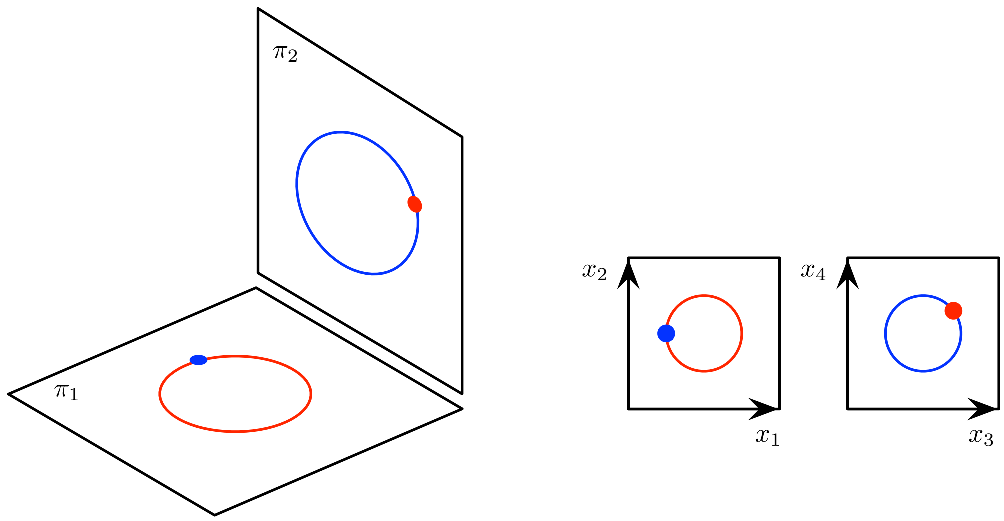

Figure 3: Rotations in $\mathbb{R}^4$. Two orthogonal rotations are possible, on two orthogonal planes π_1 and π_2. Rotations of a vector $\mathbf{v}$ (not drawn) in the plane π_1 cause the two first components of the vector to describe a circumference (red circle), leaving the other two components in π_2 unchanged (the red dot). Conversely, rotations in the plane π_2 (blue circle) leave the components in π_1 unchanged (blue dot). The sketch on the right better illustrates the situation by resigning to draw unrepresentable perspectives in $\mathbb{R}^4$, which might be misleading.

Given a unit quaternion $\mathbf{q} = e^{\mathbf{u}\theta/2}$, representing a rotation in $\mathbb{R}^3$ of θ around $\mathbf{u}$, the matrix $[\mathbf{q}]_L$ is a left-isoclinic rotation in $\mathbb{R}^4$ corresponding to the left-multiplication by the quaternion $\mathbf{q}$, and $[\mathbf{q}^*]_R$ is a right-isoclinic rotation, corresponding to the right-multiplication by the quaternion $\mathbf{q}^*$. The angles of these isoclinic rotations are exactly of magnitude $\theta/2$.[6] Then, the expression (90) represents two chained isoclinic rotations, one left- and one right-, to the 4-vector $(0, \mathbf{x})^\top$, each by half the desired rotation angle in $\mathbb{R}^3$. In one of the invariant 2-planes of $\mathbb{R}^4$, the two half angles cancel out, because they have opposite signs. In the other plane, they sum up to yield the total rotation angle θ. If we define from (90) the resulting rotation matrix, $\mathbf{R}_4$, one easily realizes that (see also (95)),

$$\mathbf{R}_4 \triangleq [\mathbf{q}^*]_R[\mathbf{q}]_L = [\mathbf{q}]_L[\mathbf{q}^*]_R = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} , \tag{96}$$

where $\mathbf{R}$ is the rotation matrix in $\mathbb{R}^3$, which clearly rotates the $\mathbb{R}^3$ subspace of $\mathbb{R}^4$, leaving the fourth dimension unchanged.

This discourse is somewhat beyond the scope of the present document. We include it here just as a means for providing yet another way to interpret rotations by quaternions,

[6] This can be checked by extracting the eigenvalues of the isoclinic rotation matrices: they are formed by pairs of conjugate complex numbers with a phase equal to $\theta/2$.

with the hope that the reader grasps more intuition about its mechanisms. The interested reader is suggested to consult the appropriate literature on isoclinic rotations in $\mathbb{R}^4$.

1.5 Quaternion conventions. My choice.

There are several ways to determine the quaternion. They are basically related to four binary choices:

- The order of its elements — real part first or last:

$$\mathbf{q} = \begin{bmatrix} q_w \\ \mathbf{q}_v \end{bmatrix} \qquad vs. \qquad \mathbf{q} = \begin{bmatrix} \mathbf{q}_v \\ q_w \end{bmatrix} \ . \tag{97}$$

- The multiplication formula — definition of the quaternion algebra:

$$ij = -ji = k \qquad vs. \qquad ji = -ij = k \ , \tag{98a}$$

  which correspond to different handedness, respectively:

$$\textit{right-handed} \qquad vs. \qquad \textit{left-handed} \ , \tag{98b}$$

  which means that, given a rotation axis $\mathbf{u}$, one quaternion $\mathbf{q}\{\mathbf{u}\,\theta\}$ rotates vectors an angle θ around $\mathbf{u}$ using the right hand rule, while the other quaternion uses the left hand rule.

- The function of the rotation operator — rotating frames or rotating vectors:

$$\textit{Passive} \qquad vs. \qquad \textit{Active.} \tag{99}$$

- In the passive case, the direction of the operation — local-to-global or global-to-local:

$$\mathbf{x}_{global} = \mathbf{q} \otimes \mathbf{x}_{local} \otimes \mathbf{q}^* \qquad vs. \qquad \mathbf{x}_{local} = \mathbf{q} \otimes \mathbf{x}_{global} \otimes \mathbf{q}^* \tag{100}$$

This variety of choices leads to 12 different combinations. Historical developments have favored some conventions over others (Chou, 1992; Yazell, 2009). Today, in the available literature, we find many quaternion flavors such as the Hamilton, the STS[7], the JPL[8], the ISS[9], the ESA[10], the Engineering, the Robotics, and possibly a lot more denominations. Many of these forms might be identical, others not, but this fact is rarely explicitly stated, and many works simply lack a sufficient description of their quaternion with regard to the four choices above.

The two most commonly used conventions, which are also the best documented, are Hamilton (the options on the left in (97–100)) and JPL (the options on the right, with the exception of (99)). Table 1 shows a summary of their characteristics. JPL is mostly used

Table 1: Hamilton vs. JPL quaternion conventions with respect to the 4 binary choices

	Quaternion type	Hamilton	JPL
1	Components order	$(q_w\ ,\ \mathbf{q}_v)$	$(\mathbf{q}_v\ ,\ q_w)$
2	Algebra	$ij = k$	$ij = -k$
	Handedness	Right-handed	Left-handed
3	Function	Passive	Passive
4	Right-to-left products mean	Local-to-Global	Global-to-Local
	Default notation, $\mathbf{q}$	$\mathbf{q} \triangleq \mathbf{q}_{\mathcal{GL}}$	$\mathbf{q} \triangleq \mathbf{q}_{\mathcal{LG}}$
	Default operation	$\mathbf{x}_\mathcal{G} = \mathbf{q} \otimes \mathbf{x}_\mathcal{L} \otimes \mathbf{q}^*$	$\mathbf{x}_\mathcal{L} = \mathbf{q} \otimes \mathbf{x}_\mathcal{G} \otimes \mathbf{q}^*$

in the aerospace domain, while Hamilton is more common to other engineering areas such as robotics —though this should not be taken as a rule.

My choice, which has been taken as early as in equation (2), is to take the Hamilton convention, which is right-handed and coincides with many software libraries of widespread use in robotics, such as Eigen, ROS, Google Ceres, and with a vast amount of literature on Kalman filtering for attitude estimation using IMUs (Chou, 1992; Kuipers, 1999; Piniés et al., 2007; Roussillon et al., 2011; Martinelli, 2012, and many others).

The JPL convention is possibly less commonly used, at least in the robotics field. It is extensively described in (Trawny and Roumeliotis, 2005), a reference work that has an aim and scope very close to the present one, but that concentrates exclusively in the JPL convention. The JPL quaternion is used in the JPL literature (obviously) and in key papers by Li, Mourikis, Roumeliotis, and colleagues (see *e.g.* (Li and Mourikis, 2012; Li et al., 2014)), which draw from Trawny and Roumeliotis' document. These works are a primary source of inspiration when dealing with visual-inertial odometry and SLAM —which is what we do.

In the rest of this section we analyze these two quaternion conventions with a little more depth.

1.5.1 Order of the quaternion components

Though not the most fundamental, the most salient difference between Hamilton and JPL quaternions is in the order of the components, with the scalar part being either in first (Hamilton) or last (JPL) position. The implications of such change are quite obvious and should not represent a great challenge of interpretation. In fact, some works with the quaternion's real component at the end (*e.g.*, the C++ library Eigen) are still considered

[7]Space Transportation System, commonly known as NASA's Space Shuttle.

[8]Jet Propulsion Laboratory.

[9]International Space Station.

[10]European Space Agency.

as using the Hamilton convention, as long as the other three aspects are maintained.

We have used the subscripts (w, x, y, z) for the quaternion components for increased clarity, instead of the other commonly used $(0, 1, 2, 3)$. When changing the order, q_w will always denote the real part, while it is not clear whether q_0 would also do —in some occasions, one might find things such as $\mathbf{q} = (q_1, q_2, q_3, q_0)$, with q_0 real and last, but in the general case of $\mathbf{q} = (q_0, q_1, q_2, q_3)$, the real part at the end would be q_3.[11] When passing from one convention to the other, we must be careful of formulas involving full 4×4 or 3×4 quaternion-related matrices, for their rows and/or columns need to be swapped. This is not difficult to do, but it might be difficult to detect and therefore prone to error.

1.5.2 Specification of the quaternion algebra

The Hamilton convention defines $ij = k$ and therefore,

$$i^2 = j^2 = k^2 = ijk = -1 \ , \quad ij = -ji = k \ , \quad jk = -kj = i \ , \quad ki = -ik = j \ , \tag{101}$$

whereas the JPL convention defines $ji = k$ and hence its quaternion algebra becomes,

$$i^2 = j^2 = k^2 = -ijk = -1 \ , \quad -ij = ji = k \ , \quad -jk = kj = i \ , \quad -ki = ik = j \ . \tag{102}$$

Interestingly, these subtle sign changes preserve the basic properties of quaternions as rotation operators. However, they have the important consequence of changing the quaternion handedness (Shuster, 1993): Hamilton uses $ij = k$ and is therefore right-handed, *i.e.*, it turns vectors following the right-hand rule; JPL uses $ji = k$ and is left-handed (Trawny and Roumeliotis, 2005). Being left- and right- handed rotations of opposite signs, we can say that their quaternions $\mathbf{q}_{left}$ and $\mathbf{q}_{right}$ are related by,

$$\mathbf{q}_{left} = \mathbf{q}^*_{right} \ . \tag{103}$$

These sign differences impact the respective formulas for rotation, composition, etc., in non-obvious ways. The formulas are thus not compatible, and we need to make a clear choice from the very start.

1.5.3 Function of the rotation operator

We have seen how to rotate vectors in 3D. This is referred to in (Shuster, 1993) as the *Active* interpretation, because operators (this affects all rotation operators) actively rotate vectors,

$$\mathbf{x}' = \mathbf{q}_{active} \otimes \mathbf{x} \otimes \mathbf{q}^*_{active} \ , \qquad\qquad \mathbf{x}' = \mathbf{R}_{active}\, \mathbf{x} \ .$$

Another way of seeing the effect of $\mathbf{q}$ and $\mathbf{R}$ over a vector $\mathbf{x}$ is to consider that the vector is steady but it is us who have rotated our point of view by an amount specified by

[11] See also footnote 1.

$\mathbf{q}$ or $\mathbf{R}$. This is called here *frame transformation* and it is referred to in (Shuster, 1993) as the *Passive* interpretation, because vectors do not move,

$$\mathbf{x}_B = \mathbf{q}_{passive} \otimes \mathbf{x}_A \otimes \mathbf{q}^*_{passive} \ , \qquad \mathbf{x}_B = \mathbf{R}_{passive}\, \mathbf{x}_A \ , \tag{104}$$

where A and B are two Cartesian reference frames, and $\mathbf{x}_A$ and $\mathbf{x}_B$ are expressions of the same vector $\mathbf{x}$ in these frames. See further down for explanations and proper notations.

The active and passive interpretations are governed by operators inverse of each other, that is,

$$\mathbf{q}_{active} = \mathbf{q}^*_{passive} \ , \qquad \mathbf{R}_{active} = \mathbf{R}^\top_{passive} \ .$$

Both Hamilton and JPL use the passive convention.

Direction cosine matrix A few authors understand the passive operator as not being a rotation operator, but rather an orientation specification, named the *direction cosine matrix*,

$$\mathbf{C} = \begin{bmatrix} c_{xx} & c_{xy} & c_{zx} \\ c_{xy} & c_{yy} & c_{zy} \\ c_{xz} & c_{yz} & c_{zz} \end{bmatrix} \ , \tag{105}$$

where each component c_{ij} is the cosine of the angle between the axis i in the local frame and the axis j in the global frame. We have the identity,

$$\mathbf{C} \equiv \mathbf{R}_{passive} \ . \tag{106}$$

1.5.4 Direction of the rotation operator

In the passive case, a second source of interpretation is related to the direction in which the rotation matrix and quaternion operate, either converting from local to global frames, or from global to local.

Given two Cartesian frames $\mathcal{G}$ and $\mathcal{L}$, we identify $\mathcal{G}$ and $\mathcal{L}$ as being the global and local frames. "Global" and "local" are relative definitions, *i.e.*, $\mathcal{G}$ is global with respect to $\mathcal{L}$, and $\mathcal{L}$ is local with respect to $\mathcal{G}$. We specify $\mathbf{q}_{\mathcal{GL}}$ and $\mathbf{R}_{\mathcal{GL}}$ as being respectively the quaternion and rotation matrix transforming vectors from frame $\mathcal{L}$ to frame $\mathcal{G}$, in the sense that a vector $\mathbf{x}_\mathcal{L}$ in frame $\mathcal{L}$ is expressed in frame $\mathcal{G}$ with the quaternion- and matrix- products

$$\mathbf{x}_\mathcal{G} = \mathbf{q}_{\mathcal{GL}} \otimes \mathbf{x}_\mathcal{L} \otimes \mathbf{q}^*_{\mathcal{GL}} \ , \qquad \mathbf{x}_\mathcal{G} = \mathbf{R}_{\mathcal{GL}}\, \mathbf{x}_\mathcal{L} \ . \tag{107}$$

The opposite conversion, from $\mathcal{G}$ to $\mathcal{L}$, is done with

$$\mathbf{x}_\mathcal{L} = \mathbf{q}_{\mathcal{LG}} \otimes \mathbf{x}_\mathcal{G} \otimes \mathbf{q}^*_{\mathcal{LG}} \ , \qquad \mathbf{x}_\mathcal{L} = \mathbf{R}_{\mathcal{LG}}\, \mathbf{x}_\mathcal{G} \ , \tag{108}$$

where

$$\mathbf{q}_{\mathcal{LG}} = \mathbf{q}^*_{\mathcal{GL}} \ , \qquad \mathbf{R}_{\mathcal{LG}} = \mathbf{R}^\top_{\mathcal{GL}} \ . \tag{109}$$

Hamilton uses local-to-global as the default specification of a frame $\mathcal{L}$ expressed in frame $\mathcal{G}$,

$$\mathbf{q}_{Hamilton} \triangleq \mathbf{q}_{[with\ respect\ to][of]} = \mathbf{q}_{[to][from]} = \mathbf{q}_{\mathcal{GL}} \ , \tag{110}$$

while JPL uses the opposite, global-to-local conversion,

$$\mathbf{q}_{JPL} \triangleq \mathbf{q}_{[of][with\ respect\ to]} = \mathbf{q}_{[to][from]} = \mathbf{q}_{\mathcal{LG}} \ . \tag{111}$$

Notice that

$$\mathbf{q}_{JPL} \triangleq \mathbf{q}_{\mathcal{LG},left} = \mathbf{q}^*_{\mathcal{LG},right} = \mathbf{q}_{\mathcal{GL},right} \triangleq \mathbf{q}_{Hamilton} \ , \tag{112}$$

which is not particularly useful, but illustrates how easy it is to get confused when mixing conventions. Notice also that we can conclude that $\mathbf{q}_{JPL} = \mathbf{q}_{Hamilton}$, but this, far from being a beautiful result, is just the source of great confusion, because the equality is only present in the quaternion values, but the two quaternions, when employed in formulas, mean and represent different things.

1.5.5 Notation

An often underestimated source of confusion when dealing with quaternion algebra is related to notation. We believe notation should be clear, lightweight and unambiguous. Of course, this applies not only to quaternions and rotation matrices, but also to the points and vectors manipulated by these. A good notation requires considering many conflicting aspects:

- Clearly distinguish scalars, vectors, matrices and functions with different font styles.
- Use the main letter to signify the physical dimension. Use prefixes, subscripts, superscripts and/or accents for details and particularities.
- Avoid tiny elements such as tildes $\tilde{\mathbf{q}}$, hats $\hat{\mathbf{q}}$, bars $\bar{\mathbf{q}}$, $\underline{\mathbf{q}}$, and other accents, as much as possible.
- Avoid certain combinations of subscripts and superscripts on the left and right hand sides, especially when they appear in multiple levels, *e.g.*, ${}^{C_i}x_{F_j}$. They produce formulas such as ${}^{i}u_j = \frac{{}^{C_i}x_{F_j}}{{}^{C_i}z_{F_j}}$ (which is the pinhole camera model) that are difficult to read because the main variables, x and z, are not salient enough.
- Make composition of chains obviously and unambiguously readable. For example, the rotation matrix $\mathbf{R}_A^B$ or the quaternion ${}^B_A\mathbf{q}$ are ambiguous because we do not know if they transform "A to B" or "B to A".
- Provide easy 1:1 translation to readable programming code. This last point becomes increasingly important as most of the works we produce are meant to be translated into algorithms.

Our notation derives from the following rules,

1. Scalars are a, x, ω; vectors are $\mathbf{a}, \mathbf{x}, \boldsymbol{\omega}$, matrices are $\mathbf{A}, \mathbf{X}, \mathbf{\Omega}$; functions are $f(), g()$.

2. We use decorations to provide details of a given magnitude. As an example, we use $\mathbf{a}$ for *acceleration*, and $\delta\mathbf{a},\ \mathbf{a}_m,\ \mathbf{a}_b,\ \hat{\mathbf{a}},\ \mathbf{A},\ \mathbf{a}_n,\ \mathbf{A}_n$ respectively for *acceleration error, measured acceleration, accelerometer bias, mean of the acceleration estimate, covariance of the acceleration estimate, acceleration noise, covariance of the acceleration noise.*

3. The only accent we use is the hat, $\hat{\mathbf{x}}$, to signify the mean of the Kalman filter Gaussian estimate. Error-state values are noted $\delta\mathbf{x}$.

4. The general translation specification $\mathbf{t}$ needs 3 decorations: a translation *from* point B / *to* point C / *expressed in* frame A,

$$\mathbf{t}_{[expressed\ in],[initial\ point][end\ point]}\ . \tag{113}$$

   This produces forms such as $\mathbf{t}_{A,BC}$. We define two convenient simplifications:

   - When $\mathbf{t}_{A,BC}$ refers to a point P, we consider its origin at the origin of the reference frame, *i.e.*, $B = A$, giving

$$\mathbf{p}_A \triangleq \mathbf{t}_{A,AP}\ . \tag{114}$$

   - When $\mathbf{t}_{A,BC}$ refers to a free vector $\mathbf{v} = \overline{BC}$, we may simply write

$$\mathbf{v}_A \triangleq \mathbf{t}_{A,\overline{BC}}\ . \tag{115}$$

   In both cases we keep the reference frame A where the point or vector is expressed in. In case of no ambiguity, when this frame is the world frame $\mathcal{W}$ of our application we allow us to drop this decoration too. For example, the position P of the robot in the world frame may be denoted simply by $\mathbf{p}$,

$$\mathbf{p} \triangleq \mathbf{p}_{\mathcal{W}} = \mathbf{t}_{\mathcal{W},\mathcal{W}P}\ . \tag{116}$$

5. The general orientation specification, $\mathbf{q}$ or $\mathbf{R}$, requires 2 decorations: a transformation *from* frame B / *to* frame A, or equivalently, the orientation *of* frame B / *with respect to* frame A,

$$\mathbf{q}_{[to][from]} \equiv \mathbf{q}_{[with\ respect\ to][of]}\ , \qquad \mathbf{R}_{[to][from]} \equiv \mathbf{R}_{[with\ respect\ to][of]}\ . \tag{117}$$

   This produces forms such as $\mathbf{R}_{AB}$ that lead unambiguously to $\mathbf{x}_A = \mathbf{R}_{AB}\mathbf{x}_B$. Stacked or composed transforms produce chains such as

$$\mathbf{x}_A = \mathbf{q}_{AB} \otimes \mathbf{q}_{BC} \otimes \mathbf{x}_C \otimes \mathbf{q}^*_{BC} \otimes \mathbf{q}^*_{AB}\ , \qquad \mathbf{x}_A = \mathbf{R}_{AB}\,\mathbf{R}_{BC}\,\mathbf{x}_C\ . \tag{118}$$

   which are readable and not prone to error. Notice how each of the two frame identifiers is always at the side of the entity nearby, creating a *chain* of identifiers. In the quaternion case, the chain of identifiers can be made more salient by recalling that $\mathbf{q}^*_{AB} = \mathbf{q}_{BA}$ and thus

$$\mathbf{x}_A = \mathbf{q}_{AB} \otimes \mathbf{q}_{BC} \otimes \mathbf{x}_C \otimes \mathbf{q}_{CB} \otimes \mathbf{q}_{BA} \ . \tag{119}$$

This allows us to easily construct and/or identify frame composition chains with absolutely no ambiguity.

Oftentimes, the world frame $\mathcal{W}$ and the frame of the main moving body B may be omitted (in cases where there is no ambiguity), yielding

$$\mathbf{q} \triangleq \mathbf{q}_{\mathcal{W}B} \ , \qquad\qquad \mathbf{R} \triangleq \mathbf{R}_{\mathcal{W}B} \ . \tag{120}$$

This, and the simplifications for points and vectors (114–116) above, lead to

$$\mathbf{x} = \mathbf{q} \otimes \mathbf{x}_B \otimes \mathbf{q}^* \ , \qquad\qquad \mathbf{x} = \mathbf{R}\,\mathbf{x}_B \ , \tag{121}$$

which are very light and easy to read. They express the transformation of vector X from the body frame B to the world frame $\mathcal{W}$.

6. We use only right-hand subscripts for frame decorations. This way, translating formulas into code and vice-versa becomes straightforward. In the code, the first letter before the first underscore is always the physical magnitude. For example, these formula and code are equivalent,

$$\mathbf{x}_A = \mathbf{R}_{AB}\,\mathbf{R}_{BC}\,\mathbf{x}_C \ , \qquad\qquad \texttt{x\_A = R\_A\_B * R\_B\_C * x\_C} \ . \tag{122}$$

1.6 Rotation composition

As we have briefly seen in (118), quaternion composition is done similarly to rotation matrices, *i.e.*, with appropriate quaternion- and matrix- products, and in the same order,

$$\mathbf{q}_{AC} = \mathbf{q}_{AB} \otimes \mathbf{q}_{BC} \ , \qquad\qquad \mathbf{R}_{AC} = \mathbf{R}_{AB}\,\mathbf{R}_{BC} \ . \tag{123}$$

The Hamilton, local-to-global convention adopted here establishes that compositions go from local to global when moving towards the left of the composition chain, or from global to local when moving right. This means that $\mathbf{q}_{BC}$ and $\mathbf{R}_{BC}$ are specifications of a frame C that is local with respect to frame B. We illustrate the composed transform using the quaternion,

$$\begin{aligned}
\mathbf{x}_A &= \mathbf{q}_{AB} \otimes \mathbf{x}_B \otimes \mathbf{q}^*_{AB} \\
&= \mathbf{q}_{AB} \otimes (\mathbf{q}_{BC} \otimes \mathbf{x}_C \otimes \mathbf{q}^*_{BC}) \otimes \mathbf{q}^*_{AB} \\
&= (\mathbf{q}_{AB} \otimes \mathbf{q}_{BC}) \otimes \mathbf{x}_C \otimes (\mathbf{q}^*_{BC} \otimes \mathbf{q}^*_{AB}) \\
&= (\mathbf{q}_{AB} \otimes \mathbf{q}_{BC}) \otimes \mathbf{x}_C \otimes (\mathbf{q}_{AB} \otimes \mathbf{q}_{BC})^* \\
&= \mathbf{q}_{AC} \otimes \mathbf{x}_C \otimes \mathbf{q}^*_{AC} \ .
\end{aligned}$$

1.7 Perturbations and time-derivatives

1.7.1 Local perturbations

A perturbed orientation $\tilde{\mathbf{q}}$ may be expressed as the composition of the unperturbed orientation $\mathbf{q}$ with a small local perturbation $\Delta\mathbf{q}_L$. Because of the Hamilton convention, this local perturbation appears *at the right hand side* of the composition product —we give also the matrix equivalent for comparison,

$$\tilde{\mathbf{q}} = \mathbf{q} \otimes \Delta\mathbf{q}_L \;, \qquad\qquad \tilde{\mathbf{R}} = \mathbf{R}\,\Delta\mathbf{R}_L \;. \tag{124}$$

If the perturbation angle $\Delta\boldsymbol{\theta}_L \triangleq \Delta\phi{\cdot}\mathbf{u}$ is small then the perturbation quaternion and rotation matrix can be approximated by the Taylor expansions of (78) and (68) up to the linear terms,

$$\Delta\mathbf{q}_L = \begin{bmatrix} 1 \\ \frac{1}{2}\Delta\boldsymbol{\theta}_L \end{bmatrix} + O(\|\Delta\boldsymbol{\theta}_L\|^2) \;, \qquad \Delta\mathbf{R}_L = \mathbf{I} + [\Delta\boldsymbol{\theta}_L]_\times + O(\|\Delta\boldsymbol{\theta}_L\|^2). \tag{125}$$

With this we can easily develop expressions for the time-derivatives. Just consider $\mathbf{q} = \mathbf{q}(t)$ as the original state, $\tilde{\mathbf{q}} = \mathbf{q}(t+\Delta t)$ as the perturbed state, and apply the definition of the derivative

$$\frac{d\mathbf{q}(t)}{dt} \triangleq \lim_{\Delta t \to 0} \frac{\mathbf{q}(t+\Delta t) - \mathbf{q}(t)}{\Delta t} \;, \tag{126}$$

to the above, with

$$\boldsymbol{\omega}_L(t) \triangleq \frac{d\boldsymbol{\theta}_L(t)}{dt} \triangleq \lim_{\Delta t \to 0} \frac{\Delta\boldsymbol{\theta}_L}{\Delta t} \;, \tag{127}$$

which, being $\Delta\boldsymbol{\theta}_L$ a local angular perturbation, corresponds to the angular rates vector in the local frame.

The development of the time-derivative of the quaternion follows (an analogous reasoning would be used for the rotation matrix)

$$\begin{aligned}
\dot{\mathbf{q}} &\triangleq \lim_{\Delta t \to 0} \frac{\mathbf{q}(t+\Delta t) - \mathbf{q}(t)}{\Delta t} \\
&= \lim_{\Delta t \to 0} \frac{\mathbf{q} \otimes \Delta\mathbf{q}_L - \mathbf{q}}{\Delta t} \\
&= \lim_{\Delta t \to 0} \frac{\mathbf{q} \otimes \left(\begin{bmatrix} 1 \\ \Delta\boldsymbol{\theta}_L/2 \end{bmatrix} - \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} \right)}{\Delta t} \\
&= \lim_{\Delta t \to 0} \frac{\mathbf{q} \otimes \begin{bmatrix} 0 \\ \Delta\boldsymbol{\theta}_L/2 \end{bmatrix}}{\Delta t} \\
&= \frac{1}{2}\,\mathbf{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_L \end{bmatrix} \;.
\end{aligned} \tag{128}$$

Defining

$$\mathbf{\Omega}(\boldsymbol{\omega}) \triangleq [\boldsymbol{\omega}]_R = \begin{bmatrix} 0 & -\boldsymbol{\omega}^\top \\ \boldsymbol{\omega} & -[\boldsymbol{\omega}]_\times \end{bmatrix} = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} , \tag{129}$$

we get from (128) and (17) (we give also its matrix equivalent)

$$\boxed{\dot{\mathbf{q}} = \frac{1}{2}\mathbf{\Omega}(\boldsymbol{\omega}_L)\,\mathbf{q} = \frac{1}{2}\mathbf{q} \otimes \boldsymbol{\omega}_L \ , \qquad \dot{\mathbf{R}} = \mathbf{R}\,[\boldsymbol{\omega}_L]_\times} \ . \tag{130}$$

1.7.2 Global perturbations

It is possible and indeed interesting to consider globally-defined perturbations, and likewise for the related derivatives. Global perturbations appear *at the left hand side* of the composition product, namely,

$$\tilde{\mathbf{q}} = \Delta\mathbf{q}_G \otimes \mathbf{q} \ , \qquad \tilde{\mathbf{R}} = \Delta\mathbf{R}_G\,\mathbf{R} \ . \tag{131}$$

The associated time-derivatives follow from a development analogous to (128), which results in

$$\boxed{\dot{\mathbf{q}} = \frac{1}{2}\,\boldsymbol{\omega}_G \otimes \mathbf{q} \ , \qquad \dot{\mathbf{R}} = [\boldsymbol{\omega}_G]_\times\,\mathbf{R}} \ , \tag{132}$$

where

$$\boldsymbol{\omega}_G(t) \triangleq \frac{d\boldsymbol{\theta}_G(t)}{dt} \tag{133}$$

is the angular rates vector expressed in the global frame.

1.7.3 Global-to-local relations

From the previous paragraph, it is worth noticing the following relation between local and global angular rates,

$$\frac{1}{2}\,\boldsymbol{\omega}_G \otimes \mathbf{q} = \dot{\mathbf{q}} = \frac{1}{2}\,\mathbf{q} \otimes \boldsymbol{\omega}_L \ . \tag{134}$$

Then, post-multiplying by the conjugate quaternion we have

$$\boldsymbol{\omega}_G = \mathbf{q} \otimes \boldsymbol{\omega}_L \otimes \mathbf{q}^* = \mathbf{R}\,\boldsymbol{\omega}_L \ . \tag{135}$$

Likewise, considering that $\Delta\boldsymbol{\theta}_R \approx \boldsymbol{\omega}\Delta t$ for small Δt, we have that

$$\Delta\boldsymbol{\theta}_G = \mathbf{q} \otimes \Delta\boldsymbol{\theta}_L \otimes \mathbf{q}^* = \mathbf{R}\,\Delta\boldsymbol{\theta}_L \ . \tag{136}$$

That is, we can transform angular rates vectors $\boldsymbol{\omega}$ and small angular perturbations $\Delta\boldsymbol{\theta}$ via frame transformation, using the quaternion or the rotation matrix, as if they were regular vectors. The same can be seen by posing $\boldsymbol{\omega} = \mathbf{u}\omega$, or $\Delta\boldsymbol{\theta} = \mathbf{u}\Delta\theta$, and noticing that the rotation axis vector $\mathbf{u}$ transforms normally, with

$$\mathbf{u}_G = \mathbf{q} \otimes \mathbf{u}_L \otimes \mathbf{q}^* = \mathbf{R}\,\mathbf{u}_L \ . \tag{137}$$

1.7.4 Time-derivative of the quaternion product

We use the regular formula for the derivative of the product,

$$(\dot{\mathbf{q}_1 \otimes \mathbf{q}_2}) = \dot{\mathbf{q}}_1 \otimes \mathbf{q}_2 + \mathbf{q}_1 \otimes \dot{\mathbf{q}}_2 \;, \qquad\qquad (\dot{\mathbf{R}_1 \mathbf{R}_2}) = \dot{\mathbf{R}}_1 \mathbf{R}_2 + \mathbf{R}_1 \dot{\mathbf{R}}_2 \;, \tag{138}$$

but noticing that, since the products are non commutative, we need to respect the order of the operands strictly. This means that $(\dot{\mathbf{q}^2}) \neq 2\,\mathbf{q} \otimes \dot{\mathbf{q}}$, as it would be in the scalar case, but rather

$$(\dot{\mathbf{q}^2}) = \dot{\mathbf{q}} \otimes \mathbf{q} + \mathbf{q} \otimes \dot{\mathbf{q}} \;. \tag{139}$$

1.7.5 Other useful expressions with the derivative

We can derive an expression for the local rotation rate

$$\boldsymbol{\omega}_L = 2\,\mathbf{q}^* \otimes \dot{\mathbf{q}} \;, \qquad\qquad [\boldsymbol{\omega}_L]_\times = \mathbf{R}^\top \dot{\mathbf{R}} \;. \tag{140}$$

and the global rotation rate,

$$\boldsymbol{\omega}_G = 2\,\dot{\mathbf{q}} \otimes \mathbf{q}^* \;, \qquad\qquad [\boldsymbol{\omega}_G]_\times = \dot{\mathbf{R}}\,\mathbf{R}^\top \;. \tag{141}$$

1.8 Time-integration of rotation rates

Accumulating rotation over time in quaternion form is done by integrating the differential equation appropriate to the rotation rate definition, that is, (130) for a local rotation rate definition, and (132) for a global one. In the cases we are interested in, the angular rates are measured by local sensors, thus providing local measurements $\boldsymbol{\omega}(t_n)$ at discrete times $t_n = n\Delta t$. We concentrate here on this case only, for which we reproduce the differential equation (130),

$$\dot{\mathbf{q}}(t) = \frac{1}{2}\mathbf{q}(t) \otimes \boldsymbol{\omega}(t) \;. \tag{142}$$

We develop zeroth- and first- order integration methods (Fig. 4), all based on the Taylor series of $\mathbf{q}(t_n + \Delta t)$ around the time $t = t_n$. We note $\mathbf{q} \triangleq \mathbf{q}(t)$ and $\mathbf{q}_n \triangleq \mathbf{q}(t_n)$, and the same for $\boldsymbol{\omega}$. The Taylor series reads,

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \dot{\mathbf{q}}_n \Delta t + \frac{1}{2!}\ddot{\mathbf{q}}_n \Delta t^2 + \frac{1}{3!}\dddot{\mathbf{q}}_n \Delta t^3 + \frac{1}{4!}\ddddot{\mathbf{q}}_n \Delta t^4 + \cdots \;. \tag{143}$$

The successive derivatives of $\mathbf{q}_n$ above are easily obtained by repeatedly applying the expression of the quaternion derivative, (142), with $\ddot{\boldsymbol{\omega}} = 0$. We obtain

$$\dot{\mathbf{q}}_n = \frac{1}{2}\,\mathbf{q}_n\,\boldsymbol{\omega}_n \tag{144a}$$

$$\ddot{\mathbf{q}}_n = \frac{1}{2^2}\,\mathbf{q}_n\,\boldsymbol{\omega}_n^2 + \frac{1}{2}\,\mathbf{q}_n\,\dot{\boldsymbol{\omega}} \tag{144b}$$

$$\dddot{\mathbf{q}}_n = \frac{1}{2^3}\,\mathbf{q}_n\,\boldsymbol{\omega}_n^3 + \frac{1}{4}\,\mathbf{q}_n\,\dot{\boldsymbol{\omega}}\,\boldsymbol{\omega}_n + \frac{1}{2}\,\mathbf{q}\,\boldsymbol{\omega}_n\,\dot{\boldsymbol{\omega}} \tag{144c}$$

$$\mathbf{q}_n^{(i \geq 4)} = \frac{1}{2^i}\,\mathbf{q}_n\,\boldsymbol{\omega}_n^i + \cdots \;, \tag{144d}$$

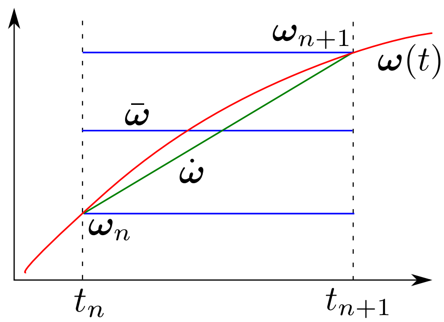

Figure 4: Angular velocity approximations for the integral: Red: true velocity. Blue: zero-th order approximations (bottom to top: forward, midward and backward). Green: first order approximation.

where we have omitted the $\otimes$ signs for economy of notation, that is, all products and the powers of $\boldsymbol{\omega}$ must be interpreted in terms of the quaternion product.

1.8.1 Zeroth order integration

Forward integration In the case where the angular rate $\boldsymbol{\omega}_n$ is held constant over the period $[t_n, t_{n+1}]$, we have $\dot{\boldsymbol{\omega}} = 0$ and the Taylor series above can be written as,

$$\mathbf{q}_{n+1} = \mathbf{q}_n \otimes \left(1 + \frac{1}{2}\boldsymbol{\omega}_n\Delta t + \frac{1}{2!}\left(\frac{1}{2}\boldsymbol{\omega}_n\Delta t\right)^2 + \frac{1}{3!}\left(\frac{1}{2}\boldsymbol{\omega}_n\Delta t\right)^3 + \frac{1}{4!}\left(\frac{1}{2}\boldsymbol{\omega}_n\Delta t\right)^4 + \cdots\right) , \tag{145}$$

in which we identify the Taylor series of the exponential $e^{\boldsymbol{\omega}_n\Delta t/2}$. From (77–78), this exponential corresponds to the quaternion representing the incremental rotation $\Delta\theta = \boldsymbol{\omega}_n\Delta t$,

$$e^{\boldsymbol{\omega}\Delta t/2} = \mathbf{q}\{\boldsymbol{\omega}\Delta t\} = \begin{bmatrix} \cos(\|\boldsymbol{\omega}\|\Delta t/2) \\ \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}\sin(\|\boldsymbol{\omega}\|\Delta t/2) \end{bmatrix} ,$$

therefore,

$$\boxed{\mathbf{q}_{n+1} \approx \mathbf{q}_n \otimes \mathbf{q}\{\boldsymbol{\omega}_n\Delta t\}} . \tag{146}$$

Backward integration We can also consider that the constant velocity over the period Δt corresponds to $\boldsymbol{\omega}_{n+1}$, the velocity measured at the end of the period. This can be developed in an similar manner with a Taylor expansion of $\mathbf{q}_n$ around t_{n+1}, leading to

$$\mathbf{q}_{n+1} \approx \mathbf{q}_n \otimes \mathbf{q}\{\boldsymbol{\omega}_{n+1}\Delta t\} . \tag{147}$$

We want to remark here that this is the typical integration method when arriving motion measurements are to be processed in real time, as the integration horizon corresponds to the last measurement (in this case, t_{n+1}). To make this more salient, we can re-label the time indices to use $\{n-1, n\}$ instead of $\{n, n+1\}$, and write,

$$\boxed{\mathbf{q}_n \approx \mathbf{q}_{n-1} \otimes \mathbf{q}\{\boldsymbol{\omega}_n\Delta t\}} . \tag{148}$$

Midward integration Similarly, if the velocity is considered constant at the median rate over the period Δt (which is not necessary the velocity at the midpoint of the period),

$$\overline{\boldsymbol{\omega}} = \frac{\boldsymbol{\omega}_{n+1} + \boldsymbol{\omega}_n}{2} \ , \tag{149}$$

we have,

$$\boxed{\mathbf{q}_{n+1} \approx \mathbf{q}_n \otimes \mathbf{q}\{\,\overline{\boldsymbol{\omega}}\Delta t\}} \ . \tag{150}$$

1.8.2 First order integration

The angular rate $\boldsymbol{\omega}(t)$ is now linear with time. Its first derivative is constant, and all higher ones are zero,

$$\dot{\boldsymbol{\omega}} = \frac{\boldsymbol{\omega}_{n+1} - \boldsymbol{\omega}_n}{\Delta t} \tag{151}$$

$$\ddot{\boldsymbol{\omega}} = \dddot{\boldsymbol{\omega}} = \cdots = 0 \ . \tag{152}$$

We can write the median rate $\overline{\boldsymbol{\omega}}$ in terms of $\boldsymbol{\omega}_n$ and $\dot{\boldsymbol{\omega}}$,

$$\overline{\boldsymbol{\omega}} = \boldsymbol{\omega}_n + \frac{1}{2}\dot{\boldsymbol{\omega}}\Delta t \ , \tag{153}$$

and derive the expression of the powers of $\boldsymbol{\omega}_n$ appearing in the quaternion derivatives (144), in terms of the more convenient $\overline{\boldsymbol{\omega}}$ and $\dot{\boldsymbol{\omega}}$,

$$\boldsymbol{\omega}_n = \overline{\boldsymbol{\omega}} - \frac{1}{2}\dot{\boldsymbol{\omega}}\Delta t \tag{154a}$$

$$\boldsymbol{\omega}_n^2 = \overline{\boldsymbol{\omega}}^2 - \frac{1}{2}\overline{\boldsymbol{\omega}}\,\dot{\boldsymbol{\omega}}\Delta t - \frac{1}{2}\dot{\boldsymbol{\omega}}\,\overline{\boldsymbol{\omega}}\Delta t + \frac{1}{4}\dot{\boldsymbol{\omega}}^2\Delta t^2 \tag{154b}$$

$$\boldsymbol{\omega}_n^3 = \overline{\boldsymbol{\omega}}^3 - \frac{3}{2}\overline{\boldsymbol{\omega}}^2\,\dot{\boldsymbol{\omega}}\Delta t + \frac{3}{4}\overline{\boldsymbol{\omega}}\,\dot{\boldsymbol{\omega}}^2\Delta t^2 + \frac{1}{8}\dot{\boldsymbol{\omega}}^3\Delta t^3 \tag{154c}$$

$$\boldsymbol{\omega}_n^4 = \overline{\boldsymbol{\omega}}^4 + \cdots \ . \tag{154d}$$

Injecting them in the quaternion derivatives, and substituting in the Taylor series (143), we have after proper reordering,

$$\mathbf{q}_{n+1} = \mathbf{q}\left(1 + \frac{1}{2}\overline{\boldsymbol{\omega}}\Delta t + \frac{1}{2!}\left(\frac{1}{2}\overline{\boldsymbol{\omega}}\Delta t\right)^2 + \frac{1}{3!}\left(\frac{1}{2}\overline{\boldsymbol{\omega}}\Delta t\right)^3 + \cdots\right) \tag{155a}$$

$$+ \ \mathbf{q}\left(-\frac{1}{4}\dot{\boldsymbol{\omega}} + \frac{1}{4}\dot{\boldsymbol{\omega}}\right)\Delta t^2 \tag{155b}$$

$$+ \ \mathbf{q}\left(-\frac{1}{16}\overline{\boldsymbol{\omega}}\,\dot{\boldsymbol{\omega}} - \frac{1}{16}\dot{\boldsymbol{\omega}}\,\overline{\boldsymbol{\omega}} + \frac{1}{24}\dot{\boldsymbol{\omega}}\,\overline{\boldsymbol{\omega}} + \frac{1}{12}\overline{\boldsymbol{\omega}}\,\dot{\boldsymbol{\omega}}\right)\Delta t^3 \tag{155c}$$

$$+ \ \mathbf{q}\left(\ \cdots\ \right)\Delta t^4 \ + \ \cdots \ , \tag{155d}$$

where in (155a) we recognize the Taylor series of the exponential $e^{\overline{\boldsymbol{\omega}}\Delta t/2}$, (155b) vanishes, and (155d) represents terms of high multiplicity that we are going to neglect. From (77–78), we have $e^{\overline{\boldsymbol{\omega}}\Delta t/2} = \mathbf{q}\{\overline{\boldsymbol{\omega}}\Delta t\}$, the quaternion representing the incremental rotation $\Delta\theta = \overline{\boldsymbol{\omega}}\Delta t$, now using the average rotation rate over the period Δt. This yields after simplification (we recover now the normal $\otimes$ notation),

$$\mathbf{q}_{n+1} = \mathbf{q}_n \otimes \mathbf{q}\{\overline{\boldsymbol{\omega}}\Delta t\} + \frac{\Delta t^3}{48}\mathbf{q}_n \otimes (\overline{\boldsymbol{\omega}} \otimes \dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}} \otimes \overline{\boldsymbol{\omega}}) + \cdots \; . \tag{156}$$

Substituting $\dot{\boldsymbol{\omega}}$ and $\overline{\boldsymbol{\omega}}$ by their definitions (151) and (149) we get,

$$\mathbf{q}_{n+1} = \mathbf{q}_n \otimes \mathbf{q}\{\overline{\boldsymbol{\omega}}\Delta t\} + \frac{\Delta t^2}{48}\mathbf{q}_n \otimes (\boldsymbol{\omega}_n \otimes \boldsymbol{\omega}_{n+1} - \boldsymbol{\omega}_{n+1} \otimes \boldsymbol{\omega}_n) + \cdots \; , \tag{157}$$

which is a result equivalent to (Trawny and Roumeliotis, 2005)'s, but using the Hamilton convention, and the quaternion product form instead of the matrix product form. Finally, since $\mathbf{a}_v \otimes \mathbf{b}_v - \mathbf{b}_v \otimes \mathbf{a}_v = 2\,\mathbf{a}_v \times \mathbf{b}_v$, see (34), we have the alternative form,

$$\boxed{\mathbf{q}_{n+1} \approx \mathbf{q}_n \otimes \mathbf{q}\{\overline{\boldsymbol{\omega}}\Delta t\} + \frac{\Delta t^2}{24}\mathbf{q}_n \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_n \times \boldsymbol{\omega}_{n+1} \end{bmatrix}} \; . \tag{158}$$

In this expression, the first term of the sum is the midward zeroth order integrator (150). The second term is a second-order correction that vanishes when $\boldsymbol{\omega}_n$ and $\boldsymbol{\omega}_{n+1}$ are collinear,[12] *i.e.*, when the axis of rotation has not changed from t_n to t_{n+1}. Given usual IMU sampling times $\Delta t \leq 0.01s$ and the usual near-collinearity of $\boldsymbol{\omega}_n$ and $\boldsymbol{\omega}_{n+1}$, this term takes values of the order of $10^{-6}\|\boldsymbol{\omega}\|^2$, or easily smaller. Terms with higher multiplicities of $\boldsymbol{\omega}\Delta t$ are even smaller and have been neglected.

Please note also that, while all zeroth-order integrators result in unit quaternions by construction (because they are computed as the product of two unit quaternions), this is not the case for the first-order integrator due to the sum in (158). Hence, when using the first-order integrator, and even if the summed term is small as stated, users should take care to check the evolution of the quaternion norm over time, and eventually re-normalize the quaternion if needed, using $\mathbf{q} \leftarrow \mathbf{q}/\|\mathbf{q}\|$.

1.9 Useful, and very useful, Jacobians of the rotation

Let us consider a rotation to a vector $\mathbf{a}$, of θ radians around the unit axis $\mathbf{u}$. Let us express the rotation specification in three equivalent forms, namely $\boldsymbol{\theta} = \theta\mathbf{u}$, $\mathbf{q} = \mathbf{q}\{\boldsymbol{\theta}\}$ and $\mathbf{R} = \mathbf{R}\{\boldsymbol{\theta}\}$. We are interested in the Jacobians of the rotated result with respect to different magnitudes.

[12]Notice also from (157) that this term would *always* vanish if the quaternion product were commutative, which is not.

1.9.1 Jacobian with respect to the vector

The derivative of the rotation with respect to the vector $\mathbf{a}$ is trivial,

$$\boxed{\frac{\partial(\mathbf{q}\otimes\mathbf{a}\otimes\mathbf{q}*)}{\partial\mathbf{a}} = \frac{\partial(\mathbf{R}\,\mathbf{a})}{\partial\mathbf{a}} = \mathbf{R}}\;. \tag{159}$$

1.9.2 Jacobian with respect to the rotation vector

The derivative of the rotation with respect to the rotation vector $\boldsymbol{\theta}$ is relatively easy. Usually, because we have a global specification $\boldsymbol{\theta}$, we must consider the infinitesimal variations $\partial\boldsymbol{\theta}$ in the global frame, having $\mathbf{R}\{\boldsymbol{\theta}+\partial\boldsymbol{\theta}\} \xrightarrow[\theta\to 0]{} (\mathbf{I}+[\partial\boldsymbol{\theta}]_\times)\mathbf{R}$. Therefore

$$\boxed{\frac{\partial(\mathbf{q}\otimes\mathbf{a}\otimes\mathbf{q}^*)}{\partial\boldsymbol{\theta}} = \frac{\partial(\mathbf{R}\,\mathbf{a})}{\partial\boldsymbol{\theta}} = \lim_{\partial\boldsymbol{\theta}\to 0}\frac{\big((\mathbf{I}+[\partial\boldsymbol{\theta}]_\times)\mathbf{R}\,\mathbf{a}-\mathbf{R}\,\mathbf{a}\big)}{\partial\boldsymbol{\theta}} = -\,[\mathbf{R}\,\mathbf{a}]_\times}\;. \tag{160}$$

1.9.3 Jacobian with respect to the quaternion

On the contrary, the derivative of the rotation with respect to the quaternion $\mathbf{q}$ is tricky. For convenience, we use a lighter notation for the quaternion, $\mathbf{q}=[w\;\;\mathbf{v}]=w+\mathbf{v}$. We make use of (33), (34), and the identity $\mathbf{a}\times(\mathbf{b}\times\mathbf{c})=(\mathbf{c}\times\mathbf{b})\times\mathbf{a}=(\mathbf{a}^\top\mathbf{c})\,\mathbf{b}-(\mathbf{a}^\top\mathbf{b})\,\mathbf{c}$, to develop the quaternion-based rotation (81) as follows,

$$\begin{aligned}
\mathbf{a}' &= \mathbf{q}\otimes\mathbf{a}\otimes\mathbf{q}*\\
&= (w+\mathbf{v})\otimes\mathbf{a}\otimes(w-\mathbf{v})\\
&= w^2\mathbf{a}+w(\mathbf{v}\otimes\mathbf{a}-\mathbf{a}\otimes\mathbf{v})-\mathbf{v}\otimes\mathbf{a}\otimes\mathbf{v}\\
&= w^2\mathbf{a}+2w(\mathbf{v}\times\mathbf{a})-\big[(-\mathbf{v}^\top\mathbf{a}+\mathbf{v}\times\mathbf{a})\otimes\mathbf{v}\big]\\
&= w^2\mathbf{a}+2w(\mathbf{v}\times\mathbf{a})-\big[(-\mathbf{v}^\top\mathbf{a})\,\mathbf{v}+(\mathbf{v}\times\mathbf{a})\otimes\mathbf{v}\big]\\
&= w^2\mathbf{a}+2w(\mathbf{v}\times\mathbf{a})-\big[(-\mathbf{v}^\top\mathbf{a})\,\mathbf{v}-\cancel{(\mathbf{v}\times\mathbf{a})^\top\mathbf{v}}+(\mathbf{v}\times\mathbf{a})\times\mathbf{v}\big]\\
&= w^2\mathbf{a}+2w(\mathbf{v}\times\mathbf{a})-\big[(-\mathbf{v}^\top\mathbf{a})\,\mathbf{v}+(\mathbf{v}^\top\mathbf{v})\,\mathbf{a}-(\mathbf{v}^\top\mathbf{a})\,\mathbf{v}\big]\\
&= w^2\mathbf{a}+2w(\mathbf{v}\times\mathbf{a})+2(\mathbf{v}^\top\mathbf{a})\,\mathbf{v}-(\mathbf{v}^\top\mathbf{v})\,\mathbf{a}\;.
\end{aligned} \tag{161}$$

With this, we can extract the derivatives $\partial\mathbf{a}'/\partial w$ and $\partial\mathbf{a}'/\partial\mathbf{v}$,

$$\frac{\partial\mathbf{a}'}{\partial w} = 2(w\mathbf{a}+\mathbf{v}\times\mathbf{a}) \tag{162}$$

$$\begin{aligned}
\frac{\partial\mathbf{a}'}{\partial\mathbf{v}} &= -2w\,[\mathbf{a}]_\times+2(\mathbf{v}^\top\mathbf{a}\,\mathbf{I}+\mathbf{v}\,\mathbf{a}^\top)-2\mathbf{a}\,\mathbf{v}^\top\\
&= 2(\mathbf{v}^\top\mathbf{a}\,\mathbf{I}+\mathbf{v}\,\mathbf{a}^\top-\mathbf{a}\,\mathbf{v}^\top-w\,[\mathbf{a}]_\times)\;,
\end{aligned} \tag{163}$$

yielding

$$\boxed{\frac{\partial(\mathbf{q}\otimes\mathbf{a}\otimes\mathbf{q}*)}{\partial\mathbf{q}} = \frac{\partial(\mathbf{R}\,\mathbf{a})}{\partial\mathbf{q}} = 2\left[\;w\mathbf{a}+\mathbf{v}\times\mathbf{a}\;\;\middle|\;\mathbf{v}^\top\mathbf{a}\,\mathbf{I}+\mathbf{v}\,\mathbf{a}^\top-\mathbf{a}\,\mathbf{v}^\top-w\,[\mathbf{a}]_\times\;\right]}\;. \tag{164}$$

2 Error-state kinematics for IMU-driven systems

2.1 Motivation

We wish to write the error-estate equations of the kinematics of an inertial system integrating accelerometer and gyrometer readings with bias and noise, using the Hamilton quaternion to represent the orientation in space or *attitude*.

Accelerometer and gyrometer readings come typically from an Inertial Measurement Unit (IMU). Integrating IMU readings leads to dead-reckoning positioning systems, which drift with time. Avoiding drift is a matter of fusing this information with absolute position readings such as GPS or vision.

The error-state Kalman filter (ESKF) is one of the tools we may use for this purpose. Within the Kalman filtering paradigm, these are the most remarkable assets of the ESKF (Madyastha et al., 2011):

- The orientation error-state is minimal (*i.e.*, it has the same number of parameters as degrees of freedom), avoiding issues related to over-parametrization (or redundancy) and the consequent risk of singularity of the involved covariances matrices, resulting typically from enforcing constraints.
- The error-state system is always operating close to the origin, and therefore far from possible parameter singularities, gimbal lock issues, or the like, providing a guarantee that the linearization validity holds at all times.
- The error-state is always small, meaning that all second-order products are negligible. This makes the computation of Jacobians very easy and fast. Some Jacobians may even be constant or equal to available state magnitudes.
- The error dynamics are slow because all the large-signal dynamics have been integrated in the nominal-state. This means that we can apply KF corrections (which are the only means to observe the errors) at a lower rate than the predictions.

2.2 The error-state Kalman filter explained

In error-state filter formulations, we speak of true-, nominal- and error-state values, the true-state being expressed as a suitable composition (linear sum, quaternion product or matrix product) of the nominal- and the error- states. The idea is to consider the nominal-state as large-signal (integrable in non-linear fashion) and the error-state as small signal (thus linearly integrable and suitable for linear-Gaussian filtering).

The error-state filter can be explained as follows. On one side, high-frequency IMU data $\mathbf{u}_m$ is integrated into a nominal-state $\mathbf{x}$. This nominal state does not take into account the noise terms $\mathbf{w}$ and other possible model imperfections. As a consequence, it will accumulate errors. These errors are collected in the error-state $\delta\mathbf{x}$ and estimated with the Error-State Kalman Filter (ESKF), this time incorporating all the noise and perturbations. The error-state consists of small-signal magnitudes, and its evolution function is correctly defined by a (time-variant) linear dynamic system, with its dynamic, control and measurement matrices computed from the values of the nominal-state. In parallel with integration of the nominal-state, the ESKF predicts a Gaussian estimate of the error-state. It only predicts, because by now no other measurement is available to correct these estimates. The filter correction is performed at the arrival of information other than IMU (*e.g.* GPS, vision, etc.), which is able to render the errors observable and which happens generally at a much lower rate than the integration phase. This correction provides a posterior Gaussian estimate of the error-state. After this, the error-state's mean is injected into the nominal-state, then reset to zero. The error-state's covariances matrix is conveniently updated to reflect this reset. The system goes on like this forever.

2.3 System kinematics in continuous time

The definition of all the involved variables is summarized in Table 2. Two important decisions regarding conventions are worth mentioning:

- The angular rates $\boldsymbol{\omega}$ are defined *locally* with respect to the nominal quaternion. This allows us to use the gyrometer measurements $\boldsymbol{\omega}_m$ directly, as they provide body-referenced angular rates.
- The angular error $\delta\boldsymbol{\theta}$ is also defined *locally* with respect to the nominal orientation. This is not necessarily the optimal way to proceed, but it corresponds to the choice in most IMU-integration works —what we could call the *classical approach*. There exists evidence (Li and Mourikis, 2012) that a globally-defined angular error has better properties. This will be explored too in the present document, Section 4, but most of the developments, examples and algorithms here are based in this locally-defined angular error.

2.3.1 The true-state kinematics

The true kinematic equations are

$$\dot{\mathbf{p}}_t = \mathbf{v}_t \tag{165a}$$
$$\dot{\mathbf{v}}_t = \mathbf{a}_t \tag{165b}$$
$$\dot{\mathbf{q}}_t = \frac{1}{2}\mathbf{q}_t \otimes \boldsymbol{\omega}_t \tag{165c}$$
$$\dot{\mathbf{a}}_{bt} = \mathbf{a}_w \tag{165d}$$
$$\dot{\boldsymbol{\omega}}_{bt} = \boldsymbol{\omega}_w \tag{165e}$$
$$\dot{\mathbf{g}}_t = 0 \tag{165f}$$

Table 2: All variables in the error-state Kalman filter.

Magnitude	True	Nominal	Error	Composition	Measured	Noise
Full state ([1])	$\mathbf{x}_t$	$\mathbf{x}$	$\delta\mathbf{x}$	$\mathbf{x}_t = \mathbf{x} \oplus \delta\mathbf{x}$		
Position	$\mathbf{p}_t$	$\mathbf{p}$	$\delta\mathbf{p}$	$\mathbf{p}_t = \mathbf{p} + \delta\mathbf{p}$		
Velocity	$\mathbf{v}_t$	$\mathbf{v}$	$\delta\mathbf{v}$	$\mathbf{v}_t = \mathbf{v} + \delta\mathbf{v}$		
Quaternion ([2,3])	$\mathbf{q}_t$	$\mathbf{q}$	$\delta\mathbf{q}$	$\mathbf{q}_t = \mathbf{q} \otimes \delta\mathbf{q}$		
Rotation matrix ([2,3])	$\mathbf{R}_t$	$\mathbf{R}$	$\delta\mathbf{R}$	$\mathbf{R}_t = \mathbf{R}\,\delta\mathbf{R}$		
Angles vector ([4])			$\delta\boldsymbol{\theta}$	$\delta\mathbf{q} = e^{\delta\boldsymbol{\theta}/2}$ $\delta\mathbf{R} = e^{[\delta\boldsymbol{\theta}]_\times}$		
Accelerometer bias	$\mathbf{a}_{bt}$	$\mathbf{a}_b$	$\delta\mathbf{a}_b$	$\mathbf{a}_{bt} = \mathbf{a}_b + \delta\mathbf{a}_b$		$\mathbf{a}_w$
Gyrometer bias	$\boldsymbol{\omega}_{bt}$	$\boldsymbol{\omega}_b$	$\delta\boldsymbol{\omega}_b$	$\boldsymbol{\omega}_{bt} = \boldsymbol{\omega}_b + \delta\boldsymbol{\omega}_b$		$\boldsymbol{\omega}_w$
Gravity vector	$\mathbf{g}_t$	$\mathbf{g}$	$\delta\mathbf{g}$	$\mathbf{g}_t = \mathbf{g} + \delta\mathbf{g}$		
Acceleration	$\mathbf{a}_t$				$\mathbf{a}_m$	$\mathbf{a}_n$
Angular rate	$\boldsymbol{\omega}_t$				$\boldsymbol{\omega}_m$	$\boldsymbol{\omega}_n$

([1]) the symbol $\oplus$ indicates a generic composition

([2]) indicates non-minimal representations

([3]) see Table 3 for the composition formula in case of globally-defined angular errors

([4]) exponentials defined as in (77, 78) and (68, 74)

Here, the true acceleration $\mathbf{a}_t$ and angular rate $\boldsymbol{\omega}_t$ are obtained from an IMU in the form of noisy sensor readings $\mathbf{a}_m$ and $\boldsymbol{\omega}_m$ in body frame, namely[13]

$$\mathbf{a}_m = \mathbf{R}_t^\top(\mathbf{a}_t - \mathbf{g}_t) + \mathbf{a}_{bt} + \mathbf{a}_n \tag{166}$$

$$\boldsymbol{\omega}_m = \boldsymbol{\omega}_t + \boldsymbol{\omega}_{bt} + \boldsymbol{\omega}_n \tag{167}$$

with $\mathbf{R}_t \triangleq \mathbf{R}(\mathbf{q}_t)$. With this, the true values can be isolated (this means that we have inverted the measurement equations),

$$\mathbf{a}_t = \mathbf{R}_t(\mathbf{a}_m - \mathbf{a}_{bt} - \mathbf{a}_n) + \mathbf{g}_t \tag{168}$$

$$\boldsymbol{\omega}_t = \boldsymbol{\omega}_m - \boldsymbol{\omega}_{bt} - \boldsymbol{\omega}_n. \tag{169}$$

[13] It is common practice to neglect the Earth's rotation rate $\boldsymbol{\omega}_\mathcal{E}$ in (167), which would otherwise be $\boldsymbol{\omega}_m = \boldsymbol{\omega}_t + \mathbf{R}_t^\top\boldsymbol{\omega}_\mathcal{E} + \boldsymbol{\omega}_{bt} + \boldsymbol{\omega}_n$. This is in most cases unjustifiably complicated.

Substituting above yields the kinematic system

$$\dot{\mathbf{p}}_t = \mathbf{v}_t \tag{170a}$$
$$\dot{\mathbf{v}}_t = \mathbf{R}_t(\mathbf{a}_m - \mathbf{a}_{bt} - \mathbf{a}_n) + \mathbf{g}_t \tag{170b}$$
$$\dot{\mathbf{q}}_t = \frac{1}{2}\mathbf{q}_t \otimes (\boldsymbol{\omega}_m - \boldsymbol{\omega}_{bt} - \boldsymbol{\omega}_n) \tag{170c}$$
$$\dot{\mathbf{a}}_{bt} = \mathbf{a}_w \tag{170d}$$
$$\dot{\boldsymbol{\omega}}_{bt} = \boldsymbol{\omega}_w \tag{170e}$$
$$\dot{\mathbf{g}}_t = 0 \tag{170f}$$

which we may name $\dot{\mathbf{x}}_t = f_t(\mathbf{x}_t, \mathbf{u}, \mathbf{w})$. This system has state $\mathbf{x}_t$, is governed by IMU noisy readings $\mathbf{u}_m$, and is perturbed by white Gaussian noise $\mathbf{w}$, all defined by

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{p}_t \\ \mathbf{v}_t \\ \mathbf{q}_t \\ \mathbf{a}_{bt} \\ \boldsymbol{\omega}_{bt} \\ \mathbf{g}_t \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} \mathbf{a}_m - \mathbf{a}_n \\ \boldsymbol{\omega}_m - \boldsymbol{\omega}_n \end{bmatrix} \qquad \mathbf{w} = \begin{bmatrix} \mathbf{a}_w \\ \boldsymbol{\omega}_w \end{bmatrix} . \tag{171}$$

It is to note in the above formulation that the gravity vector $\mathbf{g}_t$ is going to be estimated by the filter. It has a constant evolution equation, (170f), as corresponds to a magnitude that is known to be constant. The system starts at a fixed and arbitrarily known initial orientation $\mathbf{q}_t(t = 0) = \mathbf{q}_0$, which, being generally not in the horizontal plane, makes the initial gravity vector generally unknown. For simplicity it is usually taken $\mathbf{q}_0 = (1, 0, 0, 0)$ and thus $\mathbf{R}_0 = \mathbf{R}\{\mathbf{q}_0\} = \mathbf{I}$. We estimate $\mathbf{g}_t$ expressed in frame $\mathbf{q}_0$, and not $\mathbf{q}_t$ expressed in a horizontal frame, so that the initial uncertainty in orientation is transferred to an initial uncertainty on the gravity direction. We do so to improve linearity: indeed, equation (170b) is now linear in $\mathbf{g}$, which carries all the uncertainty, and the initial orientation $\mathbf{q}_0$ is known without uncertainty, so that $\mathbf{q}$ starts with no uncertainty. Once the gravity vector is estimated the horizontal plane can be recovered and, if desired, the whole state and recovered motion trajectories can be re-oriented to reflect the estimated horizontal. See (Lupton and Sukkarieh, 2009) for further justification. This is of course optional, and the reader is free to remove all equations related to graviy from the system and adopt a more classical approach of considering $\mathbf{g} \triangleq (0, 0, -9.8xx)$, with xx the appropriate decimal digits of the gravity vector on the site of the experiment, and an uncertain initial orientation $\mathbf{q}_0$.

2.3.2 The nominal-state kinematics

The nominal-state kinematics corresponds to the modeled system without noises or perturbations,

$$\dot{\mathbf{p}} = \mathbf{v} \tag{172a}$$
$$\dot{\mathbf{v}} = \mathbf{R}(\mathbf{a}_m - \mathbf{a}_b) + \mathbf{g} \tag{172b}$$
$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \otimes (\boldsymbol{\omega}_m - \boldsymbol{\omega}_b) \tag{172c}$$
$$\dot{\mathbf{a}}_b = 0 \tag{172d}$$
$$\dot{\boldsymbol{\omega}}_b = 0 \tag{172e}$$
$$\dot{\mathbf{g}} = 0. \tag{172f}$$

2.3.3 The error-state kinematics

The goal is to determine the linearized dynamics of the error-state. For each state equation, we write its composition (in Table 2), solving for the error state and simplifying all second-order infinitesimals. We give here the full error-state dynamic system and proceed afterwards with comments and proofs.

$$\dot{\delta\mathbf{p}} = \delta\mathbf{v} \tag{173a}$$
$$\dot{\delta\mathbf{v}} = -\mathbf{R}\left[\mathbf{a}_m - \mathbf{a}_b\right]_\times \delta\boldsymbol{\theta} - \mathbf{R}\delta\mathbf{a}_b + \delta\mathbf{g} - \mathbf{R}\mathbf{a}_n \tag{173b}$$
$$\dot{\delta\boldsymbol{\theta}} = -\left[\boldsymbol{\omega}_m - \boldsymbol{\omega}_b\right]_\times \delta\boldsymbol{\theta} - \delta\boldsymbol{\omega}_b - \boldsymbol{\omega}_n \tag{173c}$$
$$\dot{\delta\mathbf{a}}_b = \mathbf{a}_w \tag{173d}$$
$$\dot{\delta\boldsymbol{\omega}}_b = \boldsymbol{\omega}_w \tag{173e}$$
$$\dot{\delta\mathbf{g}} = 0. \tag{173f}$$

Equations (173a), (173d), (173e) and (173f), respectively of position, both biases, and gravity errors, are derived from linear equations and their error-state dynamics is trivial. As an example, consider the true and nominal position equations (170a) and (172a), their composition $\mathbf{p}_t = \mathbf{p} + \delta\mathbf{p}$ from Table 2, and solve for $\dot{\delta\mathbf{p}}$ to obtain (173a).

Equations (173b) and (173c), of velocity and orientation errors, require some non-trivial manipulations of the non-linear equations (170b) and (170c) to obtain the linearized dynamics. Their proofs are developed in the following two sections.

Equation (173b): The linear velocity error. We wish to determine $\dot{\delta\mathbf{v}}$, the dynamics of the velocity errors. We start with the following relations

$$\mathbf{R}_t = \mathbf{R}(\mathbf{I} + [\delta\boldsymbol{\theta}]_\times) + O(\|\delta\boldsymbol{\theta}\|^2) \tag{174}$$
$$\dot{\mathbf{v}} = \mathbf{R}\mathbf{a}_\mathcal{B} + \mathbf{g}, \tag{175}$$

where (174) is the small-signal approximation of $\mathbf{R}_t$, and in (175) we rewrote (172b) but introducing $\mathbf{a}_\mathcal{B}$ and $\delta\mathbf{a}_\mathcal{B}$, defined as the large- and small-signal accelerations in body frame,

$$\mathbf{a}_\mathcal{B} \triangleq \mathbf{a}_m - \mathbf{a}_b \tag{176}$$

$$\delta\mathbf{a}_\mathcal{B} \triangleq -\delta\mathbf{a}_b - \mathbf{a}_n \tag{177}$$

so that we can write the true acceleration in inertial frame as a composition of large- and small-signal terms,

$$\mathbf{a}_t = \mathbf{R}_t(\mathbf{a}_\mathcal{B} + \delta\mathbf{a}_\mathcal{B}) + \mathbf{g} + \delta\mathbf{g}. \tag{178}$$

We proceed by writing the expression (170b) of $\dot{\mathbf{v}}_t$ in two different forms (left and right developments), where the terms $O(\|\delta\boldsymbol{\theta}\|^2)$ have been ignored,

$$\begin{aligned}\dot{\mathbf{v}} + \dot{\delta\mathbf{v}} = \boxed{\dot{\mathbf{v}}_t} &= \mathbf{R}(\mathbf{I} + [\delta\boldsymbol{\theta}]_\times)(\mathbf{a}_\mathcal{B} + \delta\mathbf{a}_\mathcal{B}) + \mathbf{g} + \delta\mathbf{g} \\ \mathbf{R}\mathbf{a}_\mathcal{B} + \mathbf{g} + \dot{\delta\mathbf{v}} = \quad &= \mathbf{R}\mathbf{a}_\mathcal{B} + \mathbf{R}\delta\mathbf{a}_\mathcal{B} + \mathbf{R}\,[\delta\boldsymbol{\theta}]_\times\,\mathbf{a}_\mathcal{B} + \mathbf{R}\,[\delta\boldsymbol{\theta}]_\times\,\delta\mathbf{a}_\mathcal{B} + \mathbf{g} + \delta\mathbf{g}\end{aligned}$$

This leads after removing $\mathbf{R}\mathbf{a}_\mathcal{B} + \mathbf{g}$ from left and right to

$$\dot{\delta\mathbf{v}} = \mathbf{R}(\delta\mathbf{a}_\mathcal{B} + [\delta\boldsymbol{\theta}]_\times\,\mathbf{a}_\mathcal{B}) + \mathbf{R}\,[\delta\boldsymbol{\theta}]_\times\,\delta\mathbf{a}_\mathcal{B} + \delta\mathbf{g} \tag{179}$$

Eliminating the second order terms and reorganizing some cross-products (with $[\mathbf{a}]_\times\,\mathbf{b} = -\,[\mathbf{b}]_\times\,\mathbf{a}$), we get

$$\dot{\delta\mathbf{v}} = \mathbf{R}(\delta\mathbf{a}_\mathcal{B} - [\mathbf{a}_\mathcal{B}]_\times\,\delta\boldsymbol{\theta}) + \delta\mathbf{g}, \tag{180}$$

then, recalling (176) and (177),

$$\dot{\delta\mathbf{v}} = \mathbf{R}(-\,[\mathbf{a}_m - \mathbf{a}_b]_\times\,\delta\boldsymbol{\theta} - \delta\mathbf{a}_b - \mathbf{a}_n) + \delta\mathbf{g} \tag{181}$$

which after proper rearranging leads to the dynamics of the linear velocity error,

$$\boxed{\dot{\delta\mathbf{v}} = -\mathbf{R}\,[\mathbf{a}_m - \mathbf{a}_b]_\times\,\delta\boldsymbol{\theta} - \mathbf{R}\delta\mathbf{a}_b + \delta\mathbf{g} - \mathbf{R}\mathbf{a}_n}\;. \tag{182}$$

To further clean up this expression, we can often times assume that the accelerometer noise is white, uncorrelated and isotropic[14],

$$\mathbb{E}[\mathbf{a}_n] = 0 \qquad\qquad \mathbb{E}[\mathbf{a}_n\mathbf{a}_n^\top] = \sigma_a^2\mathbf{I}, \tag{183}$$

that is, the covariance ellipsoid is a sphere centered at the origin, which means that its mean and covariances matrix are invariant upon rotations (*Proof:* $\mathbb{E}[\mathbf{R}\mathbf{a}_n] = \mathbf{R}\mathbb{E}[\mathbf{a}_n] = 0$ and $\mathbf{E}[(\mathbf{R}\mathbf{a}_n)(\mathbf{R}\mathbf{a}_n)^\top] = \mathbf{R}\mathbb{E}[\mathbf{a}_n\mathbf{a}_n^\top]\mathbf{R}^\top = \mathbf{R}\sigma_a^2\mathbf{I}\mathbf{R}^\top = \sigma_a^2\mathbf{I}$). Then we can redefine the accelerometer noise vector, with absolutely no consequences, according to

$$\mathbf{a}_n \leftarrow \mathbf{R}\mathbf{a}_n \tag{184}$$

which gives

$$\boxed{\dot{\delta\mathbf{v}} = -\mathbf{R}\,[\mathbf{a}_m - \mathbf{a}_b]_\times\,\delta\boldsymbol{\theta} - \mathbf{R}\delta\mathbf{a}_b + \delta\mathbf{g} - \mathbf{a}_n}\;. \tag{185}$$

[14]This assumption cannot be made in cases where the three XYZ accelerometers are not identical.

Equation (173c): The orientation error. We wish to determine $\dot{\delta\boldsymbol{\theta}}$, the dynamics of the angular errors. We start with the following relations

$$\dot{\mathbf{q}}_t = \frac{1}{2}\mathbf{q}_t \otimes \boldsymbol{\omega}_t \tag{186}$$

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \otimes \boldsymbol{\omega}, \tag{187}$$

which are the true- and nominal- definitions of the quaternion derivatives.

As we did with the acceleration, we group large- and small-signal terms in the angular rate for clarity,

$$\boldsymbol{\omega} \triangleq \boldsymbol{\omega}_m - \boldsymbol{\omega}_b \tag{188}$$

$$\delta\boldsymbol{\omega} \triangleq -\delta\boldsymbol{\omega}_b - \boldsymbol{\omega}_n, \tag{189}$$

so that $\boldsymbol{\omega}_t$ can be written with a nominal part and an error part,

$$\boldsymbol{\omega}_t = \boldsymbol{\omega} + \delta\boldsymbol{\omega}. \tag{190}$$

We proceed by computing $\dot{\mathbf{q}}_t$ by two different means (left and right developments)

$$\begin{aligned}
(\mathbf{q} \dot{\otimes} \delta\mathbf{q}) = &\boxed{\ \dot{\mathbf{q}}_t\ } = \frac{1}{2}\mathbf{q}_t \otimes \boldsymbol{\omega}_t \\
\dot{\mathbf{q}} \otimes \delta\mathbf{q} + \mathbf{q} \otimes \dot{\delta\mathbf{q}} = & \qquad = \frac{1}{2}\mathbf{q} \otimes \delta\mathbf{q} \otimes \boldsymbol{\omega}_t \\
\frac{1}{2}\mathbf{q} \otimes \boldsymbol{\omega} \otimes \delta\mathbf{q} + \mathbf{q} \otimes \dot{\delta\mathbf{q}} = &
\end{aligned}$$

simplifying the leading $\mathbf{q}$ and isolating $\dot{\delta\mathbf{q}}$ we obtain

$$\begin{aligned}
\begin{bmatrix} 0 \\ \dot{\delta\boldsymbol{\theta}} \end{bmatrix} = \boxed{2\dot{\delta\mathbf{q}}} &= \delta\mathbf{q} \otimes \boldsymbol{\omega}_t - \boldsymbol{\omega} \otimes \delta\mathbf{q} \\
&= [\mathbf{q}]_R(\boldsymbol{\omega}_t)\delta\mathbf{q} - [\mathbf{q}]_L(\boldsymbol{\omega})\delta\mathbf{q} \\
&= \begin{bmatrix} 0 & -(\boldsymbol{\omega}_t - \boldsymbol{\omega})^\top \\ (\boldsymbol{\omega}_t - \boldsymbol{\omega}) & -\left[\boldsymbol{\omega}_t + \boldsymbol{\omega}\right]_\times \end{bmatrix} \begin{bmatrix} 1 \\ \delta\boldsymbol{\theta}/2 \end{bmatrix} + O(\|\delta\boldsymbol{\theta}\|^2) \\
&= \begin{bmatrix} 0 & -\delta\boldsymbol{\omega}^\top \\ \delta\boldsymbol{\omega} & -\left[2\boldsymbol{\omega} + \delta\boldsymbol{\omega}\right]_\times \end{bmatrix} \begin{bmatrix} 1 \\ \delta\boldsymbol{\theta}/2 \end{bmatrix} + O(\|\delta\boldsymbol{\theta}\|^2)
\end{aligned} \tag{191}$$

which results in one scalar- and one vector- equalities

$$0 = \delta\boldsymbol{\omega}^\top \delta\boldsymbol{\theta} + O(|\delta\boldsymbol{\theta}|^2) \tag{192a}$$

$$\dot{\delta\boldsymbol{\theta}} = \delta\boldsymbol{\omega} - \left[\boldsymbol{\omega}\right]_\times \delta\boldsymbol{\theta} - \frac{1}{2}\left[\delta\boldsymbol{\omega}\right]_\times \delta\boldsymbol{\theta} + O(\|\delta\boldsymbol{\theta}\|^2). \tag{192b}$$

The first equation leads to $\delta\boldsymbol{\omega}^\top\delta\boldsymbol{\theta} = O(\|\delta\boldsymbol{\theta}\|^2)$, which is formed by second-order infinitesimals, not very useful. The second equation yields, after neglecting all second-order terms,

$$\dot{\delta\boldsymbol{\theta}} = -\left[\boldsymbol{\omega}\right]_\times \delta\boldsymbol{\theta} + \delta\boldsymbol{\omega} \tag{193}$$

and finally, recalling (188) and (189), we get the linearized dynamics of the angular error,

$$\boxed{\delta\dot{\boldsymbol{\theta}} = -\left[\boldsymbol{\omega}_m - \boldsymbol{\omega}_b\right]_\times \delta\boldsymbol{\theta} - \delta\boldsymbol{\omega}_b - \boldsymbol{\omega}_n} \,. \tag{194}$$

2.4 System kinematics in discrete time

The differential equations above need to be integrated into differences equations to account for discrete time intervals $\Delta t > 0$. The integration methods may vary. In some cases, one will be able to use exact closed-form solutions. In other cases, numerical integration of varying degree of accuracy may be employed. Please refer to the Appendices for pertinent details on integration methods.

Integration needs to be done for the following sub-systems:

1. The nominal state.
2. The error-state.
   (a) The deterministic part: state dynamics and control.
   (b) The stochastic part: noise and perturbations.

2.4.1 The nominal state kinematics

We can write the differences equations of the nominal-state as

$$\begin{aligned}
\mathbf{p} &\leftarrow \mathbf{p} + \mathbf{v}\,\Delta t + \frac{1}{2}(\mathbf{R}(\mathbf{a}_m - \mathbf{a}_b) + \mathbf{g})\,\Delta t^2 && \text{(195a)}\\
\mathbf{v} &\leftarrow \mathbf{v} + (\mathbf{R}(\mathbf{a}_m - \mathbf{a}_b) + \mathbf{g})\,\Delta t && \text{(195b)}\\
\mathbf{q} &\leftarrow \mathbf{q} \otimes \mathbf{q}\{(\boldsymbol{\omega}_m - \boldsymbol{\omega}_b)\,\Delta t\} && \text{(195c)}\\
\mathbf{a}_b &\leftarrow \mathbf{a}_b && \text{(195d)}\\
\boldsymbol{\omega}_b &\leftarrow \boldsymbol{\omega}_b && \text{(195e)}\\
\mathbf{g} &\leftarrow \mathbf{g}\;, && \text{(195f)}
\end{aligned}$$

where $x \leftarrow f(x, \bullet)$ stands for a time update of the type $x_{k+1} = f(x_k, \bullet_k)$, $\mathbf{R} \triangleq \mathbf{R}\{\mathbf{q}\}$ is the rotation matrix associated to the current nominal orientation $\mathbf{q}$, and $\mathbf{q}\{v\}$ is the quaternion associated to the rotation v, according to (78).

We can also use more precise integration, please see the Appendices for more information.

2.4.2 The error-state kinematics

The deterministic part is integrated normally (in this case we follow the methods in App. C.2), and the integration of the stochastic part results in random impulses (see

App. E), thus,

$$\delta\mathbf{p} \leftarrow \delta\mathbf{p} + \delta\mathbf{v}\,\Delta t \tag{196a}$$
$$\delta\mathbf{v} \leftarrow \delta\mathbf{v} + (-\mathbf{R}\,[\mathbf{a}_m - \mathbf{a}_b]_\times\,\delta\boldsymbol{\theta} - \mathbf{R}\delta\mathbf{a}_b + \delta\mathbf{g})\Delta t + \mathbf{v_i} \tag{196b}$$
$$\delta\boldsymbol{\theta} \leftarrow \mathbf{R}^\top\{(\boldsymbol{\omega}_m - \boldsymbol{\omega}_b)\Delta t\}\delta\boldsymbol{\theta} - \delta\boldsymbol{\omega}_b\Delta t + \boldsymbol{\theta}_\mathbf{i} \tag{196c}$$
$$\delta\mathbf{a}_b \leftarrow \delta\mathbf{a}_b + \mathbf{a_i} \tag{196d}$$
$$\delta\boldsymbol{\omega}_b \leftarrow \delta\boldsymbol{\omega}_b + \boldsymbol{\omega}_\mathbf{i} \tag{196e}$$
$$\delta\mathbf{g} \leftarrow \delta\mathbf{g}\ . \tag{196f}$$

Here, $\mathbf{v_i}$, $\boldsymbol{\theta}_\mathbf{i}$, $\mathbf{a_i}$ and $\boldsymbol{\omega}_\mathbf{i}$ are the random impulses applied to the velocity, orientation and bias estimates, modeled by white Gaussian processes. Their mean is zero, and their covariances matrices are obtained by integrating the covariances of $\mathbf{a}_n$, $\boldsymbol{\omega}_n$, $\mathbf{a}_w$ and $\boldsymbol{\omega}_w$ over the step time Δt (see App. E),

$$\mathbf{V_i} = \sigma^2_{\tilde{\mathbf{a}}_n}\Delta t^2\mathbf{I} \qquad [m^2/s^2] \tag{197}$$
$$\Theta_\mathbf{i} = \sigma^2_{\tilde{\boldsymbol{\omega}}_n}\Delta t^2\mathbf{I} \qquad [rad^2] \tag{198}$$
$$\mathbf{A_i} = \sigma^2_{\mathbf{a}_w}\Delta t\mathbf{I} \qquad [m^2/s^4] \tag{199}$$
$$\boldsymbol{\Omega}_\mathbf{i} = \sigma^2_{\boldsymbol{\omega}_w}\Delta t\mathbf{I} \qquad [rad^2/s^2] \tag{200}$$

where $\sigma_{\tilde{\mathbf{a}}_n}[m/s^2]$, $\sigma_{\tilde{\boldsymbol{\omega}}_n}[rad/s]$, $\sigma_{\mathbf{a}_w}[m/s^2\sqrt{s}]$ and $\sigma_{\boldsymbol{\omega}_w}[rad/s\sqrt{s}]$ are to be determined from the information in the IMU datasheet, or from experimental measurements.

2.4.3 The error-state Jacobian and perturbation matrices

The Jacobians are obtained by simple inspection of the error-state differences equations in the previous section.

To write these equations in compact form, we consider the nominal state vector $\mathbf{x}$, the error state vector $\delta\mathbf{x}$, the input vector $\mathbf{u}_m$, and the perturbation impulses vector $\mathbf{i}$, as follows (see App. E.1 for details and justifications),

$$\mathbf{x} = \begin{bmatrix}\mathbf{p}\\ \mathbf{v}\\ \mathbf{q}\\ \mathbf{a}_b\\ \boldsymbol{\omega}_b\\ \mathbf{g}\end{bmatrix} \quad , \quad \delta\mathbf{x} = \begin{bmatrix}\delta\mathbf{p}\\ \delta\mathbf{v}\\ \delta\boldsymbol{\theta}\\ \delta\mathbf{a}_b\\ \delta\boldsymbol{\omega}_b\\ \delta\mathbf{g}\end{bmatrix} \quad , \quad \mathbf{u}_m = \begin{bmatrix}\mathbf{a}_m\\ \boldsymbol{\omega}_m\end{bmatrix} \quad , \quad \mathbf{i} = \begin{bmatrix}\mathbf{v_i}\\ \boldsymbol{\theta}_\mathbf{i}\\ \mathbf{a_i}\\ \boldsymbol{\omega}_\mathbf{i}\end{bmatrix} \tag{201}$$

The error-state system is now

$$\delta\mathbf{x} \leftarrow f(\mathbf{x}, \delta\mathbf{x}, \mathbf{u}_m, \mathbf{i}) = \mathbf{F_x}(\mathbf{x}, \mathbf{u}_m)\cdot\delta\mathbf{x} + \mathbf{F_i}\cdot\mathbf{i}, \tag{202}$$

The ESKF prediction equations are written:

$$\hat{\delta\mathbf{x}} \leftarrow \mathbf{F_x}(\mathbf{x}, \mathbf{u}_m)\cdot\hat{\delta\mathbf{x}} \tag{203}$$
$$\mathbf{P} \leftarrow \mathbf{F_x}\,\mathbf{P}\,\mathbf{F_x}^\top + \mathbf{F_i}\,\mathbf{Q_i}\,\mathbf{F_i}^\top\ , \tag{204}$$

where $\delta\mathbf{x} \sim \mathcal{N}\{\hat{\delta\mathbf{x}}, \mathbf{P}\}$[15]; $\mathbf{F_x}$ and $\mathbf{F_i}$ are the Jacobians of $f()$ with respect to the error and perturbation vectors; and $\mathbf{Q_i}$ is the covariances matrix of the perturbation impulses.

The expressions of the Jacobian and covariances matrices above are detailed below. All state-related values appearing herein are extracted directly from the nominal state.

$$\mathbf{F_x} = \left.\frac{\partial f}{\partial \delta\mathbf{x}}\right|_{\mathbf{x},\mathbf{u}_m} = \begin{bmatrix} \mathbf{I} & \mathbf{I}\Delta t & 0 & 0 & 0 & 0 \\ 0 & \mathbf{I} & -\mathbf{R}\left[\mathbf{a}_m - \mathbf{a}_b\right]_\times \Delta t & -\mathbf{R}\Delta t & 0 & \mathbf{I}\Delta t \\ 0 & 0 & \mathbf{R}^\top\{(\boldsymbol{\omega}_m - \boldsymbol{\omega}_b)\Delta t\} & 0 & -\mathbf{I}\Delta t & 0 \\ 0 & 0 & 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{I} \end{bmatrix} \tag{205}$$

$$\mathbf{F_i} = \left.\frac{\partial f}{\partial \mathbf{i}}\right|_{\mathbf{x},\mathbf{u}_m} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \mathbf{I} & 0 & 0 & 0 \\ 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \mathbf{I} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad , \quad \mathbf{Q_i} = \begin{bmatrix} \mathbf{V_i} & 0 & 0 & 0 \\ 0 & \boldsymbol{\Theta}_\mathbf{i} & 0 & 0 \\ 0 & 0 & \mathbf{A_i} & 0 \\ 0 & 0 & 0 & \boldsymbol{\Omega}_\mathbf{i} \end{bmatrix} . \tag{206}$$

Please note particularly that $\mathbf{F_x}$ is the system's transition matrix, which can be approximated to different levels of precision in a number of ways. We showed here one of its simplest forms (the Euler form). Se Appendices B to D for further reference.

Please note also that, being the mean of the error $\delta\mathbf{x}$ initialized to zero, the linear equation (203) always returns zero. You should of course skip line (203) in your code. I recommend that you write it, though, but that you comment it out so that you are sure you did not forget anything.

And please note, finally, that you should NOT skip the covariance prediction (204)!! In effect, the term $\mathbf{F_i}\,\mathbf{Q_i}\,\mathbf{F_i}^\top$ is not null and therefore this covariance grows continuously – as it must be in any prediction step.

3 Fusing IMU with complementary sensory data

At the arrival of other kind of information than IMU, such as GPS or vision, we proceed to correct the ESKF. In a well-designed system, this should render the IMU biases observable and allow the ESKF to correctly estimate them. There are a myriad of possibilities, the most popular ones being GPS + IMU, monocular vision + IMU, and stereo vision + IMU. In recent years, the combination of visual sensors with IMU has attracted a lot of attention, and thus generated a lot of scientific activity. These vision + IMU setups are very interesting for use in GPS-denied environments, and can be implemented on mobile devices (typically smart phones), but also on UAVs and other small, agile platforms.

[15] $x \sim \mathcal{N}\{\mu, \Sigma\}$ means that x is a Gaussian random variable with mean and covariances matrix specified by μ and Σ.

While the IMU information has served so far to make predictions to the ESKF, this other information is used to correct the filter. The correction consists of three steps:

1. observation of the error-state via filter correction,
2. injection of the observed errors into the nominal state, and
3. reset of the error-state.

These steps are developed in the following sections.

3.1 Observation of the error state via filter correction

Suppose as usual that we have a sensor that delivers information that depends on the state, such as

$$\mathbf{y} = h(\mathbf{x}_t) + v \ , \tag{207}$$

where $h()$ is a general nonlinear function of the system state (the true state), and v is a white Gaussian noise with covariance $\mathbf{V}$,

$$v \sim \mathcal{N}\{0, \mathbf{V}\} \ . \tag{208}$$

Our filter is estimating the error state, and therefore the filter correction equations[16],

$$\mathbf{K} = \mathbf{P}\mathbf{H}^\top(\mathbf{H}\mathbf{P}\mathbf{H}^\top + \mathbf{V})^{-1} \tag{209}$$

$$\hat{\delta\mathbf{x}} \leftarrow \mathbf{K}(\mathbf{y} - h(\hat{\mathbf{x}}_t)) \tag{210}$$

$$\mathbf{P} \leftarrow (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P} \tag{211}$$

require the Jacobian matrix $\mathbf{H}$ to be defined with respect to the error state $\delta\mathbf{x}$, and evaluated at the best true-state estimate $\hat{\mathbf{x}}_t = \mathbf{x} \oplus \hat{\delta\mathbf{x}}$. As the error state mean is zero at this stage (we have not observed it yet), we have $\hat{\mathbf{x}}_t = \mathbf{x}$ and we can use the nominal error $\mathbf{x}$ as the evaluation point, leading to

$$\mathbf{H} \equiv \left.\frac{\partial h}{\partial \delta\mathbf{x}}\right|_{\mathbf{x}} \ . \tag{212}$$

3.1.1 Jacobian computation for the filter correction

The Jacobian above might be computed in a number of ways. The most illustrative one is by making use of the chain rule,

$$\mathbf{H} \triangleq \left.\frac{\partial h}{\partial \delta\mathbf{x}}\right|_{\mathbf{x}} = \left.\frac{\partial h}{\partial \mathbf{x}_t}\right|_{\mathbf{x}} \left.\frac{\partial \mathbf{x}_t}{\partial \delta\mathbf{x}}\right|_{\mathbf{x}} = \mathbf{H_x}\ \mathbf{X}_{\delta\mathbf{x}} \ . \tag{213}$$

[16] We give the simplest form of the covariance update, $\mathbf{P} \leftarrow (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}$. This form is known to have poor numerical stability, as its outcome is not guaranteed to be symmetric nor positive definite. The reader is free to use more stable forms such as 1) the symmetric form $\mathbf{P} \leftarrow \mathbf{P} - \mathbf{K}(\mathbf{H}\mathbf{P}\mathbf{H}^\top + \mathbf{V})\mathbf{K}^\top$ and 2) the symmetric and positive *Joseph* form $\mathbf{P} \leftarrow (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}(\mathbf{I} - \mathbf{K}\mathbf{H})^\top + \mathbf{K}\mathbf{V}\mathbf{K}^\top$.

Here, $\mathbf{H_x} \triangleq \left.\frac{\partial h}{\partial \mathbf{x}_t}\right|_{\mathbf{x}}$ is the standard Jacobian of $h()$ with respect to its own argument (*i.e.*, the Jacobian one would use in a regular EKF). This first part of the chain rule depends on the measurement function of the particular sensor used, and is not presented here.

The second part, $\mathbf{X}_{\delta\mathbf{x}} \triangleq \left.\frac{\partial \mathbf{x}_t}{\partial \delta\mathbf{x}}\right|_{\mathbf{x}}$, is the Jacobian of the true state with respect to the error state. This part can be derived here as it only depends on the ESKF composition of states. We have the derivatives,

$$\mathbf{X}_{\delta\mathbf{x}} = \begin{bmatrix} \frac{\partial(\mathbf{p}+\delta\mathbf{p})}{\partial\delta\mathbf{p}} & & & & & \\ & \frac{\partial(\mathbf{v}+\delta\mathbf{v})}{\partial\delta\mathbf{v}} & & & 0 & \\ & & \frac{\partial(\mathbf{q}\otimes\delta\mathbf{q})}{\partial\delta\boldsymbol{\theta}} & & & \\ & & & \frac{\partial(\mathbf{a}_b+\delta\mathbf{a}_b)}{\partial\delta\mathbf{a}_b} & & \\ & 0 & & & \frac{\partial(\boldsymbol{\omega}_b+\delta\boldsymbol{\omega}_b)}{\partial\delta\boldsymbol{\omega}_b} & \\ & & & & & \frac{\partial(\mathbf{g}+\delta\mathbf{g})}{\partial\delta\mathbf{g}} \end{bmatrix} \tag{214}$$

which results in all identity 3×3 blocks (for example, $\frac{\partial(\mathbf{p}+\delta\mathbf{p})}{\partial\delta\mathbf{p}} = \mathbf{I}_3$) except for the 4×3 quaternion term $\mathbf{Q}_{\delta\boldsymbol{\theta}} = \partial(\mathbf{q}\otimes\delta\mathbf{q})/\partial\delta\boldsymbol{\theta}$. Therefore we have the form,

$$\mathbf{X}_{\delta\mathbf{x}} \triangleq \left.\frac{\partial \mathbf{x}_t}{\partial \delta\mathbf{x}}\right|_{\mathbf{x}} = \begin{bmatrix} \mathbf{I}_6 & 0 & 0 \\ 0 & \mathbf{Q}_{\delta\boldsymbol{\theta}} & 0 \\ 0 & 0 & \mathbf{I}_9 \end{bmatrix} \tag{215}$$

Using the chain rule, equations (17–19), and the limit $\delta\mathbf{q} \xrightarrow[\delta\boldsymbol{\theta}\to 0]{} \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta} \end{bmatrix}$, the quaternion term $\mathbf{Q}_{\delta\boldsymbol{\theta}}$ may be derived as follows,

$$\begin{aligned} \mathbf{Q}_{\delta\boldsymbol{\theta}} \triangleq \left.\frac{\partial(\mathbf{q}\otimes\delta\mathbf{q})}{\partial\delta\boldsymbol{\theta}}\right|_{\mathbf{q}} &= \left.\frac{\partial(\mathbf{q}\otimes\delta\mathbf{q})}{\partial\delta\mathbf{q}}\right|_{\mathbf{q}} \left.\frac{\partial\delta\mathbf{q}}{\partial\delta\boldsymbol{\theta}}\right|_{\hat{\delta\boldsymbol{\theta}}=0} \\ &= \left.\frac{\partial([\mathbf{q}]_L\delta\mathbf{q})}{\partial\delta\mathbf{q}}\right|_{\mathbf{q}} \left.\frac{\partial\begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta} \end{bmatrix}}{\partial\delta\boldsymbol{\theta}}\right|_{\hat{\delta\boldsymbol{\theta}}=0} \\ &= [\mathbf{q}]_L \frac{1}{2}\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \end{aligned}$$

which leads to

$$\mathbf{Q}_{\delta\boldsymbol{\theta}} = \frac{1}{2}\begin{bmatrix} -q_x & -q_y & -q_z \\ q_w & -q_z & q_y \\ q_z & q_w & -q_x \\ -q_y & q_x & q_w \end{bmatrix}. \tag{216}$$

3.2 Injection of the observed error into the nominal state

After the ESKF update, the nominal state gets updated with the observed error state using the appropriate compositions (sums or quaternion products, see Table 2),

$$\mathbf{x} \leftarrow \mathbf{x} \oplus \hat{\delta\mathbf{x}} \ , \tag{217}$$

that is,

$$\begin{aligned}
\mathbf{p} &\leftarrow \mathbf{p} + \hat{\delta\mathbf{p}} && (218a)\\
\mathbf{v} &\leftarrow \mathbf{v} + \hat{\delta\mathbf{v}} && (218b)\\
\mathbf{q} &\leftarrow \mathbf{q} \otimes \mathbf{q}\{\hat{\delta\boldsymbol{\theta}}\} && (218c)\\
\mathbf{a}_b &\leftarrow \mathbf{a}_b + \hat{\delta\mathbf{a}}_b && (218d)\\
\boldsymbol{\omega}_b &\leftarrow \boldsymbol{\omega}_b + \hat{\delta\boldsymbol{\omega}}_b && (218e)\\
\mathbf{g} &\leftarrow \mathbf{g} + \hat{\delta\mathbf{g}} && (218f)
\end{aligned}$$

3.3 ESKF reset

After error injection into the nominal state, the error state mean $\hat{\delta\mathbf{x}}$ gets reset. This is especially relevant for the orientation part, as the new orientation error will be expressed locally with respect to the orientation frame of the new nominal state. To make the ESKF update complete, the covariance of the error needs to be updated according to this modification.

Let us call the error reset function $g()$. It is written as follows,

$$\delta\mathbf{x} \leftarrow g(\delta\mathbf{x}) = \delta\mathbf{x} \ominus \hat{\delta\mathbf{x}} \ , \tag{219}$$

where $\ominus$ stands for the composition inverse of $\oplus$. The ESKF error reset operation is thus,

$$\begin{aligned}
\hat{\delta\mathbf{x}} &\leftarrow 0 && (220)\\
\mathbf{P} &\leftarrow \mathbf{G}\,\mathbf{P}\,\mathbf{G}^\top \ . && (221)
\end{aligned}$$

where $\mathbf{G}$ is the Jacobian matrix defined by,

$$\mathbf{G} \triangleq \left.\frac{\partial g}{\partial \delta\mathbf{x}}\right|_{\hat{\delta\mathbf{x}}} \ . \tag{222}$$

Similarly to what happened with the update Jacobian above, this Jacobian is the identity on all diagonal blocks except in the orientation error. We give here the full expression and proceed in the following section with the derivation of the orientation error block, $\partial\delta\boldsymbol{\theta}^+/\partial\delta\boldsymbol{\theta} = \mathbf{I} - \left[\frac{1}{2}\hat{\delta\boldsymbol{\theta}}\right]_\times$,

$$\mathbf{G} = \begin{bmatrix} \mathbf{I}_6 & 0 & 0 \\ 0 & \mathbf{I} - \left[\frac{1}{2}\hat{\delta\boldsymbol{\theta}}\right]_\times & 0 \\ 0 & 0 & \mathbf{I}_9 \end{bmatrix} \ . \tag{223}$$

In major cases, the error term $\hat{\delta\boldsymbol{\theta}}$ can be neglected, leading simply to a Jacobian $\mathbf{G} = \mathbf{I}_{18}$, and thus to a trivial error reset. This is what most implementations of the ESKF do. The expression here provided should produce more precise results, which might be of interest for reducing long-term error drift in odometry systems.

3.3.1 Jacobian of the reset operation with respect to the orientation error

We want to obtain the expression of the new angular error $\delta\boldsymbol{\theta}^+$ with respect to the old error $\delta\boldsymbol{\theta}$ and the observed error $\hat{\delta\boldsymbol{\theta}}$. Consider these facts:

- The true orientation does not change on error reset, *i.e.*, $\mathbf{q}_t^+ = \mathbf{q}_t$. This gives:

$$\mathbf{q}^+ \otimes \delta\mathbf{q}^+ = \mathbf{q} \otimes \delta\mathbf{q} \ . \tag{224}$$

- The observed error mean has been injected into the nominal state (see (218c) and (123)):

$$\mathbf{q}^+ = \mathbf{q} \otimes \hat{\delta\mathbf{q}} \ . \tag{225}$$

Combining both identities we obtain an expression of $\delta\boldsymbol{\theta}^+$,

$$\delta\mathbf{q}^+ = (\mathbf{q}^+)^* \otimes \mathbf{q} \otimes \delta\mathbf{q} = (\mathbf{q} \otimes \hat{\delta\mathbf{q}})^* \otimes \mathbf{q} \otimes \delta\mathbf{q} = \hat{\delta\mathbf{q}}^* \otimes \delta\mathbf{q} = [\hat{\delta\mathbf{q}}^*]_L \cdot \delta\mathbf{q} \ . \tag{226}$$

Considering that $\hat{\delta\mathbf{q}}^* \approx \begin{bmatrix} 1 \\ -\frac{1}{2}\hat{\delta\boldsymbol{\theta}} \end{bmatrix}$, the identity above can be expanded as

$$\begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta}^+ \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{2}\hat{\delta\boldsymbol{\theta}}^\top \\ -\frac{1}{2}\hat{\delta\boldsymbol{\theta}} & \mathbf{I} - \left[\frac{1}{2}\hat{\delta\boldsymbol{\theta}}\right]_\times \end{bmatrix} \cdot \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta} \end{bmatrix} + \mathcal{O}(\|\delta\boldsymbol{\theta}\|^2) \ , \tag{227}$$

which gives one scalar- and one vector- equations,

$$\frac{1}{4}\hat{\delta\boldsymbol{\theta}}^\top \delta\boldsymbol{\theta} = \mathcal{O}(\|\delta\boldsymbol{\theta}\|^2) \tag{228a}$$

$$\delta\boldsymbol{\theta}^+ = -\hat{\delta\boldsymbol{\theta}} + \left(\mathbf{I} - \left[\frac{1}{2}\hat{\delta\boldsymbol{\theta}}\right]_\times\right) \delta\boldsymbol{\theta} + \mathcal{O}(\|\delta\boldsymbol{\theta}\|^2) \ , \tag{228b}$$

among which the first one is not very informative in that it is only a relation of infinitesimals. One can show from the second equation that $\hat{\delta\boldsymbol{\theta}}^+ = 0$, which is what we expect from the reset operation. The Jacobian is obtained by simple inspection,

$$\boxed{\frac{\partial \delta\boldsymbol{\theta}^+}{\partial \delta\boldsymbol{\theta}} = \mathbf{I} - \left[\frac{1}{2}\hat{\delta\boldsymbol{\theta}}\right]_\times} \ . \tag{229}$$

4 The ESKF using global angular errors

We explore in this section the implications of having the angular error defined in the global reference, as opposed to the local definition we have used so far. We retrace the development of sections 2 and 3, and particularize the subsections that present changes with respect to the new definition.

A global definition of the angular error $\delta\boldsymbol{\theta}$ implies a composition *on the left hand side*, *i.e.*,

$$\mathbf{q}_t = \delta\mathbf{q} \ \otimes \mathbf{q} = \mathbf{q}\{\delta\boldsymbol{\theta}\} \otimes \mathbf{q} \ .$$

We remark for the sake of completeness that we keep the local definition of the angular rates vector $\boldsymbol{\omega}$, *i.e.*, $\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \otimes \boldsymbol{\omega}$ in continuous time, and therefore $\mathbf{q} \leftarrow \mathbf{q} \otimes \mathbf{q}\{\boldsymbol{\omega}\Delta t\}$ in discrete time, regardless of the angular error being defined globally. This is so for convenience, as the measure of the angular rates provided by the gyrometers is in body frame, that is, local.

4.1 System kinematics in continuous time

4.1.1 The true- and nominal-state kinematics

True and nominal kinematics do not involve errors and their equations are unchanged.

4.1.2 The error-state kinematics

We start by writing the equations of the error-state kinematics, and proceed afterwards with comments and proofs.

$$\begin{aligned}
\dot{\delta\mathbf{p}} &= \delta\mathbf{v} && (230a)\\
\dot{\delta\mathbf{v}} &= -\left[\mathbf{R}(\mathbf{a}_m - b\mathbf{a}_b)\right]_\times \delta\boldsymbol{\theta} - \mathbf{R}\delta\mathbf{a}_b + \delta\mathbf{g} - \mathbf{R}\mathbf{a}_n && (230b)\\
\dot{\delta\boldsymbol{\theta}} &= -\mathbf{R}\delta\boldsymbol{\omega}_b - \mathbf{R}\boldsymbol{\omega}_n && (230c)\\
\dot{\delta\mathbf{a}}_b &= \mathbf{a}_w && (230d)\\
\dot{\delta\boldsymbol{\omega}}_b &= \boldsymbol{\omega}_w && (230e)\\
\dot{\delta\mathbf{g}} &= 0 \ , && (230f)
\end{aligned}$$

where, again, all equations except those of $\dot{\delta\mathbf{v}}$ and $\dot{\delta\boldsymbol{\theta}}$ are trivial. The non-trivial expressions are developed below.

Equation (230b): The linear velocity error. We wish to determine $\dot{\delta\mathbf{v}}$, the dynamics of the velocity errors. We start with the following relations

$$\begin{aligned}
\mathbf{R}_t &= (\mathbf{I} + [\delta\boldsymbol{\theta}]_\times)\mathbf{R} + O(\|\delta\boldsymbol{\theta}\|^2) && (231)\\
\dot{\mathbf{v}} &= \mathbf{R}\mathbf{a}_\mathcal{B} + \mathbf{g} \ , && (232)
\end{aligned}$$

where (231) is the small-signal approximation of $\mathbf{R}_t$ using a globally defined error, and in (232) we introduced $\mathbf{a}_\mathcal{B}$ and $\delta\mathbf{a}_\mathcal{B}$ as the large- and small- signal accelerations in body frame, defined in(176) and (177), as we did for the locally-defined case.

We proceed by writing the expression (170b) of $\dot{\mathbf{v}}_t$ in two different forms (left and right developments), where the terms $O(\|\delta\boldsymbol{\theta}\|^2)$ have been ignored,

$$\begin{aligned}
\dot{\mathbf{v}} + \dot{\delta\mathbf{v}} = \boxed{\dot{\mathbf{v}}_t} &= (\mathbf{I} + [\delta\boldsymbol{\theta}]_\times)\mathbf{R}(\mathbf{a}_\mathcal{B} + \delta\mathbf{a}_\mathcal{B}) + \mathbf{g} + \delta\mathbf{g} \\
\mathbf{R}\mathbf{a}_\mathcal{B} + \mathbf{g} + \dot{\delta\mathbf{v}} = \quad &= \mathbf{R}\mathbf{a}_\mathcal{B} + \mathbf{R}\delta\mathbf{a}_\mathcal{B} + [\delta\boldsymbol{\theta}]_\times \mathbf{R}\mathbf{a}_\mathcal{B} + [\delta\boldsymbol{\theta}]_\times \mathbf{R}\delta\mathbf{a}_\mathcal{B} + \mathbf{g} + \delta\mathbf{g}
\end{aligned}$$

This leads after removing $\mathbf{R}\mathbf{a}_\mathcal{B} + \mathbf{g}$ from left and right to

$$\dot{\delta\mathbf{v}} = \mathbf{R}\delta\mathbf{a}_\mathcal{B} + [\delta\boldsymbol{\theta}]_\times \mathbf{R}(\mathbf{a}_\mathcal{B} + \delta\mathbf{a}_\mathcal{B}) + \delta\mathbf{g} \tag{233}$$

Eliminating the second order terms and reorganizing some cross-products (with $[\mathbf{a}]_\times \mathbf{b} = -[\mathbf{b}]_\times \mathbf{a}$), we get

$$\dot{\delta\mathbf{v}} = \mathbf{R}\delta\mathbf{a}_\mathcal{B} - [\mathbf{R}\mathbf{a}_\mathcal{B}]_\times \delta\boldsymbol{\theta} + \delta\mathbf{g} \;, \tag{234}$$

and finally, recalling (176) and (177) and rearranging, we obtain the expression of the derivative of the velocity error,

$$\boxed{\dot{\delta\mathbf{v}} = -[\mathbf{R}(\mathbf{a}_m - \mathbf{a}_b)]_\times \delta\boldsymbol{\theta} - \mathbf{R}\delta\mathbf{a}_b + \delta\mathbf{g} - \mathbf{R}\mathbf{a}_n} \tag{235}$$

Equation (230c): The orientation error. We start by writing the true- and nominal- definitions of the quaternion derivatives,

$$\dot{\mathbf{q}}_t = \frac{1}{2}\mathbf{q}_t \otimes \boldsymbol{\omega}_t \tag{236}$$

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \otimes \boldsymbol{\omega} \;, \tag{237}$$

and remind that we are using a globally-defined angular error, *i.e.*,

$$\mathbf{q}_t = \delta\mathbf{q} \otimes \mathbf{q} \;. \tag{238}$$

As we did for the locally-defined error case, we also group large- and small-signal angular rates (188–189). We proceed by computing $\dot{\mathbf{q}}_t$ by two different means (left and right developments),

$$\begin{aligned}
(\delta\mathbf{q} \dot{\otimes} \mathbf{q}) = \boxed{\dot{\mathbf{q}}_t} &= \frac{1}{2}\mathbf{q}_t \otimes \boldsymbol{\omega}_t \\
\dot{\delta\mathbf{q}} \otimes \mathbf{q} + \delta\mathbf{q} \otimes \dot{\mathbf{q}} = \quad &= \frac{1}{2}\delta\mathbf{q} \otimes \mathbf{q} \otimes \boldsymbol{\omega}_t \\
\dot{\delta\mathbf{q}} \otimes \mathbf{q} + \frac{1}{2}\delta\mathbf{q} \otimes \mathbf{q} \otimes \boldsymbol{\omega} = \quad &
\end{aligned}$$

Having $\boldsymbol{\omega}_t = \boldsymbol{\omega} + \delta\boldsymbol{\omega}$, this reduces to

$$\dot{\delta\mathbf{q}} \otimes \mathbf{q} = \frac{1}{2}\delta\mathbf{q} \otimes \mathbf{q} \otimes \delta\boldsymbol{\omega} \ . \tag{239}$$

Right-multiplying left and right terms by $\mathbf{q}^*$, and recalling that $\mathbf{q} \otimes \delta\boldsymbol{\omega} \otimes \mathbf{q}^* \equiv \mathbf{R}\delta\boldsymbol{\omega}$, we can further develop as follows,

$$\begin{aligned}
\dot{\delta\mathbf{q}} &= \frac{1}{2}\delta\mathbf{q} \otimes \mathbf{q} \otimes \delta\boldsymbol{\omega} \otimes \mathbf{q}^* \\
&= \frac{1}{2}\delta\mathbf{q} \otimes (\mathbf{R}\delta\boldsymbol{\omega}) \\
&= \frac{1}{2}\delta\mathbf{q} \otimes \delta\boldsymbol{\omega}_G \ ,
\end{aligned} \tag{240}$$

with $\delta\boldsymbol{\omega}_G \triangleq \mathbf{R}\delta\boldsymbol{\omega}$ the small-signal angular rate expressed in the global frame. Then,

$$\begin{aligned}
\begin{bmatrix} 0 \\ \dot{\delta\boldsymbol{\theta}} \end{bmatrix} = \boxed{2\dot{\delta\mathbf{q}}} &= \delta\mathbf{q} \otimes \delta\boldsymbol{\omega}_G \\
&= \Omega(\delta\boldsymbol{\omega}_G)\,\delta\mathbf{q} \\
&= \begin{bmatrix} 0 & -\delta\boldsymbol{\omega}_G^\top \\ \delta\boldsymbol{\omega}_G & -[\delta\boldsymbol{\omega}_G]_\times \end{bmatrix} \begin{bmatrix} 1 \\ \delta\boldsymbol{\theta}/2 \end{bmatrix} + O(\|\delta\boldsymbol{\theta}\|^2) \ ,
\end{aligned} \tag{241}$$

which results in one scalar- and one vector- equalities

$$0 = \delta\boldsymbol{\omega}_G^\top \delta\boldsymbol{\theta} + O(|\delta\boldsymbol{\theta}|^2) \tag{242a}$$

$$\dot{\delta\boldsymbol{\theta}} = \delta\boldsymbol{\omega}_G - \frac{1}{2}\left[\delta\boldsymbol{\omega}_G\right]_\times \delta\boldsymbol{\theta} + O(\|\delta\boldsymbol{\theta}\|^2). \tag{242b}$$

The first equation leads to $\delta\boldsymbol{\omega}_G^\top \delta\boldsymbol{\theta} = O(\|\delta\boldsymbol{\theta}\|^2)$, which is formed by second-order infinitesimals, not very useful. The second equation yields, after neglecting all second-order terms,

$$\dot{\delta\boldsymbol{\theta}} = \delta\boldsymbol{\omega}_G = \mathbf{R}\delta\boldsymbol{\omega} \ . \tag{243}$$

Finally, recalling (189), we obtain the linearized dynamics of the global angular error,

$$\boxed{\dot{\delta\boldsymbol{\theta}} = -\mathbf{R}\delta\boldsymbol{\omega}_b - \mathbf{R}\boldsymbol{\omega}_n} \ . \tag{244}$$

4.2 System kinematics in discrete time

4.2.1 The nominal state

The nominal state equations do not involve errors and are therefore the same as in the case where the orientation error is defined locally.

4.2.2 The error state

Using Euler integration, we obtain the following set of differences equations,

$$\delta\mathbf{p} \leftarrow \delta\mathbf{p} + \delta\mathbf{v}\,\Delta t \tag{245a}$$
$$\delta\mathbf{v} \leftarrow \delta\mathbf{v} + (-\left[\mathbf{R}(\mathbf{a}_m - \mathbf{a}_b)\right]_\times \delta\boldsymbol{\theta} - \mathbf{R}\delta\mathbf{a}_b + \delta\mathbf{g})\Delta t + \mathbf{v_i} \tag{245b}$$
$$\delta\boldsymbol{\theta} \leftarrow \delta\boldsymbol{\theta} - \mathbf{R}\delta\boldsymbol{\omega}_b\Delta t + \boldsymbol{\theta}_\mathbf{i} \tag{245c}$$
$$\delta\mathbf{a}_b \leftarrow \delta\mathbf{a}_b + \mathbf{a_i} \tag{245d}$$
$$\delta\boldsymbol{\omega}_b \leftarrow \delta\boldsymbol{\omega}_b + \boldsymbol{\omega}_\mathbf{i} \tag{245e}$$
$$\delta\mathbf{g} \leftarrow \delta\mathbf{g}. \tag{245f}$$

4.2.3 The error state Jacobian and perturbation matrices

The Transition matrix is obtained by simple inspection of the equations above,

$$\mathbf{F_x} = \begin{bmatrix} \mathbf{I} & \mathbf{I}\Delta t & 0 & 0 & 0 & 0 \\ 0 & \mathbf{I} & \boxed{-\left[\mathbf{R}(\mathbf{a}_m - \mathbf{a}_b)\right]_\times \Delta t} & -\mathbf{R}\Delta t & 0 & \mathbf{I}\Delta t \\ 0 & 0 & \boxed{\mathbf{I}} & 0 & \boxed{-\mathbf{R}\Delta t} & 0 \\ 0 & 0 & 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{I} \end{bmatrix} . \tag{246}$$

We observe three changes with respect to the case with a locally-defined angular error (compare the boxed terms in the Jacobian above to the ones in (205)); these changes are summarized in Table 3.

The perturbation Jacobian and the perturbation matrix are unchanged after considering isotropic noises and the developments of App. E,

$$\mathbf{F_i} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \mathbf{I} & 0 & 0 & 0 \\ 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \mathbf{I} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad , \quad \mathbf{Q_i} = \begin{bmatrix} \mathbf{V_i} & 0 & 0 & 0 \\ 0 & \boldsymbol{\Theta}_\mathbf{i} & 0 & 0 \\ 0 & 0 & \mathbf{A_i} & 0 \\ 0 & 0 & 0 & \boldsymbol{\Omega}_\mathbf{i} \end{bmatrix} . \tag{247}$$

4.3 Fusing with complementary sensory data

The fusing equations involving the ESKF machinery vary only slightly when considering global angular errors. We revise these variations in the error state observation via ESKF correction, the injection of the error into the nominal state, and the reset step.

4.3.1 Error state observation

The only difference with respect to the local error definition is in the Jacobian block of the observation function that relates the orientation to the angular error. This new block is developed below.

Using (17–19) and the first-order approximation $\delta\mathbf{q} \to \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta} \end{bmatrix}$, the quaternion term $\mathbf{Q}_{\delta\boldsymbol{\theta}}$ may be derived as follows,

$$\mathbf{Q}_{\delta\boldsymbol{\theta}} \triangleq \left.\frac{\partial(\delta\mathbf{q}\otimes\mathbf{q})}{\partial\delta\boldsymbol{\theta}}\right|_{\mathbf{q}} = \left.\frac{\partial(\delta\mathbf{q}\otimes\mathbf{q})}{\partial\delta\mathbf{q}}\right|_{\mathbf{q}} \left.\frac{\partial\delta\mathbf{q}}{\partial\delta\boldsymbol{\theta}}\right|_{\hat{\delta\boldsymbol{\theta}}=0} \tag{248a}$$

$$= [\mathbf{q}]_R \, \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{248b}$$

$$= \frac{1}{2} \begin{bmatrix} -q_x & -q_y & -q_z \\ q_w & q_z & -q_y \\ -q_z & q_w & q_x \\ q_y & -q_x & q_w \end{bmatrix} . \tag{248c}$$

4.3.2 Injection of the observed error into the nominal state

The composition $\mathbf{x} \leftarrow \mathbf{x} \oplus \hat{\delta\mathbf{x}}$ of the nominal and error states is depicted as follows,

$$\mathbf{p} \leftarrow \mathbf{p} + \delta\mathbf{p} \tag{249a}$$

$$\mathbf{v} \leftarrow \mathbf{v} + \delta\mathbf{v} \tag{249b}$$

$$\mathbf{q} \leftarrow \mathbf{q}\{\hat{\delta\boldsymbol{\theta}}\} \otimes \mathbf{q} \tag{249c}$$

$$\mathbf{a}_b \leftarrow \mathbf{a}_b + \delta\mathbf{a}_b \tag{249d}$$

$$\boldsymbol{\omega}_b \leftarrow \boldsymbol{\omega}_b + \delta\boldsymbol{\omega}_b \tag{249e}$$

$$\mathbf{g} \leftarrow \mathbf{g} + \delta\mathbf{g} \ . \tag{249f}$$

where only the equation for the quaternion update has been affected. This is summarized in Table 3.

4.3.3 ESKF reset

The ESKF error mean is reset, and the covariance updated, according to,

$$\hat{\delta\mathbf{x}} \leftarrow 0 \tag{250}$$

$$\mathbf{P} \leftarrow \mathbf{G}\mathbf{P}\mathbf{G}^\top \tag{251}$$

with the Jacobian

$$\mathbf{G} = \begin{bmatrix} \mathbf{I}_6 & 0 & 0 \\ 0 & \mathbf{I} + \left[\frac{1}{2}\hat{\delta\boldsymbol{\theta}}\right]_\times & 0 \\ 0 & 0 & \mathbf{I}_9 \end{bmatrix} \tag{252}$$

whose non-trivial term is developed as follows. Our goal is to obtain the expression of the new angular error $\delta\boldsymbol{\theta}^+$ with respect to the old error $\delta\boldsymbol{\theta}$. We consider these facts:

- The true orientation does not change on error reset, *i.e.*, $\mathbf{q}_t^+ \equiv \mathbf{q}_t$. This gives:

$$\delta\mathbf{q}^+ \otimes \mathbf{q}^+ = \delta\mathbf{q} \otimes \mathbf{q} \,. \tag{253}$$

- The observed error mean has been injected into the nominal state (see (218c) and (123)):

$$\mathbf{q}^+ = \hat{\delta\mathbf{q}} \otimes \mathbf{q} \,. \tag{254}$$

Combining both identities we obtain an expression of the new orientation error with respect to the old one and the observed error $\hat{\delta\mathbf{q}}$,

$$\delta\mathbf{q}^+ = \delta\mathbf{q} \otimes \hat{\delta\mathbf{q}}^* = [\hat{\delta\mathbf{q}}^*]_R \cdot \delta\mathbf{q} \,. \tag{255}$$

Considering that $\hat{\delta\mathbf{q}}^* \approx \begin{bmatrix} 1 \\ -\frac{1}{2}\hat{\delta\boldsymbol{\theta}} \end{bmatrix}$, the identity above can be expanded as

$$\begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta}^+ \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{2}\hat{\delta\boldsymbol{\theta}}^\top \\ -\frac{1}{2}\hat{\delta\boldsymbol{\theta}} & \mathbf{I} + \left[\frac{1}{2}\hat{\delta\boldsymbol{\theta}}\right]_\times \end{bmatrix} \cdot \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta} \end{bmatrix} + \mathcal{O}(\|\delta\boldsymbol{\theta}\|^2) \tag{256}$$

which gives one scalar- and one vector- equations,

$$\frac{1}{4}\hat{\delta\boldsymbol{\theta}}^\top \delta\boldsymbol{\theta} = \mathcal{O}(\|\delta\boldsymbol{\theta}\|^2) \tag{257a}$$

$$\delta\boldsymbol{\theta}^+ = -\hat{\delta\boldsymbol{\theta}} + \left(\mathbf{I} + \left[\frac{1}{2}\hat{\delta\boldsymbol{\theta}}\right]_\times\right)\delta\boldsymbol{\theta} + \mathcal{O}(\|\delta\boldsymbol{\theta}\|^2) \tag{257b}$$

among which the first one is not very informative in that it is only a relation of infinitesimals. One can show from the second equation that $\hat{\delta\boldsymbol{\theta}}^+ = 0$, which is what we expect from the reset operation. The Jacobian is obtained by simple inspection,

$$\boxed{\frac{\partial\delta\boldsymbol{\theta}^+}{\partial\delta\boldsymbol{\theta}} = \mathbf{I} + \left[\frac{1}{2}\hat{\delta\boldsymbol{\theta}}\right]_\times} \,. \tag{258}$$

The difference with respect to the local error case is summarized in Table 3.

A Runge-Kutta numerical integration methods

We aim at integrating nonlinear differential equations of the form

$$\dot{\mathbf{x}} = f(t, \mathbf{x}) \tag{259}$$

Table 3: Algorithm modifications related to the definition of the orientation errors.

Context	Item	local angular error	global angular error
Error composition	$\mathbf{q}_t$	$\mathbf{q}_t = \mathbf{q} \otimes \delta\mathbf{q}$	$\mathbf{q}_t = \delta\mathbf{q} \otimes \mathbf{q}$
Euler integration	$\partial\delta\mathbf{v}^+/\partial\delta\boldsymbol{\theta}$	$-\mathbf{R}\left[\mathbf{a}_m - \mathbf{a}_b\right]_\times \Delta t$	$-\left[\mathbf{R}(\mathbf{a}_m - \mathbf{a}_b)\right]_\times \Delta t$
	$\partial\delta\boldsymbol{\theta}^+/\partial\delta\boldsymbol{\theta}$	$\mathbf{R}^\top\{(\boldsymbol{\omega}_m - \boldsymbol{\omega}_b)\Delta t\}$	$\mathbf{I}$
	$\partial\delta\boldsymbol{\theta}^+/\partial\delta\boldsymbol{\omega}_b$	$-\mathbf{I}\Delta t$	$-\mathbf{R}\Delta t$
Error observation	$\mathbf{Q}_{\delta\boldsymbol{\theta}}$	$\frac{1}{2}\begin{bmatrix} -q_x & -q_y & -q_z \\ q_w & -q_z & q_y \\ q_z & q_w & -q_x \\ -q_y & q_x & q_w \end{bmatrix}$	$\frac{1}{2}\begin{bmatrix} -q_x & -q_y & -q_z \\ q_w & q_z & -q_y \\ -q_z & q_w & q_x \\ q_y & -q_x & q_w \end{bmatrix}$
Error injection		$\mathbf{q} \leftarrow \mathbf{q} \otimes \mathbf{q}\{\hat{\delta\boldsymbol{\theta}}\}$	$\mathbf{q} \leftarrow \mathbf{q}\{\hat{\delta\boldsymbol{\theta}}\} \otimes \mathbf{q}$
Error reset	$\partial\delta\boldsymbol{\theta}^+/\partial\delta\boldsymbol{\theta}$	$\mathbf{I} - \left[\frac{1}{2}\hat{\delta\boldsymbol{\theta}}\right]_\times$	$\mathbf{I} + \left[\frac{1}{2}\hat{\delta\boldsymbol{\theta}}\right]_\times$

over a limited time interval Δt, in order to convert them to a differences equation, *i.e.*,

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \int_t^{t+\Delta t} f(\tau, \mathbf{x}(\tau))d\tau \ , \tag{260}$$

or equivalently, if we assume that $t_n = n\Delta t$ and $\mathbf{x}_n \triangleq \mathbf{x}(t_n)$,

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \int_{n\Delta t}^{(n+1)\Delta t} f(\tau, \mathbf{x}(\tau))d\tau \ . \tag{261}$$

One of the most utilized family of methods is the Runge-Kutta methods (from now on, RK). These methods use several iterations to estimate the derivative over the interval, and then use this derivative to integrate over the step Δt.

In the sections that follow, several RK methods are presented, from the simplest one to the most general one, and are named according to their most common name.

NOTE: All the material here is taken from the *Runge-Kutta method* entry in the English Wikipedia.

A.1 The Euler method

The Euler method assumes that the derivative $f(\cdot)$ is constant over the interval, and therefore

$$\boxed{\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t\cdot f(t_n, \mathbf{x}_n)\ .} \tag{262}$$

Put as a general RK method, this corresponds to a single-stage method, which can be depicted as follows. Compute the derivative at the initial point,

$$k_1 = f(t_n, \mathbf{x}_n)\ , \tag{263}$$

and use it to compute the integrated value at the end point,

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t\cdot k_1\ . \tag{264}$$

A.2 The midpoint method

The midpoint method assumes that the derivative is the one at the midpoint of the interval, and makes one iteration to compute the value of $\mathbf{x}$ at this midpoint, *i.e.*,

$$\boxed{\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t\cdot f\Big(t_n + \frac{1}{2}\Delta t\ ,\ \mathbf{x}_n + \frac{1}{2}\Delta t\cdot f(t_n, \mathbf{x}_n)\Big)}\ . \tag{265}$$

The midpoint method can be explained as a two-step method as follows. First, use the Euler method to integrate until the midpoint, using k_1 as defined previously,

$$k_1 = f(t_n, \mathbf{x}_n) \tag{266}$$

$$\mathbf{x}(t_n + \tfrac{1}{2}\Delta t) = \mathbf{x}_n + \frac{1}{2}\Delta t\cdot k_1\ . \tag{267}$$

Then use this value to evaluate the derivative at the midpoint, k_2, leading to the integration

$$k_2 = f(t_n + \tfrac{1}{2}\Delta t\ ,\ \mathbf{x}(t_n + \tfrac{1}{2}\Delta t)) \tag{268}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t\cdot k_2\ . \tag{269}$$

A.3 The RK4 method

This is usually referred to as simply the Runge-Kutta method. It assumes evaluation values for $f()$ at the start, midpoint and end of the interval. And it uses four stages or iterations to compute the integral, with four derivatives, $k_1 \dots k_4$, that are obtained sequentially. These derivatives, or *slopes*, are then weight-averaged to obtain the 4th-order estimate of the derivative in the interval.

The RK4 method is better specified as a small algorithm than a one-step formula like the two methods above. The RK4 integration step is,

$$\boxed{\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{\Delta t}{6}\Big(k_1 + 2k_2 + 2k_3 + k_4\Big)} \ , \tag{270}$$

that is, the increment is computed by assuming a slope which is the weighted average of the slopes k_1, k_2, k_3, k_4, with

$$k_1 = f(t_n, \mathbf{x}_n) \tag{271}$$
$$k_2 = f\Big(t_n + \frac{1}{2}\Delta t \ , \ \mathbf{x}_n + \frac{\Delta t}{2}k_1\Big) \tag{272}$$
$$k_3 = f\Big(t_n + \frac{1}{2}\Delta t \ , \ \mathbf{x}_n + \frac{\Delta t}{2}k_2\Big) \tag{273}$$
$$k_4 = f\Big(t_n + \Delta t \ , \ \mathbf{x}_n + \Delta t \cdot k_3\Big) \ . \tag{274}$$

The different slopes have the following interpretation:

- k_1 is the slope at the beginning of the interval, using $\mathbf{x}_n$, (Euler's method);
- k_2 is the slope at the midpoint of the interval, using $\mathbf{x}_n + \frac{1}{2}\Delta t \cdot k_1$, (midpoint method);
- k_3 is again the slope at the midpoint, but now using $\mathbf{x}_n + \frac{1}{2}\Delta t \cdot k_2$;
- k_4 is the slope at the end of the interval, using $\mathbf{x}_n + \Delta t \cdot k_3$.

A.4 General Runge-Kutta method

More elaborated RK methods are possible. They aim at either reduce the error and/or increase stability. They take the general form

$$\boxed{\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t \sum_{i=1}^{s} b_i k_i} \ , \tag{275}$$

where

$$k_i = f\Big(t_n + \Delta t \cdot c_i, \mathbf{x}_n + \Delta t \sum_{j=1}^{s} a_{ij} k_j\Big) \ , \tag{276}$$

that is, the number of iterations (the order of the method) is s, the averaging weights are defined by b_i, the evaluation time instants by c_i, and the slopes k_i are determined using the values a_{ij}. Depending on the structure of the terms a_{ij}, one can have *explicit* or *implicit* RK methods.

- In explicit methods, all k_i are computed sequentially, *i.e.*, using only previously computed values. This implies that the matrix $[a_{ij}]$ is lower triangular with zero diagonal entries (*i.e.*, $a_{ij} = 0$ for $j \geq i$). Euler, midpoint and RK4 methods are explicit.

- Implicit methods have a full $[a_{ij}]$ matrix and require the solution of a linear set of equations to determine all k_i. They are therefore costlier to compute, but they are able to improve on accuracy and stability with respect to explicit methods.

Please refer to specialized documentation for more detailed information.

B Closed-form integration methods

In many cases it is possible to arrive to a closed-form expression for the integration step. We consider now the case of a first-order linear differential equation,

$$\dot{\mathbf{x}}(t) = \mathbf{A}\cdot\mathbf{x}(t) \ , \tag{277}$$

that is, the relation is linear and constant over the interval. In such cases, the integration over the interval $[t_n, t_n + \Delta t]$ results in

$$\mathbf{x}_{n+1} = e^{\mathbf{A}\Delta t}\mathbf{x}_n = \Phi\mathbf{x}_n \ , \tag{278}$$

where Φ is known as the transition matrix. The Taylor expansion of this transition matrix is

$$\Phi = e^{\mathbf{A}\Delta t} = \mathbf{I} + \mathbf{A}\Delta t + \frac{1}{2}\mathbf{A}^2\Delta t^2 + \frac{1}{3!}\mathbf{A}^3\Delta t^3 + \cdots = \sum_{k=0}^{\infty}\frac{1}{k!}\mathbf{A}^k\Delta t^k \ . \tag{279}$$

When writing this series for known instances of $\mathbf{A}$, it is sometimes possible to identify known series in the result. This allows writing the resulting integration in closed form. A few examples follow.

B.1 Integration of the angular error

For example, consider the angular error dynamics without bias and noise (a cleaned version of Eq. (173c)),

$$\dot{\delta\boldsymbol{\theta}} = -\left[\boldsymbol{\omega}\right]_\times \delta\boldsymbol{\theta} \tag{280}$$

Its transition matrix can be written as a Taylor series,

$$\Phi = e^{-[\boldsymbol{\omega}]_\times \Delta t} \tag{281}$$

$$= \mathbf{I} - [\boldsymbol{\omega}]_\times \Delta t + \frac{1}{2}[\boldsymbol{\omega}]_\times^2 \Delta t^2 - \frac{1}{3!}[\boldsymbol{\omega}]_\times^3 \Delta t^3 + \frac{1}{4!}[\boldsymbol{\omega}]_\times^4 \Delta t^4 - \dots \tag{282}$$

Now defining $\boldsymbol{\omega}\Delta t \triangleq \mathbf{u}\Delta\theta$, the unitary axis of rotation and the rotated angle, and applying (71), we can group terms and get

$$\begin{aligned}
\Phi &= \mathbf{I} - [\mathbf{u}]_\times \Delta\theta + \frac{1}{2}[\mathbf{u}]_\times^2 \Delta\theta^2 - \frac{1}{3!}[\mathbf{u}]_\times^3 \Delta\theta^3 + \frac{1}{4!}[\mathbf{y}]_\times^4 \Delta\theta^4 - \dots \\
&= \mathbf{I} - [\mathbf{u}]_\times \left(\Delta\theta - \frac{\Delta\theta^3}{3!} + \frac{\Delta\theta^5}{5!} - \cdots\right) + [\mathbf{u}]_\times^2 \left(\frac{\Delta\theta^2}{2!} - \frac{\Delta\theta^4}{4!} + \frac{\Delta\theta^6}{6!} - \cdots\right) \\
&= \mathbf{I} - [\mathbf{u}]_\times \sin\Delta\theta + [\mathbf{u}]_\times^2 (1 - \cos\Delta\theta) \ ,
\end{aligned} \tag{283}$$

which is a closed-form solution.

This solution corresponds to a rotation matrix, $\Phi = \mathbf{R}\{-\mathbf{u}\Delta\theta\} = \mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top$, according to the Rodrigues rotation formula (74), a result that could be obtained by direct inspection of (281) and recalling (68). Let us therefore write this as the final closed-form result,

$$\boxed{\Phi = \mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top} \,. \tag{284}$$

B.2 Simplified IMU example

Consider the simplified, IMU driven system with error-state dynamics governed by,

$$\dot{\delta\mathbf{p}} = \delta\mathbf{v} \tag{285a}$$

$$\dot{\delta\mathbf{v}} = -\mathbf{R}\,[\mathbf{a}]_\times\,\delta\boldsymbol{\theta} \tag{285b}$$

$$\dot{\delta\boldsymbol{\theta}} = -\,[\boldsymbol{\omega}]_\times\,\delta\boldsymbol{\theta}\ , \tag{285c}$$

where $(\mathbf{a}, \boldsymbol{\omega})$ are the IMU readings, and we have obviated gravity and sensor biases. This system is defined by the state vector and the dynamic matrix,

$$\mathbf{x} = \begin{bmatrix}\delta\mathbf{p}\\ \delta\mathbf{v}\\ \delta\boldsymbol{\theta}\end{bmatrix} \qquad \mathbf{A} = \begin{bmatrix}0 & \mathbf{P}_\mathbf{v} & 0\\ 0 & 0 & \mathbf{V}_{\boldsymbol{\theta}}\\ 0 & 0 & \Theta_{\boldsymbol{\theta}}\end{bmatrix} \,. \tag{286}$$

with

$$\mathbf{P}_\mathbf{v} = \mathbf{I} \tag{287}$$

$$\mathbf{V}_{\boldsymbol{\theta}} = -\mathbf{R}\,[\mathbf{a}]_\times \tag{288}$$

$$\Theta_{\boldsymbol{\theta}} = -\,[\boldsymbol{\omega}]_\times \tag{289}$$

Its integration with a step time Δt is $\mathbf{x}_{n+1} = e^{(\mathbf{A}\Delta t)}\cdot\mathbf{x}_n = \Phi\cdot\mathbf{x}_n$. The transition matrix Φ admits a Taylor development (279), in increasing powers of $\mathbf{A}\Delta t$. We can write a few powers of $\mathbf{A}$ to get an illustration of their general form,

$$\mathbf{A} = \begin{bmatrix}0 & \mathbf{P}_\mathbf{v} & 0\\ 0 & 0 & \mathbf{V}_{\boldsymbol{\theta}}\\ 0 & 0 & \Theta_{\boldsymbol{\theta}}\end{bmatrix}, \mathbf{A}^2 = \begin{bmatrix}0 & 0 & \mathbf{P}_v\mathbf{V}_{\boldsymbol{\theta}}\\ 0 & 0 & \mathbf{V}_{\boldsymbol{\theta}}\Theta_{\boldsymbol{\theta}}\\ 0 & 0 & \Theta_{\boldsymbol{\theta}}^2\end{bmatrix}, \mathbf{A}^3 = \begin{bmatrix}0 & 0 & \mathbf{P}_v\mathbf{V}_{\boldsymbol{\theta}}\Theta_{\boldsymbol{\theta}}\\ 0 & 0 & \mathbf{V}_{\boldsymbol{\theta}}\Theta_{\boldsymbol{\theta}}^2\\ 0 & 0 & \Theta_{\boldsymbol{\theta}}^3\end{bmatrix}, \mathbf{A}^4 = \begin{bmatrix}0 & 0 & \mathbf{P}_v\mathbf{V}_{\boldsymbol{\theta}}\Theta_{\boldsymbol{\theta}}^2\\ 0 & 0 & \mathbf{V}_{\boldsymbol{\theta}}\Theta_{\boldsymbol{\theta}}^3\\ 0 & 0 & \Theta_{\boldsymbol{\theta}}^4\end{bmatrix}, \tag{290}$$

from which it is now visible that, for $k > 1$,

$$\mathbf{A}^{k>1} = \begin{bmatrix}0 & 0 & \mathbf{P}_v\mathbf{V}_{\boldsymbol{\theta}}\Theta_{\boldsymbol{\theta}}^{k-2}\\ 0 & 0 & \mathbf{V}_{\boldsymbol{\theta}}\Theta_{\boldsymbol{\theta}}^{k-1}\\ 0 & 0 & \Theta_{\boldsymbol{\theta}}^{k}\end{bmatrix} \tag{291}$$

We can observe that the terms in the increasing powers of $\mathbf{A}$ have a fixed part and an increasing power of $\Theta_{\boldsymbol{\theta}}$. These powers can lead to closed form solutions, as in the previous section.

Let us partition the matrix Φ as follows,

$$\Phi = \begin{bmatrix} \mathbf{I} & \Phi_{\mathbf{pv}} & \Phi_{\mathbf{p}\boldsymbol{\theta}} \\ 0 & \mathbf{I} & \Phi_{\mathbf{v}\boldsymbol{\theta}} \\ 0 & 0 & \Phi_{\boldsymbol{\theta}\boldsymbol{\theta}} \end{bmatrix} , \tag{292}$$

and let us advance step by step, exploring all the non-zero blocks of Φ one by one.

Trivial diagonal terms Starting by the two upper terms in the diagonal, they are the identity as shown.

Rotational diagonal term Next is the rotational diagonal term $\Phi_{\boldsymbol{\theta}\boldsymbol{\theta}}$, relating the new angular error to the old angular error. Writing the full Taylor series for this term leads to

$$\Phi_{\boldsymbol{\theta}\boldsymbol{\theta}} = \sum_{k=0}^{\infty} \frac{1}{k!} \Theta_{\boldsymbol{\theta}}^k \Delta t^k = \sum_{k=0}^{\infty} \frac{1}{k!} \left[-\boldsymbol{\omega}\right]_\times^k \Delta t^k , \tag{293}$$

which corresponds, as we have seen in the previous section, to our well-known rotation matrix,

$$\boxed{\Phi_{\boldsymbol{\theta}\boldsymbol{\theta}} = \mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top} . \tag{294}$$

Position-vs-velocity term The simplest off-diagonal term is $\Phi_{\mathbf{pv}}$, which is

$$\boxed{\Phi_{\mathbf{pv}} = \mathbf{P}_{\mathbf{v}}\Delta t = \mathbf{I}\Delta t} . \tag{295}$$

Velocity-vs-angle term Let us now move to the term $\Phi_{\mathbf{v},\boldsymbol{\theta}}$, by writing its series,

$$\Phi_{\mathbf{v}\boldsymbol{\theta}} = \mathbf{V}_{\boldsymbol{\theta}}\Delta t + \frac{1}{2}\mathbf{V}_{\boldsymbol{\theta}}\Theta_{\boldsymbol{\theta}}\Delta t^2 + \frac{1}{3!}\mathbf{V}_{\boldsymbol{\theta}}\Theta_{\boldsymbol{\theta}}^2\Delta t^3 + \cdots \tag{296}$$

$$\Phi_{\mathbf{v}\boldsymbol{\theta}} = \Delta t \mathbf{V}_{\boldsymbol{\theta}}(\mathbf{I} + \frac{1}{2}\Theta_{\boldsymbol{\theta}}\Delta t + \frac{1}{3!}\Theta_{\boldsymbol{\theta}}^2\Delta t^2 + \cdots) \tag{297}$$

which reduces to

$$\Phi_{\mathbf{v}\boldsymbol{\theta}} = \Delta t \mathbf{V}_{\boldsymbol{\theta}} \left(\mathbf{I} + \sum_{k\geq 1} \frac{(\Theta_{\boldsymbol{\theta}}\Delta t)^k}{(k+1)!} \right) \tag{298}$$

At this point we have two options. We can truncate the series at the first significant term, obtaining $\Phi_{\mathbf{v}\boldsymbol{\theta}} = \mathbf{V}_{\boldsymbol{\theta}}\Delta t$, but this would not be a closed-form. See next section for results using this simplified method. Alternatively, let us factor $\mathbf{V}_{\boldsymbol{\theta}}$ out and write

$$\Phi_{\mathbf{v}\boldsymbol{\theta}} = \mathbf{V}_{\boldsymbol{\theta}}\Sigma_1 \tag{299}$$

with

$$\Sigma_1 = \mathbf{I}\Delta t + \frac{1}{2}\Theta_{\boldsymbol{\theta}}\Delta t^2 + \frac{1}{3!}\Theta_{\boldsymbol{\theta}}^2\Delta t^3 + \cdots . \tag{300}$$

The series Σ_1 ressembles the series we wrote for $\Phi_{\boldsymbol{\theta}\boldsymbol{\theta}}$, (293), with two exceptions:

- The powers of $\Theta_{\boldsymbol{\theta}}$ in Σ_1 do not match with the rational coefficients $\frac{1}{k!}$ and with the powers of Δt. In fact, we remark here that the subindex "1" in Σ_1 denotes the fact that one power of $\Theta_{\boldsymbol{\theta}}$ is missing in each of the members.

- Some terms at the start of the series are missing. Again, the subindex "1" indicates that one such term is missing.

The first issue may be solved by applying (71) to (289), which yields the identity

$$\Theta_{\boldsymbol{\theta}} = \frac{[\boldsymbol{\omega}]_\times^3}{\|\boldsymbol{\omega}\|^2} = \frac{-\Theta_{\boldsymbol{\theta}}^3}{\|\boldsymbol{\omega}\|^2} \,. \tag{301}$$

This expression allows us to increase the exponents of $\Theta_{\boldsymbol{\theta}}$ in the series by two, and write, if $\boldsymbol{\omega} \neq 0$,

$$\Sigma_1 = \mathbf{I}\Delta t - \frac{\Theta_{\boldsymbol{\theta}}}{\|\boldsymbol{\omega}\|^2}\left(\frac{1}{2}\Theta_{\boldsymbol{\theta}}^2\Delta t^2 + \frac{1}{3!}\Theta_{\boldsymbol{\theta}}^3\Delta t^3 + \dots\right) \,, \tag{302}$$

and $\Sigma_1 = \mathbf{I}\Delta t$ otherwise. All the powers in the new series match with the correct coefficients. Of course, and as indicated before, some terms are missing. This second issue can be solved by adding and substracting the missing terms, and substituting the full series by its closed form. We obtain

$$\Sigma_1 = \mathbf{I}\Delta t - \frac{\Theta_{\boldsymbol{\theta}}}{\|\boldsymbol{\omega}\|^2}\left(\mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top - \mathbf{I} - \Theta_{\boldsymbol{\theta}}\Delta t\right) \,, \tag{303}$$

which is a closed-form solution valid if $\boldsymbol{\omega} \neq 0$. Therefore we can finally write

$$\boxed{\Phi_{\mathbf{v}\boldsymbol{\theta}} = \begin{cases} -\mathbf{R}\,[\mathbf{a}]_\times\,\Delta t & \boldsymbol{\omega} \to 0 \\ -\mathbf{R}\,[\mathbf{a}]_\times\left(\mathbf{I}\Delta t + \dfrac{[\boldsymbol{\omega}]_\times}{\|\boldsymbol{\omega}\|^2}\left(\mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top - \mathbf{I} + [\boldsymbol{\omega}]_\times\,\Delta t\right)\right) & \boldsymbol{\omega} \neq 0 \end{cases}} \qquad \begin{matrix}(304a)\\ (304b)\end{matrix}$$

Position-vs-angle term Let us finally board the term $\Phi_{\mathbf{p}\boldsymbol{\theta}}$. Its Taylor series is,

$$\Phi_{\mathbf{p}\boldsymbol{\theta}} = \frac{1}{2}\mathbf{P_v}\mathbf{V}_{\boldsymbol{\theta}}\Delta t^2 + \frac{1}{3!}\mathbf{P_v}\mathbf{V}_{\boldsymbol{\theta}}\Theta_{\boldsymbol{\theta}}\Delta t^3 + \frac{1}{4!}\mathbf{P_v}\mathbf{V}_{\boldsymbol{\theta}}\Theta_{\boldsymbol{\theta}}^2\Delta t^4 + \cdots \tag{305}$$

We factor out the constant terms and get,

$$\Phi_{\mathbf{p}\boldsymbol{\theta}} = \mathbf{P_v}\mathbf{V}_{\boldsymbol{\theta}}\;\Sigma_2 \,, \tag{306}$$

with

$$\Sigma_2 = \frac{1}{2}\mathbf{I}\Delta t^2 + \frac{1}{3!}\Theta_{\boldsymbol{\theta}}\Delta t^3 + \frac{1}{4!}\Theta_{\boldsymbol{\theta}}^2\Delta t^4 + \cdots \,. \tag{307}$$

where we note the subindex "2" in Σ_2 admits the following interpretation:

- Two powers of $\Theta_{\boldsymbol{\theta}}$ are missing in each term of the series,

- The first two terms of the series are missing.

Again, we use (301) to increase the exponents of $\Theta_{\boldsymbol{\theta}}$, yielding

$$\Sigma_2 = \frac{1}{2}\mathbf{I}\Delta t^2 - \frac{1}{\|\boldsymbol{\omega}\|^2}\left(\frac{1}{3!}\Theta_{\boldsymbol{\theta}}^3\Delta t^3 + \frac{1}{4!}\Theta_{\boldsymbol{\theta}}^4\Delta t^4 + \cdots\right) . \tag{308}$$

We substitute the incomplete series by its closed form,

$$\Sigma_2 = \frac{1}{2}\mathbf{I}\Delta t^2 - \frac{1}{\|\boldsymbol{\omega}\|^2}\left(\mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top - \mathbf{I} - \Theta_{\boldsymbol{\theta}}\Delta t - \frac{1}{2}\Theta_{\boldsymbol{\theta}}^2\Delta t^2\right) , \tag{309}$$

which leads to the final result

$$\boxed{\Phi_{\mathbf{p}\boldsymbol{\theta}} = \begin{cases} -\mathbf{R}\left[\mathbf{a}\right]_\times \dfrac{\Delta t^2}{2} & \boldsymbol{\omega} \to 0 \quad (310a) \\ -\mathbf{R}\left[\mathbf{a}\right]_\times \left(\dfrac{1}{2}\mathbf{I}\Delta t^2 - \dfrac{1}{\|\boldsymbol{\omega}\|^2}\left(\mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top - \displaystyle\sum_{k=0}^{2}\frac{(-\left[\boldsymbol{\omega}\right]_\times \Delta t)^k}{k!}\right)\right) & \omega \neq 0 \quad (310b) \end{cases}}$$

B.3 Full IMU example

In order to give means to generalize the methods exposed in the simplified IMU example, we need to examine the full IMU case from a little closer.

Consider the full IMU system (173), which can be posed as

$$\dot{\delta\mathbf{x}} = \mathbf{A}\delta\mathbf{x} + \mathbf{B}\mathbf{w} , \tag{311}$$

whose discrete-time integration requires the transition matrix

$$\Phi = \sum_{k=0}^{\infty}\frac{1}{k!}\mathbf{A}^k\Delta t^k = \mathbf{I} + \mathbf{A}\Delta t + \frac{1}{2}\mathbf{A}^2\Delta t^2 + \dots , \tag{312}$$

which we wish to compute. The dynamic matrix $\mathbf{A}$ is block-sparse, and its blocks can be easily determined by examining the original equations (173),

$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{P_v} & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{V}_{\boldsymbol{\theta}} & \mathbf{V_a} & 0 & \mathbf{V_g} \\ 0 & 0 & \Theta_{\boldsymbol{\theta}} & 0 & \Theta_{\boldsymbol{\omega}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} . \tag{313}$$

As we did before, let us write a few powers of $\mathbf{A}$,

$$\mathbf{A}^2 = \begin{bmatrix} 0 & 0 & \mathbf{P_v V}_{\boldsymbol{\theta}} & \mathbf{P_v V_a} & 0 & \mathbf{P_v V_g} \\ 0 & 0 & \mathbf{V}_{\boldsymbol{\theta}} \Theta_{\boldsymbol{\theta}} & 0 & \mathbf{V}_{\boldsymbol{\theta}} \Theta_{\boldsymbol{\omega}} & 0 \\ 0 & 0 & \Theta_{\boldsymbol{\theta}}^2 & 0 & \Theta_{\boldsymbol{\theta}} \Theta_{\boldsymbol{\omega}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{A}^3 = \begin{bmatrix} 0 & 0 & \mathbf{P_v V}_{\boldsymbol{\theta}} \Theta_{\boldsymbol{\theta}} & 0 & \mathbf{P_v V}_{\boldsymbol{\theta}} \Theta_{\boldsymbol{\omega}} & 0 \\ 0 & 0 & \mathbf{V}_{\boldsymbol{\theta}} \Theta_{\boldsymbol{\theta}}^2 & 0 & \mathbf{V}_{\boldsymbol{\theta}} \Theta_{\boldsymbol{\theta}} \Theta_{\boldsymbol{\omega}} & 0 \\ 0 & 0 & \Theta_{\boldsymbol{\theta}}^3 & 0 & \Theta_{\boldsymbol{\theta}}^2 \Theta_{\boldsymbol{\omega}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{A}^4 = \begin{bmatrix} 0 & 0 & \mathbf{P_v V}_{\boldsymbol{\theta}} \Theta_{\boldsymbol{\theta}}^2 & 0 & \mathbf{P_v V}_{\boldsymbol{\theta}} \Theta_{\boldsymbol{\theta}} \Theta_{\boldsymbol{\omega}} & 0 \\ 0 & 0 & \mathbf{V}_{\boldsymbol{\theta}} \Theta_{\boldsymbol{\theta}}^3 & 0 & \mathbf{V}_{\boldsymbol{\theta}} \Theta_{\boldsymbol{\theta}}^2 \Theta_{\boldsymbol{\omega}} & 0 \\ 0 & 0 & \Theta_{\boldsymbol{\theta}}^4 & 0 & \Theta_{\boldsymbol{\theta}}^3 \Theta_{\boldsymbol{\omega}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} .$$

Basically, we observe the following,

- The only term in the diagonal of $\mathbf{A}$, the rotational term $\Theta_{\boldsymbol{\theta}}$, propagates right and up in the sequence of powers $\mathbf{A}^k$. All terms not affected by this propagation vanish. This propagation afects the structure of the sequence $\{\mathbf{A}^k\}$ in the three following aspects:
- The sparsity of the powers of $\mathbf{A}$ is stabilized after the 3rd power. That is to say, there are no more non-zero blocks appearing or vanishing for powers of $\mathbf{A}$ higher than 3.
- The upper-left 3×3 block, corresponding to the simplified IMU model in the previous example, has not changed with respect to that example. Therefore, its closed-form solution developed before holds.
- The terms related to the gyrometer bias error (those of the fifth column) introduce a similar series of powers of $\Theta_{\boldsymbol{\theta}}$, which can be solved with the same techniques we used in the simplified example.

We are interested at this point in finding a generalised method to board the construction of the closed-form elements of the transition matrix Φ. Let us recall the remarks we made about the series Σ_1 and Σ_2,

- The subindex coincides with the lacking powers of $\Theta_{\boldsymbol{\theta}}$ in each of the members of the series.

- The subindex coincides with the number of terms missing at the beginning of the series.

Taking care of these properties, let us introduce the series $\Sigma_n(\mathbf{X}, y)$, defined by[17]

$$\Sigma_n(\mathbf{X}, y) \triangleq \sum_{k=n}^{\infty} \frac{1}{k!}\mathbf{X}^{k-n}y^k = \sum_{k=0}^{\infty} \frac{1}{(k+n)!}\mathbf{X}^k y^{k+n} = y^n \sum_{k=0}^{\infty} \frac{1}{(k+n)!}\mathbf{X}^k y^k \tag{314}$$

in which the sum starts at term n and the terms lack n powers of the matrix $\mathbf{X}$. It follows immediately that Σ_1 and Σ_2 respond to

$$\Sigma_n = \Sigma_n(\Theta_{\boldsymbol{\theta}}, \Delta t) \ , \tag{315}$$

and that $\Sigma_0 = \mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top$. We can now write the transition matrix (312) as a function of these series,

$$\Phi = \begin{bmatrix} \mathbf{I} & \mathbf{P_v}\Delta t & \mathbf{P_v}\mathbf{V}_{\boldsymbol{\theta}}\Sigma_2 & \frac{1}{2}\mathbf{P_v}\mathbf{V_a}\Delta t^2 & \mathbf{P_v}\mathbf{V}_{\boldsymbol{\theta}}\Sigma_3\boldsymbol{\theta}_{\boldsymbol{\omega}} & \frac{1}{2}\mathbf{P_v}\mathbf{V_g}\Delta t^2 \\ 0 & \mathbf{I} & \mathbf{V}_{\boldsymbol{\theta}}\Sigma_1 & \mathbf{V_a}\Delta t & \mathbf{V}_{\boldsymbol{\theta}}\Sigma_2\boldsymbol{\theta}_{\boldsymbol{\omega}} & \mathbf{V_g}\Delta t \\ 0 & 0 & \Sigma_0 & 0 & \Sigma_1\boldsymbol{\theta}_{\boldsymbol{\omega}} & 0 \\ 0 & 0 & 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{I} \end{bmatrix} . \tag{316}$$

Our problem has now derived to the problem of finding a general, closed-form expression for Σ_n. Let us observe the closed-form results we have obtained so far,

$$\Sigma_0 = \mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top \tag{317}$$

$$\Sigma_1 = \mathbf{I}\Delta t - \frac{\Theta_{\boldsymbol{\theta}}}{\|\boldsymbol{\omega}\|^2}\left(\mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top - \mathbf{I} - \Theta_{\boldsymbol{\theta}}\Delta t\right) \tag{318}$$

$$\Sigma_2 = \frac{1}{2}\mathbf{I}\Delta t^2 - \frac{1}{\|\boldsymbol{\omega}\|^2}\left(\mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top - \mathbf{I} - \Theta_{\boldsymbol{\theta}}\Delta t - \frac{1}{2}\Theta_{\boldsymbol{\theta}}^2\Delta t^2\right) . \tag{319}$$

In order to develop Σ_3, we need to apply the identity (301) twice (because we lack three powers, and each application of (301) increases this number by only two), getting

$$\Sigma_3 = \frac{1}{3!}\mathbf{I}\Delta t^3 + \frac{\Theta_{\boldsymbol{\theta}}}{\|\boldsymbol{\omega}\|^4}\left(\frac{1}{4!}\Theta_{\boldsymbol{\theta}}^4\Delta t^4 + \frac{1}{5!}\Theta_{\boldsymbol{\theta}}^5\Delta t^5 + \dots\right) , \tag{320}$$

which leads to

$$\Sigma_3 = \frac{1}{3!}\mathbf{I}\Delta t^3 + \frac{\Theta_{\boldsymbol{\theta}}}{\|\boldsymbol{\omega}\|^4}\left(\mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top - \mathbf{I} - \Theta_{\boldsymbol{\theta}}\Delta t - \frac{1}{2}\Theta_{\boldsymbol{\theta}}^2\Delta t^2 - \frac{1}{3!}\Theta_{\boldsymbol{\theta}}^3\Delta t^3\right) . \tag{321}$$

[17]Note that, being $\mathbf{X}$ a square matrix that is not necessarily invertible (as it is the case for $\mathbf{X} = \Theta_{\boldsymbol{\theta}}$), we are not allowed to rearrange the definition of Σ_n with $\Sigma_n = \mathbf{X}^{-n}\sum_{k=n}^{\infty}\frac{1}{k!}(y\mathbf{X})^k$.

By careful inspection of the series $\Sigma_0 \ldots \Sigma_3$, we can now derive a general, closed-form expression for Σ_n, as follows,

$$\boxed{\Sigma_n = \begin{cases} \frac{1}{n!}\mathbf{I}\Delta t^n & \boldsymbol{\omega} \to 0 \quad (322a) \\ \mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top & n = 0 \quad (322b) \\ \frac{1}{n!}\mathbf{I}\Delta t^n - \frac{(-1)^{\frac{n+1}{2}}[\boldsymbol{\omega}]_\times}{\|\boldsymbol{\omega}\|^{n+1}}\left(\mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top - \sum_{k=0}^{n}\frac{(-[\boldsymbol{\omega}]_\times \Delta t)^k}{k!}\right) & n \text{ odd} \quad (322c) \\ \frac{1}{n!}\mathbf{I}\Delta t^n + \frac{(-1)^{\frac{n}{2}}}{\|\boldsymbol{\omega}\|^{n}}\left(\mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top - \sum_{k=0}^{n}\frac{(-[\boldsymbol{\omega}]_\times \Delta t)^k}{k!}\right) & n \text{ even} \quad (322d) \end{cases}}$$

The final result for the transition matrix Φ follows immediately by substituting the appropriate values of $\Sigma_n,\ n \in \{0, 1, 2, 3\}$, in the corresponding positions of (316).

It might be worth noticing that the series now appearing in these new expressions of Σ_n have a finite number of terms, and thus that they can be effectively computed. That is to say, the expression of Σ_n is a closed form as long as $n < \infty$, which is always the case. For the current example, we have $n \leq 3$ as can be observed in (316).

C Approximate methods using truncated series

In the previous section, we have devised closed-form expressions for the transition matrix of complex, IMU-driven dynamic systems written in their linearized, error-state form $\dot{\delta\mathbf{x}} = \mathbf{A}\delta\mathbf{x}$. Closed form expressions may always be of interest, but it is unclear up to which point we should be worried about high order errors and their impact on the performance of real algorithms. This remark is particularly relevant in systems where IMU integration errors are observed (and thus compensated for) at relatively high rates, such as visual-inertial or GPS-inertial fusion schemes.

In this section we devise methods for approximating the transition matrix. They start from the same assumption that the transition matrix can be expressed as a Taylor series, and then truncate these series at the most significant terms. This truncation can be done system-wise, or block-wise.

C.1 System-wise truncation

C.1.1 First order truncation: the finite differences method

A typical, widely used integration method for systems of the type

$$\dot{\mathbf{x}} = f(t, \mathbf{x})$$

is based on the finite-differences method for the computation of the derivative, *i.e.*,

$$\dot{\mathbf{x}} \triangleq \lim_{\delta t \to 0} \frac{\mathbf{x}(t + \delta t) - \mathbf{x}(t)}{\delta t} \approx \frac{\mathbf{x}_{n+1} - \mathbf{x}_n}{\Delta t}. \tag{323}$$

This leads immediately to

$$\mathbf{x}_{n+1} \approx \mathbf{x}_n + \Delta t\, f(t_n, \mathbf{x}_n) \;, \tag{324}$$

which is precisely the Euler method. Linearization of the function $f()$ at the beginning of the integration interval leads to

$$\mathbf{x}_{n+1} \approx \mathbf{x}_n + \Delta t\, \mathbf{A}\, \mathbf{x}_n \;, \tag{325a}$$

where $\mathbf{A} \triangleq \frac{\partial f}{\partial \mathbf{x}}(t_n, \mathbf{x}_n)$ is a Jacobian matrix. This is strictly equivalent to writing the exponential solution to the linearized differential equation and truncating the series at the linear term (*i.e.*, the following relation is identical to the previous one),

$$\mathbf{x}_{n+1} = e^{\mathbf{A}\Delta t}\mathbf{x}_n \approx (\mathbf{I} + \Delta t\, \mathbf{A})\, \mathbf{x}_n \;. \tag{325b}$$

This means that the Euler method (App. A.1), the finite-differences method, and the first-order system-wise Taylor truncation method, are all the same. We get the approximate transition matrix,

$$\boxed{\Phi \approx \mathbf{I} + \Delta t \mathbf{A}} \;. \tag{326}$$

For the simplified IMU example of Section B.2, the finite-differences method results in the approximated transition matrix

$$\Phi \approx \begin{bmatrix} \mathbf{I} & \mathbf{I}\Delta t & 0 \\ 0 & \mathbf{I} & -\mathbf{R}\,[\mathbf{a}]_\times\,\Delta t \\ 0 & 0 & \mathbf{I} - [\boldsymbol{\omega}\Delta t]_\times \end{bmatrix} \;. \tag{327}$$

However, we already know from Section B.1 that the rotational term has a compact, closed-form solution, $\Phi_{\boldsymbol{\theta}\boldsymbol{\theta}} = \mathbf{R}(\boldsymbol{\omega}\Delta t)^\top$. It is convenient to re-write the transition matrix according to it,

$$\Phi \approx \begin{bmatrix} \mathbf{I} & \mathbf{I}\Delta t & 0 \\ 0 & \mathbf{I} & -\mathbf{R}\,[\mathbf{a}]_\times\,\Delta t \\ 0 & 0 & \mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top \end{bmatrix} \;. \tag{328}$$

C.1.2 N-th order truncation

Truncating at higher orders will increase the precision of the approximated transition matrix. A particularly interesting order of truncation is that which exploits the sparsity of the result to its maximum. In other words, the order after which no new non-zero terms appear.

For the simplified IMU example of Section B.2, this order is 2, resulting in

$$\boldsymbol{\Phi} \approx \mathbf{I} + \mathbf{A}\Delta t + \frac{1}{2}\mathbf{A}^2\Delta t^2 = \begin{bmatrix} \mathbf{I} & \mathbf{I}\Delta t & -\frac{1}{2}\mathbf{R}\,[\mathbf{a}]_\times\,\Delta t^2 \\ 0 & \mathbf{I} & -\mathbf{R}\,[\mathbf{a}]_\times\,(\mathbf{I} - \frac{1}{2}\,[\boldsymbol{\omega}]_\times\,\Delta t)\Delta t \\ 0 & 0 & \mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top \end{bmatrix} \;. \tag{329}$$

In the full IMU example of Section B.3, the is order 3, resulting in

$$\mathbf{\Phi} \approx \mathbf{I} + \mathbf{A}\Delta t + \frac{1}{2}\mathbf{A}^2\Delta t^2 + \frac{1}{6}\mathbf{A}^3\Delta t^3 \ , \tag{330}$$

whose full form is not given here for space reasons. The reader may consult the expressions of $\mathbf{A}$, $\mathbf{A}^2$ and $\mathbf{A}^3$ in Section B.3.

C.2 Block-wise truncation

A fairly good approximation to the closed forms previously explained results from truncating the Taylor series of each block of the transition matrix at the first significant term. That is, instead of truncating the series in full powers of $\mathbf{A}$, as we have just made above, we regard each block individually. Therefore, truncation needs to be analyzed in a per-block basis. We explore it with two examples.

For the simplified IMU example of Section B.2, we had series Σ_1 and Σ_2, which we can truncate as follows

$$\Sigma_1 = \mathbf{I}\Delta t + \frac{1}{2}\Theta_{\boldsymbol{\theta}}\Delta t^2 + \cdots \approx \mathbf{I}\Delta t \tag{331}$$

$$\Sigma_2 = \frac{1}{2}\mathbf{I}\Delta t^2 + \frac{1}{3!}\Theta_{\boldsymbol{\theta}}\Delta t^3 + \cdots \approx \frac{1}{2}\mathbf{I}\Delta t^2 \ . \tag{332}$$

This leads to the approximate transition matrix

$$\Phi \approx \begin{bmatrix} \mathbf{I} & \mathbf{I}\Delta t & -\frac{1}{2}\mathbf{R}\left[\mathbf{a}\right]_\times \Delta t^2 \\ 0 & \mathbf{I} & -\mathbf{R}\left[\mathbf{a}\right]_\times \Delta t \\ 0 & 0 & \mathbf{R}(\boldsymbol{\omega}\Delta t)^\top \end{bmatrix} \ , \tag{333}$$

which is more accurate than the one in the system-wide first-order truncation above (because of the upper-right term which has now appeared), yet it remains easy to obtain and compute, especially when compared to the closed forms developed in Section B. Again, observe that we have taken the closed-form for the lowest term, *i.e.*, $\Phi_{\boldsymbol{\theta}\boldsymbol{\theta}} = \mathbf{R}(\boldsymbol{\omega}\Delta t)^\top$.

In the general case, it suffices to approximate each Σ_n except Σ_0 by the first term of its series, *i.e.*,

$$\boxed{\Sigma_0 = \mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top \ , \qquad \Sigma_{n>0} \approx \frac{1}{n!}\mathbf{I}\Delta t^n} \ . \tag{334}$$

For the full IMU example, feeding the previous Σ_n into (316) yields the approximated transition matrix,

$$\Phi \approx \begin{bmatrix} \mathbf{I} & \mathbf{I}\Delta t & -\frac{1}{2}\mathbf{R}\left[\mathbf{a}\right]_\times \Delta t^2 & -\frac{1}{2}\mathbf{R}\Delta t^2 & \frac{1}{3!}\mathbf{R}\left[\mathbf{a}\right]_\times \Delta t^3 & \frac{1}{2}\mathbf{I}\Delta t^2 \\ 0 & \mathbf{I} & -\mathbf{R}\left[\mathbf{a}\right]_\times \Delta t & -\mathbf{R}\Delta t & \frac{1}{2}\mathbf{R}\left[\mathbf{a}\right]_\times \Delta t^2 & \mathbf{I}\Delta t \\ 0 & 0 & \mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top & 0 & -\mathbf{I}\Delta t & 0 \\ 0 & 0 & 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{I} \end{bmatrix} \tag{335}$$

with (see (173))

$$\mathbf{a} = \mathbf{a}_m - \mathbf{a}_b \ , \quad \boldsymbol{\omega} = \boldsymbol{\omega}_m - \boldsymbol{\omega}_b \ , \quad \mathbf{R} = \mathbf{R}\{\mathbf{q}\} \ ,$$

and where we have substituted the matrix blocks by their appropriate values (see also (173)),

$$\mathbf{P_v} = \mathbf{I} \ , \quad \mathbf{V}_{\boldsymbol{\theta}} = -\mathbf{R}\left[\mathbf{a}\right]_\times \ , \quad \mathbf{V_a} = -\mathbf{R} \ , \quad \mathbf{V_g} = \mathbf{I} \ , \quad \Theta_{\boldsymbol{\theta}} = -\left[\boldsymbol{\omega}\right]_\times \ , \quad \Theta_{\boldsymbol{\omega}} = -\mathbf{I}$$

A slight simplification of this method is to limit each block in the matrix to a certain maximum order n. For $n = 1$ we have,

$$\Phi \approx \begin{bmatrix} \mathbf{I} & \mathbf{I}\Delta t & 0 & 0 & 0 & 0 \\ 0 & \mathbf{I} & -\mathbf{R}\left[\mathbf{a}\right]_\times \Delta t & -\mathbf{R}\Delta t & 0 & \mathbf{I}\Delta t \\ 0 & 0 & \mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top & 0 & -\mathbf{I}\Delta t & 0 \\ 0 & 0 & 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{I} \end{bmatrix} , \tag{336}$$

which is the Euler method, whereas for $n = 2$,

$$\Phi \approx \begin{bmatrix} \mathbf{I} & \mathbf{I}\Delta t & -\frac{1}{2}\mathbf{R}\left[\mathbf{a}\right]_\times \Delta t^2 & -\frac{1}{2}\mathbf{R}\Delta t^2 & 0 & \frac{1}{2}\mathbf{I}\Delta t^2 \\ 0 & \mathbf{I} & -\mathbf{R}\left[\mathbf{a}\right]_\times \Delta t & -\mathbf{R}\Delta t & \frac{1}{2}\mathbf{R}\left[\mathbf{a}\right]_\times \Delta t^2 & \mathbf{I}\Delta t \\ 0 & 0 & \mathbf{R}\{\boldsymbol{\omega}\Delta t\}^\top & 0 & -\mathbf{I}\Delta t & 0 \\ 0 & 0 & 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{I} \end{bmatrix} . \tag{337}$$

For $n \geq 3$ we have the full form (335).

D The transition matrix via Runge-Kutta integration

Still another way to approximate the transition matrix is to use Runge-Kutta integration. This might be necessary in cases where the dynamic matrix $\mathbf{A}$ cannot be considered constant along the integration interval, *i.e.*,

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) \ . \tag{338}$$

Let us rewrite the following two relations defining the same system in continuous- and discrete-time. They involve the dynamic matrix $\mathbf{A}$ and the transition matrix Φ,

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t){\cdot}\mathbf{x}(t) \tag{339}$$

$$\mathbf{x}(t_n + \tau) = \Phi(t_n + \tau | t_n){\cdot}\mathbf{x}(t_n) \ . \tag{340}$$

These equations allow us to develop $\dot{\mathbf{x}}(t_n+\tau)$ in two ways as follows (left and right developments, please note the tiny dots indicating the time-derivatives),

$$\begin{aligned}
(\Phi(t_n+\tau|t_n)\mathbf{x}(t_n))^{\cdot} = & \boxed{\dot{\mathbf{x}}(t_n+\tau)} = \mathbf{A}(t_n+\tau)\mathbf{x}(t_n+\tau) \\
\dot{\Phi}(t_n+\tau|t_n)\mathbf{x}(t_n) + \Phi(t_n+\tau|t_n)\dot{\mathbf{x}}(t_n) = & \qquad\qquad = \mathbf{A}(t_n+\tau)\Phi(t_n+\tau|t_n)\mathbf{x}(t_n) \\
\dot{\Phi}(t_n+\tau|t_n)\mathbf{x}(t_n) = & \qquad\qquad (341)
\end{aligned}$$

Here, (341) comes from noticing that $\dot{\mathbf{x}}(t_n) = \dot{\mathbf{x}}_n = 0$, because it is a sampled value. Then,

$$\dot{\Phi}(t_n+\tau|t_n) = \mathbf{A}(t_n+\tau)\Phi(t_n+\tau|t_n) \tag{342}$$

which is the same ODE as (339), now applied to the transition matrix instead of the state vector. Mind that, because of the identity $\mathbf{x}(t_n) = \Phi_{t_n|t_n}\mathbf{x}(t_n)$, the transition matrix at the beginning of the interval, $t = t_n$, is always the identity,

$$\Phi_{t_n|t_n} = \mathbf{I}\ . \tag{343}$$

Using RK4 with $f(t, \Phi(t)) = \mathbf{A}(t)\Phi(t)$, we have

$$\Phi \triangleq \Phi(t_n+\Delta t|t_n) = \mathbf{I} + \frac{\Delta t}{6}(\mathbf{K}_1 + 2\mathbf{K}_2 + 2\mathbf{K}_3 + \mathbf{K}_4) \tag{344}$$

with

$$\mathbf{K}_1 = \mathbf{A}(t_n) \tag{345}$$

$$\mathbf{K}_2 = \mathbf{A}\Big(t_n + \frac{1}{2}\Delta t\Big)\Big(\mathbf{I} + \frac{\Delta t}{2}\mathbf{K}_1\Big) \tag{346}$$

$$\mathbf{K}_3 = \mathbf{A}\Big(t_n + \frac{1}{2}\Delta t\Big)\Big(\mathbf{I} + \frac{\Delta t}{2}\mathbf{K}_2\Big) \tag{347}$$

$$\mathbf{K}_4 = \mathbf{A}\Big(t_n + \Delta t\Big)\Big(\mathbf{I} + \Delta t \cdot \mathbf{K}_3\Big)\ . \tag{348}$$

D.1 Error-state example

Let us consider the error-state Kalman filter for the non-linear, time-varying system

$$\dot{\mathbf{x}}_t(t) = f(t, \mathbf{x}_t(t), \mathbf{u}(t)) \tag{349}$$

where $\mathbf{x}_t$ denotes the true state, and $\mathbf{u}$ is a control input. This true state is a composition, denoted by $\oplus$, of a nominal state $\mathbf{x}$ and the error state $\delta\mathbf{x}$,

$$\mathbf{x}_t(t) = \mathbf{x}(t) \oplus \delta\mathbf{x}(t) \tag{350}$$

where the error-state dynamics admits a linear form which is time-varying depending on the nominal state $\mathbf{x}$ and the control $\mathbf{u}$, *i.e.*,

$$\dot{\delta\mathbf{x}} = \mathbf{A}(\mathbf{x}(t), \mathbf{u}(t)) \cdot \delta\mathbf{x} \tag{351}$$

that is, the error-state dynamic matrix in (338) has the form $\mathbf{A}(t) = \mathbf{A}(\mathbf{x}(t), \mathbf{u}(t))$. The dynamics of the error-state transition matrix can be written,

$$\dot{\Phi}(t_n + \tau | t_n) = \mathbf{A}(\mathbf{x}(t), \mathbf{u}(t)) \cdot \Phi(t_n + \tau | t_n) \ . \tag{352}$$

In order to RK-integrate this equation, we need the values of $\mathbf{x}(t)$ and $\mathbf{u}(t)$ at the RK evaluation points, which for RK4 are $\{t_n, t_n + \Delta t/2, t_n + \Delta t\}$. Starting by the easy ones, the control inputs $\mathbf{u}(t)$ at the evaluation points can be obtained by linear interpolation of the current and last measurements,

$$\mathbf{u}(t_n) = \mathbf{u}_n \tag{353}$$

$$\mathbf{u}(t_n + \Delta t/2) = \frac{\mathbf{u}_n + \mathbf{u}_{n+1}}{2} \tag{354}$$

$$\mathbf{u}(t_n + \Delta t) = \mathbf{u}_{n+1} \tag{355}$$

The nominal state dynamics should be integrated using the best integration practicable. For example, using RK4 integration we have,

$$\begin{aligned}
\mathbf{k}_1 &= f(\mathbf{x}_n, \mathbf{u}_n) \\
\mathbf{k}_2 &= f(\mathbf{x}_n + \frac{\Delta t}{2}\mathbf{k}_1, \frac{\mathbf{u}_n + \mathbf{u}_{n+1}}{2}) \\
\mathbf{k}_3 &= f(\mathbf{x}_n + \frac{\Delta t}{2}\mathbf{k}_2, \frac{\mathbf{u}_n + \mathbf{u}_{n+1}}{2}) \\
\mathbf{k}_4 &= f(\mathbf{x}_n + \Delta t \mathbf{k}_3, \mathbf{u}_{n+1}) \\
\mathbf{k} &= (\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)/6 \ ,
\end{aligned}$$

which gives us the estimates at the evaluation points,

$$\mathbf{x}(t_n) = \mathbf{x}_n \tag{356}$$

$$\mathbf{x}(t_n + \Delta t/2) = \mathbf{x}_n + \frac{\Delta t}{2}\mathbf{k} \tag{357}$$

$$\mathbf{x}(t_n + \Delta t) = \mathbf{x}_n + \Delta t \, \mathbf{k} \ . \tag{358}$$

We notice here that $\mathbf{x}(t_n + \Delta t/2) = \frac{\mathbf{x}_n + \mathbf{x}_{n+1}}{2}$, the same linear interpolation we used for the control. This should not be surprising given the linear nature of the RK update.

Whichever the way we obtained the nominal state values, we can now compute the RK4 matrices for the integration of the transition matrix,

$$\begin{aligned}
\mathbf{K}_1 &= \mathbf{A}(\mathbf{x}_n, \mathbf{u}_n) \\
\mathbf{K}_2 &= \mathbf{A}\Big(\mathbf{x}_n + \frac{\Delta t}{2}\mathbf{k}, \frac{\mathbf{u}_n + \mathbf{u}_{n+1}}{2}\Big)\Big(\mathbf{I} + \frac{\Delta t}{2}\mathbf{K}_1\Big) \\
\mathbf{K}_3 &= \mathbf{A}\Big(\mathbf{x}_n + \frac{\Delta t}{2}\mathbf{k}, \frac{\mathbf{u}_n + \mathbf{u}_{n+1}}{2}\Big)\Big(\mathbf{I} + \frac{\Delta t}{2}\mathbf{K}_2\Big) \\
\mathbf{K}_4 &= \mathbf{A}\Big(\mathbf{x}_n + \Delta t \mathbf{k}, \mathbf{u}_{n+1}\Big)\Big(\mathbf{I} + \Delta t \mathbf{K}_3\Big) \\
\mathbf{K} &= (\mathbf{K}_1 + 2\mathbf{K}_2 + 2\mathbf{K}_3 + \mathbf{K}_4)/6
\end{aligned}$$

which finally lead to,

$$\boxed{\Phi \triangleq \Phi_{t_n+\Delta t|t_n} = \mathbf{I} + \Delta t\, \mathbf{K}} \tag{359}$$

E Integration of random noise and perturbations

We aim now at giving appropriate methods for the integration of random variables within dynamic systems. Of course, we cannot integrate unknown random values, but we can integrate their variances and covariances for the sake of uncertainty propagation. This is needed in order to establish the covariances matrices in estimators for systems that are of continuous nature (and specified in continuous time) but estimated in a discrete manner.

Consider the continuous-time dynamic system,

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{w}) \ , \tag{360}$$

where $\mathbf{x}$ is the state vector, $\mathbf{u}$ is a vector of control signals containing noise $\tilde{\mathbf{u}}$, so that the control measurements are $\mathbf{u}_m = \mathbf{u} + \tilde{\mathbf{u}}$, and $\mathbf{w}$ is a vector of random perturbations. Both noise and perturbations are assumed white Gaussian processes, specified by,

$$\tilde{\mathbf{u}} \sim \mathcal{N}\{0, \mathbf{U}^c\} \quad , \quad \mathbf{w}^c \sim \mathcal{N}\{0, \mathbf{W}^c\} \ , \tag{361}$$

where the super-index $\bullet^c$ indicates a continuous-time uncertainty specification, which we want to integrate.

There exists an important difference between the natures of the noise levels in the control signals, $\tilde{\mathbf{u}}$, and the random perturbations, $\mathbf{w}$:

- On discretization, the control signals are sampled at the time instants $n\Delta t$, having $\mathbf{u}_{m,n} \triangleq \mathbf{u}_m(n\Delta t) = \mathbf{u}(n\Delta t) + \tilde{\mathbf{u}}(n\Delta t)$. The measured part is obviously considered constant over the integration interval, *i.e.*, $\mathbf{u}_m(t) = \mathbf{u}_{m,n}$, and therefore the noise level at the sampling time $n\Delta t$ is also held constant,

$$\tilde{\mathbf{u}}(t) = \tilde{\mathbf{u}}(n\Delta t) = \tilde{\mathbf{u}}_n, \quad n\Delta t < t < (n+1)\Delta t \ . \tag{362}$$

- The perturbations $\mathbf{w}$ are never sampled.

As a consequence, the integration over Δt of these two stochastic processes differs. Let us examine it.

The continuous-time error-state dynamics (360) can be linearized to

$$\dot{\delta\mathbf{x}} = \mathbf{A}\delta\mathbf{x} + \mathbf{B}\tilde{\mathbf{u}} + \mathbf{C}\mathbf{w} \ , \tag{363}$$

with

$$\mathbf{A} \triangleq \left.\frac{\partial f}{\partial \delta\mathbf{x}}\right|_{\mathbf{x},\mathbf{u}_m} \quad , \quad \mathbf{B} \triangleq \left.\frac{\partial f}{\partial \tilde{\mathbf{u}}}\right|_{\mathbf{x},\mathbf{u}_m} \quad , \quad \mathbf{C} \triangleq \left.\frac{\partial f}{\partial \mathbf{w}}\right|_{\mathbf{x},\mathbf{u}_m} \ , \tag{364}$$

and integrated over the sampling period Δt, giving,

$$\delta \mathbf{x}_{n+1} = \delta \mathbf{x}_n + \int_{n\Delta t}^{(n+1)\Delta t} \left(\mathbf{A}\delta\mathbf{x}(\tau) + \mathbf{B}\tilde{\mathbf{u}}(\tau) + \mathbf{C}\mathbf{w}^c(\tau) \right) d\tau \tag{365}$$

$$= \delta \mathbf{x}_n + \int_{n\Delta t}^{(n+1)\Delta t} \mathbf{A}\delta\mathbf{x}(\tau) d\tau + \int_{n\Delta t}^{(n+1)\Delta t} \mathbf{B}\tilde{\mathbf{u}}(\tau) d\tau + \int_{n\Delta t}^{(n+1)\Delta t} \mathbf{C}\mathbf{w}^c(\tau) d\tau \tag{366}$$

which has three terms of very different nature. They can be integrated as follows:

1. From App. B we know that the dynamic part is integrated giving the transition matrix,

$$\delta \mathbf{x}_n + \int_{n\Delta t}^{(n+1)\Delta t} \mathbf{A}\delta\mathbf{x}(\tau) d\tau = \Phi \cdot \delta \mathbf{x}_n \tag{367}$$

   where $\Phi = e^{\mathbf{A}\Delta t}$ can be computed in closed-form or approximated at different levels of accuracy.

2. From (362) we have

$$\int_{n\Delta t}^{(n+1)\Delta t} \mathbf{B}\tilde{\mathbf{u}}(\tau) d\tau = \mathbf{B}\Delta t \tilde{\mathbf{u}}_n \tag{368}$$

   which means that the measurement noise, once sampled, is integrated in a deterministic manner because its behavior inside the integration interval is known.

3. From Probability Theory we know that the integration of continuous white Gaussian noise over a period Δt produces a discrete white Gaussian impulse $\mathbf{w}_n$ described by

$$\mathbf{w}_n \triangleq \int_{n\Delta t}^{(n+1)\Delta t} \mathbf{w}(\tau) d\tau \quad , \quad \mathbf{w}_n \sim \mathcal{N}\{0, \mathbf{W}\} \quad , \quad \text{with } \mathbf{W} = \mathbf{W}^c \Delta t \tag{369}$$

   We obsereve that, contrary to the measurement noise just above, the perturbation does not have a deterministic behavior inside the integration interval, and hence it must be integrated stochastically.

Therefore, the discrete-time, error-state dynamic system can be written as

$$\delta \mathbf{x}_{n+1} = \mathbf{F}_\mathbf{x} \delta \mathbf{x}_n + \mathbf{F}_\mathbf{u} \tilde{\mathbf{u}}_n + \mathbf{F}_\mathbf{w} \mathbf{w}_n \tag{370}$$

with transition, control and perturbation matrices given by

$$\mathbf{F}_\mathbf{x} = \Phi = e^{\mathbf{A}\Delta t} \quad , \quad \mathbf{F}_\mathbf{u} = \mathbf{B}\Delta t \quad , \quad \mathbf{F}_\mathbf{w} = \mathbf{C} \quad , \tag{371}$$

with noise and perturbation levels defined by

$$\tilde{\mathbf{u}}_n \sim \mathcal{N}\{0, \mathbf{U}\} \quad , \quad \mathbf{w}_n \sim \mathcal{N}\{0, \mathbf{W}\} \tag{372}$$

with

$$\mathbf{U} = \mathbf{U}^c \quad , \quad \mathbf{W} = \mathbf{W}^c \Delta t \; . \tag{373}$$

Table 4: Effect of integration on system and covariances matrices.

Description	Continuous time t	Discrete time $n\Delta t$
state	$\dot{\mathbf{x}} = f^c(\mathbf{x}, \mathbf{u}, \mathbf{w})$	$\mathbf{x}_{n+1} = f(\mathbf{x}_n, \mathbf{u}_n, \mathbf{w}_n)$
error-state	$\dot{\delta\mathbf{x}} = \mathbf{A}\delta\mathbf{x} + \mathbf{B}\tilde{\mathbf{u}} + \mathbf{C}\mathbf{w}$	$\delta\mathbf{x}_{n+1} = \mathbf{F_x}\delta\mathbf{x}_n + \mathbf{F_u}\tilde{\mathbf{u}}_n + \mathbf{F_w}\mathbf{w}_n$
system matrix	$\mathbf{A}$	$\mathbf{F_x} = \Phi = e^{\mathbf{A}\Delta t}$
control matrix	$\mathbf{B}$	$\mathbf{F_u} = \mathbf{B}\Delta t$
perturbation matrix	$\mathbf{C}$	$\mathbf{F_w} = \mathbf{C}$
control covariance	$\mathbf{U}^c$	$\mathbf{U} = \mathbf{U}^c$
perturbation covariance	$\mathbf{W}^c$	$\mathbf{W} = \mathbf{W}^c\Delta t$

These results are summarized in Table 4. The prediction stage of an EKF would propagate the error state's mean and covariances matrix according to

$$\hat{\delta\mathbf{x}}_{n+1} = \mathbf{F_x}\hat{\delta\mathbf{x}}_n \tag{374}$$

$$\begin{aligned}\mathbf{P}_{n+1} &= \mathbf{F_x}\mathbf{P}_n\mathbf{F_x}^\top + \mathbf{F_u}\mathbf{U}\mathbf{F_u}^\top + \mathbf{F_w}\mathbf{W}\mathbf{F_w}^\top \\ &= e^{\mathbf{A}\Delta t}\mathbf{P}_n(e^{\mathbf{A}\Delta t})^\top + \Delta t^2\mathbf{B}\mathbf{U}^c\mathbf{B}^\top + \Delta t\mathbf{C}\mathbf{W}^c\mathbf{C}^\top\end{aligned} \tag{375}$$

It is important and illustrative here to observe the different effects of the integration interval, Δt, on the three terms of the covariance update (375): the dynamic error term is exponential, the measurement error term is quadratic, and the perturbation error term is linear.

E.1 Noise and perturbation impulses

One is oftentimes confronted (for example when reusing existing code or when interpreting other authors' documents) with EKF prediction equations of a simpler form than those that we used here, namely,

$$\mathbf{P}_{n+1} = \mathbf{F_x}\mathbf{P}_n\mathbf{F_x}^\top + \mathbf{Q}\ . \tag{376}$$

This corresponds to the general discrete-time dynamic system,

$$\delta\mathbf{x}_{n+1} = \mathbf{F_x}\delta\mathbf{x}_n + \mathbf{i} \tag{377}$$

where

$$\mathbf{i} \sim \mathcal{N}\{0, \mathbf{Q}\} \tag{378}$$

is a vector of random (white, Gaussian) impulses that are directly added to the state vector at time t_{n+1}. The matrix $\mathbf{Q}$ is simply considered the impulses covariances matrix. From what we have seen, we should compute this covariances matrix as follows,

$$\mathbf{Q} = \Delta t^2\,\mathbf{B}\,\mathbf{U}^c\,\mathbf{B}^\top + \Delta t\,\mathbf{C}\,\mathbf{W}^c\,\mathbf{C}^\top\ . \tag{379}$$

In the case where the impulses do not affect the full state, as it is often the case, the matrix $\mathbf{Q}$ is not full-diagonal and may contain a significant amount of zeros. One can then write the equivalent form

$$\delta\mathbf{x}_{n+1} = \mathbf{F}_\mathbf{x}\,\delta\mathbf{x}_n + \mathbf{F}_\mathbf{i}\,\mathbf{i} \tag{380}$$

with

$$\mathbf{i} \sim \mathcal{N}\{0, \mathbf{Q}_\mathbf{i}\}\ , \tag{381}$$

where the matrix $\mathbf{F}_\mathbf{i}$ simply maps each individual impulse to the part of the state vector it affects to. The associated covariance $\mathbf{Q}_\mathbf{i}$ is then smaller and full-diagonal. Please refer to the next section for an example. In such case the ESKF time-update becomes

$$\hat{\delta\mathbf{x}}_{n+1} = \mathbf{F}_\mathbf{x}\,\hat{\delta\mathbf{x}}_n \tag{382}$$

$$\mathbf{P}_{n+1} = \mathbf{F}_\mathbf{x}\,\mathbf{P}_n\,\mathbf{F}_\mathbf{x}^\top + \mathbf{F}_\mathbf{i}\,\mathbf{Q}_\mathbf{i}\,\mathbf{F}_\mathbf{i}^\top\ . \tag{383}$$

Obviously, all these forms are equivalent, as it can be seen in the following double identity for the general perturbation $\mathbf{Q}$,

$$\mathbf{F}_\mathbf{i}\,\mathbf{Q}_\mathbf{i}\,\mathbf{F}_\mathbf{i}^\top = \boxed{\mathbf{Q}} = \Delta t^2\,\mathbf{B}\,\mathbf{U}^c\,\mathbf{B}^\top + \Delta t\,\mathbf{C}\,\mathbf{W}^c\,\mathbf{C}^\top\ . \tag{384}$$

E.2 Full IMU example

We study the construction of an error-state Kalman filter for an IMU. The error-state system is defined in (173) and involves a nominal state $\mathbf{x}$, an error-state $\delta\mathbf{x}$, a noisy control signal $\mathbf{u}_m = \mathbf{u} + \tilde{\mathbf{u}}$ and a perturbation $\mathbf{w}$, specified by,

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \\ \mathbf{q} \\ \mathbf{a}_b \\ \boldsymbol{\omega}_b \\ \mathbf{g} \end{bmatrix}\ ,\quad \delta\mathbf{x} = \begin{bmatrix} \delta\mathbf{p} \\ \delta\mathbf{v} \\ \delta\boldsymbol{\theta} \\ \delta\mathbf{a}_b \\ \delta\boldsymbol{\omega}_b \\ \delta\mathbf{g} \end{bmatrix}\ ,\quad \mathbf{u}_m = \begin{bmatrix} \mathbf{a}_m \\ \boldsymbol{\omega}_m \end{bmatrix}\ ,\quad \tilde{\mathbf{u}} = \begin{bmatrix} \tilde{\mathbf{a}} \\ \tilde{\boldsymbol{\omega}} \end{bmatrix}\ ,\quad \mathbf{w} = \begin{bmatrix} \mathbf{a}_w \\ \boldsymbol{\omega}_w \end{bmatrix} \tag{385}$$

In a model of an IMU like the one we are considering throughout this document, the control noise corresponds to the additive noise in the IMU measurements. The perturbations affect the biases, thus producing their random-walk behavior. The dynamic, control and perturbation matrices are (see (363), (313) and (173)),

$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{P}_\mathbf{v} & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{V}_{\boldsymbol{\theta}} & \mathbf{V}_\mathbf{a} & 0 & \mathbf{V}_\mathbf{g} \\ 0 & 0 & \Theta_{\boldsymbol{\theta}} & 0 & \Theta_{\boldsymbol{\omega}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\ ,\quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ -\mathbf{R} & 0 \\ 0 & -\mathbf{I} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}\ ,\quad \mathbf{C} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \mathbf{I} & 0 \\ 0 & \mathbf{I} \\ 0 & 0 \end{bmatrix} \tag{386}$$

In the regular case of IMUs with accelerometer and gyrometer triplets of the same kind on the three axes, noise and perturbations are isotropic. Their standard deviations are specified as scalars as follows

$$\sigma_{\tilde{\mathbf{a}}}\ [m/s^2] \quad , \quad \sigma_{\tilde{\boldsymbol{\omega}}}\ [rad/s] \quad , \quad \sigma_{\mathbf{a}_w}\ [m/s^2\sqrt{s}] \quad , \quad \sigma_{\boldsymbol{\omega}_w}\ [rad/s\sqrt{s}] \tag{387}$$

and their covariances matrices are purely diagonal, giving

$$\mathbf{U}^c = \begin{bmatrix} \sigma_{\tilde{\mathbf{a}}}^2\mathbf{I} & 0 \\ 0 & \sigma_{\tilde{\boldsymbol{\omega}}}^2\mathbf{I} \end{bmatrix} \quad , \quad \mathbf{W}^c = \begin{bmatrix} \sigma_{\mathbf{a}_w}^2\mathbf{I} & 0 \\ 0 & \sigma_{\boldsymbol{\omega}_w}^2\mathbf{I} \end{bmatrix} . \tag{388}$$

The system evolves with sampled measures at intervals Δt, following (370–373), where the transition matrix $\mathbf{F}_\mathbf{x} = \Phi$ can be computed in a number of ways – see previous appendices.

E.2.1 Noise and perturbation impulses

In the case of a perturbation specification in the form of impulses $\mathbf{i}$, we can re-define our system as follows,

$$\delta\mathbf{x}_{n+1} = \mathbf{F}_\mathbf{x}(\mathbf{x}_n, \mathbf{u}_m)\cdot\delta\mathbf{x}_n + \mathbf{F}_\mathbf{i}\cdot\mathbf{i} \tag{389}$$

with the nominal-state, error-state, control, and impulses vectors defined by,

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \\ \mathbf{q} \\ \mathbf{a}_b \\ \boldsymbol{\omega}_b \\ \mathbf{g} \end{bmatrix} \quad , \quad \delta\mathbf{x} = \begin{bmatrix} \delta\mathbf{p} \\ \delta\mathbf{v} \\ \delta\boldsymbol{\theta} \\ \delta\mathbf{a}_b \\ \delta\boldsymbol{\omega}_b \\ \delta\mathbf{g} \end{bmatrix} \quad , \quad \mathbf{u}_m = \begin{bmatrix} \mathbf{a}_m \\ \boldsymbol{\omega}_m \end{bmatrix} \quad , \quad \mathbf{i} = \begin{bmatrix} \mathbf{v}_\mathbf{i} \\ \boldsymbol{\theta}_\mathbf{i} \\ \mathbf{a}_\mathbf{i} \\ \boldsymbol{\omega}_\mathbf{i} \end{bmatrix} , \tag{390}$$

the transition and perturbations matrices defined by,

$$\mathbf{F}_\mathbf{x} = \Phi = e^{\mathbf{A}\Delta t} \quad , \quad \mathbf{F}_\mathbf{i} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \mathbf{I} & 0 & 0 & 0 \\ 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \mathbf{I} \\ 0 & 0 & 0 & 0 \end{bmatrix} , \tag{391}$$

and the impulses variances specified by

$$\mathbf{i} \sim \mathcal{N}\{0, \mathbf{Q}_\mathbf{i}\} \quad , \quad \mathbf{Q}_\mathbf{i} = \begin{bmatrix} \sigma_{\tilde{\mathbf{a}}}^2\Delta t^2\mathbf{I} & & & 0 \\ & \sigma_{\tilde{\boldsymbol{\omega}}}^2\Delta t^2\mathbf{I} & & \\ & & \sigma_{\mathbf{a}_w}^2\Delta t\mathbf{I} & \\ 0 & & & \sigma_{\boldsymbol{\omega}_w}^2\Delta t\mathbf{I} \end{bmatrix} . \tag{392}$$

The trivial specification of $\mathbf{F_i}$ may appear surprising given especially that of $\mathbf{B}$ in (386). What happens is that the errors are defined isotropic in $\mathbf{Q_i}$, and therefore $-\mathbf{R}\sigma^2\mathbf{I}(-\mathbf{R})^\top = \sigma^2\mathbf{I}$ and $-\mathbf{I}\sigma^2\mathbf{I}(-\mathbf{I})^\top = \sigma^2\mathbf{I}$, leading to the expression given for $\mathbf{F_i}$. This is not possible when considering non-isotropic IMUs, where a proper Jacobian $\mathbf{F_i} = \begin{bmatrix}\mathbf{B} & \mathbf{C}\end{bmatrix}$ should be used together with a proper specification of $\mathbf{Q_i}$.

We can of course use full-state perturbation impulses,

$$\delta\mathbf{x}_{n+1} = \mathbf{F_x}(\mathbf{x}_n, \mathbf{u}_m)\cdot\delta\mathbf{x}_n + \mathbf{i} \tag{393}$$

with

$$\mathbf{i} = \begin{bmatrix} 0 \\ \mathbf{v_i} \\ \boldsymbol{\theta}_\mathbf{i} \\ \mathbf{a_i} \\ \boldsymbol{\omega}_\mathbf{i} \\ 0 \end{bmatrix} \quad , \quad \mathbf{i} \sim \mathcal{N}\{0, \mathbf{Q}\} \quad , \quad \mathbf{Q} = \begin{bmatrix} 0 & & & & & \\ & \sigma^2_{\tilde{\mathbf{a}}}\Delta t^2\mathbf{I} & & & 0 & \\ & & \sigma^2_{\tilde{\boldsymbol{\omega}}}\Delta t^2\mathbf{I} & & & \\ & & & \sigma^2_{\mathbf{a}_w}\Delta t\mathbf{I} & & \\ & 0 & & & \sigma^2_{\boldsymbol{\omega}_w}\Delta t\mathbf{I} & \\ & & & & & 0 \end{bmatrix} . \tag{394}$$

Bye bye.

References

Chou, J. (1992). Quaternion kinematic and dynamic differential equations. *Robotics and Automation, IEEE Transactions on*, 8(1):53–64.

Kuipers, J. B. (1999). *Quaternions and rotation sequences: a primer with applications to orbits, aerospace, and virtual reality*. Princeton University Press., Princeton, N.J.

Li, M. and Mourikis, A. (2012). Improving the accuracy of EKF-based visual-inertial odometry. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 828–835.

Li, M., Yu, H., Zheng, X., and Mourikis, A. (2014). High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 409–416.

Lupton, T. and Sukkarieh, S. (2009). Efficient integration of inertial observations into visual slam without initialization. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*.

Madyastha, V., Ravindra, V., Mallikarjunan, S., and Goyal, A. (2011). Extended Kalman Filter vs. Error State Kalman Filter for Aircraft Attitude Estimation. In *AIAA Guidance, Navigation, and Control Conference*, Portland, Oregon. American Institute of Aeronautics and Astronautics.

Martinelli, A. (2012). Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *IEEE Transactions on Robotics*, 28(1):44 –60.

Piniés, P., Lupton, T., Sukkarieh, S., and Tardós, J. D. (2007). Inertial aiding of inverse depth SLAM using a monocular camera. In *Int. Conf. on Robotics and Automation*.

Roussillon, C., Gonzalez, A., Solà, J., Codol, J. M., Mansard, N., Lacroix, S., and Devy, M. (2011). RT-SLAM: A generic and real-time visual SLAM implementation. In Crowley, J., Draper, B., and Thonnat, M., editors, *Computer Vision Systems*, volume 6962 of *Lecture Notes in Computer Science*, pages 31–40. Springer Berlin Heidelberg.

Shuster, M. D. (1993). Survey of attitude representations. *Journal of the Astronautical Sciences*, 41:439–517.

Trawny, N. and Roumeliotis, S. I. (2005). Indirect Kalman filter for 3D attitude estimation. Technical Report 2005-002, University of Minnesota, Dept. of Comp. Sci. & Eng.

Yazell, D. (2009). Origins of the unusual space shuttle quaternion definition. In *Aerospace Sciences Meetings*, pages –. American Institute of Aeronautics and Astronautics.